



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

Innovación en la Gestión de Asistencias y Participaciones en Aulas Multitudinarias: Un Enfoque Pionero en Detección de Rostros y Reconocimiento de Postura con YOLOv8 y Deep Learning

### **Autores:**

Juan Pablo Castañeda Serrano | A01752030

Aldo Daniel Villaseñor Fierro | A01637907

Francisco Castorena Salazar | A00827756

José Alfredo García Rodríguez | A00830952

### **Profesores de Reto:**

Ivan Mauricio Amaya Contreras

Edgar Covantes Osuna

Hugo Terashima Marìn

Noviembre 2023

# Índice

<b>1. Objetivo</b>	<b>2</b>
<b>2. Descripción</b>	<b>2</b>
<b>3. Algoritmos y herramientas utilizadas</b>	<b>2</b>
<b>4. Funcionamiento</b>	<b>2</b>
<b>5. Estimación de pose</b>	<b>3</b>
<b>6. Diagramas del funcionamiento de los algoritmos</b>	<b>4</b>
6.1. Algoritmo de registro de Participaciones . . . . .	4
6.2. Explicación del Sistema Integrado . . . . .	4
<b>7. Uso</b>	<b>5</b>
<b>8. Anexos</b>	<b>6</b>

## 1. Objetivo

Se busca lograr la correcta detección de asistencia y participaciones en un curso mediante el análisis de vídeo 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.

## 2. Descripción

Este programa de reconocimiento facial utiliza la biblioteca Ultralytics YOLO (You Only Look Once) para identificar y etiquetar caras en un vídeo. 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 vídeo. 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.

## 3. 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, incluyendo la detección de caras en el vídeo.
- OpenCV (Open Source Computer Vision Library): Es un 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 de realizar un seguimiento preciso de la postura de una persona en tiempo real a partir de imágenes o secuencias de vídeo mediante modelos de ML y Deep Learning.
- Face Recognition: Utiliza el módulo Face Recognition para el reconocimiento facial. Esta herramienta extrae características faciales de las imágenes de las caras conocidas y las compara con las características faciales de las caras detectadas en el vídeo 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.

Al final se decidió el utiliza YOLO-pose en lugar de MediaPipe debido a que el segundo tiene una limitación en la que no le permite funcionar con más de una persona, así que por la naturaleza de nuestro problema en donde por lo general siempre tendremos múltiples estudiantes nos es mejor utilizar una librería que detecte de forma nativa los keypoints de múltiples personas.

## 4. 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 vídeo:

- 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 vídeo en el backend.
- Abre un archivo de vídeo (por ejemplo, "./faceRecon/RecentClass/MostRecentClass.mp4").
- Para cada cuadro del vídeo, 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 una vez que este proceso ha terminado su procesamiento.

#### 4. Visualización en Tiempo Real:

- Muestra el vídeo 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: Si se ejecuta directamente en el backend, al correrlo desde la interfaz de React, ésta notifica al usuario una vez que el proceso ha finalizado.

#### 5. Eliminación del vídeo Original:

- Al finalizar la ejecución del programa, se elimina el vídeo original ("MostRecentClass.mp4").

## 5. Estimación de pose

Utilizamos YOLO-Pose para la estimación de pose y determinar las participaciones en el salón de clases. YOLO-Pose es un sistema avanzado que se basa en la detección de objetos y la estimación de poses utilizando algoritmos de aprendizaje profundo. La estimación de pose implica determinar la posición y orientación de las partes del cuerpo de una persona en una imagen o video. Esta tarea es fundamental para aplicaciones como reconocimiento de gestos, análisis de movimiento y seguimiento de personas.

Los primeros sistemas de estimación de pose se basaban en técnicas de reconocimiento de características, que identifican partes del cuerpo basándose en características específicas como la forma de los hombros, las caderas o las rodillas. Sin embargo, estas técnicas eran limitadas en su precisión y no funcionaban bien en condiciones de baja iluminación o en presencia de ruido.

