大白话《Shape Robust Text Detection with Progressive Scale Expansion Network》

whai362
whai362.github.io

论文题目：Shape Robust Text Detection with Progressive Scale Expansion Network

论文地址：arxiv.org/abs/1806.0255...

源码地址：github.com/whai362/PSEN...（代码目前还在整理中，后面会放出）

问题1：如何检测任意形状的文字块?

目前很多文字检测的方法都是基于bounding box回归的，虽然它们在常规的文字块检测任务中取得了不错的效果，但是它们很难准确地定位弯曲文字块（Fig. 1 (a)）。而基于语义分割的方法恰好能很好地解决这个问题，语义分割可以从像素级别上分割文字区域和背景区域。

Fig. 1：不同方法的文字检测效果。

问题2：如何分离靠的很近的文字块?

直接用语义分割来检测文字又会遇到新的问题：很难分离靠的很近的文字块（Fig. 1 (b)）。因为语义分割只关心每个像素的分类问题，所以即使文字块的一些边缘像素分类错误对loss的影响也不大。对于这个问题，一个直接的思路是：增大文字块之间的距离，使它们离得远一点。基于这个思路，我们引入了新的概念“kernel”，顾名思义就是文字块的核心。从Fig. 2中我们可以看到：利用“kernel”可以有效地分离靠的很近的文字块。

Fig. 2. kernel的效果。

问题3：如何通过“kernel”来构建完整的文字块?

上述的“kernel”只是文字块的核心，并不是完整的文字块，不能作为最终的检测结果。在这篇文章中，我们通过一种基于广度优先搜索的渐进扩展算法来构建完整的文字块。这个方法的核心思想是：从每个“kernel”出发，利用广度优先搜索来不断地合并周围的像素，使得“kernel”不断地扩展，最后得到完整的文字块。具体算法后面会详细介绍。

解决了以上的三个问题，相信大家对检测任意形状的文字块也有了初步的想法，下面开始讲一下PSENet的总体结构和渐进扩展算法。

首先是PSENet的总体结构。

如图Fig. 3所示，PSENet的主干网络是FPN，一张图片 I 通过FPN可以得到四个Feature Map（ P_2, P_3, P_4, P_5 ），然后通过函数 $C(\cdot)$ 合并 P_2, P_3, P_4, P_5 ，得到 F 。 $C(\cdot)$ 的具体公式如下：

$$F = C(P_2, P_3, P_4, P_5) = P_2 \parallel \mathrm{Up}_{\times 2}(P_3) \parallel \mathrm{Up}_{\times 4}(P_4) \parallel \mathrm{Up}_{\times 8}(P_5),$$

其中， “$\parallel$” 表示拼接操作， $\mathrm{Up}_{\times 2}, \mathrm{Up}_{\times 4}, \mathrm{Up}_{\times 8}$ 分别表示2，4，8倍上采样。接着，通过 F 来预测不同kernel scale的分割图 $S_1, S_2, \ldots, S_n$ 。 其中 S_1 是最小kernel scale的分割图，里面不同的连通区域都可以看作不同文字块的“kernel”。 S_n 是最大kernel scale的分割图，是完整的文字快。最后通过一个渐进扩展算法（Progressive Scale Expansion）去不断地扩展 S_1 中的每个“kernel”，直到变成 S_n 中完整的文字块。

