

场景文字处理

华中科技大学

$$\rho := \frac{1 + \sqrt{-3}}{2}$$



目 录

第一章 文字识别 (Scene Text Recognition)	1
1.1 文字识别方法介绍	1
1.1.1 DAN (AAAI2020)	1
1.1.2 GTC (AAAI2020)	3
1.1.3 TextScanner (AAAI2020)	4
1.1.4 SCATTER (CVPR2020)	5
1.1.5 SRN (CVPR2020)	6
第二章 文字检测 (Scene Text Detection)	10
2.1 文字检测方法介绍	10
2.1.1 MATI	10
2.1.2 TextTubes	12
2.1.3 DB (AAAI2020)	13
2.1.4 RelaText	14
2.1.5 PuzzleNet	15
2.1.6 DRRG (CVPR2020)	17
第三章 端到端文字识别 (Scene Text Spotting)	20
3.1 各种任意形状文本端到端识别方法介绍	21
3.1.1 MaskTextSpotter (ECCV2018)	21
3.1.2 TextDragon (ICCV2019)	23
3.1.3 CharNet (ICCV2019)	24
3.1.4 MaskRoI (ICCV2019)	26
3.1.5 Boundary (AAAI2020)	27
3.1.6 TextPerceptron (AAAI2020)	28
3.1.7 ABCNet (CVPR2020)	30

Bibliography**32**

1

文字识别 (Scene Text Recognition)

近期的文字识别的文章主要围绕两个主题进行展开：1) 如何得到更准的对齐的字符特征；2) 如何高效地利用文本的语义特征进行识别。

DAN[24], GTC[6] 以及 TextScanner[22] 都是针对第一个问题进行改进。DAN 认为注意力模型中因需要依赖前一时刻预测结果从而存在累积误差使得模型的 attention 难以对齐，文中通过解耦注意力模块和文本语义预测模块使得 attention 更为准确。GTC 认为基于 CTC 的文本识别器中，因为 CTC 的切割使得特征难以对齐。文中通过注意力模块来指导 CTC 的训练，从而使得特征在一定程度对齐。TextScanner 通过预测字符的阅读顺序从而缓解字符定位问题。

SCATTER[10] 和 SRN[2] 都是针对第二个问题进行改进。SCATTER 通过在视觉特征和语义特征之间迭代地使用注意力机制从而动态选择视觉和语义特征来加强分类效果。SRN 通过并行化注意力模块和语义模块，使得当前帧在预测时能够获得前向和后向的所有视觉以及语义特征，从而加强分类效果。

1.1 文字识别方法介绍

1.1.1 DAN (AAAI2020)

DAN[24] 的主要是解决基于 attention 机制的识别器中因需要依赖先前预测结果（存在错误累积）所导致的注意力不对齐的问题。文章中通过解耦注意力机制和语义模型来解决该问题。解耦过程如图1-1所示。

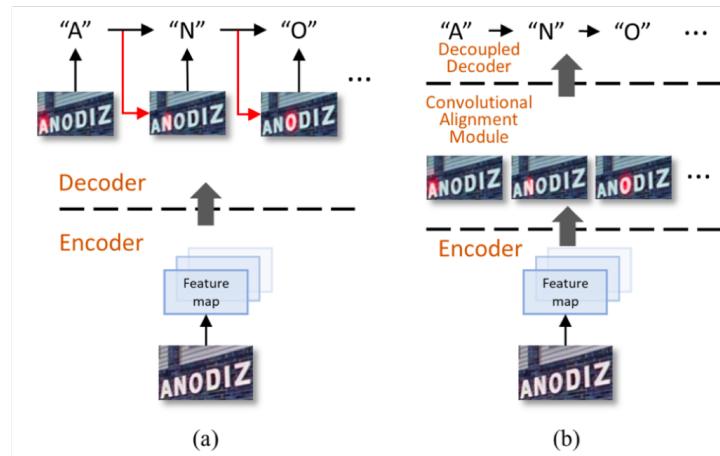


图 1-1 DAN 解耦过程：(a) 之前的基于注意力机制的方法将注意力模块和语义推理模块放在一起；(b) DAN 中先预测注意力，然后进行语义模块的预测。

1.1.1.1 DAN 的网络结构

DAN 的网络结构如图1-2所示：Decoupled Text Decoder 是基于 GRU 的，其过程和其他文字识别器的一致；CAM 用于预测每一步的 attention map。

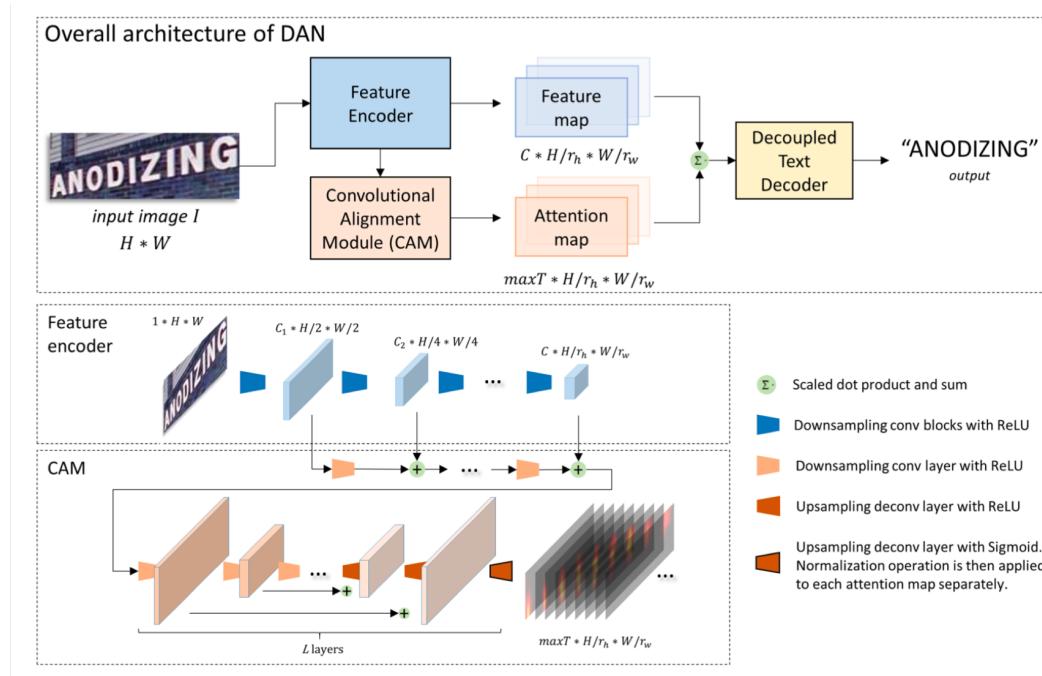


图 1-2 DAN 网络结构。

1.1.2 GTC (AAAI2020)

GTC[6] 的主要是解决基于 CTC 的文字识别器中帧数不对齐的问题，比如字符'H'由于分帧的原因会被识别为'I'。文章中通过基于注意力机制的识别 head 的监督来使得特征在一定程度上对齐。在测试阶段为了保证效率，只使用基于 CTC 的识别 head。另外，CNN 特征经过分帧后，相邻帧之间在视觉上具有一定的相似性，为了使得特征能够进一步对齐（理想情况下一帧的特征代表一个字符的特征），作者使用 GCN 来建模帧之间的关系，融合后得到对齐的帧。

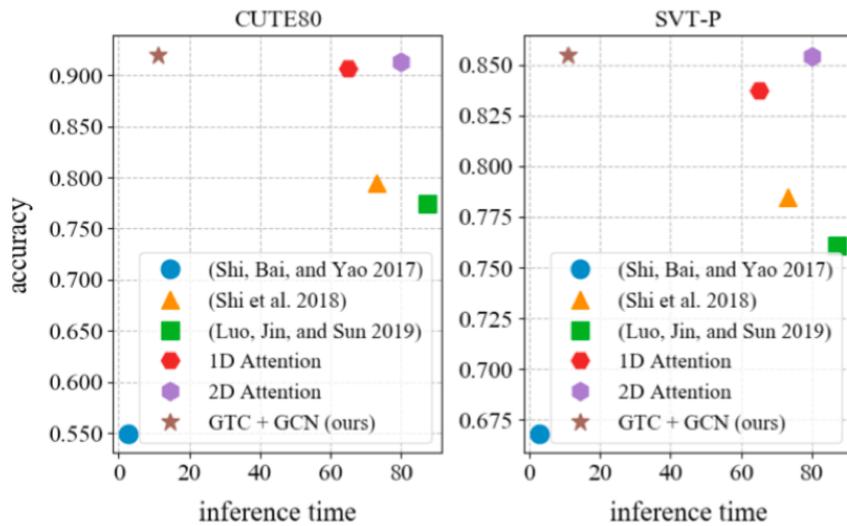


图 1-3 GTC 准确率和效率的关系，x 轴代表 ms/image

1.1.2.1 GTC 的网络结构

GTC 的网络结构如图1-4所示：网络的整体结构和 Aster 类似，不同的是，识别的 head 是基于 CTC 的，基于注意力机制的识别 head 在训练时起到辅助作用，使得特征能够起到一定的对齐作用。为保证效率，测试过程只使用 CTC 进行解码，最终的精度-效率对比如图1-3所示。

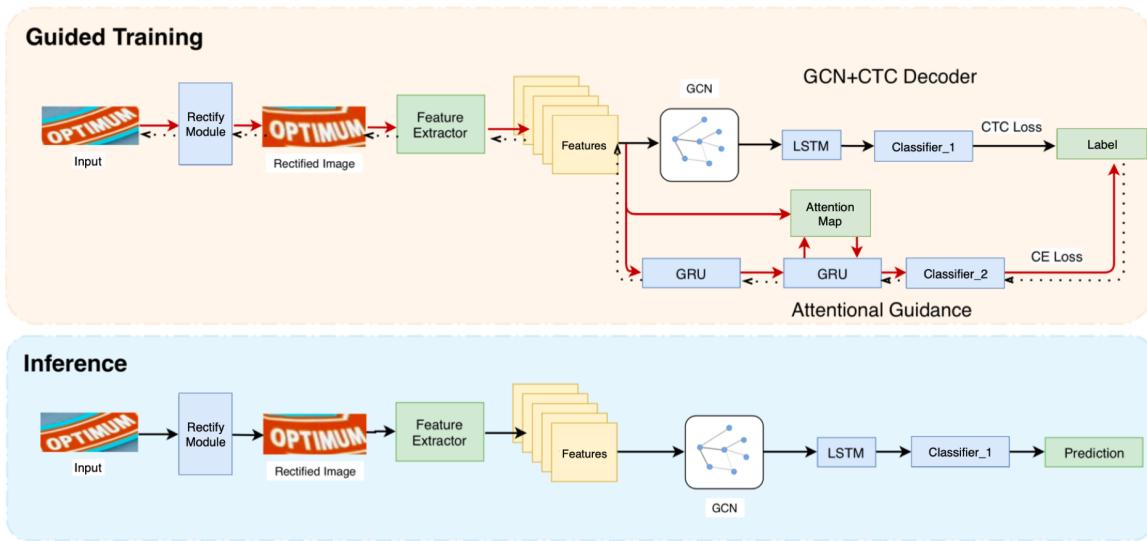


图 1-4 GTC 网络结构。

1.1.3 TextScanner (AAAI2020)

TextScanner[22]主要是解决基于分割的文本识别器中字符分割不准以及基于注意力机制的文本识别器中注意力发散的问题。目的仍然是寻找精准的字符定位从而使得特征对齐。如图1-5：基于注意力机制和分割的方法对字符的定位都存在一些问题，文章中通过预测单词的阅读顺序来解决该问题。

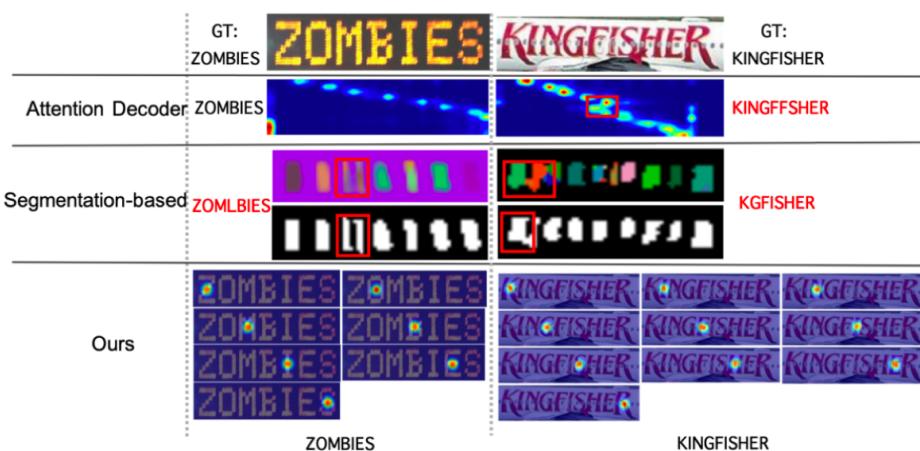


图 1-5 TextScanner 问题出发点：Attention Decoder 中注意力容易发散；基于分割的方法中因为阈值问题，容易存在欠分割或过分割问题。

1.1.3.1 TextScanner 的网络结构

TextScanner 的网络结构如图1–7所示：整体分为三部分，1) 字符分割模块，每个像素对类别进行预测；2) 字符顺序分割模块，其中 N 代表最大长度，模块进行 N 分类，预测每个像素属于哪一时间时刻；3) 字符定位模块用于定位每个字符。

