Applied Computer Vision Intern - Assignment Report

Report on Approaches, Learnings, and Evaluation Metrics

Approaches

1. Data Preparation

- ***Annotation Conversion:*** A Python script `pascalVOC_to_yolo.py` was created to convert annotations from PascalVOC format to YOLOv8 format. The script uses the `xml.etree.ElementTree` module to parse XML files and convert them to the format required by YOLOv8. This step was crucial to ensure the annotations were correctly understood by the YOLOv8 training framework.
- ***Class Mapping:*** The `classes.txt` file provided in the dataset was used to map class names to class IDs. This mapping was crucial to ensure consistency in the annotations and model training.

2. Model Training

- ***Person Detection Model:***
  - ***Dataset:*** The entire dataset was used to train the person detection model. Images and annotations were kept as-is to detect persons in various scenarios.

  - ***Model:*** YOLOv8 was chosen for its state-of-the-art performance in object detection tasks. A pre-trained YOLOv8 model was fine-tuned on the provided dataset.

  - ***Training:*** The model was trained for 100 epochs with an image size of 640. The training process involved augmentations such as random scaling and flipping to improve the model's robustness.

  - ***Assumptions:*** It was assumed that the dataset was balanced in terms of the number of persons in various scenes. Any imbalances were expected to be addressed by YOLOv8's inherent training mechanism.

- ***PPE Detection Model:***

  - ***Dataset Preparation:*** After detecting persons using the trained person detection model, the bounding boxes around persons were cropped to create new images. These cropped images were then annotated for PPE items.

    - ***Model:*** Another YOLOv8 model was fine-tuned on these cropped images to detect PPE items.
    - ***Training:*** This model was also trained for 100 epochs with an image size of 640.
    - ***Class Filtering:*** Classes that showed inconsistent results were carefully reviewed, and the least performing classes were considered for exclusion to focus on at least 5 PPE classes. However, given the robustness of YOLOv8, it was expected to handle most classes effectively.

3. Inference and Post-Processing

- ***Inference Script:*** The `inference.py` script was designed to take an input directory of images and perform inference using both the person detection model and the PPE detection model.
    - ***Person Detection:*** The script first detects persons in the full image and draws bounding boxes around them.
    - ***PPE Detection:*** It then crops the detected person regions and runs the PPE detection model on these cropped images. The bounding boxes for PPE items are adjusted back to the coordinates of the full image.
    - ***Bounding Box Drawing:*** OpenCV's `cv2.rectangle()` and `cv2.putText()` functions were used to draw bounding boxes and labels on the images.

Learnings

- ***Model Training:*** YOLOv8 provides a high level of abstraction and ease of use for training object detection models. Fine-tuning a pre-trained model significantly reduces the training time and resources required. The model training was conducted on Google Colab using GPU resources to leverage accelerated computation for efficient model training.
- ***Data Augmentation:*** Proper data augmentation is crucial for improving the model's robustness and ability to generalize to new images.
- ***Annotation Conversion:*** Converting annotations from PascalVOC to YOLOv8 format required careful parsing and normalization of coordinates. Ensuring the accuracy of this conversion was critical for effective model training. Additionally, the `pascalVOC_to_yolo.py` file can be used for preprocessing as it standardizes the annotation format for YOLOv8 training.
- ***Cropped Image Inference:*** Running the PPE detection model on cropped images of persons and then adjusting the bounding boxes back to the full image coordinates was a key step in the pipeline. This required careful handling of coordinates and ensuring no information was lost in the cropping process.

