

CSDN

博客 (<http://blog.csdn.net?ref=toolbar>)

学院 (<http://edu.csdn.net?ref=toolbar>)

下载 (<http://download.csdn.net?ref=toolbar>)

GitChat (<http://gitbook.cn/?ref=csdn>) ...

1

Q

写博

发Chat

登录 (<https://passport.csdn.net/account/login?ref=toolbar>)

注册 (<https://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister>)

new

gitchat

http://write.blog.csdn.net/postedit?ref=activity&utm\_source=csdnblog1

## 【Tensorflow】怎样为你的网络预加工和打包训练数据？（一）

原创 2017年06月16日 13:35:00

标签: TensorFlow (<http://so.csdn.net/so/search/s.do?q=TensorFlow&t=blog>) /

深度学习 (<http://so.csdn.net/so/search/s.do?q=深度学习&t=blog>) /

python (<http://so.csdn.net/so/search/s.do?q=python&t=blog>) /

神经网络 (<http://so.csdn.net/so/search/s.do?q=神经网络&t=blog>) /

机器学习 (<http://so.csdn.net/so/search/s.do?q=机器学习&t=blog>)

1166

面对五花八门的数据集，各种各样的数据存储形式，刚新手入门的我们在处理这些情况的时候是否会手足无措？反正一路走来，我的经验告诉我，deep learning的实验阶段，数据准备和处理过程往往会让你碰一鼻子灰。明明知道如何搭建网络，还是完成不了实验，究其原因，是数据工程经验的不足。我打算做这个系列，主要是记录针对不同种类，格式的数据的处理方案。数据预处理的首篇，我为大家展示一种常见情形的处理方法

## 一.问题背景

问题的背景是面对raw image数据集，但是图片按label为文件夹存放。以Office-31数据集为例。Office-31数据集是一个用于迁移学习算法性能测试的数据集，我已经上传到网上，下载地址在下面：

domain\_adaptation\_images.part1.rar  
([http://download.csdn.net/detail/mao\\_xiao\\_feng/9872209](http://download.csdn.net/detail/mao_xiao_feng/9872209))

domain\_adaptation\_images.part2.rar  
([http://download.csdn.net/detail/mao\\_xiao\\_feng/9872216](http://download.csdn.net/detail/mao_xiao_feng/9872216))

权限不够，上传了两个分卷。  
解压完以后出现这个文件



下面又是三个文件，这三个就是不同环境下拍摄的图，我们只需要进amazon即可



加入是500个文件夹，每个文件夹包含各种各样的图片，与500个文件夹共同组成一个label。



xf\_mao ([http://blog.csdn.net/mao\\_xiao\\_feng](http://blog.csdn.net/mao_xiao_feng))

+关注

([http://blog.csdn.net/mao\\_xiao\\_feng](http://blog.csdn.net/mao_xiao_feng))

原创 40 粉丝 241

码云 未开通 喜欢 6 (https://gitee.com/mao\_xiao\_feng)

### 他的最新文章

更多文章 ([http://blog.csdn.net/mao\\_xiao\\_feng](http://blog.csdn.net/mao_xiao_feng))

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/78004522](http://blog.csdn.net/mao_xiao_feng/article/details/78004522))

【Tensorflow】tf.nn.atrous\_conv2d如何实现空洞卷积？ ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/78003730](http://blog.csdn.net/mao_xiao_feng/article/details/78003730))

【Tensorflow】tf.nn.depthwise\_conv2d如何实现深度卷积？ ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/78003476](http://blog.csdn.net/mao_xiao_feng/article/details/78003476))

【Tensorflow】tf.nn.separable\_conv2d如何实现深度可分卷积？ ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/78002811](http://blog.csdn.net/mao_xiao_feng/article/details/78002811))

【Tensorflow】tf.nn.depthwise\_conv2d如何实现深度卷积？ ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/77938385](http://blog.csdn.net/mao_xiao_feng/article/details/77938385))



