Bird View Car Detection Using YOLO

Setup

Verify that Python is installed and the version is between 3.8 (inclusive) and 3.11 (exclusive). Extract the source code zip file to a directory (`source directory`). Create another directory (`videos directory`) and copy the video files and the accompanying annotation .xml files into it. Ensure that each annotation file has the same name as its corresponding video file.

On Windows:

```
python --version
```

On Ubuntu:

```
python3 --version
```

This will display the version of Python installed on your system. If the version is less than 3.8 or equal to or greater than 3.11, you will need to install a compatible version.

Navigate to the `source directory` by changing the current directory:

```
cd Smart-Plane-master
```

Create and activate a virtual environment:

On Windows:

```
python -m venv .venv
.venv\Scripts\Activate.ps1
```

On Ubuntu:

```
python3 -m venv .venv
source .venv/bin/activate
```

Note: `.venv` is the name of the virtual environment, you can change it to any other name of your choice. Now upgrade `pip` and install the required packages:

```
python -m pip install --upgrade pip
pip install -r requirements.txt
```

To prepare data:

```
python prepare_data.py --videos_dir path/to/videos/directory --data_dir data
```

This will create a `data` directory and store all extracted frames and label data within it. Now, it's time to train the network using the prepared data:

```
python train.py --epochs EPOCHS --yolo_model YOLO_MODEL
```

Replace `EPOCHS` with the number of training epochs and `YOLO_MODEL` with one of the following YOLO models:

`yolov5n, yolov5s, yolov5m, yolov5l, yolov5x`

`yolov5n` has the lowest number of parameters and the fastest speed. `yolov5x` has the maximum number of parameters and the lowest speed.

Training is a long process and requires a huge amount of system resources. To view the progress open another terminal, navigate to the `source directory` and run:

```
tensorboard --logdir runs
```

Navigate to the prompted url (e.g. `http://localhost:6006/`) in your browser to view the training curves. After the training is finished, you can validate the trained model with test set (test set is created automatically in the data preparation phase and is not used for training.):

```
python val.py --yolo_model YOLO_MODEL
```

For `YOLO_MODEL`, provide the path to a trained network's weight file (`.pt`) located in the `runs/train` directory. The validation results are save in `runs/val` directory.

To detect cars in videos and images you can simply pass a file (video/image) or a directory path containing videos/images to `detect.py` script:

```
python detect.py --yolo_model YOLO_MODEL --source FILE/FOLDER --conf_thres CONF_THRES
```

As before `YOLO_MODEL` is the path to a trained network's weights file, FILE/FOLDER is a file path to an image or video or a folder path containing images and videos. The detection results are saved in `runs/detect` directory.

Results

Train Box Loss per Epoch (YOLOv5s & YOLOv5n)

Train Object Loss per Epoch (YOLOv5s & YOLOv5n)

Validation Box Loss per Epoch (YOLOv5s & YOLOv5n)

Validation Object Loss per Epoch (YOLOv5s & YOLOv5n)

mAP, Precision and Recall per Epoch Curves