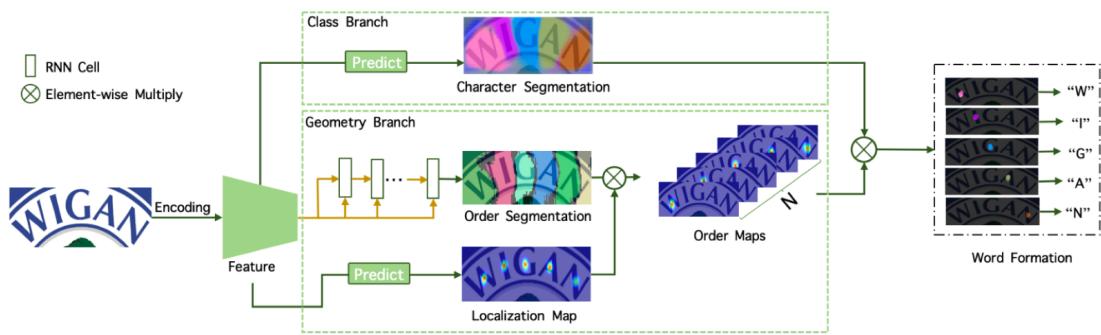


图 1–6 TextScanner 网络结构。

1.1.4 SCATTER (CVPR2020)

SCATTER[10] 的主要研究目的是让网络不断迭代地选择图像的视觉特征（CNN 的特征）和语义特征（RNN 建模出的语义特征）来强化模型的特征提取能力。而特征选择的过程通过注意力机制来完成。

1.1.4.1 SCATTER 的网络结构

SCATTER 的网络结构如图1–7所示，网络的整体框架和 Aster 一致，不同之处在于：SCATTER 在 Visual features 和 Contextual features 之间加入了多层特征选择模块进行特征的优化，并且在 Visual features 中加入了 CTC 进行监督学习。特征选择模块网络结构如图1–8所示。

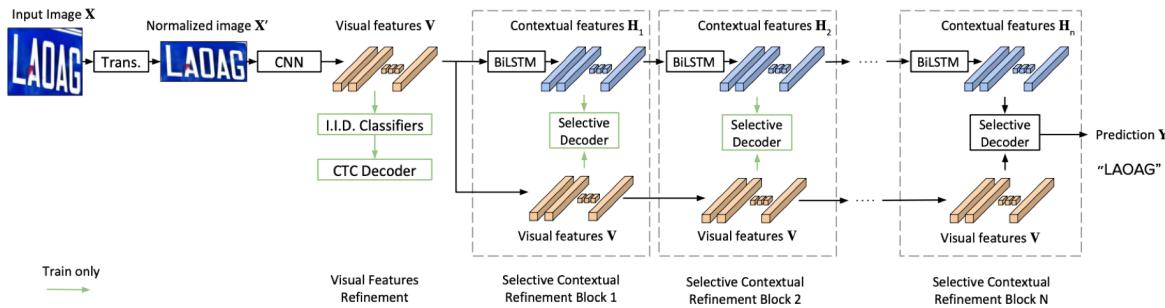


图 1-7 SCATTER 网络结构。

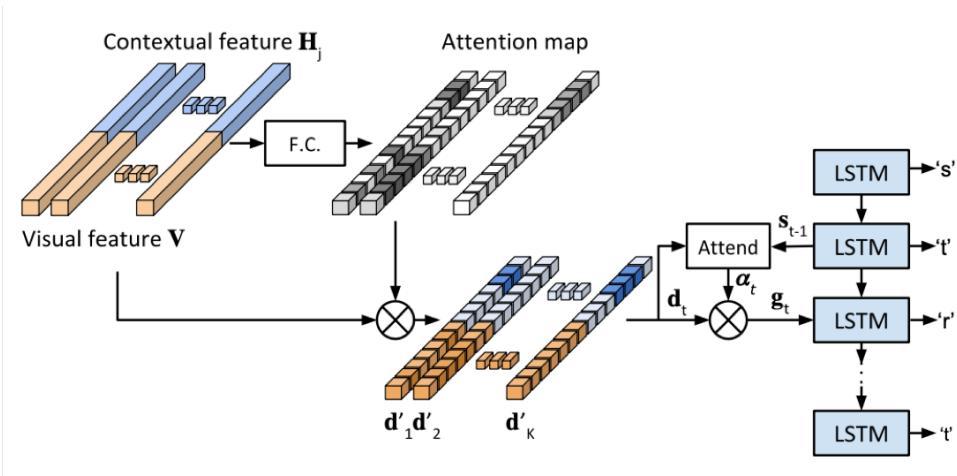


图 1-8 SCATTER 特征选择模块。

1.1.5 SRN (CVPR2020)

SRN[2] 的主要出发点是：1) 在文字识别中，由于光照，旋转等因素的影响，仅仅依靠图像的视觉特征极易引起单词中某个字符预测错误。对于这些易错的字符，如果能够利用单词的语义信息，那么将会极大地降低该字符的预测错误率。如图1-9所示，如果仅仅观察每个字符的视觉特征 (b)，某些字符容易预测错误，结合上下文语义信息能够缓解因视觉特征混淆而引起的错误预测。2) 文字识别中基于注意力机制的识别器中 [20]，注意力模块的输出大多是串行的，注意力模块中当前时刻的预测非常依赖于前一时刻的输出，导致模型难以并行处理。为解决模型的效率问题，提出并行的注意力模块。

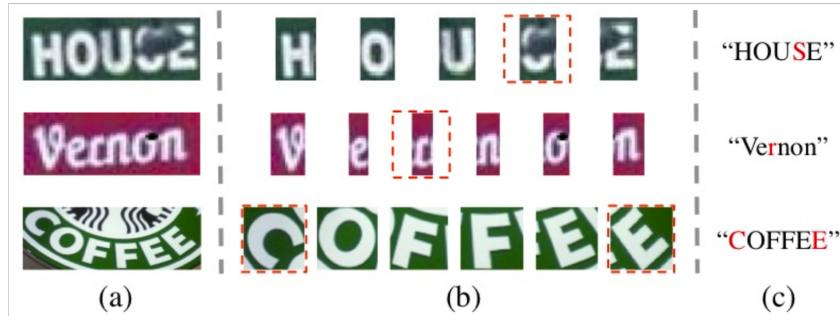


图 1-9 单词中容易预测错误的字符案例：(a) 表示原图；(b) 表示字符，红色标注为易错字符；(c) 为利用上下文语义信息预测结果。

1.1.5.1 SRN 的网络结构

SRN 的网络结构如图 1-10 所示：1) 整体网络的 backbone 为 FPN 网络，用于提取图像的视觉特征；2) 并行的视觉注意力模块 (PAVM) 用于定位每个字符；3) 全局的语义推理模块 (GSRM) 用于利用语义上下文来预测字符。

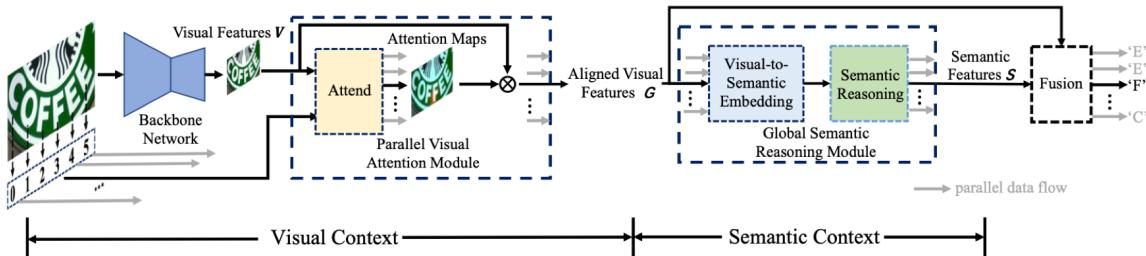


图 1-10 SRN 网络框架图。

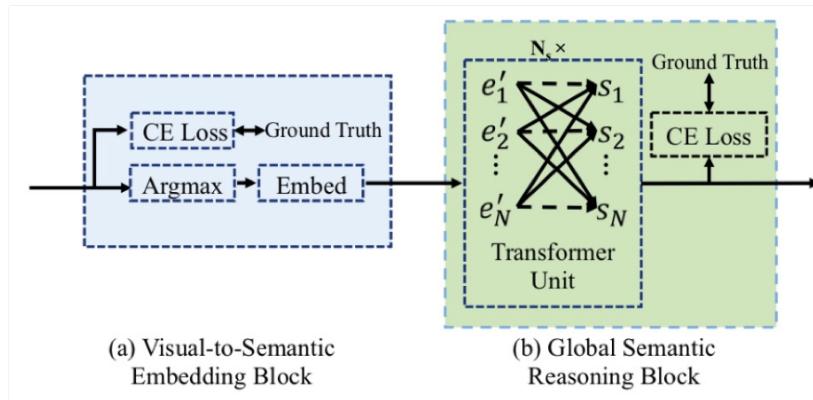


图 1-11 GSRM 模块结构。

1.1.5.2 SRN 并行以及语义推理分析

SRN 的主要优势在于利用单词的上下语义来预测困难字符，而要实现这一特性的技术难点却在于如何并行化处理序列预测。因为在预测 t 时刻的字符时，需要其他所有时刻的特征。因此，SRN 中如何将各个模块并行化是该方法技术的重点。表1-1详细地对比了 SRN 中并行化与 Aster 中串行化的区别。

对于并行化处理方面，SRN 的主要改进在于两大方面：1) 将注意力模型并行化。传统的注意力模型依赖于前一时刻的隐藏状态 h_{t-1} (在注意力模块中， h_{t-1} 作为 key 来获取相似度) 从而无法并行化。SRN 中将传统注意力模型中的 key 改为 O_t (代表字符顺序，第一个字符为 0，第二个字符为 1) 的 Embedding 特征，从而实现并行化；2) 将语义模型并行化。先前的文字识别算法中利用 rnn 来对文字语义进行建模，而 rnn 中以前一时刻的预测结果的 Embedding 特征 $f_y(y_{t-1})$ 作为输入，从而无法并行化。SRN 中将 e'_t (以 g_t 为输入的预测结果的 Embedding 向量) 来代替，从而实现并行化。另外，通过 Transformer Unit 来获取多路径的语义信息，能够考虑当前时刻的前向以及后向的所有语义信息。

表 1-1 SRN 中 PVAM, GSRM 模块和 Aster 的串行注意力机制与串行语义模块的区别。其中 TU 为 Transformer Unit, v_i 为视觉特征, h_i 为隐藏状态, y_t 为字符 label, O_t 为阅读顺序。

Module	Aster	SRN
Attention	$e_{t,i} = W_e^T \tanh(W_h h_{t-1} + W_v v_i)$ $\alpha_{t,i} = \exp(e_{t,i}) / \sum_{i'=1}^n \exp(e_{t,i'})$ $g_t = \sum_{i=1}^n \alpha_{t,i} v_i$	$e_{t,i} = W_e^T \tanh(W_o f_o(O_t) + W_v v_i)$ $\alpha_{t,i} = \exp(e_{t,i}) / \sum_{i'=1}^n \exp(e_{t,i'})$ $g_t = \sum_{i=1}^n \alpha_{t,i} v_i$
Semantic Reasoning	$(x_t, h_t) = \text{rnn}(h_{t-1}, (g_t, f_y(y_{t-1})))$	$e'_t = f_e(\text{softmax}(f_g(g_t)))$ $s_t = \text{TU}(e'_0 \dots e'_{t-1} e'_{t+1} \dots e'_n)$
Fusion		$z_t = \text{sigmoid}(W_z[g_t, s_t])$ $f_t = z_t g_t + (1 - z_t) s_t$
Classification	$p(y_t) = \text{softmax}(W x_t + b)$	$p(y_t) = \text{softmax}(W f_t + b)$

