

# 在 YOLOv3 上训练自己的图片集

作者：wych 日期：2018-09-02

## 1. 目录树

新建项目文件夹 project\_yolov3，建立如下目录树。（并不是一定要这样组织，只是很多项目，包括官网上面的，都把文件放到 darknet 的编译目录，显得文件组织结构不清晰）

project\_yolov3

- data
- - images
- - labels
- - .names
- - train\_list.txt
- - test\_list.txt
- cfg
- - .data
- - .cfg
- backup

## 2. 准备图片、标签文件

将准备训练的图片放在 images 文件夹；对应的标签放在 labels 文件夹。

每张图片对应一个同名的 label txt 文件，比如 images/abc.jpg -> labels/abc.txt。

labels 文件每行存放一个标注的 Bounding Box，一行五个数分别代表类别（从 0 开始编号），BoundingBox 中心 X 坐标，中心 Y 坐标，宽，高。这些坐标都是 0~1 的相对坐标。比如，图片的宽和高分别为 WIDTH 和 HEIGHT 像素，BoundingBox 中心 X、Y 分别为 x、y 像素，宽、高分别为 w、h 像素，label 文件里面的值可以用下面的公式计算：

$yolo\_x = x / WIDTH$

$yolo\_y = y / HEIGHT$

$yolo\_w = w / WIDTH$

yolo\_h = h/HEIGHT

未验证：

(1) 标签文件可以为空，表示对应的图片里没有目标

(2) 如果图片较多，图片和标签可以在各自目录下以相同的目录树存储。

比如：

images

- class1

- class2

labels

- class1

- class2

### 3. .names 文件

.names 文件存放训练的物体类别，比如 car、person 等，每行一个，和标签文件里面的类别编号对应，第一行在标签文件里编号为 0。

### 4. train\_list.txt 和 test\_list.txt

这两个文件，文件名称并不固定，分别存放训练和测试的图片路径，文件里每行存放一个图片路径。比如，

PATH/TO/YOUR/PROJECT/project\_yolov3/data/images/abc.jpg

### 5. .data 文件

.data 文件存放训练的物体类别数目，train\_list.txt 文件路径等。

例子：

classes= 4

train = PATH/TO/YOUR/PROJECT/project\_yolov3/data/train\_list.txt

valid = PATH/TO/YOUR/PROJECT/project\_yolov3/data/test\_list.txt

names = PATH/TO/YOUR/PROJECT/project\_yolov3/data/XXX.names

backup= PATH/TO/YOUR/PROJECT/project\_yolov3/backup

(backup 目录下文叙述)

## 6. .cfg 文件。

.cfg 文件是 darknet 描述神经网络结构的文件。

在 darknet 编译目录下面的 cfg 文件夹，有很多.cfg，比如 yolov2.cfg，yolov3.cfg 等。

需要根据自己训练的物体种类数目，做一些修改。

```
[convolutional]
```

```
size=1
```

```
stride=1
```

```
pad=1
```

```
filters=27###75
```

```
activation=linear
```

```
[yolo]
```

```
mask = 6,7,8
```

```
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326
```

```
classes=4###20
```

```
num=9
```

```
jitter=.3
```

```
ignore_thresh = .5
```

```
truth_thresh = 1
```

```
random=0###1
```

classes 的值改为物体种类数目，filters 的值改为  $3 * (\text{classes} + 5)$ ，比如 classes = 4，则 filters =  $3 * (\text{classes} + 5) = 3 * (4 + 5) = 27$ 。在.cfg 文件中搜索 “yolo”，每一个对应的 classes，filters 都需要修改。

其它的值，比如 batch 大小，输入网络的图片的像素等，可以根据自己的需要修改，当出现

“cuda, out of memory”错误，是显存不够的错误，可以改小 batch，增大 subdivisions；channels 是图片通道数，比如灰度图像为 1，rgb 图像为 3。 .cfg 文件的其它部分可以在对网络结构比较熟悉之后修改，甚至定义自己的网络结构。

batch=64

subdivisions=16 #8

width=416

height=416

channels=3

## 7. backup 目录

backup 目录存放训练过程中的权值备份，每 100 次保存一个 XXX.weights，后面产生的会覆盖之前的，所以它一直是最新的，前 9 个会同时命名为 XXX\_100.weights~XXX\_900.weights，可以作为断点备份继续训练。

## 8. 下载权值文件，训练，测试

参照官网 <https://pjreddie.com/darknet/yolo/> 上的说明，下载 darknet53.conv.74 权值文件（这是 YOLO 作者在 ImageNet 上预训练的权值）。

训练：

```
./darknet detector train PATH/TO/YOUR/PROJECT/project_yolov3/cfg/XXX.data  
PATH/TO/YOUR/PROJECT/project_yolov3/cfg/XXX.cfg darknet53.conv.74
```

测试：

```
./darknet detector test PATH/TO/YOUR/PROJECT/project_yolov3/cfg/XXX.data  
PATH/TO/YOUR/PROJECT/project_yolov3/cfg/XXX.cfg XXX.weights XXX.jpg
```