YOLOv8 Object Detection Report

1.

This report documents the approach taken to train and evaluate two YOLOv8 object detection models:

Person Detection Model - Detects persons in an image.

PPE Detection Model - Detects PPE items (hard-hat, gloves, mask, glasses, boots, vest, ppe-suit, ear-protector, safety-harness) on cropped persons.

2.

Used YOLOv8 for training a model to detect persons in images.

Dataset: Images with annotated bounding boxes for persons in PascalVOC format, converted to YOLO format.

Training Pipeline:

Data preprocessing: Converted annotations and performed data augmentation.

Model Training: Used YOLOv8 with optimized hyperparameters.

Evaluation: Measured model performance using mAP and IoU.

Logic for Training:

Trained on full images, ensuring balanced training data.

Augmented dataset with horizontal flipping and brightness adjustments.

Optimized learning rate and batch size for best performance.

3 -

Used the trained person detection model to crop each detected person from the images.

Trained a separate YOLOv8 model on these cropped images to detect PPE items.

Logic for Training:

Extracted each person detected in the first model and saved them as individual images.

Converted PPE annotations to fit cropped images using bounding box transformations.

Trained the PPE detection model with the adjusted annotations.

Step 1: Person detection model runs on the full image.

Step 2: Persons are cropped based on detected bounding boxes.

Step 3: PPE detection model is applied to cropped person images.

Step 4: The detected PPE bounding boxes are transformed back to original full-image coordinates.

Step 5: OpenCV's cv2.rectangle() and cv2.putText() are used to draw bounding boxes and confidence scores on the full image.

4.

mAP (Mean Average Precision):

Person Detection: Achieved XX% mAP at IoU 0.5.

PPE Detection: Achieved XX% mAP at IoU 0.5.

IoU (Intersection over Union): Maintained an average IoU above XX%.

Confidence Thresholding: Optimized confidence threshold to reduce false positives.

5.

Successfully trained two YOLOv8 models for multi-stage detection.

Improved detection accuracy by carefully designing the pipeline for cropped images.

Used OpenCV for bounding box visualization instead of YOLO's built-in functions.

Optimized dataset balance and augmentation strategies to improve generalization.