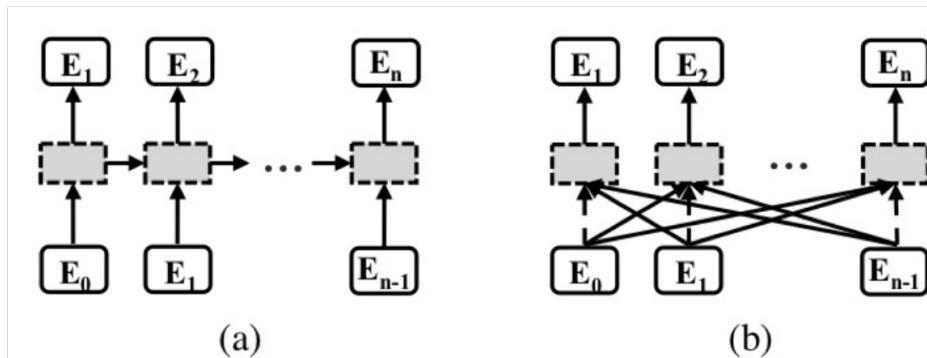


图 1-12 a) RNN 中建模单词语义模型；b) SRN 中建模单词语义模型能够考虑各个路径的信息，避免单向的错误累积。

2

文字检测 (Scene Text Detection)

近期的文字检测算法主要是解决曲形文本的检测问题。MATI[7] 和 TextTubes[19] 都是通过直接预测文本的边界来描述曲形文本。不同之处在于 MATI 是直接在全图预测边界点 (single stage)，而 TextTubes[19] 是在 MaskRCNN 的基础上将 mask head 换为预测文本的属性来描述曲形文本。

DB[9] 是基于分割的方法，意在解决文本检测的效率问题。

随后，RelaText[16]，PuzzleNet[11] 以及 DRRG[29] 都是自下而上的方法：都是首先预测文本的组件，然后通过 GCN 来预测文本组件之间的连接关系来完成检测任务。这些基于图卷积的方法不同之处在于 graph 的构建方式不同。

2.1 文字检测方法介绍

2.1.1 MATI

MATI[7] 主要是解决任意形状文本检测的问题。现有能够检测曲形文本的检测器大都是基于分割的方法，如 Textfield[28]、TextSnake[14]、CRAFT[1]、PSENet[25] 以及 MaskRCNN[5]。但是基于分割方法往往需要预测文本区域的几何属性，结合复杂的后处理过程得到任意形状的文本区域，使得检测器后处理不够高效。基于 anchor box 回归的方法虽然简单高效，但是任意形状文本难以使用包围盒进行描述。基于此，文章中提出使用直接回归的方法来检测任意形状文本。

2.1.1.1 MATI 的网络结构

MATI 的网络结构如图2-1所示：网络的整体结构基于 SSD，不同的是该方法中不需要 anchor box，整体和 east 一样进行密集预测。

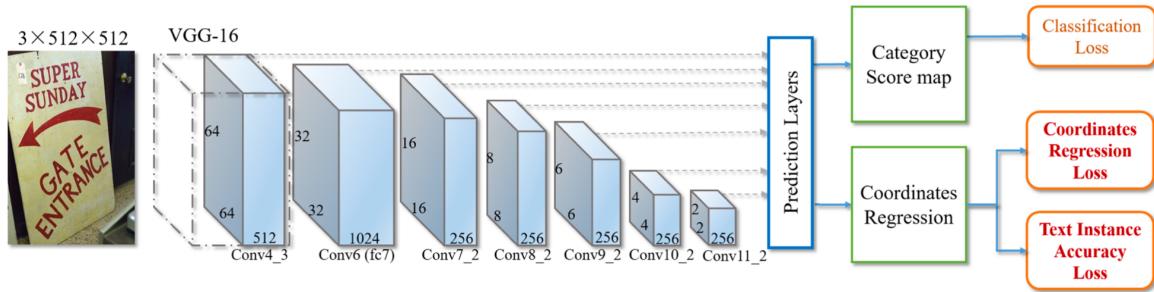


图 2-1 MATI 网络框架图。

MATI 的主要贡献在于：为了使得回归学得准确，提出 Starting-point-independent Coordinates Regression Loss 和 Text Instance Accuracy Loss。Starting-point-independent Coordinates Regression Loss 如下：

$$L_{reg} = \sum_{m \in L_{reg}^+} \min_{j \in [0, \dots, n-1]} \sum_{i=0}^{n-1} smooth_{L_1}(\hat{z}_i^m - z_{(j+i)\%n}^{m*}),$$

其中 \hat{z}_i^m 是第 m 个文本实例的第 i 个预测顶点， z_i^{m*} 是第 m 个文本实例的第 i 个顶点标注。该 Loss 的物理含义是：只需要计算预测点和标注中相邻最近的点的 loss。

Text Instance Accuracy Loss 如图 2-2 所示：将预测的点转化为 mask 和 mask 的 gt 之间计算 IOU，具体如何将预测的点转化为 mask 没有具体说明，需查看对于参考文献。

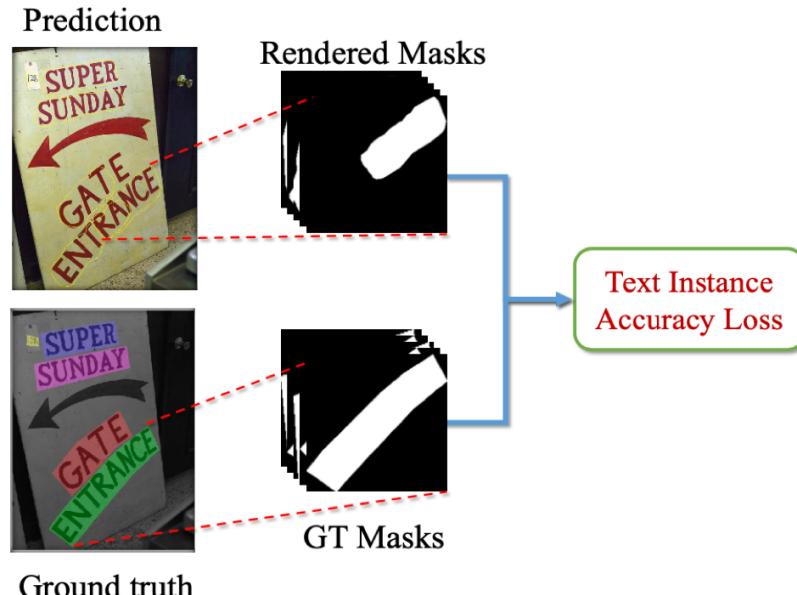


图 2-2 Text Instance Accuracy Loss。

2.1.2 TextTubes

TextTubes[19] 同样是针对曲形文本检测的问题，方法上属于基于回归的方法。该方法并不直接回归文本的边界点，而是通过回归出文本的中心轴，以及中心轴上垂直方法的半径来描述文本区域。

2.1.2.1 TextTubes 的网络结构

TextTubes 的网络结构如图2–3所示：网络的整体结构基于 MaskRCNN，将 MaskR-CNN 的 mask head 更改为该方法中的 Tube head（回归文本中心轴以及半径）。

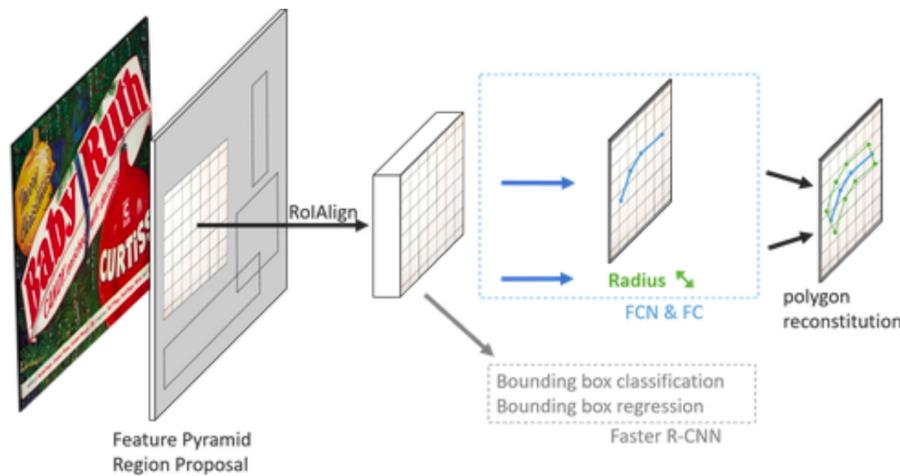


图 2–3 TextTubes 网络框架图。

文中指出：文本的中心轴在训练时难以确定（中心轴附近的点也可算为中心轴区域），因此提出新的损失函数而不是直接进行回归中心轴上的点。新的损失函数定义如下：

$$l_{tube} = l_{radius} + l_{axis} + l_{endpoints} + l_{spread}$$

其中 l_{axis} 定义如下，其中 \hat{S} 表示回归的中心线的点的集合（回归 4 个点，采样 100 个点计算 loss），

$$l_{axis}(\hat{S}, S) = 1 - \alpha \vartheta_{abs}(\hat{\gamma}, \gamma) - (1 - \alpha) \vartheta_{tan}(\hat{\gamma}, \gamma),$$

$$\vartheta_{abs}(\hat{\gamma}, \gamma) = \int_0^1 \exp\left(-\frac{d(\hat{\gamma}(t) - \gamma(t))^2}{2\sigma^2}\right) dt, \vartheta_{tan}(\hat{\gamma}, \gamma) = \int_0^1 \exp\left(-\frac{\sin^2(\hat{\theta}(t) - \theta(t))}{2\sigma^2}\right) dt$$

l_{axis} 的物理含义是：通过积分来计算预测的中心轴区域与 gt 的 loss，使得整体上能够对齐。 $\hat{\gamma}$ 的定义没有看得很明白，具体请参考原文。

2.1.3 DB (AAAI2020)

DB[9] 是基于分割的快速文本检测器。其快速的原因在于通过动态学习分割的阈值从而简化了分割图到 contours 的后处理过程。

2.1.3.1 DB 的网络结构

DB 的网络结构如图2-4所示：网络通过预测文本实例的分割图以及阈值图，然后通过比较分割图和阈值图得到二进制图（约化的二进制图）。

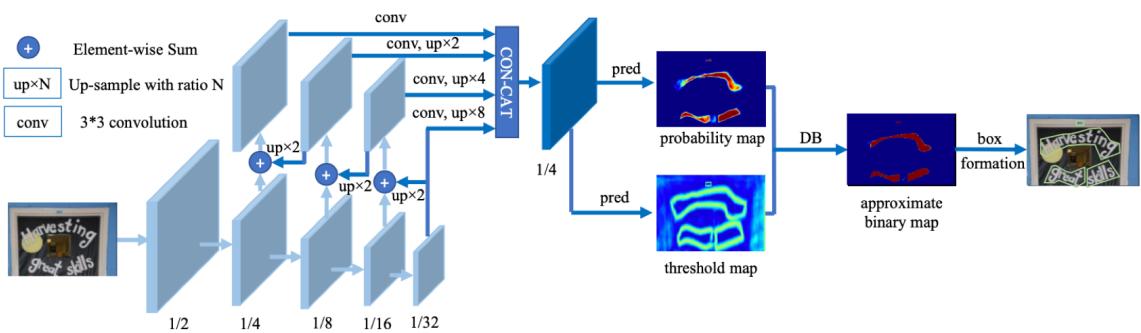


图 2-4 DB 网络框架图。

DB 的二值化函数如下：

$\hat{B}_{i,j} = \frac{1}{1+e^{-k(P_{i,j}-T_{i,j})}}$ ，其中 $P_{i,j}$ 代表分割图，其中 $T_{i,j}$ 代表阈值图。训练过程中，分割图和阈值图的 ground truth 如图2-5所示。

