

# **Mask R-cnn 分割模型训练操作流程说明**

2018 年 6 月

目录

Mask R-cnn 分割模型训练操作流程说明 ..... 1

2018 年 6 月 ..... 1

1. 模型训练前的准备工作 ..... 3

    1.1. 准备训练数据 ..... 3

        1.1.1 准备训练环境..... 3

        1.1.2rename\_JPGE.py 使用 ..... 3

        1.1.3 以 sf6 表为训练数据 ..... 3

        1.1.4 创建训练集目录..... 5

        1.1.5 生成标签文件..... 6

        1.1.6 把 16 位 mask 灰度图转 8 位灰度图 ..... 7

    1.2. 训练模型 ..... 8

        1.2.1 修改训练模型的文件 ..... 8

    1.3 测试模型..... 9

        1.3.1 修改测试模型的文件 ..... 9

# 1. 模型训练前的准备工作

## 1.1. 准备训练数据

### 1.1.1 准备训练环境

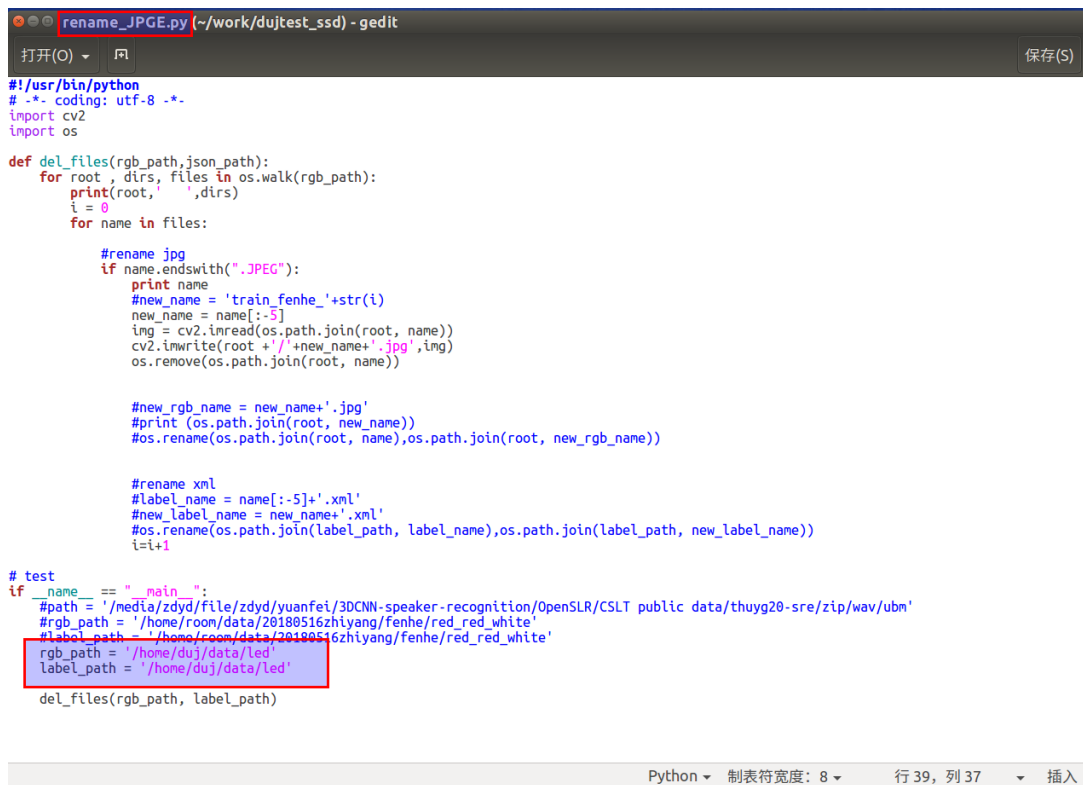
1. 首先确保自己的 ubuntu 系统中安装了 caffe。

### 1.1.2 rename\_JPGE.py 使用

此脚本为图片格式转换脚本。使用时需要修改的地方如下图所示：

✓ rgb\_path

✓ label\_path



```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import cv2
import os

def del_files(rgb_path,json_path):
    for root , dirs, files in os.walk(rgb_path):
        print(root,' ',dirs)
        i = 0
        for name in files:
            #rename jpg
            if name.endswith(".JPEG"):
                print name
                #new_name = 'train_fenhe'+str(i)
                new_name = name[:-5]
                img = cv2.imread(os.path.join(root, name))
                cv2.imwrite(root +'/' +new_name+'.jpg',img)
                os.remove(os.path.join(root, name))

