

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas
Arquitectura de Computadores y Ensambladores 2
Primer Semestre 2024

Asistente Clases Virtuales

Juan Pedro Valle Lema, Mauricio Antonio Castro Guerra , Diego Facundo Pérez Nicolau, Diana Estefanía Berducido Domingo, Marcos Arnoldo Itzep Ixmay
202101648, 202100299, 202106538, 202000277, 201907156

Introducción

OpenCV es una librería que nos permite realizar múltiples actividades relacionadas con el escaneo de objetos, rostros o lo que nosotros deseemos que pueda observar la cámara. En esta ocasión veremos la implementación de esta junto a Python y Arduino para poder escanear rostros. En esta práctica se desarrollará un dispositivo de asistencia de clases virtuales utilizando una cámara externa y la biblioteca OpenCV de Python. Este dispositivo pretende tener previamente registrado el rostro del catedrático para que al detectarlo cree una nueva sesión en Meet y se mantenga en constante Face Tracking para una experiencia mejorada de una clase virtual.

Objetivos

Generales

- Desarrollar una aplicación de asistencia de clases virtuales utilizando OpenCV y Python que permita iniciar una llamada de meet a la vez que seguir la cara del expositor y guardar esta información en una base de datos.

Específicos

- Diseñar un dispositivo que auxilie al catedrático para poder dar el curso.
- Aprender a comunicar un sistema arduino con un sistema python y poder generar eventos físicos partiendo de un proceso digital.

Desarrollo de la práctica Stack Design Framework

En el mundo actual, la tecnología juega un papel crucial en casi todos los aspectos de nuestra vida diaria. Hoy en día buscamos la automatización de nuestras tareas para de esta manera aprovechar nuestro tiempo en otras cosas. En este caso queremos presentar un asistente para las llamadas.

El reconocimiento facial es una forma de biometría que utiliza medidas y características faciales para identificar individuos. Aunque esta tecnología ha existido durante décadas, su uso se ha vuelto más prevalente con el advenimiento de las redes neuronales y el aprendizaje profundo. Sin embargo, a pesar de sus avances, el reconocimiento facial todavía enfrenta varios problemas, como la precisión y la privacidad.

En este informe, nos proponemos abordar estos problemas mediante el desarrollo de una aplicación de reconocimiento facial utilizando OpenCV, Python y Processing. Nuestro objetivo es crear un sistema que no solo sea preciso, sino que también respete la privacidad del usuario. A través de este trabajo, esperamos contribuir a la mejora de la tecnología de reconocimiento facial y, en última instancia, a la sociedad en general.

Detector de Rostros

Infraestructura del Producto

- **Hardware**
 - Arduino
 - Jumpers/Cables y LED's
 - Case del Encapsulamiento
 - Cartón
 - 2 Botones
 - Cámara externa
- **Software**
 - Código de Arduino
 - Código de Python

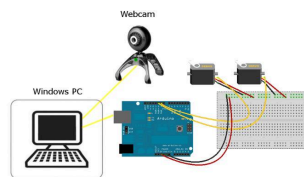


Figura 1: Prototipo del Producto

Sensores

- Cámara Web externa
- Botones

OPCIONES SECUNDARIAS

Conectividad

El dispositivo principal es un Arduino, el cual por medio de la cámara web recolecta imágenes y las procesa OpenCV para reconocer rostros a través de Python. Estos datos son enviados por medio de comunicación serial y recibidos por la aplicación que funciona con Python. Posteriormente los datos recibidos son guardados en una base de datos y subidos a la nube, a su vez manda instrucciones, al arduino de que realizar..

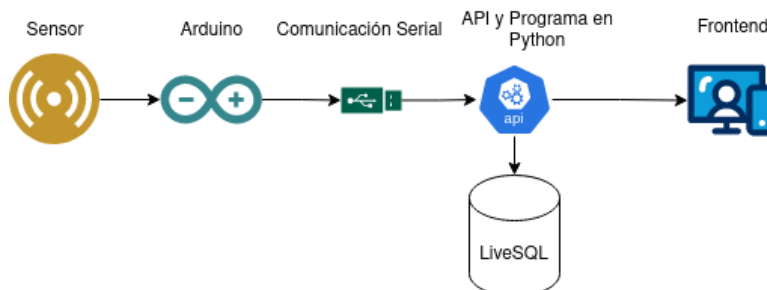


Figura 2: Conectividad del Producto

Tamaño del objeto

La maqueta tiene una altura de 10 cm por un ancho de 20 cm de largo.

