

Train YOLO v5 locally

*"Run training of YOLO v5 with custom datasets in the local machine: CPU & GPU.
Manipulate attributes to manage training process.
Demonstrate resulted charts after the training."*

Step 1: Create directory

Windows: open Anaconda Prompt.

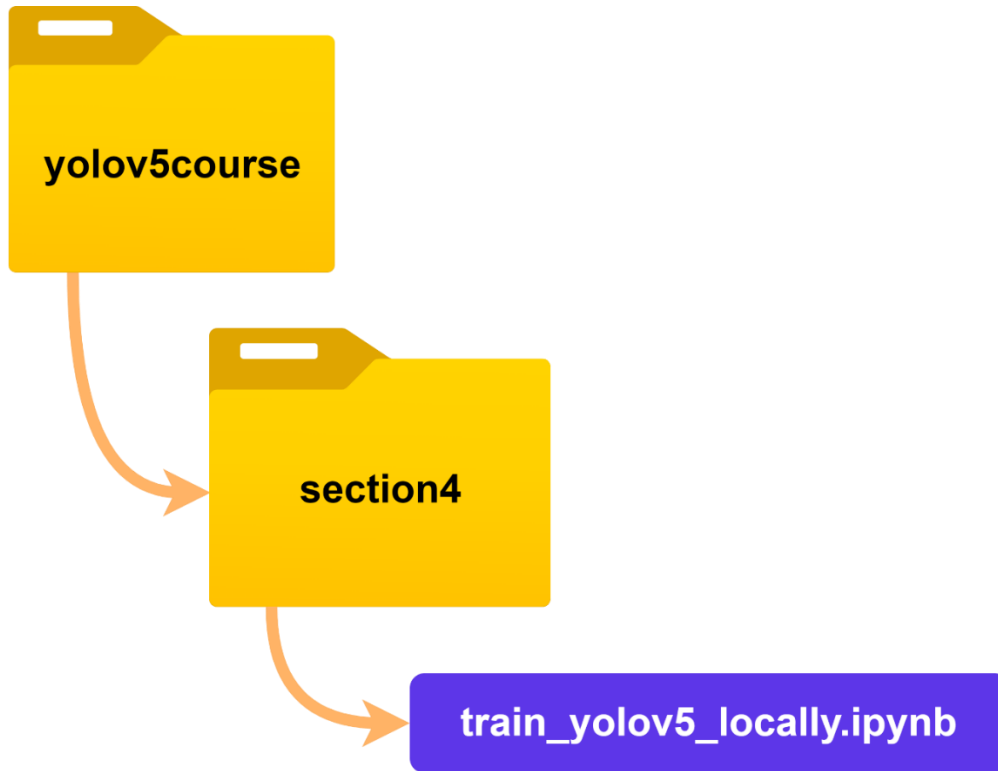
Mac or Linux: open terminal window.

All the commands are the same for Windows, Mac and Linux.

Command	Description
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>mkdir section4</code>	Creates "section4" directory

Step 2: Download code file

Go to resources of this lecture and download code file.
Place this file into created "section4" directory, that is inside "yolov5course" directory.
You should have following hierarchy:



Step 3: Run training in command line

Windows: open Anaconda Prompt.

Mac or Linux: open terminal window.

All the commands are the same for Windows, Mac and Linux.

Step 3.1: Update YOLO v5 framework

Update YOLO v5 framework	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
<code>git pull</code>	Updates YOLO v5 framework

Step 3.2: Update W&B toolkit

Update W&B toolkit	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>pip install wandb --upgrade</code>	Updates W&B toolkit

Step 3.3: Verify W&B connection

Verify W&B connection	
Command	Description
<code>wandb login</code>	Logs in into your W&B personal account
<code>wandb login --relogin</code>	Forces re-login to your or different W&B account
<code>wandb disabled</code>	Disables W&B toolkit
<code>wandb online</code>	Enables W&B toolkit

Step 3.4: Check if GPU is available

Check if GPU is available	
Navigate to the main YOLO v5 directory	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
(1) Verify successful installation of YOLO v5 to be used with GPU	
<pre>python -c "import torch; print(torch.cuda.is_available()); print(torch.cuda.device_count()); print(torch.cuda.current_device()); print(torch.cuda.get_device_name(0))"</pre>	

(1) Find more instructions in the following lecture of Section 1:

- ✓ **Quick Start: Test already trained YOLO v5**

Step 3.5: Show all available arguments to train YOLO v5 with

Show all available arguments to train YOLO v5 with	
Navigate to the main YOLO v5 directory	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
(2) Print all available arguments	
<code>python detect.py --help</code>	

(2) Navigate to `train.py` file and see all the default options for the arguments.

Step 3.6: Start training

Start training	
Navigate to the main YOLO v5 directory	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
(3) Train on custom dataset	
<code>python train.py --data ../section3/custom_dataset/yolov5dataset/dataset_updated.yaml --weights yolov5s.pt --device 0 --epochs 100 --batch-size 2 --workers 0</code>	

(3) Adjust path to `dataset_updated.yaml` file, if needed.

(3) To run on CPU only change this `--device 0` to this `--device cpu`

(4) Train on TS datasets with 4 classes

```
python train.py --data ../../section3/ts_yolo/yolov5dataset/ts4classes/dataset_updated.yaml --
weights yolov5s.pt --device 0 --epochs 100 --batch-size 2 --workers 0
```

(4) Adjust path to `dataset_updated.yaml` file, if needed.