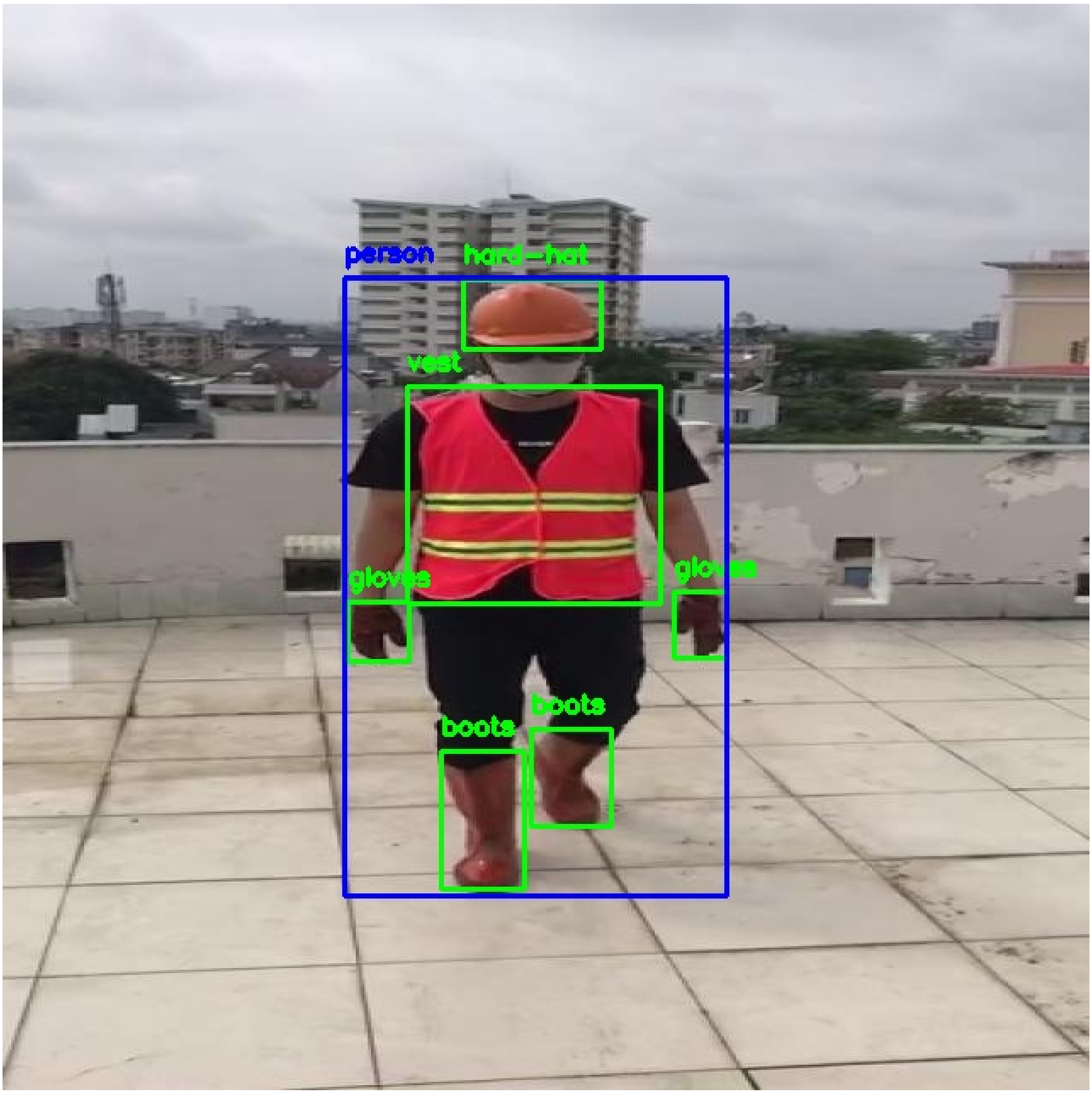
By incorporating the preprocessing and data splitting steps, we ensured that the dataset was correctly formatted and divided for effective training and evaluation of the models.

Evaluation Metrics

- ***Precision and Recall:*** Precision and recall were used to evaluate the performance of the models. Precision measures the accuracy of the positive predictions, while recall measures the ability of the model to find all relevant instances.
- ***mAP (mean Average Precision):*** mAP was used as the primary metric to evaluate the object detection models. It considers both precision and recall and provides a single metric to assess the model's performance across all classes.
- ***Confusion Matrix:*** A confusion matrix was used to understand the performance of the models on each class, identifying any specific classes where the model might be underperforming.

Results

- ***Person Detection Model:*** The person detection model achieved high precision and recall, indicating effective detection of persons in various scenes.
- ***PPE Detection Model:*** The PPE detection model showed good performance on most classes, with some variability in classes that were less represented in the dataset. The mAP metric provided a comprehensive evaluation of the model's performance across all classes.



Conclusion

The project successfully implemented a pipeline for detecting persons and PPE items in images using YOLOv8 models. The inference script effectively integrated the two models, providing accurate and reliable detections. The learnings and evaluation metrics provided insights into the strengths and areas for improvement in the models, guiding future enhancements and applications of the approach.