FOTS: Fast Oriented Text Spotting with a Unified Network

汇报人　：苏静静
汇报时间：2020/10/25

目录

CONTENTS

PART 01

背景意义

背景意义

商汤18年1月发表，被CVPR2018接收，实现了端到端多方向文字的检测和识别。
论文地址：https://arxiv.org/pdf/1801.01671.pdf
代码地址：https://github.com/jiangxiluning/FOTS.PyTorch

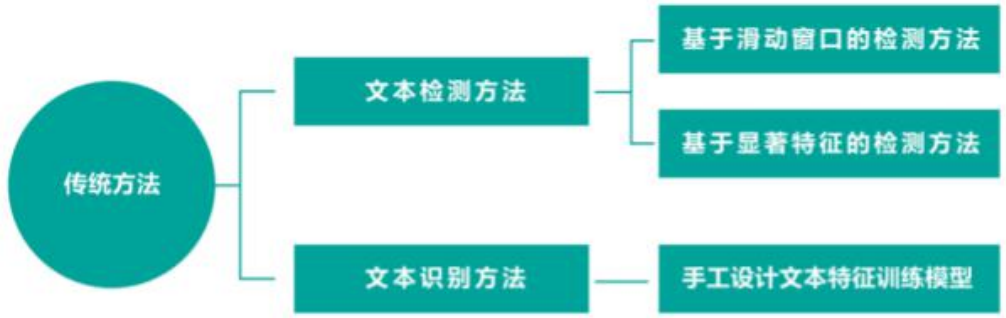
背景意义

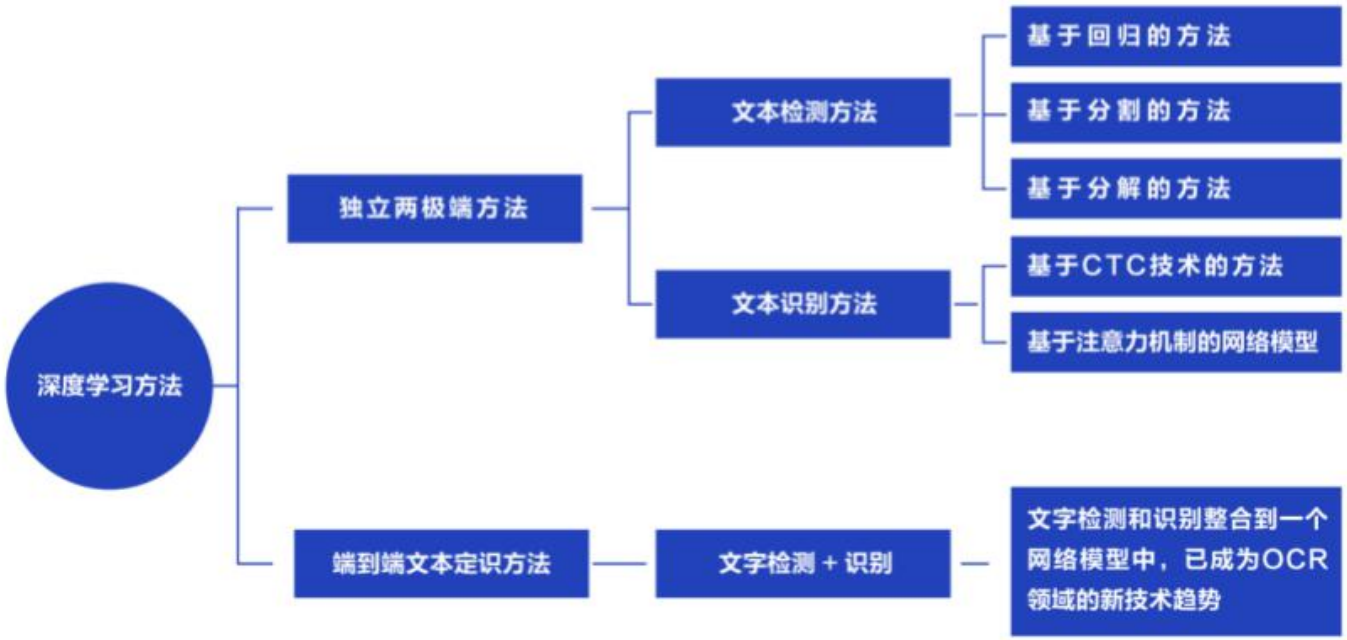
OCR 技术未来发展的三大方向主要包括一体化的端到端 OCR 模型、兼具高性能高效率的 OCR、从感知到认知的智能 OCR。

详细来说，构建一体化的端到端网络，同时对文字检测和识别进行训练，将成为 OCR 技术发展的重要趋势之一。端到端的网络设计不仅能够减少重复计算，又能够提高特征的质量，促进任务性能的改善。

