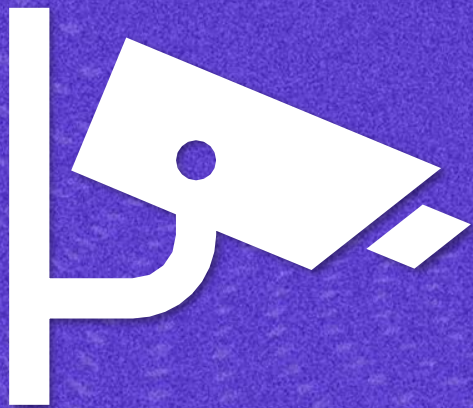




Tailgating Detection

Classy-fiers – IIT Guwahati



Problem Overview

Given CCTV camera footage of a secured door entry, detect anyone tailgating (entry without authorization).

Our Solution

Tailgating Detection Module

2 solutions:

- Mask R-CNN
- YOLO-V3

Intruder Identification Module

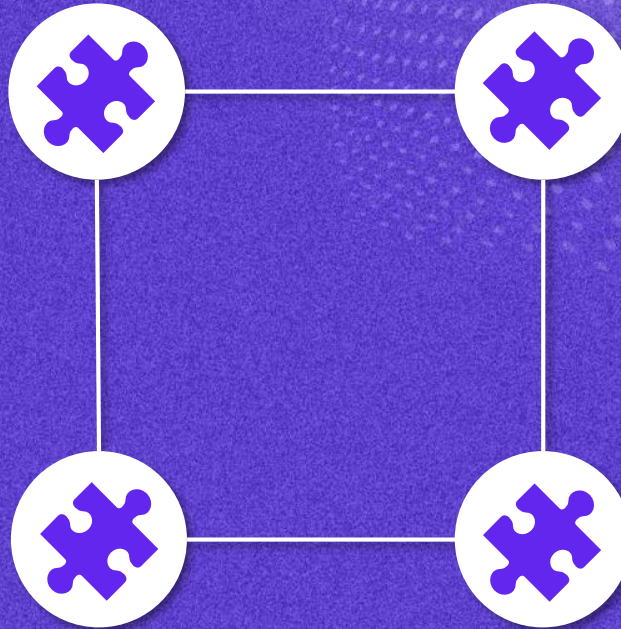
Recording and saving the masks and images of intruders.

Mailing Module

- Send mail
- Trigger an alarm

User Interface

Developed an Interactive User interface for real-time inference.



Developed Systems:

High Power System:

- Model: Mask R-CNN
- Tracking: Centroid Tracking

High Efficiency System:

- Model: YOLO-v3
- Tracking: Deep SORT



This is followed by a common Integrated Alert and Mailing Module.
These approaches will be explained in the following slides.

