Smart Surveillance: Real-Time Person Detection

Authored by Sai Bharath, Sai Harsha, Sairam

Introduction:

The aim of this project is to utilize a computer vision model to detect and track individuals in surveillance footage, specifically to monitor and count the number of people entering and exiting a mall. For this project, we used the pretrained YOLO11s model, the latest version in the YOLO (You Only Look Once) family of object detectors. We selected the smaller "YOLO11s" model due to computational constraints and it offered a balanced trade-off between accuracy (mAP validation) and performance speed. Although larger versions of YOLO offer marginal improvements in accuracy, we felt that the increase in processing time makes them less suitable for our project. Our priority is rapid detection, ensuring that the system can accurately and promptly detect people as they enter and exit through an entrance (for ex: shopping mall or subway entrance).

The following image compares different versions within [1] YOLO11 which we used to choose an ideal model for this project.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

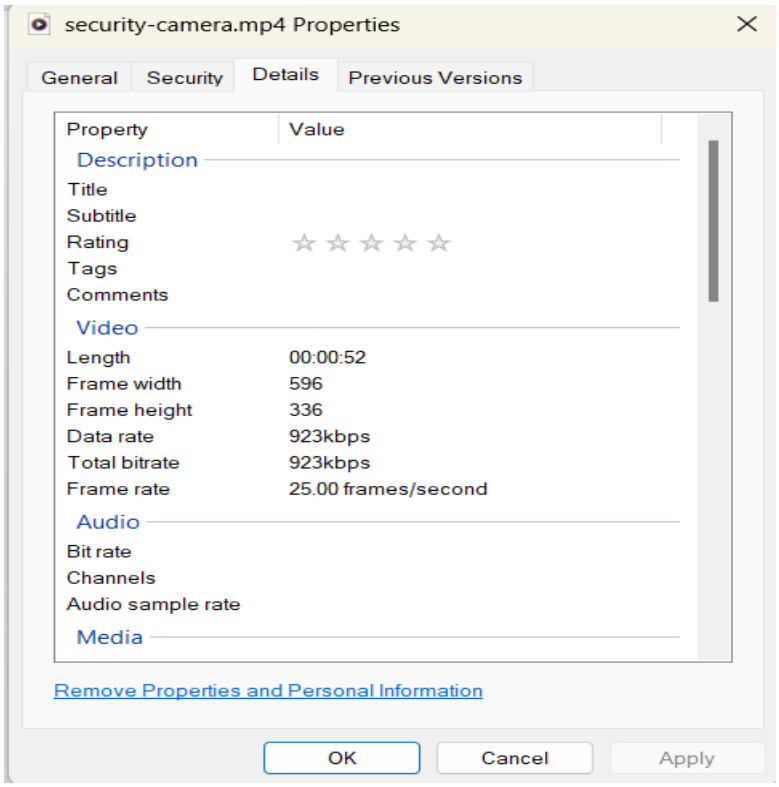
Objective:

Our objective is to count the number of people entering and exiting through an entrance using video captured by a surveillance camera focused on the entrance. This will allow us to track patterns in the number of visitors over daily, monthly, and yearly periods. Additionally, this system can be used to estimate the time individuals spend in a location, such as a shopping mall, providing insights that businesses can leverage to enhance customer experience and optimize product engagement. However, further exploration of these business applications is beyond the current scope and is left for future consideration.

Implementation:

We utilized the **YOLO** model from the **Ultralytics** package for object detection in this project. Our objective was to focus on detecting a single class: **"person"**. For analysis, we used a 52 second long video sourced from [2] Shutterstock, as it closely resembled the surveillance footage typically seen in malls.

The following picture shows the properties of this video.



