

# Instructions

## Step1:

对原数据进行缩放或者切割; (支持格式: jpeg、png)  
Resize or crop raw data; (data format: jpeg,png)

## Step2:

创建每个图像对应的 mask 图像; (支持格式: jpeg、png) (mask 图像必须为单通道图)

Create mask image for each data. (Mask image: jpeg, png; single channel)

步骤: (Steps :)

- 1、下载程序 JS Segment Annotator (见 readme 文件), 解压缩 (运行环境 linux);  
Download program - JS Segment Annotator & unzip it.
- 2、按照说明在/data 文件夹下, 新建样本标注文件 xxx.json, 修改 or 增加标注类别, 分别按照样例添加图像和原始 mask 生成路径和名字; (样例文件 example.json)
- 3、修改/js 文件夹下 main.js 文件, 使 dataURL 指向 你自己建立的 xxx.json 文件;  
Follow the user guide of JS Segment Annotator, and then set your own labels and data.

4、运行主目录下文件 index.html 进入图像界面进行标注; (可以使用默认的 firefox 浏览器)

Make your own mask files.

- 5、保存每个图像生成的原始 mask 文件;

Save all mask files.

- 6、运行 segnet\_labeling.m 程序 (需要第三方函数 natsortfiles (见 readme 文件)), 将前面生成的原始 mask 图像 (默认支持 png) 转换为 segnet 的 mask 图像, 并同时生成可视化图像 (后缀为\_colour)。在弹出对话框中分别输入: 选择原始 mask 图像文件夹; 生成的 segnet mask 图像保存文件夹;

Run Matlab code - segnet\_labeling.m to convert masks which you made above to the SegNet format. (Support png format default)

## Step3:

建立 list 文件; (可以使用 create\_list.sh)

Create list file.

格式: (数据与 mask 之间空一格)

```
/SegNet /train/0001TP_006690.png /SegNet /trainannot/0001TP_006690.png  
/SegNet /train/0001TP_006720.png /SegNet /trainannot/0001TP_006720.png
```

## Step4:

根据输入图像大小, 修改网络 upsample 层的参数;

Change to your own parameter of upsample layer, according to input size.

### Step5:

计算并修改 `class_weighting` 参数（在 loss 层）；

Compute the parameters (`class_weighting`) of loss layer.

可以使用 `class_weighting_compute.m`；使用方法：(Method:)

Matlab（2016）运行，在弹出对话框中分别输入：类别总数；选择 `mask` 图片文件夹；结果保存的 `txt` 文件夹路径；`txt` 文件保存的名字。（注：`mask` 文件夹中图片不可太多，可能会溢出，可以分别做成若干个 `txt` 文件，再把对应类权重求均值，且 `mask` 图像格式默认为 `png`，如果为 `jpeg` 需要进行修改）

Run code `class_weighting_compute.m`, and you will get a `txt` file. Then set the value to loss layer.

### Step6:

训练+测试。

Train & test.

注：test set 测试网络精度时才需要 `mask` 文件，如果仅仅是使用训练好网络做 `implement`，那么只需要输入数据，不需要对应的 `mask` 文件。