-----《智能文字识别（OCR）能力测评与应用白皮书》

背景意义





-----《智能文字识别（OCR）能力测评与应用白皮书》

背景意义

论文动机

文本检测和识别是分为两项任务进行的，分开更容易实现，可是也带来了先验知识的损失。FOTS提出了端到端多方向文字的检测和识别方法，将文本检测和文本识别这两项监督进行融合互补。

主要贡献

1. 提出了FOTS，一个端到端的，可训练的，多方向的场景文本识别框架。
2. 提出一个新颖的RoIRotate操作，使得检测和识别统一到一个端到端的系统中。
3. 通过共享卷积特征，文本识别步骤计算开销基本没有，这也使得作者的系统分可以在实时的速度下运行。

PART 02

方法介绍

方法介绍

网络结构

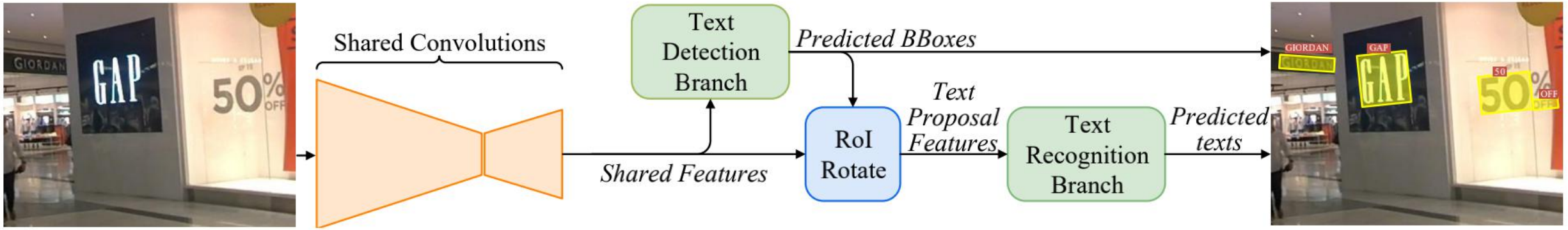


Figure 2: Overall architecture. The network predicts both text regions and text labels in a single forward pass.