Considering our computational constraints, we limited our processing to 1001 frames, which corresponds to 40.04 seconds of the 52-second video. This allowed us to maintain a balance between efficiency and accuracy, ensuring that our model performs effectively within the given resource limitations while still capturing meaningful data from the surveillance footage.

```
Number of Frames processed: 1001
```

The model was able to detect the person in the frame with confidence scores ranging between 0.37-0.94. The confidence score kept increasing as the person became more evident in the frame.

```
0: 480x640 1 person, 190.0ms
Speed: 2.0ms preprocess, 190.0ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)
**********Box********** [435, 60, 468, 168]
Confidence Score: 0.3753049373626709
```

```
0: 480x640 1 person, 156.0ms
Speed: 2.0ms preprocess, 156.0ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)
**********Box********** [2, 308, 239, 600]
Confidence Score: 0.9426749348640442
```

Within each frame of the video, we defined two rectangles that serve as key regions for tracking movement. When a person enters any of these rectangles after passing through the other, a bounding box is drawn around them. We track the bottom-right corner of the bounding box to determine whether the person is entering or exiting the mall.

The following is the detailed explanation of our approach.

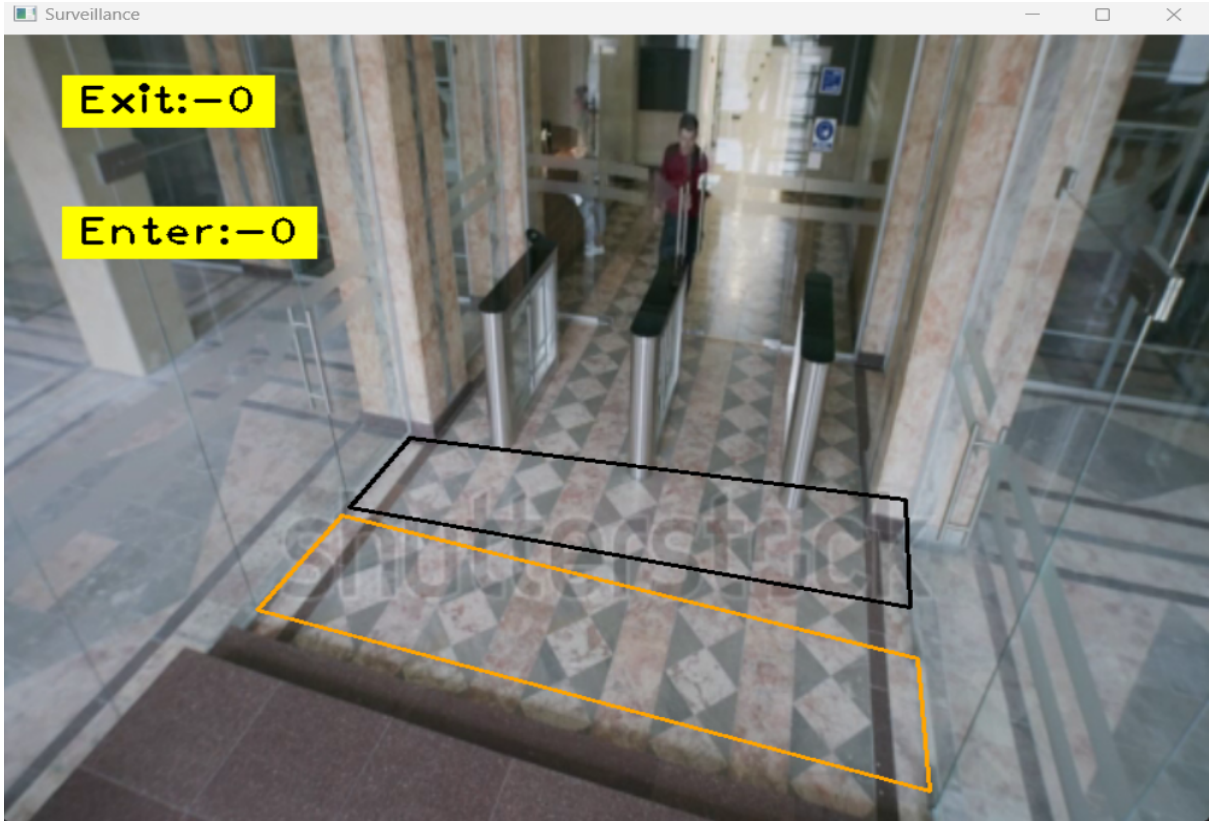
For each frame of the video, we first resize it to dimensions of (800, 600) to standardize input size and improve processing efficiency. This resized frame is then passed to the YOLO model, which performs object detection. When the model detects a "person" class in the frame, we track the bounding box coordinates of the detected person.