                #new_rgb_name = new_name+'.jpg'
                #print (os.path.join(root, new_name))
                #os.rename(os.path.join(root, name),os.path.join(root, new_rgb_name))

                #rename xml
                #label_name = name[:-5]+'.xml'
                #new_label_name = new_name+'.xml'
                #os.rename(os.path.join(label_path, label_name),os.path.join(label_path, new_label_name))
                i=i+1

# test
if __name__ == "__main__":
    #path = '/media/zdyd/file/zdyd/yuanfei/3DCNN-speaker-recognition/OpenSLR/CSLT_public_data/thuyg20-sre/zip/wav/ubm'
    #rgb_path = '/home/room/data/20180516zhongyang/fenhe/red_red_white'
    #label_path = '/home/room/data/20180516zhongyang/fenhe/red_red_white'
    rgb_path = '/home/duj/data/led'
    label_path = '/home/duj/data/led'
    del_files(rgb_path, label_path)
```

### 1.1.3 以 sf6 表为训练数据

- 以 sf6 表为例进行，进行说明，sf6 表的原始图片如下：



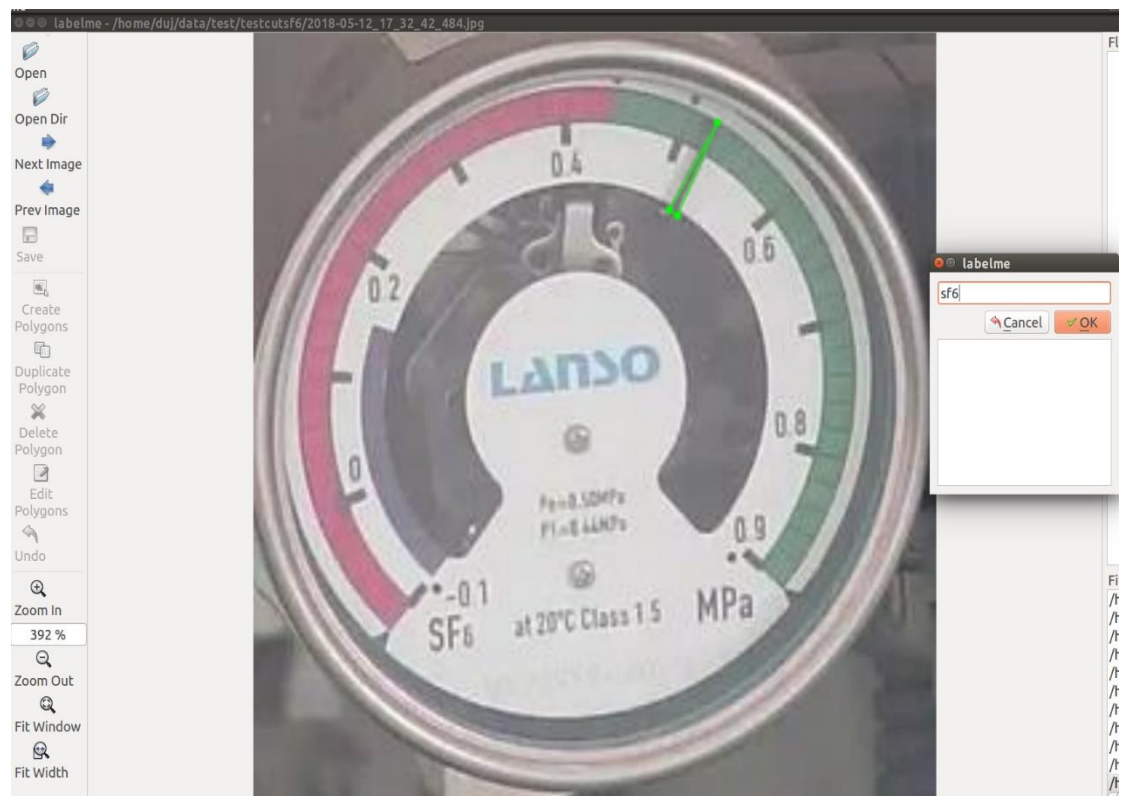
- 首先用 caffe 分类模型的数据先要通过 SSD 的面检测模型，把 sf6 表的面抠出来。SSD 扣出来的测试数据图片如下图：



图-1

- 使用 resize.py 把抠出来的图片变成统一的 256\*256 大小的。  
修改 resize.py 中的 data\_path 和 output\_path，这两个是 sf6 表的路径。我自己配置文件 data\_path 和 output\_path 都为：  
/home/duj/data/test/testcutsf6  
说明：如果不改成 256\*256 大小的尺寸，训练数据的时候会报错。
- 使用 labelme 标注工具对图-1 进行标注，标注显示的图如下：