成功挽回前男友

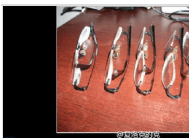


铁屑压块机



广告

美国免费空间



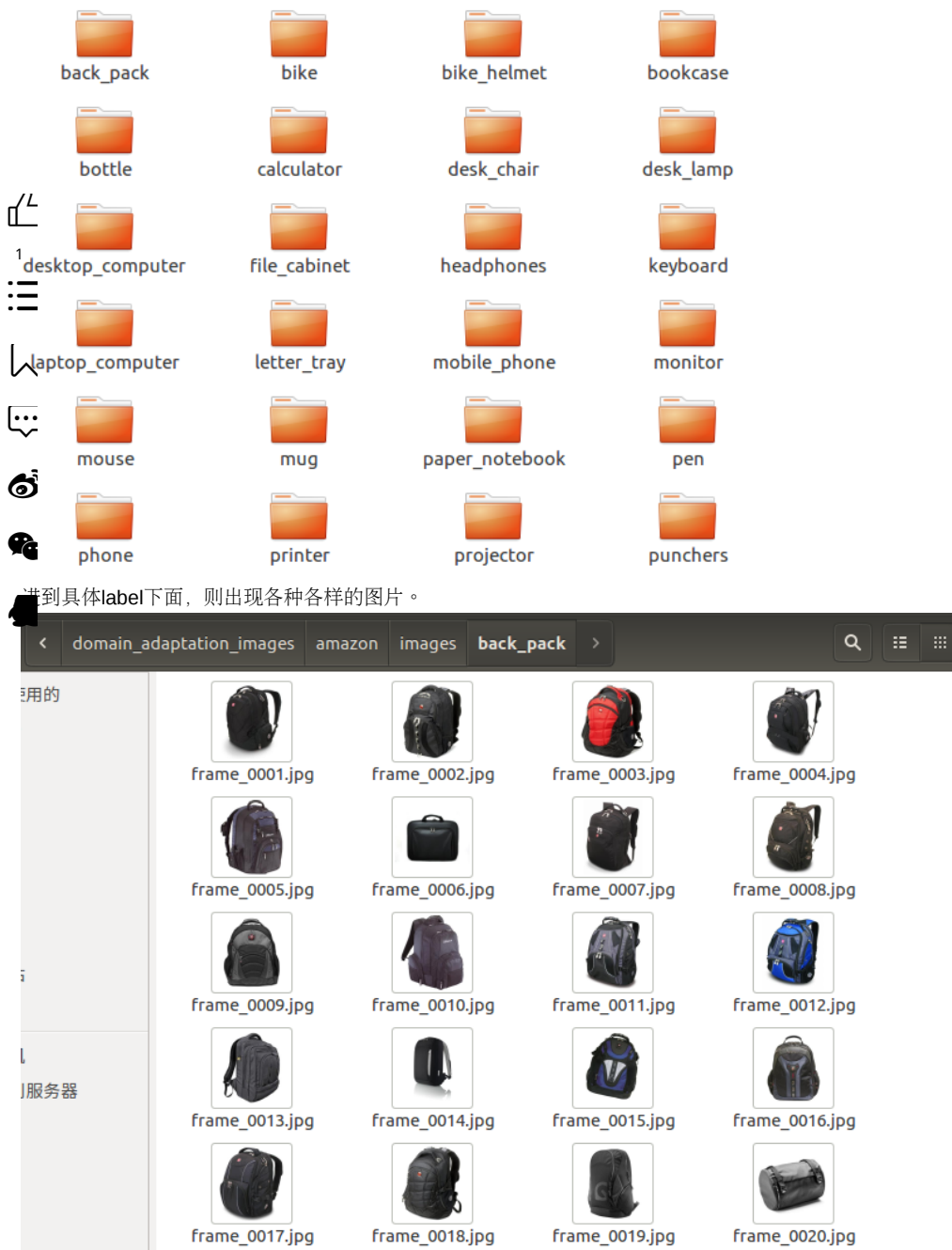
近视从600度降到

### 博主专栏

登录

注册

×



进入到具体label下面，则出现各种各样的图片。

描述这样一个问题背景是有意义的，因为实际上很多图片数据集都是以这样的形式来存放。以此为范例，下面来记录一个这个问题的具体解决方案。

## 二.解决方案

首先说一下需要用到辅助工具，前一篇讲到的skimage（【Tensorflow】辅助工具篇——scikit-image介绍 ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/73251440](http://blog.csdn.net/mao_xiao_feng/article/details/73251440))），cPickle，matplotlib  
鉴于这里有三个domain的数据，我们只做amazon这个文件夹下图片的处理  
先上代码。