Two polygonal areas are plotted on the frame, representing specific entry and exit zones. A bounding box is drawn around a detected person once they enter one of the polygons after having crossed the other. The system tracks the bottom-right corner of the bounding box to determine whether the person has passed through both zones. If a person crosses from one polygon to the other, it allows us to classify their movement as either entering or exiting the mall. If this point moves from "area" to "area1," we classify the person as entering the mall. Conversely, if the point moves from "area1" to "area," we classify the person as exiting the mall.

The following are the coordinates of the rectangle that we defined.

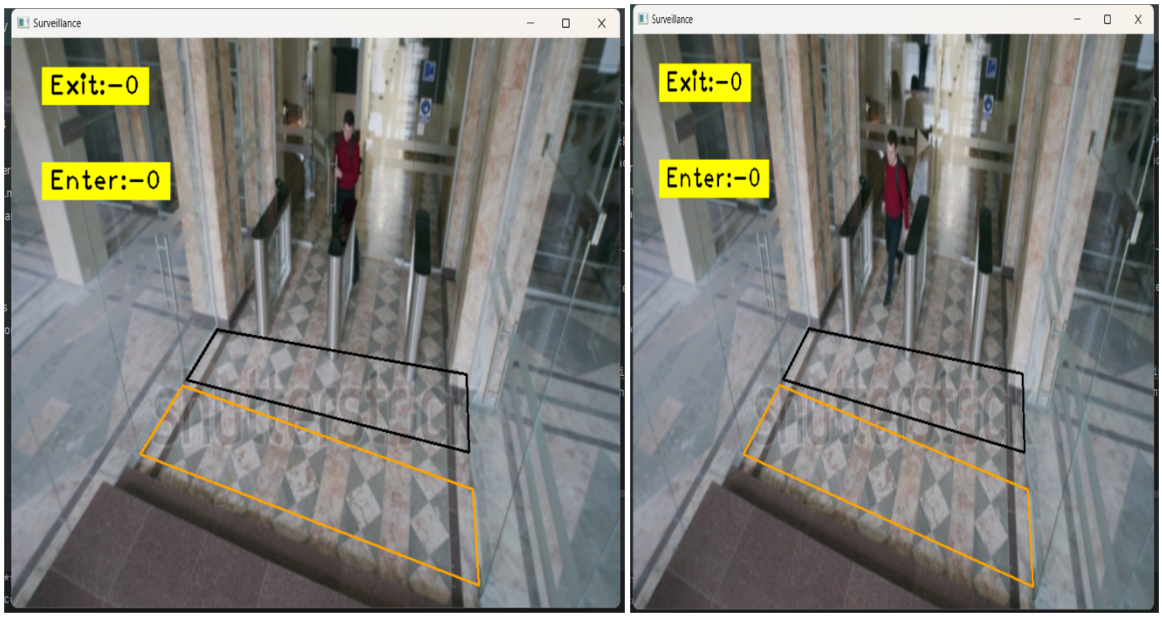
```
area=[(270,307),(230,360),(601,436),(598,354)]
area1=[(225,366),(169,438),(614,576),(606,475)]
```

