



(<http://lib.csdn.net/base/deeplearning>)

深度学习 (<http://lib.csdn.net/base/deeplearning>) - 深度学习应用 (<http://lib.csdn.net/deeplearning/node/747>) - 图像检测 (<http://lib.csdn.net/deeplearning/knowledge/1726>)

👁 1224 💬 0

基于Faster-rcnn及FCN的中文OCR文本定位

作者：wqzghost (<http://my.csdn.net/wqzghost>)

启发

光学字符识别 (Optical Character Recognition, OCR) 是指对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程。一般分为两个步骤：文字定位，即找到文字在图片中的位置；文字识别，即识别出找到的文字。文字定位也可能包含一些二值化，矫正的步骤。

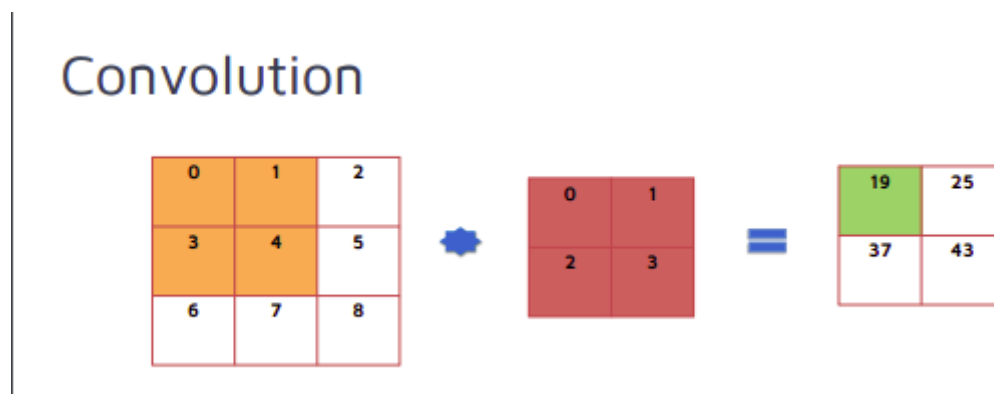
文字定位常见的算法MSER +SVM或者MSER + CNN，是一种自底向上的方法：首先产出大量MSER区域，进而用SVM或者CNN进行二分类过滤掉非文字区域，最后合并文字区域产生文字区域。这种方法一方面产生大量候选MSER区域，另一方面，文字合并中考虑到中文特殊的左右，上下结构，过程会非常复杂，结果难以把控。

受启发于ImageNet比赛中的Image Classification任务，能否将Faster-rcnn 以及FCN 算法应用于OCR*的文本定位过程中。答案应该时肯定的。现在相当于进行一个只有两类的Image Classification，一类是图片背景，另一类则是文本行。不在像MSER那样把字切碎了，在合并起来。而是直接找到一整行文字。这就是它的优势所在。

之所以选择Faster-rcnn和FCN算法，一方面它们是目前效果比较好的算法，另一方面它们代表了两种不同的思路：Faster-rcnn是以 bounding box为训练目标，FCN则是一种逐像素labelling的方法。在给出实验结果之前，比较一下这两种算法。

Faster-rcnn

Faster-rcnn的思想在于利用神经网络中conv + pooling并没有改变特征在图像中的相对位置。真正改变是在fc层。如下图：



可以清晰的看到左侧图片，在中间卷积核的作用下，最终得到数值19（观察图片中矩阵带有颜色的部分）。也就是19这个特征值对应图片中的左上角，这个值在图片中有明显的位置关系。同样，特征值25对应图片中的右上角，即值为1，2，4，5的部分。

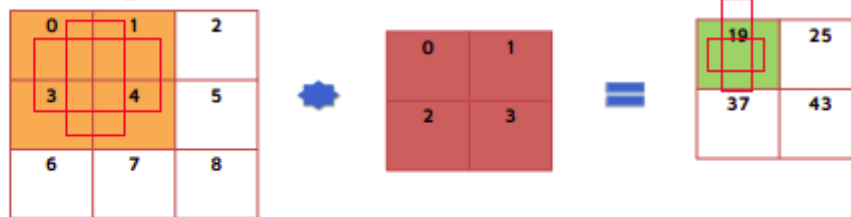
pooling也有同样的效果：

Max pooling



实际上这也是rpn网络的原理，rpn网络在全卷积网络，即只有conv + pooling，的输出上，每一个特征值产生9个anchor，对应原图中的相应位置。

Convolution



rpn网络的巧妙应该有三点：1) rpn利用了conv + pooling不改变相对位置关系的特性。2) cnn特征不仅用来做分类，即特征产生标签，还用来做bounding box的回归，即特征产生bounding box坐标。3) rpn突破了我们理解，在bounding box回归上，判断网络优良的标准是生成的box和ground truth的IOU到底有多大。

