文本检测–PSENet-1s

<Excerpt in index | 首页摘要>
Shape Robust Text Detection with Progressive Scale Expansion Network
KeyWords Plus:　　**CVPR2019**　　**Curved Text**　　**Face++**

- **paper**： new version paper
- **Github**: PSENet

<The rest of contents | 余下全文>

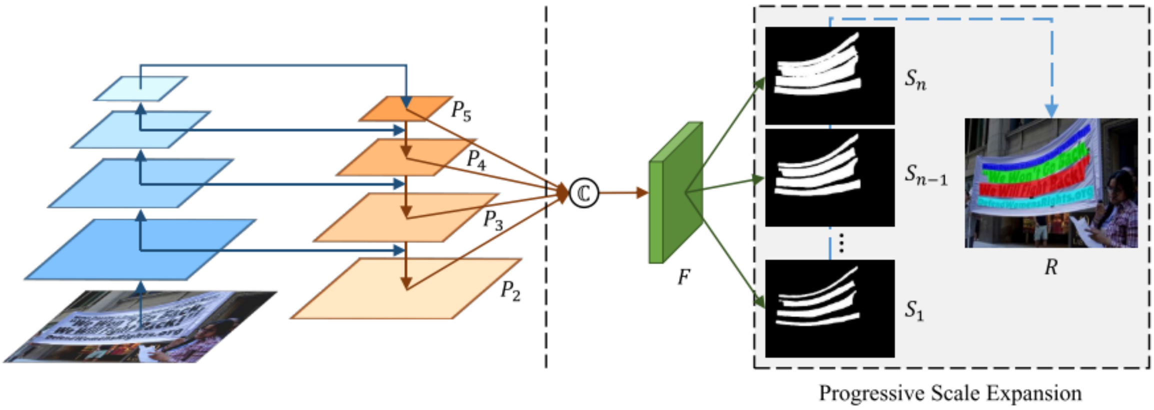
Introduction

PSENet 分好几个版本，最新的一个是**19年的CVPR**，这是一篇南京大学和face++合作的文章（好像还有好几个机构的人），19年出现了很多不规则文本检测算法，TextMountain、Textfield等等，不过为啥我要好好研究这个**（因为这篇文章开源了代码。。。）。**

