**深度学习--检测模型实验手册**

该实验分成两个部分：**训练过程**和**应用过程**。

带标注框的数据集

检测模型

训练模型参数

训练好的检测模型

一张未知图片

输出检测结果

训练过程

应用过程

图1 深度学习的训练和应用过程

**训练过程：**首先准备带检测标签的数据集，然后建立一个检测模型，模型的初始参数随机设置，将数据集输入模型，计算输出结果，将输出的检测结果与该数据真实的检测的差值作为参数调整的依据进行参数调整，重复上述过程直到差值很小为止，这时，保存下模型的参数，作为最终的模型参数值。

**应用过程：**建立跟训练过程一样的检测模型，读入保存下来的模型参数，准备一个未知数据**（未知数据需要和训练数据一样的格式）**，将未知数据输入模型，等待模型计算后输出结果，输出的结果就是这种未知图片的检测结果。

图像数据集的制作

在检测模型中，数据集是由一系列图片和标注框构成。

|  |  |
| --- | --- |
| 类别 | 标签名 |
| 猫 | cat |
| 狗 | dog |
| 人 | person |

数据集目录（VOC数据集格式）如下

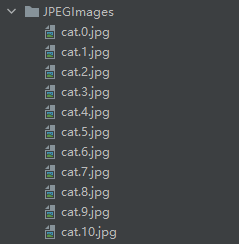
Annotations : 存放标注后生成的.xml文件

ImageSets\Main: 存放划分数据集的.txt文件

JPEGImages: 存放图片文件

dataSet.py: 用来划分数据集的代码

1. 确定检测类别，收集若干张图片，所有图片的物体加起来，每个类别的物体均不少于500个，放入JPEGImages文件夹，如下图所示：



⚠️图片的命名随意，用英文或拼音字母来表示，尽量不要使用中文汉字。

（实例的数据集是偷懒找了分类的数据集，图片名称就为类别名，但是不影响。）

⚠ 单张图片可以有多个类别的物体，并且同一类别的物体可以有多个。

⚠ 图片后缀名必须是.jpg（查看源码就知道）

1. 使用labelImage软件，对图片进行标注。
2. 下载labelImg软件

这里建议下载exe文件，python源代码用起来有点麻烦。

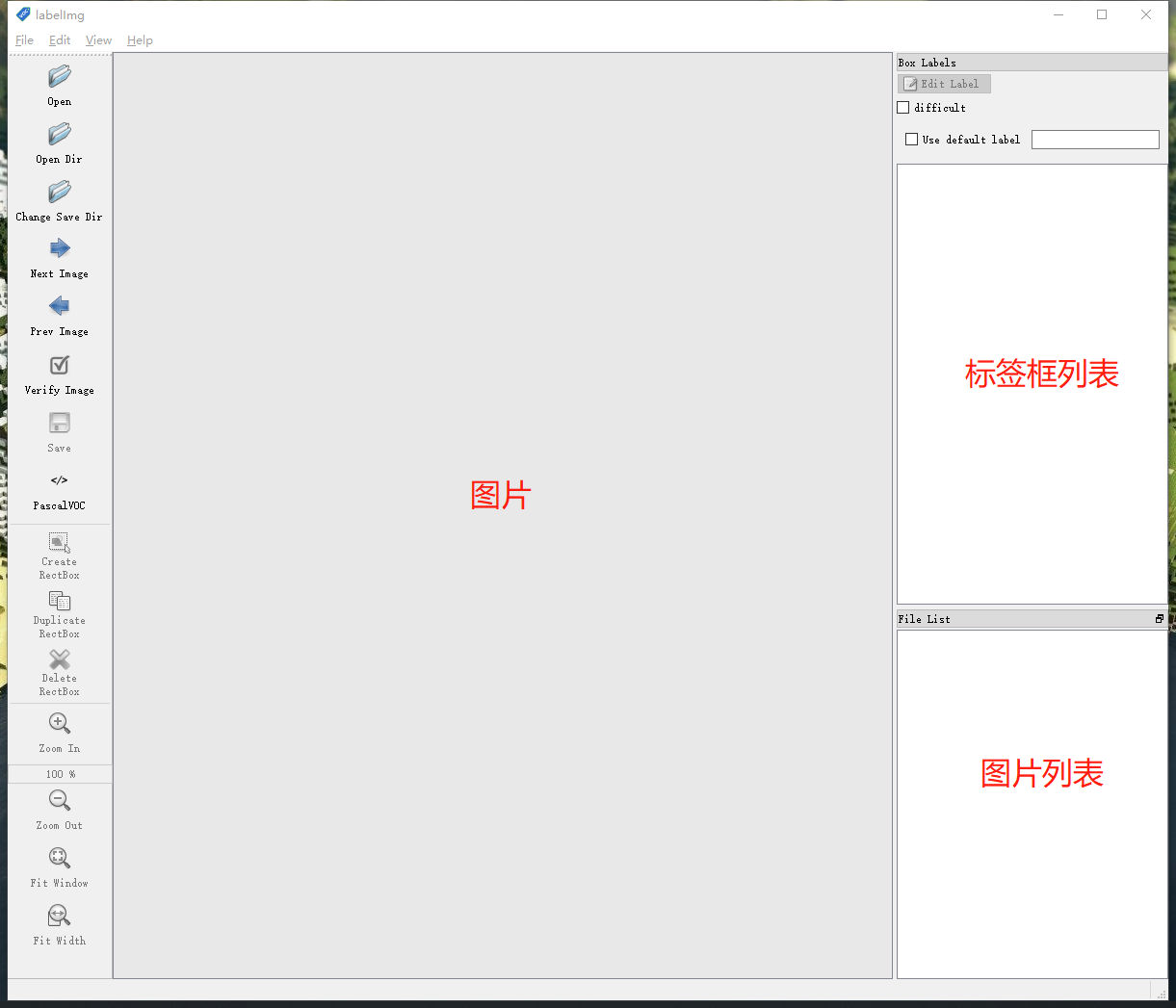
exe下载地址：

官方：<http://tzutalin.github.io/labelImg/> (需要翻墙)