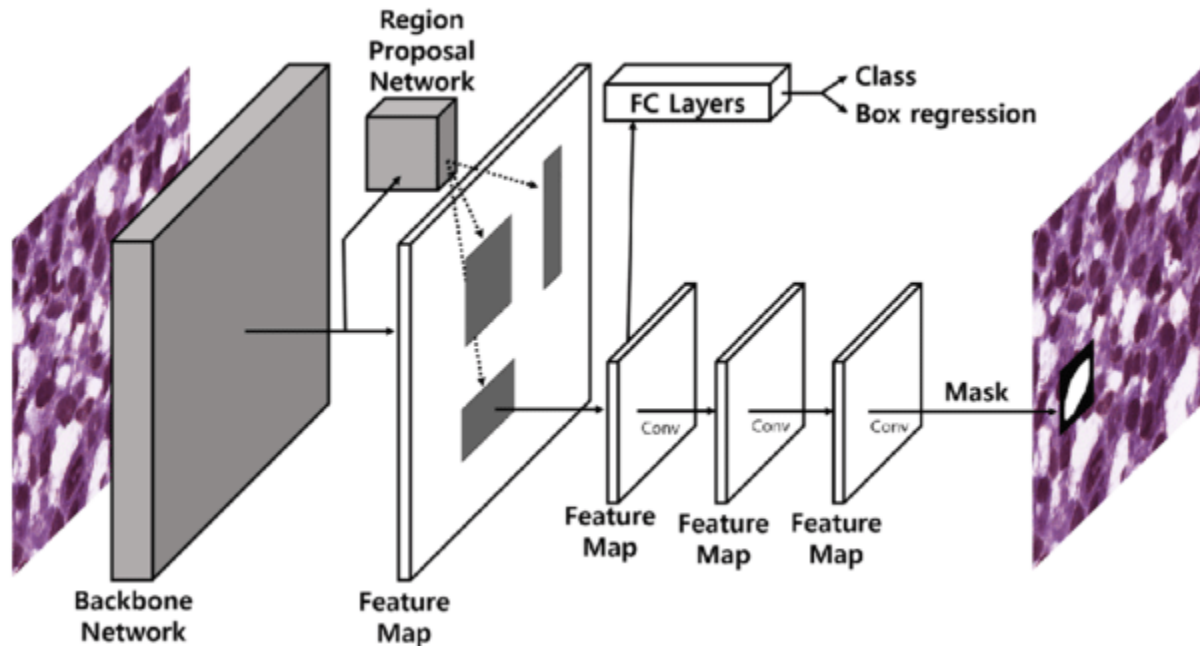


System 1

- High Power System

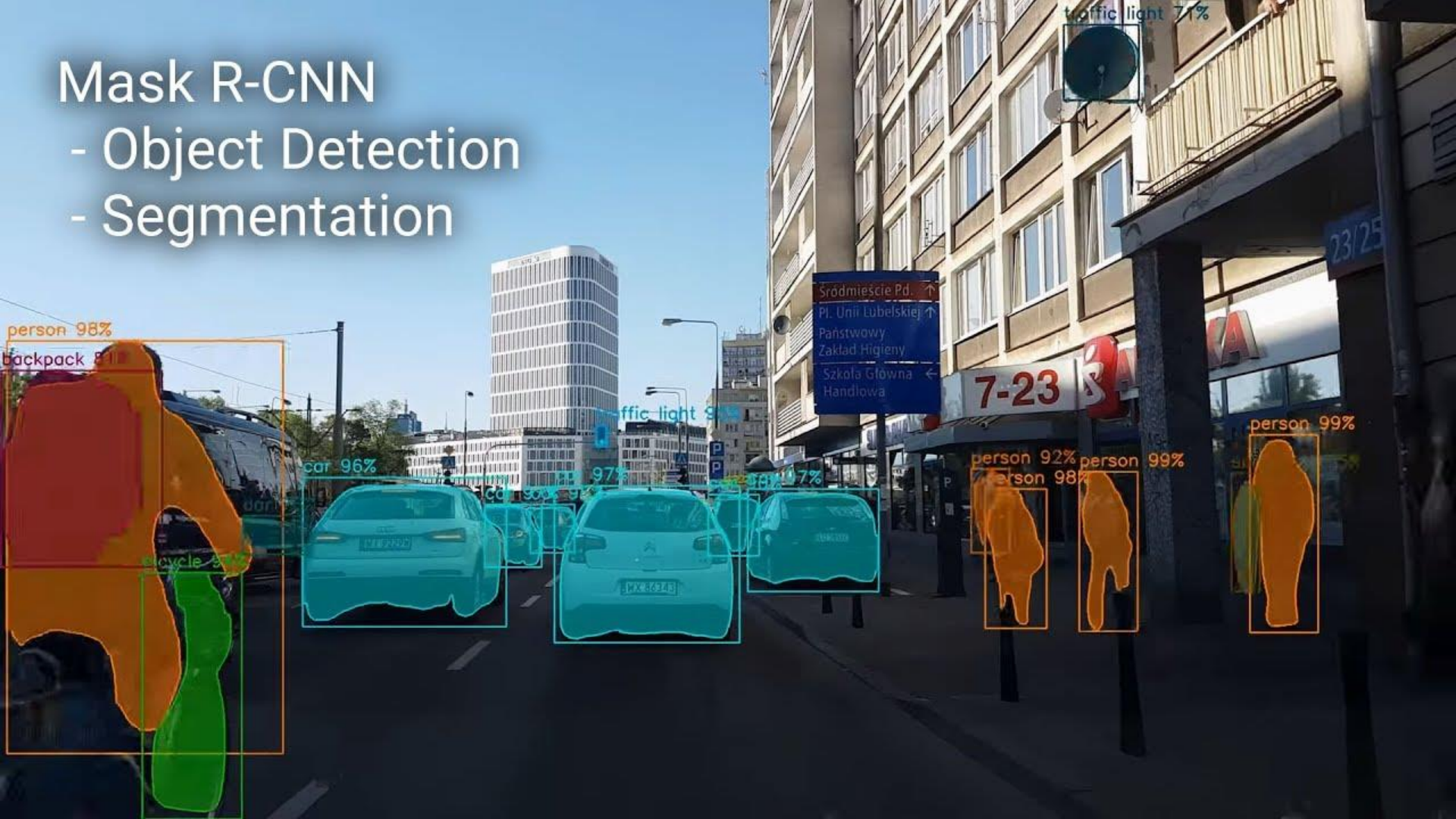
Mask R-CNN

Mask R-CNN is a popular model choice for Instance Segmentation



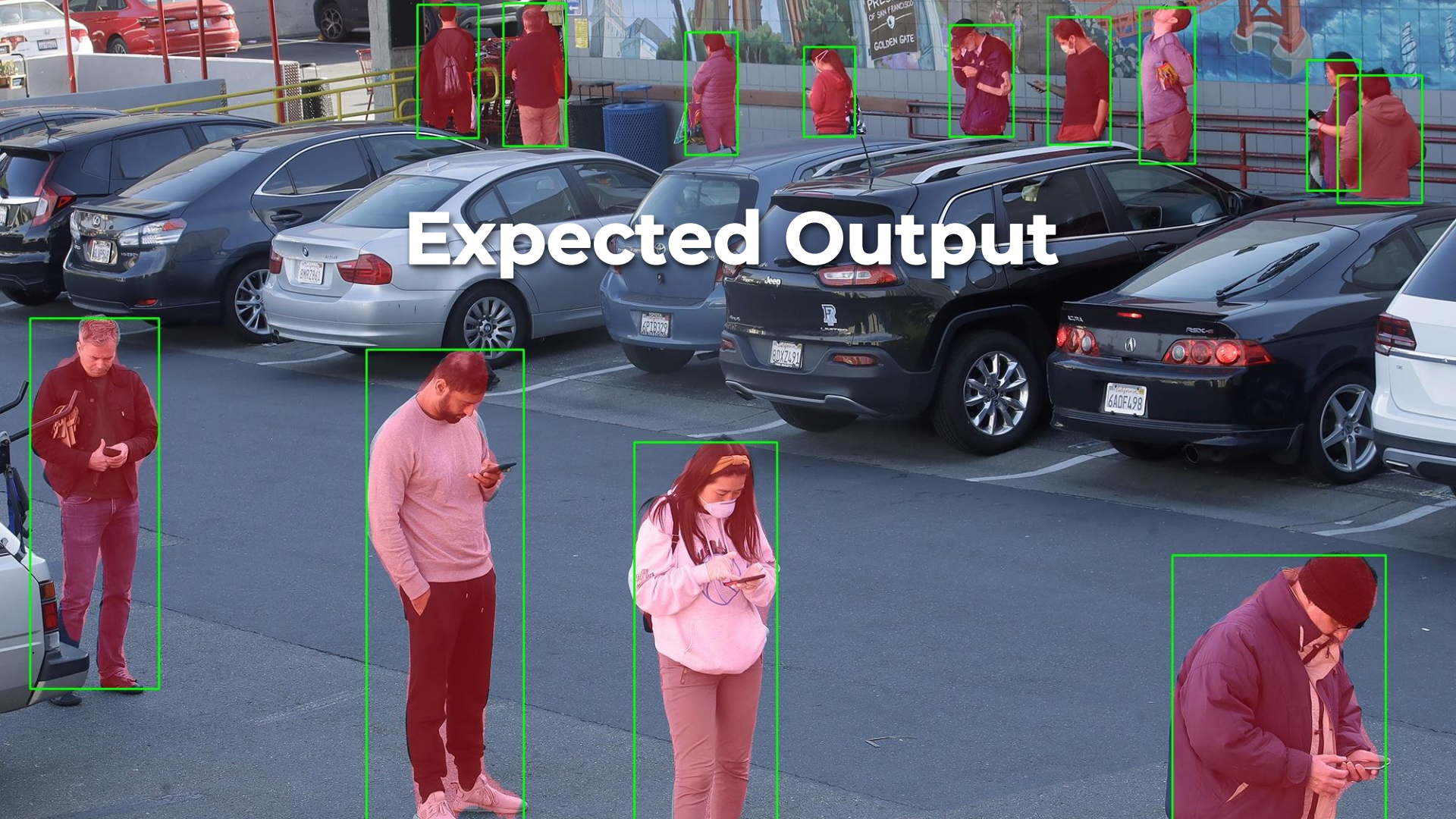
Mask R-CNN

- Object Detection
- Segmentation

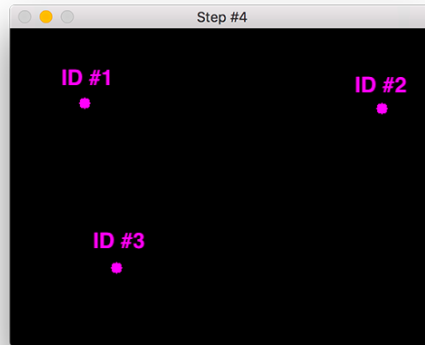
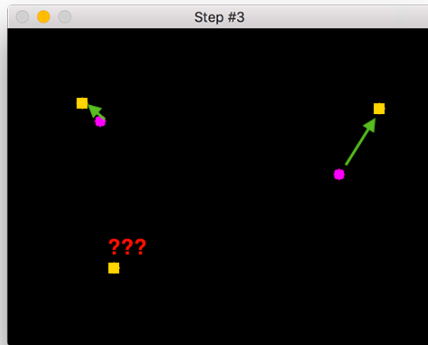
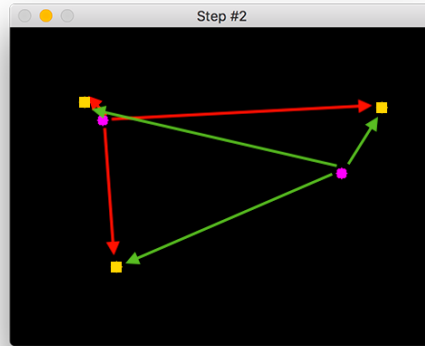
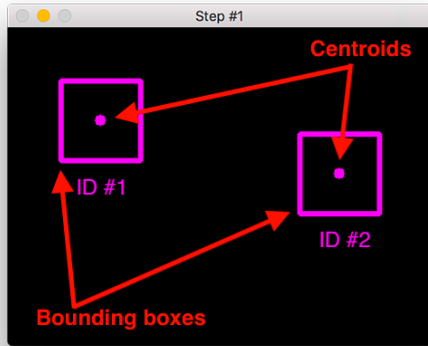


- Mask R-CNN based on a Resnet50 backbone
- A pre-trained Mask RCNN model is used in our project, as it is very tedious to train this architecture by ourselves based on our current computational capabilities.
- The pre-trained weights are trained on the **COCO Dataset**, which consists of around 91 classes. But for our specific problem, we only need Humans as the class/mask output.
- To tackle this problem, we employed **Transfer Learning**, and finetuned the pre-existing architecture to predict only humans. Transfer Learning was achieved using the **Penn Fudan Database**, which is a pedestrian dataset. Hence our model now predicts and outputs only humans, thus reducing computation costs significantly.

