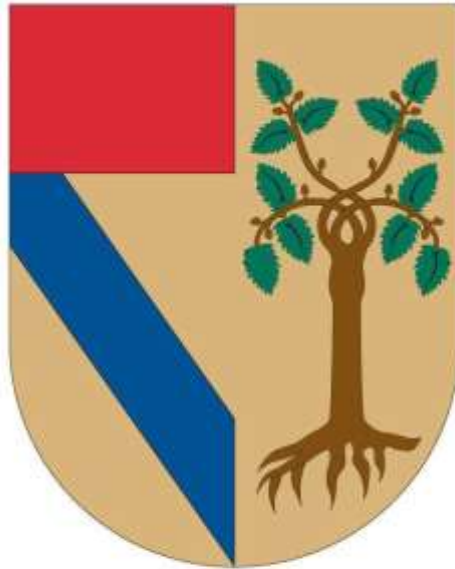


REPORTE PROYECTO FINAL
RECONOCIMIENTO FACIAL



UNIVERSIDAD
PANAMERICANA

ROBOTICA INDUSTRIAL
PYTHON & OPEN CV
JESÚS MARIO DAVIZON LÓPEZ

A 18 de Mayo del 2020



RECONOCIMIENTO FACIAL

El proyecto consiste en realizar reconocimiento facial en tiempo real utilizando la cámara web de la computadora. Para esto se utilizaron diversas librerías de open cv que fueron implementadas en python.

Para lograr ejecutar este proyecto de manera exitosa es importante llevar a cabo los siguientes pasos:


1. En primer lugar es necesario instalar OpenCV. Gracias a que se trabaja con código abierto es posible utilizar las librerías y los diferentes algoritmos implementados para adaptarlos a nuestras necesidades. La instalación la podemos realizar ejecutando las siguientes líneas de código:

```
pip install opencv-python  
pip install opencv-contrib-python
```

2. Posteriormente es importante hacer uso de los siguientes archivos, ya que son parte fundamental de la ejecución del proyecto:

 [main.ipynb](#)

 [audio.ipynb](#)

 [dataset.ipynb](#)

 [detectar_identificar_rostros.ipynb](#)

 [entrenamiento.ipynb](#)

3. Explicaciones:

a) [main.ipynb](#)

- Con este script se inicia la ejecución del proyecto. Al inicializarlo la cámara se ejecuta y empieza la detección de rostros mostrando “Desconocido” si no reconoce el rostro o mostrando el nombre encima del rostro en caso de reconocerlo. Si se desea agregar un nuevo rostro a reconocer se pasa a la ejecución del script [audio.ipynb](#)

b) [audio.ipynb](#)

- Para poder ejecutar de manera correcta este script es necesario realizar la instalación de SpeechRecognition:

```
!pip install SpeechRecognition
```



- En caso de tener problemas con la instalación, ver el siguiente tutorial donde se explica como resolver dichos problemas. Esto se debe a que en algunas ocasiones suele haber problemas con la ejecución de pyaudio. El video agrega la dirección del repositorio donde están los files requeridos para la instalación.

<https://www.youtube.com/watch?v=XksalwvNckU&list=LLat-eo-IOvEQHvPHIqLNHOA&index=10>

- Este script pide al usuario el nombre a guardar, de manera que el usuario por medio del micrófono dice el nombre, mismo que es guardado en una variable de texto. Posteriormente se crea un archivo de texto en el que se agrega el valor previamente guardado (correspondiente al nombre) para así ser utilizado en el script del set de datos. En caso de que no sea claro lo que dice el usuario, el código seguirá pidiendo el nombre al usuario hasta que pueda reconocerlo. Una vez finalizado, se ejecuta el script data.set

```
text = r.recognize_google(audio) # language='pt-BR'
print("El nombre guardado es: {}".format(text, 'es-ES'))

path = folder + text
os.mkdir(path)

archivo_texto=open("archivo.txt", "w")
archivo_texto.write(text)
archivo_texto.close()
```

Fig 1. Reconocimiento de audio y creación de archivo .txt donde se escribe el nombre guardado.

c) dataset.ipynb

- Este script de Python nos permite realizar el set de datos que posteriormente servirá para la detección de rostros. En primer lugar ejecuta la cámara para tomar fotografías al rostro detectado y así crear una carpeta con el nombre asignado para que después, en caso de reconocer ese nombre, le asigne el nombre por medio de una etiqueta. Actualmente el código realiza 400 capturas, este dato puede variarse según el número de capturas que se desee (ver fig.2)

```
elif count >= 400:
    break
```

Fig 2. Variable count que representa el número de capturas realizadas al nuevo rostro detectado.