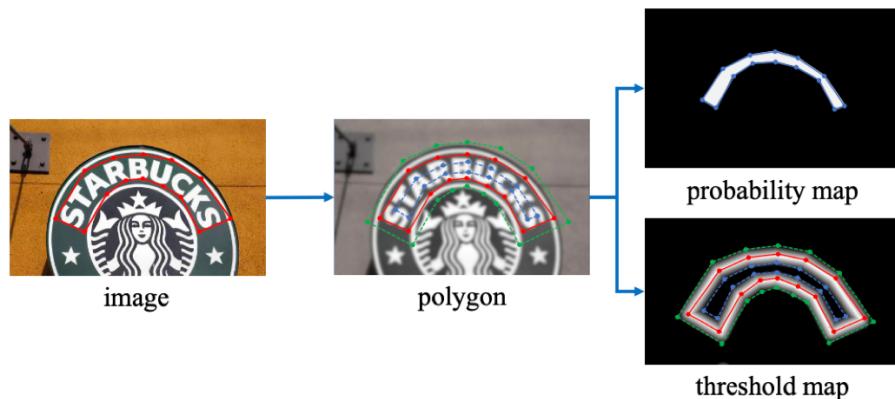


图 2-5 DB label 制作过程。

2.1.4 RelaText

RelaText[16] 属于 bottom-up 类型的任意形状文本检测器。

2.1.4.1 RelaText 的网络结构

RelaText 的网络结构如图2-6所示：网络的整体结构基于 FPN，采用 Anchor-Free RPN[31] 的架构。网络首先预测出文本的组件（由四边形构成，如图2-7所示），然后将经过 nms 后保留的每个 graph 经过 GCN 来获得最终的文本实例区域。

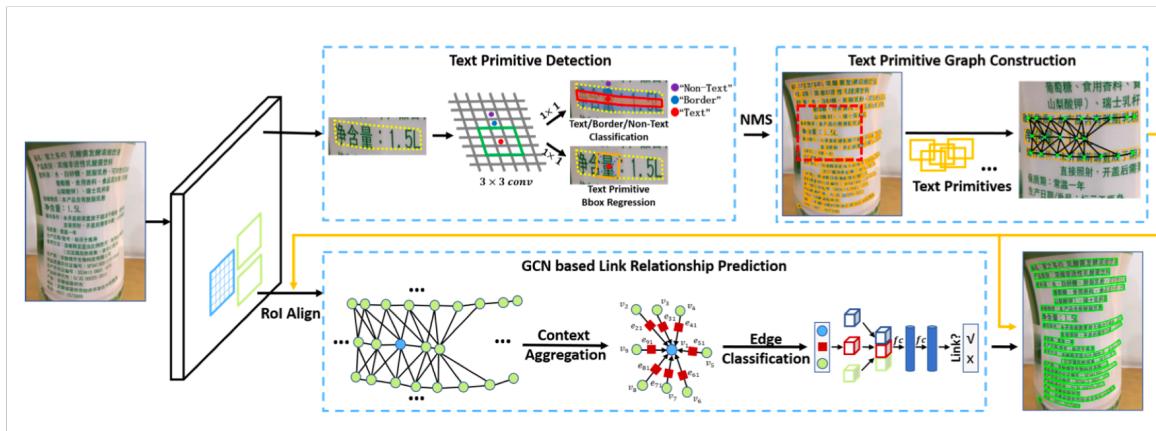


图 2-6 RelaText 网络框架图。

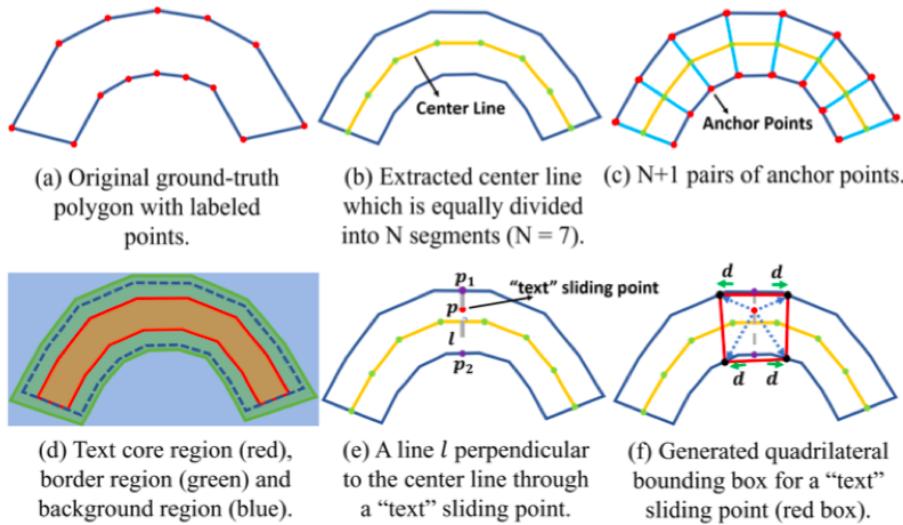


图 2-7 RelaText Label 制作过程。

在构建 graph 时，基本原则为：同一 graph 的组件只来自于 FPN 中的同一层，并且这些组件的空间位置相近。具体地：文本组件在 graph 中表现为一个节点，某一 graph 的节点的相邻节点来自于该节点的 K 近邻（两个文本组件的距离定义为组件中心的空间距离）。所有的 graph 构建完成后，通过预测 graph 中的边的分类结果来确定组件是否属于同一文本。边分类如图2-8所示：每条边的特征包括两个节点的视觉特征以及几何特征。几何特征定义为 $[\Delta(b_i, b_j), \Delta(b_i, b_{ij}), \Delta(b_j, b_{ij})]$ ，共 18 维度。其中 Δb_i , Δb_j , Δb_{ij} 分别代表两组件的 box 以及两个 box 的并集。 $\Delta(b_i, b_j) = (t_x^{ij}, t_y^{ij}, t_w^{ij}, t_h^{ij}, t_x^{ji}, t_y^{ji})$ 定义如下：

$$\begin{aligned} t_x^{ij} &= (x^i - x^j)/w^i, \quad t_y^{ij} = (y^i - y^j)/h^i, \\ t_w^{ij} &= \log(w^i/w^j), \quad t_h^{ij} = \log(h^i/h^j), \\ t_x^{ji} &= (x^j - x^i)/w^j, \quad t_y^{ji} = (y^j - y^i)/h^j. \end{aligned}$$

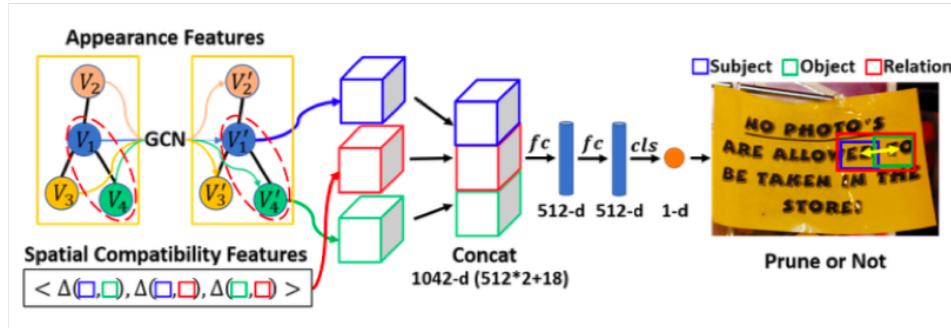


图 2-8 RelaText GCN 中边分类过程。

2.1.5 PuzzleNet

PuzzleNet[11] 属于 bottom-up 类型的任意形状文本检测器。通过 GCN 来连接各个文本组件。和 ReLaText 的不同在于 graph 的构建：ReLaText 是将同一尺度，K 近邻的文本组件构建成一个 graph，而 PuzzleNet 是一张图像内的所有组件构建一个 graph。因为 ReLaText 中每个文本实例由大量的文本组件构成（比较 PuzzleNet 和 ReLaText 网络生成文本组件），为了高效，所以一张图像需要构建多个 graph 单独处理。

2.1.5.1 PuzzleNet 的网络结构

RelaText 的网络结构如图2-9所示，网络的 backbone 基于 RPN。首先预测出文本组件（旋转长方形），然后将 nms 后的所有文本组件构建一个 graph（一张图像一个 graph），通过 GCN 网络来判断相邻节点是否连接。

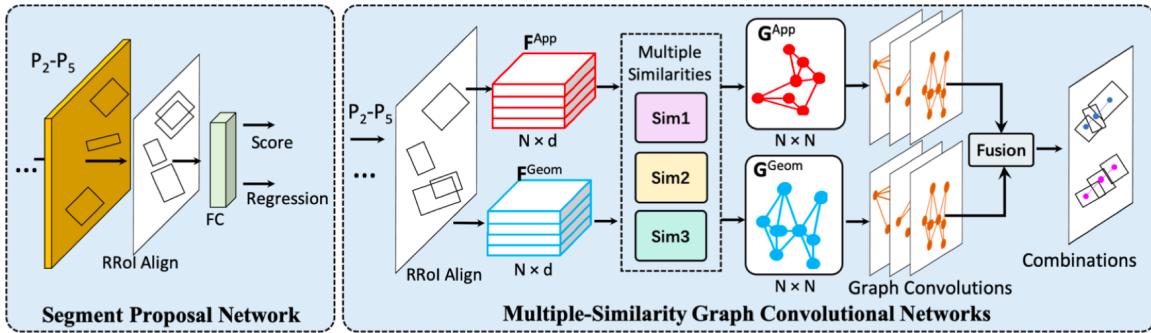


图 2–9 PuzzleNet 网络框架图。

具体来说，网络的 MSGCN 模块中分为两个支路（appearance graph 和 geometry graph），其过程可用下式表达。

$$link = classifier(cat(K(F_{App}, F_{App})F_{App}W_{App} + F_{App}, K(F_{Geo}, F_{Geo})F_{Geo}W_{Geo} + F_{Geo}))$$

其中， $F_{App} \in \mathbb{R}^{N \times d}$ 代表视觉特征， $F_{Geo} \in \mathbb{R}^{N \times d}$ 代表几何特征， $W \in \mathbb{R}^{d \times d}$ ， K 代表相似性度量函数， $K(*) \in \mathbb{R}^{N \times N}$ ， $cat(*) \in \mathbb{R}^{N \times 2d}$ 。相似性度量函数定义如下：

$$K = \beta_1 K_1 + \beta_2 K_2 + \beta_3 K_3, s.t. \beta_1 + \beta_2 + \beta_3 = 1,$$

$K_1(y_1, y_2) = \frac{y_1 y_2^T}{\|y_1\| \|y_2\|}$, $K_2(y_1, y_2) = \exp(-\frac{\|y_1 - y_2\|^2}{2\sigma^2})$, $K_3(y_1, y_2) = \exp(-\frac{JSD(y_1, y_2)}{2\sigma^2})$, JSD 代表 Jensen-Shannon Divergence。

可以看出 $K(F_{App}, F_{App})F_{App}W_{App} + F_{App}$ 和 NonLocal 模块的含义一致，起到特征加强的作用。

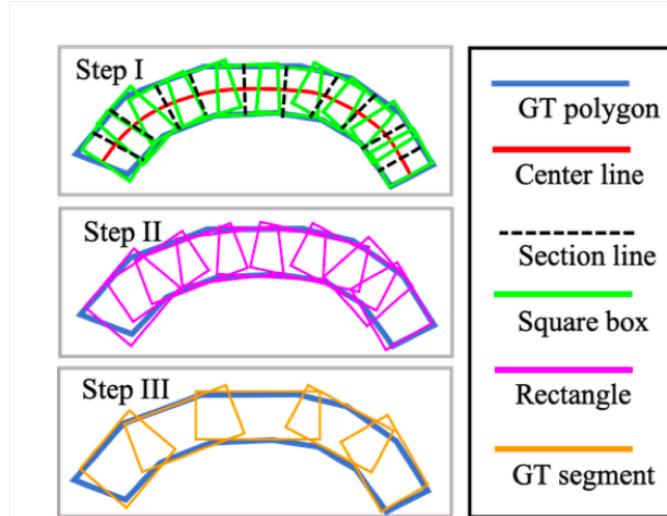


图 2–10 PuzzleNet Label 制作过程：1) 生成大量的小的文本组件，2) 相邻的小文本组件融合为长方，3) 将角度相近的长方形融合。

2.1.6 DRRG (CVPR2020)

DRRG[29] 属于 bottom-up 类型的任意形状文本检测器。将文本区域表示为多个文本组件，通过 GCN 来预测各个组件之间是否连接。和 ReLaText 一样，一张图像的所有文本组件构建为多个 graph，构建 graph 的算法和 ReLaText 不同。

2.1.6.1 DRRG 的网络结构

DRRG 的网络结构如图2–11所示：网络通过 Text Compont Prediction 模块来预测文本组件的属性（其 backbone 如图2–12所示，文本组件生成过程如图2–13所示），然后通过 IPS[26] 算法构建多个 Local Graphs，然后 Relational Reasoning 模块预测 Local Graphs 中边的连接关系，最后融合 Local Graphs 的预测结果（一个文本实例的文本组件可能处于多个 graph 中），得到最后完整的文本组件的连接关系。