Expected Output



Centroid Tracking



- I. Accept bounding box coordinates and compute centroids
- II. Compute Euclidean distance between new bounding boxes and existing objects
- III. Update (x, y) -coordinates of existing objects
- IV. Register new objects
- V. Deregister old objects



person 0

person 1

**Line of
Surveillance**

How can you avoid unauthorized access of a restricted area?

Tailgating Detected



**Centroid
crossed the line**

How can you avoid unauthorized access of a restricted area?

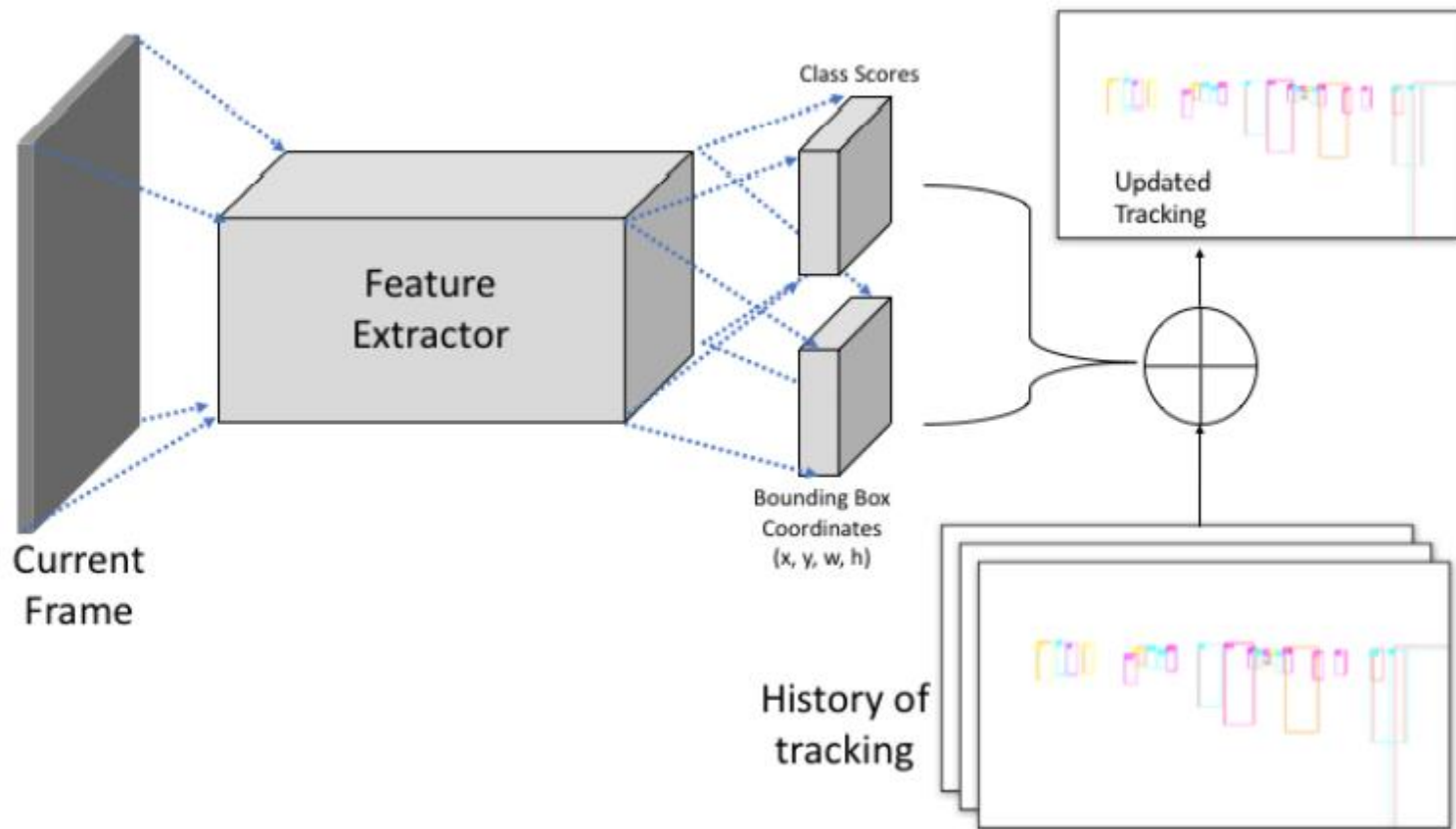
clideo.com



System 2

- High Efficiency System

YOLO-v3 + Deep SORT



In the problem setting of Multiple Object Tracking, each frame has more than one object to track. A generic method to solve this has two steps:

- **Detection**: First, all the objects are detected in the frame. There can be single or multiple detections.
