

学校代码：10246
学号：17210240269

復旦大學

硕士 学位 论文 (专业学位)

基于局部分割的自然场景图像字符检测与识别

Local Segmentation based Scene Text Detection and Recognition

院 系：计算机科学技术学院

专业学位类别（领域）：计算机技术

姓 名：周钊

指 导 教 师：薛向阳 教授

完 成 日 期：2019年 10月 8日

指导小组成员

薛向阳 教 授
张玥杰 教 授
金 城 教 授
李 斌 青年研究员

目 录

| | |
|---------------------------|------------|
| 摘要 | vii |
| Abstract | ix |
| 第1章 绪论 | 1 |
| 1.1 字符检测 | 1 |
| 1.2 字符识别 | 2 |
| 1.3 问题和挑战 | 3 |
| 1.4 主要贡献 | 4 |
| 第2章 研究现状 | 7 |
| 2.1 字符检测 | 7 |
| 2.1.1 传统方法字符检测 | 7 |
| 2.1.2 深度学习字符检测 | 7 |
| 2.2 字符识别 | 15 |
| 2.2.1 传统方法字符识别 | 15 |
| 2.2.2 深度学习字符识别 | 16 |
| 第3章 任意形状的字符检测 | 23 |
| 3.1 算法流程 | 23 |
| 3.2 局部分割网络 | 24 |
| 3.2.1 网络结构 | 24 |
| 3.2.2 候选框生成 | 25 |
| 3.2.3 文字分割 | 26 |
| 3.2.4 损失函数 | 26 |
| 3.3 弯曲连接 | 27 |
| 3.3.1 分割区域连接 | 28 |
| 3.3.2 多边形文字边缘生成 | 28 |
| 3.4 字符检测数据集 | 29 |
| 3.4.1 任意方向数据集 | 29 |
| 3.4.2 任意形状数据集 | 30 |
| 3.4.3 评价指标 | 31 |
| 3.5 实验结果和比较 | 31 |
| 3.5.1 候选框标签生成 | 31 |
| 3.5.2 数据增强 | 32 |

| | |
|-------------------------|-----------|
| 3.5.3 模型训练 | 32 |
| 3.5.4 实验结果和比较 | 32 |
| 第 4 章 任意形状的字符识别 | 35 |
| 4.1 算法流程 | 35 |
| 4.2 特征提取改进 | 36 |
| 4.3 识别算法网络结构 | 38 |
| 4.4 字符识别数据集 | 38 |
| 4.5 实验结果与比较 | 39 |
| 第 5 章 总结和展望 | 45 |
| 5.1 总结 | 45 |
| 5.2 展望 | 45 |
| 参考文献 | 47 |
| 致谢 | 53 |

插图

| | |
|--|----|
| 1-1 字符检测 | 2 |
| 1-2 自然场景下字符检测的示例 | 3 |
| 1-3 字符检测算法流程 | 4 |
| 1-4 字符识别 | 4 |
| 2-1 CTPN 的算法流程图 ^[1] | 8 |
| 2-2 DeepText 的网络结构图 ^[3] | 8 |
| 2-3 TextBoxes 的网络结构图 ^[4] | 9 |
| 2-4 RRPN 的算法流程图 ^[5] | 9 |
| 2-5 DMPNet 的候选框样例图 ^[6] | 10 |
| 2-6 SegLink 的算法流程图 ^[7] | 10 |
| 2-7 EAST 的网络结构图 ^[2] | 11 |
| 2-8 R ² CNN 的算法流程图 ^[8] | 11 |
| 2-9 CornerText 的算法流程图 ^[9] | 12 |
| 2-10 WordSup 的半监督字符级检测 ^[10] | 12 |
| 2-11 TextBoxes++ 的候选框回归详解 ^[11] | 13 |
| 2-12 FOTS 的算法流程图 ^[12] | 13 |
| 2-13 CTD 的算法流程图 ^[13] | 14 |
| 2-14 SLPR 的算法流程图 ^[14] | 14 |
| 2-15 TextSnake 的算法流程图 ^[15] | 15 |
| 2-16 PSENet 的算法流程图 ^[16] | 15 |
| 2-17 TextField 的算法流程图 ^[17] | 16 |
| 2-18 CRNN 的算法流程图 ^[8] | 16 |
| 2-19 RARE 的算法流程图 ^[18] | 17 |
| 2-20 R2AM 的算法流程图 ^[19] | 17 |
| 2-21 STAR-Net 的算法流程图 ^[20] | 18 |
| 2-22 GRCNN 的算法流程图 ^[21] | 18 |
| 2-23 ATR 的算法流程图 ^[22] | 18 |
| 2-24 FAN 的算法流程图 ^[23] | 19 |
| 2-25 Char-Net 的算法流程图 ^[24] | 19 |
| 2-26 AON 的算法流程图 ^[25] | 20 |
| 2-27 EP 的算法流程图 ^[26] | 20 |
| 2-28 SSFL 的算法流程图 ^[27] | 21 |

| | |
|--|----|
| 3-1 局部分割网络的结构图, C 和 DC 分别表示了连接操作和反卷积操作 | 23 |
| 3-2 MASK-RCNN 的流程图 ^[28] | 24 |
| 3-3 ResNet 残差结构 ^[29] | 25 |
| 3-4 ResNet 网络结构 ^[29] | 26 |
| 3-5 局部分割网络的候选框和分割示意图 | 27 |
| 3-6 分割合并的效果 | 28 |
| 3-7 SynthText 数据集效果展示 | 31 |
| 3-8 算法在 CTW1500 数据集上的效果展示 | 33 |
| 3-9 算法在 Total-Text 数据集上的效果展示 | 34 |
| | |
| 4-1 识别四阶段流程图示例 | 36 |
| 4-2 SEBlock 结构示例 ^[30] | 37 |
| 4-3 SEBlock 对 ResNet 的改进 | 37 |
| 4-4 SEBlock 对 Inception 的改进 | 38 |
| 4-5 字符识别数据集展示 | 39 |
| 4-6 IIIT 数据集识别正确的样例展示 | 40 |
| 4-7 IIIT 数据集识别错误的样例展示 | 41 |
| 4-8 ICDAR13 数据集识别正确的样例展示 | 41 |
| 4-9 ICDAR13 数据集识别错误的样例展示 | 43 |

表格

| | | |
|-----|--|----|
| 3-1 | CTW1500 数据集上不同弯曲字符检测算法的结果 | 32 |
| 3-2 | Total-Text 数据集上不同弯曲字符检测算法的结果 | 32 |
| 4-1 | 特征提取网络配置 | 42 |
| 4-2 | 特征压缩配置 | 43 |
| 4-3 | 识别在多个数据集上比较的结果 | 43 |

摘要

信息技术和网络通信的高速发展，为人们打开了一个全新的世界。但是在给人们生活带来便捷的同时，它也在不断地改变着人们的生活方式。信息高度发达的今天，使得物理距离不再是阻碍人们交流的鸿沟。与此同时，世界充斥着大量的数据交换。这些数据包括了语音、文本、图像、视频等各种形式。这些丰富的数据是一堆亟待开发的有效资源，等着人们进行合理利用。但是如何从海量数据中甄别出有效数据，并且进一步挖掘出有效数据的价值，一直是一个困扰众多研究者的难题。

在众多的数据资源中，图像数据由于其丰富的内涵以及直观的呈现方式成为了这些资源中的明珠。目前，使用计算机视觉技术理解图像内容一直是研究的热点问题，例如图像分类、目标检测等。同时，文字是一种承载思想，传递思想信息的重要工具，对文字的研究也一直未曾中断。本文研究的主要内容是自然场景的字符检测与识别。

图像字符检测的目的是定位图像上字符所在的位置。作为图像内容理解的基础任务之一，字符检测对于准确提取图像文字信息至关重要。本文从解决自然场景下任意形状字符检测的问题出发，提出了一种能够适用任意形状的字符检测框架，主要包括局部分割网络与曲线连接两大组件。局部分割网络得到文本区域具有高度重叠的水平矩形候选框和文字精细的轮廓，而曲线连接算法利用局部分割网络的输出结果，将多个候选框根据一定的规则进行合并，以此来拟合任意形状的文字。我们在两个公开的自然场景下的弯曲字符检测数据集进行了实验对比，以此验证了提出方法的有效性。

字符识别作为图像提取文字信息的另一个基础任务，要求能够很好的抗拒复杂的背景以及多变的文字。随着循环神经网络和空间变换网络的发展，开始解决弯曲文本的识别问题，本文基于现有的空间转化阶段、特征提取阶段、序列建模阶段和结果预测阶段的识别框架，同时借鉴了 Squeeze-and-excitation Block 的思路^[30]，优化了识别流程中特征提取阶段，使得识别结果有显著的提高。我们在多个公开的自然场景识别数据集上做了相应的实验，并将结果与其它的识别方法进行对比，证明我们的改进对提升识别精度的有效性。

关键字：自然场景，字符检测，字符识别，卷积神经网络

中图分类号：TP391

Abstract

The rapid development of information technology and network communication has opened up a whole new world for people. But while it brings convenience to people's lives, it is constantly changing the way people live. Today, with highly developed information, physical distance is no longer a barrier to communication. At the same time, the world is full of data exchange. These data include various forms such as voice, text, images, and video. These rich data are a bunch of effective resources to be developed, waiting for people to develop and use. But how to identify effective data from massive data and further explore the value of effective data has always been a problem that has plagued many researchers.

