

Servicio Social, Trimestre: 25-I, 2025

CONSTRUCCIÓN DE UNA BASE DE DATOS DE VIDEOS AÉREOS Y SU ANÁLISIS VÍA HERRAMIENTAS DE IA

Martínez Buenrostro Jorge Rafael.

Universidad Autónoma Metropolitana
Unidad Iztapalapa, México
molap96@gmail.com

Resumen: Este documento describe el proyecto cuya meta es contar con una caracterización de los grupos de humanos que se desplazan juntos. Identificando las características estadísticas de los mismos.

1. Introducción

1.1. Descripción general del proyecto

La simulación de una red de comunicaciones en donde intervienen dispositivos personales de comunicación requiere contar con modelos que representen fielmente los patrones de movimiento de las personas. De otra manera, la utilidad de las conclusiones que se puedan obtener de esa simulación es limitada.

Para avanzar hacia la definición de un modelo de movilidad humana grupal, se propone la construcción de una base de datos de videos aéreos (capturados por un dron) y su análisis vía herramientas de IA. Esto nos permitirá determinar algunas características de la movilidad de interés.

1.2. Objetivos y propósitos

El objetivo principal del proyecto es contar con una caracterización de los grupos de humanos que se desplazan juntos. Identificando las características estadísticas de los grupos de humanos que se desplazan juntos.

Los propósitos del proyecto son:

- Construir una base de datos de videos aéreos de grupos humanos.
- Usar un modelo de IA que permita identificar y caracterizar los grupos humanos en los videos.
- Analizar los patrones de movimiento y las interacciones entre los grupos humanos.

1.3. Alcance del sistema

El sistema se enfoca en la recolección de datos aéreos de grupos humanos y su análisis utilizando herramientas de IA. El alcance incluye:

- Captura de videos aéreos mediante un dron en diferentes escenarios y condiciones.
- Almacenamiento y organización de los videos en una base de datos estructurada.
- Procesamiento de los videos para la detección y seguimiento de individuos y grupos humanos.
- Extracción de características relevantes sobre la movilidad y las interacciones grupales.
- Generación de reportes y visualizaciones de los resultados obtenidos.

No se considera dentro del alcance el desarrollo de hardware de drones ni la implementación de modelos de IA desde cero; se utilizarán herramientas y modelos existentes.

2. Requisitos

2.1. Requisitos del sistema

- Base de datos PostgreSQL 17.5.1
- Versión mínima de Python 3.13.3
- Dependencias principales:
 - torch
 - torchvision
 - numpy
 - pandas
 - scikit-learn
 - matplotlib
 - seaborn
- Dependencias procesamiento de imágenes/video:
 - ultralytics
- Dependencias de tracking:
 - deep-sort-realtime
- Dependencias de multimedia:
 - ffmpeg-python
- Dependencias de base de datos:
 - sqlalchemy
 - psycopg2-binary
- Dependencias de variables de entorno:
 - python-dotenv
- Dependencias de barras de progreso:
 - tqdm

2.2. Instrucciones de instalación

1. Instalar PostgreSQL 17.5.1 desde la página oficial de PostgreSQL.
2. Clonar el repositorio del proyecto desde Github el proyecto se encuentra dentro de la carpeta **Implementación**.
3. Crear un entorno virtual en Python:

```
python -m venv .venv
source .venv/bin/activate # Linux/Mac
.\.venv\Scripts\activate  # Windows
```

4. Instalar las dependencias del proyecto:

```
pip install -r requirements.txt
```

5. Configurar variable de entorno:

```
cp .env.example .env
nano .env # Editar valores
```

6. Verificar instalación:

```
python tests/check_requirements/
```

3. Arquitectura

3.1. Diagrama de la estructura del proyecto

```
Proyecto/
├── .venv/
├── src/
│   ├── database/
│   │   ├── connection.py
│   │   ├── creationOfModels.py
│   │   └── models/
│   ├── features/
│   ├── menus/
│   │   └── shared/
├── tests/
└── requirements.txt
```

Proyecto/ Raíz del proyecto que contiene todos los archivos y directorios necesarios para el desarrollo y ejecución de la aplicación.

.venv/ Directorio del entorno virtual de Python, que aísla las dependencias del proyecto para evitar conflictos con otros proyectos.

src/ Contiene el código fuente principal de la aplicación, donde se organizan los módulos y paquetes necesarios para su funcionamiento.

tests/ Directorio dedicado a las pruebas automatizadas, que incluye tanto pruebas unitarias como de integración para asegurar la calidad del código.

requirements.txt Archivo que lista las dependencias del proyecto, especificando las versiones de los paquetes necesarios para su correcto funcionamiento.

3.2. Descripción de los componentes

Aquí estará la descripción de los componentes del proyecto, incluyendo sus responsabilidades y cómo interactúan entre sí. Cada componente debe ser claramente definido para facilitar la comprensión de la arquitectura general.

3.3. Flujo de datos

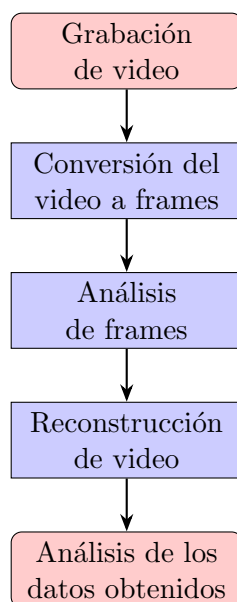


Figura 1: Flujo de datos

4. Configuración

4.1. Archivos de configuración

4.2. Parámetros ajustables