01 共享特征

02 检测分支

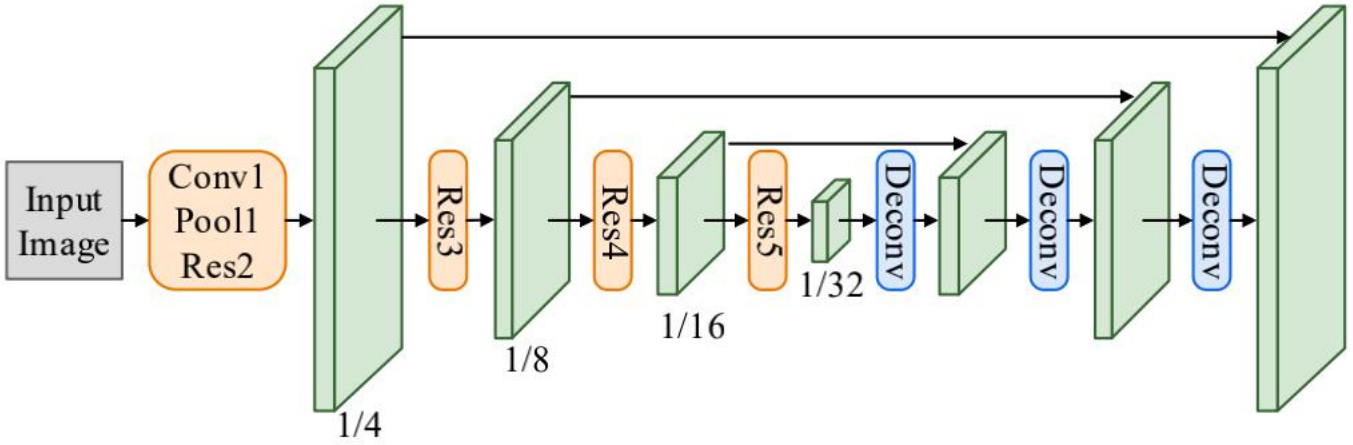
03 ROI Rotate

04 识别分支

方法介绍

共享特征

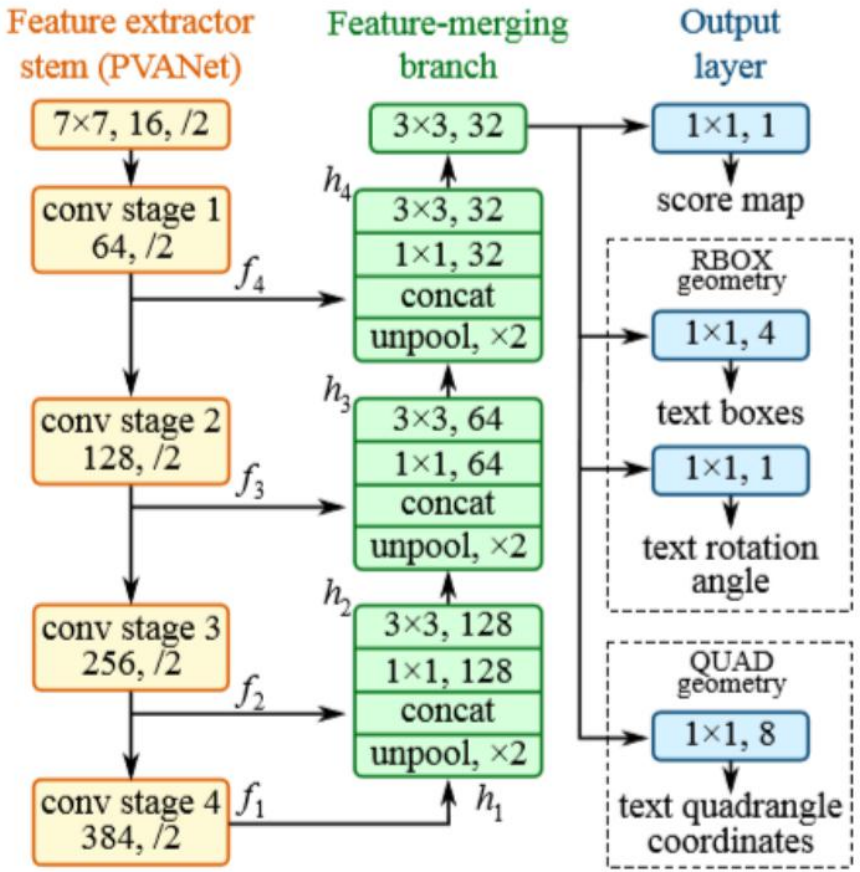
FOTS的基础网络结构为Resnet50，共享卷积层采用了类似U-net的卷积的共享方法。输出的特征图大小为原图的1/4。



方法介绍

检测分支

和EAST基本一样，输出6个channel的feature map：第一个channel表示每个pixel属于正样本（文本区域）的概率；后面的4个channel表示该pixel到包围它的bounding box的上、下、左、右边界的距离；最后一个channel预测相应bounding box的方向（旋转角度）。



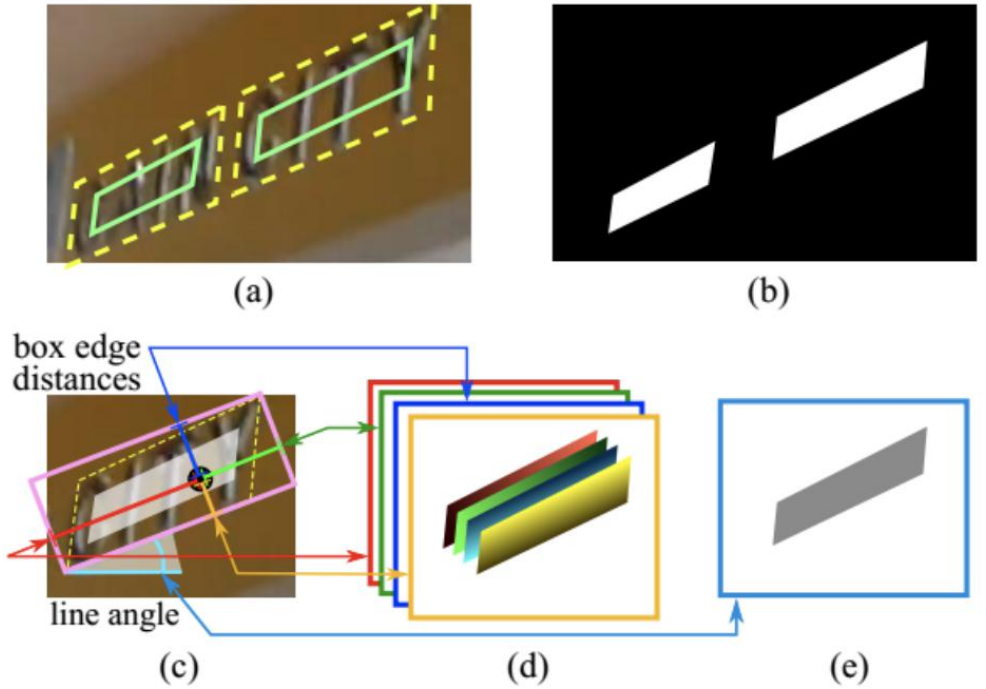


Figure 4. Label generation process: (a) Text quadrangle (yellow dashed) and the shrunk quadrangle (green solid); (b) Text score map; (c) RBOX geometry map generation; (d) 4 channels of distances of each pixel to rectangle boundaries; (e) Rotation angle.

