ASSIGNMENT REPORT

1. Introduction

This project aims to develop two object detection models using YOLOv8:

- Detecting persons
- Detecting personal protective equipment (PPE)

Along with objective to fulfil:

i. Program to convert annotations from PascalVOC format to yolov8 format.
ii. Program to infer the results using results from both models.

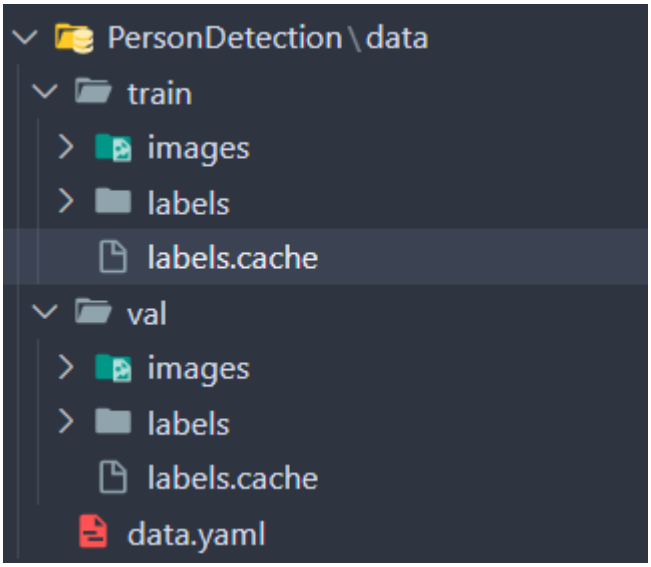
2. Dataset Preparation

The dataset consists of images and annotations for the following classes:

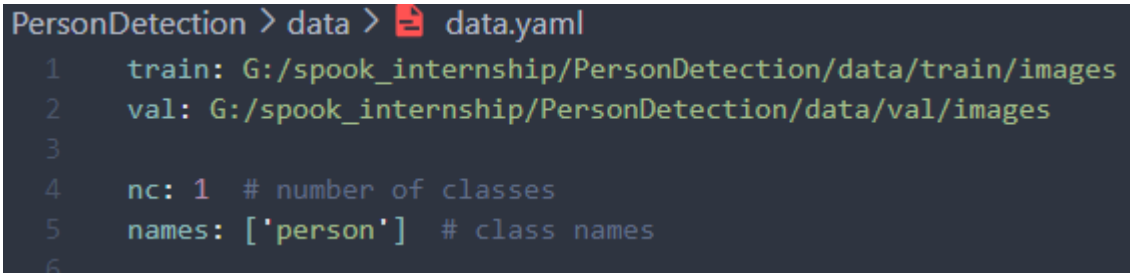
[Person, hard-hat, gloves, mask, glasses, boots, vest, ppe-suit, ear-protector, and safety-harness.]

The annotations were in PascalVOC format and were converted to YOLOv8 format using a custom script, using `***pascalVOC_to_yolo.py*** ` file.

- Dataset Preparation for PERSON DETECTION
  By modifying `*pascalVOC_to_yolo.py*` file to include only annotation mapped to person, we prepared modified annotation to each existing image. Used `make_dataset.py` to split the data into train and validation set (image 1). Also added `data.yaml ` file (image 2).



(*Image 1*)



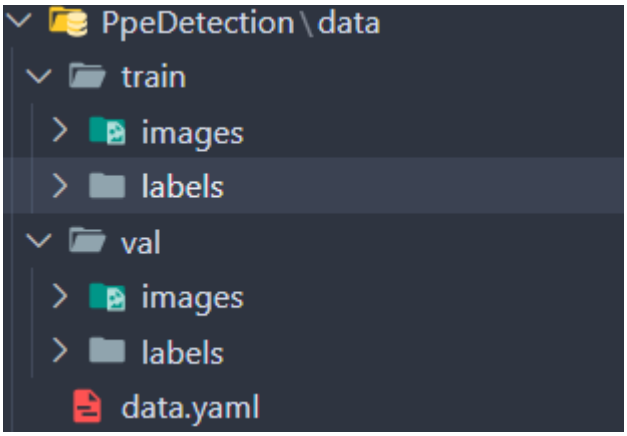
(Image 2)

No. of images in train: **332**

No. of images in validation split: **84**

- Dataset Preparation for PPE DETECTION

  Used the person annotation to crop the persons from the original image, preparing a 1214 image dataset. Again by modifying `*pascalVOC_to_yolo.py*` file to exclude 'person' and less occurring classes we prepared annotation for PPE detection. So final classes include ['hard-hat', 'gloves', 'boots', 'vest', 'ppe-suit'] . Used `make_dataset.py` to split the data into train and validation set (image 3). Also added `data.yaml` file (image 4).



(image 3)

```
PpeDetection > data > data.yaml
1   train: G:/spook_internship/PpeDetection/data/train/images
2   val: G:/spook_internship/PpeDetection/data/val/images
3
4   nc: 5   # number of classes
5   names: ['hard-hat', 'gloves', 'boots', 'vest', 'ppe-suit']  # class names
6
```

(Image 4)

No. of images in train: **968**

No. of images in validation split: **246**

3. Model Training

- Person Detection

  The first model was trained to detect persons in the entire image. The YOLOv8 model was used for this purpose. The training process involved the following steps:

  Data Preparation: refer to the dataset preparation for person detection section.

