Overview:

The project is designed to act as a **surveillance system using drone-captured images and videos**. The system is divided into two major parts:

- **Aerial Capture Module (Drone side)**
- **Ground Processing Module (Face Recognition system)**

How it works (Conceptual Flow):

1. A drone flies over an area during surveillance missions.
2. It captures **images or video frames** from the sky.
3. These images/videos are **sent in real-time (or near real-time)** to the **ground control system**.
4. The ground system performs **face detection and recognition** on each image/frame received.
5. The system:
    - Identifies all the faces in the image
    - Compares them against a known database
    - Labels known individuals with their names
    - Adds new unknown faces to the database
6. The output is shown with:
    - **Bounding boxes with names** on the detected faces
    - A **list of all recognized individuals** displayed on the image window itself

This system allows operators to **instantly know who is present in a captured image**, making it highly useful for **security, surveillance, crowd monitoring, and investigation purposes.**

System Modules

1. Aerial Capture Module (Drone Side)

- Captures high-resolution images and videos of the surroundings.
- Sends these images/videos wirelessly to the ground system for processing.

2. Ground Processing Module (Recognition System)

- This is the main part of the project and is divided into two modes for optimization:

Mode	Description
Single-Person Recognition	Designed for cases where only **one face** is expected in the image. It processes faster and uses simpler logic.
Multiple-People Recognition	Detects and processes **multiple faces** present in the image, recognizes each of them, and shows results on screen.

- Each mode performs the following tasks:
    1. Detect faces in the image.
    2. Generate unique numerical encodings of the faces.
    3. Compare detected faces with already known encodings in the database.
    4. If matched → label with the existing name.
       If not matched → register it as a new person.
    5. Show final annotated image with names and recognized people list.

Tools and Technologies Used

Tool / Library	Purpose
Python 3.9	Core programming language
face_recognition	Face detection, encoding, and matching
OpenCV (cv2)	Image processing, drawing boxes, display
NumPy	Array and numerical operations
Pickle	Storing and loading known face encodings
OS	Handling folders and files for storing logs

Algorithms and Concepts Used

Step	Description
Face Detection	Locates faces in the input image using face_recognition (HOG-based face detector by default).
Face Encoding	Extracts a 128-d numerical feature vector for each face.
Face Matching	Compares the new face encoding with existing encodings using Euclidean distance. If distance < threshold → same person.
Face Registration	Stores encodings of new/unknown faces along with assigned names for future recognition.

Process Flow

1. Drone captures image/video.
2. Images are sent to ground system.
3. Ground system loads previously known face encodings (if available).
4. Detects faces in the incoming image.
5. Generates encodings for each face.
6. Matches faces with known database.
7. Unknown faces are registered as new entries.
8. Draws boxes + names on the image.
9. Displays the image along with a side list of all recognized people.

Current Output

- Faces detected and highlighted with **green bounding boxes**.
- **Names displayed** above each face.
- **List of all recognized people** shown on the image window for easy reading.