Object Detection Using YOLOv8

Overview

I implemented an object detection system using YOLOv8 to identify people and various Personal Protective Equipment (PPE) items. The system uses custom-trained YOLOv8 models and includes:

- Converting PascalVOC format XML annotations to YOLO format
- Training a YOLOv8 model on a custom dataset for person detection
- Training a second YOLOv8 model on a custom dataset for PPE detection
- Creating an inference pipeline
- Generating predictions with visualized results showing bounding boxes

Workflow

➢ Converting PascalVOC Annotations to YOLO Format

The dataset annotations were initially in PascalVOC XML format and needed conversion to YOLO format for training. This conversion ensures each line in the .txt file represents an object with normalized bounding box coordinates:

- **XML Annotation Parsing:** Extracted class names and bounding box coordinates (xmin, ymin, xmax, ymax)
- **Bounding Box Conversion:** Transformed coordinates into YOLO format
- **Image Dimension Handling:** Retrieved image dimensions using OpenCV to normalize coordinates
- **YOLO Annotation Generation:** Created a .txt file for each XML annotation file with class index and normalized coordinates
- **Class Label Management:** Maintained a class list with numerical IDs

The conversion code was saved in the **PascalVOC_To_Yolo.py** file.

➢ Person Detection Model

After preparing the dataset:

- The YOLOv8 model was trained on the custom dataset containing images and YOLO-format labels
- The dataset was split into training, testing, and validation sets
- The model was trained for 140 epochs, with automated handling of image loading, annotations, and weight saving
- The resulting model was specialized for person detection only
- The best weights were saved as **best_Person.pt**

➢ PPE Detection Model

The second model was trained to detect PPE items in cropped person images:

- The model was designed to detect: hard-hats, gloves, masks, glasses, boots, vests, PPE-suits, ear-protectors, and safety-harnesses
- The dataset included images and corresponding YOLO-format labels
- The data was split into training, testing, and validation sets
- Training ran for 140 epochs with automated image loading, annotation handling, and weight saving
- The best weights were saved as **best_PPE.pt**

➢ Inference Pipeline

I created a comprehensive pipeline for detection:

- The system takes an image directory as input and saves results to another directory
- First, the person detection model processes full images to identify people
- Then, detected person regions are cropped and passed through the PPE detection model
- PPE detections from cropped images are converted back to full-image coordinates
- Results are visualized with bounding boxes and confidence scores using OpenCV functions

- Processed images are saved in the output directory in two folders: annotated and cropped

The inference code was saved as **inference.py.**

Results

The project successfully demonstrates YOLOv8's effectiveness in detecting people and PPE items, with high-confidence detections of various safety equipment in real-world images.