

---

# **YOLOv3 Documentation**

***Release 0.0.1***

**Anthony DeGennaro**

**Dec 30, 2018**



## CONTENTS:

1	YOLOv3 train	1
2	YOLOv3 detect	3
3	YOLOv3 InputFile	5
4	YOLOv3 models	7
5	YOLOv3 NetworkTrainer	9
6	YOLOv3 Target	11
7	YOLOv3 fcn_sigma_rejection	13
8	YOLOv3 fcn_sigma_rejection	15
9	YOLOv3 datasetProcessing	17
10	YOLOv3 datasets	19
11	YOLOv3 utils	21
12	YOLOv3 utils_xview	23
13	YOLOv3 unittests	25
14	Indices and tables	27
	Python Module Index	29
	Index	31



## YOLOV3 TRAIN

`src.train2.main(inputs)`

Main driver script for training the YOLOv3 network.

**Inputs:**

*inputs*: an input file formatted according to the InputFile class

**Outputs:**

*inputs.outdir/results.txt*: output metrics for each training epoch

*inputs.loaddir/latest.pt*: checkpoint file for latest network configuration

*inputs.loaddir/best.pt*: checkpoint file for best current network configuration

*inputs.loaddir/backup.pt*: checkpoint file for backup purposes



## YOLOV3 DETECT

```
class src.detect.ConvNetb(num_classes=60)
```

```
    forward(x)
```

Defines the computation performed at every call.

Should be overridden by all subclasses.

---

**Note:** Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

---





## YOLOV3 INPUTFILE



## YOLOV3 MODELS

**class** `src.models.Darknet` (*config\_path, img\_size=416*)  
YOLOv3 object detection model

**forward** (*x, targets=None, requestPrecision=False, weight=None, epoch=None*)  
Defines the computation performed at every call.  
Should be overridden by all subclasses.

---

**Note:** Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

---

**class** `src.models.EmptyLayer`  
Placeholder for 'route' and 'shortcut' layers

**class** `src.models.YOLOLayer` (*anchors, nC, img\_dim, anchor\_idxs*)

**forward** (*p, targets=None, requestPrecision=False, weight=None, epoch=None*)  
Defines the computation performed at every call.  
Should be overridden by all subclasses.

---

**Note:** Although the recipe for forward pass needs to be defined within this function, one should call the `Module` instance afterwards instead of this since the former takes care of running the registered hooks while the latter silently ignores them.

---

`src.models.create_modules` (*module\_defs*)  
Constructs module list of layer blocks from module configuration in *module\_defs*

`src.models.create_yolo_config_file` (*template\_file\_path, output\_config\_file\_path, n\_anchors,*  
*n\_classes, anchor\_coordinates*)  
Creates a yolo-v3 layer configuration file from desired options

`src.models.parse_model_config` (*path*)  
Parses the yolo-v3 layer configuration file and returns module definitions



## YOLOV3 NETWORKTRAINER



## YOLOV3 TARGET

```
class src.targets.Target.Target(inputs)
```

Class for handling target pre-processing tasks.

```
apply_mask_to_filtered_data()
```

Method to apply mask to filtered data variables.

```
compute_bounding_box_clusters_using_kmeans(n_clusters)
```

Method to compute bounding box clusters using kmeans.

**Inputs:**

*n\_clusters*: number of desired kmeans clusters

```
compute_cropped_data()
```

Method to crop image data based on the width and height. Filtered variables are then computed based on the updated image coordinates.

```
compute_filtered_data_mask()
```

Method to compute filtered data by applying several filtering operations.

```
compute_filtered_variables_from_filtered_coords()
```

Method to compute filtered variables from filtered coordinates.

```
compute_filtered_variables_from_filtered_xy()
```

Method to compute filtered variables from filtered xy.

```
compute_image_weights_with_filtered_data()
```

Method to compute image weights from filtered data. Weight is simply inverse of class frequency.

```
edge_requirements(w_lim, h_lim, x2_lim, y2_lim)
```

Method to compute filtering based on edge specifications.

**Inputs:**

*w\_lim*: limit for image width

*h\_lim*: limit for image height

*x2\_lim*: limit for image x2

*y2\_lim*: limit for image y2

**Outputs:**

indices where filtered variables satisfy the dimension requirements.

**load\_target\_file()**

Method to load a targetfile of type specified in the input file. Supported types: .json.

**manual\_dimension\_requirements** (*area\_lim, w\_lim, h\_lim, AR\_lim*)

Method to compute filtering based on specified dimension requirements.

**Inputs:**

*area\_lim*: limit for image area

*w\_lim*: limit for image width

*h\_lim*: limit for image height

*AR\_lim*: limit for image aspect ratio

**Outputs:**

indices where filtered variables satisfy the dimension requirements.

**process\_target\_data()**

Method to perform all target processing.

**set\_image\_w\_and\_h()**

Method to set width and height of images associated with targets.

**sigma\_rejection\_indices** (*filtered\_data*)

Method to compute a mask based on a sigma rejection criterion.

**Inputs:**

*filtered\_data*: data to which sigma rejection is applied and from which mask is computed

**Outputs:**

*mask\_reject*: binary mask computed from sigma rejection

**strip\_image\_number\_from\_chips\_and\_files()**

Method to strip numbers from image filenames from both chips and files.



