“Real-Time PPE Compliance Detection using YOLOv8 and Jetson Orin Nano”

ECHS End-Sem Jury Project Report

Class: LY-AIEC-1

Group ID: ECHS25-04

Group Members:

Wani Tantarpale MITU22BTCS1006

Sujal Bhogade MITU22BTCS0871

Syna Bernard MITU22BTCS0894

Tashu Dhote MITU22BTCS0916

“Real-Time PPE Compliance Detection using YOLOv8 and Jetson Orin Nano”

1. Introduction

Personal Protective Equipment (PPE) compliance is a critical aspect of workplace safety, especially in industrial and construction environments. Ensuring that workers wear safety helmets, vests, gloves, and boots reduces the risk of injuries and fatalities. However, manual supervision of PPE compliance is inefficient, labor-intensive, and prone to human error. Automating this process with real-time computer vision systems can significantly enhance workplace safety monitoring.

This project presents a real-time PPE compliance detection system based on YOLOv8, a state-of-the-art deep learning model for object detection. The system identifies whether workers are wearing required PPE gear such as helmets and vests using live video feeds. The trained model is deployed on an NVIDIA Jetson Orin Nano (8 GB) developer kit for edge inference, utilizing an RTSP camera feed for real-time detection.

2. Literature Review

Author / Year	Approach	Highlights	Limitations
R. Singh et al. (2022)	CNN-based helmet detection	Basic CNN architecture for safety gear detection	Limited to small dataset and low accuracy
A. Sharma et al. (2023)	Transfer learning with YOLOv5	Good accuracy on PPE dataset	Requires optimization for edge devices
Roboflow PPE Dataset (2023)	Community-annotated PPE images	Includes helmets, vests, gloves, and boots with consistent labels	May require retraining for specific environments
Ultralytics (2024)	YOLOv8 object detection model	Supports real-time detection with high precision	Requires TensorRT optimization for embedded deployment

Earlier works focused on helmet detection or limited PPE categories, often using smaller models or non-optimized architectures. YOLOv8 provides a balance between speed and accuracy, making it ideal for real-time PPE compliance detection on Jetson Orin Nano.

3. Methodology

3.1 Dataset

- Source: Roboflow PPE Detection Dataset
- Total images: ≈ **800**
- Number of classes: **7**
- Data split: **80% train, 10% validation, 10% test**
- Format: **YOLOv8 annotated dataset (.txt bounding boxes, .yaml metadata)**

3.2 Proposed System

The system pipeline consists of the following stages:

1. Data Preparation:

- Dataset downloaded and verified from Roboflow with augmentations (flips, rotations, HSV shifts).

2. Model Training:

- Framework: Ultralytics YOLOv8
- Model variant: YOLOv8-Small (v8s) for improved accuracy
- Optimizer: SGD with cosine learning rate scheduler
- Epochs: ≈ 25
- Image size: 600 × 600 px

3. Model Export & Deployment:

- Best weights (`best.pt`) exported to **ONNX/TensorRT**

- Deployed on **Jetson Orin Nano 8 GB**
- Real-time inference using **TP-Link Tapo C200** IP camera via RTSP

4. Real-Time Detection:

- RTSP stream processed by Jetson Nano
- Detected class and confidence displayed on frame
- Optionally send classification results to a local dashboard or actuator

3.3 Block Diagram

DATA ACQUISITION MODULE
(Roboflow dataset / RTSP IP camera)

▼

TRAINING & MODEL CREATION
(YOLOv8 on Google Colab)

▼

MODEL OPTIMIZATION
(ONNX / TensorRT Conversion)

▼

EDGE DEPLOYMENT (Jetson Orin)
+ RTSP Video Inference Pipeline

▼

OUTPUT: PPE COMPLIANCE STATUS
(Visual overlay / alert mechanism)

3.4 Working Principle

- The RTSP IP camera captures live video streams from the work environment.
- The YOLOv8 TensorRT-optimized model deployed on Jetson Orin Nano performs real-time PPE detection.
- Detected objects (helmet, vest, gloves, boots) are displayed with bounding boxes and confidence scores.
- Non-compliance alerts (e.g., missing helmet or vest) can be logged or sent to a dashboard for safety monitoring.