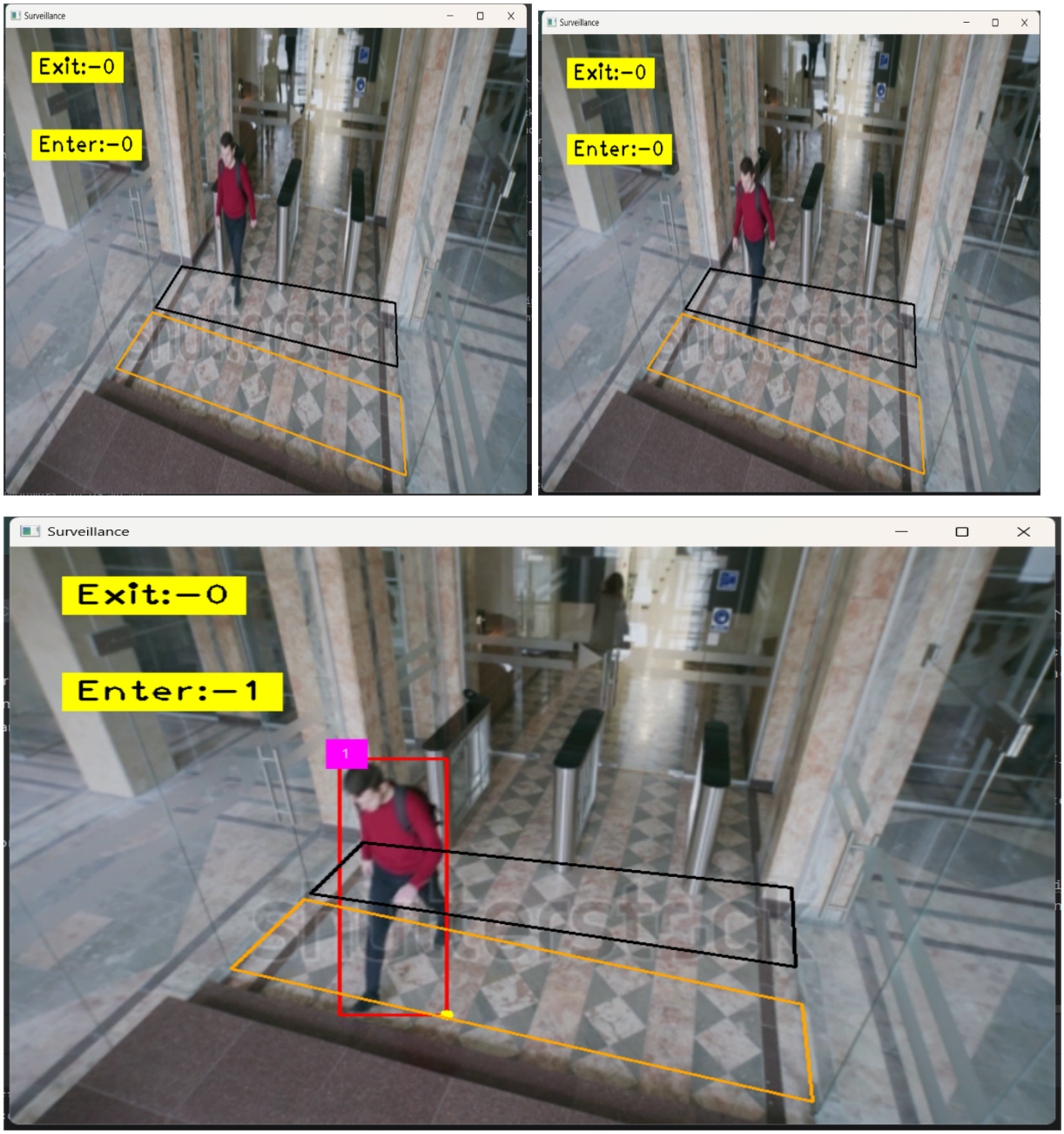
The following figure shows the rectangles that we plotted using the coordinates mentioned above.



The rectangle with a black outline is drawn using the coordinates of "area," while the rectangle with an orange outline is drawn using the coordinates of "area1". If the bottom-right corner of the bounding box enters the black-outlined rectangle first and then the orange-outlined rectangle, the person is classified as entering the mall. Conversely, if the point moves from the orange rectangle to the black one, the person is classified as exiting.

