Assignment Report

1. Format Conversion (PascalVOC to YOLOv8)

1.1 Introduction

The dataset provided for this assignment was annotated in PascalVOC format, which is commonly used in computer vision tasks. However, YOLOv8 requires annotations in a specific format to train its models effectively. The first step was to convert the annotations from PascalVOC format to YOLOv8 format.

1.2 Conversion Process

- **Input**: The PascalVOC format uses XML files for each image, where bounding boxes are defined by the coordinates of the top-left and bottom-right corners (xmin, ymin, xmax, ymax). Each annotation also contains the class label of the object within the bounding box.
- **Output**: YOLOv8 format requires annotations in a text file with the same name as the image but with a `.txt` extension. Each line in this file corresponds to one object, containing the class ID followed by the normalized center coordinates (x_center, y_center) and the normalized width and height of the bounding box.

1.3 Conversion Script

The conversion was done using a Python script (`pascalVOC_to_yolo.py`) that:

1. **Parsed the PascalVOC XML files** to extract the bounding box coordinates and class labels.
2. **Converted the bounding box coordinates** from the original format to YOLO's required format.
3. **Normalized the coordinates** relative to the image dimensions.
4. **Saved the annotations** in a `.txt` file with each line containing the class ID and the normalized bounding box coordinates.

How to run the script?

```
python3 pascalVOC_to_yolo.py /path/to/dataset /path/of/output_dir/
```

Note that the 'dataset' directory should contain the directory called 'pascalVOC_labels' which contains .xml files. If not, please rename the directory to 'pascalVOC_labels'.

---

2. Training the Person Detection Model (YOLOv8)

2.1 Introduction The objective of this step was to train a YOLOv8 model to accurately detect persons in images. The YOLO (You Only Look Once) architecture is well-suited for real-time object detection tasks, and YOLOv8, the latest iteration, provides significant improvements in speed and accuracy.

2.2 Dataset The dataset provided in the assignment included images with annotations specifying the locations of persons in PascalVOC format, which was converted to YOLOv8 format as described earlier.

2.3 Model and Training Configuration

- **Model Used**: YOLOv8n (YOLOv8 Nano), which balances speed and accuracy, making it suitable for training on a laptop with a dedicated GPU.
- **Training Environment**: The training was performed on a laptop equipped with an NVIDIA RTX 4050 GPU, which provided sufficient computational power for this task.
- **Epochs**: The model was trained for 50 epochs, a reasonable number given the dataset size and the computational resources available.
- **Batch Size**: The default batch size was used.
- **Learning Rate**: The learning rate was set to the default value, which dynamically adjusted during training based on the model's performance.

2.4 Training Process

1. **Data Augmentation**: YOLOv8 applies various data augmentation techniques such as random scaling, flipping, and color adjustments to increase the diversity of the training data, helping the model generalize better.
2. **Loss Function**: The model optimizes a composite loss function that includes bounding box regression loss, objectness score loss, and classification loss.

3. **Optimization**: The default **SGD (Stochastic Gradient Descent)** optimizer was used, which is effective for large-scale machine learning problems. The momentum term in SGD helps in accelerating gradients vectors in the right direction, leading to faster converging.

2.5 Results The person detection model showed steady improvement in both training and validation metrics over the 50 epochs. The loss curves indicated that the model was learning effectively without overfitting.

---

3. Training the PPE Detection Model on Cropped Dataset

3.1 Introduction After training the person detection model, the next step was to train a second YOLOv8 model to detect Personal Protective Equipment (PPE) on cropped images of detected persons. The challenge here was to extract and adjust the PPE annotations from full images to fit the cropped images.

3.2 Dataset Preparation

- **Original Annotations**: The original annotations contained bounding boxes for PPE items within full images. These needed to be transformed to fit the cropped images.
- **Logic for Cropping and Annotation Adjustment**:
    1. **Person Detection**: The person detection model was used to detect and crop each person from the full image.
    2. **PPE Annotations**: For each cropped person image, the corresponding PPE annotations were adjusted by subtracting the top-left coordinates of the person's bounding box from the PPE bounding box coordinates.
    3. **Saving Annotations**: The adjusted annotations were then saved in YOLOv8 format, creating a new "cropped dataset."

3.3 Model and Training Configuration

- **Model Used**: YOLOv8n, the same variant as used for person detection.
- **Training Environment**: The model was trained on the same laptop with RTX 4050 GPU.
- **Epochs**: The training was conducted for 50 epochs, consistent with the person detection model.
- **Training Strategy**: The same default settings for batch size, learning rate, and optimizer were used.

3.4 Results The PPE detection model was able to effectively detect various PPE items such as hard-hats, gloves, and boots within the cropped images. The loss curves and evaluation metrics demonstrated a successful learning process similar to the person detection model.