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！  
1. `def build_dataset(data_dir, out_dir, weight=100, height=100):`



如何为Tensorflow训练打包数据和预处理？  
([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/16035.html](http://blog.csdn.net/mao_xiao_feng/article/details/16035.html))  
7429

### 他的热门文章

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/53444333](http://blog.csdn.net/mao_xiao_feng/article/details/53444333))  
33270

【TensorFlow】tf.nn.softmax\_cross\_entropy\_with\_logits的用法 ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/53382790](http://blog.csdn.net/mao_xiao_feng/article/details/53382790))  
25670

【TensorFlow】tf.nn.max\_pool实现池化操作 ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/53453926](http://blog.csdn.net/mao_xiao_feng/article/details/53453926))  
19299

【TensorFlow】tf.concat的用法 ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/53366163](http://blog.csdn.net/mao_xiao_feng/article/details/53366163))  
16640

对于随机森林的通俗理解 ([http://blog.csdn.net/mao\\_xiao\\_feng/article/details/52728164](http://blog.csdn.net/mao_xiao_feng/article/details/52728164))  
14096



小程序代理商

it培训机构排名

激光除锈机

大学生云服务器

```

2.     data_dir = os.path.join(data_dir, "images")
3.     for _, dirnames, _ in os.walk(data_dir):
4.         for dirname in dirnames:
5.             index = dirnames.index(dirname)
6.             workdir = os.path.join(data_dir, dirname)
7.             #images = io.imread_collection(workdir + '/*.jpg')
8.             processed_images = io.ImageCollection(workdir + '/*.jpg', load_func=process_image,
9.             label = np.full(len(processed_images), fill_value=index, dtype=np.int32)
10.            images = io.concatenate_images(processed_images)
11.            if index == 0:
12.                data = images
13.                labels = label
14.            else:
15.                data = np.vstack((data, images))
16.                labels = np.append(labels, label)
17.        if not os.path.exists(out_dir):
18.            os.makedirs(out_dir)
19.        print "data shape:", data.shape
20.        print "label shape:", labels.shape
21.        save_pickle(data, out_dir+'/'+ 'amazon_images.pkl')
22.        save_pickle(labels, out_dir+'/'+ 'amazon_labels.pkl')

```

解决思路还是比较传统的。首先要遍历文件夹，对于每一个文件夹下面的所有图片，用skimage批量读出来读取的过程是通过imread\_collection函数将所有jpg图片读取出来，返回一个类（注意此时这个类并不是np数组，而是skimage中的ImageCollection类，所以他并不能直接使用，我们要通过concatenate\_images函数将多个图片连接起来成为一个np数组）

但是我们没有使用imread\_collection函数，而是使用了ImageCollection类的构造函数，直接构造一个ImageCollection类，主要是因为如果图像的大小像素不同会导致连接的时候报错（维度不同），所以我们要先完成图像的预处理，处理完了将所有的图resize到相同的大小。构造ImageCollection类的时候可以load进去一个处理函数，在这里是process\_image函数：

```

[python]
1. def process_image(image, weight, height):
2.     img = io.imread(image)
3.     img = transform.resize(img, (weight, height), mode='reflect')
4.     return img

```

当然process\_image函数里面我们还可以添加其他内容（裁剪，填充等）

另外，如果是可以保证原始图像的像素全部相等，那么我们也可以imread\_collection读进来以后统一处理。这里我们主要针对的是更复杂的情况。

最后，使用pkl文件来保存。

```

[python]
1. def save_pickle(data, path):
2.     with open(path, 'wb') as f:
3.         pickle.dump(data, f, pickle.HIGHEST_PROTOCOL)
4.         print ('Saved %s..' %path)

```

在数据量不大的情况下，pkl是一种常用的保存手段，同时使用gzip来压缩，（我这里为了方便没有用），最常见的mnist就是用的pkl.gz这种后缀。同时cPickle又是pickle的升级版，压缩率好过pickle，大家可以尝试一下。

但是在数据量很大的情况下，我们一般使用hdf5，hdf5在性能方面是好过cPickle很多。这种方法后面会介绍。

当然也可以构造图片预处理的pipeline。这种方法是所有方案的终极版，专门针对超大数据集

（ImageNet，CoCo）不可能全部load到内存中使用的，例如用CoCo数据集来做style transfer训练的时候用的就是线程读图片的方式，同时这种方法也是最难去实现的，同样后面也会介绍。