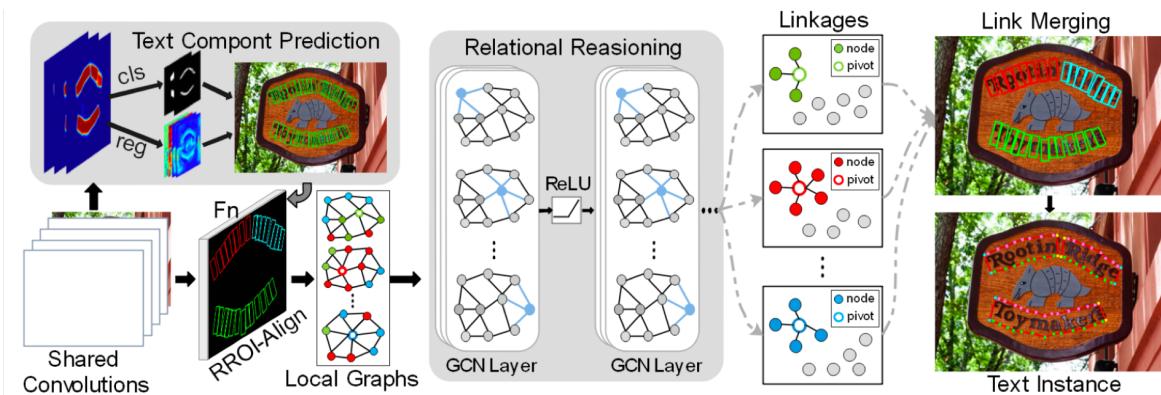


图 2–11 DRRG 网络框架图。

Local Graphs 的构建过程如下：1) 选定 pivot (按照文献 [26] 的做法，依次将每个文本组件作为 pivot); 2) 以 pivot 为中心，2-hop (暂时理解为往外延伸 2 代节点，每个节点最多 4 个邻域，该理解需要进一步确认) 的文本组件构建为一个 graph；寻找领域时的相似性度量如下所示： $E_s = 1 - D(p, v_i)/\max(H_m, W_m)$, $v_i \in V_p$, $D(p, v_i)$ 代表节点的文本组件中心的距离， H_m, W_m 代表图像高宽。

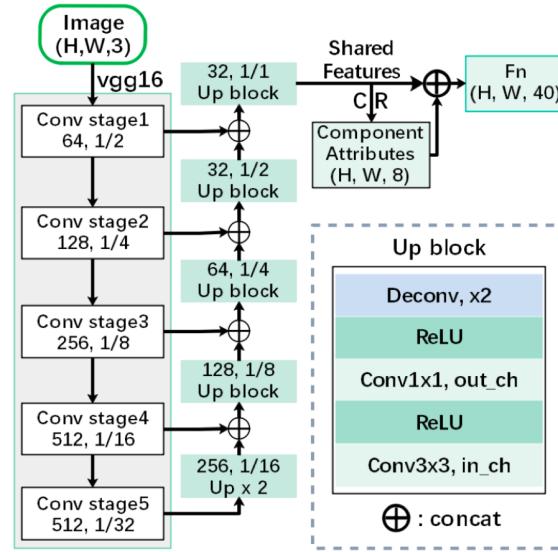


图 2–12 DRRG 的 backbone。

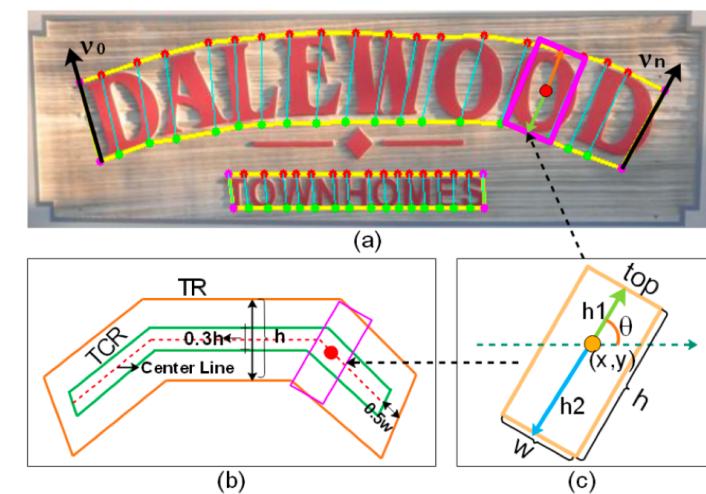


图 2–13 文本组件构建过程。

Local Graphs 中节点特征以及连接矩阵如图2–14所示：节点特征由视觉特征 F_r 和几何特征构成 F_g 构成。 F_r 由 RRoI-Align 进行提取， F_g 由预测的几何向量 $(x, y, h, w, \cos\theta, \sin\theta)$ 经变换函数 ε 构成。

$$\varepsilon_{2i} = \cos\left(\frac{z}{1000^{2i/C_\varepsilon}}\right), i \in (0, C_\varepsilon/2 - 1),$$

$$\varepsilon_{2i+1} = \sin\left(\frac{z}{1000^{2i/C_\varepsilon}}\right), i \in (0, C_\varepsilon/2 - 1).$$

其中 C_ε 代表每个几何属性生成的 Embedding 向量的维度，最终几何特征 F_g 的维度为 $6C_\varepsilon$ 。

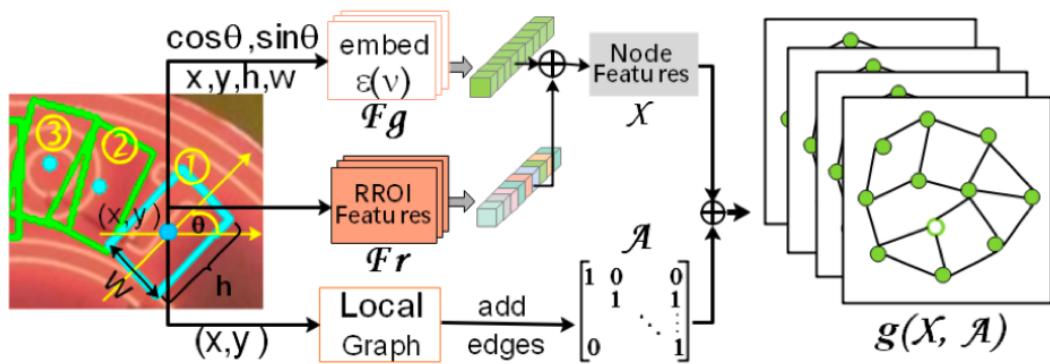


图 2-14 Graph 中节点特征以及连接矩阵：节点特征由视觉特征和几何特征构成。

3

端到端文字识别 (Scene Text Spotting)

2018 年以前，关于 Scene Text Spotting 的论文 [12]，主要集中在解决旋转文本的端到端识别问题，几乎没有论文解决曲形文本端到端识别的问题。自从发表在 ECCV2018 中的论文，MaskTextSpotter[15]，开始试图解决曲形文本端到端识别的问题以来，大量的工作开始致力于该问题的研究，如 [15, 8, 3, 27, 18, 23, 13, 17]。自此，端到端文本识别方法在理论上能够解决任意形状文本识别的问题。在以下表述中以“任意形状文本端到端识别方法”来统称既能处理多方向又能处理曲形文本的端到端文本识别方法。

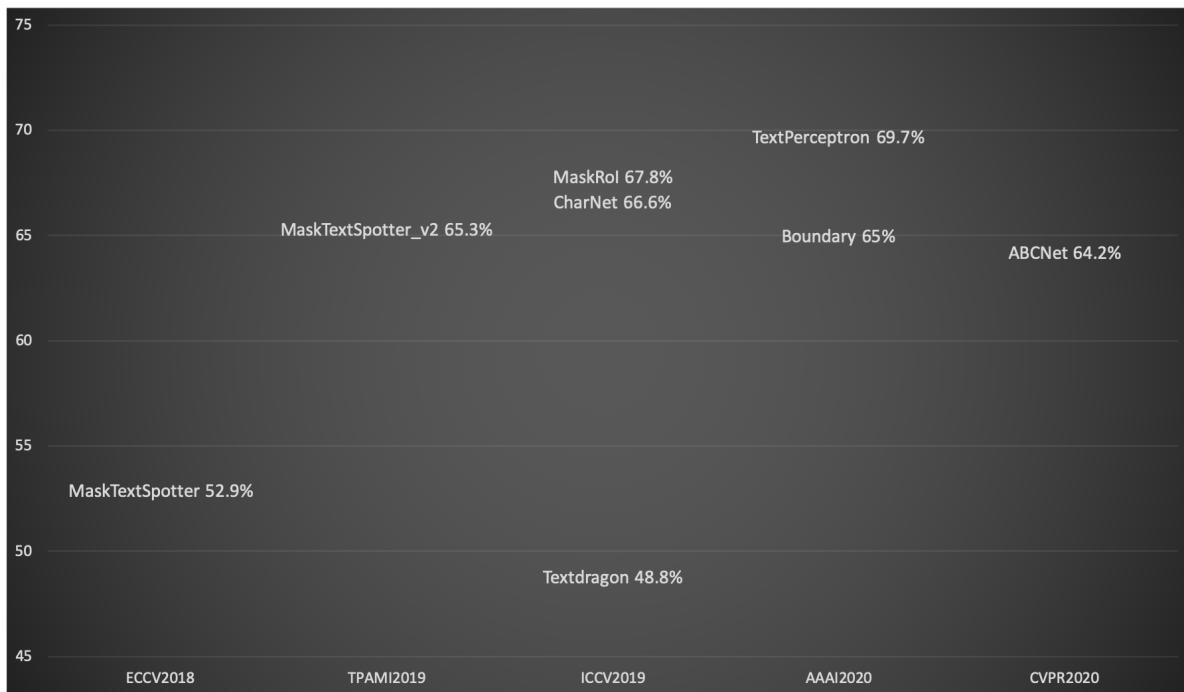


图 3-1 2018 年到 2020 年 3 月期间，曲形文本端到端文本识别方法在 TotalText 上的性能。

首先，我们从问题的角度出发，概述各种任意形状文本端到端识别方法之间的关

系。然后将从方法，实验结果，该方法的优缺点等方面来分别探讨各种方法，这些方法包括：MaskTextSpotter[15, 8]、TextDragon[3]、CharNet[27]、MaskRoI[18]、Boundary[23]、TextPerceptron[17]以及ABCNet[13]。最后，进一步总结归纳以上方法的特点。

3.1 各种任意形状文本端到端识别方法介绍

3.1.1 MaskTextSpotter (ECCV2018)

3.1.1.1 MaskTextSpotter 网络结构

MaskTextSpotter 作为首个任意形状文本端到端识别方法，其思路是将每个文字字符作为一个类别进行检测。其网络框架如图3-2所示，网络主体部分与 MaskRCNN[5]一致。由于常规的 MaskRCNN 网络（分割分支进行 1 通道的分割，表示是否为文字两个类别）只能完成文字检测任务，无法进行文字识别，因此作者将 Mask 分支设计为 37 个通道（26 个英文字符加上 10 个数字以及表示是否为字符区域的 1 通道）加上检测的 1 个类别共 38 个类别进行分割，其分割分支如图3-3所示。在测试阶段，检测的 1 通道能够检测任意形状的文本。根据另外 37 个通道的分割信息，按照从左到右的顺序连接每个字符，完成识别任务。

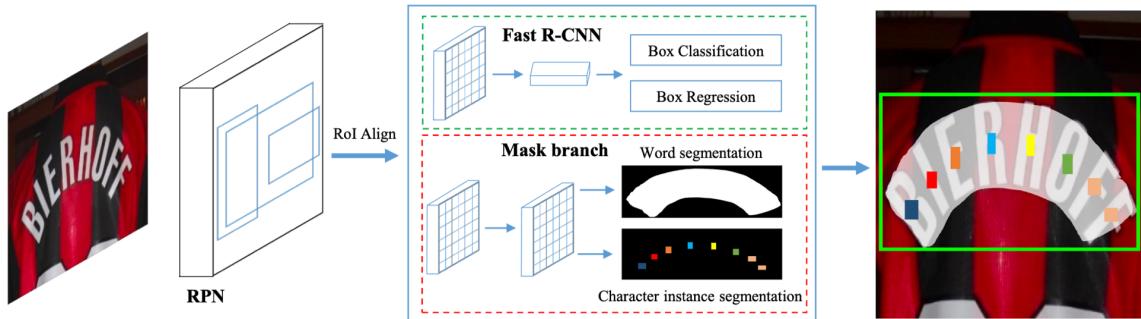


图 3-2 MaskTextSpotter 框架图。

3.1.1.2 MaskTextSpotter 实验细节

MaskTextSpotter 的训练过程分为两部分：合成数据集预训练和真实数据集微调阶段。合成数据集使用的是 SynthText[4]，batch size 为 8，输入图像以短边为 800，保持长宽比进行训练。在真实数据微调阶段，batch size 为 8，采用多尺度训练的策略，短

