Staff Tag Detection System – Documentation

Overview

This project detects people in a video using YOLO, identifies if they wear a staff tag, and saves confirmed staff images into a folder. It supports continuous video processing, tagging, memory-based identity matching, and output logging.

How to Run

1. Place your video into the videos/ folder.
2. Ensure YOLO models are in the project directory.
3. Run the app: python main.py

Dataset Preparation & Verification

Dataset Source

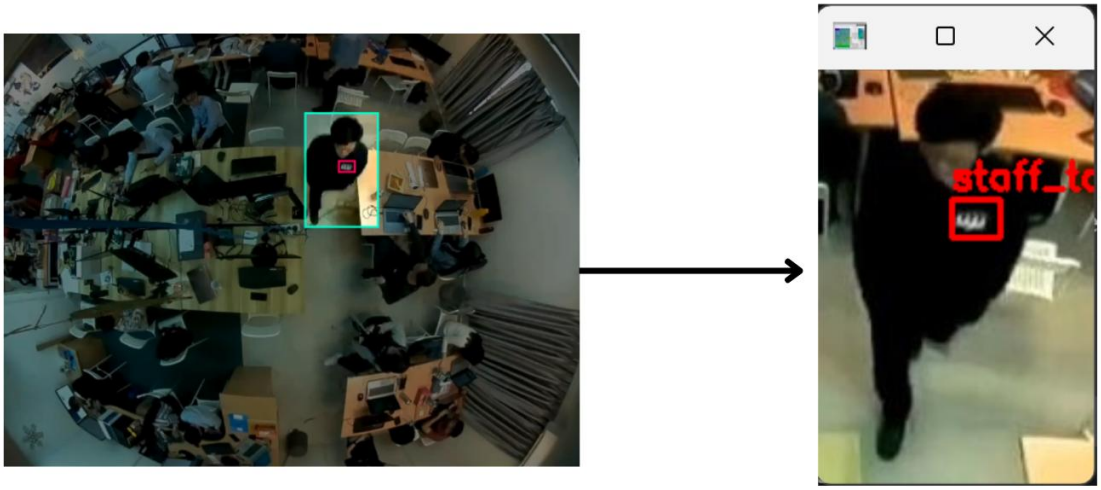
This project uses a custom YOLOv8-compatible dataset originally from Roboflow:
Staff Tag Detection (Roboflow)

Preprocessing Goals

- Keep only images that contain both a person (class 0) and a staff tag (class 1)
- Crop each image based on the person bounding box
- Retain only tag annotations that fall within the cropped person region
- Export a clean dataset for two-stage YOLO detection

Step 1: Dataset Cropping

File: preprocess/crop_staff_dataset.py
Goal: Crop image to the first person (class 0), and keep staff tag (class 1) annotations inside the crop.

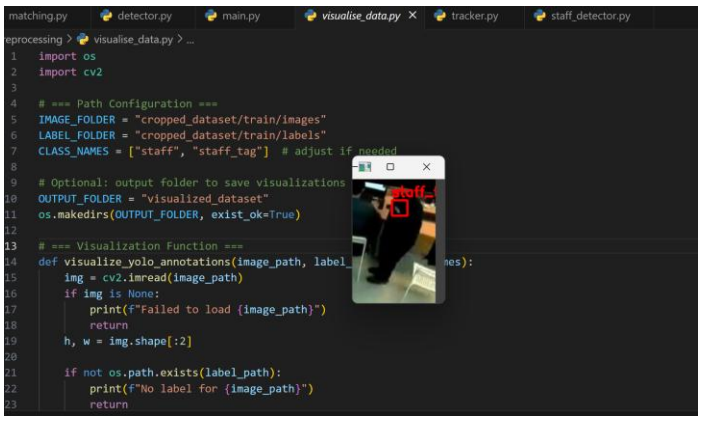


Process Summary:

1. Read image + YOLO label.
2. Find the first bounding box with class 0 (human).
3. Crop image around that bounding box with margin.
4. For each class 1 tag inside the cropped box:
    - Convert coordinates to cropped image format.
    - Save cropped image and adjusted label if at least one tag exists.

