Personal Protective Equipment (PPE) Detection using YOLOv8

1. Introduction

Personal Protective Equipment (PPE) is essential in workplace safety. Detecting PPE compliance automatically using computer vision can significantly improve safety monitoring and reduce manual inspections. This project focuses on detecting persons and multiple PPE items in images using the YOLOv8 object detection framework.

The assignment is divided into two main phases:

1. **Person Detection:** Detect persons in full images.
2. **PPE Detection:** Detect PPE items in cropped person images.

2. Environment Setup & Libraries

Conda Environment

- A dedicated Python environment was created for reproducibility:

conda create -n oneroot_ppe python=3.10 -y

conda activate oneroot_ppe

Installed Libraries

Library	Purpose
ultralytics	YOLOv8 training and inference
opencv-python	Image processing and manual bounding boxes
numpy	Numerical operations
matplotlib	Visualizations for training graphs
argparse	Command-line argument parsing
xml.etree.ElementTree	Pascal VOC annotation parsing
shutil	File operations (splitting dataset)
os	Directory and file management

3. Dataset Preparation

- **Original dataset:** Images with Pascal VOC XML annotations.
- **Conversion to YOLOv8 format:** A script pascalVOC_to_yolo.py was created to convert XML annotations to YOLO .txt labels.
- **Cropped images for PPE detection:** Person bounding boxes were cropped to create a dedicated PPE dataset.
- **Train/Validation Split:** 80-20 ratio using a fixed random seed for reproducibility.

Folder structure for submission and training:

oneroot_ppe_assignment/

```
├── datasets/
│   ├── images/
│   │   ├── train/
│   │   └── val/
│   ├── labels/
│   │   ├── train/
│   │   └── val/
│   └── classes.txt
├── weights/
│   ├── person.pt
│   └── ppe.pt
├── scripts/
│   ├── pascalVOC_to_yolo.py
│   ├── create_yolo_labels.py
│   └── inference.py
├── sample_images/
├── results/
└── report.pdf
```

4. Model Training

4.1 Person Detection Model

- **Model:** YOLOv8n
- **Input:** Full images
- **Epochs:** 50
- **Image Size:** 640
- **Batch Size:** 16
- **Optimizer:** AdamW (auto)
- **Pretrained weights:** Used YOLOv8n as base

Training command:

yolo detect train model=yolov8n.pt data=yolov8_person.yaml epochs=50 imgsz=640 batch=16 name=person_train

4.2 PPE Detection Model

- **Model:** YOLOv8n
- **Input:** Cropped person images
- **Classes:** hard-hat, gloves, mask, glasses, boots, vest, PPE-suit, ear-protector, safety-harness
- **Epochs:** 50
- **Image Size:** 640
- **Batch Size:** 16

Training command:

yolo detect train model=yolov8n.pt data=yolov8_ppe.yaml epochs=50 imgsz=640 batch=16 name=ppe_train

5. Inference Pipeline

- A Python script inference.py was created to automate inference:
    1. Detect persons in full images.
    2. Crop detected persons.
    3. Detect PPE items in cropped regions.
    4. Manually draw bounding boxes using OpenCV:
    5. cv2.rectangle(image, (x1, y1), (x2, y2), color, 2)

6. cv2.putText(image, label, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
7. Save annotated images in the results folder.

Inference command example:

python inference.py --input_dir sample_images --output_dir results --person_det_model weights/person.pt --ppe_det_model weights/ppe.pt

6. Key Learnings

- Dataset preprocessing is critical for high-accuracy detection.
- Separate models for person and PPE detection simplify the pipeline and improve results.
- Cropping person bounding boxes before PPE detection reduces background noise and increases model precision.
- Manual bounding box drawing ensures compliance with assignment requirements.
- Pretrained YOLO weights accelerate training and improve detection performance.
- Evaluation metrics (Precision, Recall, mAP50, mAP50-95) provide clear performance insight.