Among the many data resources, image data has become the jewel of these resources due to its rich connotation and intuitive presentation. At present, using computer vision technology to understand image content has always been a hot topic of research, such as image classification, target detection and so on. At the same time, writing is an important tool for carrying ideas and conveying ideological information. The study of words has not been interrupted. The main content of this paper is the text detection and recognition of natural scenes.

Text detection, in order to find the location of text, as one of the most basic tasks of extracting text information from images, is essential for extracting text information on images accurately. Aiming to solve the problem of arbitrary shape text detection in real-world scenarios, this paper proposes a text detection framework that can be applied to arbitrary shapes: partial segmentation network and curve connection. Through the local segmentation network, a horizontal rectangle proposal frame with a highly overlapping text region and a fine outline of the text are obtained. The curve connection uses the output of the local segmentation network to combine multiple proposal frames according to certain rules, and to fit the text of an arbitrary shape. We experimented with the curved text detection dataset in two open scenarios, and compared the results with the previous methods to prove the effectiveness of the method.

Text recognition, as another basic task for extracting text information from images, is required to be able to fit complex backgrounds and varied texts. With the development of cyclic neural network and spatial transformation network, how to identifying curved text is a problem. This paper adopts the Transformation stage, Feature extraction stage, Sequence modeling stage and Prediction stage framework at the moment, and draws on the idea of Squeeze-and-excitation Block^[30] to optimize the phrase of feature extraction, find out that improved the result significantly. We also performed corresponding experiments on open datasets at present, and compared the results with other recognition methods to prove the effectiveness of our method on improving recognition accuracy.

Keywords: Scene text; text detection; text recognition; convolutional neural networks

Abstract

CLC number: TP391

第1章 绪论

当今社会飞速发展，随着各种图像捕捉设备的发展以及图像传输技术的发展，人们每天都能产生以及传输大量的图像数据。我们无时无刻不在接收和发送的数据，但是如何提取海量数据背后相关的信息呢？目前比较普遍的方式是通过人工分析和提取图像内的信息，例如在海关行业中，报关员需要拿到发票的信息，然后手工地录入到系统中。如果一单单据中拥有大量的商品，往往需要消耗报关员大量的时间在录入信息上。这种方法效率低下并且人工成本过高。随着人工智能技术的发展以及在某些领域上的应用，使我们看到了机器去自动识别图像上的信息的希望，这也成为当下研究的一个热点。

图像上有很多丰富的信息，包括各种各样的目标以及各国的文字。目前有许多的研究是针对如何提取图像上的目标，比如目标检测，目标分割等。这些技术有助于提取图像里的物体特征。但是由于文字所独有的特点，与目标检测有一定的区别（例如文字具有较大的长宽比），通常的目标检测方法并不能很好的解决文字的检测识别问题，因此有大量的研究人员聚焦于如何更好的提取任意场景图片内的文字信息。

文字信息提取大体可以分为两个基础部分：字符检测和字符识别。

1.1 字符检测

字符检测的作用是框定图片上的文字区域。如图1-1所示，给定的一张包含文字的图片，定位里面所有的文字区域，如图1-1中的右图绿色框所示。框定区域的方式从最开始的水平矩形，随后为了能够更加贴合文字转变成带有角度的矩形，再到之后的任意四边形，最近演变成了使用任意的多边形去框定文字的区域。框定方法的不断变化不仅反映了文字形状、字体的多变，同时也体现了文字背景的日渐复杂。

文本是人类沟通和表达信息的最重要方式。提取图像的文本信息有助于我们理解图像所要表达的信息。字符检测作为图像字符识别的基本任务之一，是许多多媒体任务例如视觉分类和视频分析的重要先决条件。一个优秀的检测方法能够更加精准地确定文字的位置，同时解决各种形状的文字定位，为识别任务减少背景的干扰，降低字符识别的难度。例如当识别弯曲文本时，如果检测的结果能够更加贴近给出文字的区域，可以利用空间变换网络(STN)^[31]将文字转换成水平方向，更有利提高识别精度。

在卷积神经网络出现之前，字符检测大多使用了图像特征的信息，例如利用图像的直方图，边缘特征等方法。整个算法流程可以分为两个方向，分别为基于滑动窗口以及连通区域的两种方法。

滑动窗口的方法是事先定义一些候选框，然后在图片上根据一定的步长进行滑动。采集每一个窗口内的图像特征，利用特征判断窗口内是否含有文字。之后利用非极大值抑制等方法对多余候选框进行选取，最后确定最为合适的候选框。

连通区域的方法是根据算法计算出一些图像的特征（例如笔画的粗细），然后根据这些特征计算出连通区域，通过一些聚类的算法，最终确定检测的结果。



图 1-1 字符检测

不管是基于滑动窗口的方式还是基于连通区域的方式，都需要根据实际的图像进行大量的参数调整。当遇到一些比较严重的干扰时，检测结果往往会遭遇失败或出现混乱。

近年来随着卷积神经网络的发展以及通用物体检测的发展，许多研究者尝试将目标检测算法根据文字的特征进行细微的修改，从而达到字符检测的效果。字符检测早期在解决扫描文本上的字符时，由于扫描文本上大多字体都是水平方向的，因此关注焦点是如何确定水平文字的位置。后来字符检测被逐步应用于各种场景，著名字符检测的比赛 ICDAR，整理出了大量的各种场景下的字符数据集，例如 ICDAR2013^[32]，ICDAR2015^[33] 以及 total-text^[34] 等数据集，成为之后检验字符检测算法的基准数据集。再后来，提出了各种方向的文字以及轻微形变的检测算法（例如，在^[2, 5, 6] 中）。

1.2 字符识别

字符识别的作用是提取出图像上的文字。其中，字符识别的输入是由字符检测算法得到的文字区域，如图1-4所示。传统的识别算法将文字序列按字符切分，将每个字符输入一个分类器，最后通过后处理得到文字序列。传统识别算法的性能取决于切分方法的性能。字符级的切分对于不同的语言存在较大的差异。例如，对于汉字序列“树林”字来说，切分算法可能将其切分为“木”、“又”、“寸”、“木”和“木”，也有可能将其切分为“木”、“对”和“林”，因为汉字结构的多变性，字符级的切分存在较大的问题。对于英文来说，由于英文字符的间距较小，切分算法也难以切分。所以，基于字符切分的方法在识别领域并不能取得较好的效果。字符识别目前比较流行的算法有 CRNN^[35]，利用卷积网络提取特征，再利用循环神经网络学习图像上下文的信息。以及解决弯曲文本识别的 RARE^[18] 等。

现阶段，得益于深度学习技术的快速发展，目前最流行的字符识别算法将字符识别问题看成 sequence-to-sequence 的问题。该识别架构由空间转化阶段、特征提取阶段、序列建模阶段和结果预测阶段四个阶段组成，在多个公开的数据集上均取得了不错的效果。我们沿用了四个阶段的方式进行字符识别，并且对于特征提取阶段，我们借鉴了 SEBlock^[30] 的



图 1-2 自然场景下字符检测的示例

结构，提高了特征提取的质量，得到更好的识别效果。

目前某些算法采用了检测识别端到端的形式，将检测的结果直接输入识别中，并且检测和识别共用一个基础网络，在减少资源消耗的同时，提高了整个检测识别的速度。同时，利用识别到的信息，直接产生损失反馈到基础网络中，有助于更好的训练基础网络，达到更好的效果。例如论文 FOTS^[12] 就采用了这种模式，在 ICDAR2013 和 ICDAR2015 的数据集上都取得了很好的效果。

1.3 问题和挑战

随着大众审美的提高，各种新颖的文字设计出现在大街小巷，在表达信息的同时，也给人们带来了艺术的欣赏，例如各种曲线词，徽标以及各种形态的广告词等。这对字符检测的性能又提出了更高的要求，任意方向字符检测算法已经无法解决弯曲字符的检测。图1-2中示出了一些这样的样本图像，其中第一行是输入的图像，第二行是 CTPN^[1] 的检测效果，EAST^[2] 的效果是第三行以及本文的方法第四行。从图中可以看到，CTPN 在近似与水平方