4. Inference Script

4.1 Overview The inference script (`inference.py`) was designed to take an input directory of images, perform inference through both the person detection model and the PPE detection model, and save the annotated results in another directory.

4.2 Inference Logic

1. **Input and Output Handling**: The script accepts four command-line arguments—input directory, output directory, person detection model, and PPE detection model.
2. **Person Detection**:
    - For each image, the person detection model is run to detect all persons.
    - Each detected person is cropped from the image.
3. **PPE Detection**:
    - The cropped images are fed into the PPE detection model to detect PPE items.
    - The PPE bounding boxes are then mapped back to the original full image coordinates.
4. **Drawing and Saving Results**:
    - Bounding boxes and labels for both persons and PPE items are drawn on the original image using OpenCV.
    - The annotated images are saved in the specified output directory.

How to run ?

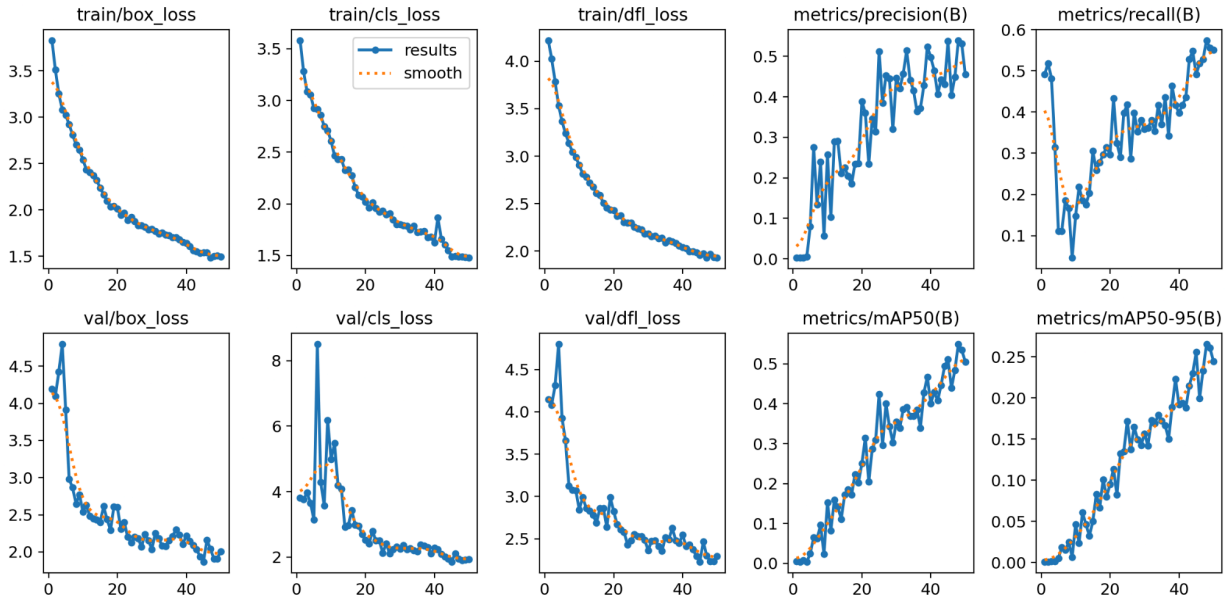
```
python3 inference.py --input_dir /path/to/input/images --output_dir
/path/to/save/output --person_det_model /path/to/person/detection/model
--ppe_detection_model /path/to/ppe/detection/model
```

---

5. Learning and Evaluation Metrics

5.1 Training Metrics

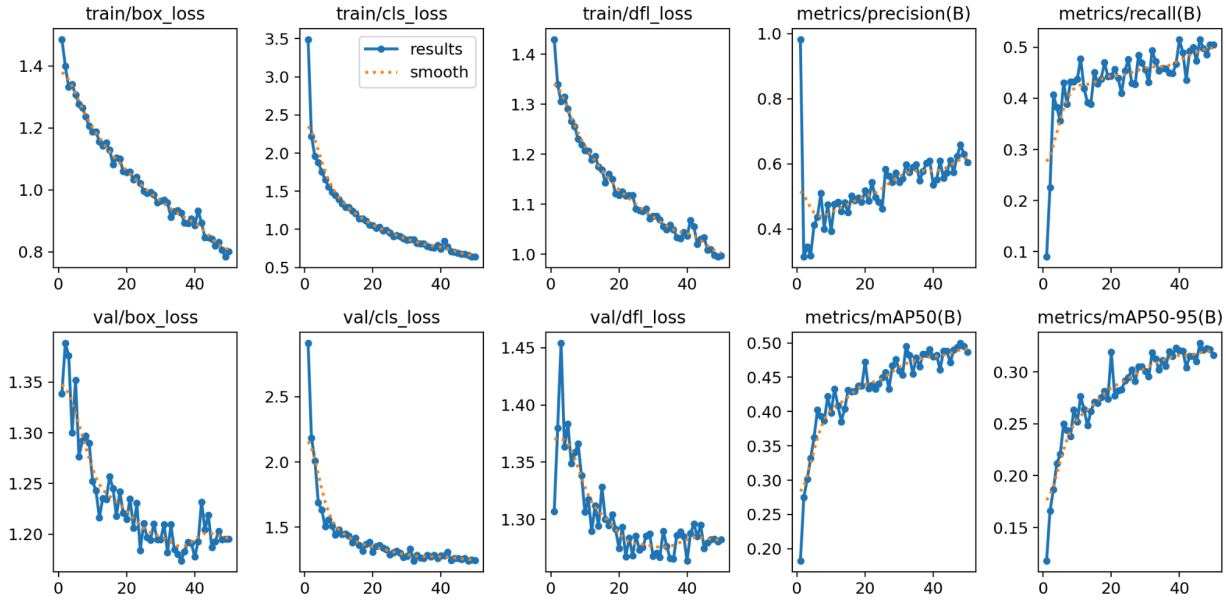
Person Detection Model:



