

Sistema De Reconocimiento Facial Para El Control De Asistencia En Aulas Universitarias

Integrantes:

Líder: Ricardo Andrey Flórez Torres

Julián David Merchán Quiroz

Jorge Iván Monroy Martínez

Alejandro David Contreras Parra

Unidades Tecnológicas de Santander

Tecnología en Desarrollo de Sistemas Informáticos

Docente: Ing. Pedro Ramírez

Bucaramanga, Santander

Fase: **Desarrollo**

1.introducción: En la fase de desarrollo del proyecto “Sistema de Reconocimiento Facial para el Control de Asistencia en Aulas Universitarias”, es esencial seleccionar las herramientas de software adecuadas para asegurar la correcta implementación del sistema. Estas herramientas incluyen programas para la captura y procesamiento de imágenes, algoritmos de reconocimiento facial, almacenamiento de datos, y desarrollo de interfaces de usuario.

2. Programas Seleccionados

2.1. Captura y Procesamiento de Imágenes

Para la fase de captura y procesamiento de imágenes, es necesario utilizar un conjunto de herramientas que permitan la manipulación de imágenes en tiempo real, la detección facial y el pre procesamiento para alimentar el algoritmo de reconocimiento.

Herramientas seleccionadas:

OpenCV (Open Source Computer Vision Library)

Descripción: OpenCV es una biblioteca de visión artificial de código abierto que proporciona funciones avanzadas para el procesamiento de imágenes y video. Se utilizará para capturar las imágenes faciales desde las cámaras y realizar las tareas de detección y pre procesamiento de las imágenes.

Funcionalidades clave:

- Captura de imágenes en tiempo real.
- Detección de rostros en las imágenes.
- Pre procesamiento de imágenes (escalado, normalización).

Razón de selección: OpenCV es ampliamente compatible con lenguajes como Python y C++, lo que facilita su integración en el sistema.

2.2. Reconocimiento Facial

Para el reconocimiento facial, se utilizarán herramientas y bibliotecas especializadas en la identificación de rostros utilizando algoritmos de machine learning y redes neuronales.

Herramientas seleccionadas:

Dlib

Descripción: Dlib es una biblioteca de aprendizaje automático que contiene un modelo robusto de reconocimiento facial. Ofrece alta precisión y es adecuada para aplicaciones en tiempo real.

Funcionalidades clave:

- Detección de características faciales clave (ojos, nariz, boca).
- Reconocimiento de rostros basado en modelos entrenados.

Razón de selección: Dlib ha demostrado un rendimiento óptimo para el reconocimiento facial en aplicaciones con restricciones de tiempo real, además de ser de código abierto.

DeepFace (opcional)

Descripción: DeepFace es una biblioteca en Python que simplifica la implementación de algoritmos de reconocimiento facial avanzados, permitiendo el uso de múltiples modelos preentrenados.

Funcionalidades clave:

- Reconocimiento facial con redes neuronales convolucionales (CNN).
- Integración sencilla con bases de datos y sistemas de autenticación.

Razón de selección: Permite experimentar con diferentes modelos para mejorar la precisión del sistema.

2.3. Almacenamiento y Gestión de Datos

Es necesario implementar una base de datos que almacene la información de los estudiantes y los registros de asistencia. Para ello, se utilizará un sistema de gestión de bases de datos relacional.

Herramientas seleccionadas:

MySQL

Descripción: MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que proporciona alta escalabilidad y flexibilidad.

Funcionalidades clave:

- Almacenamiento de datos biométricos y registros de asistencia.
- Consultas eficientes para obtener la asistencia de los estudiantes.

Razón de selección: MySQL es una opción robusta y ampliamente utilizada en aplicaciones a gran escala, lo que asegura confiabilidad y facilidad de uso en el sistema de asistencia.

SQLite (para pruebas o prototipos)

Descripción: SQLite es una base de datos ligera y embebida, ideal para pruebas rápidas o para aplicaciones que no requieren la complejidad de un servidor de bases de datos completo.

Funcionalidades clave:

- Almacenamiento de datos en un archivo local sin necesidad de configuración de servidor.

Razón de selección: Se utilizará durante la fase de prototipado para simplificar el desarrollo.

2.4. Desarrollo de la Interfaz de Usuario

Para proporcionar una interfaz de administración del sistema, se utilizarán frameworks web para construir un panel de control accesible a través de navegadores.

Herramientas seleccionadas:

Flask

Descripción: Flask es un microframework web de Python, ideal para construir aplicaciones ligeras y modulares. Se utilizará para desarrollar la interfaz que permita a los administradores (profesores) visualizar los registros de asistencia.

Funcionalidades clave:

- Desarrollo de aplicaciones web ligeras y personalizables.
- Fácil integración con bases de datos y otros servicios.

Razón de selección: Flask es flexible, fácil de usar y permite un desarrollo rápido de aplicaciones web.

Bootstrap

Descripción: Bootstrap es un framework de diseño web que facilita la creación de interfaces responsivas y visualmente atractivas.

Funcionalidades clave:

- Diseño de interfaces responsivas para visualizar en múltiples dispositivos (computadoras, tabletas, celulares).
- Componentes visuales pre contruidos (tablas, botones, formularios).

Razón de selección: Facilita la creación de una interfaz de usuario moderna y responsiva sin necesidad de diseñar desde cero.

2.5. Plataforma para Entrenamiento del Modelo (opcional)

Si se opta por entrenar un modelo de reconocimiento facial personalizado, es posible utilizar plataformas en la nube para acelerar el proceso de entrenamiento.

Herramientas seleccionadas:

Google Colab

Descripción: Google Colab es una plataforma gratuita que proporciona acceso a GPU y entornos de Python pre configurados para ejecutar scripts de machine learning.

Funcionalidades clave:

- Entrenamiento de modelos de reconocimiento facial con aceleración por GPU.
- Entorno interactivo de Python.

Razón de selección: Google Colab es gratuito y proporciona recursos avanzados que aceleran el proceso de entrenamiento de modelos.