百度云：链接：https://pan.baidu.com/s/1MtwKm2yuu2fQcKHYJkbyZQ

提取码：f4zs

1. 标注数据
2. labelImg主界面



1. 常用快捷键

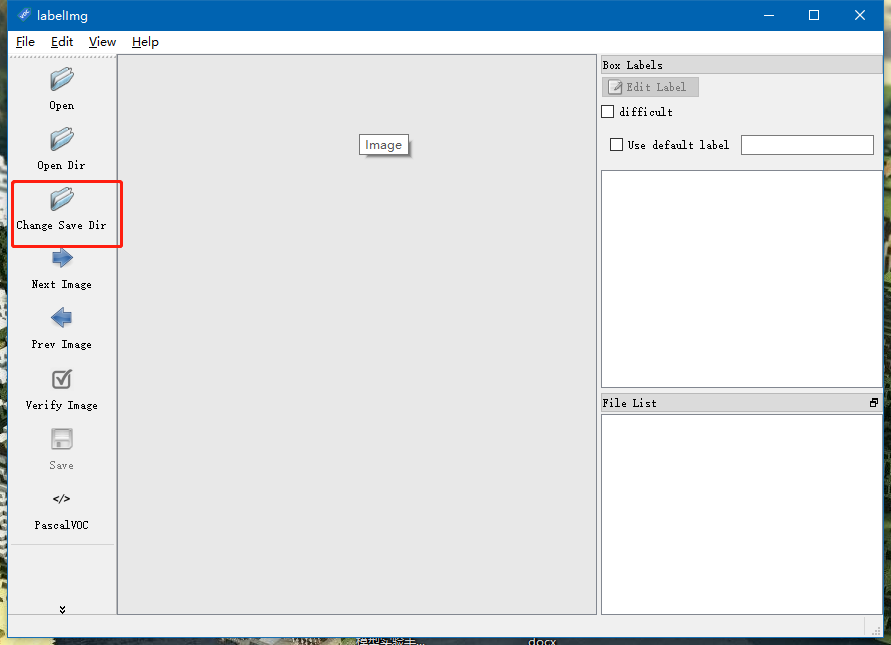
W : 创建标签框

A ：上一张图片

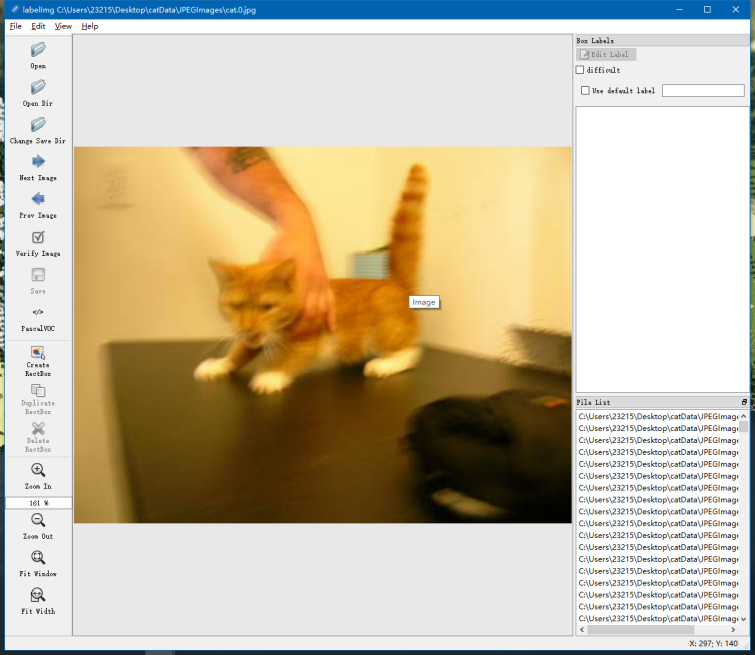
D ：下一张图片

Ctrl + S ： 保存.xml文件

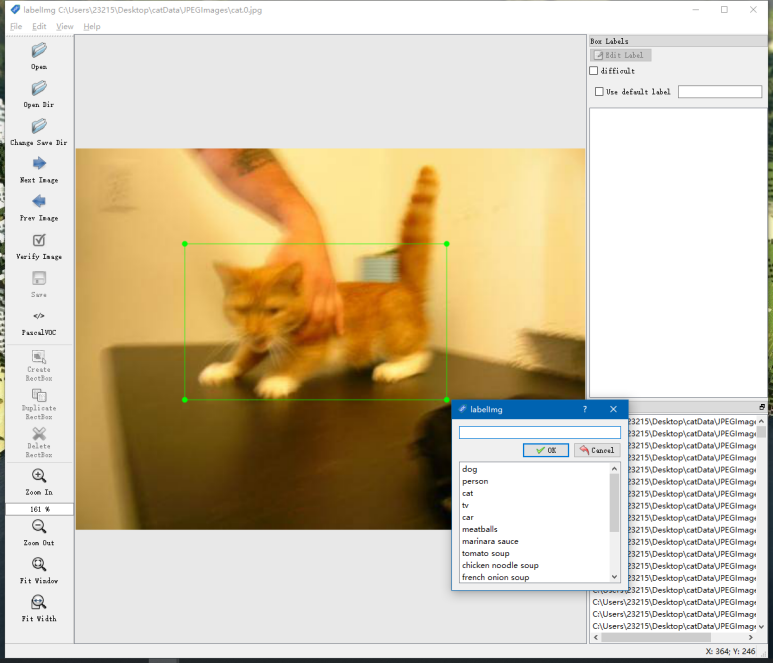
1. 标注步骤