En contraste, YOLO-Pose se refiere a la identificación precisa de puntos específicos en una imagen, conocidos como keypoints [1]. Estos puntos clave representan partes distintivas o características relevantes de un objeto, como articulaciones o puntos de referencia significativos. Estos keypoints se expresan típicamente mediante coordenadas 2D [x, y] o 3D [x, y, visible], siendo cruciales para comprender la posición relativa o la estructura de un objeto dentro de una imagen [1].

## 6. Diagramas del funcionamiento de los algoritmos

### 6.1. Algoritmo de registro de Participaciones

Frame tras frame se evaluará para cada persona si una de sus muñecas está por encima de su nariz, si se cumpliera esta lógica, se llamará al proceso de reconocimiento facial definido y explicado en la subsección anterior para identificar de los alumnos quien es aquel que está levantando la mano.

Seguido de esto, simplemente se tiene que cumplir un umbral de 20 frames para que se registre la participación en el sistema y sea enviado a la base de datos como una participación por parte de un alumno específico.

En la figura 1 se puede observar un diagrama de flujo que resume el proceso mediante el cual se registran las participaciones.

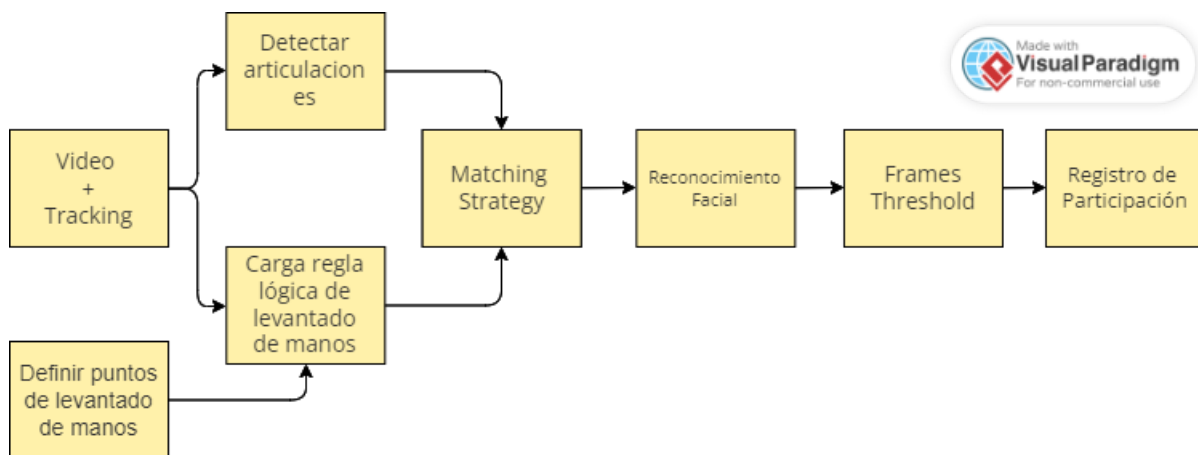


Figura 1: Diagrama de flujo de pasos a seguir por el algoritmo de registro de participaciones implementado.

