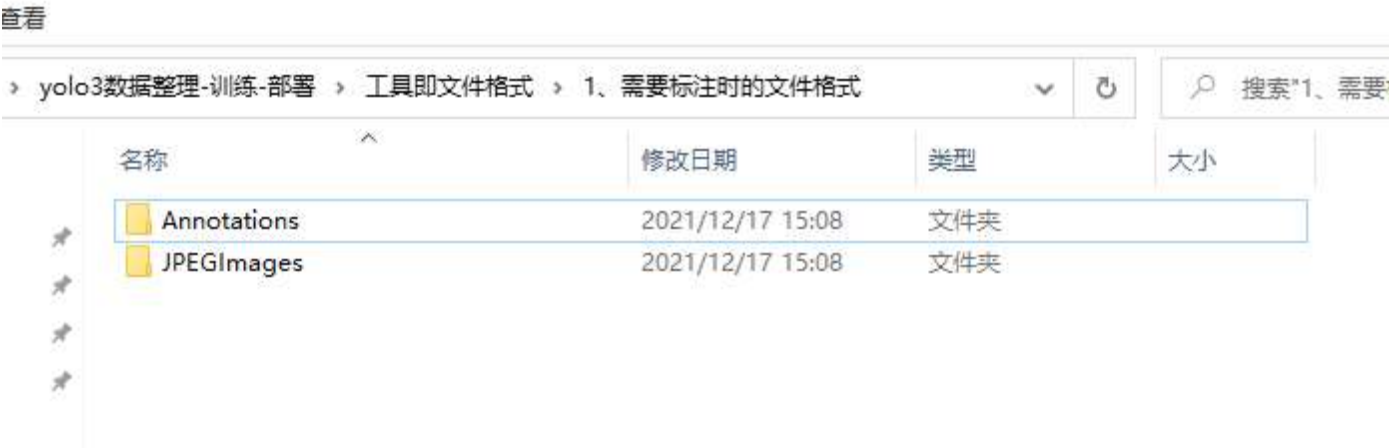


yolo3数据整理-训练-部署-示例读码器训练

make by haifeny

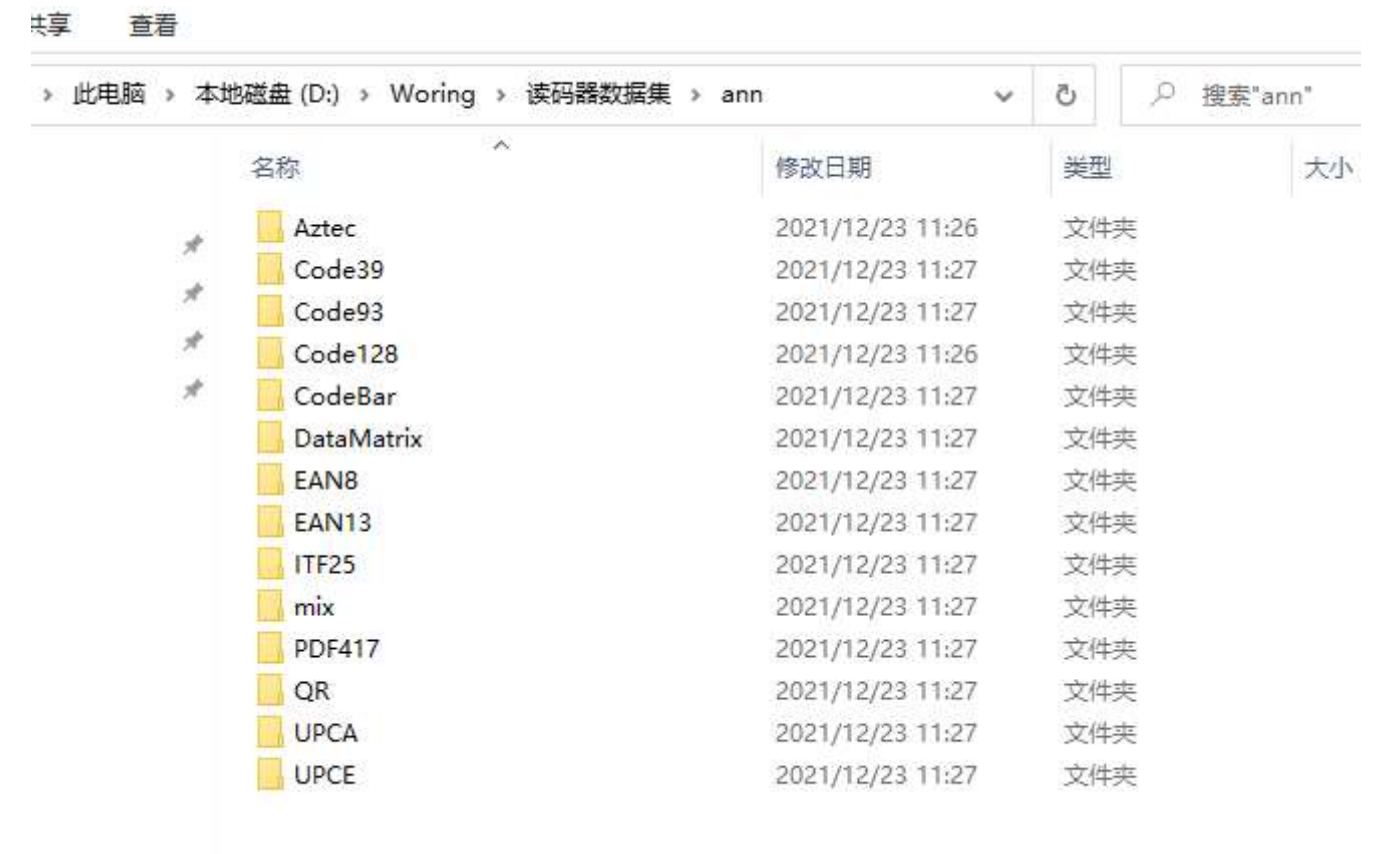
数据整理

创建Annotations和JPEGImages两个文件夹



图像拷贝整理

获取到的数据这样：



查看

图片工具

磁盘 (D:) > Woring > 读码器数据集 > image > Aztec

搜索"Aztec"

10200000

10200001

10200002

10200003

10200004

10200005

10200006

10200007

10200008

10200009

10200010

10200011

10200012

10200013

10200014

10200015

10200016

10200017

10200018

10200019

10200020

10200021

10200022

10200023

10200024

10200025

10200026

10200027

看














本地磁盘 (D:) > Woring > 读码器数据集 > image

搜索"image"

名称	修改日期	类型	大小
Aztec	2021/12/23 11:27	文件夹	
Code39	2021/12/23 11:28	文件夹	
Code93	2021/12/23 11:28	文件夹	
Code128	2021/12/23 11:28	文件夹	
CodeBar	2021/12/23 11:29	文件夹	
DataMatrix	2021/12/23 11:29	文件夹	
