RCNN论文阅读笔记

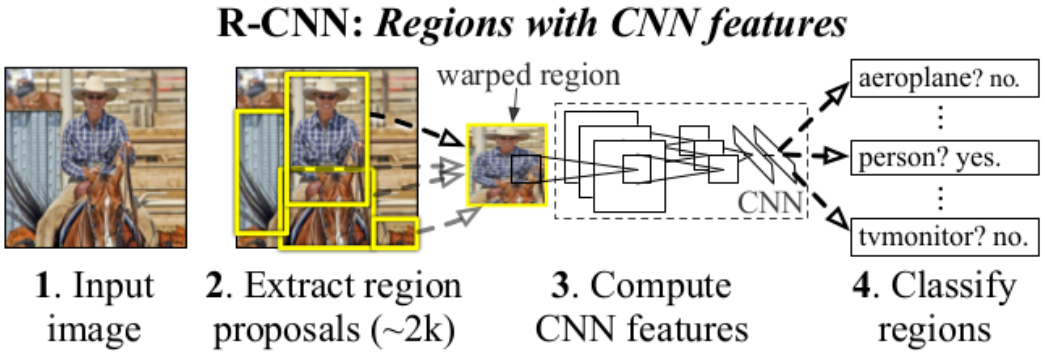
1.前言

图片分类和物体检测是不同的,物体检测需要定位出物体的位置,而图片分类其实是逻辑回归。
随着深度学习的发展，物体检测有了新的发展。从14年以来的发展成果有：

RCNN -> SPPNET -> Fast-RCNN ->Faster RCNN -> YOLO ->SSD

2.RCNN

RCNN是Ross Girshick在2014年Rich feature hierarchies for accurate object detection and semantic segmentation中提出的。



首先输入一张图片，我们先定位出2000个物体候选框，然后采用CNN提取每个候选框中图片的特征向量，特征向量的维度为4096维，接着采用svm算法对各个候选框中的物体进行分类识别。也就是总个过程分为三个程序：
a、找出候选框；
b、利用CNN提取特征向量；
c、利用SVM进行特征向量分类。

---

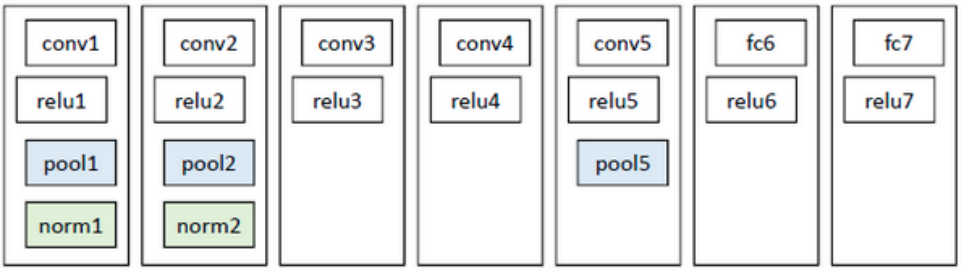
a.提取候选框:

当输入一张图片时，要提取物体的候选框（一看到这个，我最先想到的是用不同size的窗口去滑动搜索).
RCNN采用的是selective search（ Uijlings, Jasper RR, et al. “Selective search for object recognition.”）。
selective search 大概的思路是
1. 使用 Efficient Graph-Based Image Segmentation的方法获取原始分割区域R={r1,r2,…,rn}
2. 初始化相似度集合S=∅
3. 计算两两相邻区域之间的相似度（见第三部分），将其添加到相似度集合S中
4. 从相似度集合S中找出，相似度最大的两个区域 ri 和rj，将其合并成为一个区域 rt，从相似度集合中除去原先与ri和rj相邻区域之间计算的相似度，计算rt与其相邻区域（原先与ri或rj相邻的区域）的相似度，将其结果添加的到相似度集合S中。同时将新区域 rt 添加到 区域集合R中。
5. 获取每个区域的Bounding Boxes，这个结果就是物体位置的可能结果L.
而其中相似度的计算是采用了多元化的计算，是颜色相似度， 文理相似度， 大小相似度， 吻合相似度四者的加权和。（具体参考原论文和参考资料2和3）
最后我们是的出了大概2000个候选框。
候选框为了适应物体的多尺度问题(同一类物体在图像中的成像是不同的)，候选框大小是不同的，但是cnn的输入图片的大小是固定的（论文中是277×277）。
所以原文中采用了各项异性缩放进行缩放，不管图片的长宽比例，直接将其缩放到277*277,这简直太粗暴了，因为直接缩放会导致图像扭曲，会对cnn提取体征有影响，也是本文的缺点之一。