  Model Configuration: The base pretrained YOLOv8 was used to train the model with configuration batch_size = 16, patience=10, epochs = 30, imagsz=640.

- PPE Detection

  The second model was trained to detect PPE on cropped images of persons. The steps involved were **:**

  Cropping Images: Using the person detection model, images were cropped to include only the bounding boxes of detected persons.

  Data Preparation: refer to the dataset preparation for Ppe detection section.

  Model Configuration: The base pretrained YOLOv8 was used to train the model with configuration batch_size = 16, patience=10, epochs = 20, imagsz=640.

4. Inference Pipeline

An inference pipeline `inference.py` was developed to perform detection using both models. The pipeline involves the following steps:

- **Person Detection**: The weights from person detection model is used to detect persons in the input images.
- **PPE Detection**: For each detected person, the image is cropped, and the PPE detection model weight is used to detect PPE within the cropped image.
- **Bounding Box Drawing**: The detected PPE bounding boxes are adjusted on the original image and confidence scores are drawn on the original image using OpenCV.

Inference Script

A Python script named `***inference.py***` was created to handle the inference process. The script takes input and output directories, and paths to the person detection and PPE detection models as arguments.

5. Results and Evaluation

The models were evaluated based on standard object detection metrics such as precision, recall, and their inter related curves along with their confidence. The results showed that the models performed well on the validation set, with high precision and recall for both person and PPE detection.

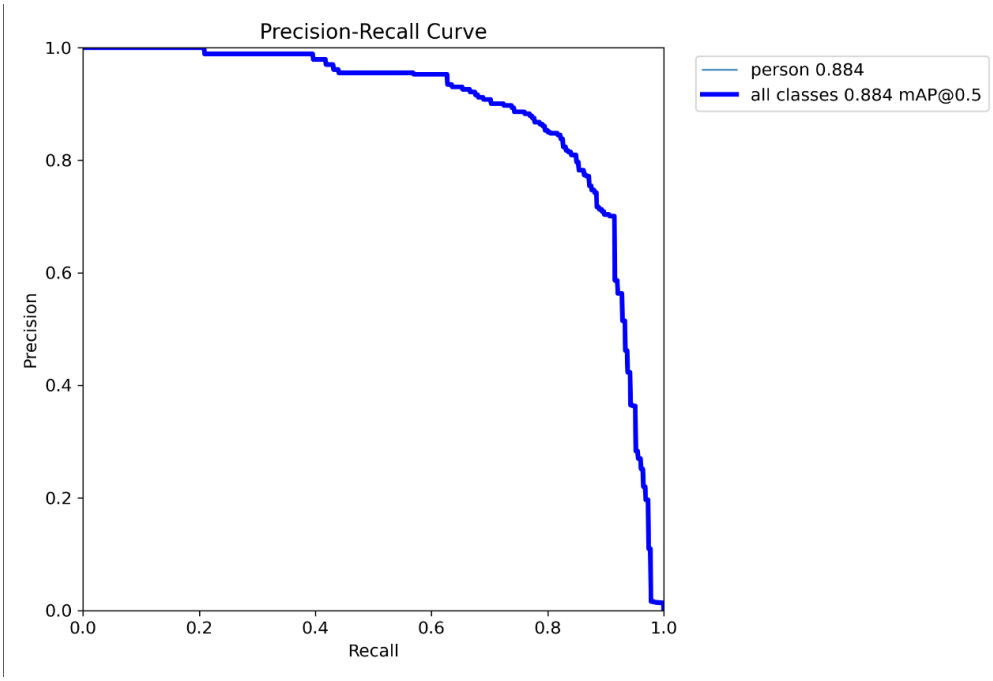
Confidence matrix: Confidence scores represent the probability that a given prediction is correct. They are crucial for model inference and interpretability.

Precision recall curve: The precision-recall (PR) curve is a valuable tool for evaluating the performance of a classification model, especially in scenarios where the classes are imbalanced. Here's why it's used:

1. Imbalanced Datasets: As we see in PPE label classes, vests and ppe-suits are less represented than other classes.
2. Trade-off Visualization: The PR curve helps visualize the trade-off between precision (the accuracy of positive predictions) and recall (the ability to find all positive instances).
3. Threshold Selection: The PR curve aids in selecting an optimal threshold for classification. Depending on the application, you might prioritize precision (minimizing false positives) or recall (minimizing false negatives).