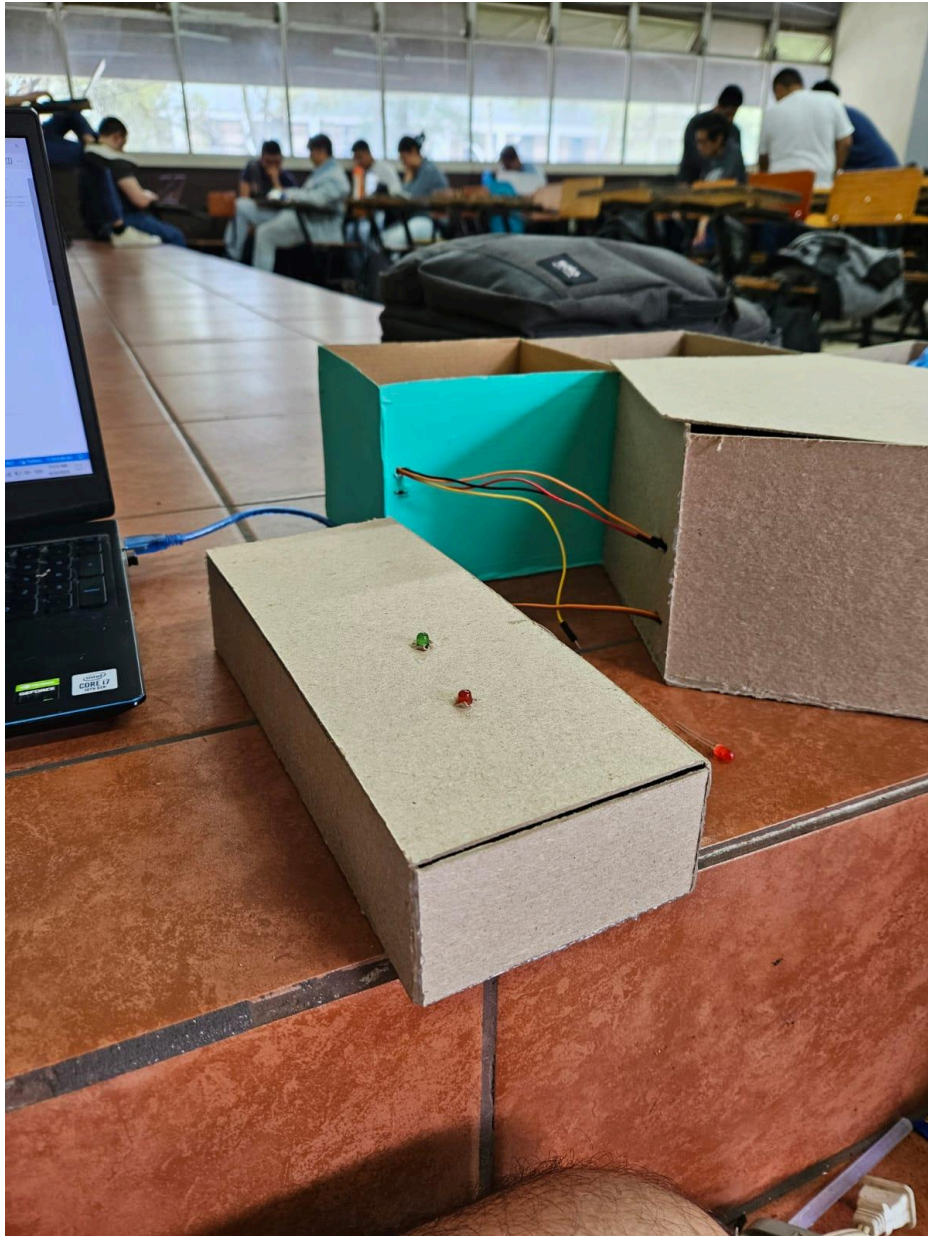


Figura 3: Imagen del Prototipo

Entorno del objeto

La idea es usarlo como un dispositivo móvil que podría ser conectado a cualquier computadora que tenga una conexión a una cámara web integrada o externa.

Consumo de Energía

De 46-90 mA

Analítica

La analítica parte de este dispositivo fue enfocada en el sector de datos, brindando un análisis descriptivo de ellos, esto con el fin de poder encontrar tendencias y patrones respecto a los rostros presentados a la cámara:

