

## 构建训练 batch

```
model.batch_input(model.train_dataset)
```

```
dataset.get_batch_op
```

传入 300 个 filename 的 list 构建一个 queue

```
reader.read(filename_queue)
```

返回结果用 NyuDataset 类的 parse\_example 函数解析即可

唯一的处理就是对图片进行 crop

```
dataset.preprocess_op(self._input_width, self._input_height)
```

## JointDetectionModel

```
dms, poses = data.preprocess.data_aug(dms, poses, cfgs, coms)
```

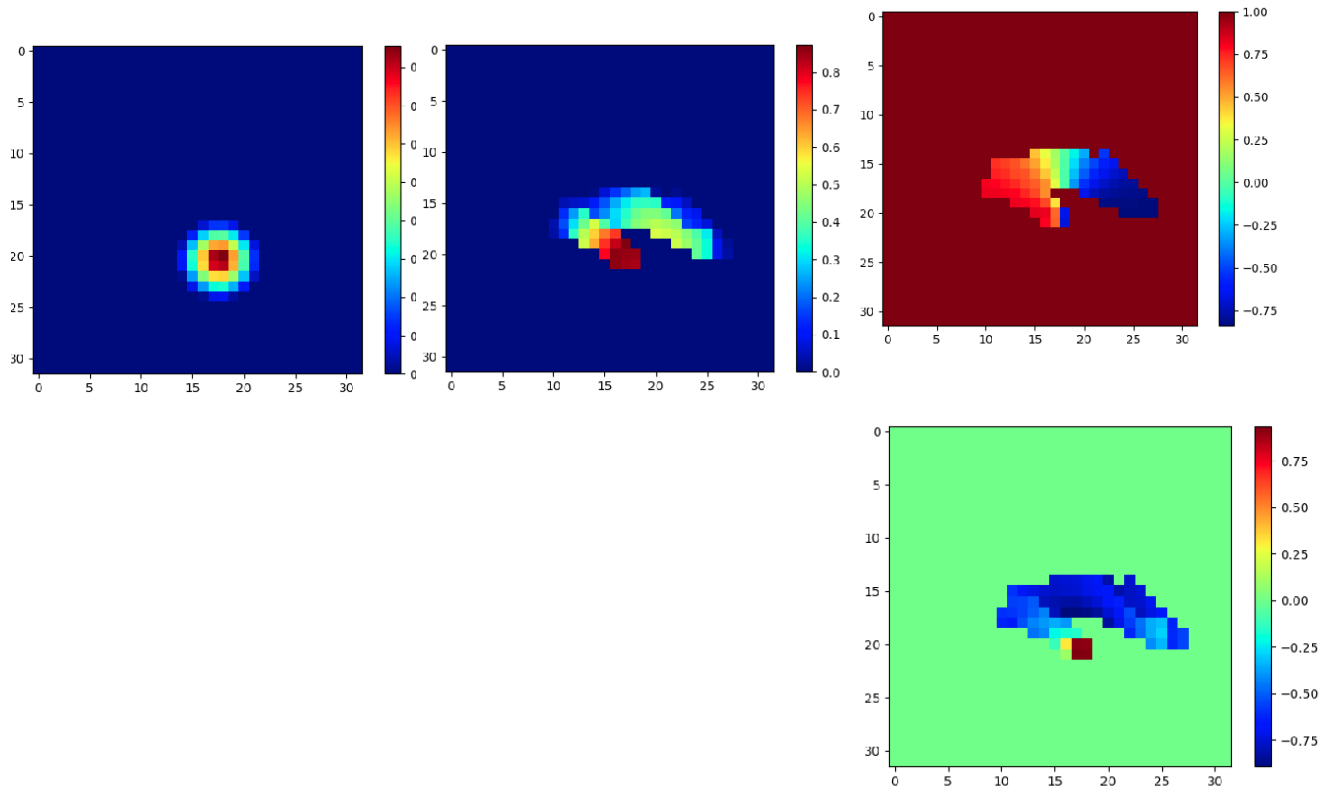
数据增强

生成标签

gt\_hms  
[3,32,32,14]

gt\_hm3s  
[3,32,32,14]

gt\_ums  
[3,32,32,42]



```
end_points = self.inference(normed_dms, cfgs,  
coms, reuse_variables=None, is_training=True)
```