标签为：sf6



■

## 1.1.4 创建训练集目录

### ■ 训练数据集目录

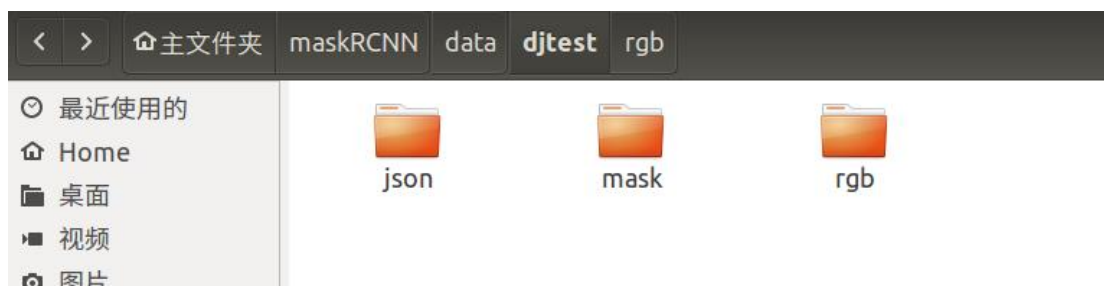
训练集文件 ----|-----json 存放.json 标签文件

|-----mask

|-----rgb 存放.jpg 图片

以上三个文件夹我本机放在/home/duj/maskRCNN/data/djtest

目录下。可以根据自己的习惯放。



- ✓ .jpg 图片数据统一命名为 rgb\_0.jpg, rgb\_1.jpg...格式
- ✓ 对应的.json 标签文件统一命名为 rgb\_0.json, rgb\_1.json...格式

说明：.jpg 图片数据的统一命名和对应的.json 标签文件的统一命名  
可以通过执行 rename\_rgb.py 脚本来实现。

此脚本中需要修改 rgb\_path 和 json\_path ( 即 rgb 图片和 json 文件的存放路径 )

### 1.1.5 生成标签文件

先给 json2label.sh 文件赋权限

```
chmod -R 777 json2label.sh
```

运行./json2label.sh ( 也可以 bash json2label.sh )

- 根据自己机器的训练数据修改训练集路径及 i 值，详细说明如下：

```
#!/bin/bash
```

自己本机训练数据的路径

```
s1="/home/duj/maskRCNN/data/djtest/json/rgb_"
```

```
s2=".json"
```

```
for((i=0;i<54;i++))
```

```
do
```

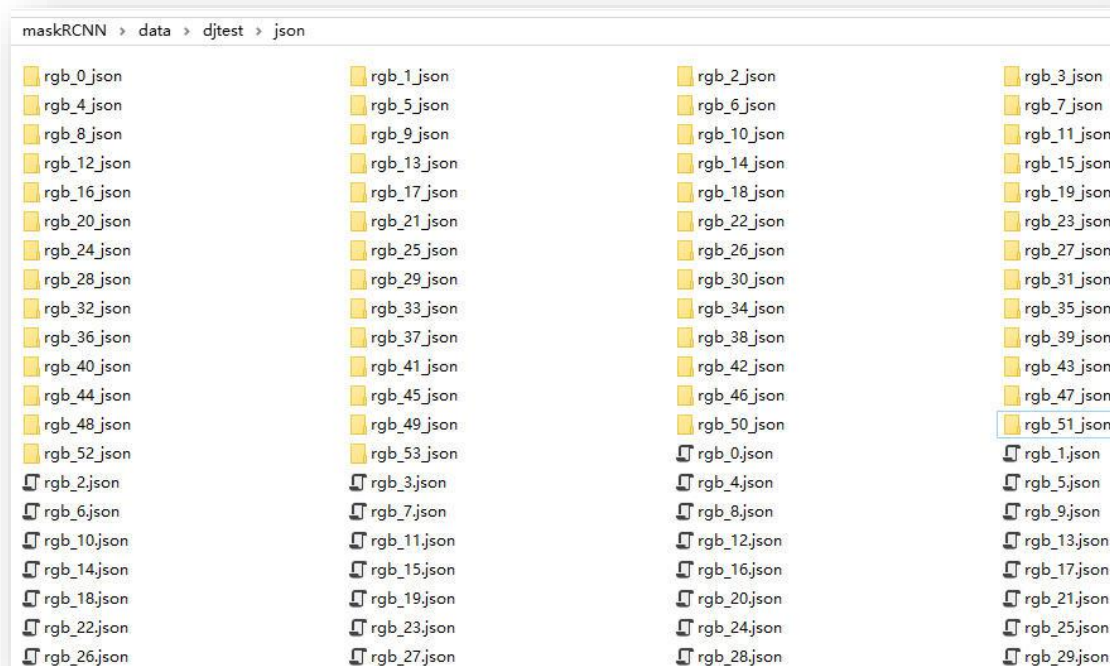
```
s3=${i}
```

训练图片数+1，我自己的训练图片是 53 张，所以我这配置文件的 i 值是 54

```
labelme_json_to_dataset ${s1}${s3}${s2}
```

done

json2label.sh，运行后将会在数据集文件夹下的 json 文件夹中生成标签，一个 json 文件对应一个标签文件夹。如下图所示：



### 1.1.6 把 16 位 mask 灰度图转 8 位灰度图

- 修改 mask\_label.cpp 中的路径及 i 值，修改的地方如下图所示：

```
for(int i=0;i<54;i++)
```

```
{
```