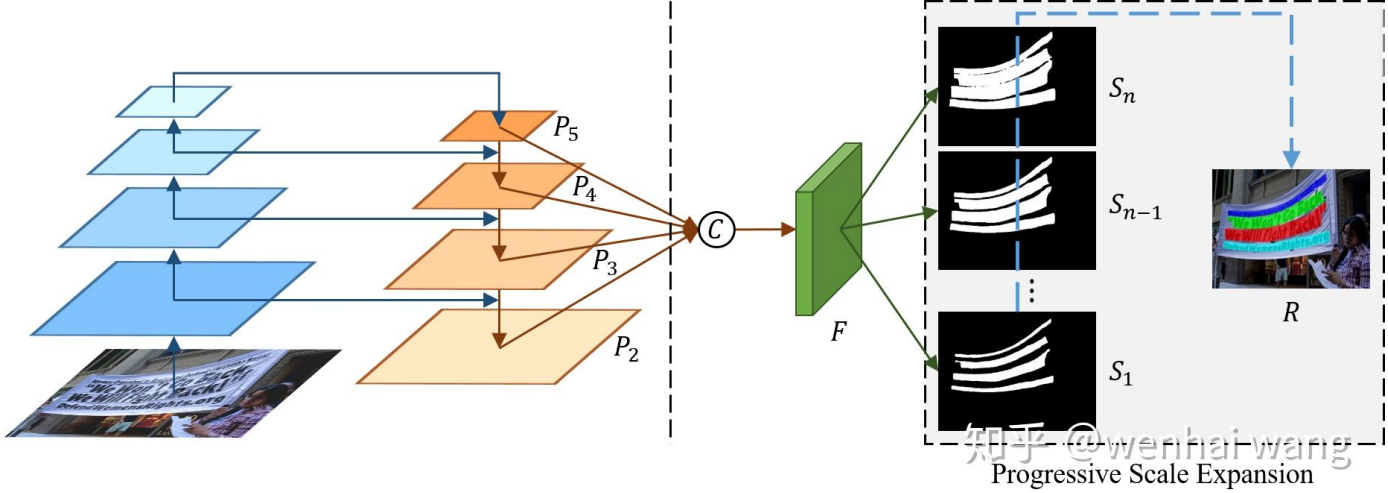

Fig. 3. PSENet的总体结构。

下面讲渐进扩展算法（Progressive Scale Expansion）。

渐进扩展算法的输入是不同kernel scale的分割图 $S_1, S_2, \ldots, S_n$，输出是最后的检测结果 R。不妨令 $n = 3$，则渐进扩展的算法的过程如Fig. 4所示。首先，对 S_1 求连通区域，得到不同文字块的“kernel”（Fig. 4 (b)）。然后，通过Fig. 4 (g)所示的扩展操作合并 S_2 中的文字像素，得到扩展后的结果（Fig. 4 (c)）。最后，使用同样的扩展操作合并 S_3 中的文字像素，得到完整的文字块（Fig. 4 (d)）。其中Fig. 4 (g)所示的扩展操作是基于广度优先搜索实现的。