方法介绍

检测损失

分类loss（cross entrop）和坐标回归loss（IOU loss）

$$
\begin{aligned}
L_{\text{cls}} &= \frac{1}{|\Omega|}\sum_{x\in\Omega}\mathrm{H}(p_x, p_x^*) \\
&= \frac{1}{|\Omega|}\sum_{x\in\Omega}(-p_x^*\log p_x - (1-p_x^*)\log(1-p_x))
\end{aligned}
\tag{1}
$$

其中, Ω是通过OHEM算法在score map上选取的正样本区域, |Ω|表示像素点数, H(px, px*)表示交叉熵。

$$
L_{\text{reg}} = \frac{1}{|\Omega|}\sum_{x\in\Omega}\mathrm{IoU}(\mathbf{R}_x, \mathbf{R}_x^*) + \lambda_\theta(1-\cos(\theta_x, \theta_x^*)) \tag{2}
$$

$$
L_{\text{detect}} = L_{\text{cls}} + \lambda_{\text{reg}} L_{\text{reg}} \tag{3}
$$

方法介绍

ROI Rotate

RoIRotate applies transformation on oriented feature regions to obtain axis-aligned feature maps.

方法介绍

ROI Rotate

RoI pooling：将大小不同的feature map 池化成大小相同的feature map。

RoIAlign：解决了ROI Pooling操作中两次量化造成的区域不匹配(mis-alignment)的问题，取消量化操作，使用双线性内插的方法获得坐标为浮点数的像素点上的图像数值,从而将整个特征聚集过程转化为一个连续的操作。

RRoI pooling：结合双线性插值求值，将旋转区域转为固定大小。

ROI Rotate：使得输出的特征长度可变，更适用文本识别任务。

方法介绍

ROI Rotate

变换分为两步。
一是通过text proposals的预测值或者text proposals真实值计算仿射变换矩阵M；
二是分别对检测分支得到的不同的文本框实施仿射变换，通过双线性插值获取水平feature map。

（训练的时候输入给recognize的部分，是gt的部分经过roirotate得到的feature map；预测的时候输入给recognize的部分是使用的east回归的出来的参数进行roirotate得到的feature map。所以说训练和预测阶段的处理有点不太一样。）

方法介绍

ROI Rotate

$$t_x = l * \cos\theta - t * \sin\theta - x \tag{4}$$

$$t_y = t * \cos\theta + l * \sin\theta - y \tag{5}$$

$$s = \frac{h_t}{t+b} \tag{6}$$

$$w_t = s * (l+r) \tag{7}$$

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= s \begin{bmatrix} \cos\theta & -\sin\theta & t_x\cos\theta - t_y\sin\theta \\ \sin\theta & \cos\theta & t_x\sin\theta + t_y\cos\theta \\ 0 & 0 & \frac{1}{s} \end{bmatrix} \tag{8}$$

M:仿射变换矩阵，包含旋转，缩放，平移
ht:仿射变换后的特征图的高度，实验中为8
wt:仿射变换后的特征图的宽度

(x,y):特征图中的点的位置

(t; b; l; r) :特征图中的点距离旋转的框的上下左右的距离

θ:检测框的角度

1.先计算ROI内的点(x,y)(在ROI内的坐标为(l, t))旋转至水平后要平移的距离 tx、ty。
2.计算目标高度ht与预测高度(t+b)的比值s，为保持ROI的高宽比，预测的长度(l+r)进行s倍的放缩。
3.平移(tx, ty)，缩放，旋转得到水平的ROI。
4.按照仿射矩阵M，可得目标ROI上每个像素点对应的原ROI上的坐标。

方法介绍

ROI Rotate

$$\begin{pmatrix} x_i^s \\ y_i^s \\ 1 \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \quad (9)$$

$$V_{ij}^c = \sum_n^{h_s} \sum_m^{w_s} U_{nm}^c k(x_{ij}^s - m; \Phi_x) k(y_{ij}^s - n; \Phi_y) \quad (10)$$

hs,ws表示ROIRotate的高和宽
Umn表示原ROI中(n, m, c)处的像素值
k()表示通用采样核(generic sampling kernel), 其参数φx, φy定义了采样方式, 此处取双线性插值采样。

