

Sistema de Vigilancia de Ingreso Basado en Reconocimiento Facial

Nayibe Yesenia Arias Cortez

22 de julio de 2025

Índice

1. Introducción	2
2. Justificación	2
3. Flujo de trabajo	2
3.1. Descripción general	2
3.2. Diagrama de Flujo Resumido	2
4. Componentes y herramientas principales	3
4.1. face_recognition	3
4.2. OpenCV (cv2)	3
5. Explicación detallada del sistema	3
5.1. Carga de datos	3
5.2. Procesamiento en tiempo real	3
5.3. Reconocimiento facial	3
5.4. Registro y UI	3
6. Resultados esperados	4
7. Conclusiones	4
8. Recomendaciones futuras	4
9. Diagrama de flujo completo del sistema	4

1. Introducción

El presente trabajo describe el desarrollo de un sistema de vigilancia de ingreso automatizado utilizando técnicas de reconocimiento facial. El objetivo principal es registrar de forma fiable el acceso de personas a una institución educativa, optimizando tiempo y reduciendo errores humanos.

El sistema está implementado en Python y emplea las librerías **OpenCV** (**cv2**) para el procesamiento de imágenes y **face_recognition** para la detección y el reconocimiento facial mediante modelos de aprendizaje profundo.

2. Justificación

El control de asistencia manual es susceptible a imprecisiones y falta de escalabilidad. Con la adopción de un sistema de reconocimiento facial se automatiza el registro de ingreso, incrementando la seguridad y la eficiencia. Además, la posibilidad de utilizar múltiples imágenes por persona mejora la robustez del reconocimiento bajo distintas condiciones de iluminación, ángulos o expresiones faciales.

3. Flujo de trabajo

3.1. Descripción general

El sistema sigue la secuencia de pasos ilustrada en el diagrama de flujo (Figura 9). Cada cuadro representa un proceso importante dentro de la aplicación, desde la carga inicial de datos y configuración hasta la visualización de la interfaz de usuario.

3.2. Diagrama de Flujo Resumido

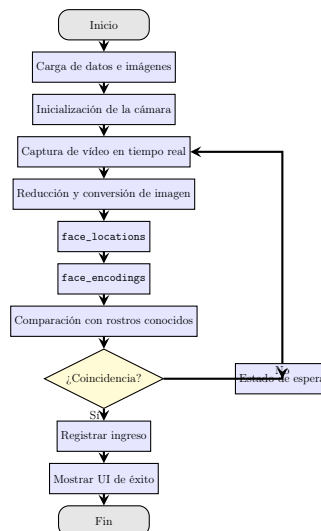


Figura 1: Diagrama de flujo del sistema de vigilancia de ingreso.

4. Componentes y herramientas principales

4.1. face_recognition

La librería `face_recognition` se basa en el modelo dlib CNN y genera vectores de 128 dimensiones que representan características faciales. Las funciones principales utilizadas son:

- `face_locations`: detecta los rostros presentes en la imagen.
- `face_encodings`: produce las codificaciones faciales.
- `face_distance`: compara las codificaciones con las previamente almacenadas usando la distancia euclidiana.

4.2. OpenCV (cv2)

OpenCV se emplea para la adquisición de vídeo, manipulación de imágenes y construcción de la interfaz gráfica. Entre las funciones más relevantes se encuentran:

- `VideoCapture`: acceso a la cámara web.
- `resize`: reducción de resolución para acelerar el procesamiento.
- `cvtColor`: conversión de BGR a RGB.
- `rectangle` y `putText`: creación de elementos visuales.
- `imshow`: visualización en tiempo real.

5. Explicación detallada del sistema

5.1. Carga de datos

Se lee el archivo `Persona_data.csv` que contiene la información de cada usuario y se cargan las imágenes correspondientes desde la carpeta designada. Cada imagen se codifica y se almacena junto con el nombre de la persona.

5.2. Procesamiento en tiempo real

Cada fotograma capturado se reduce (factor de reducción `FACTOR_REDUCCION`) para disminuir el costo computacional y se transforma de BGR a RGB, formato requerido por la biblioteca `face_recognition`.

5.3. Reconocimiento facial

Se comparan las codificaciones del rostro capturado con las codificaciones almacenadas. Si la distancia es menor al umbral `TOLERANCIA_COINCIDENCIA`, se considera una coincidencia.

5.4. Registro y UI

En caso de coincidencia, se registra el ingreso en `registro_log.csv`. El sistema muestra un marco verde, un ícono de verificación y la tarjeta de perfil del usuario reconocido.

6. Resultados esperados

- Reconocimiento preciso de usuarios registrados.
- Registro automático de fecha y hora de ingreso.
- Interfaz amigable con retroalimentación inmediata.

7. Conclusiones

Se demostró la factibilidad de implementar un sistema de reconocimiento facial para control de acceso utilizando herramientas de código abierto. La combinación de OpenCV y `face_recognition` ofrece una solución robusta y escalable.

8. Recomendaciones futuras

- Integrar autenticación multifactor.
- Migrar a una base de datos relacional o NoSQL.
- Implementar detección de rostros con mascarilla.
- Crear una versión web o móvil del sistema.

9. Diagrama de flujo completo del sistema

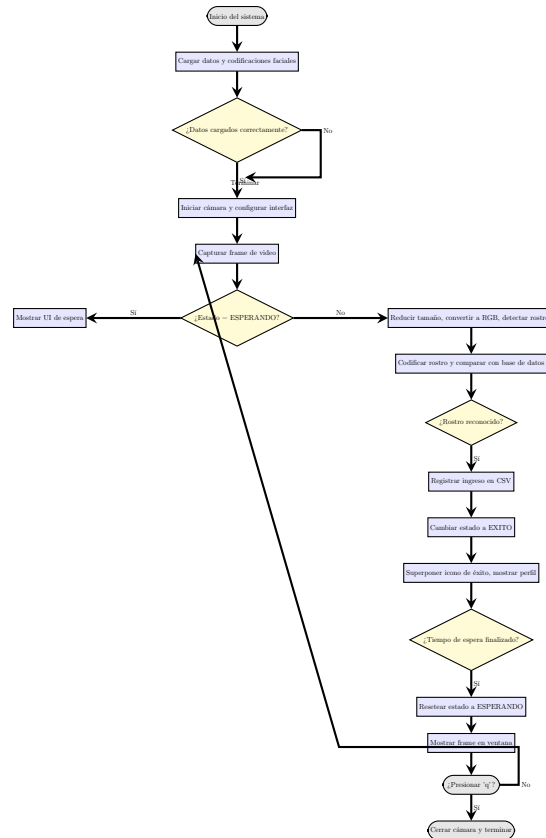


Figura 2: Diagrama de flujo completo del sistema de vigilancia de ingreso.