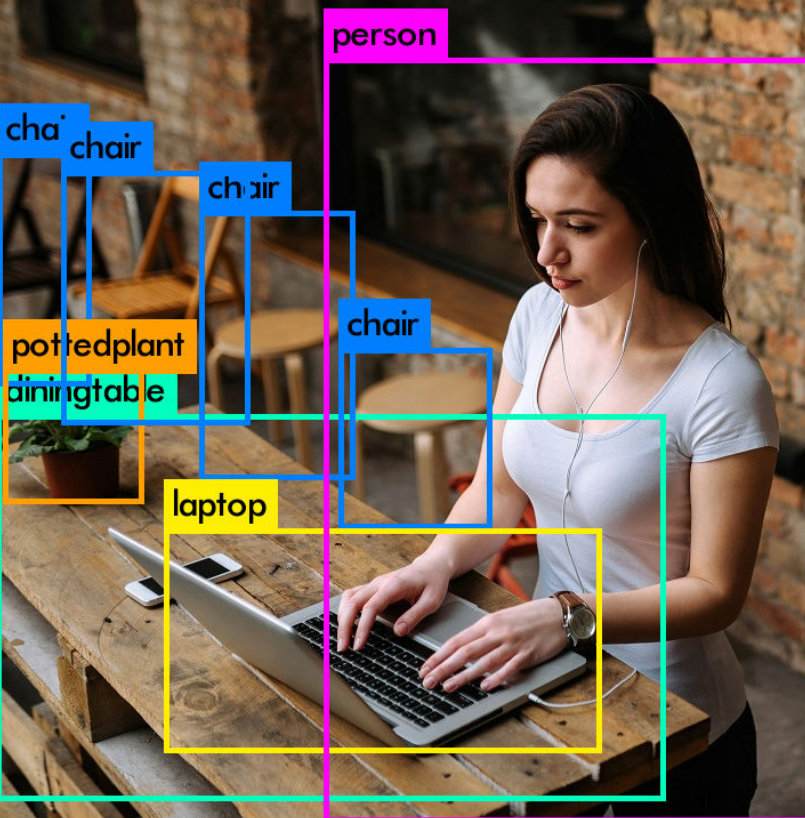


Training YOLO v3 for Objects Detection with Custom Data

*Setting up
cfg files*



Setting up configuration files

One more step is needed before training on two datasets from *Section-4* and *Section-5*, that is creating appropriate *configuration files*. We will use as a base existing *configuration file* and update only some specific numbers as described below leaving everything else the same.

Create four cfg files

Find existing *configuration file* in *cfg* folder inside root *Darknet* directory with name *yolov3.cfg*. Create four new files with names as shown below and copy content from *yolov3.cfg* to these files.

In *Section-4*, dataset for *Car*, *Bicycle wheel* and *Bus* was created. In *Section-5*, dataset of *Traffic Signs* was created. For every dataset two *configuration files* are needed, one for training and one for testing.

darknet/

cfg/

yolov3.cfg

darknet/

cfg/

yolov3_ts_train.cfg

yolov3_ts_test.cfg

yolov3_custom_train.cfg

yolov3_custom_test.cfg

Update batch and subdivisions parameters

Open two files for *training*, uncomment lines `# batch=64`, `# subdivisions=16` and delete lines `batch=1`, `subdivisions=1`. Save results. Open two files for *testing* and delete lines `# batch=64`, `# subdivisions=16`. Save results. You should have following lines as shown below.

yolov3_ts_train.cfg	yolov3_ts_test.cfg	yolov3_custom_train.cfg	yolov3_custom_test.cfg
[net]	[net]	[net]	[net]
# Training	# Testing	# Training	# Testing
batch=32	batch=1	batch=32	batch=1
subdivisions=16	subdivisions=1	subdivisions=16	subdivisions=1
width=416	width=416	width=416	width=416
height=416	height=416	height=416	height=416
...

It is also possible to set different numbers but only for training. For example, you can set in training files for *batch* and for *subdivisions* following numbers as shown below.

For training files

yolov3_ts_train.cfg and yolov3_custom_train.cfg

batch=	subdivisions=
64	8 or 16 or 32
32	8 or 16
16	8 or 4
8	4 or 2

If there is *memory overload error*, then it is needed to *decrease batch* number and *increase subdivisions*. Pay attention, *subdivisions* parameter has to be less than *batch*.