大功告成了！最后看一看结果吧。

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

同样还是用matplotlib来显示多个图片

登录

注册

×

```
[python]
1. import cPickle
2. from mpl_toolkits.axes_grid1 import ImageGrid
3. import matplotlib.pyplot as plt
4.
5.
6. def imshow_grid(images, shape=[2, 8]):
7.     """Plot images in a grid of a given shape."""
8.     fig = plt.figure(1)
9.     grid = ImageGrid(fig, 111, nrows_ncols=shape, axes_pad=0.05)
10.
11.     size = shape[0] * shape[1]
12.     for i in range(size):
13.         grid[i].axis('off')
14.         grid[i].imshow(images[i]) # The AxesGrid object work as a list of axes.
15.
16.     plt.show()
17.
18.
19. def load_amazon():
20.     data = cPickle.load(open('processed_data/amazon_images.pkl'))
21.     labels = cPickle.load(open('processed_data/amazon_labels.pkl'))
22.     return data, labels
23.
24. data, labels = load_amazon()
25. print "show image..."
26. imshow_grid(data[90:106])
27. print labels[90:106]
```

Figure 1



图片和label，看到是可以对上了，然后我们就可以下一步了。

```
[python]
1. [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1]
```

不要忘了数据的归一化！！比较简单的做法是计算（像素值-127.5）/127.5，这种做法是归一到-1到1之间，也可以算每个通道的均值，然后每个通道分别归一。  
归一化相信大家都会，就不赘述了。

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



### 三.实验源码

[python]

```
1. import tensorflow as tf
2. import tensorflow.contrib.slim as slim
3. import os
4. from skimage import io,transform
5. from mpl_toolkits.axes_grid1 import ImageGrid
6. import argparse
7. import numpy as np
8. import cPickle
9. import matplotlib.pyplot as plt
10.
11. parser = argparse.ArgumentParser(description='')
12. parser.add_argument('--dataset', dest='dataset', default='amazon', help='dataset name')
13.
14. def save_pickle(data, path):
15.     with open(path, 'wb') as f:
16.         cPickle.dump(data, f, cPickle.HIGHEST_PROTOCOL)
17.         print ('Saved %s..' %path)
18.
19. def imshow_grid(images, shape=[2, 8]):
20.     """Plot images in a grid of a given shape."""
21.     fig = plt.figure(1)
22.     grid = ImageGrid(fig, 111, nrows_ncols=shape, axes_pad=0.05)
23.
24.     size = shape[0] * shape[1]
25.     for i in range(size):
26.         grid[i].axis('off')
27.         grid[i].imshow(images[i]) # The AxesGrid object work as a list of axes.
28.
29.     plt.show()
30.
31. def process_image(image, weight, hight):
32.     img = io.imread(image)
33.     img = transform.resize(img, (weight,hight), mode='reflect')
34.     return img
35.
36. def build_dataset(data_dir, out_dir, name,weight=100,hight=100):
37.     data_dir = os.path.join(data_dir,"images")
38.     for _, dirnames, _ in os.walk(data_dir):
39.         for dirname in dirnames:
40.             index = dirnames.index(dirname)
41.             workdir = os.path.join(data_dir, dirname)
42.             #images = io.imread_collection(workdir + '/*.jpg')
43.             processed_images = io.ImageCollection(workdir + '/*.jpg', load_func=process_image,
44.             label = np.full(len(processed_images), fill_value=index, dtype=np.int32)
45.             images = io.concatenate_images(processed_images)
46.             if index == 0:
47.                 data = images
48.                 labels = label
49.             else:
50.                 data = np.vstack((data,images))
51.                 labels = np.append(labels,label)
52.
53.             if not os.path.exists(out_dir):
54.                 os.makedirs(out_dir)
55.             print("data shape:")
56.             print(data.shape)
57.             print("label shape:")
58.             print(labels.shape)
59.             save_pickle(data, out_dir+'/'+name+'_images.pkl')
60.             save_pickle(labels, out_dir+'/'+name+'_labels.pkl')
61.
62. def load_amazon():
63.     images = cPickle.load(open('prosessed_data/amazon/amazon_images.pkl'))
64.     labels = cPickle.load(open('prosessed_data/amazon/amazon_labels.pkl'))
65.     images = images*2 - 1
66.     print ('finished loading amazon image dataset..!')
67.     return images,labels
68.
69. images = cPickle.load(open('prosessed_data/dslr/dslr_images.pkl'))
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×


