《Shape Robust Text Detection with Progressive Scale Expansion Network》论文推荐

贾程皓
21821117

摘 要

由于深度卷积神经网络的快速发展，场景文本检测领域在近几年以来取得了快速的发展。但是其在工业界上还无法进行实际应用。主要由于存在着两大挑战。一方面，大多数现有的方法都需要采用四边形的边框来确定边界，但这种边界框在定位任意形状的文字时的性能很差，可想而知精确度很低。另一方面，对于场景中彼此十分接近甚至互相干扰的文本，目前的技术可能会产生误检测。传统的基于分段的技术手段可以缓解四边形边界框的性能问题，但是通常无法解决误检测问题。

我要推荐的这篇文章是 CVPR 2019 录用的一篇有关场景文字检测的文章。在该篇文章中提出了一种新颖的渐进式尺度可扩展网络（PSENet），该模型可以精确鲁棒地检测场景中任意形状的文本实例。

在本文中，首先会对该论文的工作进行阐释和复述，接着会解释我选择这篇文章推荐的理由，并且结合自己的一些简单实验阐述自己的一些想法和思考。

1. 简介

近年以来，自然场景下的文本检测因为其众多应用方向而受到关注。比如文本检测结果可用于场景理解、产品识别、自动驾驶等等。然而，这项工作存在很多挑战。由于前景文本和背景物体的变化，以及极端情况下的光照条件或者遮挡物等原因，自然场景中的文本检测面临比较大的挑战。

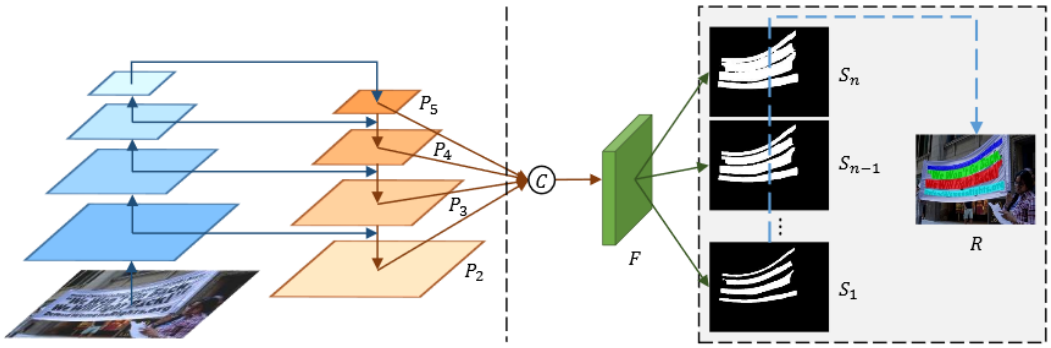
深度卷积神经网络是应用于计算机视觉方向上的利器，利用深度卷积神经网络，基于边界框回归思想的方法，已有许多工作可以成功地利用特定方向的矩形或四边形来定位文本目标。目前，该工作存在着两个重要的难题。一是矩形或四边形边框无法检测具有任意形状的文本实例。二是传统的逐像素分割方法虽然可

以提取任意形状的文本实例的区域，但是对于比较接近的，甚至可能互相干扰的实例区域，该方法仍然可能无法将它们分开，而把它们误检测为一个文本实例。

为了解决这些问题，本文提出了渐进式的，可扩展的模型结构。首先，该模型作为基于分割的方法，它可以定位具有任意形状的文本。第二，该模型采用一种渐进式扩展算法，利用它可以成功识别紧密相邻的文本实例。具体来说，首先该模型会为每个文本实例分配多个预测的分割区域，也就是文中所谓的“核”。每个核与文本实例有相似的形状，它们位于相同的中心点，但是比例不同。然后受到广度优先搜索算法的启发，模型会从最小的内核开始，通过逐渐在更大的内核中包含更多的像素来扩展区域，直到搜索到最大的内核。