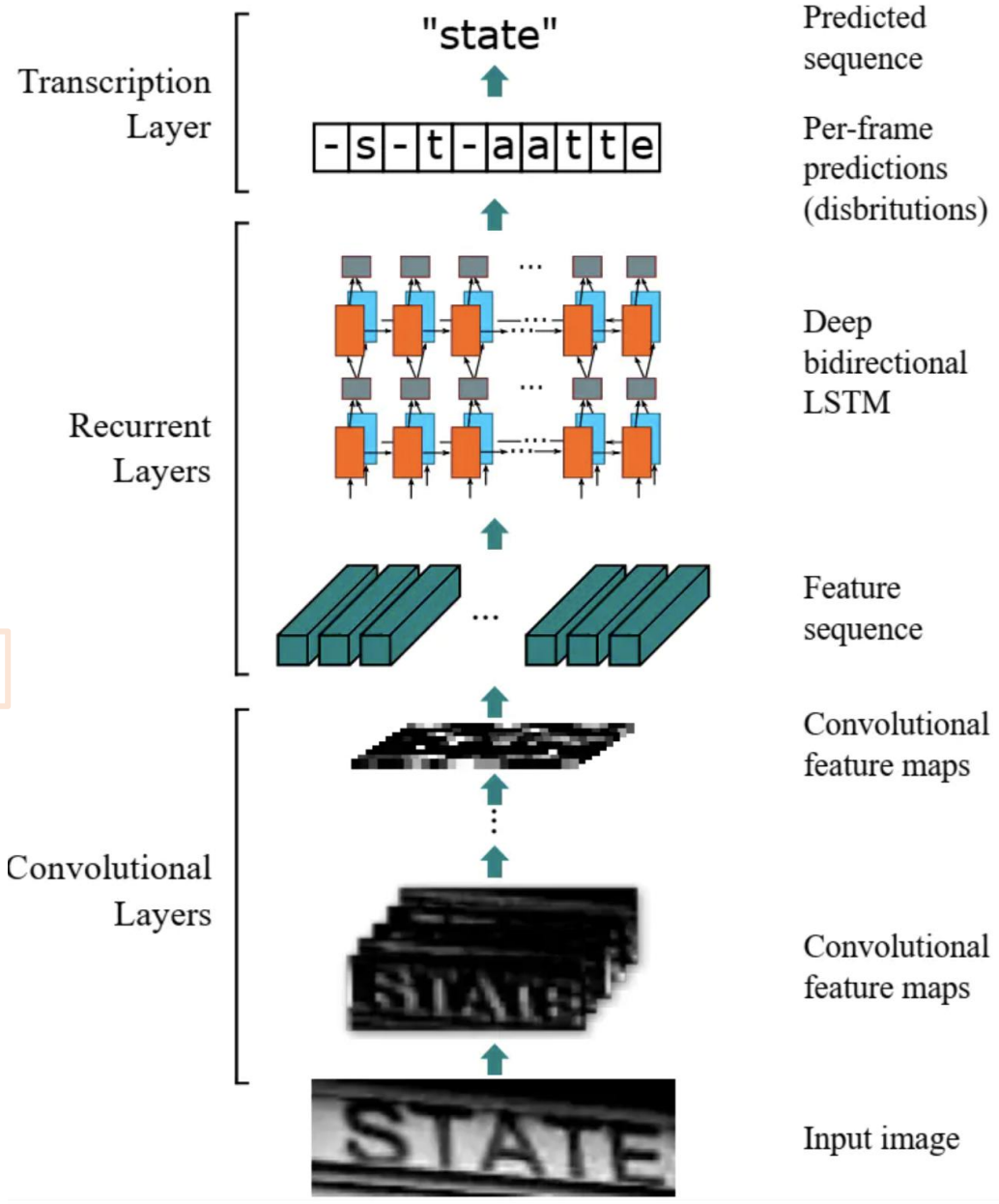
处理旋转好的目标ROI在宽的方向用0填充至固定宽度, 填充的部分不计算loss。