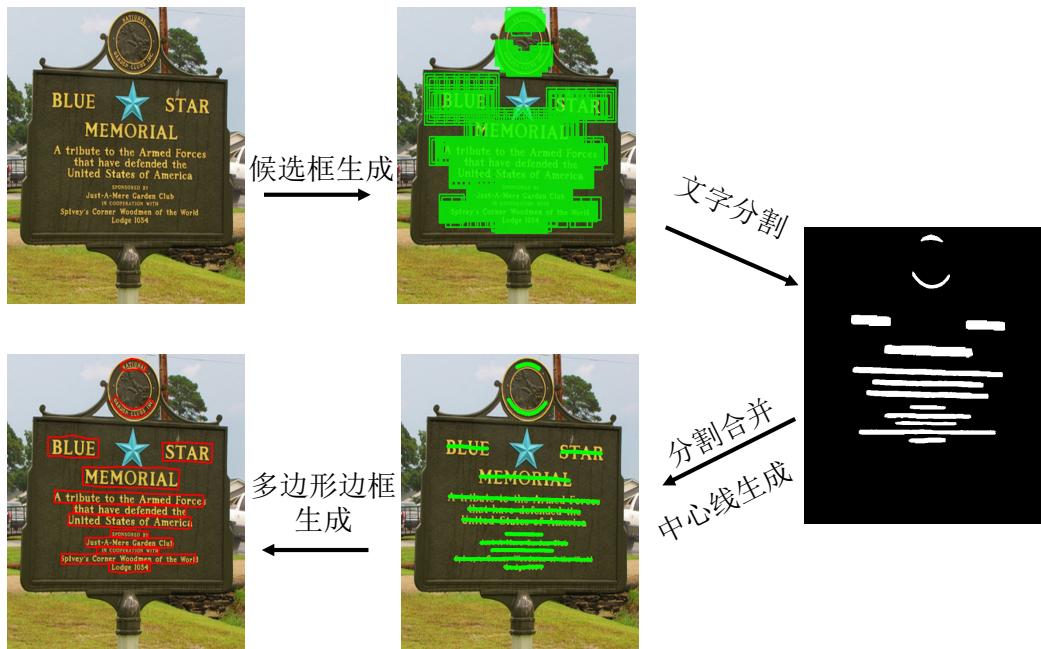


图 1-3 字符检测算法流程



图 1-4 字符识别

向的文字时效果很好，基本上都能够很好的覆盖文字区域，但是面对弯曲文本时，就会引入大量的背景或者直接漏检。EAST^[2]能够检测出的更多的文字，但是在长字上的效果不如CTPN的稳定，同时对于弯曲文本也是束手无策。图1-2的最后一行是本文算法的表现，可以发现检测的记过能够很好的拟合文字区域，得到更加精准的检测结果。同样，在字符识别方面，如何更好的确定任意形状文字的位置，成为字符识别的一大问题与挑战。

1.4 主要贡献

为了解决不同形状文字的定位问题，本文提出了一个结合局部分割网络（LSN）和曲线连接的新文本检测框架。流程如图 1-3 所示。LSN 的设计具有两个功能，即候选框的生成和文本区域的精确定位。由于 ResNet-50^[29] 较强的图像特征提取能力，我们选取它作为 LSN 的骨干网络。为了适应文本的不同比例，我们使用了不同大小的特征层进行候选框的产生。这些生成的候选框都是与文本高度重叠并且紧贴文本区域的正方形候选框，LSN 网络对候选框只进行分类操作，而不回归候选框的中心点坐标或者候选框的长度。本文通过 ROIAlign^[28] 将所有的不同大小的候选框提取的特征统一到同一个尺度，利用分割的方法确定精细的文本区域。算法的第二步是曲线连接，利用 LSN 输出结果生成中心线和文本区域多边形，最后得到检测结果。我们比较了通过分割区域直接生成检测结果和从 LSN 输出的结果进行区曲线连接的方法，证明了后者对算法的精度有很大的提高。为了评估我们提出

的框架的性能，我们在最近提出的曲线文本检测数据集上进行评估，即 CTW1500^[13] 数据集和 Total-Text^[34]，并且我们将我们的结果和最近的几篇关于弯曲字符检测方法进行对比。在仅仅采用训练数据集进行训练的情况下，以 TextSnake^[15] 作为对比，我们在两弯曲文本数据集上有一些提高。

为了解决识别不同形状字体的文字，我们采用了四个阶段的识别方法，及空间转化阶段、特征提取阶段、序列建模阶段和结果预测阶段。在空间转化转换阶段，我们使用了 STN 网络进行输入图像的空间转化，在特征提取阶段，我们使用了 ResNet 作为我们的基础网络，并且利用 SEBlock 的结构提升网络的性能。在序列建模阶段，我们使用 LSTM 获取文字的前后文信息。结果预测阶段使用 CTC 计算网络的损失。

本文的主要贡献如下：

- 提出了一种新颖的神经网络框架，结合局部分割网络与曲线连接的方式进行曲线文本检测。我们的想法是通过局部的分割确定文字的精准位置，然后通过全局的弯曲连接来实现检测不同尺度，不同形状的文字。
- 提出了利用局部分割信息，进行全局连接的策略，利用 LSN 产生的精细文字区域，来形成稳定的弯曲字符连接，通过产生中心线的方式更好确定文本区域的两个端点，形成更加精准的多边形边框，以提高长文本词和任意形状文本检测的性能。
- 提出了利用 SEBlock 的结构优化特征提取阶段的性能，使得识别的精度得到提高。
- 我们将框架应用于最新提出的两个自然场景下弯曲文本的检测数据集，并将我们的框架和已有的弯曲字符检测算法进行比较，在使用相同较少训练集进行训练时，能够具有更优的精度。同时我们比较了当前公开的字符识别数据集，修改后的四阶段识别框架比其它方法具有更高的识别精度。

第 2 章 研究现状

2.1 字符检测

字符检测作为图像文字提取的基础任务之一，一直以来都是研究的热点。从最开始的在扫描图像上检测出比较工整的文字，到现在开始检测艺术字，弯曲字体等各种变化的字体，对字符检测问题有提出了更高的要求。如何更好的确定各种形式字体的位置，以及如何区分艺术字体的特征，成为在自然场景下检测文字的难点。本节简要的描述了一下字符检测的发展，包含了从传统的图像特征的方法到现在的使用神经网络进行字符检测的方法。

2.1.1 传统方法字符检测

传统的方法大多聚焦在如何设计出特征，使得文字能够很好的和背景分开。然后根据一些聚类的方法，将相同特征的区域合并成一个区域后，作为一个文字区域输出。

SWT^[36] 利用的思想就是笔画宽度的特征，认为具有相同文字区域内的文字笔画的宽度是相似的，利用算法提取文字的边缘用，利用梯度算出文字笔画的宽度。然后通过一些聚类的方式，将相同笔画宽度的文字形成一个连通区域，作为检测的结果。

MSER^[37] 的思想是在灰度图中，一个文字区域内像素的值的变化幅度是很小的，这些区域成为 MSER。算法的流程大体上是通过不断的变化图像的阈值，判断哪些区域是 MSER 区域，然后通过聚类将 MSER 区域进行合并，最后生成检测结果。

Selective Search^[38] 的思想是先根据区域的特征，生成一个个比较小的连通区域。然后通过将相似的连通区域合并，逐渐得到比较大的连通区域，最后得到检测的结果。

传统的方法的好处是效率高，大部分图像处理的传统方法不需要进行的训练，在比较单一的字符检测上能够达到比较好的效果。但传统的字符检测的方法的缺点也很明显，需要调节大量的参数，基本上稍微有一些背景变化的图像适用的参数就不太一致。同时，在有很多干扰的情况下，传统的方法无法抗拒这些干扰，使得检测效果无法满足需求。

2.1.2 深度学习字符检测

随着深度学习的不断发展，在图像分类上取得了很大的成功。同时，随着目标检测的快速发展，出现了像 RCNN^[39]，Fast-RCNN^[40]，Faster-RCNN^[41]，Mask-RCNN^[28]，SSD^[42] 以及 YOLO^[43] 等目标检测算法，目标检测的精度产生了翻天覆地的变化。然而，与目标检测算法不同，字符检测需要精准的预测文字的文字，通常细微的差别就有可能导致检测的失败。因此，研究者们做了许多目标检测算法在字符检测上的改进，也取得了令人惊喜的效果。根据检测文字的形状的不同，我们可以大概将字符检测算法分为三个阶段：水平字符检测，四边形字符检测以及多边形字符检测。

水平字符检测

CTPN^[1] 提出了将文本检测看成一些小的候选框的序列，如图2-1所示，CTPN 使用了宽度固定为 16 的候选框，首先根据卷积神经网络提取图像的特征，然后将这些特征拉成一维的向量后放入了 LSTM 中，利用 LSTM 来提取上下文的信息，最后输出每一个候选框的是否是文字的预测分数，坐标以及调整的幅度，最后合并相邻的文字候选框作为文本行。CTPN 将文字作为一个序列，提高了字符检测的精度。

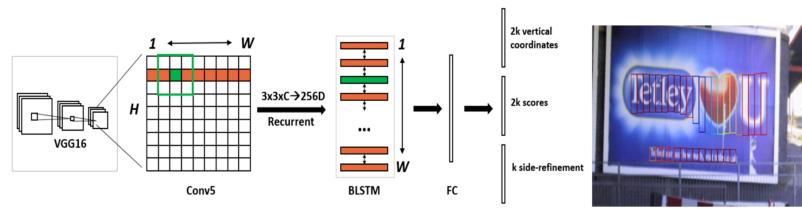


图 2-1 CTPN 的算法流程图^[1]

DeepText^[3] 对目标检测算法 Faster-RCNN 进行了三点的改进：1. 为了适应文字形状的候选框，Deep Text 将 Faster-RCNN 中的 RPN 改成了 Inception-RPN 的结果，可以更好的提取文字形状的特征。2. 为了更好的区分文字区域和背景，DeepText 在原始二分类的基础上，添加了一个模糊类，以此来更加精准的区分文字区域和非文字区域。3. 修改了原始的 Faster-RCNN 只用到了一层的特征提取候选框，DeepText 选用了最后两层的特征，以此来提取更加多样的文字特征。如图2-2所示，对 Faster-RCNN 的三点改进在字符检测中取得了不错的效果。