但是，rpn网络也有它的缺点：rpn生成的anchor，有位置关系以及一定的长宽比例，整个训练过程通过调节anchor的坐标来达到与ground truth的最大IOU。如果anchor长宽比例设置不合理的话，对一个长宽比例失调的物体就会很难找到一个将它整个包围的bounding box。以瓶子为例，因为它比较高，所以瓶子上端和下端特征距离比较远，这样很可能产生下面两个anchor，都不能把瓶子包围住。或许瓶子中间特征产生的anchor，通过调节是可以的，或者还是有误差。这里只是个例子，为了说明情况而已。



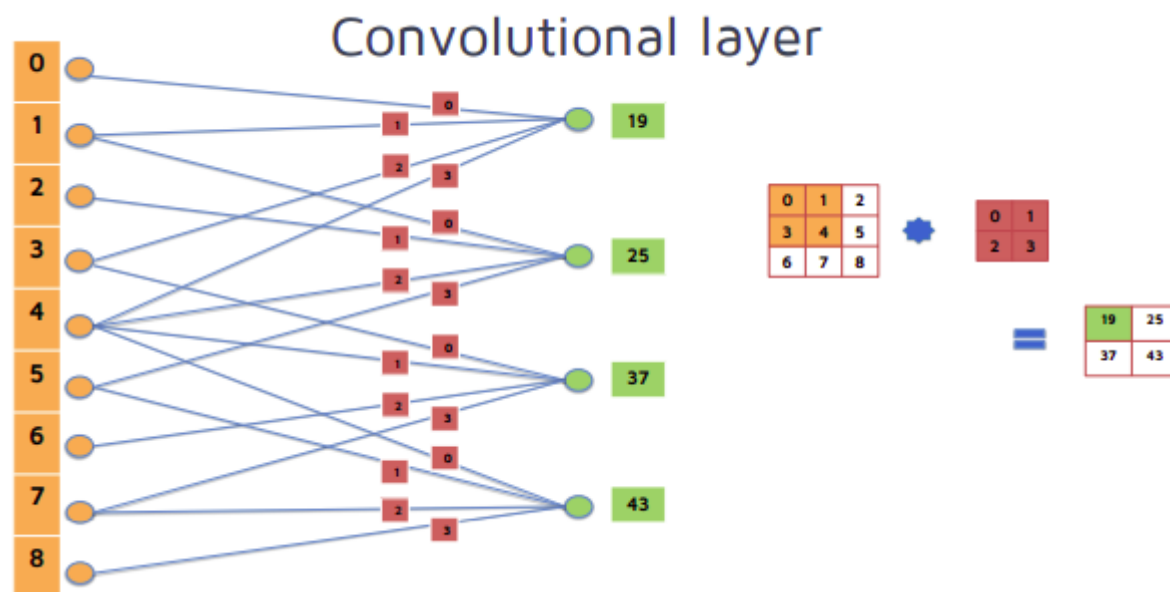
另外一个问题就是，Faster-rcnn采用了nms，这样会导致互相有遮盖的物体不能同时被检查出来，如果发现问题可以尝试把nms去掉试试。

FCN

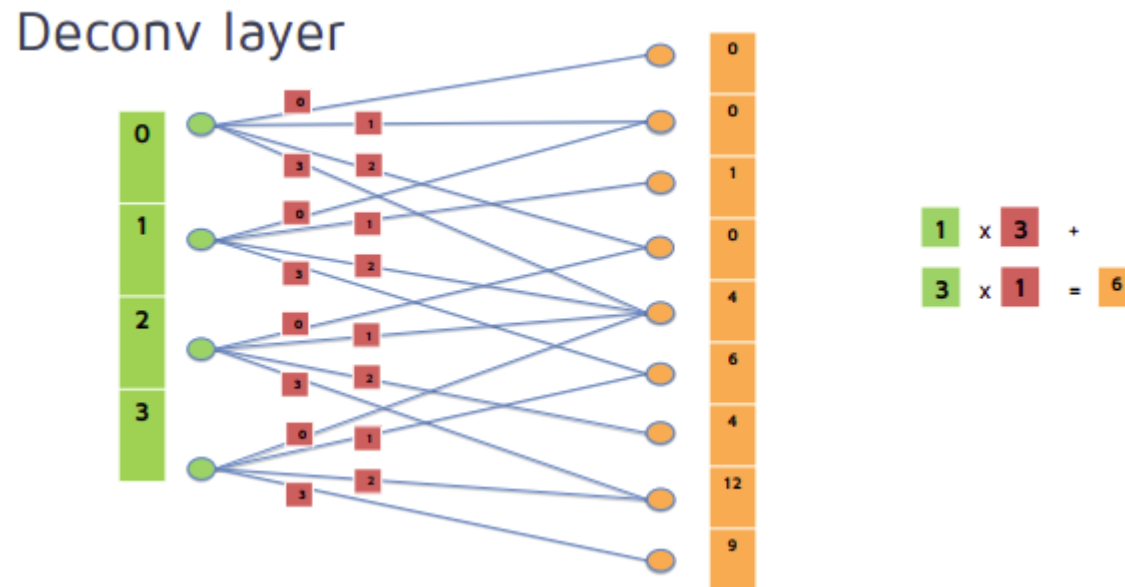
FCN (fully convolutional network)，全卷积网络，没有fc层。所以它在整个网络中是不会丢失位置信息的，与Faster-rcnn不同的是FCN对每一个像素分类，在OCR里面每个像素分成两类：文字行和非文字行（背景）。比如一张300*400的图片，最后FCN会对所有300*400个像素做分类。