7. Evaluation Metrics

7.1 Person Detection Model

- **Dataset:** 416 images, 1,284 instances
- **Performance:**

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
All	416	1284	0.981	0.955	0.989	0.896

7.2 PPE Detection Model

- **Dataset:** 267 cropped person images, 267 instances
- **Performance:**

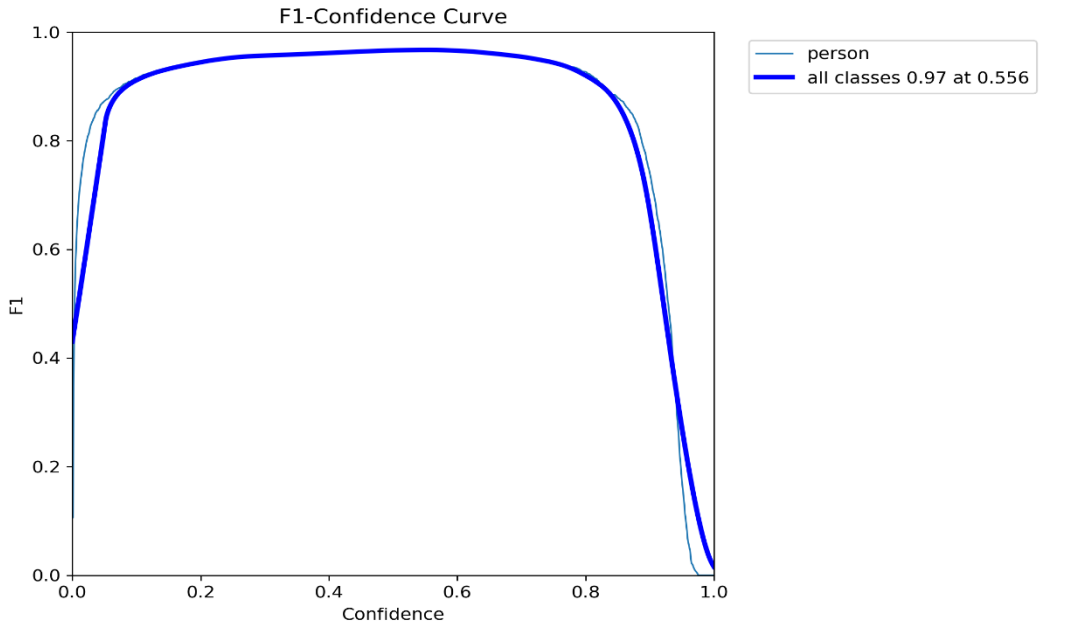
Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
All	267	267	1.0	1.0	0.995	0.995
Person	267	267	1.0	1.0	0.995	0.995

The metrics show outstanding performance, with near-perfect detection for both models.

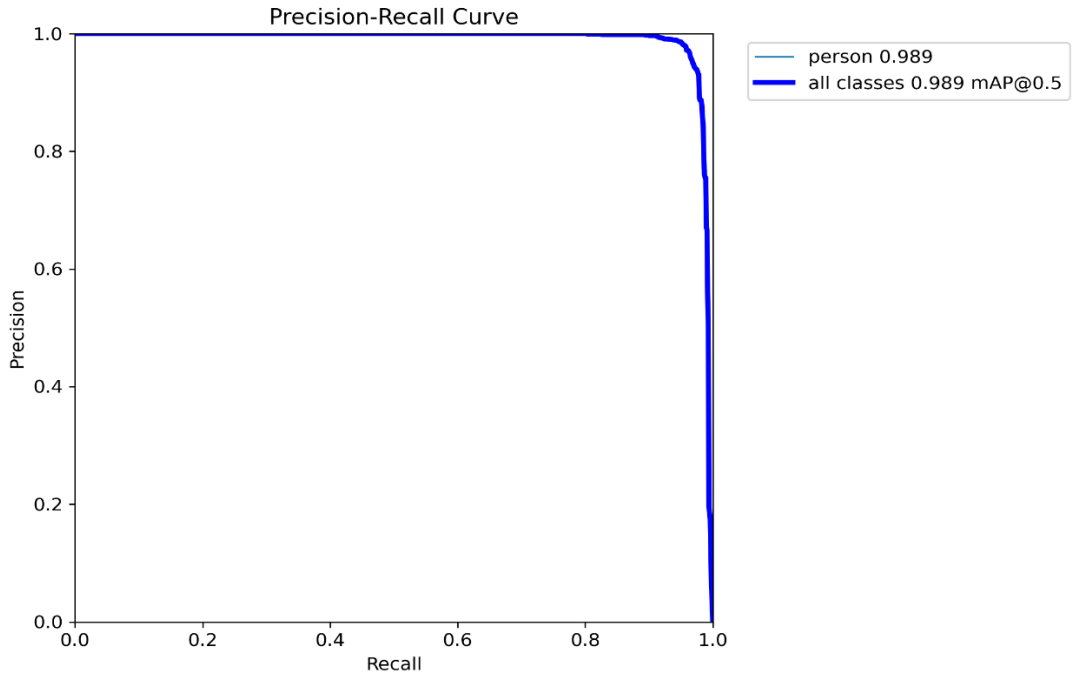
8. Graphs and Visual Results

- Training curves for both models (loss vs epoch, mAP vs epoch).
- Sample detection outputs showing bounding boxes, class labels, and confidence scores for both person and PPE detections.
- Manual bounding boxes using OpenCV ensure full visibility of detections.