3.5 Jetson Orin Nano Deployment Setup

Hardware Configuration

- **Board:** NVIDIA Jetson Orin Nano 8 GB Developer Kit
- **Storage:** 512 GB NVMe SSD (used as primary boot and storage device)
- **Camera:** TP-Link Tapo C200 IP Camera (connected via RTSP stream)
- **Network:** Gigabit Ethernet / Wi-Fi for remote access and RTSP streaming
- **Power Supply:** 5 V ⎓ 4 A USB-C power input

Software Environment

- **Operating System:** JetPack 6.2.1 (Linux Ubuntu 22.04 LTS + L4T)
- **Flashing Tool:** NVIDIA SDK Manager (v2.x) — used to flash JetPack onto the NVMe SSD.

Python Environment:

```
python3 -m venv yolov8-env

source yolov8-env/bin/activate

pip install ultralytics opencv-python onnxruntime
```

Model Deployment:

The trained YOLOv8 model (`best.pt`) was transferred from Google Colab to the Jetson via `scp`.
The model was then converted for optimized inference:

yolo export model=best.pt format=onnx

yolo export model=best.pt format=engine

- The **TensorRT (`.engine`) file** was used for real-time inference to achieve higher FPS.

RTSP Camera Integration

The Tapo C200 camera stream was accessed through its RTSP URL:

rtsp://<username>:<password>@<camera_ip>:554/stream1

- The live feed was decoded using OpenCV:

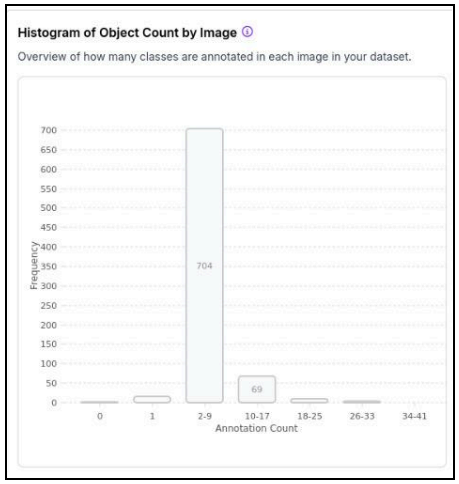
  import cv2

  cap = cv2.VideoCapture("rtsp://user:pass@192.168.0.xxx:554/stream1")

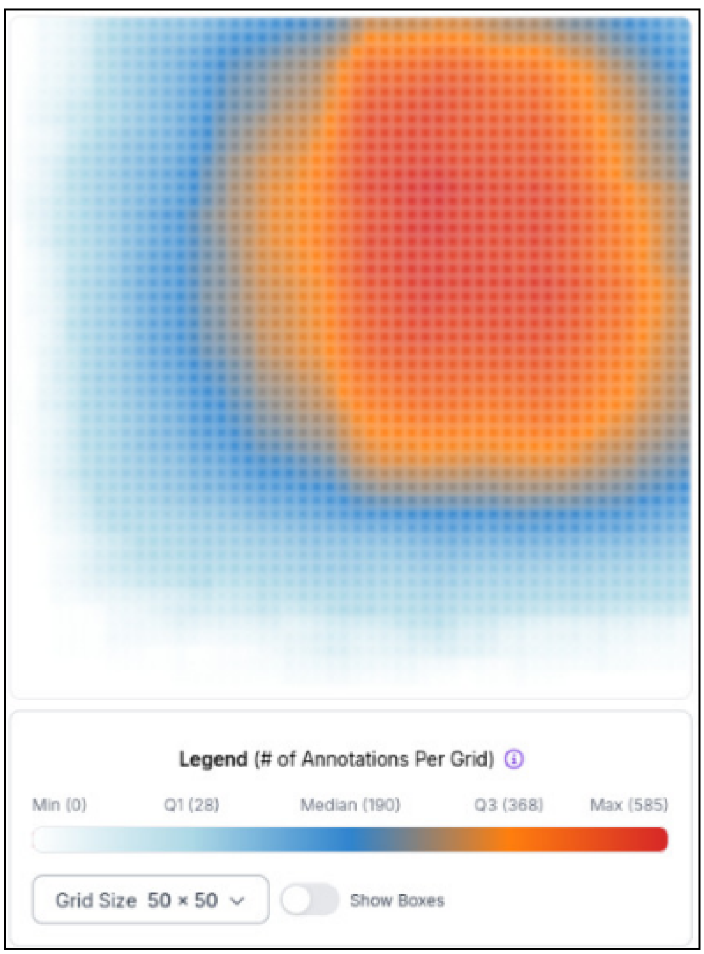
- Frames were passed to the YOLOv8 TensorRT model for detection and visualized in real time.

