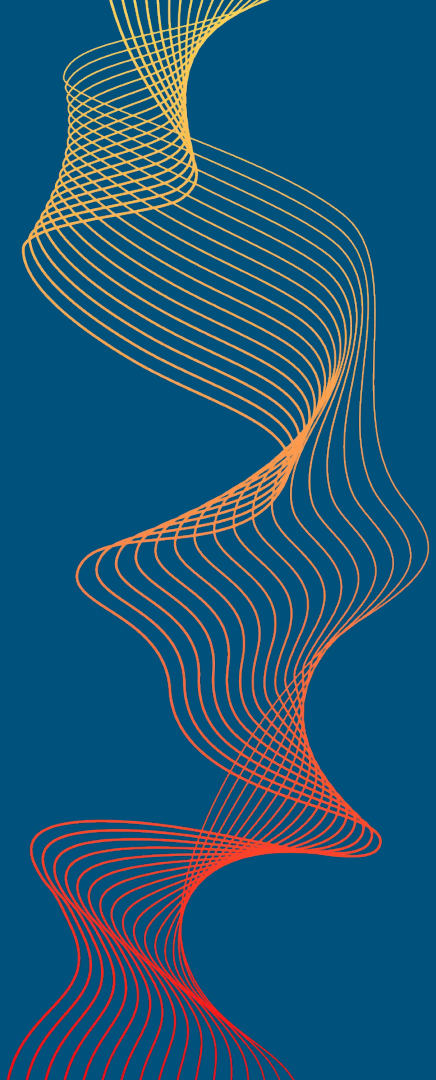
Object Detection and Tracking: Enhancing Video Analysis

A Solution Utilizing YOLOv8 and Advanced Tracking Models for People Counting

Payal Jain

Content

- Object Detection, Tracking and People Counting Workflow
- Model for object detection - YOLOv8
- Model for object tracking - Centroid based
- Model for object tracking - SORT
- Model for object tracking - SDeepSORT
- Enhancements for Accuracy
- Enhancements for Eficiency



Object Detection and Tracking for People Counting

- Object detection and tracking algorithms play a pivotal role in achieving precise people counting:
  - Object Detection: Identifying and localizing individuals within a frame.
  - Object Tracking: Continuously monitoring and assigning unique IDs.
- The synergy between these algorithms enables us to:
  - Overcome occlusions and obstacles.
  - Maintain consistent counts even in dynamic environments.

Workflow

Object Detection → Object Tracking → People Counting

- Different versions of YOLO (You Only Look Once) are available.
- Used YOLOv8 model

- Experimented with :
  - Centroid based object tracking
  - SORT
  - DeepSort
- Used DeepSort in the final solution

- Calculate Centre Co-ordinates of Detected Objects
- Define Region of Interest(ROI)
- Number of IDs - Total people count
- Direction of movement of IDs w.r.t ROI - in/out count

Centroid-based Object Tracking

Tracks objects based on the movement of their central points between frames.

Advantages

01 Simple to implement

02 Faster with 9 FPS

Disadvantages

01 Assigning different IDs to same person takes longer steps

02 Less accuracy when people occluded

SORT(Simple Online and Realtime tracking)

Based on the Kalman filter for state estimation, and employs data association techniques for accurate tracking.

Advantages

01 Better accuracy when people taking longer steps

02 Faster than DeepSort

Disadvantages

01 Slower than centroid based - 6 FPS

02 Suffered reduced accuracy when obscured. Tracking IDs were frequently lost due to occlusion with the gates consequently impacting the accuracy of in/out counts.

DeepSORT

Based on deep learning techniques combined with the SORT algorithm

Advantages

01 Tracks people better when far away.

02 Tracks people better when occluded

Disadvantages

01 HELL SLOW!! (specially for my CPU based system) 3 FPS

Despite being slow, used DeepSORT for my solution since it had better accuracy.

Enhancements

Accuracy

- Fine tune YOLOv8 model with custom annotated data
  - Two people very near to each other
  - People far away from the camera
  - People behind windows
  - People bending
- Implement logic to consolidate the bounding regions of individual, especially when they are identified as two separate entities in crowded scenarios.
- In/Out count logic keeps track of centre points of ids from the ROI whether they fall in positive or negative region of it.
  As soon as the region of the centre point from the desired line changes, the in/out count is updated.
  When dense crowds gather near the frisking area, the center points may inadvertently cross the reference line due to shifting bounding regions, resulting in miscounts.

Efficiency

- Opting for a GPU-based system due to enhanced processing capabilities, as compatibility issues with Python 3.10 in Colab led to numerous dependency errors with our code.
- Using smaller object detection model within deepSORT which would provide satisfactory result but uses less resources
- Utilizing multiple cores of the CPU, you can explore multithreading to parallelize the tracking process across cores.

Thank you for your time and attention 🙂