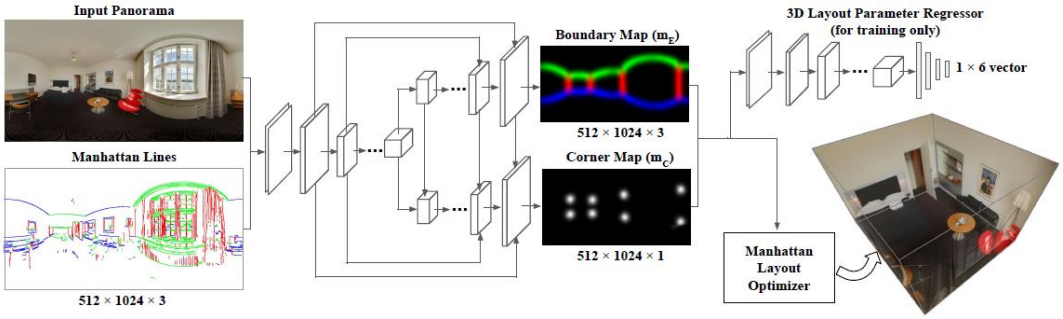
Report 18-4-4

1. 读了一下 tensorflow 实现的 SegNet 代码，大概知道怎么用了，具体细节还没看完；

2. 今天看到一篇发表在 CVPR2018 做 Layout 的文章: LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image. 这篇有提供代码，于是精读了一下。

   但是读完之后发现这篇主要还是做从单张**全景图片**生成 **3D** 的 layout 表达。主要评估标准也是两个 3D 数据集上做的。



第一步：估计地板平面朝向，将图像与地板对齐在一个水平线上(保证所有墙-墙边界都是垂线). 并从全景图检测 Manhattan lines；
第二步：用一个带 skip layer 的 encoder-decoder 网络预测语义边界概率图和 keypoints 的概率图，网络输入是全景图和 Manhattan lines(按通道连接)；
第三步：以语义边界和 keypoints 为输入，训练一个回归 3D Layout 参数的网络，实际主要作用是监督提高语义边界和 keypoints 的准确率；
第四步：后处理优化方法，对 3D Layout 加上 Manhattan word 的约束，优化 3D Layout 的参数。

但对于 2D 的 perspective images 的布局估计效果，并比不上 RoomNet，以下是他们补充材料里的数据:

Method	Pixel Error (%)
Schwing et al. [27]	12.8
Del Pero et al. [6]	12.7
Dasgupta et al. [4]	9.73
LayoutNet (ours)	9.69
RoomNet recurrent 3-iter [16]	**8.36**

Performance on Hedau dataset [11]. We show

Method	Keypoint Error (%)	Pixel Error (%)
Hedau et al. [11]	15.48	24.23
Mallya et al. [22]	11.02	16.71
Dasgupta et al. [4]	8.20	10.63
LayoutNet (ours)	7.63	11.96
RoomNet recurrent 3-iter [16]	**6.30**	**9.86**
RoomNet basic [16]	6.95	10.46

Table 6. Performance on LSUN dataset [9]. LayoutNet ranks sec-

实际他们在做 perspective images 时，并没有用到上述 pipe-line 中的一、三、四步，网络结构也与 pipeline 图中有一些区别，基本上是参照 RoomNet 的方法做的，归纳与之不同的点如下：

- LayoutNet 采用 joint trianing，同时训语义边界和 keypoints
- LayoutNet 网络层数较 RoomNet 浅，另外还引入了 skip layers
- LayoutNet 可以加后处理使得最终结果满足曼哈顿假设，但是 Roomnet 没有
- LayoutNet 采用了简易式的 keypoints 表达，每张图只预测 8 个通道的关键点(因为最多 8 个可见)，而 RoomNet 则是分拓扑一共 48 个通道；

- LayoutNet 在预测 keypoints 时采用的交叉熵 loss，而 RoomNet 采用的是 L2 loss；作者认为效果比不过 roomnet 是因为：The lower accuracy in pixel error mainly results from our simplified room keypoint representation.

