

MDM Lab

Assignment 3

Name: Kaustubh Wagh

PRN: 202201070021

Batch: T1

Documentation File

1. Overview

This project demonstrates how to use the **Ultralytics platform** for object detection using a **YOLOv11 model**. The goal was to train a model to detect **people** in images using a custom dataset. The process involves preparing the dataset, training the YOLOv11 model, running inference on test images, and evaluating the model's performance.

2. Setup

Required Libraries

To get started with the Ultralytics YOLOv11 model, we need to install the following libraries:

- **Ultralytics:** for YOLOv11 model training and inference.
- **Roboflow:** for accessing and managing datasets.

Run the following commands to install the required libraries:

```
!pip install ultralytics
```

```
!pip install roboflow
```

Ultralytics Installation

Install **Ultralytics** and check the environment setup:

```
import ultralytics
```

```
ultralytics.checks()
```

3. Dataset Preparation

The dataset used for this project was obtained from **Roboflow**. We used the **People Detection** dataset, which was exported in YOLO format. The dataset was then downloaded and unzipped for training.

1. **Download Dataset:** The dataset was accessed via the Roboflow API, which allows fetching datasets in YOLOv8 format:

```
from roboflow import Roboflow  
  
rf = Roboflow(api_key="your_api_key")  
  
project = rf.workspace("leo-ueno").project("people-detection-o4rdr")  
  
version = project.version(9)  
  
dataset = version.download("yolov8")
```

2. **Extract Dataset:** The dataset was unzipped and its structure verified, with a focus on the **data.yaml** file, which defines the dataset configuration.

4. Model Training

To train the YOLOv11 model, we used the following command:

```
!yolo task=detect mode=train data={dataset.location}/data.yaml  
model="yolov11n.pt" epochs=10 imgsz=640
```

This command initiates the training process for **10 epochs** with a batch size of 16 and an image size of 640x640. The model's weights were saved after training.

Training Command Breakdown:

- task=detect: Object detection task.
- mode=train: Specifies training mode.
- data={dataset.location}/data.yaml: Dataset configuration file.
- model="yolov11n.pt": Pre-trained YOLOv11 model.
- epochs=10: Number of training epochs.
- imgsz=640: Image size for training.

5. Inference and Evaluation

After training, the YOLOv11 model was used to make predictions on test images. The inference was run with the following command:

```
!yolo task=detect mode=predict  
model="/content/runs/detect/train/weights/best.pt" conf=0.25  
source=image2.jpg save=True
```

This command runs object detection on the image **image2.jpg** with a confidence threshold of 0.25, and the results are saved.

Evaluation Metrics:

The model's performance was evaluated using the following metrics:

- **Mean Average Precision (mAP@50, mAP@50-95):** Measures the model's accuracy across different Intersection over Union (IoU) thresholds.
- **Precision:** Proportion of true positive predictions out of all positive predictions.
- **Recall:** Proportion of true positive predictions out of all actual positives.
- **F1 Score:** The harmonic mean of precision and recall, balancing both metrics.

The model performed well in terms of accuracy and object detection precision.

6. Conclusion

The YOLOv11 model successfully detected people in images with high accuracy. The **Ultralytics platform** provided a seamless environment for model training, evaluation, and inference. The project showcased the effectiveness of YOLOv11 for real-time object detection, providing accurate results even on a custom dataset.