1、网络结构构建 model.py 以及loss函数



127

255

pic\_size = 140#1,4,4,512 conv4-4

pic\_size = 280#1,21,21,512 conv4-4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer | conv1-2 | conv2-2 | conv3-4 | conv4-4 | correlation |
| filter | 3\*3 | 3\*3 | 3\*3 | 3\*3 |  |
| stride | 2 | 2 | 2 | 2 | 1 |
| channel | 64 | 128 | 256 | 512 | 4 |
| output size\_ search image | 278\*278\*64 | 134\*134\*128 | 59\*59\*256 | 21\*21\*512 | 141\*141\*4 |
| output size\_ example image | 138\*138\*64 | 64\*64\*128 | 24\*24\*256 | 4\*4\*512 |
| layer | conv1 | conv2 | conv3 | conv4 | concatenate |
| filter | 1\*1 | 1\*1 | 1\*1 | 1\*1 |  |
| stride | 1 | 1 | 1 | 1 |  |
| channel | 1 | 1 | 1 | 1 | 5 |
| output size\_ search image | 278\*278\*1 | 134\*134\*1 | 59\*59\*1 | 21\*21\*1 | 278\*278\*4 |
| output size\_ example image | 138\*138\*1 | 64\*64\*1 | 24\*24\*1 | 4\*4\*1 | 124\*124\*5 |

2、辅助函数 ut.py 已经被替代

用于读取图片文件，产生重新缩放后的图片，在测试网络时使用

3、get\_shuffled\_list\_from\_video.py

用于读取cfnet-validation 数据集信息，生成shuffled\_data\_list.txt数据内容如下：

z,x,z\_pos\_x, z\_pos\_y, z\_target\_w, z\_target\_h ,x\_pos\_x, x\_pos\_y, x\_target\_w, x\_target\_h

每一种类物体都是前一张图片做z模板图像，后一张做x搜索图像，以此往下循环，然后把所有的这样的图像对打乱顺序排列，就生成了30507组数据

region\_to\_bbox.py输出模板图像z中心的x，y位置（占原图的比例，原图为1），z的宽和高（也是占原图的比例）

图片的左上角为原点，groundtruth含义分别为xx，xy，xw，xh

4、prepare\_training\_dataset.py

读取shuffled\_data\_list.txt内容，将z,x图像转化为700\*700\*3大小的字符串格式后，与其他标签一起存入protocol buffer数据结构，然后写入training\_dataset.tfrecords文件中

5、read\_training\_dataset.py

将training\_dataset.tfrecords保存的数据读取后，输出batch=8大小的tensor

一个batch包含的内容如下：像素值被归一化到[-1,1]

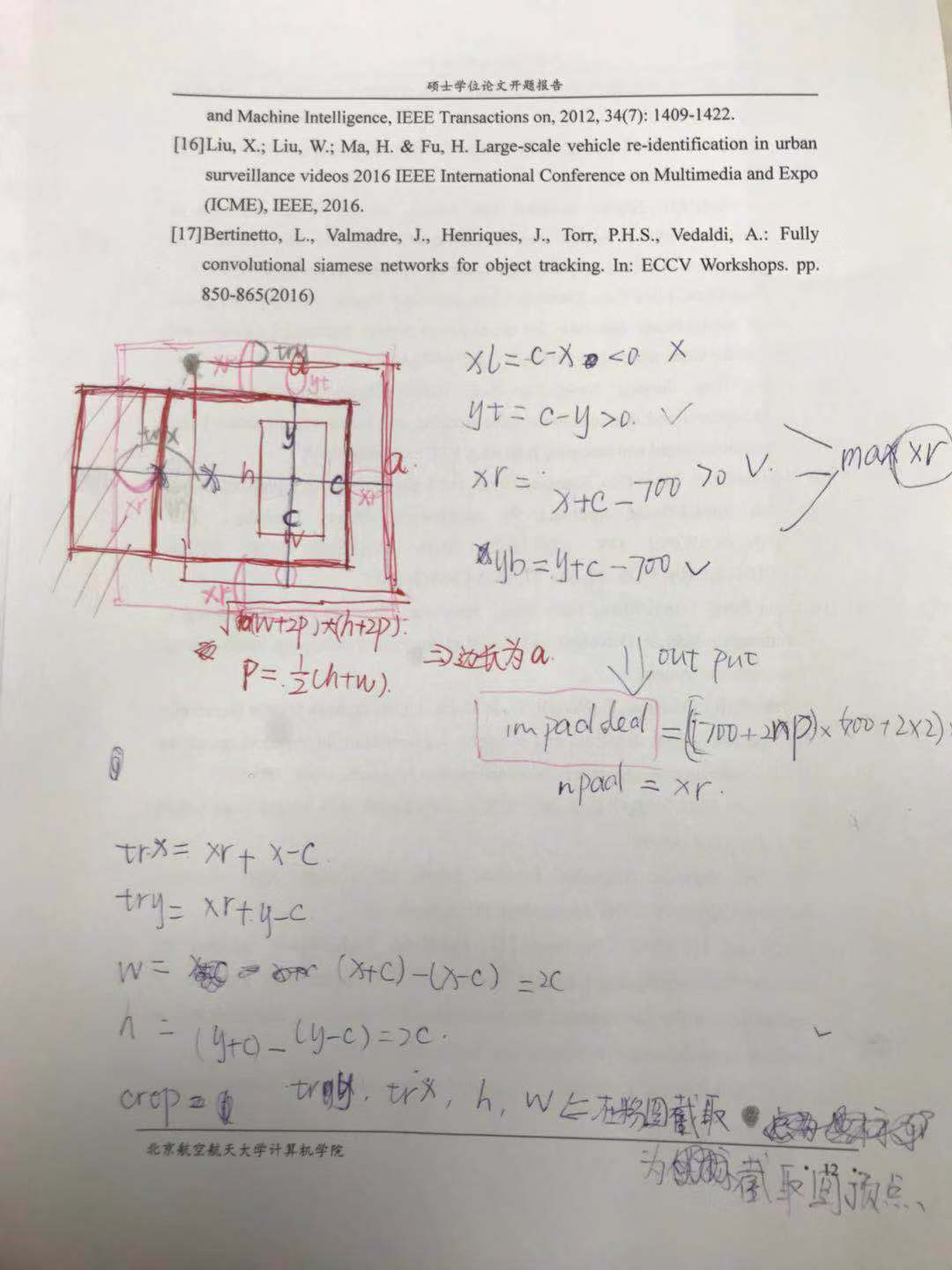
Shape of a: [<tf.Tensor 'shuffle\_batch\_4:0' shape=(8, 700, 700, 3) dtype=float64>, <tf.Tensor 'shuffle\_batch\_4:1' shape=(8, 700, 700, 3) dtype=float64>, <tf.Tensor 'shuffle\_batch\_4:2' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:3' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:4' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:5' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:6' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:7' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:8' shape=(8,) dtype=int32>, <tf.Tensor 'shuffle\_batch\_4:9' shape=(8,) dtype=int32>]

6、crop.py思路如下图所示

padframe输出填充好的图像，保证各个边距离中心点都是c，以后才方便裁剪，输出被填充好的一张图片向量，输出tensor的size[填充值2\*npad+imsize700，填充值2\*npad+imsize700，3]

extractcropz提取模板图像z图片[1,127,127,3]

extractcropx提取模板图像x图片[3,255,255,3]



6、siamese.py思路