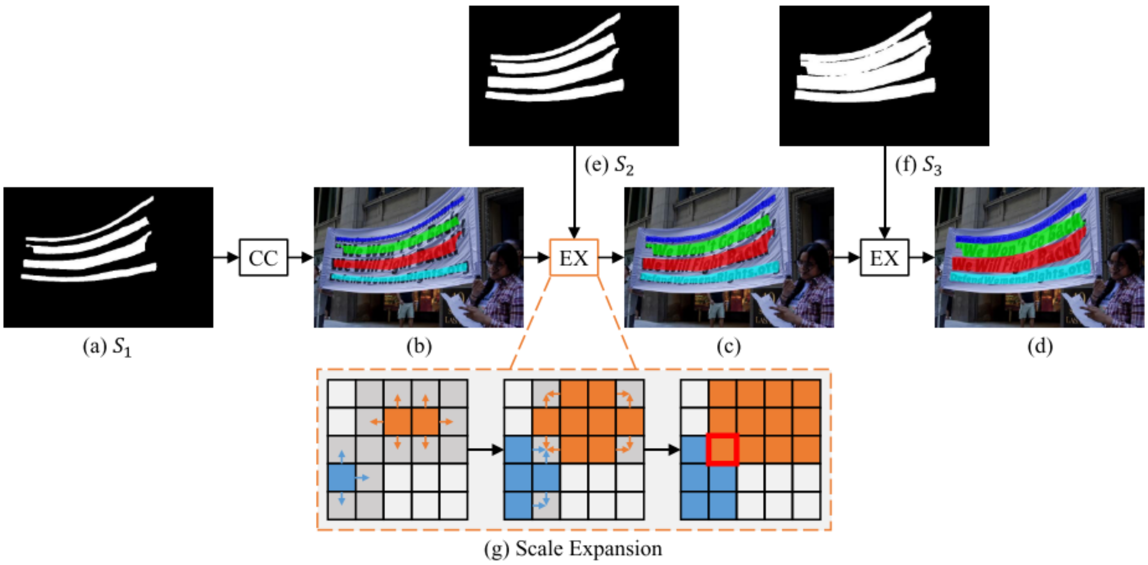
1、论文创新点

1、Propose a novel kernel-based framework, namely, **Progressive Scale Expansion Network (PSENet)**
2、Adopt a progressive scale expansion algorithm based on **Breadth-First-Search (BFS)**:
（1）、Starting from the kernels with **minimal scales** (instances can be distinguished in this step)
（2）、**Expanding their areas** by involving more pixels in larger kernels gradually
（3）、Finish- ing until the complete text instances (**the largest kernels)** are explored.

这个文章主要做的创新点大概就是**预测多个分割结果，分别是S1,S2,S3...Sn**代表不同的等级面积的结果，S1最小，基本就是文本骨架，Sn最大。然后在后处理的过程中，先用**最小的预测结果去区分文本，再逐步扩张成正常文本大小。。。**



2、算法主体



We firstly get four 256 channels feature maps **(i.e. P2, P3, P4, P5)** from the backbone. To further combine the semantic features from low to high levels, we fuse the four feature maps to get **feature map F with 1024 channels** via the function C(·) as:

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

先backbone下采样得到**四层的feature maps**，再通过**fpn**对四层feature分别进行**上采样2,4,8倍**进行融合得到输出结果。

如上图所示，网络有三个分割结果，分别是S1,S2,S3.首先利用最小的kernel生成的**S1来区分四个文本实例，然后再逐步扩张成S2和S3**

3、label generation

产生不同尺寸的S1….Sn需要**不同尺寸的labels**

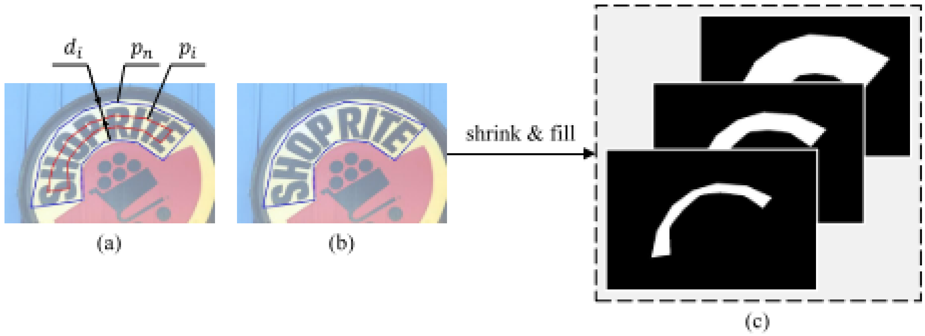


Figure 5. The illustration of label generation. (a) contains the annotations for d, p_i and p_n. (b) shows the original text instances. (c) shows the segmentation masks with different kernel scales.

不同尺寸的labels生成如上图所示，缩放比例可以用下面公式计算得出：

$$d_i = \frac{\text{Area}(p_n) \times (1 - r_i^2)}{\text{Perimeter}(p_n)},$$

这个d_i表示的是缩小后mask边缘与正常mask边缘的距离，缩放比例rate r_i可以由下面计算得出：

$$r_i = 1 - \frac{(1 - m) \times (n - i)}{n - 1},$$

m是最小mask的比例，n在m到1之间的值，成线性增加。

4、Loss Function

Loss 主要分为**分类的text instance loss和shrunk losses**，L是平衡这两个loss的参数。分类loss主要用了交叉熵和dice loss。

$$L = \lambda L_c + (1 - \lambda) L_s,$$

The dice coefficient **D(Si, Gi)** 被计算如下：

$$D(S_i, G_i) = \frac{2 \sum_{x,y} (S_{i,x,y} \times G_{i,x,y})}{\sum_{x,y} S_{i,x,y}^2 + \sum_{x,y} G_{i,x,y}^2},$$