Person Prediction :

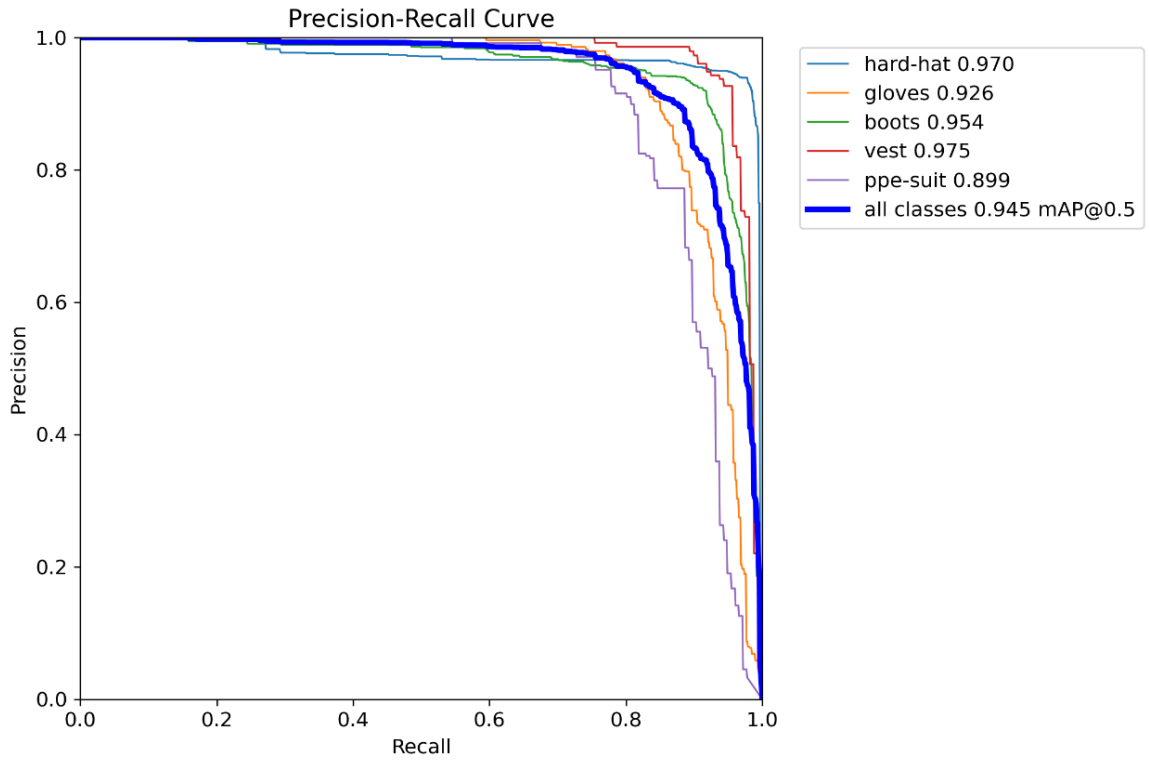
(image 5: preson prediction on val data)



(image 6: Precision-recall curve for person prediction model)

PPE Prediction :

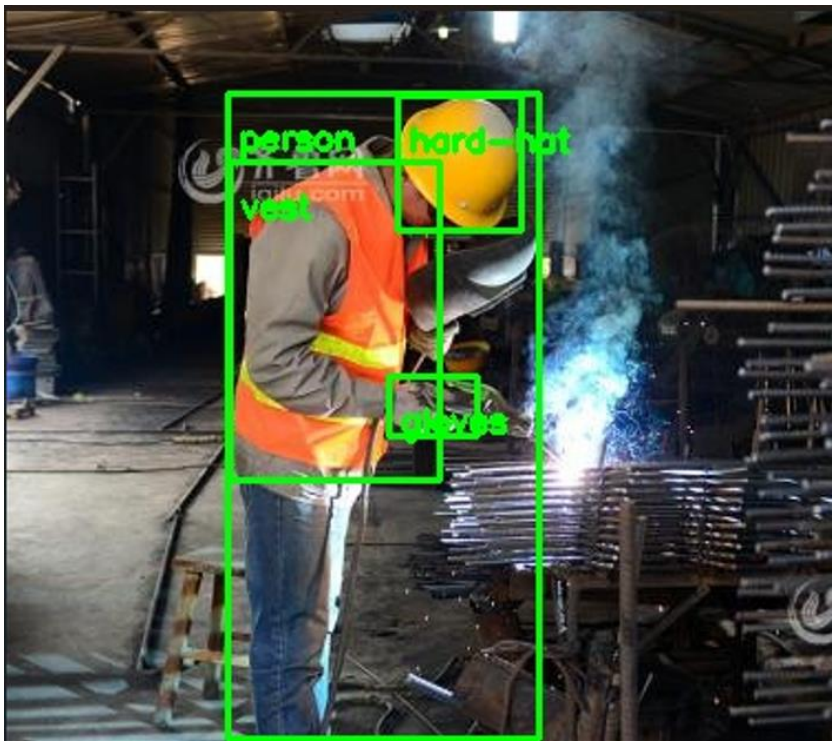
(image 7: PPE prediction on val data of cropped images



(image 8: Precision-recall curve for PPE prediction model)

6. Conclusion

This project successfully converted pascal VOC annotation to YOLO annotations. Developed two YOLOv8 models for person and PPE detection. Also able to build a pipeline to use both model weights and infer from it . The models were trained on a given dataset and evaluated using standard metrics. The inference pipeline was implemented to perform detection on new images and draw bounding boxes using OpenCV.



(Image 9: sample output from inference.py)

(Image 10: sample output from inference.py)