
**Prink Hapaliya**

**Trainee ai/ml engineer**

# Smart Surveillance System with Behavioral Analytics

➔ **Introduction**

- This system aims to analyze the behavior of individuals using surveillance camera feeds. It involves several components:

- app.py: This is the main file to initiate the system. It reads a JSON file containing camera information, extracts relevant data, and organizes it into dictionaries.

- Camera Manager: This class manages all cameras. Each camera is handled in a separate thread. It creates instances of the CameraObject and DetectObject classes, improves video speed, and stores values in dictionaries.

- Camera Object: This object processes frames from cameras, generates desk URLs for live streaming, and stores frame values in queues. These queues are accessed by the CameraManager.

- Detect Object: This component utilizes a custom-trained YOLOv8s model to detect people in frames. It also employs object tracking to assign IDs to individuals and determine if they are within designated desk regions.

- Tracker: This layer tracks objects and assigns IDs to detected individuals. It communicates with the DetectObject to update coordinates and IDs, storing values in dictionaries for individual person detection.

- Desk Manager: Responsible for detecting individuals within desk regions. It creates desk objects, manages their status (occupied or vacant), and counts the number of people in each frame.

- Desk Object: Class responsible for creating and managing desk objects. It tracks start and end times of occupancy, calculates total time spent by individuals at desks, and updates status based on frame counts.

- Database: A file used to store data collected by the system, including occupancy times and desk statuses.

➔ **Algorithmic Flow**

- Initialization:
- Read camera information from a JSON file.
- Extract relevant data such as camera name, height, width, and points for desk division.
- Organize data into dictionaries for easy access.

- Camera Management:
- Manage cameras concurrently using threads.
- Create CameraObject and DetectObject instances for each camera.
- Improve video speed for processing efficiency.
- Store values in dictionaries for communication between components.

- Object Detection:
- Process frames from cameras.
- Utilize a custom-trained YOLOv8s model to detect people.
- Employ object tracking to assign IDs to individuals and track their movements.
- Determine if individuals are within desk regions and update their statuses accordingly.

- Desk Management:
- Detect individuals within desk regions.
- Create and manage desk objects.
- Track occupancy start and end times.
- Update desk status based on individual presence.
- Count the number of people occupying desks in each frame.

- Data Storage:
- Store collected data, including occupancy times and desk statuses, in a database file for future analysis.