(4) To run on CPU only change this `--device 0` to this `--device cpu`

(5) Train on TS datasets with 4 classes

```
python train.py --data ../../section3/ts_yolo/yolov5dataset/ts43classes/dataset_updated.yaml --
weights yolov5s.pt --device 0 --epochs 100 --batch-size 2 --workers 0
```

(5) Adjust path to `dataset_updated.yaml` file, if needed.

(5) To run on CPU only change this `--device 0` to this `--device cpu`

Step 3.7: Resume training

Resume training	
Navigate to the main YOLO v5 directory	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
(6) Resume training	
<code>python train.py --resume</code>	Automatically find latest checkpoint (searches in <code>yolov5</code> directory)
<code>python train.py --resume runs/train/exp16/weights/last.pt</code>	Specifies resume checkpoint

(6) You may not change settings when resuming, and no additional arguments other than `--resume` should be passed, with an optional path to the checkpoint you'd like to resume from. If no checkpoint is passed the most recently updated `last.pt` in your `yolov5` directory is automatically found and used.

Step 3.8: Validate training results

Validate training results	
Navigate to the main YOLO v5 directory	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
(7) Validate on custom dataset	
<pre>python val.py --data ../../section3/custom_dataset/yolov5dataset/dataset_updated.yaml --weights runs/train/exp7/weights/best.pt --batch-size 2 --workers 0 --device 0 --task test</pre>	

(7) Navigate to `val.py` file to see all the default options for the arguments.

(7) Adjust path to `dataset_updated.yaml` file and path to `best.pt`, if needed.

(7) To validate accuracy on `train` sub-dataset, change this `--task test` to this `--task train`

(7) To validate accuracy on `val` sub-dataset, change this `--task test` to this `--task val`

(7) To run on CPU only change this `--device 0` to this `--device cpu`

Step 3.9: Tensorboard

Investigate training results in Tensorboard	
Navigate to the main YOLO v5 directory	
Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>cd yolov5course</code>	Navigates to "yolov5course" directory
<code>cd section1</code>	Navigates to "section1" directory
<code>cd yolov5</code>	Navigates to "yolov5" directory
(8) Run Tensorboard	
<pre>tensorboard --logdir runs/train/exp7</pre>	

(8) Adjust path to directory `runs/train/exp7` with log file, if needed.

Step 4: Run training in Jupyter Notebook

Windows: open Anaconda Prompt.

Mac or Linux: open terminal window.

All the commands are the same for Windows, Mac and Linux.

Command	Description
<code>conda activate yolov5env</code>	Activates "yolov5env" environment
<code>jupyter notebook</code>	Runs Jupyter Notebook

Solutions to the most frequent issues

Issue 1

OMP: Error #15: Initializing libiomp5.dylib, but found libiomp5.dylib already initialized.

Solution 1

Add to `train.py` and `val.py` files following two code lines in the top:

```
import os
```

```
os.environ['KMP_DUPLICATE_LIB_OK']='True'
```

Link with more details, examples and other solutions:

✓ <https://github.com/dmlc/xgboost/issues/1715>

Issue 2

[WinError 1455] The paging file is too small for this operation to complete. Error loading "torch\lib\caffe2_detectron_ops_gpu.dll" or one of its dependencies

Solution 2

Add to the command following arguments:

```
--batch-size 2 --workers 2
```

Link with more details, examples and other solutions:

✓ <https://github.com/ultralytics/yolov3/issues/1643>

Issue 3

Error when resuming a training
train.py, line 567, in main train(opt.hyp, opt, device, callbacks)

Solution 3

Run one of the following commands:

```
python train.py --resume
```

automatically find latest checkpoint

```
python train.py --resume path/to/last.pt
```

specify resume checkpoint

Link with more details, examples and other solutions:

✓ <https://github.com/ultralytics/yolov5/issues/7394>

Issue 4

torch\multiprocessing\reductions.py, line 36, in __del__ AttributeError:
'NoneType' object has no attribute '_free_weak_ref'

Solution 4

Add to the command following argument:

```
--workers 0
```

Link with more details, examples and other solutions:

✓ <https://github.com/pytorch/pytorch/issues/74016>

Issue 5

Reusing TensorBoard on port 6006 (pid 17596), started 1 day, 23:56:21 ago.
(Use '!kill 17596' to kill it.)

Solution 5

- ✓ Clean previously ran Tensorboard sessions.
- ✓ Delete temp logs (not the permanent, that we got after the training).
- ✓ Move log file two levels up, so make the path as short as possible, e.g. place the log file into the directory `runs`

Link with more details, examples and other solutions:

✓ <https://github.com/tensorflow/tensorboard/issues/2481>

To delete temp logs in Linux, run following in Terminal:

```
rm -rf ./logs/
```

To delete temp logs in Windows, run following in CMD (command line) one-by-one:

```
taskkill /im tensorboard.exe /f
```

```
del /q %TMP%\tensorboard-info\*
```

Links

Check out additional links with extra information for further readings:

- ✓ [Yolo v5 pretrained checkpoints](#)
- ✓ [Yolo v5 models](#)
- ✓ [Tensorboard local logging](#)
- ✓ [Weights & Biases system of record for your model training](#)
- ✓ [Weights & Biases settings of your Profile](#)
- ✓ [Weights & Biases quick access to your API key](#)
- ✓ [Weights & Biases quickstart - Documentation](#)