Step – 1: Creating an environment

Please use the following commands to set up the environment.

conda create -n Name of the environment python=3.9

I personally, used anaconda environment and used above command in the conda cmd.

Step:2 : Installation required libraries and software's:

Please navigate to the original folder.
Example: datascience_project

C:\Users\makula\Desktop\Fall 2022\Data Science\Project\datascience_project>

Use the following commands to install

pip install -r requirements_prereq.txt

pip install -r requirements_gpu.txt

Step-3: Model modification or tuning hyperparameters:

Training:

Cuda support: (Please include –device 0)

```
#Model-1
# Cuda Supported code - requires GPU support

#os.system("python wf_ml_training.py --workers 1 --device 0 --batch-size 10 --epochs 100
--img 640 640 --data data_processing/train_test/custom_data.yaml --hyp
data_processing/train_test/hyp.scratch.custom.yaml --cfg
cfg/training/yolov7-custom.yaml --name Data_Munging_visualisation --weights
data_processing/cfg/yolov7_pretarined_weights.pt")

#os.system("python wf_ml_training.py --workers 1 --device 0 --batch-size 10 --epochs 10
--img 640 640 --data data_processing/train_test/custom_data.yaml --hyp
data_processing/train_test/hyp.scratch.custom.yaml --cfg
cfg/training/yolov7-custom.yaml --name Data_Munging_visualisation --weights
data_processing/cfg/yolov7_pretarined_weights.pt")


# Uncomment if the machine doesn't support the CUDA or GPU
os.system("python wf_ml_training.py --workers 1 --batch-size 10 --epochs 150 --img 640
640 --data data_processing/train_test/custom_data.yaml --hyp
data_processing/train_test/hyp.scratch.custom.yaml --cfg
cfg/training/yolov7-custom.yaml --name Data_Munging_visualisation --weights
data_processing/cfg/yolov7_pretarined_weights.pt")


#Model - 2 Reducing the epoch size and evaluating the accuracies
```

```
#os.system("python wf_ml_training.py --workers 1 --batch-size 10 --epochs 50 --img 640
640 --data data_processing/train_test/custom_data.yaml --hyp
data_processing/train_test/hyp.scratch.custom.yaml --cfg
cfg/training/yolov7-custom.yaml --name Data_Munging_visualisation --weights
data_processing/cfg/yolov7_pretarined_weights.pt")

#Model - 3 Reducing the batch size and epochs for evaluating the accuracies

#os.system("python wf_ml_training.py --workers 1 --batch-size 4 --epochs 10 --img 640 640
--data data_processing/train_test/custom_data.yaml --hyp
data_processing/train_test/hyp.scratch.custom.yaml --cfg
cfg/training/yolov7-custom.yaml --name Data_Munging_visualisation --weights
data_processing/cfg/yolov7_pretarined_weights.pt")

#Model - 4 Using YoloV7_tiny pretrained model and evaluating the accuracies

#os.system("python wf_ml_training.py --workers 1 --batch-size 10 --epochs 100 --img 640
640 --data data_processing/train_test/custom_data.yaml --hyp
data_processing/train_test/hyp.scratch.custom.yaml --cfg
cfg/training/yolov7-custom.yaml --name Data_Munging_visualisation --weights
data_processing/cfg/yolov7-tiny_pretarined_weights.pt")

print("Training is sucessful!!")
```

Predicition:

```
os.system("python wf_ml_prediction.py --weights trained_models/best.pt --conf 0.5
--img-size 640 --source data_processing/prediction_samples/video.mp4")
```