```
71.     labels = cPickle.load(open('processed_data/dslr/dslr_labels.pkl'))
72.     images = images * 2 - 1
73.     print ('finished loading dslr image dataset...!')
74.     return images,labels
75.
76. def load_webcam():
77.     images = cPickle.load(open('processed_data/webcam/webcam_images.pkl'))
78.     labels = cPickle.load(open('processed_data/webcam/webcam_labels.pkl'))
79.     images = images * 2 - 1
80.     print ('finished loading webcam image dataset...!')
81.     return images, labels
82.
83. args = parser.parse_args()
84.
85. def main():
86.     print "make dataset..."
87.
88.     if args.dataset == 'amazon':
89.         build_dataset("domain_adaptation_images/amazon", "processed_data
/amazon",args.dataset,weight=64,height=64)
90.         print "read dataset..."
91.         images,label = load_amazon()
92.         print "show image..."
93.         imshow_grid((images[90:106]+1)/2)
94.         print label[90:106]
95.
96.     elif args.dataset == 'dslr':
97.         build_dataset("domain_adaptation_images/dslr", "processed_data
/dslr",args.dataset, weight=64, height=64)
98.         print "read dataset..."
99.         images, label =load_dslr()
100.        print "show image..."
101.        imshow_grid((images[90:106]+1)/2)
102.        print label[90:106]
103.
104.    elif args.dataset == 'webcam':
105.        build_dataset("domain_adaptation_images/webcam", "processed_data
/webcam",args.dataset, weight=64, height=64)
106.        print "read dataset..."
107.        images, label =load_webcam()
108.        print "show image..."
109.        imshow_grid((images[90:106]+1)/2)
110.        print label[90:106]
111.
112.    else:
113.        raise Exception("wrong args!!")
114.
115.    print "loading successful!"
116.
117.
118. if __name__ == "__main__":
119.     main()
```

版权声明：本文为博主原创文章，转载请注明出处。

本文已收录于以下专栏：如何为Tensorflow训练打包数据和预处理？(<http://blog.csdn.net/column/details/16035.html>)




楼主 load之后报错: &#39;gbk&#39; codec can&#39;t decode byte 0x80 in position 0: illegal multibyte sequence  
请问是什么原因呢?  
应该怎么修改呢?  
谢谢  
回复

 qq\_37226149 (/qq\_37226149) 2017-06-25 17:45 1楼  
(qq\_37226149)下这个的源码吗?  
1 回复 1条回复




## 【Tensorflow】怎样为你的网络预加工和打包训练数据? (二): 小数据集的处理方...

实验环境: python2.7 第二篇我们来讲一讲小数据集的处理方法, 小数据集一般多以文本存储为主, csv是一种流行的数据格式, 另外也有txt等。当然也会有.mat或者.npy这种经过处理的格式。一...

 mao\_xiao\_feng ([http://blog.csdn.net/mao\\_xiao\\_feng](http://blog.csdn.net/mao_xiao_feng)) 2017年06月25日 14:53 1025

## 基于Tensorflow, OpenCV. 使用MNIST数据集训练卷积神经网络模型, 用于手写数字识...

基于Tensorflow, OpenCV使用MNIST数据集训练卷积神经网络模型, 用于手写数字识别ocr.py一个单层的神经网络, 使用MNIST训练, 识别准确率较低cnn\_ocr.py两层的卷积神经网络...

 SugarAnnie (<http://blog.csdn.net/SugarAnnie>) 2017年10月30日 11:23 163

## 《神秘的程序员们》 【揭秘】程序员升职加薪的捷径来了!



在岗5年, 总想着闲下来的时候应该如何安排自己的程序人生呢? 无意中看到这个! 眼睛亮了..

([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjT3P160IZ0qnfK9ujYzP1nsrjDz0Aw-5Hc3rHnYnHb0TAq15HfLPWRznb0T1dBPh7hmWtkPWlBnjNhnAcY0AwY5HDdnHf4Pj6sPHm0lgF\\_5y9YIZ0IQzqBTLn8mLPbUB48ugfEUiqYULKGmzq-uZNxug99UHqdAdxTvqdThP-5yF\\_UvTkn0KzujY4rHb0mhYqn0KsTWYs0ZNGujYkPHTYn1mk0AQGuYknWb3rjDY0APGuYLnWm4n1c0ULI85H00TZbqnW0v0APzm1Yzrm4nf](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjT3P160IZ0qnfK9ujYzP1nsrjDz0Aw-5Hc3rHnYnHb0TAq15HfLPWRznb0T1dBPh7hmWtkPWlBnjNhnAcY0AwY5HDdnHf4Pj6sPHm0lgF_5y9YIZ0IQzqBTLn8mLPbUB48ugfEUiqYULKGmzq-uZNxug99UHqdAdxTvqdThP-5yF_UvTkn0KzujY4rHb0mhYqn0KsTWYs0ZNGujYkPHTYn1mk0AQGuYknWb3rjDY0APGuYLnWm4n1c0ULI85H00TZbqnW0v0APzm1Yzrm4nf)))