FCN问题的关键是如何将卷积后的特征还原成原始图片大小，这个过程在许多深度学习框架里叫做 deconvolution（参加），许多人认为并不准确，因为整个过程并不是真正的deconvolution（参见wikipedia (<https://en.wikipedia.org/wiki/Deconvolution>))，而是应该称为transposed convolutional network。这是一个什么样的过程呢？

如果将图片reshape成一个一维矩阵，然后在convolution，情况是这样的：



deconvolution是这样的：



可以看到deconvolution实际是convolution的一个反过程，熟悉caffe的同学可以看下deconv_layer (https://github.com/BVLC/caffe/blob/80f44100e19fd371ff55beb3ec2ad5919fb6ac43/src/caffe/layers/deconv_layer.cpp#L32) 实际是改变了forward和backward的调换位置。细心的同学也会发现，虽然过程相反，但实际上值不一样。其实，deconvolution并不是要还原成原来一模一样的图片，而是将特征值恢复到相应的位置。而且，在实际应用中，deconvolution也可以利用bilinear卷积核，也就是双线性插值的一个resize的过程。

相对于convolution来说，恢复max pooling容易的多，只需要在做max的时候记录下最大值的位置，在unpooling的时候根据相应规则赋值相应位置即可。

FCN的优势在于利用deconvolution，unpooling等操作，将特征矩阵恢复到原图大小，然后对每一个位置上的像素做预测，形成一个heat map，整个过程并没有fc的出现。可想而知，不会出现Faster-rcnn的情况，对一些比例失调的物体鲁棒性差。当然，它也有自身的问题，就是计算时间复杂度大。

实验对比

这里仅给出FCN在人工数据集上的效果，该数据集来自于vgg的《SynthText in the Wild Dataset》，参见这里 (<http://www.robots.ox.ac.uk/~vgg/data/scenetext/>)。效果如下图：







图中用红色描出了FCN识别出的区域，总结起来大概有两个问题：

- 1) 对大号字的识别效果不佳，原因可能是因为字号太大，卷积窗口太小，所以找到整个字的特征，解决方法可以添加多尺度，但是这样做的结果是带来更大的时间复杂度。
- 2) 当行与行间距过小的话，很难将行分开，就是图中看到的连成一片的区域。一些论文中针对这个问题都会进行两阶段处理：第一阶段找出图中文字区域，大概就是现在这个样子；第二阶段对找到的区域继续fine tuning。有兴趣的读者可以参考《Multi-Oriented Text Detection with Fully Convolutional Networks》，《Accurate Text Localization in Natural Image with Cascaded Convolutional Text Network》。

我本人采用的另一种方式：针对每一个文本行，我只取中间部分作为训练标签。这个中间部分占整个文本行高度的0.4到0.8不等，分别实验。最后虽然取得了一定效果，但是相对小号字上的定位效果就差了些。希望有思路的同学也能分享下。

引用

A guide to convolution arithmetic for deep learning (<https://arxiv.org/pdf/1603.07285v1.pdf>)

[查看原文>>](http://blog.csdn.net/wqzghost/article/details/53228468) (<http://blog.csdn.net/wqzghost/article/details/53228468>)



1

看过本文的人也看了：

- 深度学习知识结构图
(<http://lib.csdn.net/base/deeplearning/structure>)
- 【转载】SSD 下的 MySQL IO 优化
(<http://lib.csdn.net/article/deeplearning/53060>)
- SSD的配置及运行
(<http://lib.csdn.net/article/deeplearning/53859>)
- faster rcnn源码理解（二）之AnchorTarge...
(<http://lib.csdn.net/article/deeplearning/50134>)
- Faster-RCNN+ZF用自己的数据集训练模...
(<http://lib.csdn.net/article/deeplearning/61652>)
- windows 下 编译py-faster-rcnn , py-rfcn...
(<http://lib.csdn.net/article/deeplearning/55081>)

发表评论

输入评论内容


[发表](#)

0条评论

[公司简介](http://www.csdn.net/company/about.html) | [招贤纳士](http://www.csdn.net/company/recruit.html) | [广告服务](http://www.csdn.net/company/marketing.html) | [联系方式](http://www.csdn.net/company/contact.html) | [版权声明](http://www.csdn.net/company/statement.html) | [法律顾问](http://www.csdn.net/company/layer.html) | [问题报告](mailto:webmaster@csdn.net) | [合作伙伴](http://www.csdn.net/friendlink.html) | [论坛反馈](http://bbs.csdn.net/forums/Service)

网站客服 杂志客服 (<http://wpa.qq.com/msgrd?v=3&uin=2251809102&site=qq&menu=yes>) 微博客服 (<http://e.weibo.com/csdnsupport/profile>) webmaster@csdn.net

400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved  (<http://www.hd315.gov.cn/beian/view.asp?bianhao=010202001032100010>)