方法介绍

识别分支

传统文字识别:单字切割和分类任务

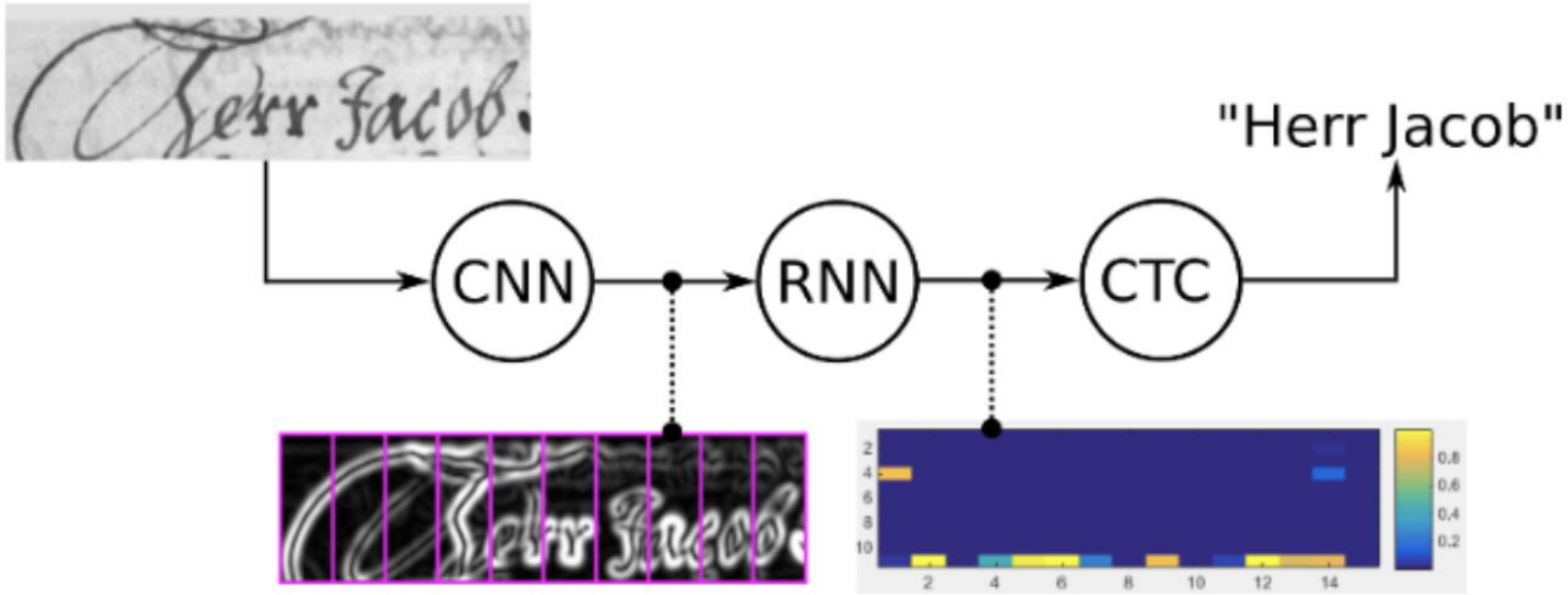
CRNN



方法介绍

识别分支

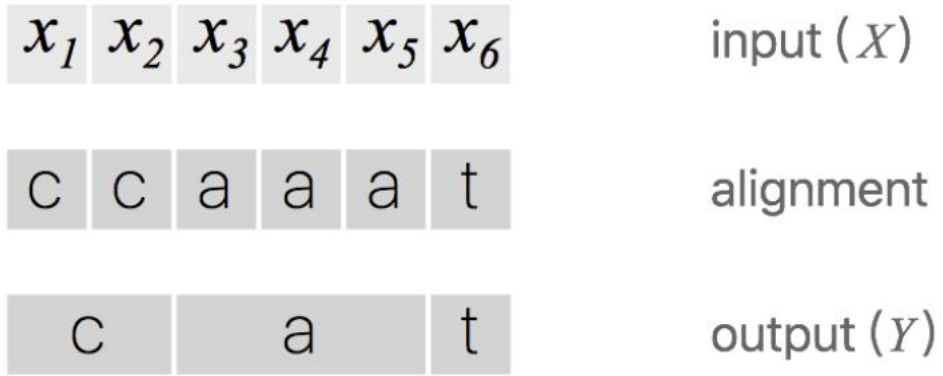
1.卷积层，使用CNN，作用是从输入图像中提取特征序列;
2.循环层，使用RNN，作用是预测从卷积层获取的特征序列的标签（真实值）分布;
3.转录层，使用CTC，作用是把从循环层获取的标签分布通过去重整合等操作转换成最终的识别结果(输出哪些像素范围对应的字符);



方法介绍

CTC

CTC算法如何将输入和输出进行对齐的?

