



Visión Artificial

Título: PROYECTO

Nombre: Cesar Eduardo Campos Virgen

Registro: 22110354

6°G

1. Objetivos del Proyecto

El objetivo principal de este proyecto es desarrollar un sistema de reconocimiento facial que, utilizando la cámara de una computadora, sea capaz de identificar a la persona que se encuentra frente a ella. En caso de no encontrar coincidencias en la

base de datos, el sistema marcará al individuo como “desconocido”.

2. Explicación del Proyecto

El sistema utiliza la cámara web para capturar imágenes en tiempo real. Luego, mediante técnicas de visión computacional y aprendizaje automático, compara los rostros capturados con una base de datos de rostros previamente almacenados y etiquetados. Si se encuentra una coincidencia, se muestra el nombre de la persona. Si no, se muestra “Desconocido”.

Herramientas y tecnologías utilizadas:

- Lenguaje de programación: Python
- Librerías:
- OpenCV: para el manejo de la cámara y procesamiento de imágenes
- face_recognition: para la detección y reconocimiento facial
- numpy: para manejo de matrices
- Base de datos local de imágenes (carpeta con rostros conocidos)

3. Código

```
import cv2
import face_recognition
import os
import numpy as np

# Cargar imágenes conocidas
ruta = 'rostros_conocidos'
imagenes = []
nombres = []

for archivo in os.listdir(ruta):
    img = cv2.imread(f'{ruta}/{archivo}')
    imagenes.append(img)
    nombres.append(os.path.splitext(archivo)[0])

# Función para codificar rostros
def codificar_rostros(imagenes):
    codificaciones = []
    for img in imagenes:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        codificacion = face_recognition.face_encodings(img)[0]
        codificaciones.append(codificacion)
    return codificaciones

rostros_codificados = codificar_rostros(imagenes)

# Activar cámara
```

```

camara = cv2.VideoCapture(0)

while True:
    exito, frame = camara.read()
    rostros_en_frame = face_recognition.face_locations(frame)
    codigos_en_frame = face_recognition.face_encodings(frame, rostros_en_frame)

    for rostro_codificado, ubicacion in zip(codigos_en_frame, rostros_en_frame):
        resultados = face_recognition.compare_faces(rostros_codificados,
rostro_codificado)
        distancias = face_recognition.face_distance(rostros_codificados,
rostro_codificado)
        indice = np.argmin(distancias)

        if resultados[indice]:
            nombre = nombres[indice]
        else:
            nombre = "Desconocido"

        y1, x2, y2, x1 = ubicacion
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        cv2.putText(frame, nombre, (x1, y2 + 20), cv2.FONT_HERSHEY_SIMPLEX, 1,
(255, 255, 255), 2)
        cv2.imshow('Reconocimiento Facial', frame)

        if cv2.waitKey(1) == ord('q'):
            break

camara.release()
cv2.destroyAllWindows()

```

4. Pruebas y Errores

Durante el desarrollo se presentaron varios errores, entre ellos:

- Algunas imágenes no eran bien codificadas, por lo que se agregó un control para verificar que cada imagen tenga un rostro válido.
- Problemas de iluminación que afectaban la precisión del reconocimiento. Se mejoró la iluminación del entorno para mejores resultados.
- En algunos casos, rostros parcialmente cubiertos (mascarillas, gafas) eran detectados como “desconocidos”.

5. Resultados y Conclusión

El sistema funciona correctamente en condiciones óptimas de iluminación y con una base de datos bien definida. Reconoce con buena precisión a los usuarios registrados y marca adecuadamente como desconocidos a quienes no lo están.