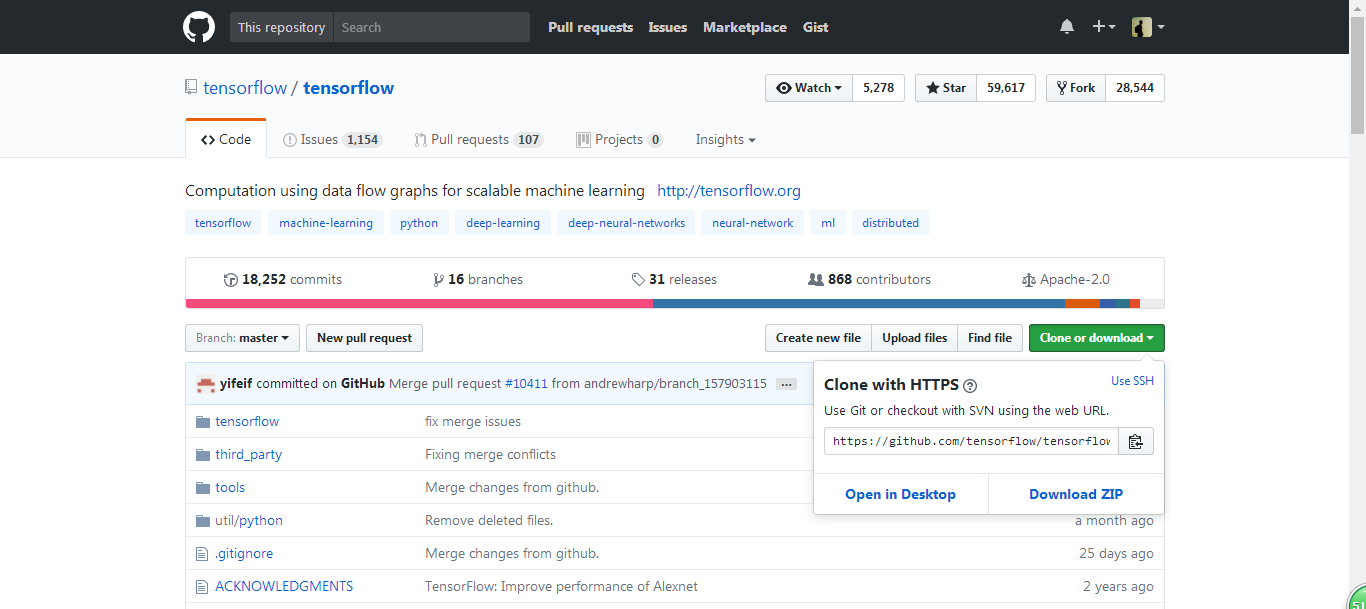
# inception v3模型的微调

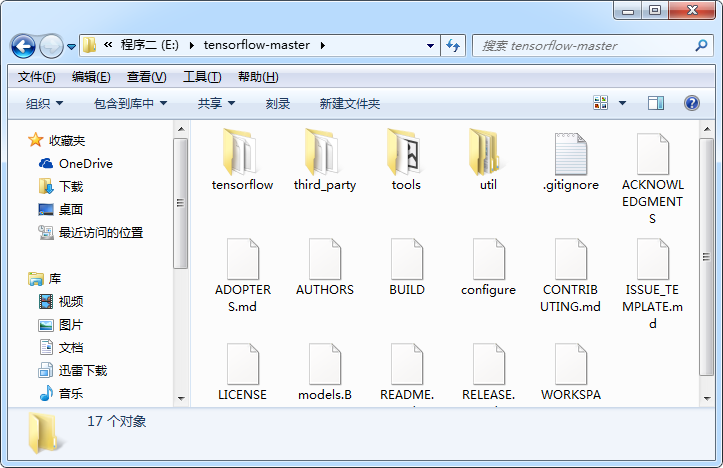
## 训练

## 下载tensorflow源码

github上下载tensorflow源码:https://github.com/tensorflow/tensorflow



解压得到E盘：



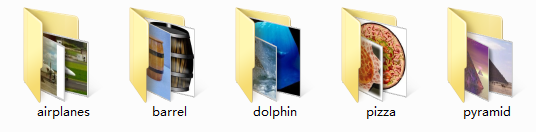
## 1.2下载inception v3模型

在E盘下新建一个名为retrain的文件夹，下载inception v3模型，放在E:\retrain文件夹下，下载地址：

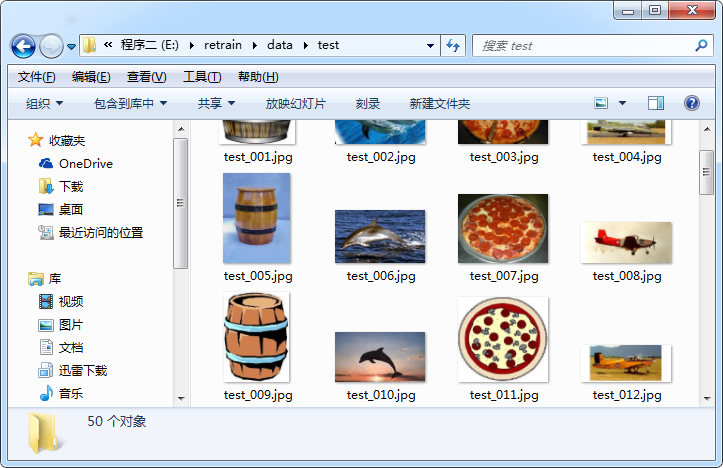
<http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz>

### 准备数据

准备5个类别数据，文件夹以类别名称命名，放在一个名为E:/retrain/data/train文件夹下：



准备测试数据，放在E:/retrain/data/test文件夹下：



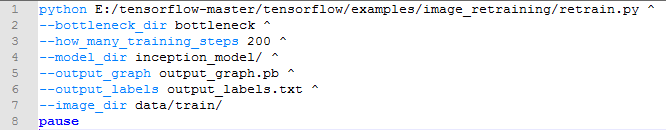
### Bottleneck文件夹

在retrain目录下新建一个bottleneck文件夹，该文件夹用于保存训练的中间结果

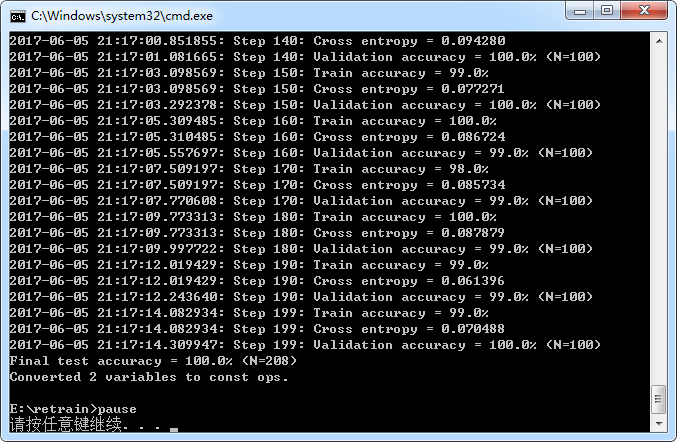
### 训练脚本

编写一个retrain.bat文件，放置在E：/retrain目录下，内容：

|  |
| --- |
| python E:/tensorflow-master/tensorflow/examples/image\_retraining/retrain.py ^  --bottleneck\_dir bottleneck ^  --how\_many\_training\_steps 200 ^  --model\_dir inception\_model/ ^  --output\_graph output\_graph.pb ^  --output\_labels output\_labels.txt ^  --image\_dir data/train/  pause |



执行脚本：



## 测试

准备以下程序：

|  |
| --- |