Update number of iterations for training

Next, it is needed to change *max_batches* that is total number of iterations for training and *steps* that are used for updating learning rate.

max_batches is updated according to the number of classes. General equation is as following:

$$\text{max_batches} = \text{classes} * 2000$$

(but not less than 4000)

steps are calculated as 80% and 90% from *max_batches*.

For example, if number of classes is equal to 2, then:

- *max_batches*=4000
- *steps*=3200,3600

For *Traffic Signs* dataset there are 4 classes, and for *Custom* dataset (Car, Bicycle wheel, Bus) there are 3 classes. You should have following numbers as shown below.

yolov3_ts_train.cfg	yolov3_ts_test.cfg	yolov3_custom_train.cfg	yolov3_custom_test.cfg
[net]	[net]	[net]	[net]
...
max_batches=8000	max_batches=8000	max_batches=6000	max_batches=6000
steps=6400,7200	steps=6400,7200	steps=4800,5400	steps=4800,5400
...

Update number of classes in 3 [yolo] layers and filters in 3 [convolutional] layers

Next, it is needed to update number of *classes* in every of three [yolo] layers in the end of the configuration files. Also, it is needed to update number of *filters* in [convolutonal] layers right before such every [yolo] layers but *not anywhere else*. It is needed in order to properly connect [convolutonal] layer that is right before [yolo] layer in accordance with number of *classes* in dataset.

General equation that represents how to calculate proper number of *filters* in three [convolutonal] layers right before every of three [yolo] layers is as following:

$$\text{filters} = (\text{classes} + \text{coordinates} + 1) * \text{masks}$$

For example, for COCO dataset it will be as following:

$$\text{filters} = (80 + 5) * 3 = 255$$

Number of *coordinates* and number of *masks* are the same for our four *configuration files*, and we will not change them. We change only number of *classes*.

For *Traffic Signs* dataset there are 4 classes:

$$\text{filters} = (4 + 5) * 3 = 27$$

For *Custom* dataset (Car, Bicycle wheel, Bus) there are 3 classes:

$$\text{filters} = (3 + 5) * 3 = 24$$

You should have following numbers as shown below.

yolov3_ts_train.cfg	yolov3_ts_test.cfg	yolov3_custom_train.cfg	yolov3_custom_test.cfg
[convolutional] size=1 stride=1 pad=1 filters=27 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=27 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=24 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=24 activation=linear
[yolo] classes=4 ...	[yolo] classes=4 ...	[yolo] classes=3 ...	[yolo] classes=3 ...
[convolutional] size=1 stride=1 pad=1 filters=27 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=27 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=24 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=24 activation=linear
[yolo] classes=4 ...	[yolo] classes=4 ...	[yolo] classes=3 ...	[yolo] classes=3 ...
[convolutional] size=1 stride=1 pad=1 filters=27 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=27 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=24 activation=linear	[convolutional] size=1 stride=1 pad=1 filters=24 activation=linear
[yolo] classes=4 ...	[yolo] classes=4 ...	[yolo] classes=3 ...	[yolo] classes=3 ...

Parameters inside configuration file

Inside *yolo_v3.cfg* we have parameters that are used for training and testing. Some of them are described below. Read full description in **Useful Links** section below.

[net] section:

- `batch=64` – number of samples that will be processed in one batch
- `subdivisions=16` – number of *mini batches* in one batch; GPU processes *mini batch samples at once*; the weights will be updated for batch samples, that is 1 iteration processes batch images
- `width=608` – every image will be resized during training and testing to this number
- `height=608` – every image will be resized during training and testing to this number
- `channels=3` – every image will be converted during training and testing to this number

Optimization:

- `momentum=0.9` – hyperparameter for optimizer that defines how much history will influence further updating of weights
- `decay=0.0005` – decay the learning rate over the period of the training
- `max_batches = 500200` – total number of iterations
- `learning_rate=0.001` – initial learning rate for training

Training:

- `angle=0` – parameter that randomly *rotates* images during training
- `saturation=1.5` – parameter that randomly *changes saturation* of images during training
- `exposure=1.5` – parameter that randomly *changes brightness* of images during training
- `hue=.1` – parameter that randomly *changes hue* of images during training

Useful Links

Check out additional links with detailed explanation of algorithm and other useful information for further reading:

- [1] [Parameters in \[net\] section](#) – description of parameters inside *configuration file* in the section *[net]*
- [2] [Parameters in layers](#) – description of parameters inside *configuration file* in the layers