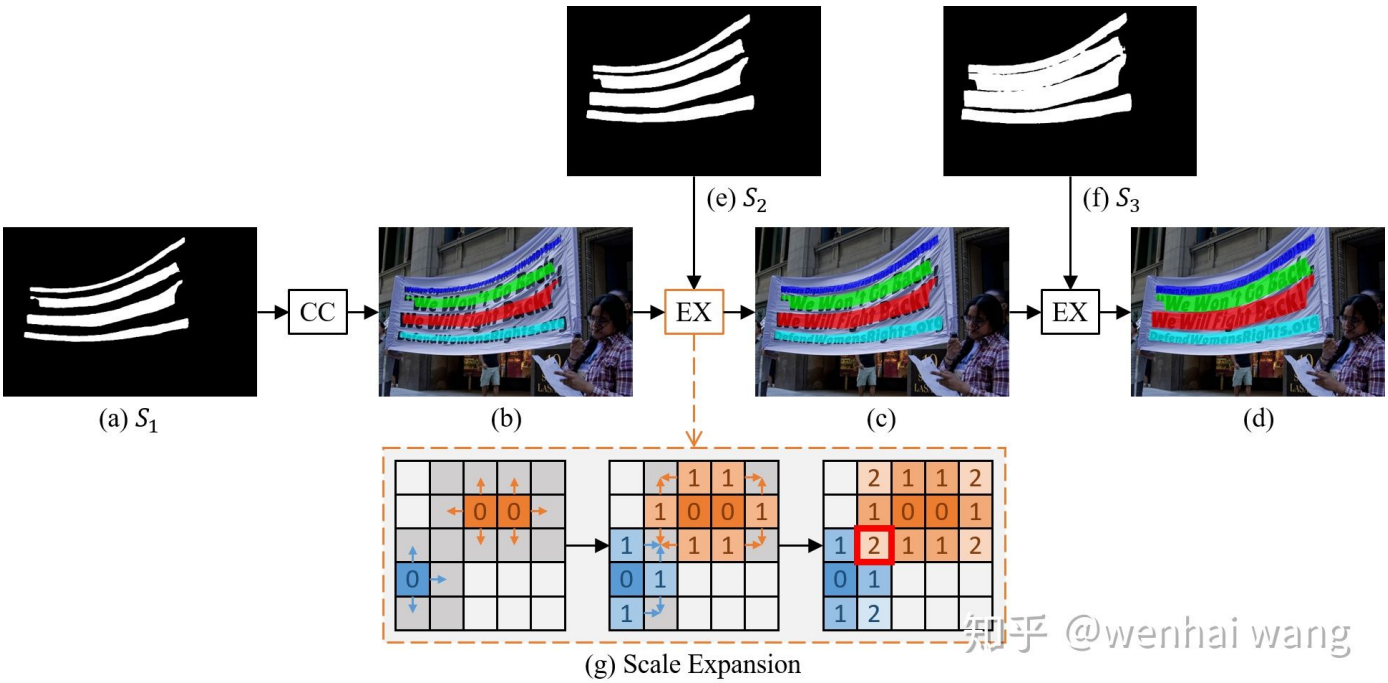

Fig. 4. 渐进扩展算法的过程。其中CC表示求连通区域，EX表示扩展操作。

到这里PSENet已经差不多讲完，接下来我们看看PSENet在不同数据集上的表现，如Table 1所示，PSENet在常规数据集ICDAR 2015和ICDAR 2017 MLT上的表现不弱于主流的检测算法，**但在弯曲文字数据集SCUT-CTW1500上的表现超过了之前最好的结果6.37%**。Fig. 5是PSENet在不同数据集上的一些结果。

Method	IC15				IC17-MLT			SCUT-CTW1500		
	P (%)	R (%)	F (%)	FPS	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
CTPN [27]	74.22	51.56	60.85	7.1	-	-	-	60.4*	53.8*	56.9*
SegLink [24]	74.74	76.50	75.61	-	-	-	-	42.3*	40.0*	40.8*
SSTD [9]	80.23	73.86	76.91	7.7	-	-	-	-	-	-
WordSup [10]	79.33	77.03	78.16	-	-	-	-	-	-	-
EAST [30]	83.27	78.33	80.72	6.52	-	-	-	78.7*	49.1*	60.4*
R^2CNN [13]	85.62	79.68	82.54	-	-	-	-	-	-	-
FTSN [2]	88.65	80.07	84.14	-	-	-	-	-	-	-
SLPR [31]	85.5	83.6	84.5	-	-	-	-	80.1	70.1	74.8
linkage-ER-Flow [1]	-	-	-	-	44.48	25.59	32.49	-	-	-
TH-DL [1]	-	-	-	-	67.75	34.78	45.97	-	-	-
TDN SJTU2017 [1]	-	-	-	-	64.27	47.13	54.38	-	-	-
SARI FDU RRPN v1 [1]	-	-	-	-	71.17	55.50	62.37	-	-	-
SCUT DLVClab1 [1]	-	-	-	-	80.28	54.54	64.96	-	-	-
Lyu et al. [20]	89.5	79.7	84.3	3.6	83.8	55.6	66.8	-	-	-
FOTS [18]	91.00	85.17	87.99	7.5	80.95	57.51	67.25	-	-	-
CTD+TLOC [19]	-	-	-	-	-	-	-	77.4	69.8	73.4
PSENet-4s (ours)	87.98	83.87	85.88	12.38	75.98	67.56	71.52	80.49	78.13	79.29
PSENet-2s (ours)	89.30	85.22	87.21	7.88	76.97	68.35	72.40	81.05	79.10	80.06
PSENet-1s (ours)	88.71	85.51	87.08	2.33	77.01	68.40	72.45	82.50	79.89	81.17

Table 1. PSENet在不同数据集上的表现

(a) ICDAR 2015 (b) ICDAR 2017 MLT (c) SCUT-CTW1500

Fig. 5. PSENet在不同数据集上的结果

更多详情请看原文，如有错误请指正。

编辑于 2018-06-10