2. 修改.xml文件的保存路径



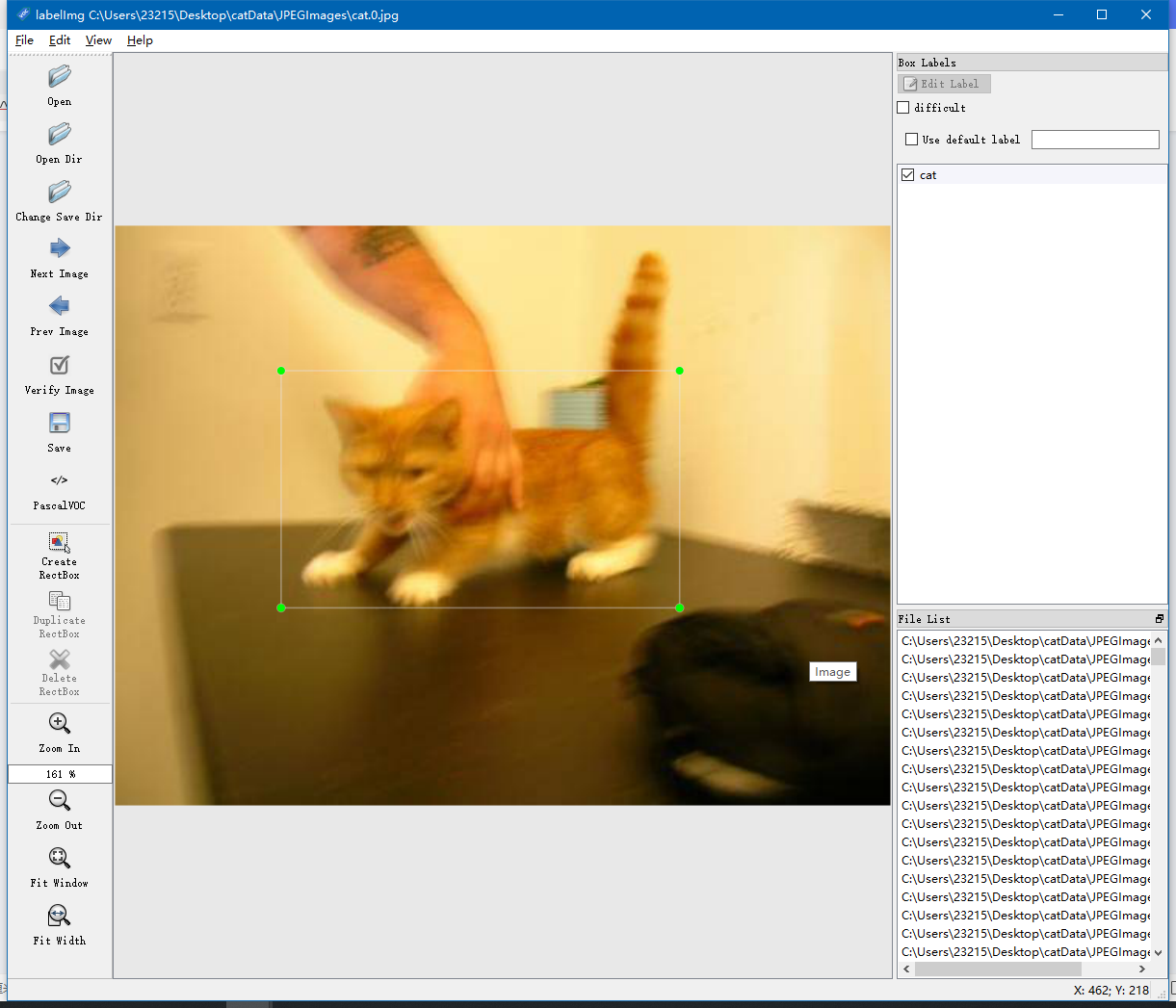
1. 打开文件夹



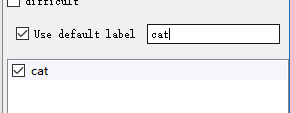
1. 创建标签框

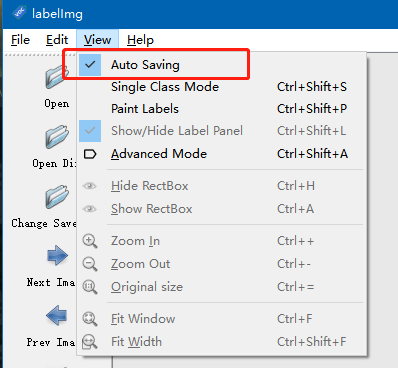


1. 输入类别名



1. 保存
2. 如果嫌每次都需要输入类别名很麻烦，可以使用默认标签和自动保存。



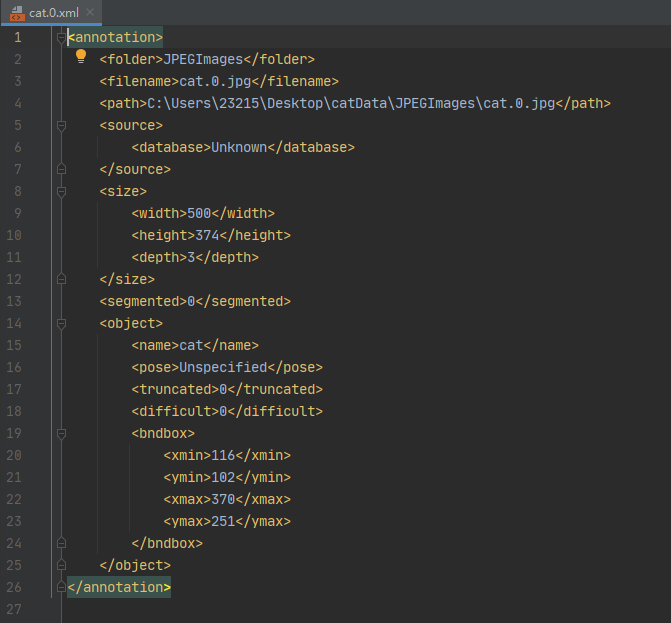


⚠️labelImage软件闪退：

方法1：将labelImage软件移到没有中文的路径下。

方法2：删除 C:\Users\用户名\labelImgSettings.pkl。

1. .xml文件内容



folder: 图片所在文件夹名称

filename: 图片名

path: 图片所在路径

size: 图片属性

Object: 类别（可以有多个。子标签bndbox为标签框，也可以有多个）

⚠ path标签是图片当时所在的绝对地址，但是我们不需要这个标签（源码会生成图片所在的地址并保存），所以我们存放图片的路径可以随意，不用担心因为该标签而在导入数据集时找不到图片。

