**Índice**

[**Introducción y objetivos** 2](#_Toc156067545)

[**Descripción Técnica** 2](#_Toc156067546)

[**Cámara Perfil: Conteo y Detección de Personas** 2](#_Toc156067547)

[*Detección* 2](#_Toc156067548)

[*Controlador y Contador* 3](#_Toc156067549)

[*Código Principal* 4](#_Toc156067550)

[**Cámara Frontal: Análisis de Género y Edad** 5](#_Toc156067551)

[**Conclusión y Posibles Ampliaciones** 5](#_Toc156067552)

[**Tecnologías Usadas** 5](#_Toc156067553)

[**Repositorios** 5](#_Toc156067554)

**Introducción y objetivos**

El presente estudio se focaliza en la identificación, recuento y análisis de la edad y género de individuos mediante el empleo de *YOLOv8* y *MediaPipe*. El propósito fundamental es poder examinar y supervisar el tránsito de personas dentro de un establecimiento, posibilitando la generación de estadísticas detalladas sobre la cantidad y tipología de personas que ingresan a lo largo del día.

La disposición de una cámara posicionada lateralmente permite la detección de personas que atraviesan la entrada del establecimiento, llevándose a cabo un seguimiento para determinar si están ingresando o saliendo del edificio. Una vez que una persona cruza la línea límite, indicando su entrada, se activa una segunda cámara ubicada frontalmente en la entrada. Esta cámara captura una imagen y realiza un análisis para determinar el género y la edad de la persona detectada.

Los datos recopilados se almacenan en una base de datos dedicada, la cual se utilizará para generar estadísticas detalladas sobre el flujo de personas en el establecimiento a lo largo del año. La información, tanto en forma de vídeos de las cámaras como de métricas y gráficos, estará accesible a través de un servidor web especialmente desarrollado para este propósito.

**Descripción Técnica**

## **Cámara Perfil: Conteo y Detección de Personas**

Este proceso se divide en dos partes: detección de las personas que capture la cámara y el conteo de cuántas entran y salen del lugar. Solo se puede contar una persona que pase a la vez por lo que, en esta fase, el proyecto solo está pensado para establecimientos o servicios cuyas entradas solo den capacidad para una persona.

### *Detección*

Para la detección, se han desarrollado dos clases fundamentales dentro del archivo *detection.py*: **Person** y **YOLOPersonDetector**.

***Clase Person***

La primera clase, denominada Person, desempeña un papel esencial en la gestión individual de cada persona detectada. Su función principal consiste en almacenar información relevante sobre la persona, como las coordenadas de la caja delimitadora de su figura, la posición central, la trayectoria, la dirección del movimiento, tiempo de vida de la persona siendo detectada y un indicador que señala si ha superado el tiempo máximo de seguimiento.

Este seguimiento temporal se realiza mediante métodos como **age\_one()**, que incrementa la edad de la persona en una unidad, y **is\_time\_out()**, que verifica si la persona ha excedido su tiempo de seguimiento máximo.

Asimismo, la clase posee el método **update\_coords()** para actualizar las coordenadas de la persona y el método **calculate\_dir()** para calcular su dirección de movimiento con relación a unos límites especificados.

***Clase YOLOPersonDetector***

La segunda clase, YOLOPersonDetector, se encarga de integrar el modelo YOLO para la detección de personas en una imagen.

Su método principal, **predict()**, acepta una imagen como entrada y devuelve una lista de las coordenadas de las personas detectadas. Este método utiliza la biblioteca *ultralytics* para realizar predicciones a partir del modelo YOLO, extrayendo las cajas delimitadoras de los objetos detectados y verificando la categoría del objeto para determinar si corresponde a una persona. En caso afirmativo, las coordenadas se agregan a la lista de las coordenadas de las personas detectadas y se devuelve como resultado.

Estas clases han sido diseñadas con un enfoque modular y orientado a objetos, facilitando la implementación y el mantenimiento del sistema.

### *Controlador y Contador*

Se han diseñado dos clases fundamentales para gestionar la visualización de información y llevar a cabo el seguimiento y conteo de individuos detectados en cada frame. Estas clases, denominadas **Painter** y **PersonCounterController**, cumplen roles clave en el sistema y se detallan a continuación.

***Clase Painter***

La clase **Painter** se encarga de la presentación visual de la información en la imagen procesada. Su función principal es dibujar líneas representativas, contar el número de personas detectadas y resaltar visualmente la presencia de cada individuo en la imagen.

Se utilizan diversos métodos, como **paint\_lines()**, que dibuja líneas divisorias y texto en la imagen para etiquetar las líneas de delimitan los límites de control, y **paint\_counter()**, que muestra un contador de personas en una esquina de la imagen. Además, el método **paint\_person()** resalta visualmente cada persona detectada, marcando su caja delimitadora, centro y etiqueta.

***Clase PersonCounterController***

La segunda clase, **PersonCounterController**, opera como el controlador principal de la aplicación. Se encarga de orquestar la interacción entre el detector de personas, el presentador visual y de avisar al servidor de que una persona ha entrado al local.

Al recibir un frame de la cámara, redimensiona la imagen según las especificaciones proporcionadas y realiza predicciones de detección de personas mediante el objeto **YOLOPersonDetector**. Luego, utiliza el objeto **Painter** para visualizar líneas divisorias, contadores y la presencia de personas en la imagen.

El método **track()** se encarga de seguir el movimiento de las personas detectadas y calcular su dirección en relación con límites específicos en la imagen con el método realizado en la clase **Person**, **calculate\_dir()**.

Por otro lado, el método **count()** actualiza el contador de personas en función de la dirección de movimiento de cada persona detectada y, a su vez, va enviando esta información al servidor, avisando a la **Cámara Frontal** de que tiene que realizar una foto cuando el contador suma 1.

### *Código Principal*

En el script principal, llamado **main.py**, se establece la lógica central para la ejecución del sistema de conteo y seguimiento de personas.

En la primera sección, se importan las clases esenciales desde los módulos **control** y **camera**. La clase **PersonCounterController** se encarga del control del conteo de personas, mientras que **CameraServer** se utiliza para el envío de frames al servidor desde la cámara.

La función principal, **main()**, inicializa instancias de las clases mencionadas y configura parámetros clave como el tamaño de la imagen, límites de conteo, edad máxima y el modelo YOLO a utilizar. Posteriormente, se utiliza la cámara del sistema, inicializada mediante OpenCV, para capturar frames en un bucle continuo. Cada frame capturado es procesado por el objeto **PersonCounterController**, llevando a cabo la detección de personas, así como el seguimiento y conteo.

En resumen, **main.py** funciona como el punto de entrada del sistema, coordinando la adquisición y procesamiento de frames, y presentando visualmente los resultados de la detección y conteo de personas en tiempo real. La modularidad de las clases facilita una fácil expansión y mantenimiento del sistema.

## **Cámara Frontal: Análisis de Género y Edad**

**Servidor Web**

# **Conclusión y Posibles Ampliaciones**

# **Tecnologías Usadas**

# **Repositorios**