EAN8	2021/12/23 11:30	文件夹	
EAN13	2021/12/23 11:29	文件夹	
ITF25	2021/12/23 11:30	文件夹	
mix	2021/12/23 11:31	文件夹	
PDF417	2021/12/23 11:31	文件夹	
QR	2021/12/23 11:31	文件夹	
UPCA	2021/12/23 11:32	文件夹	
UPCE	2021/12/23 11:32	文件夹	

本地磁盘 (D:) > Woring > 读码器数据集 > ann > Aztec

搜索"Aztec"

名称	修改日期	类型	大小
 10200000	2021/12/16 8:42	XML 文档	1 KB
 10200001	2021/12/16 14:28	XML 文档	3 KB
 10200002	2021/12/16 14:29	XML 文档	1 KB
 10200003	2021/12/16 14:29	XML 文档	1 KB
 10200004	2021/12/16 8:54	XML 文档	1 KB
 10200005	2021/12/16 8:55	XML 文档	1 KB
 10200006	2021/12/16 8:56	XML 文档	1 KB
 10200007	2021/12/16 14:29	XML 文档	1 KB
 10200008	2021/12/16 8:56	XML 文档	1 KB
 10200009	2021/12/16 8:56	XML 文档	1 KB
 10200010	2021/12/16 8:57	XML 文档	1 KB
 10200011	2021/12/16 8:57	XML 文档	1 KB
 10200012	2021/12/16 14:29	XML 文档	1 KB