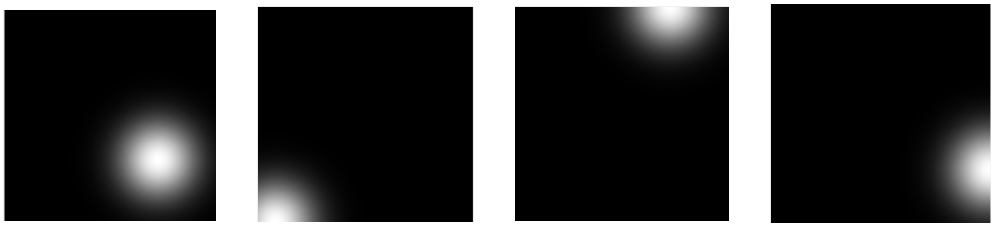
  在他们提供的代码里也没有找到对 perspective images 估计室内布局的部分。只有对全景图片生成 3D Layout 的代码。

3. 做项目汇报 PPT。

Report 18-4-3

1. 把昨天得到的 8000 张训练图片 resize 到 320x320，把相应的 keypoint 坐标转换到 320x320 分辨率以及 40x40 分辨率两组。40x40 是 roomnet 用的值，是为了减少网络参数，提高训练速度，但是精度应该会有所下降，我打算先实现 roomnet-basic，于是做了一组与输入 320x320 匹配的坐标数据。

2. 按照 320x320 和 40x40 分别生成两组 2D 高斯的关键点 ground truth，根据作者回复的邮件，每个点生成一个通道的 ground truth，距离关键点 2.5σ像素视作背景像素，其高斯值置 0.(函数值小于 0.0439)效果如下：

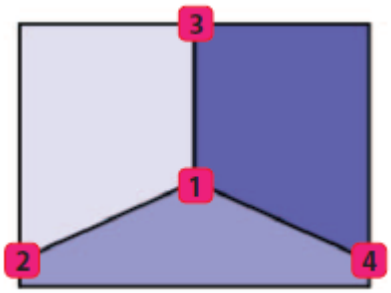


3. RoomNet-Basic 是在 SegNet 的基本结构上修改实现的。我在 github 上找到一个第三方基于 tensorflow 实现的 SegNet. 先熟悉下 tensorflow 实现 SegNet 的代码。
   https://github.com/tkuanlun350/Tensorflow-SegNet

Report 18-4-2

1. 继续写脚本检测剩余的训练数据，并手动校准。另外之前检测时没有考虑到边界上关键点的相对位置约束，部分拓扑有少量漏检测的错例，重新校准。现在已将 LSUN 训练集全部校准完毕。

2. 数据扩充，把所有训练集图片水平翻转。镜像翻转后关键点的坐标和顺序都会发生改变，其中顺序按不同拓扑有不同可能，需要对翻转的训练数据分不同拓扑调整关键点的顺序。

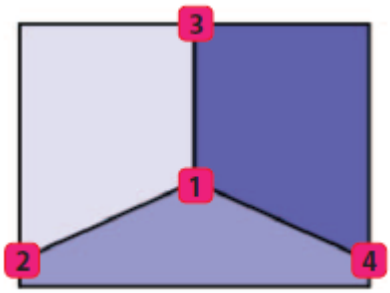


如这种拓扑的图像水平翻转后，需要把点 2 和点 4 的储存顺序交换。
最终得到 8000 张训练图片（一半是水平翻转），和相应按官方规定排序的 keypoints 坐标。

接下来需要按照每个关键点坐标生成相应的 2D 高斯 ground truth，但是有两个小问题不确定，一个是关于 ground truth 的通道数，不知道是对一张图生成 1 张 ground truth 还是对图上的每个点都生成 1 张 ground truth. 还有一个是不知道距离关键点多少个像素时使高斯值置 0. 给作者发了邮件询问。

Report 18-4-1

LSUN 官方数据集中有个别图的关键点 ground truth 储存顺序与他们定义的不大一样。对此，先分 11 种拓扑按照关键点间的相对位置关系，写脚本检测错误的 ground truth，然后手动校准。目前检测并校准了 8 类拓扑(约 3200 张)的数据，还差 3 类拓扑(约 800 张)待矫正。