- **Association**: Once we have detections for the frame, a matching is performed for similar detections with respect to the previous frame. The matched frames are followed through the sequence to get the tracking for an object

The Deep SORT tracker uses a vector of states to store the historical information for previous detections. If a new frame comes, we can either use pre-stored bounding box detections or compute them using object detection methods. Finally, using current observation of bounding box detections and previous states, the current tracking is estimated

YOLO-v3 Approach

Tailgating Incident
without Integration Alert

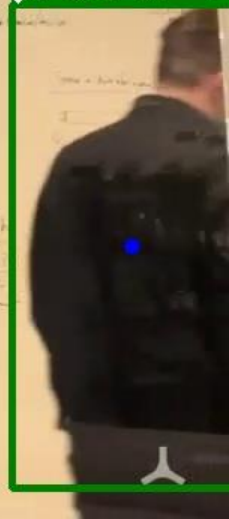


person-3



Region of
Surveillance

person-5



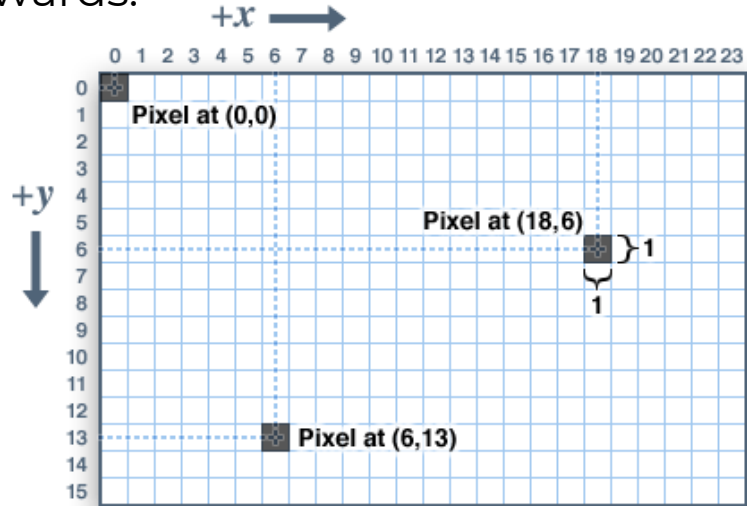
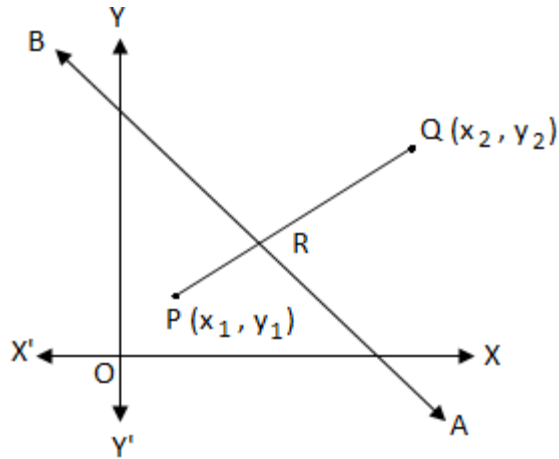
Surveillance Area Logic

In this system, we detect tailgating by counting the number of people crossing the region of surveillance between two consecutive swipes. This is achieved through tracking the respective centroids of the humans detected. Also we take little help from Coordinate Geometry.

