# **AI-Based People Counting System Using Dual-ROI Door Tracking and YOLOv8**

### **Objective:**

To develop a computer vision system that counts the total number of people passing through a door area using YOLOv8 object detection, tracking, and a two-boundary logic to avoid double-counting and missed events.

### **Overview:**

The system detects and tracks people in a video using YOLOv8, identifies when they cross a specific “door region,” and increments a counter. To ensure robustness against partial detections (e.g., a person starting half inside the door), the algorithm introduces an **outer and inner boundary** around the door area.

### **System Design & Methodology:**

#### **1. Model & Detection**

* Used **YOLOv8n (pretrained on COCO)** for person detection (class\_id = 0).
* YOLO’s built-in **tracking (ByteTrack)** keeps consistent IDs for each person across frames.

#### **2. Region of Interest (ROI) Setup**

* The **door** is represented by **two rectangular boundaries**:  
  + **Outer ROI** → defines the entire doorway area.
  + **Inner ROI** → a smaller rectangle placed *inside* the outer one.

Example:  
  
 door\_outer = (85, 45, 220, 290)

door\_inner = (100, 60, 200, 270)

#### **3. Dual-Boundary Logic (Core Innovation)**

This solves the biggest problem in people counting —

“A person might start inside the door or linger around it, causing duplicate or missed counts.”

To fix this:

* The system only counts a person **once** when their center point moves from the **outer box → inner box**, or vice versa.
* This transition ensures:  
  + A **clear directional pass-through**, not just hovering at the boundary.
  + **No double counting**, since one crossing event equals one count.
  + **Reliable detection** even if the person starts inside the door region.

##### **Algorithm Steps:**

1. Track each detected person’s centroid (cx, cy).
2. Check if (cx, cy) lies within **outer** and **inner** boxes.
3. Maintain a state for each person ID:  
   * state = "outside", "between", "inside".
4. If a person transitions:  
   * From **outside → inside** → increment counter (+1).
   * Reset their state to avoid multiple increments per pass.

This logic is both **direction-independent** and **stable under partial visibility**.

### 

### **4. Counting Rule**

The counter increases whenever a tracked person crosses the door region (outer→inner or inner→outer).  
 It tracks **total crossings**, not distinguishing entry vs exit.

### **5. Backend Integration**

* Implemented using **FastAPI**.
* Accepts video uploads via HTTP POST (/count endpoint).

Saves video temporarily, runs counter(video\_path) function, and returns a JSON response:  
  
 {

"status": "success",

“Video\_name” : test2.mp4

"people\_passed": 2

}

### **6. System Architecture**

Video Input

↓

YOLOv8 Person Detection → Object Tracking (ID persistence)

↓

Dual-Box Door Crossing Logic (Outer & Inner ROI)

↓

Count Increments (total passes)

↓

FastAPI JSON Response

### **7. Tech Stack**

| **Component** | **Technology** |
| --- | --- |
| Object Detection | YOLOv8 (Ultralytics) |
| Tracking | Built-in YOLO Tracker (ByteTrack/SORT) |
| ROI Logic | Custom Dual-Boundary Cross Detection |
| Backend | FastAPI |
| Libraries | OpenCV, Ultralytics, FastAPI, Uvicorn |
| Language | Python 3.12 |

### **8. Advantages of Dual-Boundary System**

| **Issue** | **Single ROI** | **Dual ROI Fix** |
| --- | --- | --- |
| Person starts inside the door | Missed count | Detected once they leave inner→outer |
| Person lingers near door | Double count | Counted only on full cross |
| Partial entry or partial exit | Unstable | Requires complete cross of both boundaries |
| Bidirectional flow | Ambiguous | Counts total transitions clearly |

### **9. Applications**

* Smart office entry/exit tracking
* Retail store visitor analytics
* Public transport or corridor flow monitoring
* Automated footfall statistics for events

### **10. Future Enhancements**

* Adaptive ROI based on door detection model.
* Direction-aware counting (entry vs exit).
* Live dashboard with Streamlit or React front-end.
* Real-time edge deployment on NVIDIA Jetson or Raspberry Pi.

### **11. Result**

✅ The dual-boundary system provides stable and accurate total counts even when:

* People start inside the door area, or
* Stand still near the boundary.

This approach eliminates double counting, reduces noise, and achieves **reliable real-world performance** under varied scenarios.

### **12. Sample Outputs**