b、利用CNN提取特征向量

由于物体检测中，标签好的物体数据较少，所以直接随机初始化网络的话，样本的数量是不够的。采用迁移学习的方法，使用已经训练好的图片分类的网络，在使用标签好的物体数据进行有监督的参数微调（fine-tuning）。其中fine-tuning的

过程要在最后加一层网络，假设要检测的物体类别有N类，那么我们就需要把上面预训练阶段的CNN模型的最后一层给替换掉，替换成N+1个输出的神经元(加1，表示还有一个背景)，然后这一层直接采用参数随机初始化的方法，其它网络层的参数不变；接着就可以开始继续SGD训练了。
网络架构可以选择Alexnet和VGG16，Alexnet的精度（58.5%）小于VGG16的（66%）
VGG精度较小，学习速率较小，但是计算量是Alexnet的7倍。采用的是alexnet



Alexnet中conve5层神经元个数为9216、 f6、f7的神经元个数都是4096，通过这个网络训练完毕后，最后提取特征每个输入候选框图片都能得到一个4096维的特征向量。

c、利用SVM进行特征向量分类

CNN f7层特征被提取出来，那么我们将为每个物体累训练一个svm分类器。当我们用CNN提取2000个候选框，可以得到2000*4096这样的特征向量矩阵，然后我们只需要把这样的一个矩阵与svm权值矩阵4096*N点乘(N为分类类别数目，因为我们训练的N个svm，每个svm包好了4096个W)，就可以得到结果了.

3.RCNN检测结果

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression (BB) is described in Section C. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN $pool_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc_6	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc_7	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT $pool_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc_6	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc_7	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc_7 BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**
DPM v5 [20]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [28]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [31]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

Table 2: Detection average precision (%) on VOC 2007 test. Rows 1-3 show R-CNN performance without fine-tuning. Rows 4-6 show results for the CNN pre-trained on ILSVRC 2012 and then fine-tuned (FT) on VOC 2007 trainval. Row 7 includes a simple bounding-box regression (BB) stage that reduces localization errors (Section C). Rows 8-10 present DPM methods as a strong baseline. The first uses only HOG, while the next two use different feature learning approaches to augment or replace HOG.

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	**35.7**	62.1	64.0	66.5	**71.2**	62.2
R-CNN O-Net BB	**73.4**	**77.0**	**63.4**	**45.4**	**44.6**	**75.1**	**78.1**	**79.8**	**40.5**	**73.7**	**62.2**	**79.4**	**78.1**	**73.1**	**64.2**	35.6	**66.8**	**67.2**	**70.4**	71.1	**66.0**

Table 3: Detection average precision (%) on VOC 2007 test for two different CNN architectures. The first two rows are results from Table 2 using Krizhevsky et al.'s architecture (T-Net). Rows three and four use the recently proposed 16-layer architecture from Simonyan and Zisserman (O-Net) [43].

4.总结

提出了使用cnn的模型用于物体检测,采用的是多个region +多个CNN + pooling.
缺点：速度慢，wrapping缩放变形，重复计算较多。

目录———KAM_FANG

参考资料：
1.RCNN原文:
https://arxiv.org/abs/1311.2524
2. http://www.cnblogs.com/zhao441354231/p/5941190.html
3. http://blog.csdn.net/surgewong/article/details/39316931
4. http://blog.csdn.net/hjimce/article/details/50187029
5.Selective_Search原文:
https://www.researchgate.net/publication/262270555_Selective_Search_for_Object_Recognition