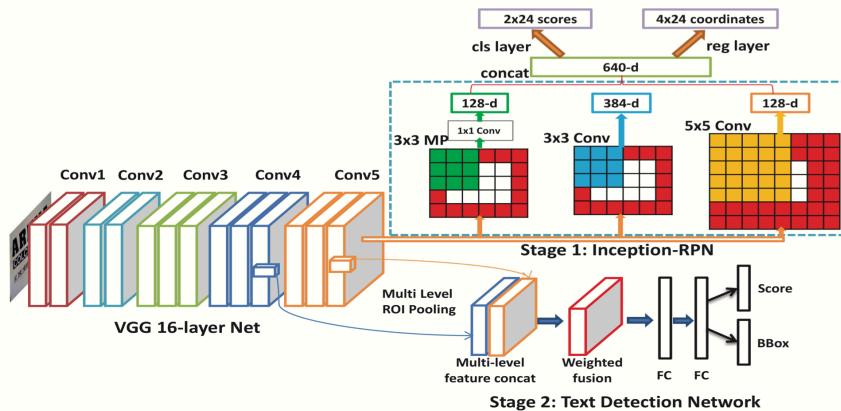


图 2-2 DeepText 的网络结构图^[3]

TextBoxes^[4] 是对目标检测算法 SSD 的改进，其中包括了 1. 根据文字的形状特征，重新设计了基础的候选框的长宽比。2. 修改了分类的卷积核，从 SSD 的 3×3 变成 1×5 ，提高分类的精度。3. 多尺度训练图像，让不同大小的文字都得到很好的训练，提高了模型的泛化性。4. 将检测后的结果接入到识别中，并且利用识别的结果更好的提升检测的效果。如图2-3所示，TextBoxes 最大的贡献点就是加入了识别结果，做到了端到端的字符检测识别。

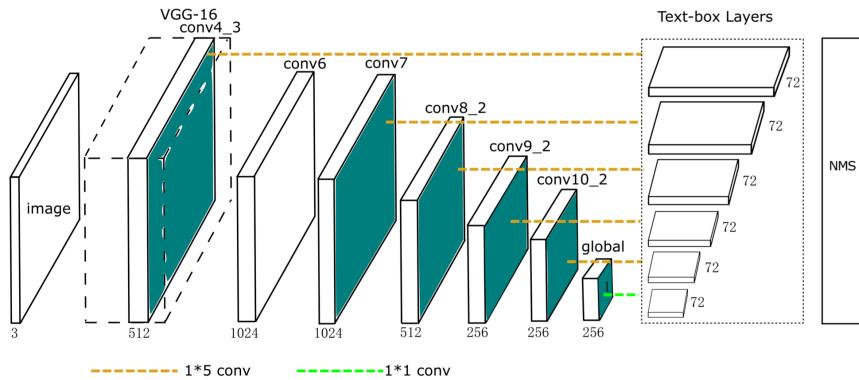


图 2-3 TextBoxes 的网络结构图^[4]

四边形字符检测

RRPN^[5] 提出了利用有角度的矩形框作为文字的候选框来检测任意角度的文字。如图2-4所示，在 Faster-RCNN 的 RPN 的基础上，加入了一个角度的参数，利用 5 元组 (x, y, h, w, θ) 表示，使得候选框变成有一定倾斜角度的矩形框。同时，为了提取出任意角度候选框的特征，提出了 RROI 的结构，将生成的候选框映射到由 VGG 网络产生的特征上，相比于原始的 ROI，能够更加准确的提取候选框的特征。RRPN 第一次提出了带角度的候选框，更好的拟合文字的区域，使得检测结果有很大的提升。

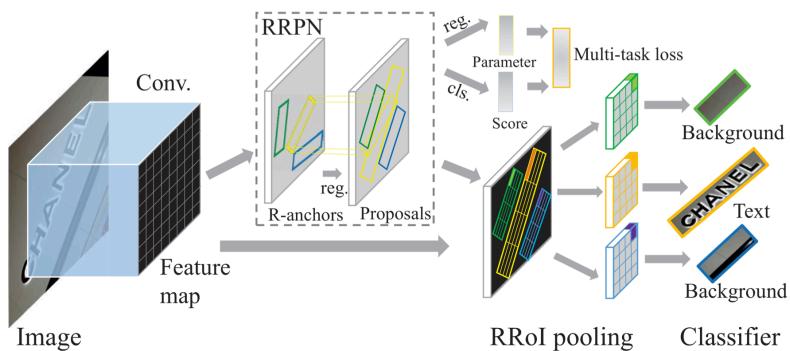


图 2-4 RRPN 的算法流程图^[5]

DMPNet^[6]修改了候选框的形状，如图2-5所示，DMPNet在原始的候选框中，分别在水平和竖直的候选框中添加了两个平行四边形，在正方形的候选框中添加两个 45° 的矩形框，提高候选框的多样性以此来获取更好的文字特征。在计算两个候选框的相交面积上，DMPNet提出了Shared Monte-Carlo的方法，该方法利用采样的思路，在两个候选框分别均匀采样，然后计算在两个相交面积上的采样点占所有采样点的比例，从而计算出需要的IOU。该方法比之前的方法更加的精准，并且可以计算任意四边形相交的面积。同时DMPNet设计了Smooth LN Loss，得到更好的检测效果。

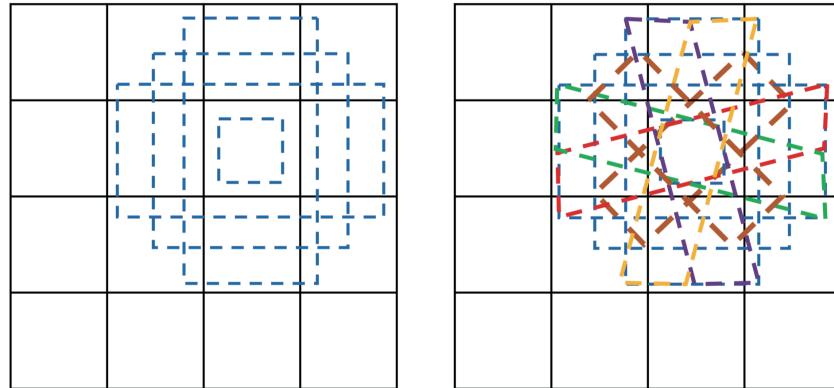


图 2-5 DMPNet 的候选框样例图^[6]

SegLink^[7]提出了分割和连接的思路，如图2-6所示，该方法分为两个部分，分割和连接。分割模块，借鉴了目标检测SSD的思路，利用VGG网络提取图像的特征，然后在多层的网络提取的特征图上分别预测出候选框，候选框由五个元素表示，分别为 x, y, w, h, θ ，分别表示候选框的中心点坐标 x, y ，候选框的宽度和高度 w, h 以及候选框的角度 θ 。连接模块利用网络去预测了每一个候选框与相邻候选框的关系，每一个候选框定义了在同一层特征上八个相邻的候选框的连接情况以及和前一层的四个候选框的连接情况。该方法利用网络去预测各个候选框的连接情况，然后利用这些连接情况组织成检测结果。具有较好的泛化性。

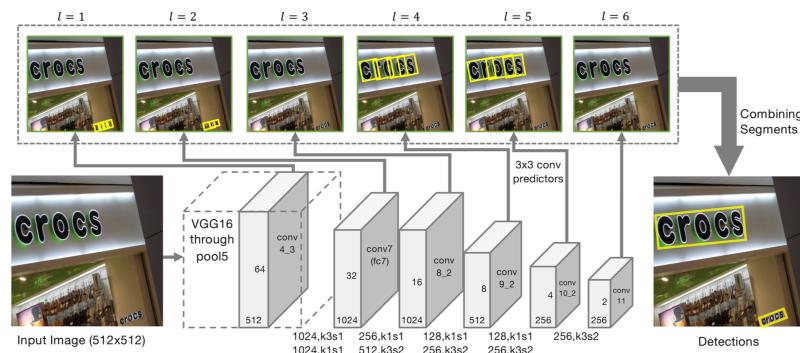


图 2-6 SegLink 的算法流程图^[7]

EAST^[2] 总结了前面的算法的流程，发现大多算法都需要好几个步骤，整个过程比较繁琐，而 EAST 算法流程只分为分割和 NMS 两个部分。如图2-7所示，分割网络借鉴了 UNet 的思路，连接不同的特征层后进行候选框的回归预测以及分类预测。EAST 设计了两种形式的候选框，一种是带有角度的矩形框，另外是任意的四边形。同时，在生成候选框的标签时，利用算法将文字区域缩小一定的比例，有助于更精准的预测文字的位置。在最后的 NMS 上，EAST 提出了 Locality-Aware NMS 算法，将算法的时间复杂度从之前的 $O(n^2)$ 变成了 $O(n)$ ，同时相比较与基础的 NMS，精度也有所提高。

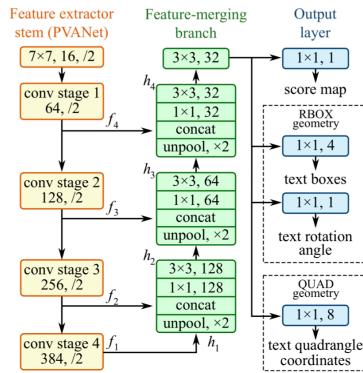


图 2-7 EAST 的网络结构图^[2]

R^2CNN ^[8] 的结构如图2-8所示，在 Faster-RCNN 的基础上，首先对 ROI-Pooling 进行了修改，分别利用 7×7 、 3×11 、 11×3 进行采样，可以更好的提取文字特征。其次，利用两个角点的坐标以及高度来确定文字的区域，这样可以生成任意角度的矩形框。同时，根据传统的 NMS 在字符检测上的缺点， R^2CNN 提出了 inclined nms，更好的解决字符 NMS 容易遗漏的问题。