划分数据集

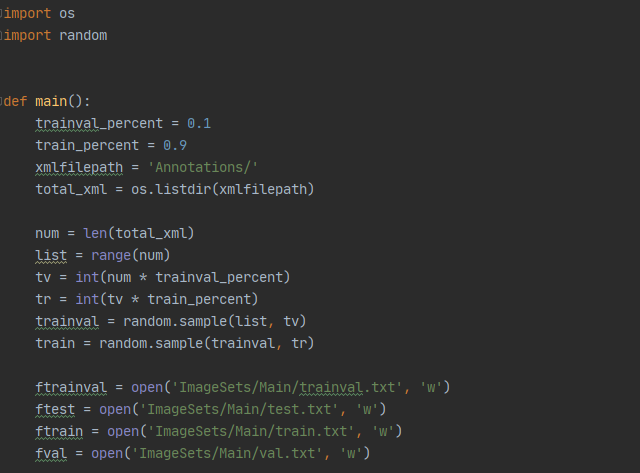
收集并标注完数据集后，需要使用代码读取图像形成数据集，一般数据集分为四个部分：

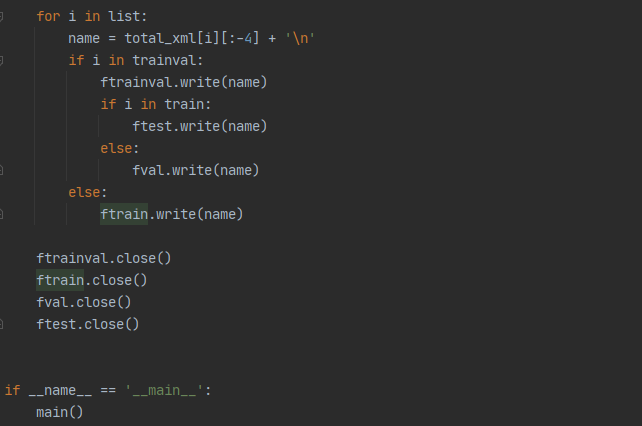
* 训练图像
* 训练图像标签
* 测试图像
* 测试图像分类标签

从收集的图片库中进行读取，其中随机选取90%的图片作为训练图像，剩下的10%作为测试图像。

数据集的入口就是ImageSets\Main文件夹下的.txt文件。通过读取.txt文件获取图片名，将数据集划分为测试集、训练集、训练验证集、验证集。（虽然这么分，但实际上我们使用的只有训练集，源码里没有测试环节，所以测试集没有用到），再利用获取到的图片名，从指定文件夹读取图片和标签信息，输入到模型中。

在数据集的根目录创建dataSet.py文件，输入以下代码并运行。





代码获取及配置

1. 环境配置

Python 3.6

tensorflow-gpu 1.14.0

（CUDA和CUDNN自行配置）

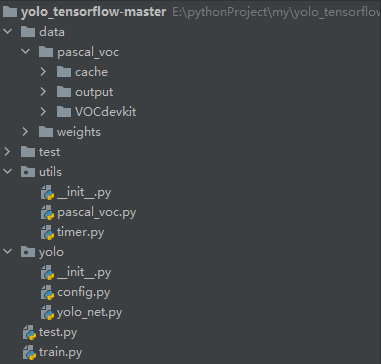
1. 代码获取

YOLO v1 tf版本的源码

1. Github: https://github.com/hizhangp/yolo\_tensorflow
2. 百度云：链接：https://pan.baidu.com/s/1dGAEkVel2H2P6e8E9NoZaA

提取码：tfwr

源码目录如下：



data: 存放数据(需要自己创建)，其子文件夹pascal\_voc存放数据集，子文件夹weights存放权 重。

test: 存放测试用的图片

utils: 与数据集有关的代码包

yolo: 与模型有关的代码包

test.py: 测试代码

train.py: 训练代码

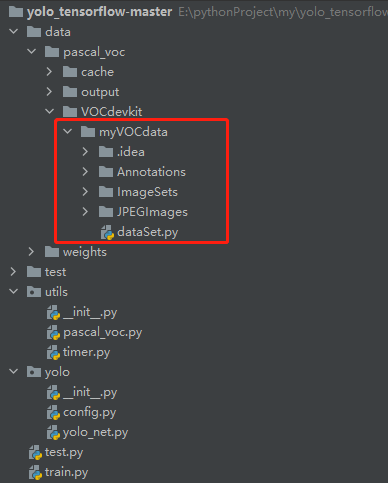
其中，

data\pascal\_voc\VOCdevkit文件夹、data\weights文件夹要自己创建。 data\pascal\_voc\cache文件夹在运行train.py后会生成，里面的pascal\_train\_gt\_labels.pkl文件的内容是图片地址与其处理后的标签（标签被转化为[中心横坐标, 中心纵坐标, 宽度, 高度, 置信度，类别1的概率，类别2的概率，...]）一一对应，如果存在，训练模型时直接加载，如果不存在，则重新生成写入。所以更换数据集记得把该文件夹删除或清空。

data\pascal\_voc\output文件夹在运行train.py后会生成，里面保存训练出来的权重文件。

1. 数据集安置

将数据集复制到data\pascal\_voc\VOCdevkit文件夹里，如下图：



1. 代码修改

由于该项目的源码是针对VOC2007数据集的，因此再训练自制的数据集之前，需要修改一些参数配置。

1. yolo\config.py文件



修改为：CLASSES = [‘类别名1’, ‘类别名2’, ‘...’]， 类别名的顺序不分先后

1. utils\pascal\_voc.py文件



修改为: self.data\_path = os.path.join(self.devkil\_path, ‘自制数据集根目录文件夹名称’)