4. Results and Discussion

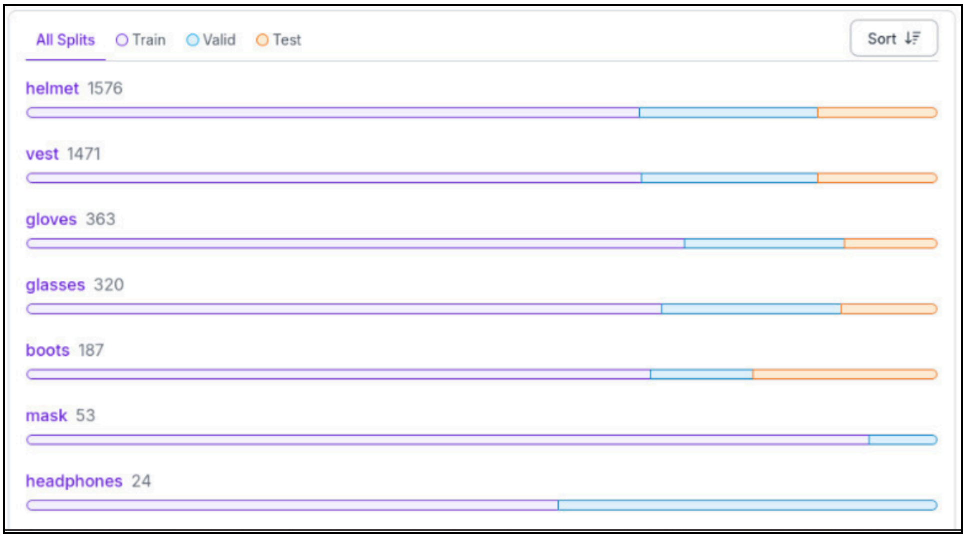
- Object count by image



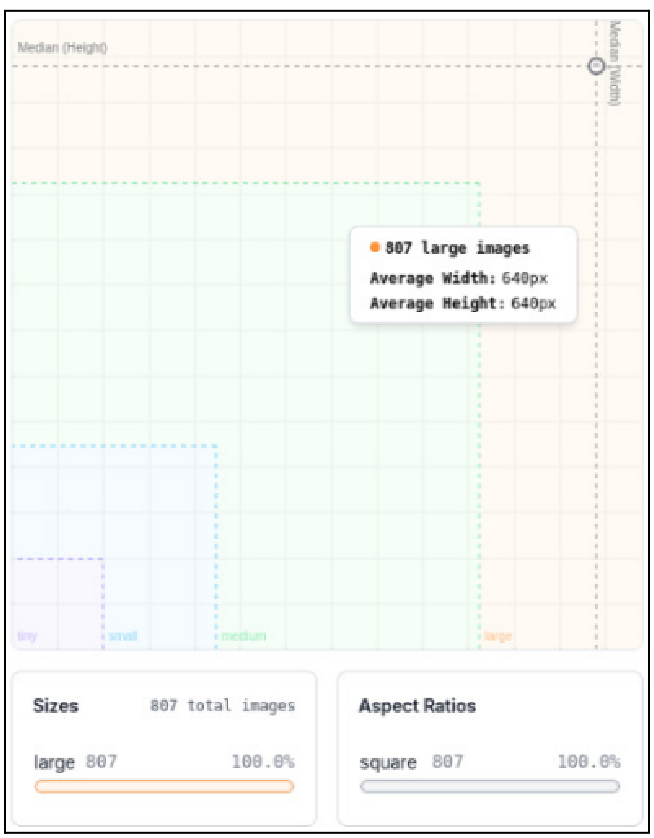
- Heat Annotation Map



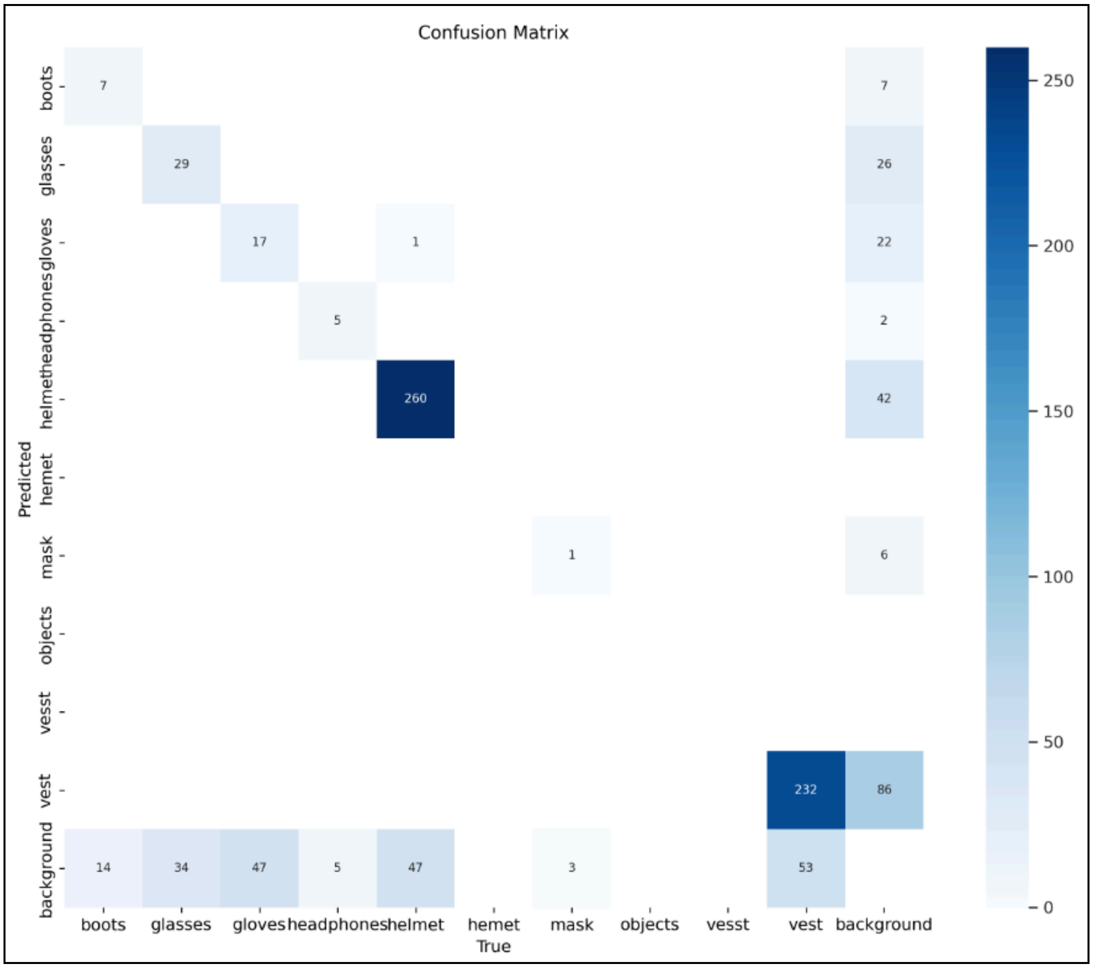
- Dataset



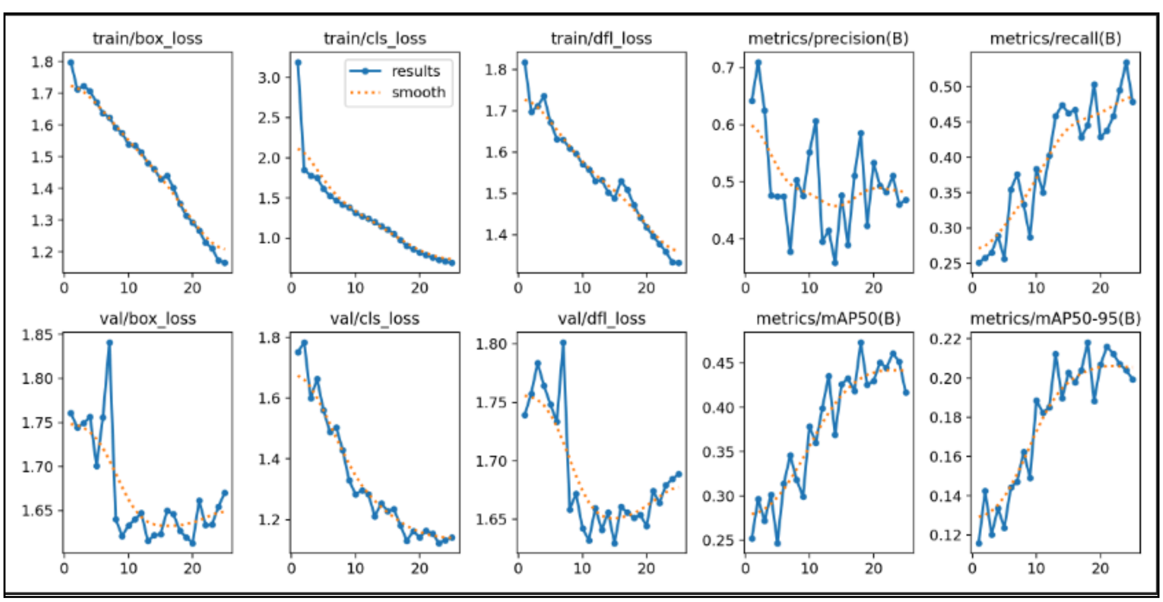
- Dimension Insights



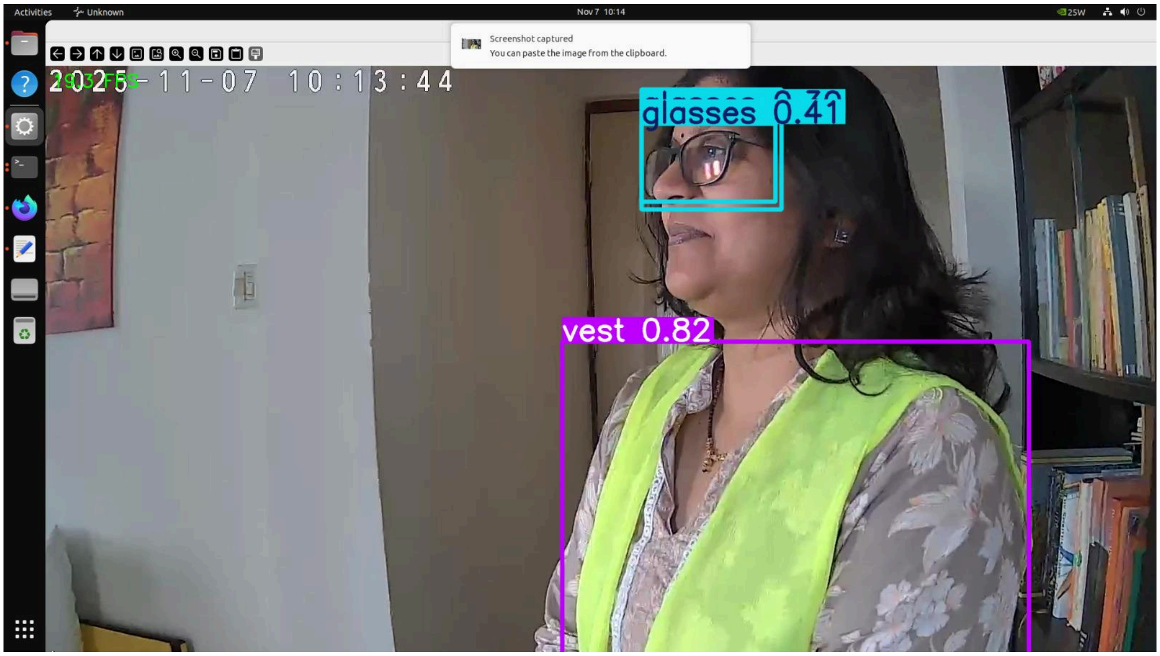
- Confusion Matrix

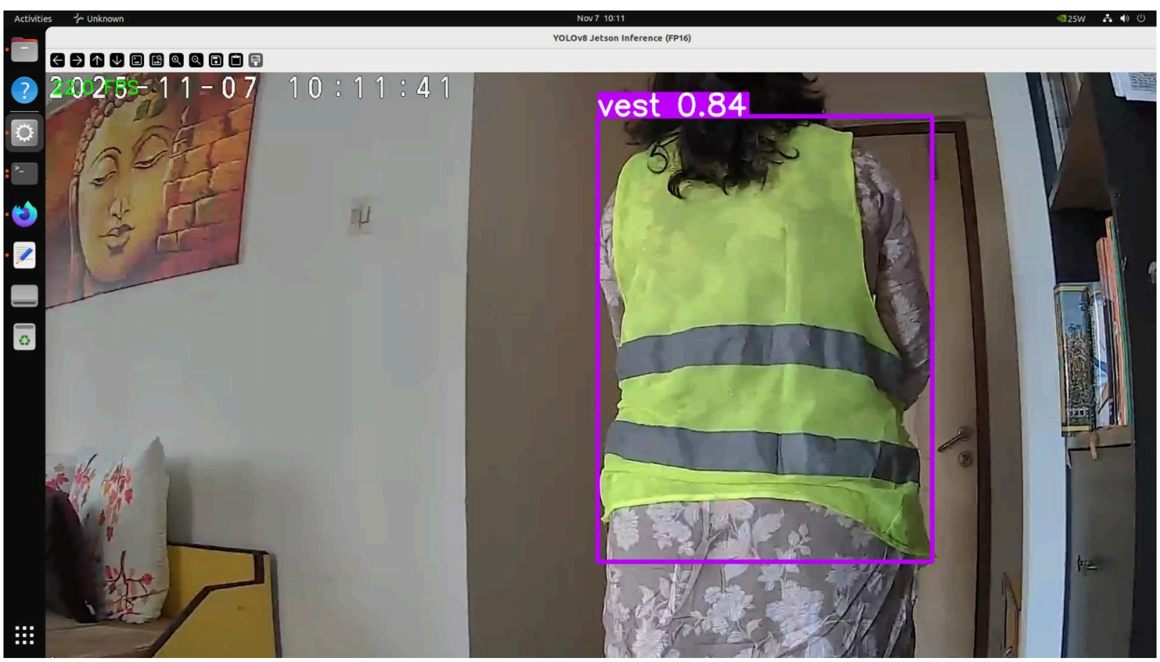


- Result Graphs



- Real-time Inference





5. Conclusion

The proposed system successfully demonstrates the feasibility of real-time PPE compliance detection using YOLOv8 on Jetson Orin Nano. With TensorRT optimization, the model achieves high-speed inference suitable for live RTSP streams. This approach can automate safety monitoring across industrial and construction sites, reducing human supervision requirements and improving workplace safety.

Future work includes expanding the dataset to include more PPE types, integrating IoT-based alert systems, and optimizing model performance under varying lighting and environmental conditions.

6. References

- Ultralytics YOLOv8 Documentation, https://docs.ultralytics.com/
- Roboflow PPE Detection Dataset, https://app.roboflow.com/
- Singh R. et al., “Deep Learning-Based Helmet Detection System for Industrial Safety,” IEEE Access, 2022.
- Sharma A. et al., “Real-Time PPE Detection Using Transfer Learning,” IJERT, 2023.
- NVIDIA Jetson Orin Nano Developer Kit Guide, https://developer.nvidia.com/embedded/jetson-orin-nano