将ann下一级所有(14个文件夹内)文件夹内的标注xml文件(00200000.xml)拷贝到yolo3项目下的data/Annotations文件夹中,将image下一级所有(14个文件夹内)文件夹内的图像bmp文件拷贝到yolo3项目下的data/JPEGImages文件夹中

yolo数据集格式转换1

由于图像是单通道的bmp图像，而模型需要使用3通道的图像，所以需要将图像转为三通道。

在yolo3/data下创建.py运行文件，加入如下代码：

In []:

```
import cv2
import numpy as np
import os
import random
from os import listdir, getcwd

path = getcwd()
imgfilepath = path + '/JPEGImages'
aimfilepath = path + '/images'

total_img = os.listdir(imgfilepath)

for name in total_img:
    namepath = imgfilepath + "/" + name
    img = cv2.imread(namepath)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img2 = np.zeros_like(img)
    img2[:, :, 0] = gray
    img2[:, :, 1] = gray
    img2[:, :, 2] = gray
    imgusename = name.split(".", 1)
    save = aimfilepath + "/" + imgusename[0] + ".jpg"
    cv2.imwrite(save, img2)
```

环境搭建

下载yolo3模型，下载第八版

<https://github.com/ultralytics/yolov3/releases> (<https://github.com/ultralytics/yolov3/releases>)



v8 - Final Darknet Compatible Release

This is the final release of the darknet-compatible version of the <https://github.com/ultralytics/yolov3> repository. This release is backwards-compatible with darknet *.cfg files for model configuration.

All pytorch (*.pt) and darknet (*.weights) models/backbones available are attached to this release in the Assets section below.

Breaking Changes

There are no breaking changes in this release.

Bug Fixes

- Various

Added Functionality

- Various

- 1、打开解压的文件夹下，在顶部路径栏处输入cmd，进入命令行提示符
- 2、输入：conda create -n yolo3 python==3.7.2
- 3、输入：conda activate yolo3

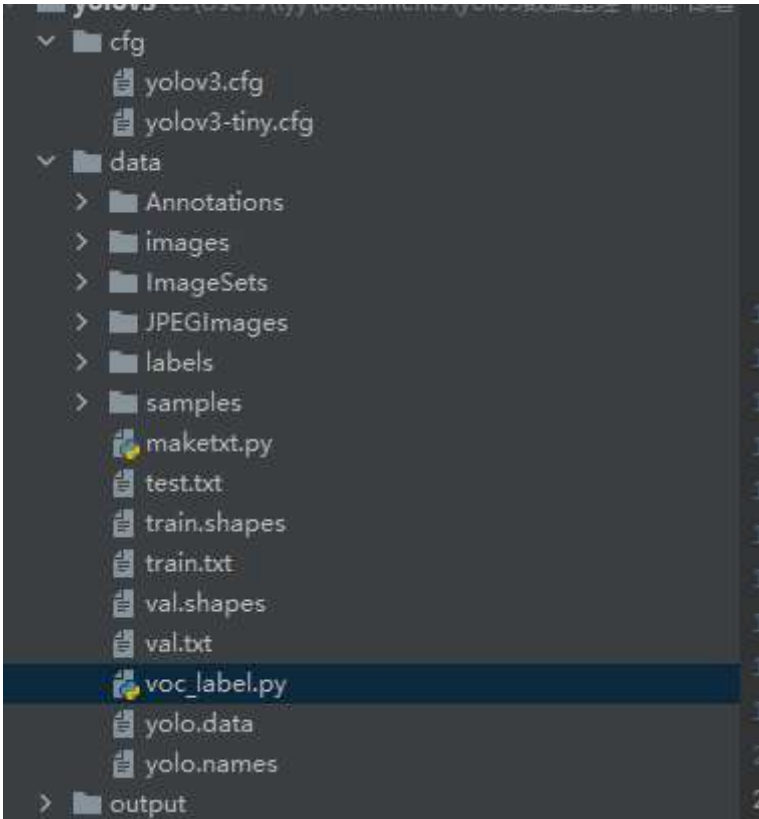
```
4、输入：CPU: conda install pytorch torchvision torchaudio cpuonly -c pytorch
GPU:conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch
4、输入： pip install requirements.txt
```