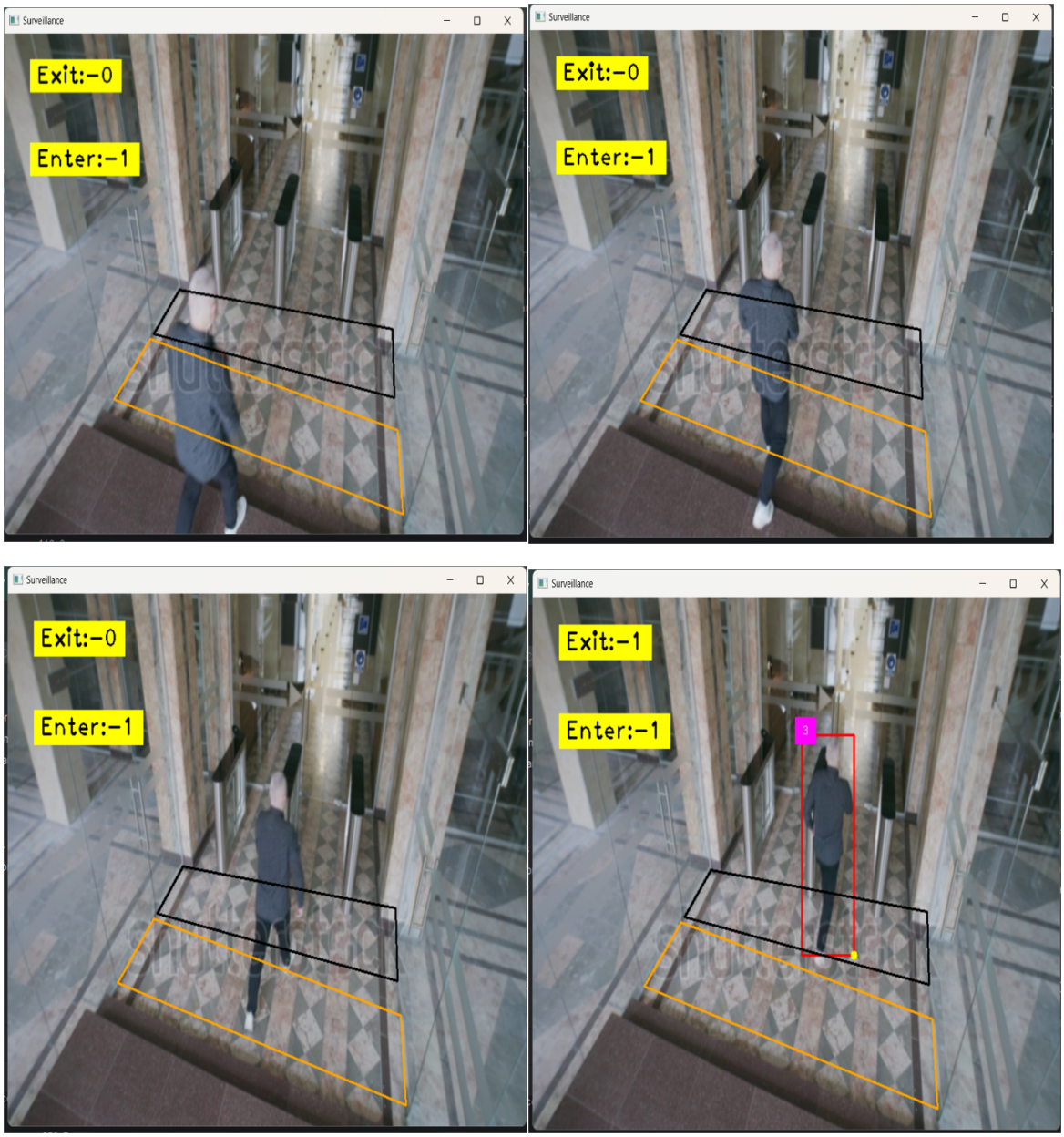
Now let us see a scenario where a person enters the mall.