L_s 被计算如下：

$$L_s = 1 - \frac{\sum_{i=1}^{n-1} D(S_i \cdot W, G_i \cdot W)}{n-1},$$

$$W_{x,y} = \begin{cases} 1, & if\ S_{n,x,y} \geq 0.5; \\ 0, & otherwise. \end{cases}$$

4、Datasets

SynthText

Contains about **800K** synthetic images.

TotalText

Newly-released benchmark for text detection. Besides horizontal and multi-Oriented text instances.The dataset is split into **training and testing sets with 1255 and 300 images**, respectively.

CTW1500

CTW1500 dataset **mainly consisting of curved text**. It consists of **1000 training images and 500 test images**. Text instances are annotated with polygons with **14 vertexes.**

ICDAR 2015

Icdar2015 is a commonly used dataset for text detection. It contains a **total of 1500 pictures**, 1000 of which are used for training and the remaining are for testing. The

ICDAR 2017 MLT

ICDAR 2017 MIL is a large scale multi-lingual text dataset, which includes **7200 training images, 1800 validation images and 9000 testing images.**

5、Experiment Results

Implementation Details

All the networks are optimized by using stochastic gradient **descent (SGD).**The **data augmentation** for training data is listed as follows: 1) the images are rescaled with ratio {0.5, 1.0, 2.0, 3.0} randomly; 2) the images are horizon- tally flipped and rotated in the range [−10◦, 10◦] randomly; 3) 640 × 640 random samples are cropped from the trans- formed images.

Method	Ext	Total-Text			
		P	R	F	FPS
SegLink [32]	-	30.3	23.8	26.7	-
EAST [43]	-	50.0	36.2	42.0	-
DeconvNet [2]	-	33.0	40.0	36.0	-
TextSnake [26]	✓	82.7	74.5	78.4	-
PSENet-1s	-	81.77	75.11	78.3	3.9
PSENet-1s	✓	84.02	77.96	**80.87**	3.9
PSENet-4s	✓	84.54	75.23	79.61	**8.4**

Total-Text

Method	Ext	CTW1500			
		P	R	F	FPS
CTPN [36]	-	60.4*	53.8*	56.9*	7.14
SegLink [32]	-	42.3*	40.0*	40.8*	10.7
EAST [43]	-	78.7*	49.1*	60.4*	**21.2**
CTD+TLOC [24]	-	77.4	69.8	73.4	13.3
TextSnake [26]	✓	67.9	85.3	75.6	-
PSENet-1s	-	80.57	75.55	78.0	3.9
PSENet-1s	✓	84.84	79.73	**82.2**	3.9
PSENet-4s	✓	82.09	77.84	79.9	8.4

CTW1500

Method	Ext	IC15			
		P	R	F	FPS
CTPN [36]	-	74.22	51.56	60.85	7.1
SegLink [32]	✓	73.1	76.8	75.0	-
SSTD [11]	✓	80.23	73.86	76.91	7.7
WordSup [13]	✓	79.33	77.03	78.16	-
EAST [43]	-	83.57	73.47	78.2	**13.2**
RRPN [28]	-	82.0	73.0	77.0	-
R^{2}CNN [16]	-	85.62	79.68	82.54	-
DeepReg [12]	-	82.0	80.0	81.0	-
PixelLink [4]	-	82.9	81.7	82.3	7.3
Lyu et al. [27]	✓	94.1	70.7	80.7	3.6
RRD [20]	✓	85.6	79.0	82.2	6.5
TextSnake [26]	✓	84.9	80.4	82.6	1.1
PSENet-1s	-	81.49	79.68	80.57	1.6
PSENet-1s	✓	86.92	84.5	**85.69**	1.6
PSENet-4s	✓	86.1	83.77	84.92	3.8

