how_to_train_yolo_model

November 7, 2024

1 Environment

YOLO can be run oon MacOS and WSL directly, there is no compatibility problems. In order to run on Windows, we need to install some specific libraries.

Before that, make sure cuda is installed on computer if using Windows or WSL. Cuda can be installed by the link: https://developer.nvidia.com/cuda-downloads

Run the following code to check nvidia version.

```
[ ]: !nvidia-smi
```

Before training, we need to create virtual env to install all relevant dependencies.

```
[ ]: !python -m venv venv
```

There are various way to activate venv based on different systems. On Windows:

```
[ ]: !./venv/Script/Activate
```

On Linux or MacOS:

```
[ ]: !source venv/bin/activate
```

1.1 Windows

On Windows, some versions of torch are incompatible with cuda, resulting in the inability to use gpu during the training process. Here is the torch for cuda121.

```
[ ]: !pip install ultralytics
     !pip uninstall torch torchvision torchaudio -y
     !pip cache purge
     !pip install torch torchvision torchaudio --extra-index-url https://download.
      ↪pytorch.org/whl/cu121
```

Or install the version with CUDA support as per your choice from here

1.2 WSL and MacOS

Easy to train the model on WSL and MacOS since there is no incompatible problem. YOLO can be installed by the following code.

[ ]:
```
!pip install ultralytics
```

2 Training

YOLO is easy to train as YOLO using pre-trained model. The architecture of YOLO is set up in advance but still could be changed.

[ ]:
```
from ultralytics import YOLO
```

Steps of training YOLOV8 model are: 1. dataset with YOLOV8 format should be downloaded on Roboflow, with a *.yaml* file on the root folder 2. choose yolov8m-seg.pt as pretrained model for training. 3. set up hyperparameters like image size, batch size, epochs, etc.

Note that early stop is applied during training as default.

[ ]:
```
model = YOLO('yolov8m-seg.pt')
model.train(data='path/to/data/yaml', imgsz=1280, batch=-1, epochs=300)
```

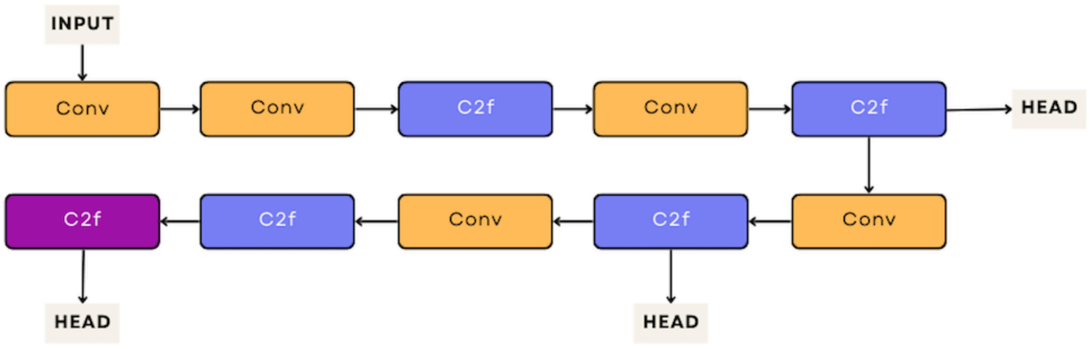
The training results will be stored in runs folder including training loss, confusion matrix, mAP, etc.

3 Architecture of YOLOV8

If you download ultralytics through git, there will be a folder named *ultralytics* in your repo. The architecture of YOLOV8 is demonstrated in a file named yolov8-seg.yaml, path *'ultralytics/cfg/models/v8'*. There are two main parts, one is backbone, the other is head. The differences of these two parts are as follows:

1. backbone:
    - Mainly responsible for extracting features from raw input images.
    - Typically consists of a pretrained deep neural network such as VGG, ResNet, Darknet, etc.
    - The output of feature extraction is passed on to other parts of the model, such as the head or other task-specific branches.
2. head:
    - Usually follows the backbone and is responsible for performing specific tasks like classification, detection, or segmentation.
    - May include classifiers, regressors, or segmenters, depending on the model's intended task.
    - The head receives feature representations from the backbone and outputs final predictions or feature representations.

In YOLOv8, the "backbone" refers to the main architecture of the model responsible for extracting features from input images. YOLOv8 typically uses a deep neural network as its backbone, often a pretrained convolutional neural network such as Darknet or other commonly used networks like ResNet. This backbone network extracts features from the raw images to facilitate subsequent object detection tasks. The choice and design of the backbone network significantly influence the model's performance and speed.



Here is an example of backbone:

[ ]:
```
# - [from, repeats, module, args]
# - [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
# - [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
# - [-1, 3, C2f, [128, True]]
# - [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
# - [-1, 6, C2f, [256, True]]
# - [-1, 1, Conv, [512, 3, 2]] # 5-P4/16
# - [-1, 6, C2f, [512, True]]
# - [-1, 1, Conv, [1024, 3, 2]] # 7-P5/32
# - [-1, 3, C2f, [1024, True]]
# - [-1, 1, SPPF, [1024, 5]] # 9
```

We can either change number or kind of layers or change repeats, as per your choice.