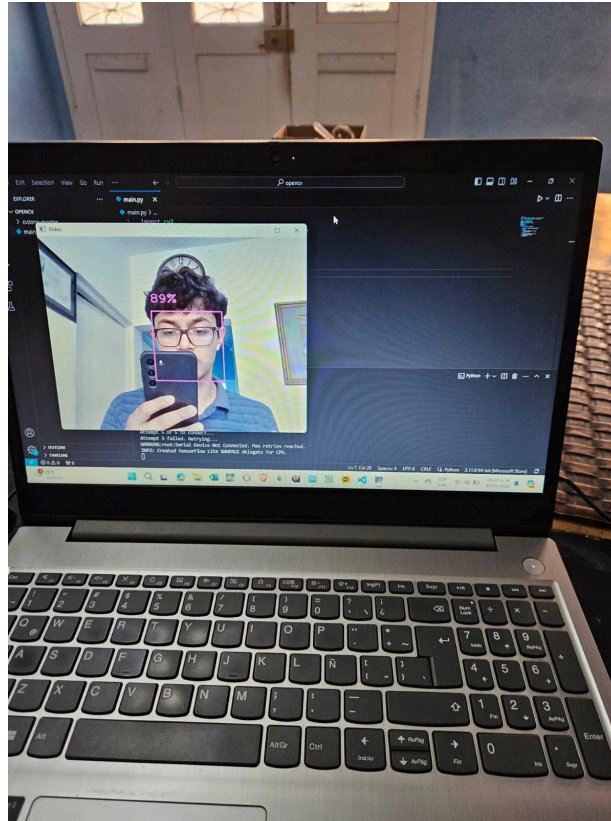


Figura 4: Visualización porcentaje facial

```
import cv2
from cvzone.FaceDetectionModule import FaceDetector
from cvzone.SerialModule import SerialObject
import time

arduino = SerialObject('COM6')
cap = cv2.VideoCapture(0)
detector = FaceDetector()

caras = []
contador = 0
while True:
    success, img = cap.read()

    img, bboxes = detector.findFaces(img)
    file = open("contador.txt", "r")
    datos_previos = int(file.read().strip())

    if len(bboxes) > 0:
        score = bboxes[0]['score'][0]
        if score > 0.9:
            arduino.sendData([1, 0, 1, 0])
            contador = datos_previos + 1
            print(contador)
            face = "VERDADERO"
            cv2.putText(img, f'CARA DETECTADA: {(face)}', (20, 70), cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0), 2)
            file = open("contador.txt", "w")
            file.write(str(contador))
            file.close()
            time.sleep(1)
        elif score < 0.9:
            arduino.sendData([0, 1, 0, 1])
            face = "FALSO"
            cv2.putText(img, f'CARA DETECTADA: {(face)}', (20, 70), cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0), 2)
            time.sleep(4)
```

Figura 5: Código Python

Imagen prototipo

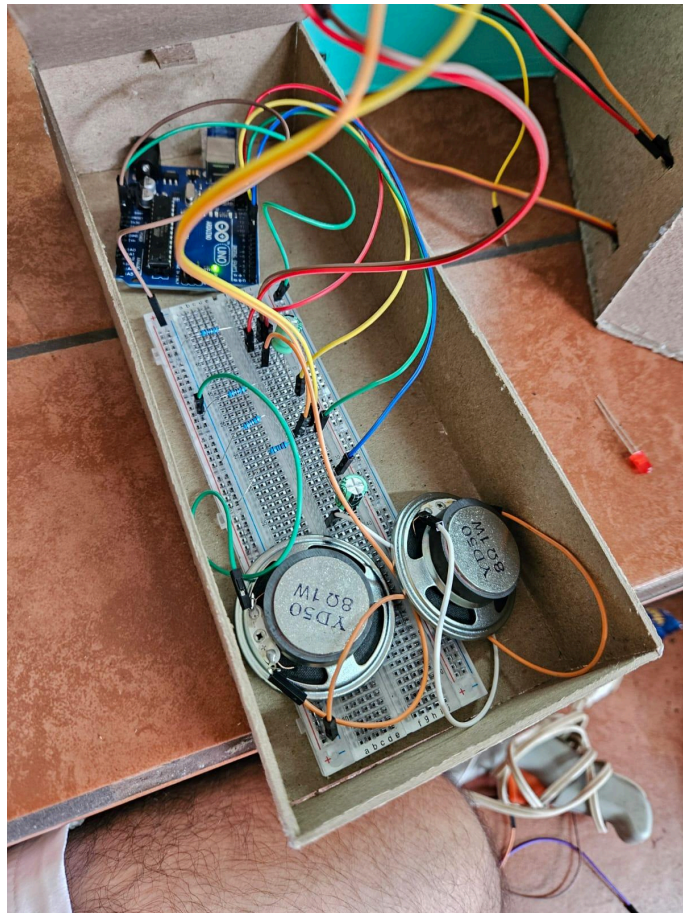


Figura 6: Imagen del Prototipo

Link repositorio: https://github.com/DFacundoPerezN/ACE2_1S24_G7.git