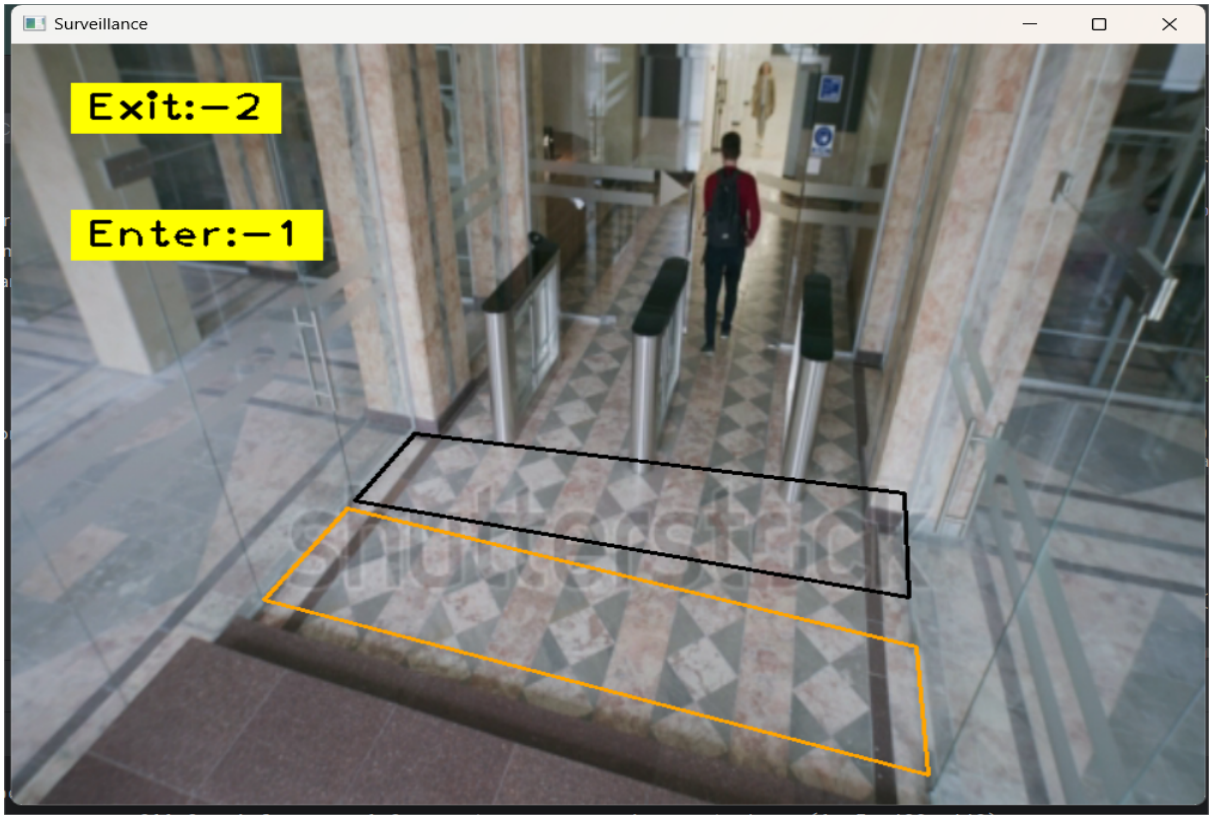


The above series of images illustrate a scenario where a person is entering the mall. Although the model detects the person as soon as they appear in the frame, the bounding box is only displayed once the person crosses the black rectangle and enters the orange rectangle. This is depicted in the last image of the sequence, where the bottom-right corner of the bounding box is highlighted to show how we distinguish between entries and exits. The system checks which rectangle this point touches first. In this case, since the point crossed the black rectangle before the orange one, we incremented the "Enter" count and displayed it.

Let’s see a scenario where a person is exiting the mall



The above series of images illustrate a scenario where a person is exiting the mall. Although the model detects the person as soon as they appear in the frame, the bounding box is only displayed once the person crosses the orange rectangle and enters the black rectangle. This is depicted in the fourth image of the sequence, where the bottom-right corner of the bounding box is highlighted to show how we distinguish between entries and exits. The system checks which rectangle this point touches first. In this case, since the point crossed the orange rectangle before the black one, we incremented the "Exit" count and displayed it.



The above image shows the final count of “Entries” and “Exits”, after processing 40.04 seconds of the video.

Conclusion:

This project successfully demonstrates the use of computer vision techniques, specifically object detection with a YOLO model, to count the number of people entering or exiting a monitored area. While the current implementation effectively detects individuals, it can be extended to support more advanced tasks. For example, as discussed earlier, the system could track individual customers and log their entry and exit timestamps, allowing analysis of their time spent within the mall. This information could offer valuable insights to businesses, enabling them to refine customer engagement strategies and enhance overall experiences.

References:

[1]. https://github.com/ultralytics/ultralytics?tab=readme-ov-file
[2]. https://www.shutterstock.com/video/clip-1099731461-view-security-cameras-entrance-reception-hotel-business