这两处的6修改为：5+类别数（比如，如果有3个类别，那就是8）。

模型的训练设置与训练过程

1. 模型的训练设置

模型启动训练前，需要设置一些训练参数。主要是修改yolo\config.py文件的以下几个参数。

LEARNING\_RATE = 0.0001 # 学习率

DECAY\_STEPS = 30000 # 学习率的衰减步数（这里是官方源码的，我并未修改）

DECAY\_RATE = 0.1 # 学习率衰减率

BATCH\_SIZE = 8 # （显卡不行建议改小）

MAX\_ITER = 15000 # 最大迭代次数（类似于周期）

SUMMARY\_ITER = 10 # 日志记录迭代步数

SAVE\_ITER = 1000 # 权重保存步数

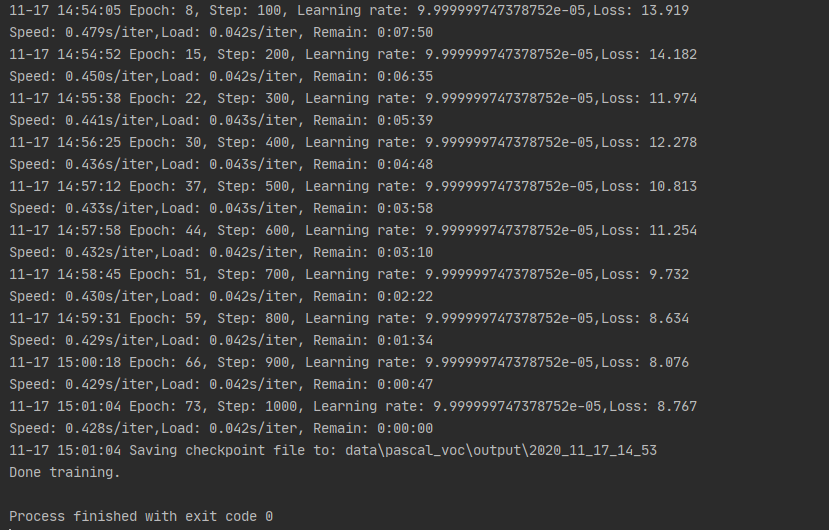
1. 模型的训练过程

调完参数直接运行train.py即可。

下图是训练过程（一些关于GPU的Warning未截出）





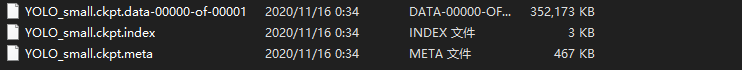


权重文件如下：

(这里训练了1000个迭代步数，所以权重文件只保存一次。三个为一组)



从训练得到的权重里挑一个最好的，复制到 yolo\_tensorflow-master\data\weights文件夹下，并改名为YOLO\_small.ckpt（这是之前训练好的权重）



⚠️官方权重（VOC2007数据集）

(1)下载地址：

链接：https://pan.baidu.com/s/10h0FMjXdqOJ4SjLzjgz9lg

提取码：wcr2

(2)类别名： ['aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus',

'car', 'cat', 'chair', 'cow', 'diningtable', 'dog', 'horse',

'motorbike', 'person', 'pottedplant', 'sheep', 'sofa',

'train', 'tvmonitor']

模型的测试

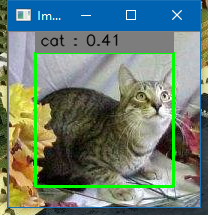
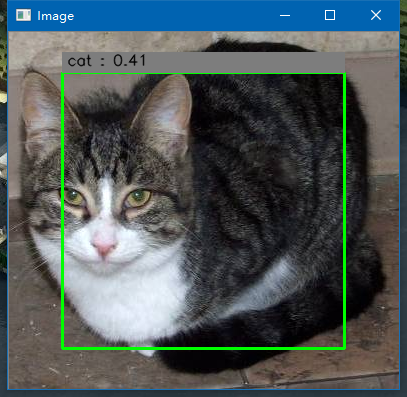
当模型训练并保存权重完后，就可以新建一个模型的时候直接调用已经训练好的参数直接进行预测。

1. 修改test.py文件



这里的test/test.jpg修改为要检测的图片

1. 运行test.py，多次结果如下图（截图大小为原图大小）：

（这里的数据集只有500多个cat类别标签框，训练了5000个迭代步数，选取最后一次的权重来检测）