如图是 LSUN 定义的点序，但是实际 ground truth 中有点序与定义不符合的情况，例如原本标 3 的地方存的是点 4 的坐标，原本标 4 的地方存的点 3 的坐标. Roomnet 需要区分这种点序，因此需要对数据校准。

数据校准记录：

Type0: 2/651　(错误 ground truth 占该拓扑总体比例)　数字表示训练集中第 i 张图片
keypoint error, type0:　　1996
keypoint error, type0:　　3129

Type1: 27/166
keypoint error, type1:　　235
keypoint error, type1:　　328

keypoint error, type1: 423
keypoint error, type1: 623
keypoint error, type1: 642
keypoint error, type1: 867
keypoint error, type1: 955
keypoint error, type1: 1434
keypoint error, type1: 1475
keypoint error, type1: 1700
keypoint error, type1: 1810
keypoint error, type1: 1885
keypoint error, type1: 1953
keypoint error, type1: 2045
keypoint error, type1: 2185
keypoint error, type1: 2194
keypoint error, type1: 2259
keypoint error, type1: 2485
keypoint error, type1: 2612
keypoint error, type1: 3113
keypoint error, type1: 3407
keypoint error, type1: 3572
keypoint error, type1: 3606
keypoint error, type1: 3691
keypoint error, type1: 3733
keypoint error, type1: 3817
keypoint error, type1: 3847

Type2: 0/2

Type3: 11/27 (错误 ground truth 占该拓扑总体比例) 数字表示训练集中第 i 张图片

keypoint error, type3: 400
keypoint error, type3: 1059
keypoint error, type3: 1682
keypoint error, type3: 2052
keypoint error, type3: 2061
keypoint error, type3: 2736
keypoint error, type3: 2893
keypoint error, type3: 3408
keypoint error, type3: 3583 (type4)
keypoint error, type3: 3699
keypoint error, type3: 3889

Type4: 41/1002

keypoint error, type4: 66
keypoint error, type4: 169

keypoint error, type4: 277
keypoint error, type4: 392
keypoint error, type4: 518
keypoint error, type4: 529
keypoint error, type4: 550
keypoint error, type4: 556
keypoint error, type4: 570
keypoint error, type4: 601
keypoint error, type4: 669
keypoint error, type4: 670
keypoint error, type4: 933
keypoint error, type4: 1239
keypoint error, type4: 1393
keypoint error, type4: 1411
keypoint error, type4: 1546
keypoint error, type4: 1557
keypoint error, type4: 1641
keypoint error, type4: 1915
keypoint error, type4: 2107
keypoint error, type4: 2109
keypoint error, type4: 2126
keypoint error, type4: 2162
keypoint error, type4: 2293
keypoint error, type4: 2470
keypoint error, type4: 2495
keypoint error, type4: 2559
keypoint error, type4: 2657
keypoint error, type4: 2663
keypoint error, type4: 2849
keypoint error, type4: 2916
keypoint error, type4: 3249
keypoint error, type4: 3321
keypoint error, type4: 3376
keypoint error, type4: 3449
keypoint error, type4: 3500
keypoint error, type4: 3689
keypoint error, type4: 3745
keypoint error, type4: 3844
keypoint error, type4: 3849

Type5: 2/1808 (错误 ground truth 占比)

keypoint error, type5: 549
keypoint error, type5: 1704

Type6： 7/77

keypoint error, type6: 283
keypoint error, type6: 862
keypoint error, type6: 2195
keypoint error, type6: 2744
keypoint error, type6: 2831
keypoint error, type6: 3391
keypoint error, type6: 3519

Type7: 0/5

Type8: 0/4

Type9: 12/212
keypoint error, type9: 37
keypoint error, type9: 253
keypoint error, type9: 427
keypoint error, type9: 1297
keypoint error, type9: 1943
keypoint error, type9: 2145
keypoint error, type9: 2581
keypoint error, type9: 2603
keypoint error, type9: 2768
keypoint error, type9: 3133
keypoint error, type9: 3357
keypoint error, type9: 3853

Type10: 0/46