最后，在pycharm中加载yolo3的环境，打开yolo3的代码

建议：环境自己搭建，代码使用本次提供的代码，因为本代码增加数据处理、修改训练参数等操作，而github上没有提供针对自己数据集的训练部分，仅能运行测试demo，开启不了训练。

yolo数据集格式转换2

找到yolo3文件夹下的data文件夹



1、先运行makeTxt.py,将数据分成训练集，测试集和验证集

训练-部署 > 代码 > 训练代码 > yolov3 > data > ImageSets				搜索"Image"	
名称	修改日期	类型	大小		
test	2021/12/8 11:11	文本文档	1 KB		
train	2021/12/8 11:11	文本文档	5 KB		
trainval	2021/12/8 11:11	文本文档	1 KB		
val	2021/12/8 11:11	文本文档	1 KB		

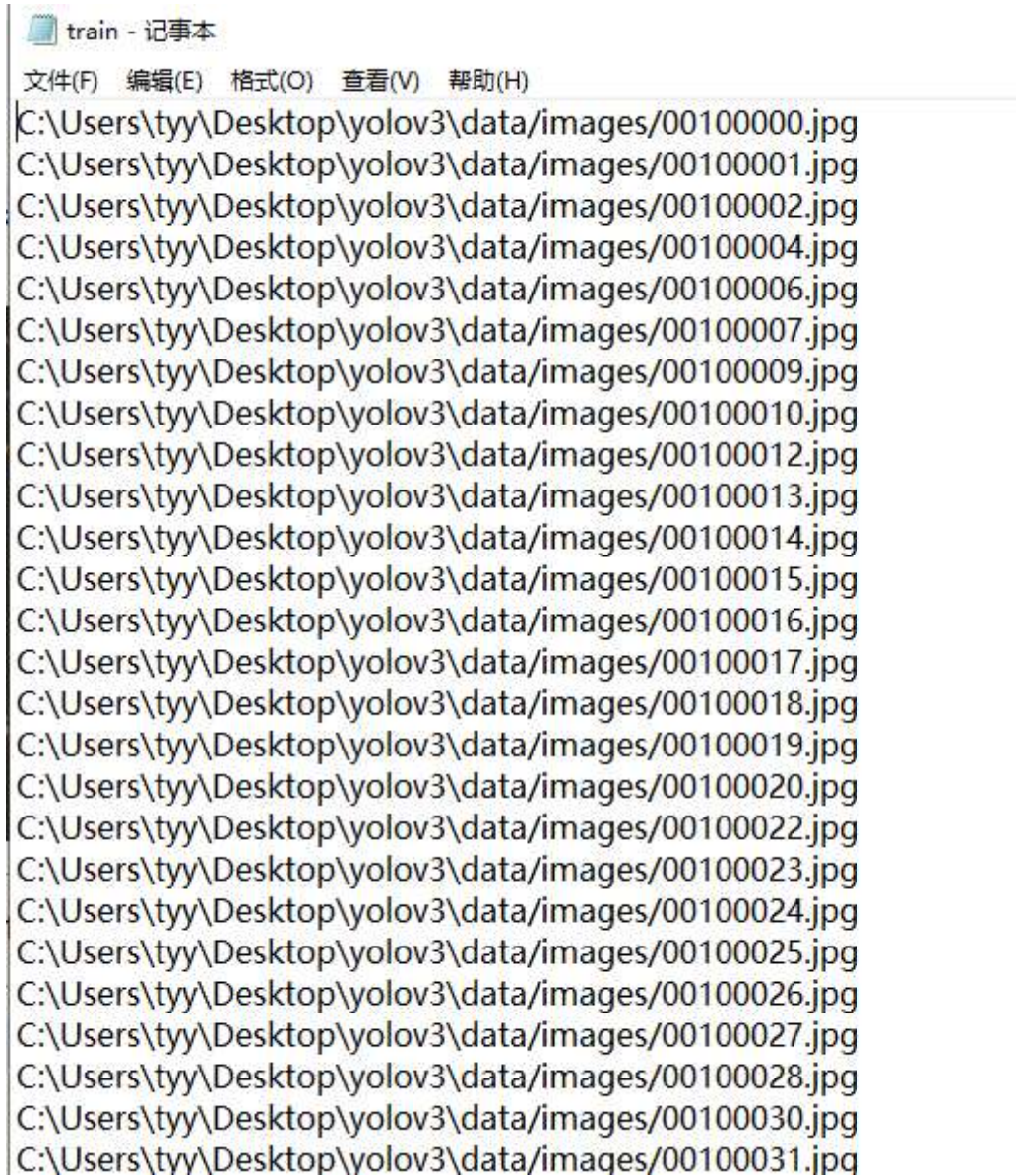


2、在voc_label.py中第9行将classes = []改为你的类别，如本次改为classes = ['BarCode','DataCode']



3、在运行voc_label.py,得到labels的具体内容以及data目录下的train.txt, test.txt, val.txt

test	2021/12/8 14:06	文本文档	5 KB
train.shapes	2021/12/8 13:49	SHAPES 文件	8 KB
train	2021/12/8 14:06	文本文档	47 KB
val.shapes	2021/12/8 14:06	SHAPES 文件	1 KB
val	2021/12/8 14:06	文本文档	1 KB



```
train - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
C:\Users\tyy\Desktop\yolov3\data/images/00100000.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100001.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100002.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100004.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100006.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100007.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100009.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100010.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100012.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100013.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100014.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100015.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100016.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100017.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100018.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100019.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100020.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100022.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100023.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100024.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100025.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100026.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100027.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100028.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100030.jpg
C:\Users\tyy\Desktop\yolov3\data/images/00100031.jpg
```

4、修改该目录下yolo.names文件，该目录下为你需要的标签类别：一行一个类别

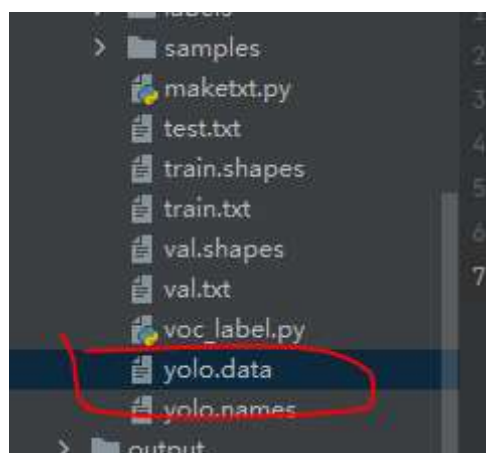
面包屑: yolo3数据整理-训练-部署 > 代码 > 训练代码 > yolov3 > data