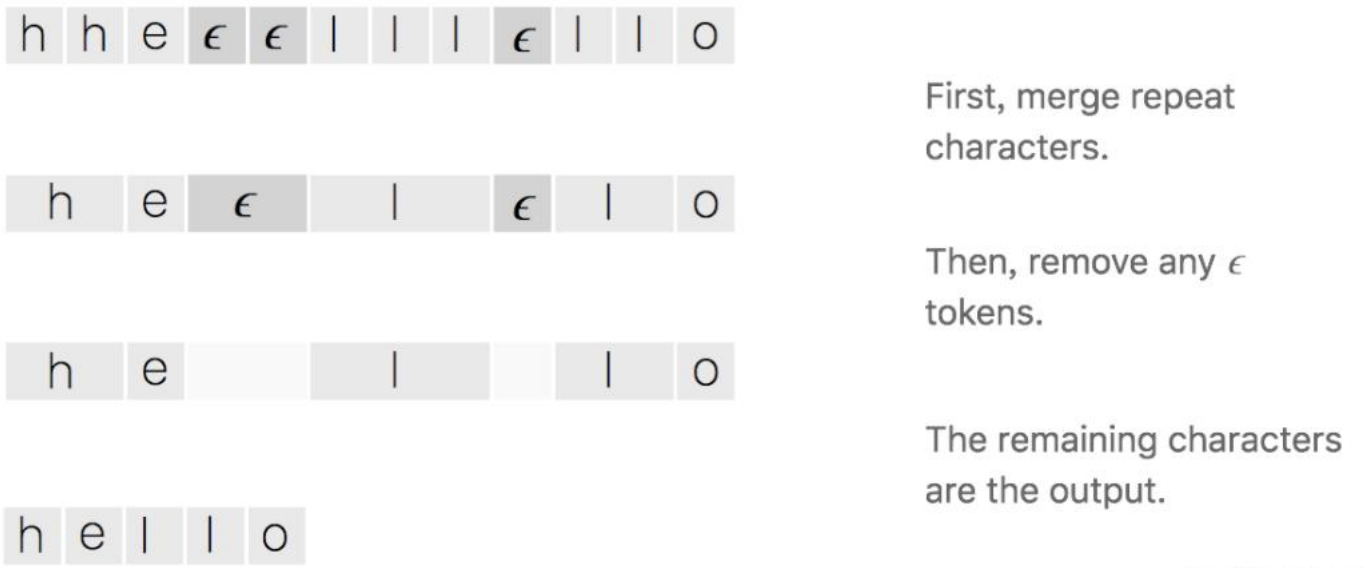


有两个问题：

1.通常这种对齐方式是不合理的。比如在语音识别任务中，有些音频片可能是无声的，这时候应该是没有字符输出的
2.对于一些本应含有重复字符的输出，这种对齐方式没法得到准确的输出。例如输出对齐的结果为[h , h , e , l , l , l , o] [h,h,e,l,l,l,o][h,h,e,l,l,l,o]，通过去重操作后得到的不是“hello”而是“helo”

方法介绍

CTC



CTC算法引入的一个新的占位符用于输出对齐的结果。这个占位符称为空白占位符，通常使用符号ϵ，这个符号在对齐结果中输出，但是在最后的去重操作会将所有的ϵ删除得到最终的输出。

方法介绍

识别损失

$$p(\boldsymbol{y}^*|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\boldsymbol{y}^*)} p(\pi|\boldsymbol{x}) \qquad (11) \qquad \boldsymbol{y}^* = \{y_1, ..., y_T\}, T \leqslant W$$

$$L_{\text{recog}} = -\frac{1}{N} \sum_{n=1}^{N} \log p(\boldsymbol{y}_n^*|\boldsymbol{x}) \qquad (12)$$

N是输入图像中文本区域的数量，yn*是识别标签，y*是ground truth标签序列

方法介绍

FOTS损失

$$L = L_{\text{detect}} + \lambda_{\text{recog}} L_{\text{recog}}$$

λrecog controls the trade-off be_x0002_tween two losses. λrecog is set to 1 in our experiments.

PART 03

实验结果

实验结果

Method	Detection			Method	End-to-End			Word Spotting		
	P	R	F		S	W	G	S	W	G
SegLink [43]	74.74	76.50	75.61	Baseline OpenCV3.0+Tesseract [26]	13.84	12.01	8.01	14.65	12.63	8.43
SSTD [13]	80.23	73.86	76.91	Deep2Text-MO [51, 50, 20]	16.77	16.77	16.77	17.58	17.58	17.58
WordSup [17]	79.33	77.03	78.16	Beam search CUNI+S [26]	22.14	19.80	17.46	23.37	21.07	18.38
RRPN [39]	83.52	77.13	80.20	NJU Text (Version3) [26]	32.63	-	-	34.10	-	-
EAST [53]	83.27	78.33	80.72	StradVision_v1 [26]	33.21	-	-	34.65	-	-
NLPR-CASIA [15]	82	80	81	Stradvision-2 [26]	43.70	-	-	45.87	-	-
R^2CNN [25]	85.62	79.68	82.54	TextProposals+DictNet [7, 19]	53.30	49.61	47.18	56.00	52.26	49.73
CCFLAB_FTSN [4]	88.65	80.07	84.14	HUST_MCLAB [43, 44]	67.86	-	-	70.57	-	-
Our Detection	88.84	82.04	85.31	Our Two-Stage	77.11	74.54	58.36	80.38	77.66	58.19
FOTS	91.0	85.17	87.99	FOTS	81.09	75.90	60.80	84.68	79.32	63.29
FOTS RT	85.95	79.83	82.78	FOTS RT	73.45	66.31	51.40	76.74	69.23	53.50
FOTS MS	**91.85**	**87.92**	**89.84**	FOTS MS	**83.55**	**79.11**	**65.33**	**87.01**	**82.39**	**67.97**

Table 2: Comparison with other results on ICDAR 2015 with percentage scores. "FOTS MS" represents multi-scale testing and "FOTS RT" represents our real-time version, which will be discussed in Sec. 4.4. "End-to-End" and "Word Spotting" are two types of evaluation protocols for text spotting. "P", "R", "F" represent "Precision", "Recall", "F-measure" respectively and "S", "W", "G" represent F-measure using "Strong", "Weak", "Generic" lexicon respectively.