2. 模型

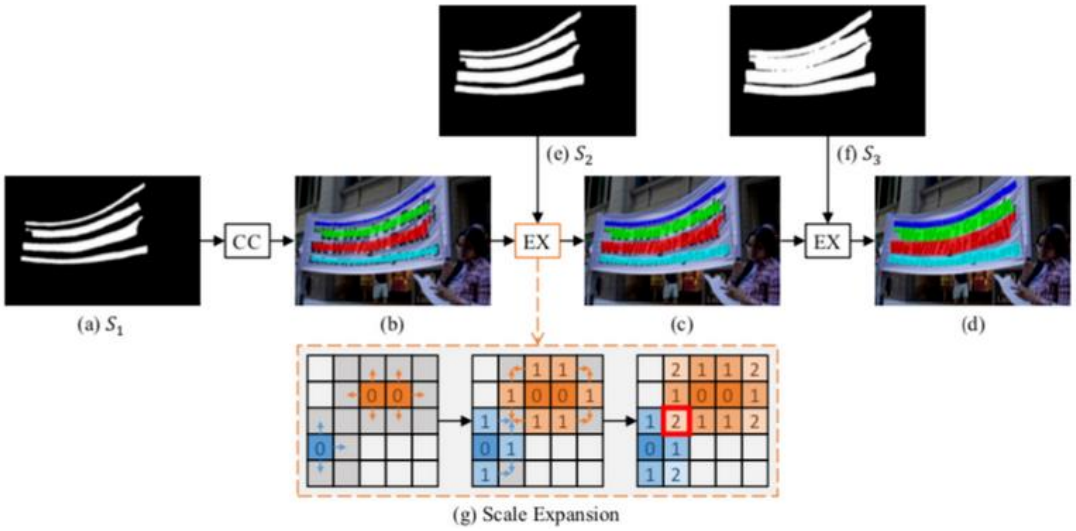
首先介绍 PSENet 的总体结构。PSENet 的总体结构如下图所示：



Progressive Scale Expansion

PSENet 模型的主干网络采用 resnet，网络的框架与 FPN 结构类似。如图所示，模型会先利用 resnet 在图像中提取出四层 Feature maps，然后将得到的四层特征图进行融合得到特征图 F。然后特征图 F 会被映射到 n 个分支来产生不同的分割结果，即 S1，…，Sn，而每一个 S 都会在某个规模上作为所有文本实例的一个分割掩码。其中 S1 是最小尺寸的分割图，里面不同的连通区域都可以看作不同文字块的核。而 Sn 是最大规模的分割图，其结果是完整的文字块。最后，再通过一个渐进扩展算法不断地扩展每个核，直到变成完整的文字块。

接下来简单讲解一下 PSE 渐进扩展算法，这既是本文提出模型名字的由来，也是这篇论文的主要贡献所在。该算法如下图所示：



渐进扩展算法的输入是不同尺寸的核的分割图，输出则是最后的检测结果。渐进扩展算法的提出是为了解决相近文本实例易被误检测为一个整体的文本的问题，其思路是受到广度优先搜索算法的启发。该算法会首先从最小的分割结果 S1 开始，找出若干个分割区域，这样就能得到所有文本的中心区域。之后将这个分割结果与 S2 进行合并。合并的方法与广度优先搜索算法思路类似：考察分割结果周围的像素点，根据 S2 的结果，判断该像素属于哪一个文本实例，然后将该像素点归到该实例中。对于临界的、可能产生冲突的像素点，采用“先到先得”的方式来解决。

3. 实验

文中作者将 PSENet 应用在了不同的数据集上。PSENet 在常规的数据及 ICDAR 2015 和 ICDAR 2017 MLT 上的表现不弱于主流的检测算法。但是在弯曲文字数据集 SCUT-CTW1500 上的表现相较于之前最好的结果还要高 6.37%。

4. 推荐理由

我之所以推荐这篇文章有以下几点理由：

首先是实际效果方面。这篇文章通过提出 PSENet 模型，为每个文本实例生成不同比例的核，并将最小比例的核逐步扩展生成完整形状比例的核来适应不同大小的文本示例。通过这样的方法，由于最小尺度的核之间存在较大的几何边距，从而有效地分割了场景中一些紧密的文本实例，从而更容易地使用分段方法来检

测任意形状的文本实例。而这篇文章提出的渐进式扩展算法，能够比较精确地分割开距离较近的几个文本实例，有效解决了传统算法误检测的问题。

其次是文章背后折射出的作者思路。文章想要解决的问题是任意形状的文本的检测和误检测问题的解决。由于在之前的工作中，使用基于分割的方法做文本检测得到了很好的结果，所以本文很简单朴素却抓住了问题本质的思路，使用基于分割的方法，但针对文本毗邻时的单例分割做不好的问题，提出从文本中心开始，扩散到整个文本，分步完成预测的方法。

这启示我们在解决问题的时候，尽量应该避免“无中生有”，而应该在既有工作的基础上，针对特定问题进行特定方式的改进来解决问题。

5. 思考与实验

本文的工作及代码已经在 github 上开源，而且有 pytorch 和 tensorflow 两种框架的实现。另一方面，由于我的专业研究方向和这一主题交集较少，所以我并没有再重新进行一次实现。但是我下载了 tensorflow 版本的全部代码并进行了研读和理解，从而对论文内容有了更深刻和正确的认识。

但我自己也做了一些实验。首先，我按照文章核对代码确认无误后进行实验，经过三次实验后发现，并得不到文章中 87.08%如此之高的 F 值，只有约 84.8%，当然，这比其他 baseline 的成绩也要好很多。这暴露出该模型的一个缺点，需要调整的参数和超参数比较多，所以相对来说鲁棒性并不是那么好。同时，数据集改变之后，还必须重新调整参数。

另一方面，我注意到文章中的损失函数定义时采用的是分割常用的 dice coefficient，而不是交叉熵函数。所以我将损失函数修改为交叉熵函数重新跑一遍实验，这时最终的测试结果还要再降低大约一个点。这说明，目前深度学习的可解释性仍然有待加强。同样具有解释意义的损失函数却可能在最终模型的泛化性能上有很大不同的影响。