## YOLOV3 FCN\_SIGMA\_REJECTION



## YOLOV3 FCN\_SIGMA\_REJECTION



## YOLOV3 DATASETPROCESSING



## YOLOV3 DATASETS





## YOLOV3 UTILS

`utils.utils.bbox_iou (box1, box2, x1y1x2y2=True)`

Returns the IoU of two bounding boxes

`utils.utils.build_targets (pred_boxes, pred_conf, pred_cls, target, anchor_wh, nA, nC, nG, requestPrecision)`  
returns nGT, nCorrect, tx, ty, tw, th, tconf, tcls

`utils.utils.compute_ap (recall, precision)`

Compute the average precision, given the recall and precision curves. Code originally from <https://github.com/rbgirshick/py-faster-rcnn>. # Arguments

recall: The recall curve (list). precision: The precision curve (list).

**# Returns** The average precision as computed in py-faster-rcnn.

`utils.utils.load_classes (path)`

Loads class labels at 'path'



## YOLOV3 UTILS\_XVIEW



## YOLOV3 UNITTESTS

```
class tests.unittests.DataProcessingTests (methodName='runTest')
```

```
    setUp ()
```

```
        Hook method for setting up the test fixture before exercising it.
```

```
class tests.unittests.DatasetTests (methodName='runTest')
```

```
    setUp ()
```

```
        Hook method for setting up the test fixture before exercising it.
```

```
class tests.unittests.GPUtests (methodName='runTest')
```

```
    setUp ()
```

```
        Hook method for setting up the test fixture before exercising it.
```

```
class tests.unittests.ModelsTests (methodName='runTest')
```

```
    setUp ()
```

```
        Hook method for setting up the test fixture before exercising it.
```

```
class tests.unittests.TargetTests (methodName='runTest')
```

```
    setUp ()
```

```
        Hook method for setting up the test fixture before exercising it.
```



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## PYTHON MODULE INDEX

### S

- `src.detect`, 3
- `src.InputFile`, 5
- `src.models`, 7
- `src.NetworkTrainer`, 9
- `src.targets.fcn_sigma_rejection`, 13
- `src.targets.per_class_stats`, 15
- `src.targets.Target`, 11
- `src.train2`, 1

### t

- `tests.unittests`, 25

### u

- `utils.datasetProcessing`, 17
- `utils.datasets`, 19
- `utils.utils`, 21
- `utils.utils_xview`, 23



## A

`apply_mask_to_filtered_data()`  
(*src.targets.Target.Target method*), 11

## B

`bbox_iou()` (*in module utils.utils*), 21  
`build_targets()` (*in module utils.utils*), 21

## C

`compute_ap()` (*in module utils.utils*), 21  
`compute_bounding_box_clusters_using_kmeans()`  
(*src.targets.Target.Target method*), 11  
`compute_cropped_data()`  
(*src.targets.Target.Target method*), 11  
`compute_filtered_data_mask()`  
(*src.targets.Target.Target method*), 11  
`compute_filtered_variables_from_filtered_data()`  
(*src.targets.Target.Target method*), 11  
`compute_filtered_variables_from_filtered_xy()`  
(*src.targets.Target.Target method*), 11  
`compute_image_weights_with_filtered_data()`  
(*src.targets.Target.Target method*), 11  
`ConvNetb` (*class in src.detect*), 3  
`create_modules()` (*in module src.models*), 7  
`create_yolo_config_file()` (*in module src.models*), 7

## D

`Darknet` (*class in src.models*), 7  
`DataProcessingTests` (*class in tests.unittests*), 25  
`DatasetTests` (*class in tests.unittests*), 25

## E

`edge_requirements()` (*src.targets.Target.Target method*), 11  
`EmptyLayer` (*class in src.models*), 7

## F

`forward()` (*src.detect.ConvNetb method*), 3  
`forward()` (*src.models.Darknet method*), 7  
`forward()` (*src.models.YOLOLayer method*), 7

## G

`GPUtests` (*class in tests.unittests*), 25

## L

`load_classes()` (*in module utils.utils*), 21  
`load_target_file()` (*src.targets.Target.Target method*), 12

## M

`main()` (*in module src.train2*), 1  
`minimal_dimension_requirements()`  
(*src.targets.Target.Target method*), 12  
`ModelsTests` (*class in tests.unittests*), 25

## P

`parse_model_config()` (*in module src.models*), 7  
`process_target_data()` (*src.targets.Target.Target method*), 12

## S

`set_image_w_and_h()` (*src.targets.Target.Target method*), 12  
`setUp()` (*tests.unittests.DataProcessingTests method*), 25  
`setUp()` (*tests.unittests.DatasetTests method*), 25  
`setUp()` (*tests.unittests.GPUtests method*), 25  
`setUp()` (*tests.unittests.ModelsTests method*), 25  
`setUp()` (*tests.unittests.TargetTests method*), 25  
`sigma_rejection_indices()`  
(*src.targets.Target.Target method*), 12  
`src.detect` (*module*), 3  
`src.InputFile` (*module*), 5  
`src.models` (*module*), 7  
`src.NetworkTrainer` (*module*), 9  
`src.targets.fcn_sigma_rejection` (*module*), 13  
`src.targets.per_class_stats` (*module*), 15  
`src.targets.Target` (*module*), 11  
`src.train2` (*module*), 1  
`strip_image_number_from_chips_and_files()`  
(*src.targets.Target.Target method*), 12

## T

`Target` (*class in `src.targets.Target`*), 11  
`TargetTests` (*class in `tests.unittests`*), 25  
`tests.unittests` (*module*), 25

## U

`utils.datasetProcessing` (*module*), 17  
`utils.datasets` (*module*), 19  
`utils.utils` (*module*), 21  
`utils.utils_xview` (*module*), 23

## Y

`YOLOLayer` (*class in `src.models`*), 7