To do this task, you can use a combination of YOLO (You Only Look Once) for real-time person detection and tracking algorithms like SORT (Simple Online and Realtime Tracking) or DeepSORT for tracking people across frames. Here's an outline of the approach:

1. **Initialization:**
   * Import necessary libraries (OpenCV, NumPy, etc.).
   * Load the YOLO model (e.g., YOLOv3, YOLOv4) for person detection.
   * Load the SORT or DeepSORT tracking algorithm.
2. **User Input:**
   * Prompt the user for the choice between unique counts or same person counts.
   * Take user input for the region of interest (ROI) where counting needs to be done.
3. **Processing:**
   * Read frames from a live video stream or a video file.
   * Use YOLO to detect people in each frame.
   * For each detected person, track them using SORT or DeepSORT.
   * Keep track of people entering and exiting the ROI based on their centroids.
   * Update the counts accordingly based on user input.
4. **Output:**
   * Draw bounding boxes around detected and tracked people.
   * Display the live count of people entering and exiting the ROI.
   * Save the processed video with bounding boxes and counts as an mp4 file.
5. **Documentation:**
   * detailed explanation of the logic and algorithms used in the program in a doc file.
   * Include pseudo code to explain the flow of the program.

initialize YOLO model

initialize SORT/DeepSORT tracker

initialize count\_in, count\_out

while video\_stream.is\_running():

frame = video\_stream.read\_frame()

detections = YOLO.detect\_people(frame)

tracked\_objects = tracker.update(detections)

for obj in tracked\_objects:

if obj.enters\_roi():

if user\_input == unique\_counts:

count\_in += 1

else:

count\_in += 2

if obj.exits\_roi():

if user\_input == unique\_counts:

count\_out += 1

else:

count\_out += 2

draw\_boxes(frame, tracked\_objects)

display\_live\_count(count\_in, count\_out)

save\_frame\_to\_video\_output(frame)

save\_processed\_video()