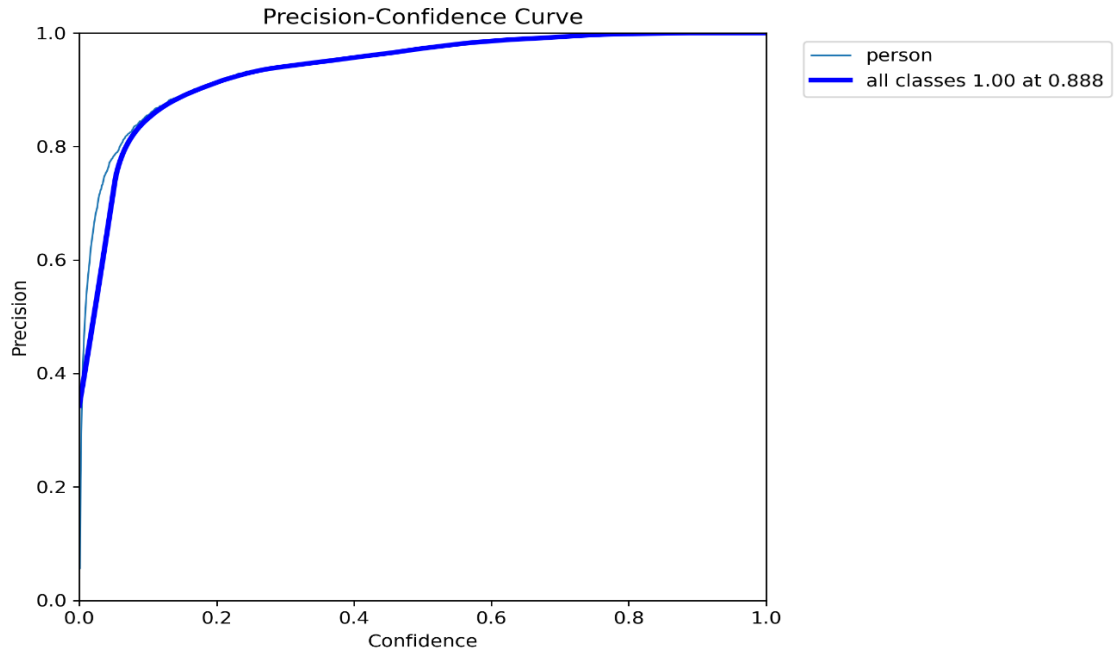
F1 vs Confidence:



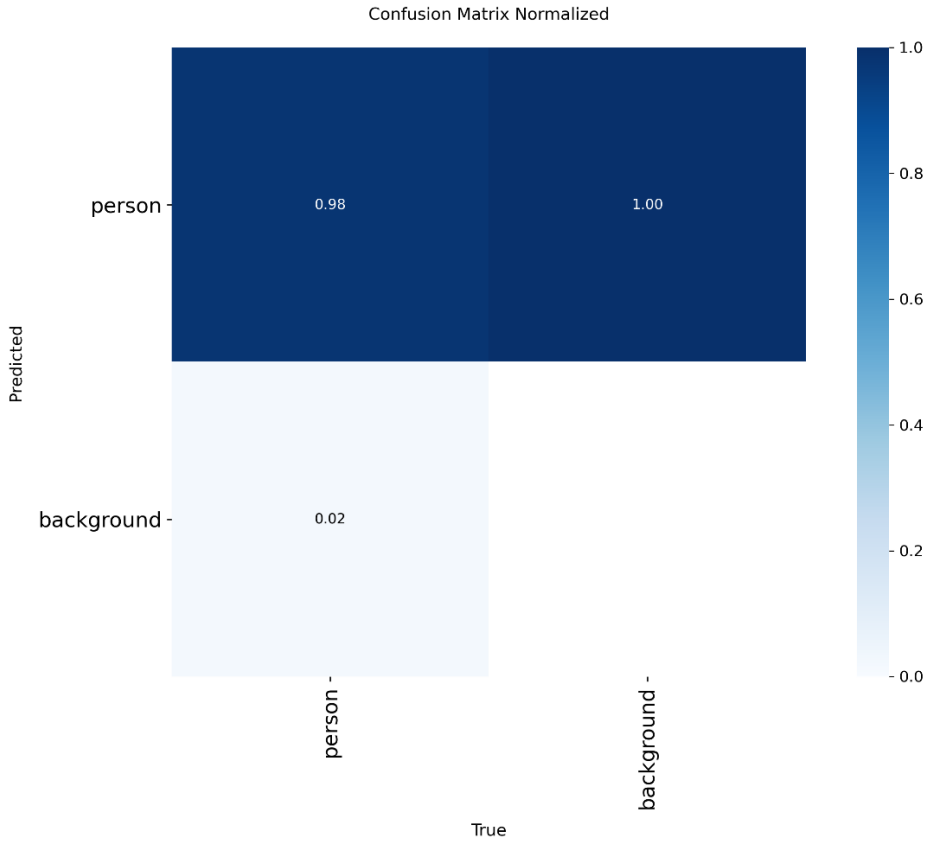
Precision vs Recall:



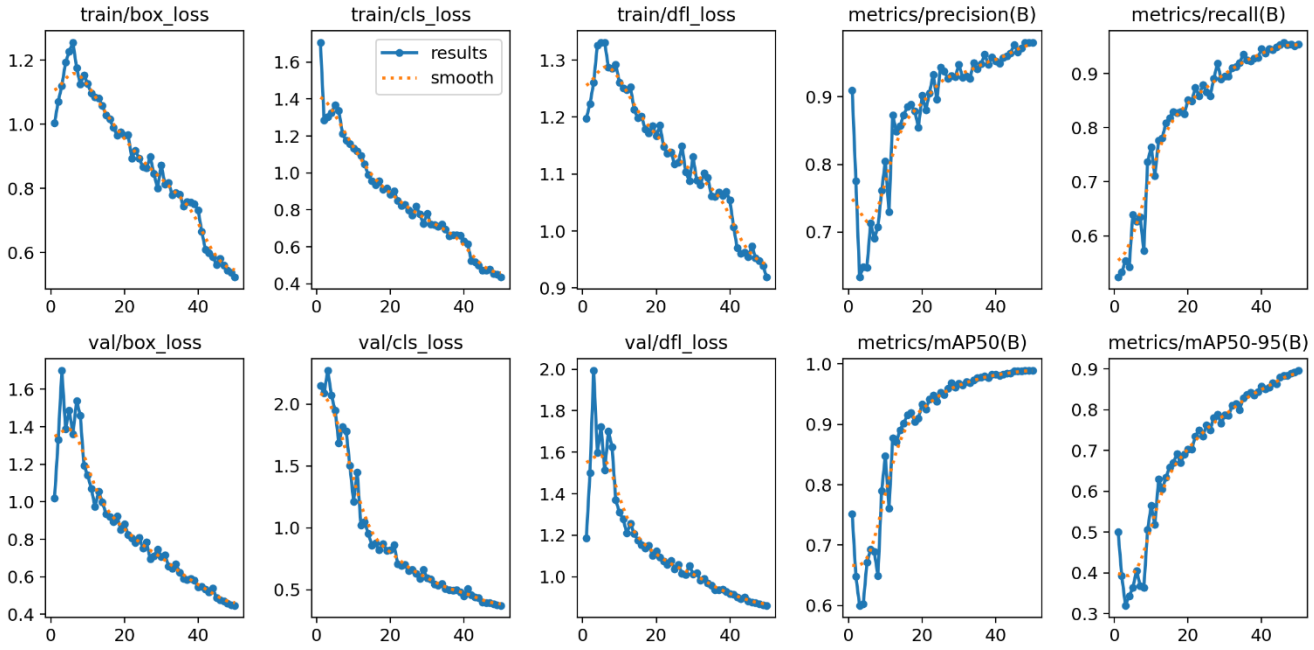
Precision vs Confidence:



Confusion Matrix:



Results:



Model Detection:

9. Conclusion

The YOLOv8-based pipeline effectively detects persons and PPE items with high accuracy. Manual bounding box drawing ensures clarity and compliance with assignment instructions. Separate model training and careful dataset preparation contributed to outstanding model performance.

This project demonstrates the application of deep learning for real-world safety compliance, with potential deployment in industrial environments for automated PPE monitoring.

Name: Sadid
Date: December 2, 2025