ICDAR 2015

Methods	P	R	F
PSENet (ResNet50)	73.7	68.2	70.8
PSENet (ResNet101)	74.8	68.9	71.7
PSENet (ResNet152)	75.3	69.2	72.2

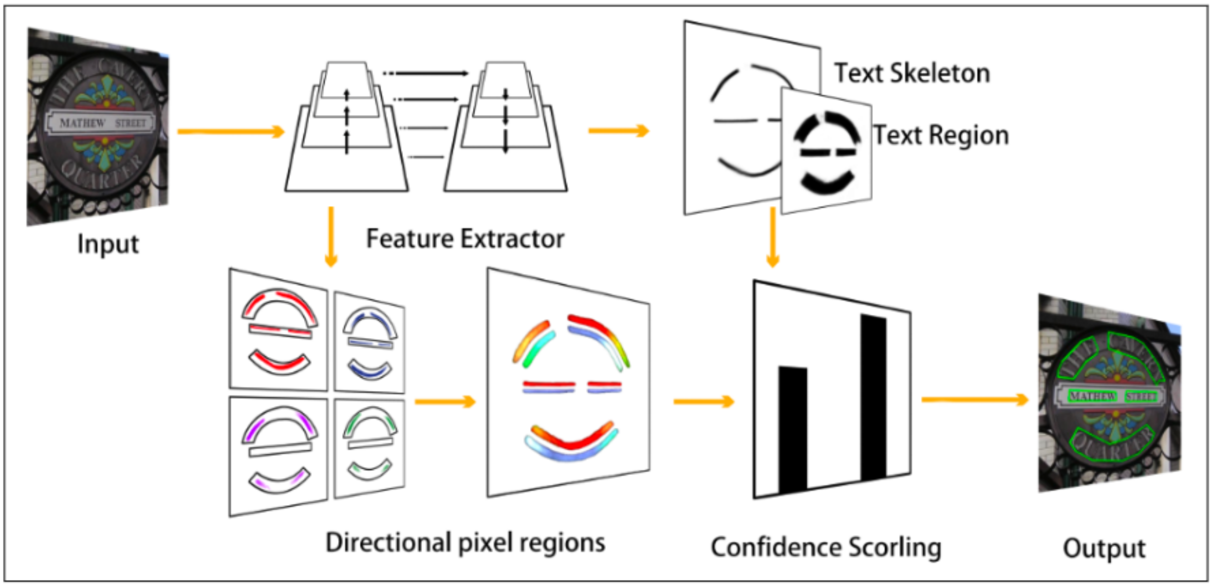
Table 1. Performance grows with deeper backbones on IC17-MLT. “P”, “R” and “F” represent the precision, recall and F-measure respectively.

IC17-MLT

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

6、Conclusion and Future work

个人观点：这个文章其实做的只是一件事情，就是用**预测得到的小的mask区分文本，然后逐渐扩张形成正常大小的文本mask**，个人最近发了一篇比较水的会议论文也是检测不规则文本的：TextCohesion: Detecting Text for ArbitraryShapes，其实本质是和这个文章是差不多的（我发之前还没看过这个文章，好像也没有被收录），不过算法主体是不一样的，我这个文章过几天也会挂到arxiv上，主要也是用小的mask区分文本实例，但是我不是进行扩展，我是讲除了**文本骨架外的文本像素给不同的方向预测，使得四周的文本像素对文本骨架有一个类似于“聚心力”的作用，最终形成一个文本实例。**pipeline如下（在total和ctw1500上实验指标蛮高，暂时第一）：



反馈与建议

- 微博：@柏林designer
- 邮箱：weijia_wu@yeah.net