Step 2: Visualize Cropped Annotations

File: preprocess/visualize_data.py
Goal: Manually confirm YOLO labels are valid after cropping.

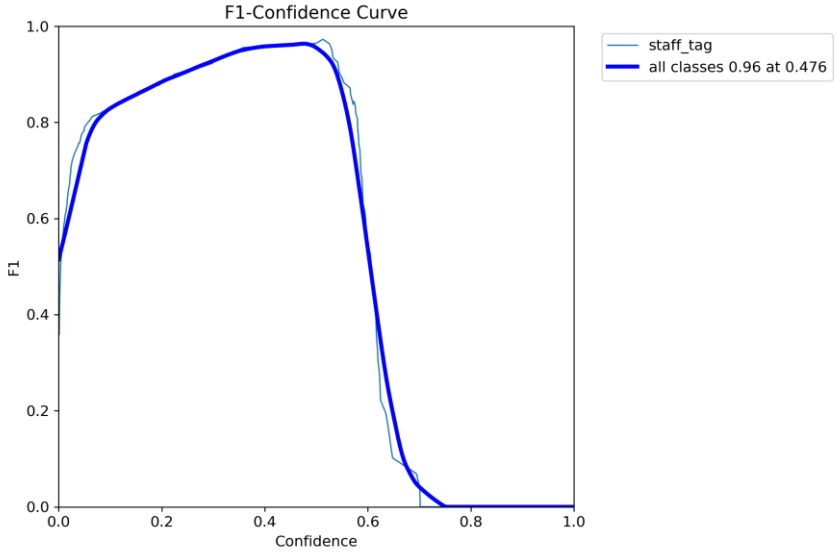


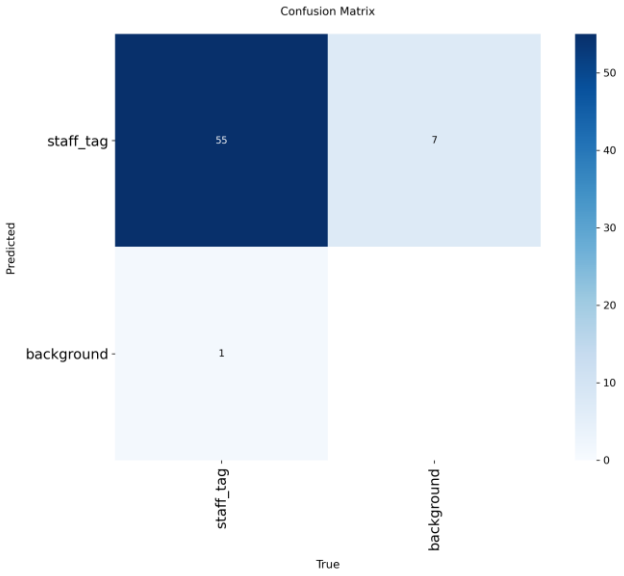
YOLOv12s Fine-Tuning

After preprocessing and cleaning the dataset, the cropped data (person regions with embedded staff tags) is used to fine-tune a YOLOv8s model for detecting staff tags only.