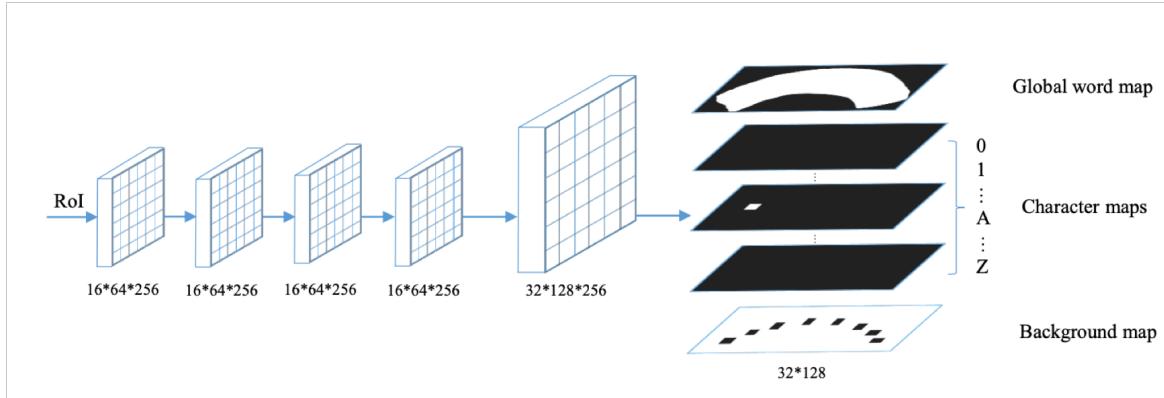


图 3-3 MaskTextSpotter 的分割分支结构图。

边分为 (600, 800, 1000) 三个尺度进行训练，使用的数据集有 SynthText, ICDAR2013, ICDAR2015, TotalText 以及来自 [30] 的 1162 张图像。

3.1.1.3 MaskTextSpotter_v2 网络结构

由于 MaskTextSpotter 中将通过将每个字符作为单独的类别分割出来作为识别结果，这样会导致识别过程中忽略文字的语义特征。基于此，MaskTextSpotter_v2 的主要出发点是将文字的语义信息融入到识别分支中，最终其网络结构如图3-4所示。可以看出，该网络结构和 MaskTextspotter 相比，在识别分支加入了序列识别分支。其识别分支具体结构如图3-5所示。

识别分支由两部分组成：基于字符分割的识别分支和基于序列识别的识别分支。每个识别分支在输出识别结果的同时对识别结果的置信度进行打分，最终的识别结果取置信度较高的分支的识别结果。

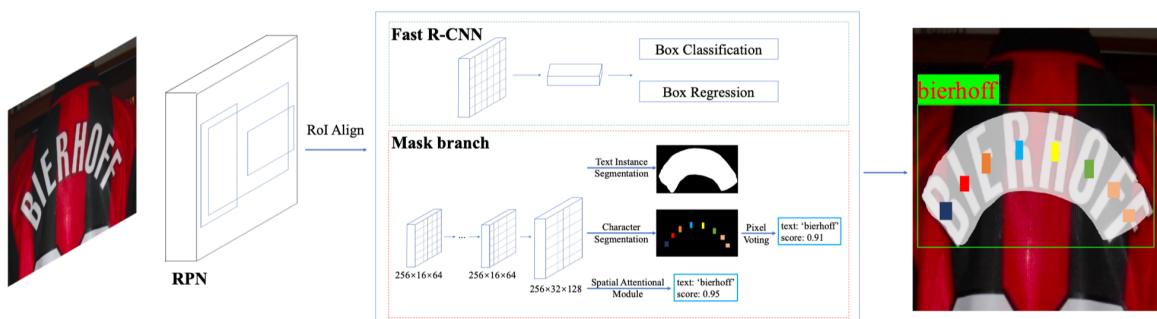


图 3-4 MaskTextSpotter_v2 框架图。

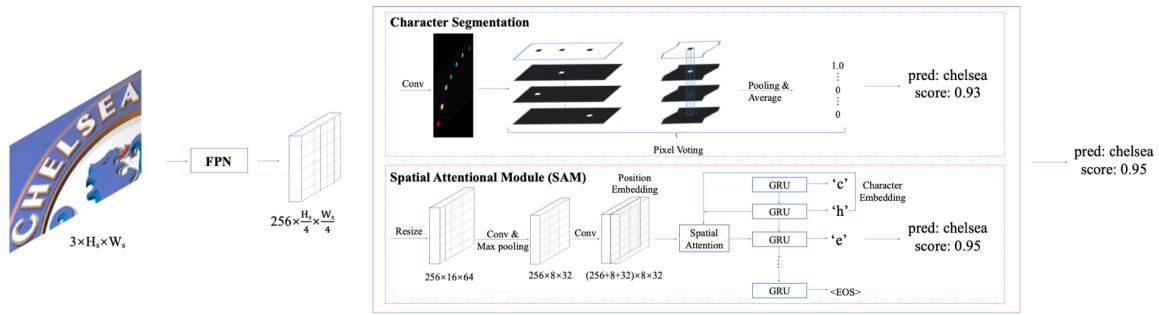


图 3-5 MaskTextSpotter_v2 识别分支结构图。

3.1.1.4 MaskTextSpotter_v2 实验细节

MaskTextSpotter 的训练过程分为两部分：合成数据集预训练和真实数据集微调阶段。合成数据集使用的是 SynthText[4]，batch size 为 8，输入图像以短边为 800，保持长宽比，学习率从 0.001 开始，第 100k, 200k 次下降 0.1，训练 270k 步。在真实数据微调阶段，batch size 为 8，采用多尺度训练的策略，短边分为 (600, 800, 1000, 1200, 1400) 三个尺度进行训练，使用的数据集有 SynthText, ICDAR2013, ICDAR2015, TotalText 以及来自 [30] 的 1162 张图像。学习率从 0.001 开始，第 100k 下降 0.1，训练 150k 步。

3.1.2 TextDragon (ICCV2019)

虽然 MaskTextSpotter 能够进行任意形状文本端到端的识别，但是数据集字符级别的标注的要求使得网络的训练比较昂贵。TextDragon 意在只使用单词级别的标注来设计任意形状文本端到端识别系统。

3.1.2.1 TextDragon 网络结构

TextDragon 对任意形状文本的表示方式主要来自于文字检测方法 TextSnake[14]，也就是将任意形状的文本表示为一系列的带方向正方形。然后从属于同一文本实例的带方向正方形中聚合，采样一个子集形成文本区域。基于该子集的带方向正方形，则有：1) 从这些带方向正方形中提取文字边界点来表示检测结果，2) 对每个正方形区域的特征进行字符分类，通过 CTC 解码成文本字符串。TextDragon 的网络结构如图3-6所示。

具体地，文字的表示方法如图3-7所示。文本实例由文本的中心线、正方形边长以及正方形旋转方向构成。在测试阶段，推理过程如下：1) 通过文本中心线来获取每个文本实例大致区域；2) 在每个文本实例的中心线上获得所有的预测的带方向正方形，并保留这些 IOU 大于 0.5，旋转角度差小于 45 度的正方形区域；3) 根据其位置将这些保

留的正方形进行排序，形成表示该文本区域的正方形子集。4) 最终通过 RoISlide 从这些矫正后的文本区域中获取特征，进行字符串识别，而文本边界由正方形子集形成的边界点构成。

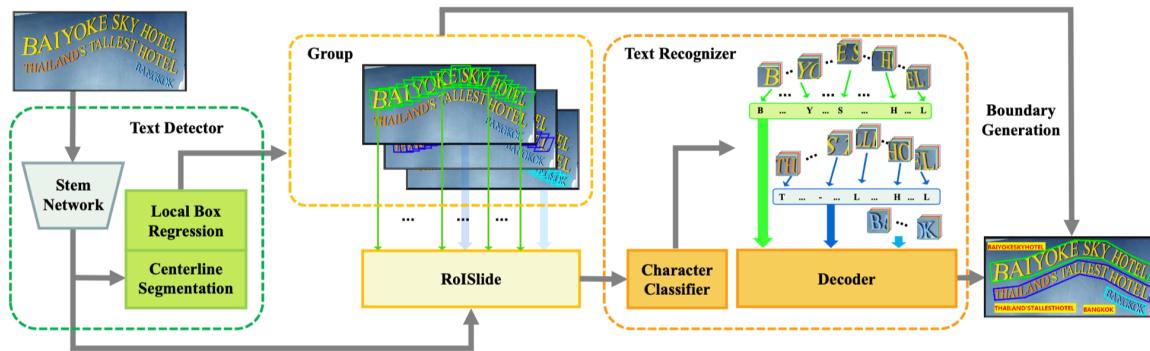


图 3-6 TextDragon 框架图。

3.1.2.2 TextDragon 实验细节

TextDragon 的训练过程分为两部分：合成数据集预训练和真实数据集微调阶段。合成数据集使用的是 SynthText，输入图像大小为 512*512，学习率为 0.01，训练 600k 步。在真实数据微调阶段，输入图像大小为 512*512，数据集为相对应数据集的训练集，学习率为 0.001，训练 120k 步。

3.1.3 CharNet (ICCV2019)

CharNet 与 TextDragon 一样是在任意形状文本端到端识别算法只有 MaskTextSpotter 的背景下出现的方法。他的主要出发点是：当前端到端识别的方法中，都是 two stage 的（这里 two stage 是指检测网络得到检测结果，再根据检测结果利用 ROI 操作获取特征进行识别），作者认为 two stage 中 ROI 提取很难提取准确（主要是检测存在误差），并且 two stage 过程繁琐，不便使用。因此，作者想设计一个 single stage 的网络，同时输出文本实例的检测和识别结果。

3.1.3.1 CharNet 网络结构

CharNet 在核心问题上和 MaskTextSpotter 一致，都是通过字符级别的分割解决任意形状文本的识别问题。MaskTextSpotter 是在 ROI 内进行分割，而 CharNet 是在全图进行字符级别的分割。那么，CharNet 就剩下最后一个需要解决的问题：如何将分割出的字

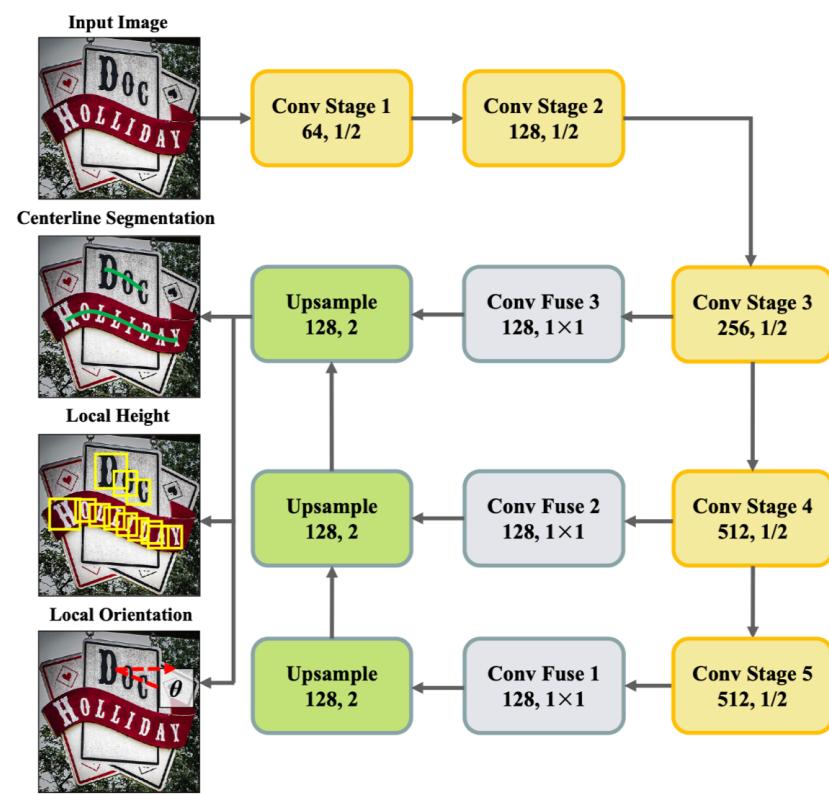


图 3-7 TextDragon 文本实例表示方法。

符 group 成为一个文本区域？如 CharNet 网络结构图3-8所示，其 Detection Branch 的作用便是预测一些信息来将分割出的字符聚合成字符串。对于多方向文本，其 Detection Branch 的表示方法为 EAST[32] 的表示方式，利用预测的四边形框的 IoU 聚合每个字符为字符串。对于曲形文本，其 Detection Branch 的表示方法为 TextField[28] 的表示方法。