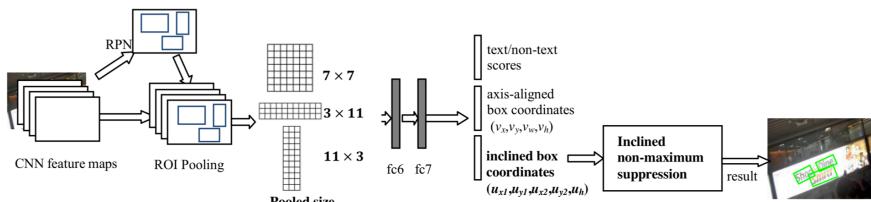
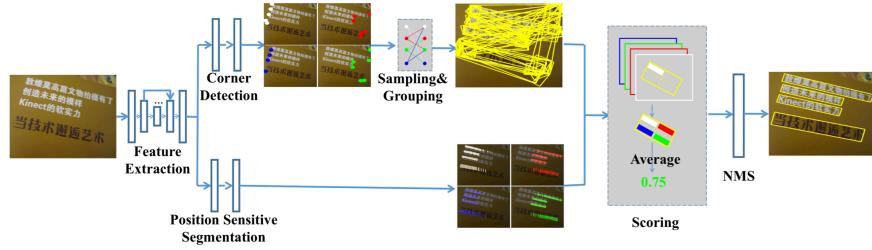
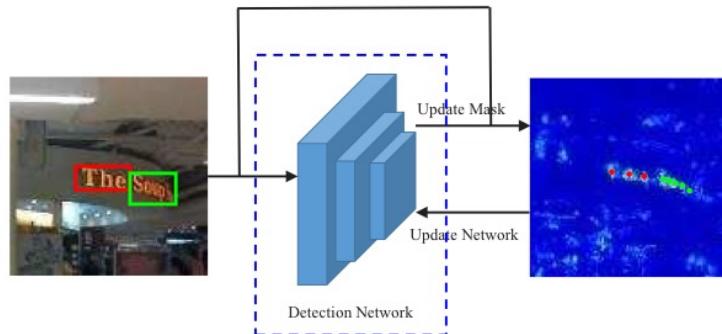


图 2-8 R^2CNN 的算法流程图^[8]

CornerText^[9] 采用了 VGG 作为基础网络，为了更好的使用文字不同的感受野，采用了 DSSD 的结果，如图2-9所示。CornerText 提出的想法是直接预测出文字区域的四个角点，这样同时也带来如果直接预测坐标点是否是角点在出现重叠的情况下无法判断当前角点属于哪一个文字区域，为了解决这个问题，作者提出了位置敏感的分割方法，每一个角点预测了四种类型点的偏移量。在优化 loss 方面，采用了 OHEM 的方式，最后通过后处理得出检测的结果。

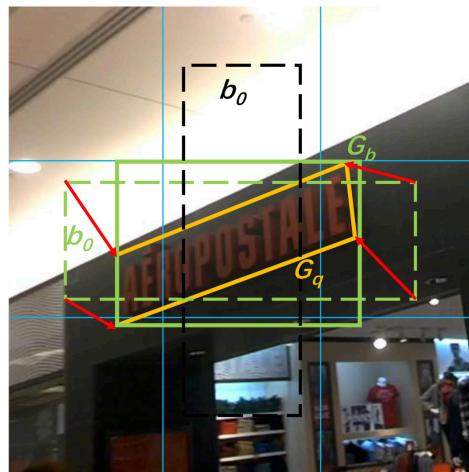
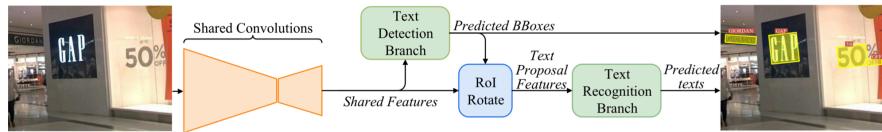
图 2-9 CornerText 的算法流程图^[9]

WordSup^[10] 提出了一种利用单词级标注来学习字符位置的方法，如图2-9所示。利用单词级标注的结果，通过检测网络生成检测结果的同时，生成字母级别的中心点分割图，通过半监督的方式，利用生成的分割图又反馈给检测网络，优化检测网络。

图 2-10 WordSup 的半监督字符级检测^[10]

TextBoxes++^[11] 是 TextBoxes 的一个改进，整体的结果与 TextBoxes 类似，不同的是 TextBoxes++ 在预测出文字区域的矩形框后，同时回归了四个角点的坐标，如图2-11所示，由此便可以检测任意的四边形的矩形框。

FOTS^[12] 提出了一种端到端的检测识别方法，如图2-12所示，首先利用网络进行检测，随后利用检测结果在原来的特征图上进行 RoiRotate，将得到的特征输入到识别网络中，直接得到识别的结果。在训练的过程中使用识别的结果进行梯度的反传，更好的优化基础网络，使得检测的结果得到提升。

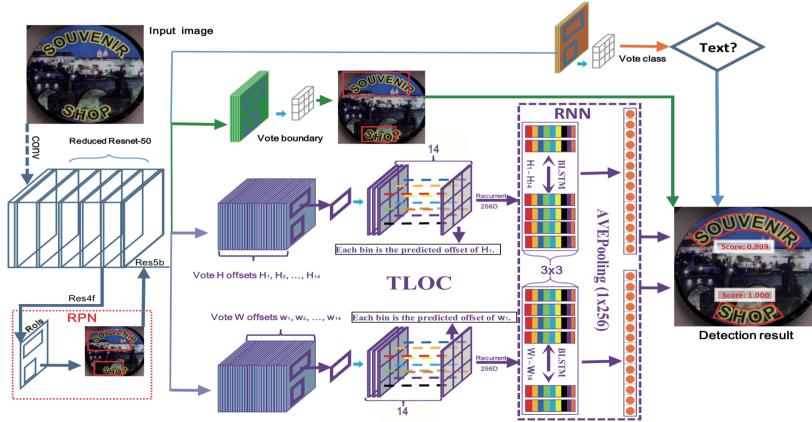
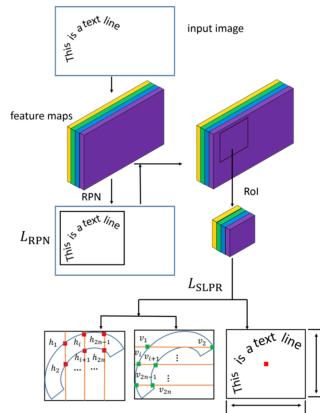
图 2-11 TextBoxes++ 的候选框回归详解^[11]图 2-12 FOTS 的算法流程图^[12]

多边形字符检测

CTD^[13] 首先搜集了一个专门针对检测弯曲文本的数据集 CTD，该数据集包含了 1500 张图片其中 1000 张为训练集，500 张为测试数据集，包括了 10,751 个文本框，其中 33% 为弯曲文本。在标注数据上，CTW 数据集利用 14 个点代表了任意多边形，其中上、下边界各由 7 个点表示。在提出数据集的基础上，论文设计了弯曲文本检测的算法 CTD，网络结构如图2-13所示，整体的过程和 Faster-RCNN 相似，利用 RPN 提取候选框，然后利用每一个候选框中，分别预测 14 个坐标点的高和宽的偏移量，作者发现 14 个点各自预测出来形成的曲线不够光滑，因此在预测出结果后输入到双向的 LSTM 中，得到更加光滑的曲线。

SLPR^[14] 类似于 Faster-RCNN 的思路，首先利用 RPN 产生存在文字的候选框。然后在候选框的基础上，等间距的确定 n 个位置，分别在这 n 个位置预测出两个点，表示文字区域的上下边界，如图2-14所示。得到一些可以确定文字区域的坐标点后，利用不同的算法分别对多边形和矩形进行拟合，最终得到检测结果。

TextSnake^[15] 提出了任意形状文字的表示方法，利用一系列的圆以及相应的角度可以

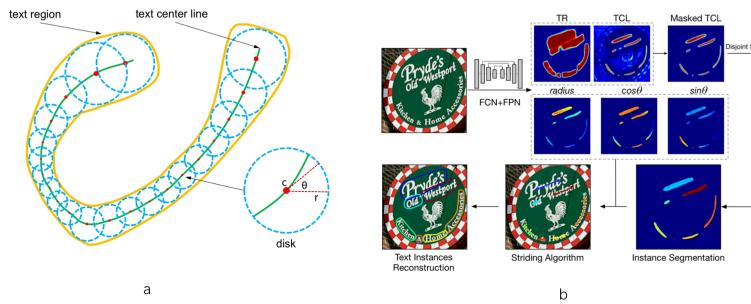
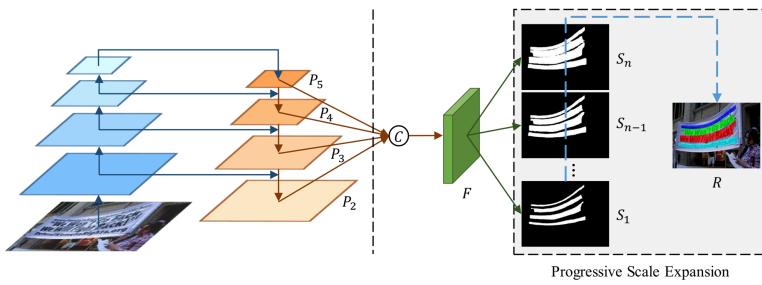
图 2-13 CTD 的算法流程图^[13]图 2-14 SLPR 的算法流程图^[14]

