



Campus Monterrey

Inteligencia artificial avanzada para la ciencia de datos II (Gpo 501)

Momento de Retroalimentación: Reto Documentación  
Documentación del programa de Reconocimiento Facial

Juan Pablo Castañeda Serrano	A01752030
Aldo Daniel Villaseñor Fierro	A01637907
José Alfredo García Rodríguez	A00830952
Francisco Castorena Salazar	A00827756

## Objetivo

Se busca lograr la correcta detección de asistencia y participaciones en un curso mediante el análisis de video tomado durante la impartición de la clase. Se requiere de un algoritmo que de forma automática y correcta logre identificar a todos los alumnos participantes en el curso mediante reconocimiento facial. También se busca que el algoritmo sea capaz de identificar cuando hay una participación de un alumno y que dicha participación sea asociada al alumno que la realizó. Todos los datos recopilados por el algoritmo deben ser almacenados en una base de datos para su posterior despliegue en una página web.

## Descripción

Este programa de reconocimiento facial utiliza la biblioteca Ultralytics YOLO (You Only Look Once) para identificar y etiquetar caras en un video. Además, utiliza la biblioteca Face Recognition para reconocer rostros conocidos a partir de imágenes predefinidas, también utiliza la librería de OpenCV en Python para encasillar los rostros que se muestran en el video. El programa se conecta a una base de datos MongoDB para realizar un seguimiento de la asistencia de estudiantes y almacena la información en una colección llamada "estudiantes", de igual manera cuenta con una interfaz gráfica en la arquitectura de React para la visualización de la base de datos y la grabación de una nueva clase a analizar.

## Funcionamiento

El programa consta de las siguientes partes principales:

### 1. Conexión a la Base de Datos MongoDB:

- El programa se conecta a una base de datos MongoDB local llamada "fcRecog".
- Crea una colección llamada "estudiantes" para almacenar información sobre los estudiantes, su asistencia y sus participaciones

### 2. Carga de Caras Conocidas:

- Dentro de la interfaz de react, en la sección de “Nuevo estudiante” se toma una foto directamente desde el navegador, junto con los datos del alumno a registrar.
- El programa carga imágenes de rostros conocidos desde el directorio "./faceRecon/faces".
- Para cada imagen, se calcula una lista de 128 puntos que describen de mejor forma la cara mostrada y se almacena el código de reconocimiento facial y el nombre de la persona en listas separadas.

### 3. Reconocimiento de Caras en el Video:

- En la sección de “Grabar clase” en nuestra interfaz de react, se abre la cámara y se graba la clase. Se presiona el botón de “terminar clase” para detener la grabación y por consiguiente el botón “Actualizar base de datos” para que empiece el procesamiento del video en el backend
- Abre un archivo de video (por ejemplo, `./faceRecon/RecentClass/MostRecentClass.mp4`).
- Para cada cuadro del video, identifica las caras utilizando Face Recognition y YOLO.
- Utiliza Face Recognition para comparar las caras con las caras conocidas y etiqueta las caras correspondientes con los nombres de las personas si son reconocidas.
- Actualiza el recuento de asistencia en la base de datos para las personas reconocidas.
- Se notifica en la interfaz de react como este ya terminó su procesamiento

#### 4. Visualización en Tiempo Real:

- Muestra el video con las caras etiquetadas en una ventana de visualización en tiempo real.
- Permite al usuario presionar la tecla 'q' para detener la visualización y cerrar la ventana.
- Nota: Esto en caso de correrlo directamente en el backend, de hacerlo por la interfaz de react, la interfaz gráfica notifica al usuario una vez que acabe

#### 5. Eliminación del Video Original:

- Al final de la ejecución del programa, elimina el vídeo de origen (`"MostRecentClass.mp4"`).

### Algoritmos y herramientas utilizadas

- YOLO (You Only Look Once): El programa utiliza el modelo YOLO de Ultralytics para la detección de objetos en tiempo real, lo que incluye la detección de caras en el video.
- OpenCV(Open Source Computer Vision Library): es una framework de código abierto diseñado para resolver problemas de visión por computadora y procesamiento de imágenes. Esta biblioteca está escrita en C++ y cuenta con enlaces para Python, lo que la hace muy popular en la comunidad de desarrollo de Python.
- MediaPipe: Es una biblioteca desarrollada por Google capaz e realizar un seguimiento preciso de la postura de una persona en tiempo real a partir de imágenes o secuencias de video a partir de modelos de ML y Deep Learning.

- Face Recognition: Utiliza el módulo Face Recognition para el reconocimiento facial. Face Recognition extrae características faciales de las imágenes de las caras conocidas y las compara con las características faciales de las caras detectadas en el video para determinar si coinciden.
- MongoDB: El programa se integra con una base de datos MongoDB para almacenar información sobre los estudiantes y llevar un registro de su asistencia. Utiliza la biblioteca pymongo para interactuar con la base de datos.

## Estimación de pose

El algoritmo hace uso de la biblioteca mediapipe para la estimación de la posición de las articulaciones en la imagen. Como ya se mencionó mediapipe se basa en modelos de machine learning y deep learning para estimar las coordenadas de las articulaciones de la persona. En una investigación preliminar se descubrió que este framework es el más o sino es que uno de los frameworks más robustos y eficientes par la realización de esta tarea, sin embargo solo funciona en imágenes en donde aparece únicamente una persona. Para solucionar este impedimento es necesario detectar a una persona en la imagen y delimitar el área en la que aparece dicha persona, de esta manera nos es posible recortar la imagen para que únicamente aparezca una persona y realizar la estimación de la pose para cada una de las personas de la imagen, esta tarea es realizada YOLOv8 el cual analiza una imagen y es capaz de delimitar correctamente un cuadrado que encierra a cada una de las personas que aparecen en el frame. Una vez hecho esto solo es necesario dar la foto recortada a mediapipe para la estimación de pose y repetir este proceso para cada una de las personas que aparecen en la imagen.

## Uso

Para que el programa funcione correctamente necesita una instalación de MongoDB corriendo en el puerto 27017. Con una bd llamada "fcRecog" y las colecciones "estudiantes" y "images". Esto es una solución temporal hasta que la base de datos sea desplegada en un ip publico

Para correr el programa se necesita correr 2 diferentes partes. La primera es el backend:

### Backend

Navegar a la carpeta de "backend"  
Correr el comando python server.py

### Frontend

Luego correr el frontend:

Navegar a la carpeta de "frontend"  
Correr el comando npm install  
Correr el comando npm start

Ir a la ruta <http://localhost:3000> en su navegador

Dar click en "nuevo estudiante"

Capturar los datos

Dar click en "Comenzar cámara"

Dar click en "Capturar camara y datos"

En la derecha dar click a "Estadísticas"

Comprobar los datos

Dar click en "Grabar clase"

Dar click en "Comenzar clase"

Dar click en "Comenzar grabación"

Dar click en "Terminar clase"

Dar click en "Actualizar base de datos"