3.1.3.2 CharNet 实验细节

CharNet 的训练过程分为两部分：合成数据集预训练和真实数据集微调阶段。合成数据集使用的是 SynthText，batch size 为 32，学习率为 0.0002，数据集迭代 5 epochs。在真实数据微调阶段，数据集为相对应数据集的训练集，学习率为 0.002，分三步进行迭代训练，三步训练回合分别为 100, 400, 800 epochs。这里每步迭代训练是指：利用先前的模型获得训练集的字符级别标注（检测框），过滤出正确的字符级别标注来训练模型，迭代 n（100, 400 或 800）个 epochs。

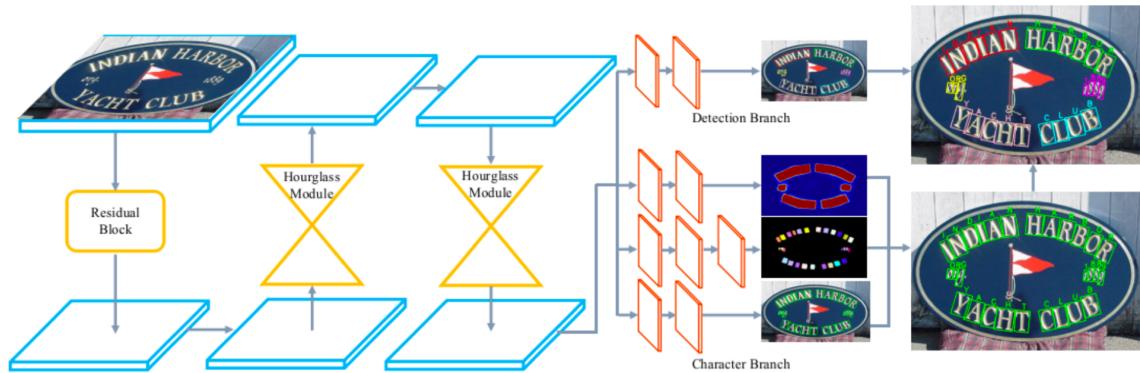


图 3-8 CharNet 框架图。

3.1.4 MaskRoI (ICCV2019)

MaskRoI 与 CharNet 以及 TextDragon 是同时期文章，该方法不需要字符级别的标注，同时不需要将任意形状文本矫正为水平文本进行识别。

3.1.4.1 MaskRoI 网络结构

如图3-9所示，MaskRoI 和 MaskTextSpotter一样，也是基于 MaskRCNN 框架进行改进的。识别分支采用基于 attention 的序列识别方案。为了解决任意形状文本的 RoI 容易采样到背景或者相邻文本特征的问题，在进行序列识别之前，进行了特征过滤操作。该操作就是将文本实例分割图和 RoI 的特征进行相乘。

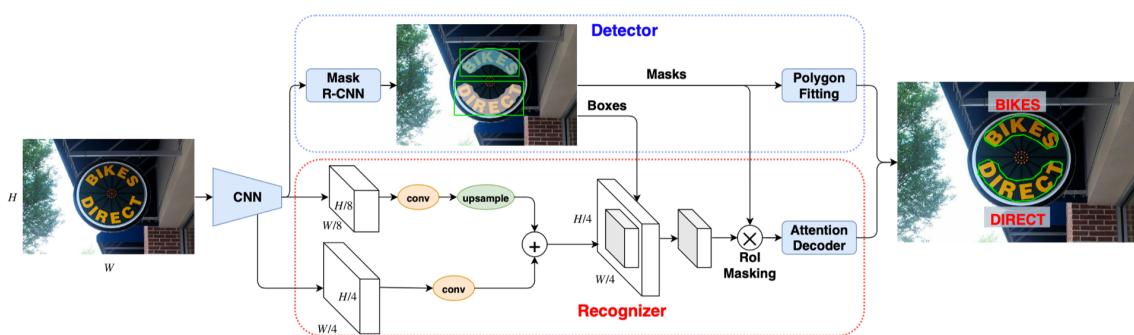


图 3-9 MaskRoI 框架图。

3.1.4.2 MaskRoI 实验细节

MaskRoI 采用一步训练的方式，数据集包括 SynthText, ICDAR2015, COCOText, ICDAR-MLT, TotalText 以及网络收集的通过 Google OCR API 标注的 30k 张图像。采用多尺度训练的策略，短边为 480 到 800 之间。

3.1.5 Boundary (AAAI2020)

Boundary[23] 和基于分割的方法得到任意形状文本的边界不同，该文章通过回归的方式获得任意形状文本的边界，从而避免负责的后处理过程使得检测识别两个过程完全端到端可训练。

3.1.5.1 Boundary 网络结构

Boundary 的网络结构如图3-10所示：网络整体基于 FPN 网络，首先检测出文本的长方形包围盒，提取该长方形内的特征（这样相当于将文本在角度上进行了归一化，使得边界点的回归更为准确）；然后检测文本的边界点，最后通过 Arbitrary RoIAlign 将曲形文本矫正为水平文本特征进行识别。

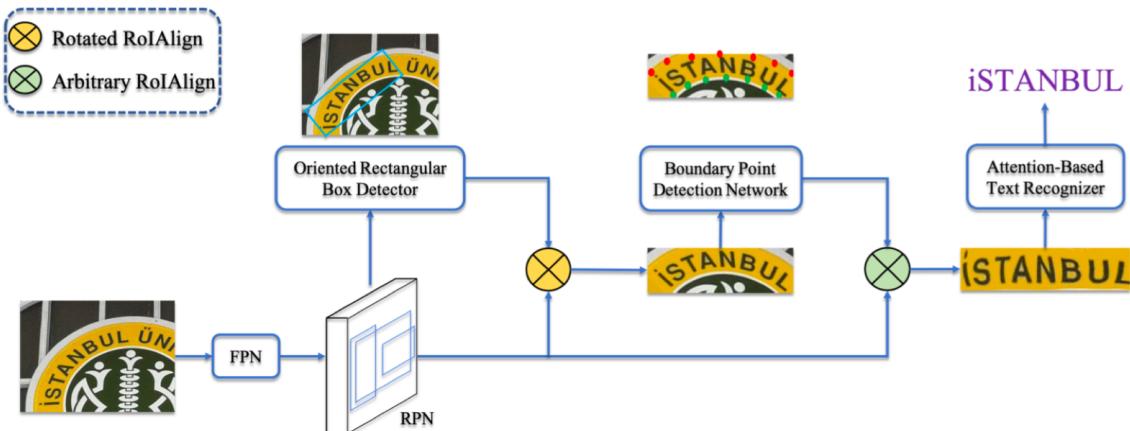


图 3-10 Boundary 框架图。

Boundary 详细的回归细节如图3-11所示：准确回归出文本边界点的重要细节在于，在 (b) 处将文本的方向信息进行了归一化（即将文本旋转为水平进行边界点的检测），这样使得边界点的分布更为稳定。另外，文章中并不是直接预测边界点的坐标，而是在预设的 anchor points 基础上进行边界点的回归，这样同样可以简化边界点的回归任务。

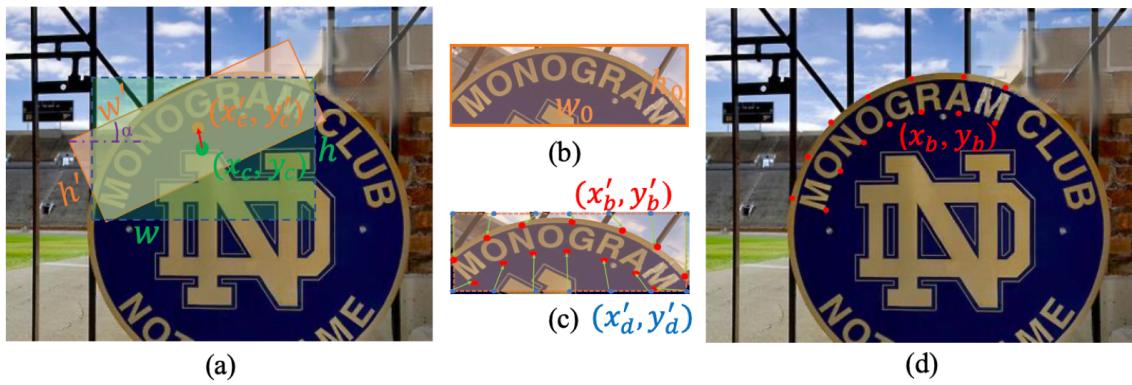


图 3-11 Boundary 回归过程：(a) 回归出文本的带方向的长方形包围盒；(b) 归一化文本角度；(c) 从 anchor points 处回归到 boundary points。

3.1.6 TextPerceptron (AAAI2020)

TextPerceptron 的主体思路也是文本实例边界点检测 + TPS+ 序列识别。和 Boundary 不同之处在于文本边检点检测过程，具体地说，是通过文本实例的几何属性和后处理得到边界点。

3.1.6.1 TextPerceptron 网络结构

TextPerceptron 的网络结构如图3-12所示。边界点的检测是基于分割的方法，预测的文本实例的几何属性包括：1) 文本上下边界；2) 文本实例的开端；3) 文本实例的结尾；4) 文本实例的中间区域；5) 开端以及结尾处的角点回归；6) 文本中心区域的边界点回归。其中 5) 和 6) 的标签定义如图3-13所示。

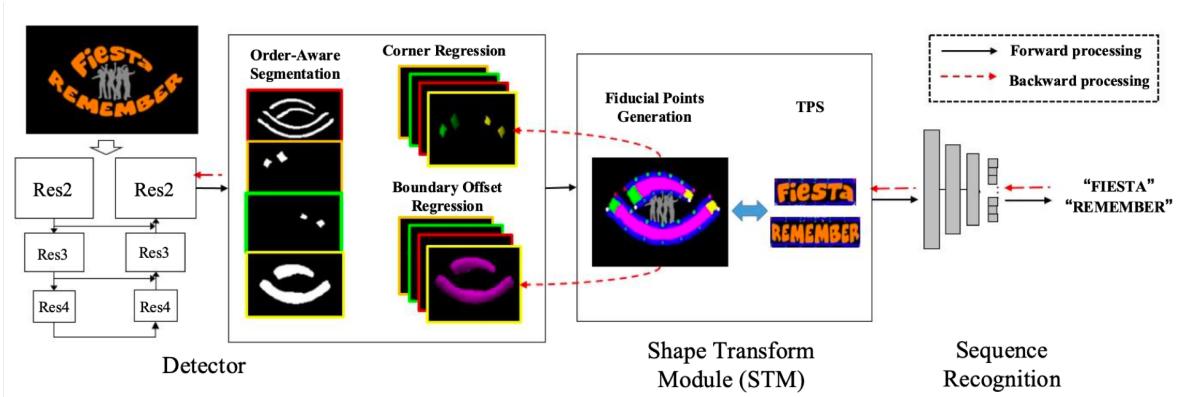


图 3-12 TextPerceptron 框架图。

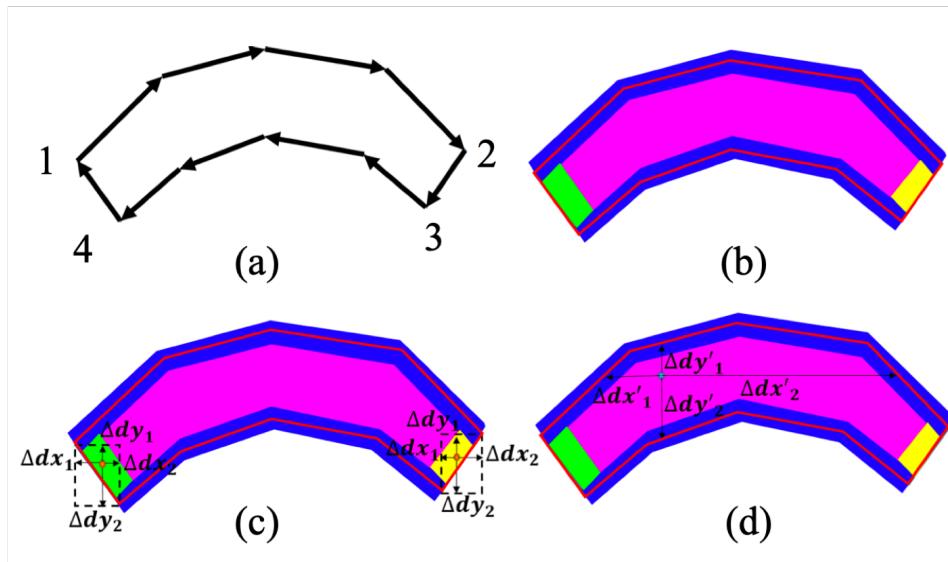


图 3-13 TextPerceptron 角点和边界点回归的定义。

3.1.6.2 TextPerceptron 边界点获取过程

根据预测的文本实例的几何属性，后处理得到边界点的过程如下：1) 根据中心区域和开端以及结尾的匹配程度可以获得开端、结尾匹配对，上下边界可以用于区分相邻的文本实例；2) 在开端、结尾分割图处获取文本实例的 4 个角点；3) 如图3-14所示，获得较长边的角点对的中心点，作垂线，获得该点对所处边界的交点作为一个边界点，以此类推，获得所有边界点。