实验结果

Method	Precision	Recall	F-measure
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 [39]	71.17	55.50	62.37
SCUT_DLVClab1 [1]	80.28	54.54	64.96
Our Detection	79.48	57.45	66.69
FOTS	80.95	57.51	67.25
FOTS MS	**81.86**	**62.30**	**70.75**

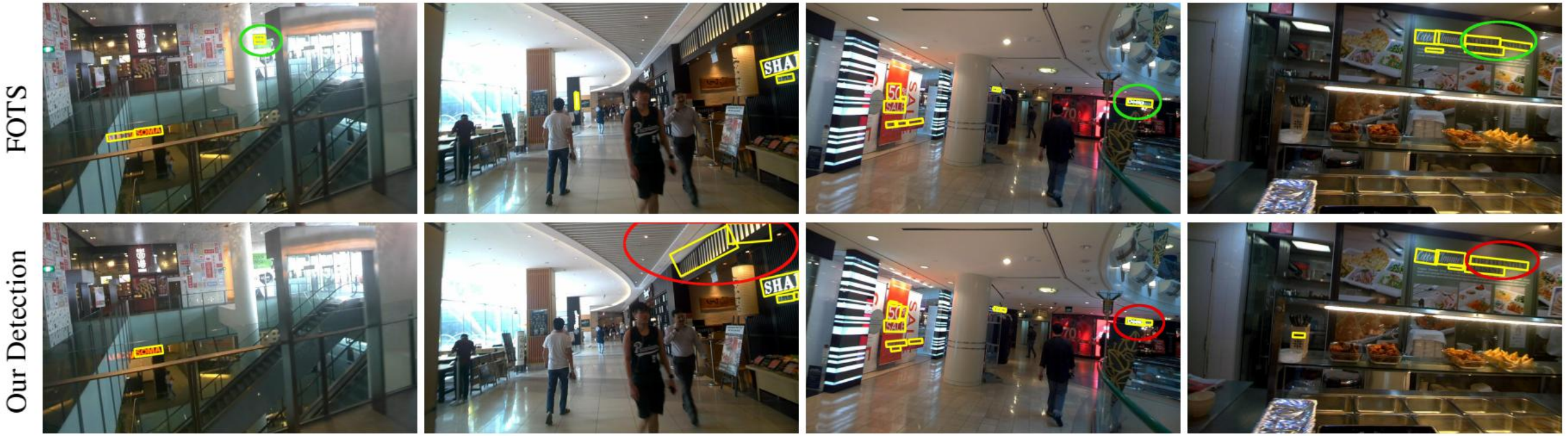
Table 3: Comparison with other results on ICDAR 2017 MLT scene text detection task.

实验结果

Method	Detection		Method	End-to-End			Word Spotting		
	IC13	DetEval		S	W	G	S	W	G
TextBoxes [34]	85	86	NJU Text (Version3) [27]	74.42	-	-	77.89	-	-
CTPN [49]	82.15	87.69	StradVision-1 [27]	81.28	78.51	67.15	85.82	82.84	70.19
R^2CNN [25]	79.68	87.73	Deep2Text II+ [51, 20]	81.81	79.47	76.99	84.84	83.43	78.90
NLPR-CASIA [15]	86	-	VGGMaxBBNet(055) [20, 19]	86.35	-	-	90.49	-	76
SSTD [13]	87	88	FCRNall+multi-filt [10]	-	-	-	-	-	84.7
WordSup [17]	-	90.34	Adelaide_ConvLSTMs [32]	87.19	86.39	80.12	91.39	90.16	82.91
RRPN [39]	-	91	TextBoxes [34]	91.57	89.65	83.89	93.90	91.95	85.92
Jiang *et al.* [24]	89.54	91.85	Li *et al.* [33]	91.08	89.81	84.59	94.16	92.42	**88.20**
Our Detection	86.96	87.32	Our Two-Stage	87.84	86.96	80.79	91.70	90.68	82.97
FOTS	88.23	88.30	FOTS	88.81	87.11	80.81	92.73	90.72	83.51
FOTS MS	**92.50**	**92.82**	FOTS MS	**91.99**	**90.11**	**84.77**	**95.94**	**93.90**	87.76

Table 4: Comparison with other results on ICDAR 2013. "IC03" and "DetEval" represent F-measure under ICDAR 2013 evaluation and DetEval evaluation respectively.

实验结果



(a) Miss (b) False (c) Split (d) Merge

参考

1.https://blog.csdn.net/zl535320706/article/details/80603187

2.https://blog.csdn.net/sinat_30822007/article/details/89294068

3.https://blog.csdn.net/qq_14845119/article/details/84635847

4.https://blog.csdn.net/ft_sunshine/article/details/95381610?utm_medium=distribute.pc_relevant.none-task-blog-title-4&spm=1001.2101.3001.4242

5.https://zhuanlan.zhihu.com/p/51090538

6.https://zhuanlan.zhihu.com/p/106759252

THANKS FOR YOUR WATCHING

汇报人：苏静静