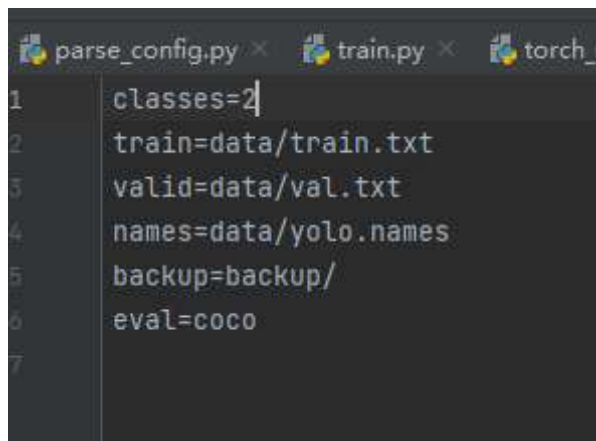
名称	修改日期	类型	大小
Annotations	2021/12/17 14:05	文件夹	
images	2021/12/17 14:06	文件夹	
ImageSets	2021/12/17 14:06	文件夹	
JPEGImages	2021/12/17 14:06	文件夹	
labels	2021/12/17 14:07	文件夹	
samples	2021/12/17 14:07	文件夹	
PC maketxt	2021/12/7 16:32	JetBrains PyChar...	1 KB
test	2021/12/8 14:06	文本文档	5 KB
train.shapes	2021/12/8 13:49	SHAPES 文件	8 KB
train	2021/12/8 14:06	文本文档	47 KB
val.shapes	2021/12/8 14:06	SHAPES 文件	1 KB
val	2021/12/8 14:06	文本文档	1 KB
PC voc_label	2021/12/8 14:05	JetBrains PyChar...	2 KB
yolo.data	2021/12/7 17:08	DATA 文件	1 KB
yolo.names	2021/12/7 17:03	NAMES 文件	1 KB

```
train.py x yolo.data x To3aisle.py x yolov3-tiny.cfg x yolo.names x ma
1 Barcode
2 DataCode
```

注意 标签的排列顺序必须与voc_label.py,第9行中classes = ['BarCode','DataCode']的类别标签排列顺序一致。

修改该目录下yolo.data中的classes值，其代表了类别的数量。在读码器项目中，填2





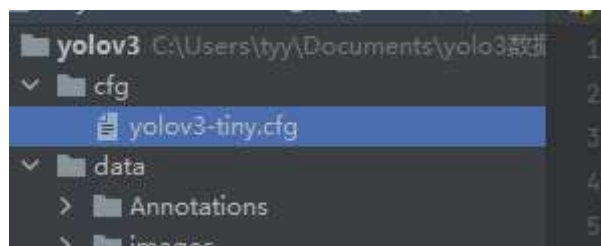
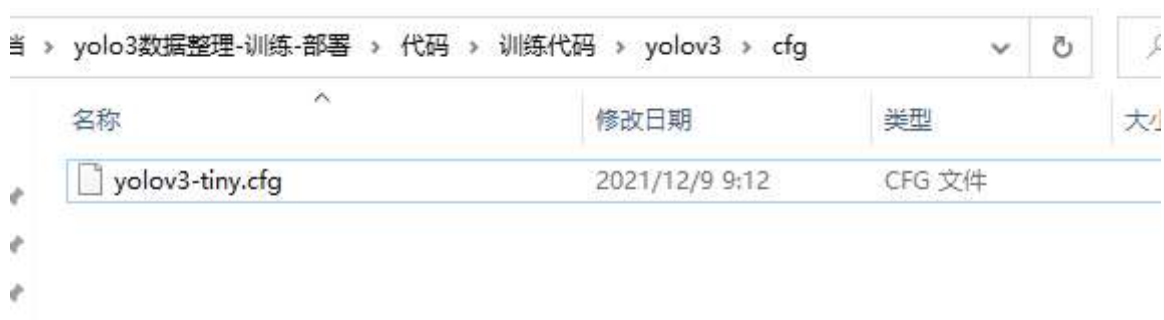
```
1 classes=2  
2 train=data/train.txt  
3 valid=data/val.txt  
4 names=data/yolo.names  
5 backup=backup/  
6 eval=coco  
7
```

5、打开yolo3下的cfg目录下的yolo3-tiny.cfg，修改[yolo3]下的classes，此为你需要训练的分类数

修改其上[convolutional]处的filters，计算公式

$\text{filter} = 3 * (\text{classes} + 5)$

eg: 此处 $\text{filter} = 3 * (2 + 5) = 21$



yolo3-tiny.cfg下工4处要修改:

```

118
119 [convolutional]
120 size=1
121 stride=1
122 pad=1
123 filters=21 ← 1
124 activation=linear
125
126 [yolo]
127 mask = 3,4,5
128 anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
129 classes=2 ← 2
130 num=6
131 jitter=.3
132 ignore_thresh = .7
133