- Una vez realizadas las capturas, el script lee el archivo .txt donde se encuentra guardado el nombre dicho por el usuario, de manera que se crea una nueva carpeta con dicho nombre donde serán guardadas las capturas correspondientes a esa persona. La lectura del archivo de texto así como la creación de la carpeta con las imágenes se observa en las fig. 3 y 4

```
archivo_texto=open("archivo.txt","r")
texto=archivo_texto.read()
archivo_texto.close()

folder='images/'
name= texto + '/' + texto
path = folder + name
```

Fig 3. Se abre archivo .txt para leer el valor guardado y asignarlo a la carpeta referenciando el nombre.

```
cv2.imwrite(path+str(count)+".jpg", grises[y:y+h, x:x+w])
cv2.imshow("Creando Dataset", imagen_marco)
```

Fig 4. Se crea carpeta con nombre leído en archivo .txt y se guardan capturas.

- Una vez realizado el set de datos, la cámara web se finaliza y se corre el script de entrenamiento (ver fig. 5)

```
web_cam.release()
cv2.destroyAllWindows()
%run entrenamiento.ipynb
```

Fig 5. Finalización de cámara web e inicio de ejecución de script de entrenamiento

d) entrenamiento.ipynb

- Este script carga el archivo .xml correspondiente al tipo de detección (en este caso se utilizó detección de rostros frontal) y ejecuta el método de reconocimiento a partir de un entrenamiento. De igual manera, se generan las etiquetas de los nombres existentes y los asigna a los tipos de rostros guardados para el entrenamiento. Finalmente guarda el entrenamiento en el archivo entrenamiento.yml y ejecuta el script de detección de rostros.



```
reconocimiento.train(x_entrenamiento, np.array(y_etiquetas))
reconocimiento.save("entrenamiento.yml")
%run detectar_identificar rostros.ipynb
```

Fig 6. Entrenamiento con arreglos de etiquetas y generación de archivo .yml con resultados del entrenamiento.

e) detectar_identificar_rostros.ipynb

- Este script es el encargado de ejecutar la cámara web, detectar los rostros, encerrarlos en un recuadro y mostrar el nombre correspondiente en caso de reconocer el rostro. Si la predicción del rostro se encuentra entre 4 y 85 se dice que es una buena predicción de manera que se muestra la etiqueta con el nombre referente al rostro que más se asemeja al reconocimiento. En caso de que la variable solo sea mayor a 50 lo toma como desconocido.

```
# reconocimiento
id_, conf = reconocimiento.predict(roi_gray)
if conf >= 4 and conf < 85:
    #print(id_)
    #print(etiquetas[id_])
    font = cv2.FONT_HERSHEY_SIMPLEX

    nombre = etiquetas[id_]

    if conf > 50:
        #print(conf)
        nombre = "Desconocido"
```

Fig 7. Ejecución de cámara web y reconocimiento facial. Rostro es encerrado y nombre se muestra en parte superior.

- Si se desea capturar otro rostro, se pulsa la tecla “q” para finalizar la ejecución de la cámara web y posteriormente la tecla “a” para comenzar a grabar y solicitarle el nuevo nombre al usuario, mismo que será guardado utilizando el micrófono.



```
web_cam.release()
cv2.destroyAllWindows()

# Cuando todo esté hecho, liberamos la captura
option=input("\n \nIngresar la tecla a para guardar una nueva persona: ")


if option=='a':
    %run audio.ipynb

else:
    print("\nUn gusto trabajar con usted\n")
    #exit()
```

Fig 8. Se cierra cámara y se le pregunta al usuario si desea continuar o terminar con el programa.

4. Uso de carpeta images:

- Esta carpeta es aquella en la que se generan las nuevas carpetas con los nombres asignados por el usuario. En cada una de estas carpetas se guardan las imágenes correspondientes a las capturas tomadas referentes al nombre asignado.



Mario

Messi

Obama

Fig 9. Ejemplos de carpetas con nombres guardados por el usuario.



Lo explicado anteriormente representa la base para la detección de rostros por medio de la cámara web haciendo uso de OpenCV. A esta funcionalidad se le pueden agregar diferentes scripts para adecuarlo a diferentes necesidades y aplicaciones. Como complemento extra, además del reconocimiento de voz, se utiliza un script que convierte en texto en audio, de manera que al detectar el nombre, se puede escuchar el nombre. La ejecución de este script se dejó comentada, ya que retrasa de manera considerable la ejecución del programa, sin embargo, la implementación funciona y es correcta. Para poder realizar la conversión de texto a audio es necesario instalar lo siguiente:

```
!pip install pyttsx3
```

```
import pyttsx3
import win32com

eng=pyttsx3.init()

eng.say('Jesús')
eng.runAndWait()
eng.setProperty('rate', 200)
```

Fig 10. Ejemplo de código de conversión de texto en audio. Se trabaja con el texto “Jesús”.