Two given points $P(x_1, y_1)$ and $Q(x_2, y_2)$ will lie on the **same** side of the line $ax + by + c = 0$ if $ax_1 + by_1 + c$ and $ax_2 + by_2 + c$ will have **same** signs.

On the other hand, $P(x_1, y_1)$ and $Q(x_2, y_2)$ will lie on the **opposite** sides of the line $ax + by + c = 0$ if $ax_1 + by_1 + c$ and $ax_2 + by_2 + c$ will have **opposite** signs.

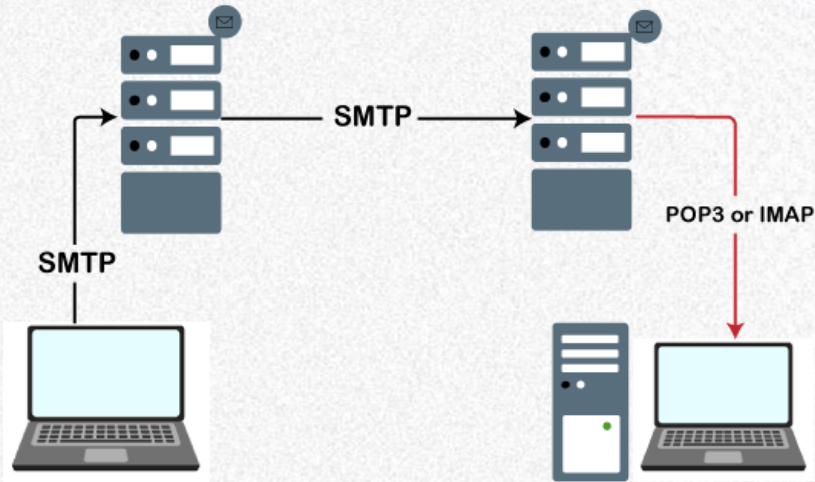
In Python and OpenCV, the origin of a 2D matrix is located at the top left corner starting at $x, y = (0, 0)$. The coordinate system is left-handed where x -axis points positive to the right and y -axis points positive downwards.



Hence, to check if the centroid is between those two lines, we can apply this logic, where one of those two points will be the origin and the other will be the respective centroid.

We have provided the user with two distinct solutions based on their computational capabilities. One approach is an exhaustive end-to-end solution while the other is highly efficient and less computationally expensive.

Mailing Module



- After detecting an intruder, our main objective is to raise an alarm and alert the security personnel regarding the potential security breach.
- This was achieved using SMTP in python. We send a mail to the security department and the person who swiped the card in the first place.
- We also trigger a physical alarm, based on the employers need.

Intruder Identification Module

We maintain a logger file, which would consist of frame/mask of the alleged intruder.

> This PC > Desktop > Projects > VMWare > Project Code > Intruders



Intruder_1



Intruder_3



Intruder_5

Integrated Alert Module

Warning! Tailgating Detected ➤



ameyrambatla@gmail.com

to rishondz ▾

Swiped Card User

4:40 AM (10 hours ago)



Dear Rishon Dsouza,

Real-time Alert

A Tailgating incident was detected behind you at the time 04:40:12. We have attached an image of the possible intruder. You are requested to report at the security frontend immediately.

