Report final project: People tracking and counting

Clément Amate

February 8, 2025

1 Introduction

In this project, we developed a people tracking and counting system using artificial intelligence and computer vision. The goal is to detect and track individuals in motion to count those who cross a virtual line in a video stream. This type of solution has various applications, such as managing pedestrian flow in public spaces, analyzing customer traffic in retail stores, and improving security and surveillance systems.

To achieve this, we used YOLOv8 models, a deep learning model specialized in real-time object detection and tracking. Unlike traditional frame-by-frame detection approaches, YOLOv8 assigns a unique identifier to each detected person, allowing continuous tracking over time and preventing duplicate counting. All the code used to make this report is available on the project's github repository [1].

This report outlines the system setup, the techniques used for tracking and counting, the experimental results, and potential improvements for future development.

2 People detection and tracking

To accurately count individuals crossing a line, our system must first detect and track them effectively. This section details the technologies and methodologies used to achieve robust real-time tracking.

2.1 YOLOv8 for Detection

YOLOv8 is a state-of-the-art deep learning model designed for fast and efficient object detection. Unlike traditional methods that process images frame by frame without retaining object identity, YOLOv8 can detect and track objects across multiple frames. This capability is essential for counting people while ensuring that each individual is only counted once. In our implementation, YOLOv8 was used to detect humans in each frame, returning the following outputs:

- Bounding boxes: The coordinates defining the detected objects.
- Class labels: The category of the detected object (human, bottle, computer).
- Tracking IDs: Unique identifiers assigned to each detected entities, allowing persistent tracking.

2.2 Tracking individuals

Tracking is a crucial step in distinguishing between new and previously detected individuals. YOLOv8's built-in tracking functionality ensures that each detected person retains a consistent tracking ID across multiple frames. This prevents repeated counts when the same person moves back and forth within the scene. This tracking id can be access by reaching the *id* parameter: *results[0].boxes.id*. Furthermore we define bounding boxes around individuals, the box coordinate can be accessed via the *results[0].boxes* parameter. The label of detected object are accessible with the parameter *results[0].boxes.cls*, in our task only humans are important, we then remove all detected object that have a different label. On YOLO, humans have the label '0'.

[1]Project repository: `GitHub Repository`

Figure 1: Results presentation of human detection. The model shows good accuracy and robustness to other details.

2.3 Counting mechanism

To count people crossing a predefined line, we designed a system that monitors the trajectory of each individual:

- The system stores the previous and current positions of each tracked person.
- If a person moves from one side of the line to the other, they are flagged as having crossed.
- To avoid duplicate counting, we ensure that each tracking ID is counted only once.

In order to do that efficiently, we chose to only consider the center of the bounding box. This method can have weaknesses in diverse situation where humans can be unpredictable. We stores this information in a dictionary with the entity's id. By comparing with the current position and the previous ones we can determine if a person has crossed the line. We do that by validating conditions based on previous and current person coordinate and the line coordinate. We also need to check wether this id is already known and if it has crossed the line. This method ensures no duplicate counting.

3 Results and Performance Analysis

To validate the effectiveness of our people tracking and counting system, we designed a series of experiments focusing on real-time detection performance, accuracy, and robustness.

3.1 Experimental data

Our tests were conducted using a real-time camera from the same computer the code is running. We try our detection system with diverse situation containing one and two humans. These individuals are moving across the camera, crossing each other, and making diverse movements.

3.2 Accuracy of Detection and Tracking

The model achieved a good detection accuracy and managed to detect every individuals seen by the camera (c.f 1). The movements were detected perfectly and the bounding box was fitting the humans. Human crossing each other were not a problem since the model treated them as two different entities with two different id.

3.3 Counting Performance

The counting mechanism was tested in different scenarios with one or two individuals. The results are satisfying since every human crossing the line is counted. The unique id parameter of YOLOv8 counter the duplicates and prevent the same individual to be counted twice if going back and forth.

4 Challenges and Limitations

Despite the promising results, the system encountered several challenges that could impact its reliability in more complex environments.

4.1 Occlusions and Missed Detections

In crowded settings, individuals often overlap, making it difficult to maintain consistent tracking IDs. Partial occlusions led to occasional ID reassignment errors, causing undercounting or duplicate entries.

4.2 Multiple Crossings and Re-identification Issues

The system encountered an ID problem when individuals were leaving the vision's field of the camera and came back. This brief disappearance forced the model to re-assign an ID to the same person leading to a duplicate counting. This problem is due to the difficulty of the task to maintain in memory each ID appearance. Implementing an artificial memory was not effective and lead to the same results. To solve this problem, a better model like YOLOv8l can be used or an another generation of model such as YOLOv11 could be a solution.

5 Conclusion

In this project, we successfully developed a people tracking and counting system using YOLOv8. The model demonstrated good accuracy in detecting and tracking individuals, ensuring reliable counting in defined environments. By leveraging YOLOv8's tracking IDs and bounding box information, we effectively minimized duplicate counts and improved the robustness of our system.

Despite these successes, some limitations remain, particularly in cases of occlusion and re-identification when individuals leave and re-enter the camera's field of view. Future work could focus on enhancing tracking stability by integrating additional techniques such as feature-based re-identification or using more advanced models like YOLOv8l or upcoming versions such as YOLOv11.

Overall, this project highlights the potential of deep learning-based tracking systems for real-world applications, including crowd monitoring, security surveillance, and retail analytics. Further optimizations could make this approach even more adaptable to complex and dynamic environments.