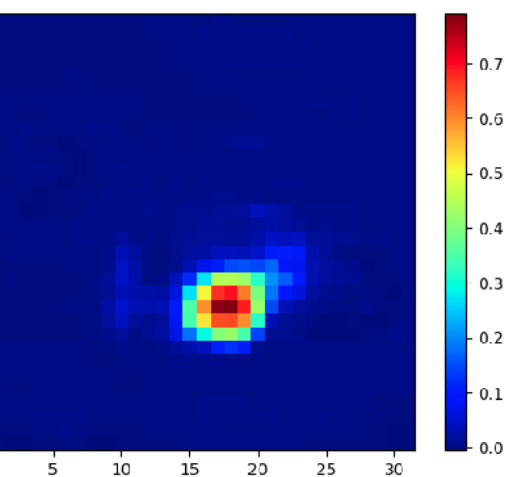
这里实现了论文中的网络结构

cfgs 并没有用到，这是用在将分割后的图像转换成原始坐标，使之能与原始数据比较。

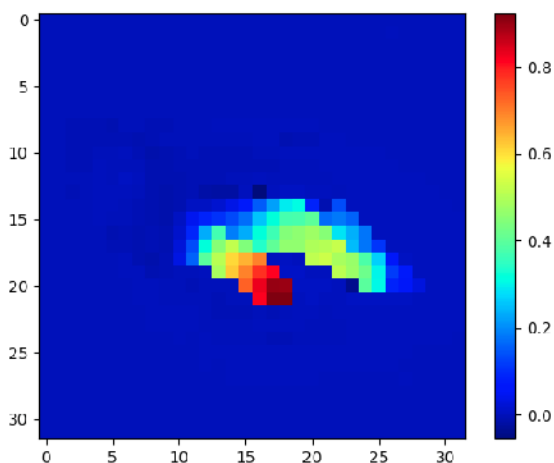
接下来会用到

```
est_normed_poses = self._xyz_estimation(est_hms,
est_oms, est_hm3s, tiny_normed_dms, cfigs, coms)
```

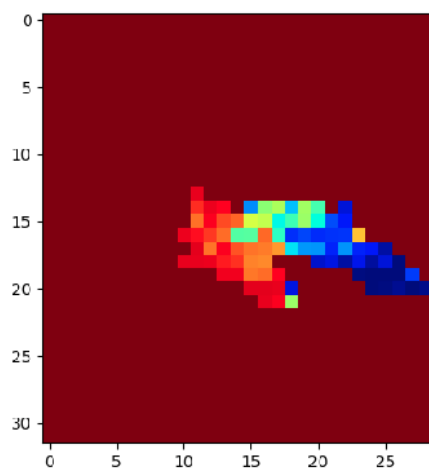
est\_hms  
[3,32,32,14]



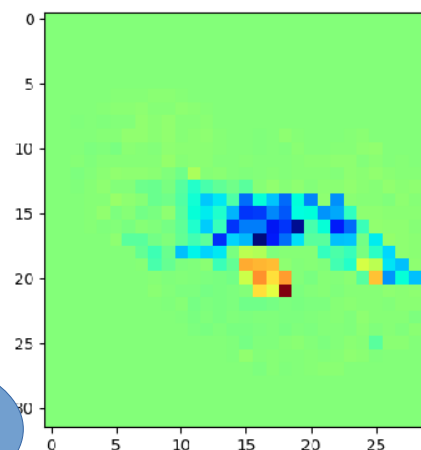
est\_hm3s  
[3,32,32,14]



est\_ums  
[3,32,32,42]



est\_oms  
[3,32,32,42]



从 3d 热图和单位偏移向量恢复偏移图

0.8 · (1-317) ·  $\begin{matrix} um.x \\ um.y \\ um.z \end{matrix}$  (3 32 32 42)

`_xyz_estimation`  
use meanshift to get the final estimation

深度圖中的每個點的坐標 +om  
 $xyzs = xyzs + oms$

2D3D 合在一起得到一个精细化的 hm  
再把深度图中比较远的点去掉  
`refined_hms = tf.multiply(hms+1.0, hm3s)`  
`refined_hms = tf.multiply(refined_hms, dms_mask)`

对于每个关节点用 `refined_hms` 选择 5 个 `xyzs` 中的初始点  
`self._generate_candidates`

权重测量 `xyz_pts` 如何适合 2d hms 估计和 dms 观测  
就是这个候选的点在 2D 上的得分  
`self._get_candidate_weights`

最终的 meanshift 就只看这 5 个点而已。