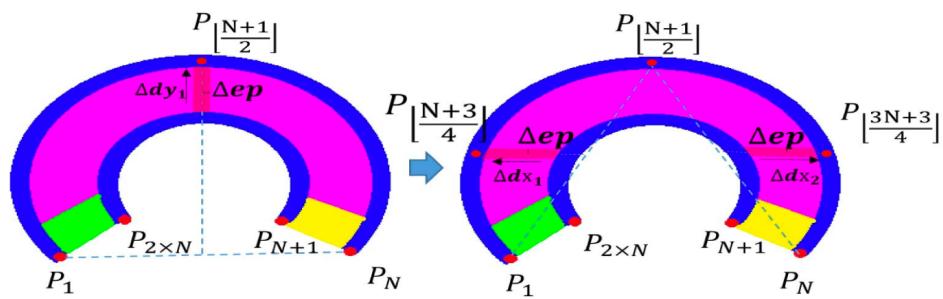


图 3-14 TextPerceptron 后处理过程。

3.1.6.3 TextPerceptron 实验细节

TextPerceptron 分为 3 个阶段，1) 训练检测分支，在 SynthText 上以学习率 0.002 训练 5 epochs；2) 训练识别分支，在 SynthText 上以学习率 0.002 训练 5 epochs；3) 检测识别联合训练，在各自训练集上以学习率 0.001 训练 80 epochs，每 20 epochs 学习率乘以 0.1。

3.1.7 ABCNet (CVPR2020)

ABCNet 的主体思路也是文本实例边界点检测 +TPS+ 序列识别。边界点的检测采样回归的方法，和 Boundary 不同的是，检测部分采用 anchor-free 的方法。论文的框架主要基于 FCOS[21] 上进行改进。

3.1.7.1 ABCNet 网络结构

ABCNet 采用贝塞尔曲线来表示文本实例的边界，贝塞尔曲线描述效果如图3-16所示。网络框架如图3-15所示，FCOS 采用密集预测的方式。从代码中可以看出，检测部分的预测信息包括：1) 每个 bbox 的得分；2) 中心区域预测；3) bbox 回归；4) 贝塞尔曲线控制点预测。获得贝塞尔曲线后，

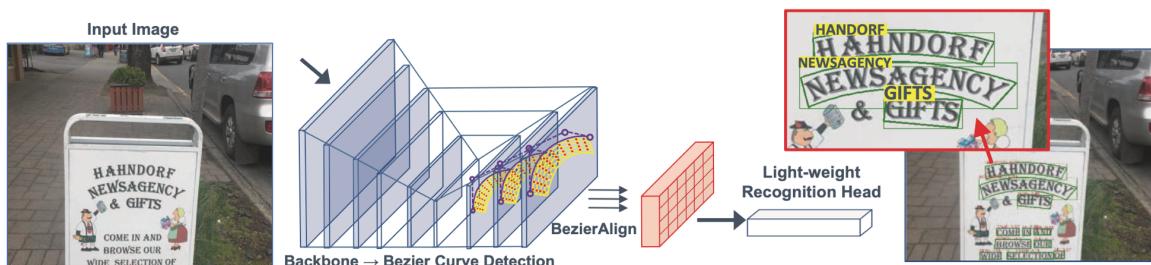


图 3-15 ABCNet 框架图。

贝塞尔曲线获得的边界如图3-17所示。

3.1.7.2 ABCNet 实验细节

TextPerceptron 分为 2 个阶段：1) 合成的 150k 合成数据集，15k 的 COCOText，7k 的 ICDAR-MLT 预训练；2) 相应的训练集训练。

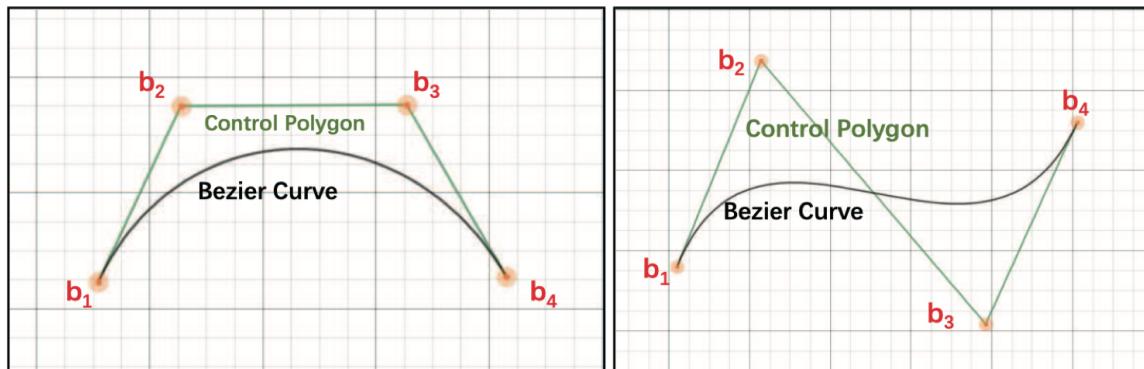


图 3-16 贝塞尔曲线。

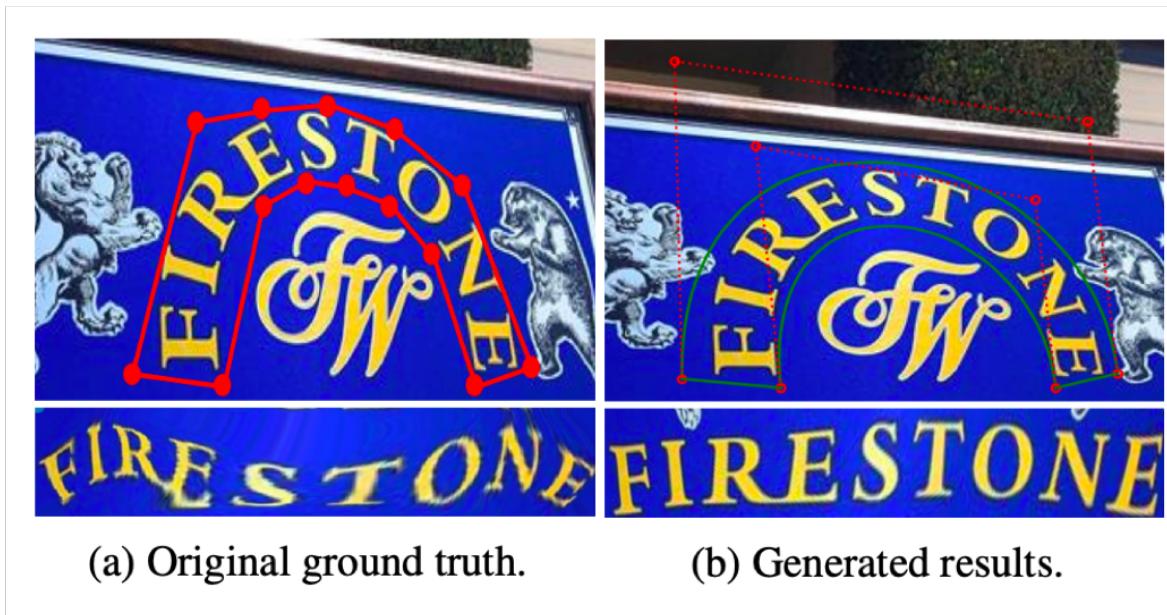


图 3-17 贝塞尔曲线。

Bibliography

- [1] Youngmin Baek et al. “Character region awareness for text detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9365–9374.
- [2] Yu Deli et al. “Towards Accurate Scene Text Recognition with Semantic Reasoning Networks”. In: (2020).
- [3] Wei Feng et al. “TextDragon: An End-to-End Framework for Arbitrary Shaped Text Spotting”. In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [4] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. “Synthetic data for text localisation in natural images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2315–2324.
- [5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [6] Wenyang Hu et al. “GTC: Guided Training of CTC Towards Efficient and Accurate Scene Text Recognition”. In: *arXiv preprint arXiv:2002.01276* (2020).
- [7] XiaoQian Li, Jie Liu, ShuWu Zhang, and GuiXuan Zhang. “Learning to Predict More Accurate Text Instances for Scene Text Detection”. In: *arXiv preprint arXiv:1911.07423* (2019).
- [8] Minghui Liao et al. “Mask textsspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes”. In: *IEEE transactions on pattern analysis and machine intelligence* (2019).
- [9] Minghui Liao et al. “Real-time Scene Text Detection with Differentiable Binarization”. In: *arXiv preprint arXiv:1911.08947* (2019).
- [10] Ron Litman et al. “SCATTER: Selective Context Attentional Scene Text Recognizer”. In: (Mar. 2020).

- [11] Hao Liu et al. “PuzzleNet: Scene Text Detection by Segment Context Graph Learning”. In: *arXiv preprint arXiv:2002.11371* (2020).
- [12] Xuebo Liu et al. “Fots: Fast oriented text spotting with a unified network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5676–5685.
- [13] Yuliang Liu et al. “ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network”. In: *arXiv preprint arXiv:2002.10200* (2020).
- [14] Shangbang Long et al. “Textsnake: A flexible representation for detecting text of arbitrary shapes”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 20–36.
- [15] Pengyuan Lyu et al. “Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 67–83.
- [16] Chixiang Ma, Lei Sun, Zhuoyao Zhong, and Qiang Huo. “ReLaText: Exploiting Visual Relationships for Arbitrary-Shaped Scene Text Detection with Graph Convolutional Networks”. In: *arXiv preprint arXiv:2003.06999* (2020).
- [17] Liang Qiao et al. “Text Perceptron: Towards End-to-End Arbitrary-Shaped Text Spotting”. In: *arXiv* (2020), arXiv–2002.
- [18] Siyang Qin et al. “Towards unconstrained end-to-end text spotting”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 4704–4714.
- [19] Joël Seytre, Jon Wu, and Alessandro Achille. “TextTubes for Detecting Curved Text in the Wild”. In: *arXiv preprint arXiv:1912.08990* (2019).
- [20] Baoguang Shi et al. “Aster: An attentional scene text recognizer with flexible rectification”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.9 (2018), pp. 2035–2048.
- [21] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. “Fcos: Fully convolutional one-stage object detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 9627–9636.
- [22] Zhaoyi Wan et al. “TextScanner: Reading Characters in Order for Robust Scene Text Recognition”. In: *arXiv preprint arXiv:1912.12422* (2019).
- [23] Hao Wang et al. “All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting”. In: *arXiv preprint arXiv:1911.09550* (2019).

- [24] Tianwei Wang et al. “Decoupled Attention Network for Text Recognition”. In: *arXiv preprint arXiv:1912.10205* (2019).
- [25] Wenhui Wang et al. “Shape robust text detection with progressive scale expansion network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9336–9345.
- [26] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. “Linkage based face clustering via graph convolution network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1117–1125.
- [27] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R. Scott. “Convolutional Character Networks”. In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [28] Yongchao Xu et al. “TextField: learning a deep direction field for irregular scene text detection”. In: *IEEE Transactions on Image Processing* 28.11 (2019), pp. 5566–5579.
- [29] Shi-Xue Zhang et al. “Deep Relational Reasoning Graph Network for Arbitrary Shape Text Detection”. In: *arXiv preprint arXiv:2003.07493* (2020).
- [30] Zhuoyao Zhong, Lianwen Jin, Shuye Zhang, and Ziyong Feng. “Deeptext: A unified framework for text proposal generation and text detection in natural images”. In: *arXiv preprint arXiv:1605.07314* (2016).
- [31] Zhuoyao Zhong, Lei Sun, and Qiang Huo. “An anchor-free region proposal network for Faster R-CNN-based text detection approaches”. In: *International Journal on Document Analysis and Recognition (IJDAR)* 22.3 (2019), pp. 315–327.
- [32] Xinyu Zhou et al. “EAST: an efficient and accurate scene text detector”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 5551–5560.