## Tensorflow框架下Faster-RCNN实践 (二) ——用自己制作的数据训练Faster-RCNN网...

修改make.sh #!/usr/bin/env bash TF\_INC=\$(python -c 'import tensorflow as tf; print(tf.sysconfig.get\_i...

 ll666634 (<http://blog.csdn.net/ll666634>) 2017年12月25日 21:40 31


## 用tensorflow实现VGG网络, 训练mnist数据集 (<http://blog.csdn.net/masterjames/article...>

本文主要用TENSORFLOW通过使用VGG网络模型训练mnist数据集。

 masterjames (<http://blog.csdn.net/masterjames>) 2017年11月13日 21:04 85

## TensorFlow训练mnist数据集 (卷积神经网络lenet5) (<http://blog.csdn.net/u013841196/...>

TensorFlow使用经典卷积神经网络实现对手写数字mnist的训练和测试两个卷积, 两个池化, 两个全连接层。代码如下:  
(在上一篇的基础上更改网络lenet5) 代码分为3部分, 网络-训练和...

 u013841196 (<http://blog.csdn.net/u013841196>) 2017年07月30日 23:53 251



7



0.75/只  
专业厂家生产汽车LED  
大灯用H7电子线 2芯 3

8



**10w大功率**  
摩托罗拉对讲机  
85.00/台  
欣阳对讲机 户外无线  
大功率酒店车载 手持

9



2200.00/只  
原装正品摩托罗拉防爆  
对讲机GP328全国联

## Tensorflow框架下Faster-RCNN实践（二）——用自己制作的数据训练Faster-RCNN网...

Tensorflow框架下Faster RCNN实践

Best\_Coder ([http://blog.csdn.net/Best\\_Coder](http://blog.csdn.net/Best_Coder)) 2017年08月03日 11:12 4096

## 利用tensorflow训练自己的图片数据（3）——建立网络模型 (<http://blog.csdn.net/ywx1832990>...

说明在上一博客——利用tensorflow训练自己的图片数据（2）中，我们已经获得了神经网络的训练输入数据：image\_batch, label\_batch。接下就是建立神经网络模型，笔者的网络...

ywx1832990 (<http://blog.csdn.net/ywx1832990>) 2017年11月23日 10:23 155

## 利用tensorflow训练自己的图片数据（5）——测试训练网络 (<http://blog.csdn.net/ywx1832990>...

说明上一篇博客中，我们已经将建立好的网络模型训练好了，并将训练的网络参数保存在相应的文件中；下面我们就开始测试网络，验证网络的训练效果；本次测试为随机的单图片测试，即随机的从训练集或测试集中读取...

ywx1832990 (<http://blog.csdn.net/ywx1832990>) 2017年11月24日 16:57 186

## tensorflow实现训练代码（mnist数据集&断点续训） (<http://download.csdn.net/download/baimafujinji/10155111>...

 <http://download.csdn.net/download/baimafujinji/10155111> 2017年11月04日 12:14 66.86MB 下载()

## tensorflow实现AlexNet训练mnist数据 (<http://download.csdn.net/download/baimafujinji/10155111>...

 <http://download.csdn.net/download/baimafujinji/10155111> 2017年05月07日 11:52 5KB 下载()

## 在TensorFlow中保存已经训练好的神经网络模型 (<http://blog.csdn.net/baimafujinji/article/details/505806>...

通常训练一个具有一定实用价值的深度神经网络都是非常消耗计算时间的。所以在使用时，最好的方法是导入已经训练好的模型，重用它，而不是每次都重新训练。本文就主要介绍在TensorFlow中如何保存和导入已经...

baimafujinji (<http://blog.csdn.net/baimafujinji>) 2016年01月27日 14:08 5806

## tensorflow 训练 MNIST数据集 (<http://download.csdn.net/download/u01551111/10155111>...

 <http://download.csdn.net/download/u01551111/10155111> 2017年08月15日 23:22 11.06MB 下载()

## Tensorflow自我训练进阶（代码+注解）【2】第一个神经网络--简单感知器处理MNIST ...

继续我们的tf~~~

Skyfish2016 (<http://blog.csdn.net/Skyfish2016>) 2017年05月15日 00:24 338

## 深入浅出Tensorflow（三）：训练神经网络模型的常用方法 ([http://blog.csdn.net/fu\\_shuwu](http://blog.csdn.net/fu_shuwu)...

[http://www.infoq.com/cn/articles/introduction-of-tensorflow-part03?utm\\_campaign=rightbar\\_v2&utm\\_sour...](http://www.infoq.com/cn/articles/introduction-of-tensorflow-part03?utm_campaign=rightbar_v2&utm_sour...)

fu\_shuwu ([http://blog.csdn.net/fu\\_shuwu](http://blog.csdn.net/fu_shuwu)) 2017年07月26日 21:32 1151  
加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×



## tensorflow实现最基本的神经网络 + 对比GD、SGD、batch-GD的训练方法 (<http://blog....>)