```

```

163 [convolutional]
164 size=1
165 stride=1
166 pad=1
167 filters=21 ← 3
168 activation=linear
169
170 [yolo]
171 mask = 0,1,2
172 anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
173 classes=2 ← 4
174 num=6
175 jitter=.3
176 ignore_thresh = .7
177 truth_thresh = 1
178 random=1
179

```

训练

使用本次配置好的代码即可，点击yolo3下的train.py

对原代码修改的地方如下所示，共4处，同时也是需要注意的点：

- 1、394行，epoch：是训练迭代次数，建议一般设为270次。
- 2、395行，batch_size：是一次迭代输入的图像数量，在此由于本计算机内存小，故设为8，如果是GPU训练，可以设为16或者32。
- 3、409行，device，训练是cpu还是gpu。

重要参数说明参数：

- 1、394行，epoch：是训练迭代次数，建议一般设为270次。
- 2、395行，batch_size：是一次迭代输入的图像数量，在此由于本计算机内存小，故设为8，如果是GPU训练，可以设为16或者32。
- 3、396行，cfg：是配置参数文件的路径，已将配置好了，请勿修改。
- 4、397行，data：是训练数据的输入地址，将配置好了，请勿修改。
- 5、399行，img-size，实验过，该参数请勿修改。
- 6、409行，device，训练是cpu还是gpu。

```

392 if __name__ == '__main__':
393     parser = argparse.ArgumentParser()
394     parser.add_argument('--epochs', type=int, default=40) # 500200 batches at bs 16, 117263 COCO images = 273 epochs
395     parser.add_argument('--batch-size', type=int, default=8) # effective bs = batch_size * accumulate = 16 * 4 = 64
396     parser.add_argument('--cfg', type=str, default='cfg/yolov3-tiny.cfg', help='*.cfg path')
397     parser.add_argument('--data', type=str, default='data/yolo.data', help='*.data path')
398     parser.add_argument('--multi-scale', action='store_true', help='adjust (67% - 150%) img_size every 10 batches')
399     parser.add_argument('--img-size', nargs='+', type=int, default=[320, 640], help='[min_train, max_train, test]')
400     parser.add_argument('--rect', action='store_true', help='rectangular training')
401     parser.add_argument('--resume', action='store_true', help='resume training from last.pt')
402     parser.add_argument('--nosave', action='store_true', help='only save final checkpoint')
403     parser.add_argument('--notest', action='store_true', help='only test final epoch')
404     parser.add_argument('--evolve', action='store_true', help='evolve hyperparameters')
405     parser.add_argument('--bucket', type=str, default='', help='gsutil bucket')
406     parser.add_argument('--cache-images', action='store_true', help='cache images for faster training')
407     parser.add_argument('--weights', type=str, default='weights/yolov3-tiny.weights', help='initial weights path')
408     parser.add_argument('--name', default='', help='renames results.txt to results_name.txt if supplied')
409     parser.add_argument('--device', default='cpu', help='device id (i.e. 0 or 0,1 or cpu)')
410     parser.add_argument('--adam', action='store_true', help='use adam optimizer')
411     parser.add_argument('--single-cls', action='store_true', help='train as single-class dataset')
412     parser.add_argument('--freeze-layers', action='store_true', help='Freeze non-output layers')
413     opt = parser.parse_args()
  
```

在初始阶段建议将[394]行的epchs的default设为10，训练完成，检测一下是否满足要求，满足则将该值设为200，继续训练

模型部署

- 1、点击lovo3下的ptToweights.py,将模型转换成可调用格式
- 2、拷贝weights文件夹下的best.weights模型权重，data文件夹下的yolo.names标签文件，cfg文件夹下的yolov3-tiny.cfg模型参数，

到所要部署的c#软件中。例如本次示例中，拷贝到...部署代码\Yolo3\Yolo3\bin\x64\Debug\model文件夹下。使用yolo3调用接口demo跑程序