| # coding: utf-8  import tensorflow as tf  import os  import numpy as np  import re  from PIL import Image  import matplotlib.pyplot as plt  lines = tf.gfile.GFile('E:/retrain/output\_labels.txt').readlines()  uid\_to\_human = {}  #一行一行读取数据  for uid,line in enumerate(lines) :  #去掉换行符  line=line.strip('\n')  uid\_to\_human[uid] = line  def id\_to\_string(node\_id):  if node\_id not in uid\_to\_human:  return ''  return uid\_to\_human[node\_id]  #创建一个图来存放google训练好的模型  with tf.gfile.FastGFile('E:/retrain/output\_graph.pb', 'rb') as f:  graph\_def = tf.GraphDef()  graph\_def.ParseFromString(f.read())  tf.import\_graph\_def(graph\_def, name='')  with tf.Session() as sess:  softmax\_tensor = sess.graph.get\_tensor\_by\_name('final\_result:0')  #遍历目录  for root,dirs,files in os.walk('E:/retrain/data/test'):  for file in files:  #载入图片  image\_data=tf.gfile.FastGFile(os.path.join(root,file), 'rb').read()  predictions = sess.run(softmax\_tensor,{'DecodeJpeg/contents:0': image\_data})#图片格式是jpg格式  predictions = np.squeeze(predictions)#把结果转为1维数据  #打印图片路径及名称  image\_path = os.path.join(root,file)  print(image\_path)  #显示图片  img=Image.open(image\_path)  plt.imshow(img)  plt.axis('off')  plt.show()  #排序  top\_k = predictions.argsort()[::-1]  print(top\_k)  for node\_id in top\_k:  #获取分类名称  human\_string = id\_to\_string(node\_id)  #获取该分类的置信度  score = predictions[node\_id]  print('%s (score = %.5f)' % (human\_string, score))  print() |

执行结果：