表示任意的任意形状的文字，即，如图2-15a所示。算法的整体流程如图2-15b所示，首先输入的图片通过FCN和FPN后，分别得到文字中心线的分割图，文字区域的分割图以及每一个圆的半径和角度的sin, cos值。最后根据这些结果，进行后处理后输出检测的区域。

PSENet^[16]提出了一种类似于膨胀的算法，如图2-16所示，整体的思路还是基于图像分割的思路。利用不同的分割尺度，从最小的尺度开始，不断扩大分割的范围，最后通过后处理形成最终检测的结果，该方法思路简单，得到比较好的检测结果。

TextField^[17]基于图像分割的思路，首先对图像进行文字区域的分割，区分出文字的区域。为了解决分割的黏连文字问题，TextField设计了在文字区域的每一个点预测出向中心点的回归方向，如图2-17所示，然后利用这些回归方向去区分相邻的文字区域。

从传统方法到现在深度学习的方法，字符检测的情况越来越复杂，检测算法的鲁棒性也越来越强，检测的效果也越来越好。目标检测算法对字符检测有很大的影响，例如Faster-RCNN的出现，使得字符检测出现DeepText, SLPR等算法。字符检测深度学习的算法可以大体归纳到两个大类中，第一种是基于候选框的方法，另外一种则是用分割的方法。目前后一种方法成为主流，不但检测的效果更加好，而且能够较为容易的检测任意形状的文字。

图 2-15 TextSnake 的算法流程图^[15]图 2-16 PSENet 的算法流程图^[16]

本文结合了候选框的方法和分割的方法，利用候选框粗略的筛选出文字的位置，然后使用分割的方法精细的确定文字的位置，之后设计曲线连接的方式，形成任意文字的检测结果。在下一章中我们将详细讲解我们的检测方法。

2.2 字符识别

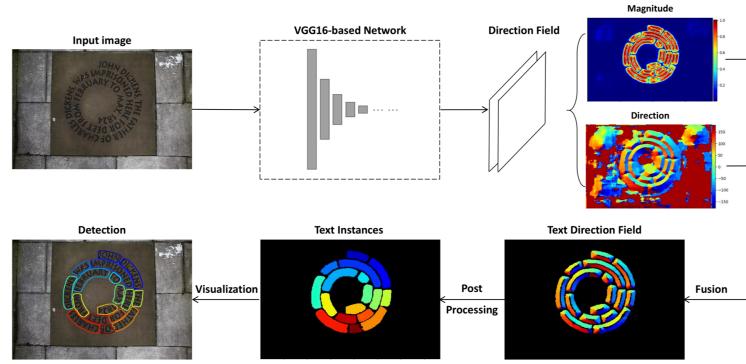
字符识别一直是计算机视觉方向的一项重要研究方向。大多国家都拥有自己的语言，这也提高了字符识别任务的复杂程度。本节简单的描述一些比较知名的字符识别算法，包括传统的字符识别以及使用深度学习方法的字符识别。

2.2.1 传统方法字符识别

Wang et al.^[12] 使用了 HoG 特征去训练一个分类器来区分文字与背景，之后使用滑动窗口的方式将所有的字母连接层单词。

Neumann and Matas^[44] 利用支持向量机进行分类，设计来一些列低纬度的特征，例如长宽比例，开孔面积等。

Strokelets^[45] 设计来一套特征生成算法，来生成能够区分文字的特征，以此来达到识别

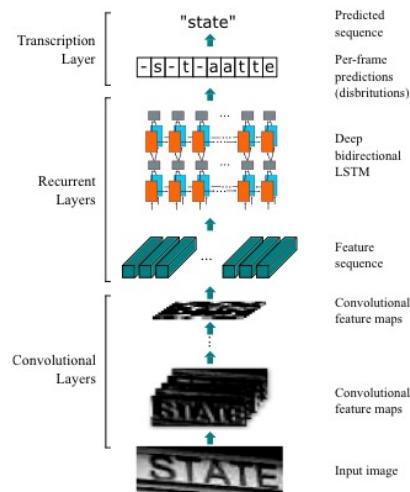
图 2-17 TextField 的算法流程图^[17]

文字的效果。

传统的字符识别算法需要很多手工设计的特征，这些特征当遇见场景比较复杂的情况下就容易出现错误，而且对于差异较大的情况下，手工设计的特征就很难满足识别的需求。

2.2.2 深度学习字符识别

CRNN^[35] 分为三个部分，卷积层，循环层以及转换层。如图2-18所示，卷积层的作用是对于输入的图像进行特征的提取，有卷积操作和池化操作组成。循环层的作用是将图像转化为序列，在提取上下文信息的同时，能够处理任意长度的文本，作者在该层使用来双向的 LSTM 进行特征转化。在转换层使用来 CTC 的损失，将循环神经网络的输出转化成最终的结果。利用动态规划进行加速后，得到概率最大的标签序列。

图 2-18 CRNN 的算法流程图^[8]

RARE^[18] 模型实现对不规则文本端到端的识别，RARE 由两大部分组成，分别是 STN(Spatial Transformer Network) 以及 SRN(Sequence Recognition Network)。STN 处理来任意形状的文字，利用空间变化，将输入的不规则文本进行矫正，得到形状规则的水平文本。SRN 如图2-19所示，是一个基于注意力机制的网络结构，包含了编码和解码两个部分。编码部分类似与

CRNN，解码部分是由注意力机制的循环网络。STN 和 SRN 的组合实现 sequence to sequence 的文本识别。

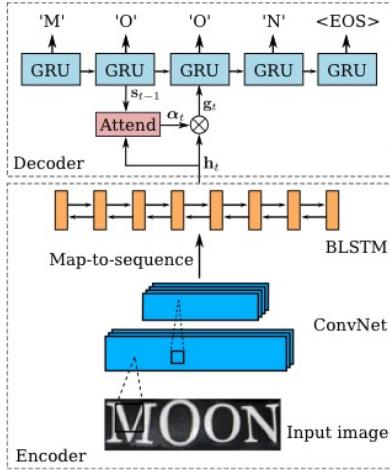


图 2-19 RARE 的算法流程图^[18]

R2AM^[19] 提出了一种带有注意力机制的递归循环网络模型，如图2-20所示，R2AM 有两个部分组成，第一部分图像特征提取部分，在该部分中，为了提高模型获取上下文的范围，及提高网络的感受野，使用了递归神经网络，在参数量相同的情况下，增加了网络的深度。第二部分是字符级别的文字建模，采用了循环神经网络和注意力机制的方法，使得模型更好的结合图像特征与语言统计。

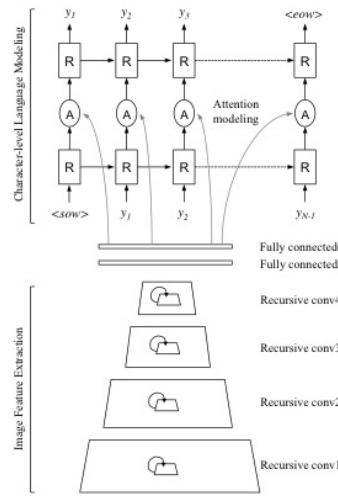
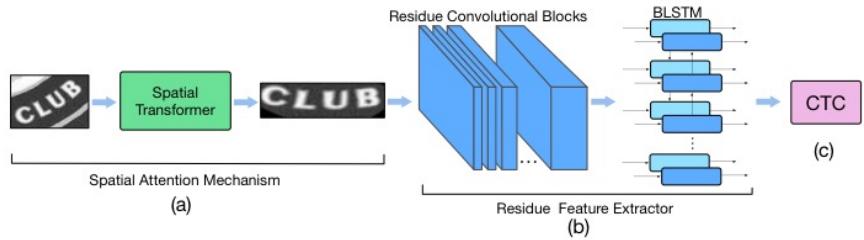


图 2-20 R2AM 的算法流程图^[19]

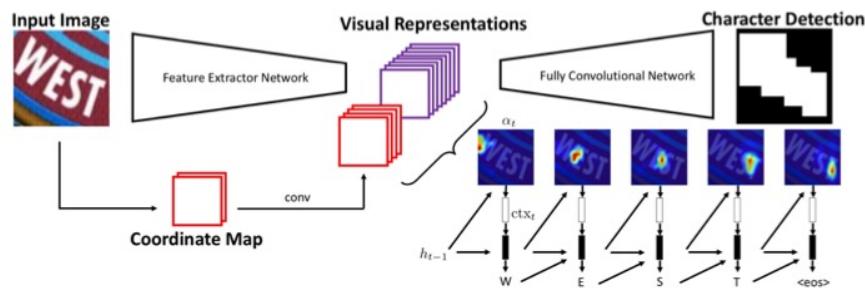
STAR-Net^[20] 采用了空间注意机制，对于任意输入形状的文字进行空间转化，使得后续特征提取器聚焦在规整的文本区域上。STAR-Net 还利用了残差卷积结构来构建一个非常深的特征提取器，这对于成功提取这种细粒度识别任务和判别性文本特征至关重要。如图2-21所示，STAR-Net 结合空间注意机制和残差卷积结构，实现了一个能够端到端的字符识别模型。