```
# -*- coding:utf-8 -*- # 将tensorflow引入并命名tf import tensorflow as tf # 矩阵操作库numpy, 命名为np impo...
```



Barry\_J ([http://blog.csdn.net/Barry\\_J](http://blog.csdn.net/Barry_J)) 2017年12月08日 10:48 54



## tensorflow的基本用法(五)——创建神经网络并训练 (<http://blog.csdn.net/Quincuntial/art...>)

1

tensorflow的基本用法(五)——创建神经网络并训练



Quincuntial (<http://blog.csdn.net/Quincuntial>) 2017年04月12日 23:11 1497



## tensorflow的基本用法(七)——使用MNIST训练神经网络 (<http://blog.csdn.net/Quincunti...>)

tensorflow的基本用法(七)——使用MNIST训练神经网络



Quincuntial (<http://blog.csdn.net/Quincuntial>) 2017年04月16日 00:27 1233



## TensorFlow 入门之第一个神经网络训练 MNIST (<http://blog.csdn.net/u012373815/article...>)

1. 构建神经网络构建一个神经网络，用于学习神经网络的结构。本文构建的神经网络是典型的三层神经网络，输入层、隐藏层、输出层。输入层有一个输入参数也就是有一个神经元，隐藏层定义了10个神经元，输出层...



u012373815 (<http://blog.csdn.net/u012373815>) 2017年08月15日 23:25 525

## 深入浅出Tensorflow（三）：训练神经网络模型的常用方法 ([http://blog.csdn.net/qq\\_31...](http://blog.csdn.net/qq_31...))

本文将介绍优化训练神经网络模型的一些常用方法，并给出使用TensorFlow实现深度学习的最佳实践样例代码。为了更好的介绍优化神经网络训练过程，我们将首先介绍优化神经网络的算法——梯度下降算法。然后在...



qq\_31001583 ([http://blog.csdn.net/qq\\_31001583](http://blog.csdn.net/qq_31001583)) 2017年07月21日 09:14 1255

## (尤其是训练集验证集的生成)深度学习 tensorflow 实战(2) 实现简单神经网络以及随机...

在之前的实战(1)中，我们将数据清洗整理后，得到了'notMNIST.pickle'数据。本文将阐述利用tensorflow创建一个简单的神经网络以及随机梯度下降算法。[ja...



u010159842 (<http://blog.csdn.net/u010159842>) 2017年03月06日 11:15 2054