CSRNet说明：

1. 使用shanghai\_tech数据集，包含part\_A\_final和part\_B\_final两个，每个里面包含train和test两个文件夹，每个文件夹分别包含images图片和ground\_truth两个文件夹，分别为图片和mat文件（GT）
2. 对数据集进行预处理，即根据mat文件中标注的点，首先建一张和图片大小相同的全为0的矩阵，然后根据mat文件中点的位置将全0矩阵相应的位置置1，然后使用高斯核进行卷积（如一个点为1，卷积后和为1），卷积核的方差由距离这个点最近的K个点的平均距离决定（belta为0.3）（查找最近的K个点使用KD数搜索）
3. io.loadmat with h5py.File(‘’,’w’) as hf: h5py.File(‘’,’r’)
4. np.nonzero() 返回数组、矩阵中非零元素的位置，可以这样用

pts = np.array(list(zip(np.nonzero(gt)[1], np.nonzero(gt)[0])))

1. 使用scipy.spatial.KDTree实现KD树

from scipy import spatial

tree = spatial.KDTree(pts.copy(),leafsize = ) #构建KD树

distances,location = tree.query(pts,K = 4) #从构建的KD树查询（pts：要查询的数组 k:返回的最近邻的个数） 返回值：distance：到最近邻居的距离 location：最近邻位置

PS：返回的location和distance的第一列为自身，location为最近邻的索引，distance为距离

参考：<https://vimsky.com/examples/usage/python-scipy.spatial.KDTree.query.html>

1. scipy.ndimage.filters.gaussian\_filter():多维度高斯滤波

输入：矩阵 高斯方差 mode：边界问题（constant）

参考：<https://blog.csdn.net/pursuit_zhangyu/article/details/94138048>

1. 网络结构：vgg16网络结构和6层空洞卷积构成，最后加一个1\*1卷积，load\_weights预训练模型
2. 将生成的数据集密度图resize到原来的1/8，这样网络的输出和GT的大小相同。
3. 损失函数：torch.nn.MSELoss(reduction = ‘sum’) mse损失函数求和
4. torch.nn.DataParallel的使用：

device\_ids = [0,1] net = torch.nn.DataParallel(net,device\_ids)即可使用多GPU训练

参考：<https://zhuanlan.zhihu.com/p/102697821>

1. SGD中的momentum和weight\_decay参数:weight\_decay是正则项前面的系数（L2正则化），momentum为动量

参考：<https://www.jianshu.com/p/1a1339c4acd7>

参考重点：<https://www.zhihu.com/question/24529483>

参考：<https://pytorch-cn.readthedocs.io/zh/latest/package_references/torch-optim/>

其他参考：<https://www.manongdao.com/article-55196.html>

<https://www.pianshen.com/article/1215147034/>

<https://blog.csdn.net/qq_40356092/article/details/108405260>

<https://download.csdn.net/download/qq_40356092/12811426>

<https://blog.csdn.net/qq_40356092/article/details/108140273>

<https://github.com/leeyeehoo/CSRNet-pytorch>