图 2-21 STAR-Net 的算法流程图^[20]

GRCNN^[21] 是对 RCNN 算法的一种改进，提出了 GRCNN 的新架构。如图2-22所示，GRCNN 设计了一个门来调节算法 RCNN 中的循环连接，从而能够动态的选择上下文信息，并灵活地将前馈网络部分与循环网络部分相结合。将与识别内容不相关的上下文信息使用门结构进行隔离。之后的算法与 RCNN 类似，使用了双向的 LSTM 产生标签序列，形成识别的结果。

图 2-22 GRCNN 的算法流程图^[21]

ATR^[22] 提出了一种不用通过 STN 网络预处理的弯曲字符识别方法，如图2-23所示，ATR 网络利用全卷积网络设计了一个字母级的检测结构，利用字母级的检测算法，确定每一个字母的位置，从而达到识别弯曲文字的效果。其它的结构与 CRNN 类似，使用卷积网络提取图像的特征，在通过循环神经网络输出识别的结果。

图 2-23 ATR 的算法流程图^[22]

FAN^[23] 提出了注意力机制的一个问题，注意力的区域大多没有完全的契合目标文字的位置，即注意力偏移问题，解释了注意力机制在某些图像上糟糕的效果。为了解决这个问题，研究人员提出了许多改进方法，例如引入多头注意力机制、使用掩码或硬注意力等。

题, FAN 提出了全新的模块: 聚焦网络, 纠正注意力机制产生的偏移, 如图2-24所示。FAN 以 ResNet 作为特征提取的网络, 丰富图像特征的表示。

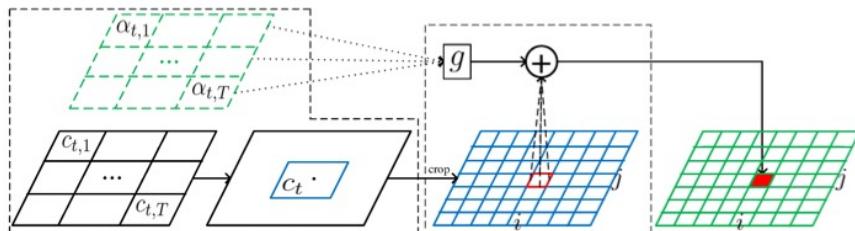


图 2-24 FAN 的算法流程图^[23]

Char-Net^[24] 与 ATR 类似, 流程如图2-25所示, Char-Net 结合了单词级编码器, 字符级编码器和 LSTM 的解码器。为了解决弯曲文本的识别, Char-Net 提出了分层注意力机制 (HAM), HAM 包含了两个部分, 第一部分为循环 RoIWarp 层, 该层从单词级编码器产生的特征映射中依次提取对应于字符的特征区域, 并将其反馈到单词级编码器。第二部分是字符级的注意力层, 该层利用字符级的编码器产生的特征组成上下文向量, 最后将上下文的特征送入到 LSTM 解码。

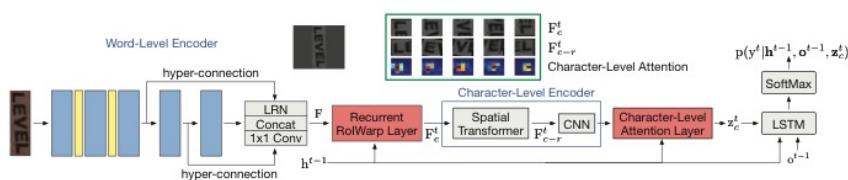
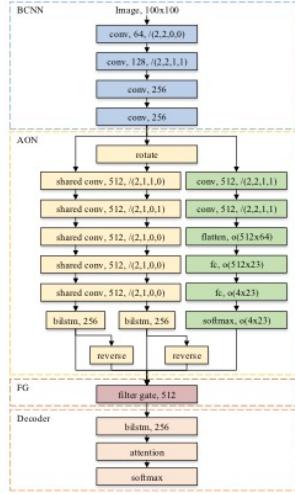
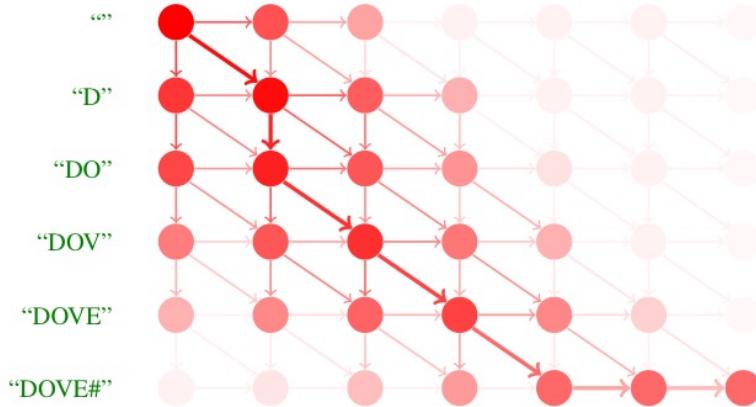


图 2-25 Char-Net 的算法流程图^[24]

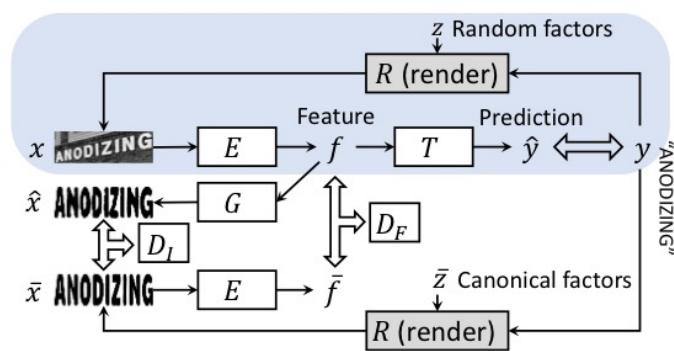
AON^[25] 的提出是为了解决弯曲字体的识别问题, 之前的算法对文字的处理都是相同方向的处理方式, 如从左到右的方式。AON 提取的四个方向的文字编码, 利用特征长度相同的四个方向的特征表示文字。同时设计了一个滤波器开关, 来融合四个方向的特征。AON 算法结果如图2-26所示, BCNN 负责提取基础的图像特征, 最后利用 LSTM 得到结果序列。

图 2-26 AON 的算法流程图^[25]

EP^[26] 提出现有技术基于注意力的编码器-解码器框架下的场景文本识别问题。现有方法通常采用逐帧最大似然损失来优化模型。当我们训练模型时，标签与注意力的概率分布输出序列之间的不一致将混淆和误导训练过程，从而使训练成本高昂并降低识别准确度。为了解决这个问题，EP 提出了一种新的方法，称为编辑概率（EP），用于场景文本识别。EP有效地估计从输入图像上的概率分布到输出序列生成字符串的概率，同时考虑可能出现的丢失或者多余字符。EP 的计算过程如图2-27所示，该方法的优点在于训练过程可以集中在缺失的、多余的和未识别的字符上。

图 2-27 EP 的算法流程图^[26]

SSFL^[27] 的方法核心是通过生成一张干净背景的识别图去辅助字符识别。如图2-28所示，提出了一套具有编码器-发生器-鉴别器-解码器架构的新型多任务网络，该网络通过干净的图像去知道网络获取更好的图像特征，提高识别的精度。

图 2-28 SSFL 的算法流程图^[27]

第3章 任意形状的字符检测

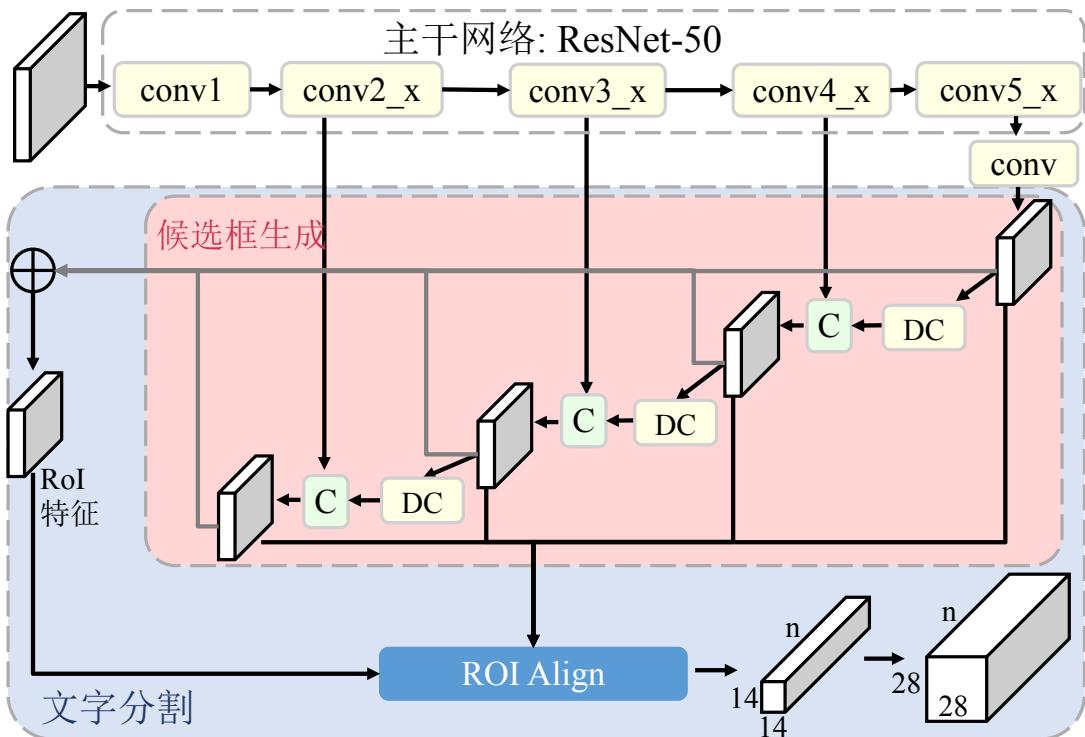


图 3-1 局部分割网络的结构图，C 和 DC 分别表示了连接操作和反卷积操作

为了能够检测出任意形状的文字，并且结合候选框方法和分割的方法的优点，本文设计了一种基于局部分割和全局连接的方法，首先利用候选框在图像上粗略的筛选出文字的位置，这一步可以帮助我们过滤掉大量的背景，减少后期分割的区域。随后在每一个单独的候选框中，进行图像分割，得到精细的分割结果。本节将详细的介绍整个算法的流程以及实现的细节。