根据自己本机的  
路径进行设置

```
    sprintf(buff1, "/home/duj/maskRCNN/data/djtest/json/rgb_%d.json",
```

```
    i);
```

根据自己本机的  
路径进行设置

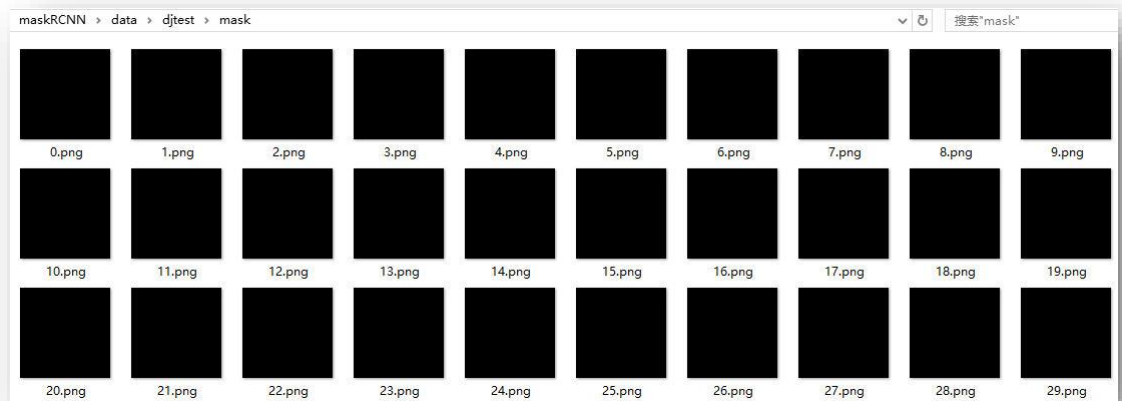
```
sprintf(buff2, "/home/duj/maskRCNN/data/djtest/mask/%d.png", i);
```

- mask\_label.cpp 文件修改完成后，运行

```
g++ -o mask_label mask_label.cpp `pkg-config opencv --cflags --libs`
```

生成名称叫 mask\_label 的可执行文件

- 右键运行 mask\_label，会将标签文件中的 mask 灰度图转为 8 位  
保存到训练集文件夹下的 mask 文件夹中。如下图所示：



## 1.2. 训练模型

### 1.2.1 修改训练模型的文件

- 修改 train\_shapes.py，修改以下三个地方：

- ✓ 修改 load\_shapes 函数中的标签名称，标签编号从 1 开始（先查找到 load\_shapes）

```
self.add_class("shapes", 1, "sf6")
```

说明：以 sf6 表为例，只标注了表的指针。所以标签是 1，标签名称为

sf6，如果训练数据标了两个标签则为：



```
self.add_class("shapes", 1, "led1")
```

```
self.add_class("shapes", 1, "led2")
```

以此类推

#### ✓ 修改 load\_mask 函数中的标签

✚ 先查找到 load\_mask 函数，修改以下内容：

```
for i in range(len(labels)):
    if labels[i].find("sf6") != -1:
        # print "box"
        labels_form.append("sf6 ")
```

标签名称

#### ✓ 修改训练集路径 dataset\_root\_path

#基础设置

```
dataset_root_path="/media/room/dujing/maskRCNN/data/djtest/"
```

用于训练的数据  
所存放的路径

#### ■ 执行 python train\_shapes.py

- ✓ 训练生成的模型保存在 logs 文件夹下，默认名称为 shapes+训练时间

## 1.3 测试模型

### 1.3.1 修改测试模型的文件

#### ■ 修改 test\_shapes.py

- ✓ 参考 train\_shapes.py 修改标签名及路径
- ✓ 修改需要测试的模型路径 model\_path
- ✓ 修改测试图片路径 root\_dir 及输出结果保存路径 output\_dir

■ 修改以下内容：

1. dataset\_root\_path="/media/room/dujing/maskRCNN/data/djtest/"
2. model\_path = os.path.join(ROOT\_DIR, "logs/shapes20180622T1341/mask\_rcnn\_shapes\_0002.h5")
3. root\_dir = '/media/room/dujing/sf6'
4. output\_dir = '/media/room/dujing/maskRCNN/testresultsf6/'

训练数据的路径

模型存放的路径

测试图片的路径

测试结果保存的路径

■ 修改完成后，执行测试

python test\_shapes.py