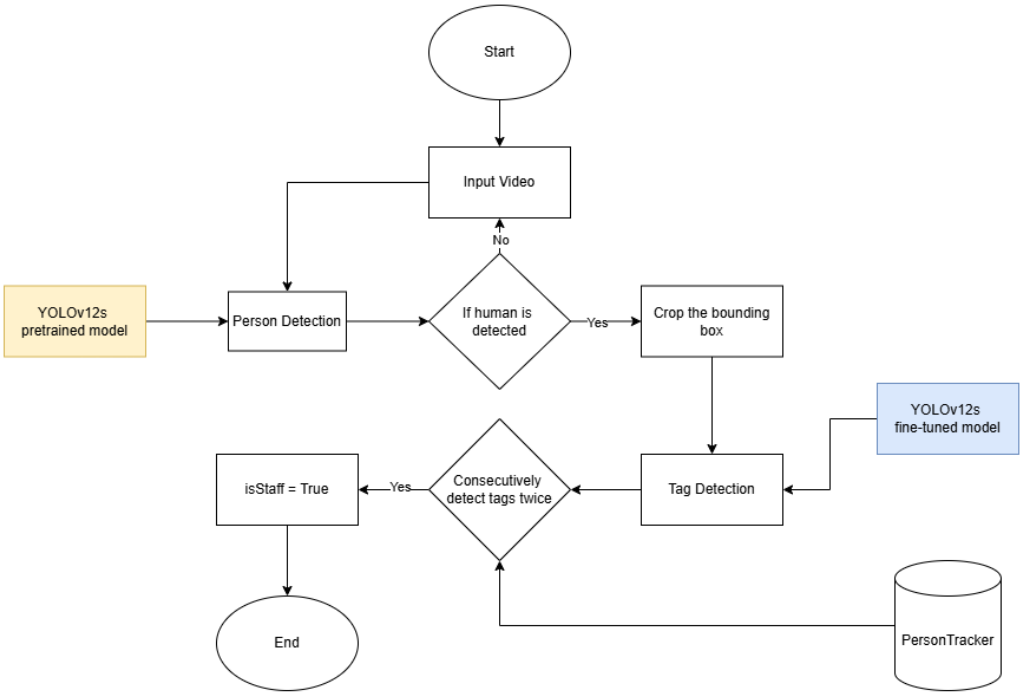
```
100 epochs completed in 0.256 hours.
Optimizer stripped from runs\detect\train5\weights\last.pt, 5.4MB
Optimizer stripped from runs\detect\train5\weights\best.pt, 5.4MB

Validating runs\detect\train5\weights\best.pt...
Ultralytics 8.3.158  Python-3.9.6 torch-2.7.1+cu118 CUDA:0 (NVIDIA GeForce RTX 4060 Laptop GPU, 8188MiB)
YOLOv12n summary (fused): 159 layers, 2,526,971 parameters, 0 gradients, 5.8 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|          | 2/2 [
                   all         56         56      0.962      0.964      0.975      0.397
Speed: 0.2ms preprocess, 2.5ms inference, 0.0ms loss, 0.9ms postprocess per image
Speed: 0.2ms preprocess, 2.5ms inference, 0.0ms loss, 0.9ms postprocess per image
Results saved to runs\detect\train5
```





Flowchart


Start
Input Video
No
YOLOv12s pretrained model
Person Detection
If human is detected
Yes
Crop the bounding box
YOLOv12s fine-tuned model
isStaff = True
Yes
Consecutively detect tags twice
Tag Detection
End
PersonTracker

Module Descriptions

PersonDetector (in detectors.py)

- Loads the YOLOv8 model for person detection.
- Outputs bounding boxes for class person (class 0).

TagDetector (in detectors.py)

- Loads the trained model for detecting staff tags.
- Filters predictions based on confidence (default 0.56).
- Returns tag_found: bool and confidence: float.

PersonTracker (in tracker.py)

- Maintains a memory of previous person positions (cx, cy).
- Matches new detections with old ones using distance threshold.
- Uses a hit counter to confirm staff tags over multiple frames.

StaffDetector (in staff_detector.py)

- Orchestrates the detection pipeline.
- Reads frames, detects people, checks for tags.
- Annotates the video and saves staff crops to /staff.
- Logs tag scores in tag_scores.txt.

Outputs

- output/output_video.mp4 – video with bounding boxes + labels.
- output/detected_staff_dataset/– cropped images of detected staff.
- output/tag_scores.txt – line-by-line log of tag detection scores:

Future Improvements (Suggestions)

- Save to CSV/JSON with IDs and timestamps
- Add ByteTrack or DeepSORT for long-term ID tracking