	Ingeniería electrónica		
	Proyectos computacionales aplicados a ingeniería electrónica		
	Ing. José Anibal Silva de Los Angeles		
	Aux. Fernando Mardoqueo Paxtor Sam		
Carné:	201700923	Fecha:	9/3/2022
Nombre:	Héctor Fernando Carrera Soto		
Registro admin:	3505043180101		

1. Resultados

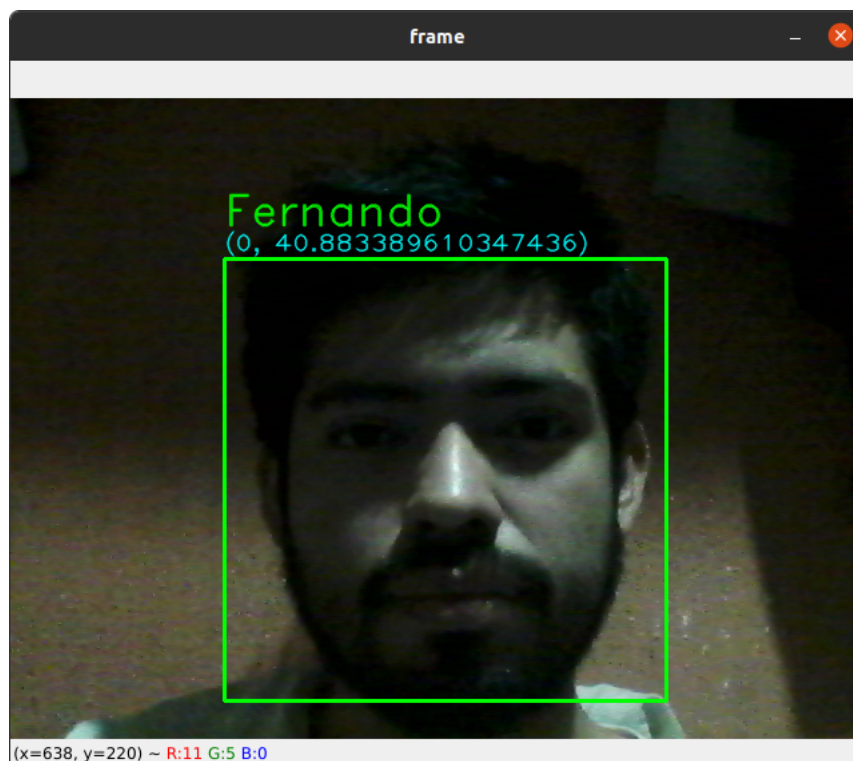


Figura 1: Programa de reconocimiento de rostros, usando opencv.

Fuente: Elaboración propia.

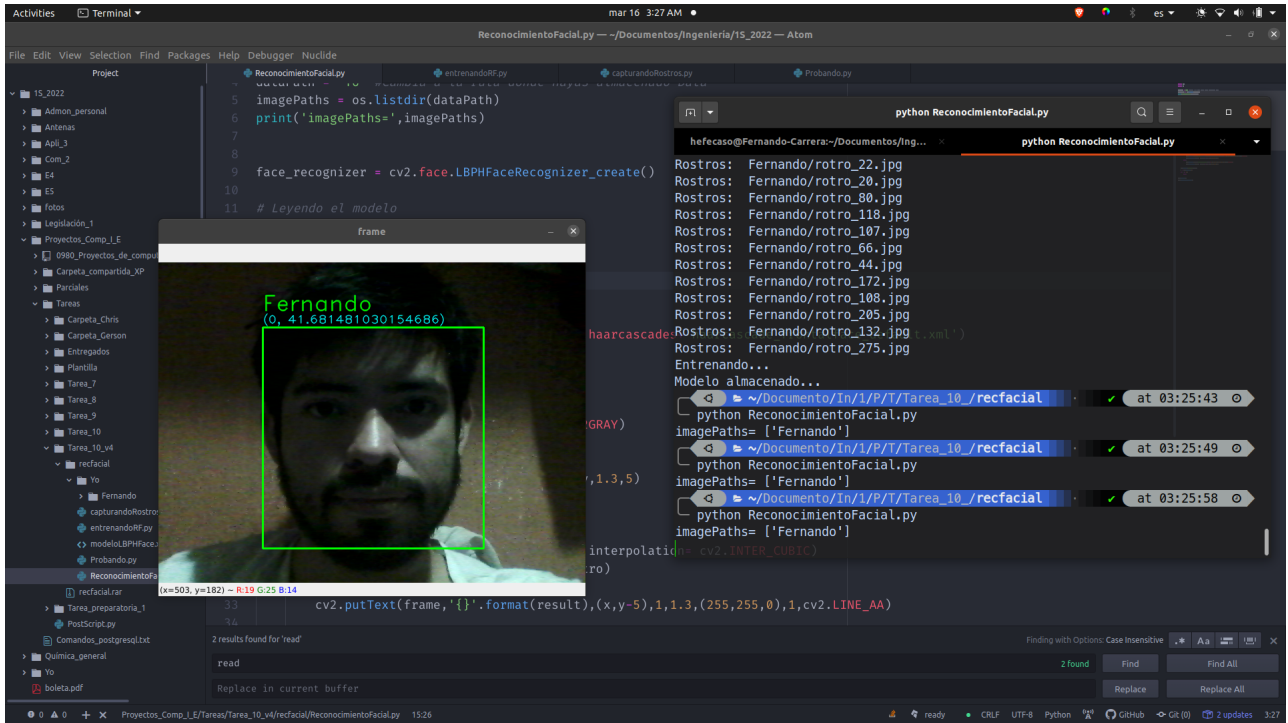


Figura 2: Programa de reconocimiento de rostros, usando opencv.
Fuente: Elaboración propia.

2. Observaciones para el uso en UBUNTU

1. Modificar "cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)" por "cap = cv2.VideoCapture(-1)"
2. Dar permisos de administrador a la cámara usando el siguiente comando en la terminal
"\$ sudo chmod 777 /dev/video*"
3. Verificar que aparezca entre los permisos el acceso a la cámara con
"\$ id"
4. Instalar los siguientes complementos para openCV desde ubuntu:
"\$ python -m pip install --user opencv-contrib-python"

3. Código para captura de datos biométricos

```
import cv2
import os
import imutils

personName = 'Fernando'
dataPath = 'Yo'#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(-1)
#cap = cv2.VideoCapture('Video.mp4')

faceClassif = cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_defau
count = 0

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)

    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 + '/roto_{}.jpg'.format(count),rostro)
        count = count + 1
    cv2.imshow('frame',frame)

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

cap.release()
cv2.destroyAllWindows()
```

4. Código para entrenamiento biométrico

```
import cv2
import os
import numpy as np

dataPath = 'Yo' #Cambia a la ruta donde hayas almacenado Data
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

labels = []
facesData = []
label = 0

for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName,0))

    label = label + 1

# 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))

# Almacenando el modelo obtenido

face_recognizer.write('modeloLBPHFace.xml')
print("Modelo almacenado...")
```

5. Código de programa de reconocimiento biométrico

```
import cv2
import os

dataPath = 'Yo' #Cambia a la ruta donde hayas almacenado Data
imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)

face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Leyendo el modelo

face_recognizer.read('modeloLBPHFace.xml')

cap = cv2.VideoCapture(-1)
#cap = cv2.VideoCapture('Video.mp4')

faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')

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)

    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)

        cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)

    # LBPHFace
    if result[1] < 70:
        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()
```