- **Training Losses**:
  - **Box Loss**: The box loss steadily decreased from approximately 3.5 to 1.5 over the 50 epochs, indicating the model's improving ability to localize persons accurately within the images.
  - **Class Loss**: The class loss decreased from about 3.5 to around 1.5, showing that the model's accuracy in classifying the detected objects as "persons" improved with training.
  - **DFL (Distribution Focal Loss)**: This loss metric dropped from 4.0 to around 2.0, further confirming the model's increasing confidence in its predictions.
- **Validation Losses**:
  - **Box Loss**: The validation box loss started higher and fluctuated more than the training loss but showed a general downward trend from 4.5 to approximately 2.0.

    - **Class Loss**: The class loss on the validation set began at around 8.0 and reduced significantly to about 2.0, though it showed some volatility, which may indicate areas for further tuning or more data.
    - **DFL Loss**: Similarly, the validation DFL loss decreased from 4.5 to about 2.5, mirroring the trend seen in training.
- **Evaluation Metrics**:
    - **Precision**: The precision metric started low but improved steadily, reaching around 0.4 by the end of training, indicating a moderate level of accuracy in detecting persons without too many false positives.
    - **Recall**: Recall also improved significantly, starting at around 0.1 and ending near 0.55, showing that the model became more effective at detecting most persons present in the images.
    - **mAP50**: The mean Average Precision at an IoU threshold of 0.5 increased from around 0.1 to 0.55, demonstrating the model's improving ability to correctly predict the bounding boxes at a less strict IoU threshold.
    - **mAP50-95**: This metric, which averages the mAP across IoU thresholds from 0.5 to 0.95, also increased from about 0.0 to 0.25, indicating the model's growing accuracy across various IoU thresholds.

PPE Detection Model:



- **Training Losses**:

    - **Box Loss**: The box loss for the PPE detection model decreased from approximately 1.4 to 0.8, showing the model's improved ability to localize PPE items within the cropped images.
    - **Class Loss**: The class loss decreased from about 3.5 to around 0.5, indicating strong performance in correctly identifying different types of PPE.
    - **DFL Loss**: This loss metric dropped from 1.4 to just above 1.0, confirming the model's growing confidence in its predictions.
- **Validation Losses**:
    - **Box Loss**: The validation box loss showed a downward trend, reducing from 1.35 to approximately 1.2, although with some fluctuation.
    - **Class Loss**: The validation class loss decreased from 2.5 to about 1.5, with notable fluctuation early in the training process but stabilizing towards the end.
    - **DFL Loss**: The validation DFL loss decreased consistently from 1.45 to about 1.3, mirroring the training loss trend.
- **Evaluation Metrics**:
    - **Precision**: Precision improved steadily, reaching around 0.6 by the end of the training process, indicating that the model was making accurate predictions with fewer false positives.
    - **Recall**: Recall started low but improved significantly, reaching approximately 0.5 by the end, showing that the model became more effective at detecting most PPE items present in the images.
    - **mAP50**: The mean Average Precision at an IoU threshold of 0.5 increased from around 0.2 to 0.5, indicating improved accuracy in detecting PPE items with good localization.
    - **mAP50-95**: This metric also showed a steady increase from about 0.1 to 0.3, reflecting the model's improving accuracy across a range of IoU thresholds.

5.2 Summary of Results

- Both models showed consistent improvement in precision, recall, and mAP metrics across the 50 epochs, indicating effective learning and model convergence.
- The person detection model performed well in detecting persons, with mAP50 reaching 0.55, while the PPE detection model also showed strong performance in detecting various PPE items with mAP50 reaching 0.5.
- The fluctuations observed in the validation losses, particularly in the early stages of training, suggest that further tuning or additional data could help stabilize and possibly improve the models' performance.