### 6.2. Explicación del Sistema Integrado

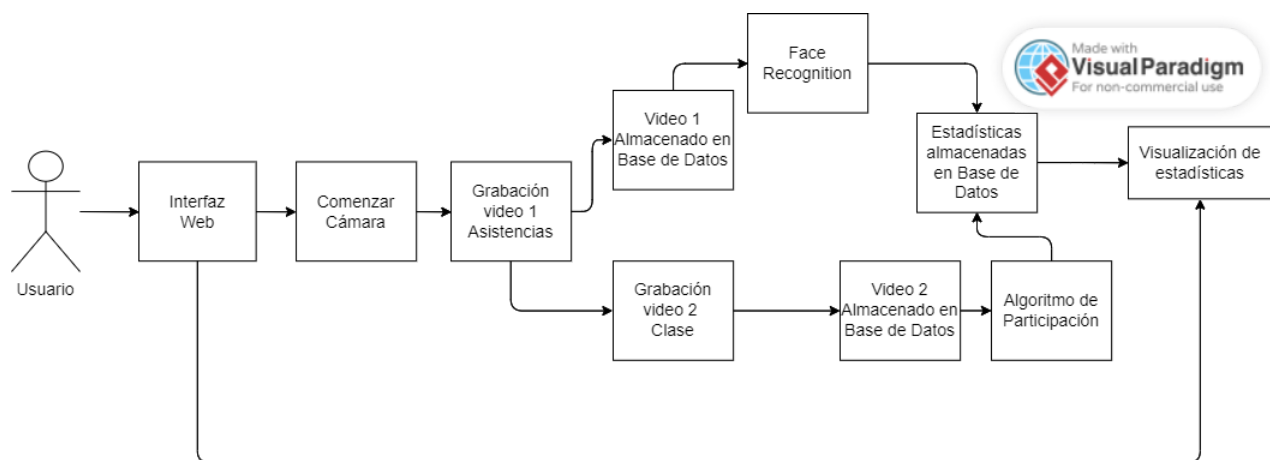


Figura 2: Diagrama de flujo del proceso realizado en el sistema implementado para la toma de asistencias y participaciones en el aula.

Como se muestra en la figura 2, el sistema inicia con la interacción del usuario en una interfaz web. Esta interfaz coordina funciones y servicios para grabar, almacenar, y procesar vídeos de clases y asistencias. Además, gestiona el registro de nuevos alumnos y la presentación de estadísticas mediante gráficas y valores estadísticos.

Una vez en la interfaz web, el usuario puede activar la cámara para el registro de asistencias y participaciones. Al iniciar la cámara, se debe hacer clic en "Empezar grabación de asistencia para comenzar a grabar el vídeo. Este vídeo se envía al backend para el reconocimiento facial, donde se mapean los puntos descriptores de las caras con la base de datos de alumnos y profesores. Los participantes deben pasar frente a la cámara durante este proceso.

Una vez que todos han sido registrados, se hace clic en "Empezar grabación de clase, que graba el vídeo de la clase y envía el vídeo de asistencias (vídeo 1) al backend. Este vídeo se almacena en la base de datos y se procesa en paralelo con la grabación de la clase para actualizar las estadísticas. Al finalizar la clase, se hace clic en "Terminar la clase, lo que almacena el vídeo de clase (vídeo 2) en la base de datos.

Posteriormente, el vídeo 2 es procesado por la parte del backend que contiene el algoritmo de registro de participaciones. La información se envía a la base de datos para actualizar las estadísticas respectivas. Este proceso puede tardar algunos minutos según la duración del vídeo.

Una vez que ambos vídeos se han procesado y los datos se han registrado en la base de datos, el usuario puede acceder a la sección de estadísticas desde la interfaz. Aquí, se presentan gráficas y datos relevantes sobre estudiantes y clases.

## 7. Uso

Para que el programa funcione correctamente necesita una instalación de MongoDB corriendo en el puerto 27017. Con una base de datos llamada "fcRecogz las colecciones ".estudiantesz images". Esto es una solución temporal hasta que la base de datos sea desplegada en una dirección IP pública.

Para ejecutar el programa se requiere llevar a cabo 2 partes diferentes. La primera es el backend:

### Backend

1. Navegar hasta la carpeta "backend".
2. Ejecutar el comando `python server.py`.

Luego, para el frontend:

### Frontend

1. Navegar hasta la carpeta "frontend".
2. Ejecutar el comando `npm install`.
3. Ejecutar el comando `npm start`.
4. Abrir la ruta `http://localhost:3000` en su navegador.
5. Hacer clic en "nuevo estudiante".
6. Capturar los datos.
7. Hacer clic en "Comenzar cámara".

8. Hacer clic en Capturar cámara y datos".
9. En la sección derecha, hacer clic en .Estadísticas".
10. Verificar los datos.
11. Hacer clic en “Grabar clase”.
12. Hacer clic en “Comenzar clase”.
13. Hacer clic en “Comenzar grabación”.
14. Hacer clic en “Terminar clase”.
15. Hacer clic en “Actualizar base de datos”.

## Referencias

- [1] Ultralytics, “Pose.” [Online]. Available: <https://docs.ultralytics.com/tasks/pose/>

## 8. Anexos

Repositorio de github: <https://github.com/a01752030/PortafolioRetoFC>

\*En caso de requerir acceso, mandar correo a cualquier integrantes del equipo A00830952@tec.mx