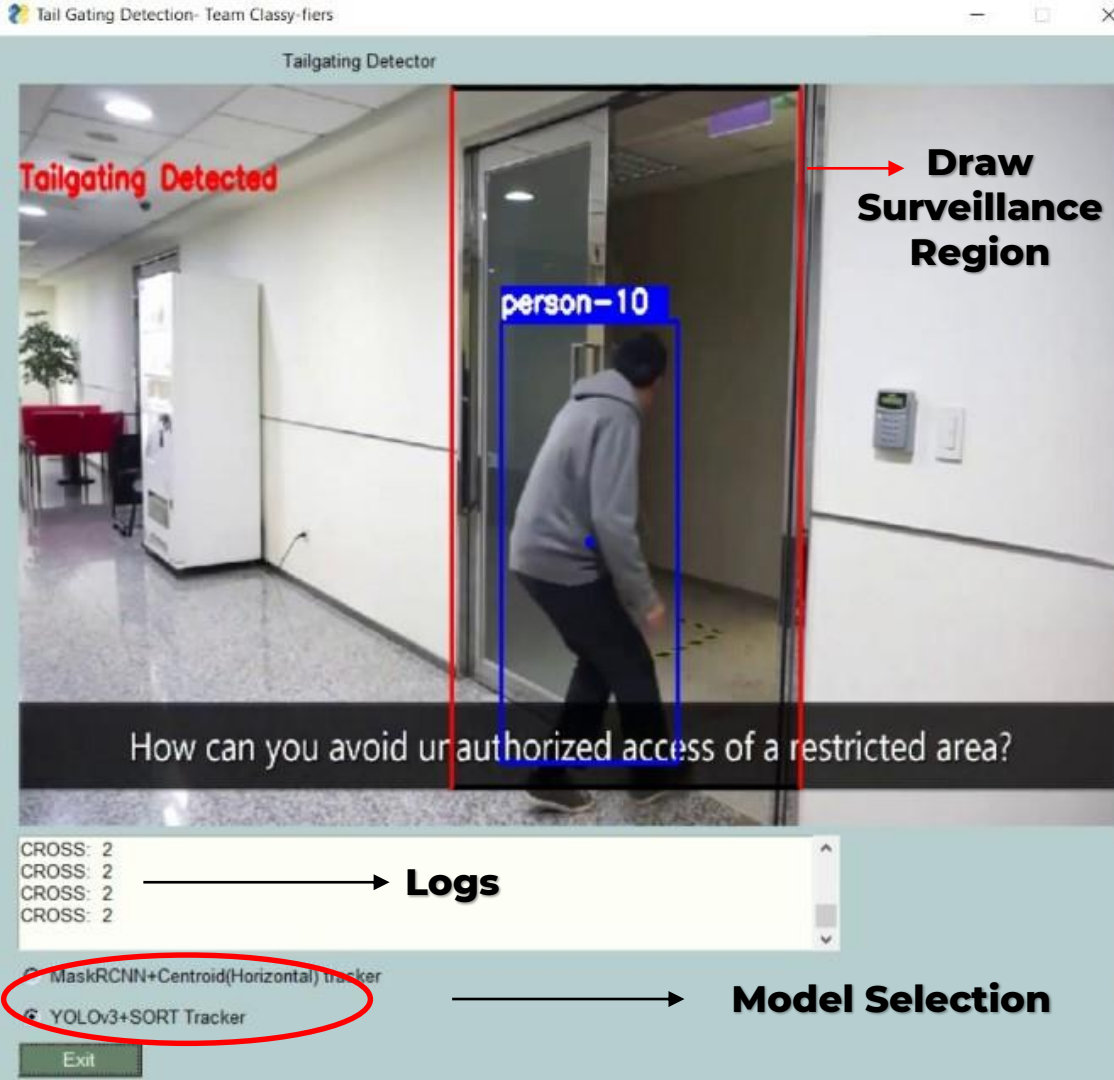
If this was done knowingly, please refrain from doing so and follow the guidelines as laid down by the authority.

Stay Alert.

Regards,
Security Department



Intruder Identification Module



User Interface

To provide the user control over this system, we have developed an Interactive GUI, which provides options such as:

- **Select** required System
- **Draw** the line/region of surveillance w.r.t the camera placement.



Thank You

The Team



**Rishon
Dsouza**



**Amey
Rambatla**



**Aishik
Rakshit**