3.1 算法流程

算法的流程如图1-3所示。总体可以分为三个步骤，第一步是候选框生成，第二步是分割结果生成，第三步是后处理。首先对于输入的一张图片，利用候选框生成算法生成所有的候选框，每个文字区域都是由密集的候选框覆盖。如图1-3第二张图所示。随后将每一个候选框送入到分割网络，得到精细的文字的区域的分割图像，如图1-3的第三张图所示。在得到比较精细的文本结果后，利用分割的结果将同一个文字区域的候选框聚集成一个集合，随后利用主曲线回归算法生成的中心线。再由中心线等间距获取七个点，利用这七个点来精细的生成文字区域。下面就每一个模块进行详细的讲解。

3.2 局部分割网络

目前大多的利用图像分割方法做字符检测直接使用原图进行分割，这样会出现大量的背景，造成分割的浪费。MASK-RCNN^[28]中先利用候选框大致的筛选出目标的位置，然后利用分割网络切分出精细的物体的边界。但在目标检测中，大多检测的问题都比较接近正方形，字符检测却存在许多长宽比很大的目标。因此如果依旧是利用 RPN 的思路无法很好的覆盖所有的文字情况，比较难解决文字的长度比较大的情况。为了能解决任意形状，任意长度的字符检测问题，我们采用了拼接的方式，首先利用 RPN 获取一系列的正方形的水平候选框，然后将这些候选框送入分割网络得到精细的文字区域，因此我们设计了一个局部分割网络。

3.2.1 网络结构

目标检测算法 MASK-RCNN 的流程如图3-2所示。MASK-RCNN 的大体流程是，先产生目标物体的候选框后，利用更加精细的 ROI-Align 提取每一个候选框的特征，利用提取的特征进行图像分割，得到物体的轮廓。

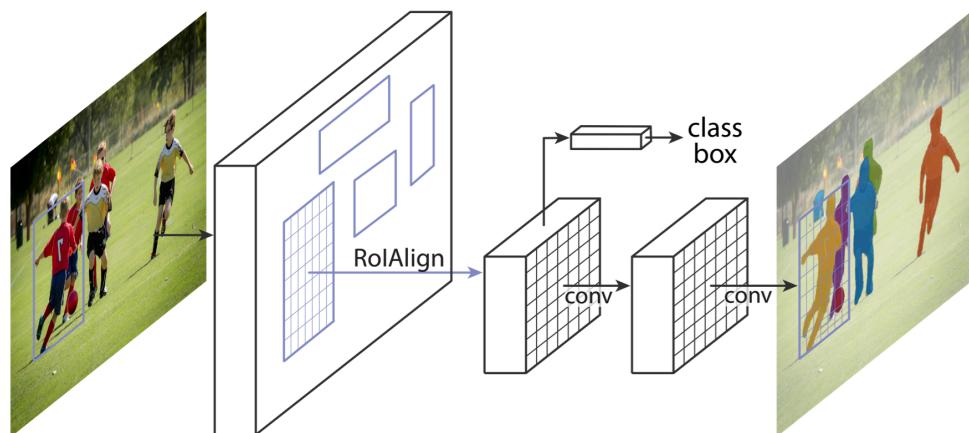
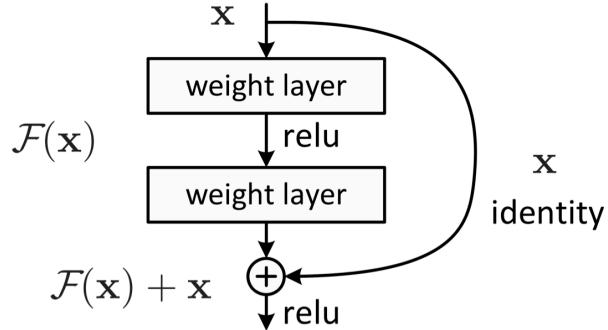


图 3-2 MASK-RCNN 的流程图^[28]

残差网络（ResNet）^[29]设计了 Shortcut Connections 的结构，为了解决梯度消失等问题，同时也将低层的信息添加到高层中，在图像分类上取得了比较好的结果。残差结构如图3-3所示。利用这些残差结构，构建出了残差网络，如图3-3所示，很大程度上增加了网络的深度，具备了更好的性能。至今有很多的任务依旧使用了残差网络作为基础网络，得到更好的基础特征。

借鉴 MASK-RCNN 的网络结构，我们设计了局部分割网络的网络结构。网络结构如图3-1所示，局部分割网络的骨干网络选用了 ResNet50^[29]，具体结构见图3-4以此来捕获更精细的文字特征。此外，我们删除了基础网络 ResNet50 的最后一个全连接层，并连接了额外的卷积层，以获得具有更大感受野的更深层特征。

图 3-3 ResNet 残差结构^[29]

在候选框生成部分，网络结构如图3-1红色部分，为了能够检测不同大小的文字，学习FCN^[46]的思路，我们将骨干网络得到的结果不断利用反卷积和卷积分别得到不同大小的特征层，最大的特征图是原图的1/8。为了更好的融合底层和高层的特征，学习了FPN^[47]的思想，我们将ResNet四个block的输出特征与反卷积出来的结果进行拼接，得到更好的特征表示。对于每个特征合并部分，我们使用了具有 1×1 内核的卷积层使两个合并特征具有相同的维度。同时，四个不同大小的特征层分别预测候选框，能够更好的覆盖图像上所有的文字，达到更优的检测效果。为了减少回归问题对精度的影响，与其它候选框方法不同的是我们只对候选框进行了分类，确定候选框中是否含有文字，而不做其它的预测。候选框的选择详见下一小节。

在分割部分，我们将四层的特征合并成一个特征后，使用ROIAlign^[28]来获得具有不同尺寸的正方形的相同尺寸特征。然后我们对提取的特征进行反卷积操作，将特征的大小从 14×14 变成了 28×28 ，获取更加精准的分割结果。

3.2.2 候选框生成

不管是目标检测的Faster-RCNN还是字符检测的候选框的方法，在生成候选框的时候，他们都需要预测候选框的偏移，例如DeepText需要预测 x, y, w 与 h 的偏移，RRPN需要预测 x, y, w, h 和 θ 的偏移，然而偏移总是存在着误差，无法很精准的预测出文字的位置，例如RRPN中，如果角度 θ 的预测出现了一些偏差，就有可能导致检测结果不正确。

因此，在候选框生成的过程中，我们只进行了分类任务，如图3-5顶部所示，任何文字区域都可以有若干个正方形候选框覆盖。我们通过 (x, y, l) 定义每个正方形候选框。其中 x, y 表示正方形的中心坐标， l 表示正方形的边长。我们假设从网络中提取的特征图的大小是 w 和 h 。候选框坐标与特征图之间的关系可以表示为以下等式：

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|------------|-------------|---|---|---|--|--|
| conv1 | 112×112 | | | 7×7, 64, stride 2 | | |
| | | | | 3×3 max pool, stride 2 | | |
| conv2_x | 56×56 | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ |
| | 1×1 | | | average pool, 1000-d fc, softmax | | |
| FLOPs | | 1.8×10^9 | 3.6×10^9 | 3.8×10^9 | 7.6×10^9 | 11.3×10^9 |

图 3-4 ResNet 网络结构^[29]

$$\begin{aligned} x_i &= i \times \text{stride}, i \in \{0, 1 \dots w - 1\} \\ y_j &= j \times \text{stride}, j \in \{0, 1 \dots h - 1\} \end{aligned} \quad (3.1)$$

$l \in \{s \times k | s = 8, 16, 32, 64. k = 2, 2.5, 3, 3.5\}$ 是可以覆盖几乎所有文本比例的正方形的宽度，其中 s 代表特征的步长， k 代表了候选框的边长对于步长的倍数。整个网络只做了分类任务，因此对于每一种候选框，预测的通道数量为 2，分别代表非文字和文字的概率。每一层特征预测 4 种类型候选框，因此每一层的候选框预测的输出的通道数为 8。

3.2.3 文字分割

和 MASK-RCNN 类似，在第一步获取文字的大致位置后，我们在候选框生成分类之后预测每个正方形的分割结果。我们利用基础网络产生的特征和反卷积产生的特征进行融合，之后将这四层的特征通过采样的方式变成相同的分辨率，利用叠加的方式融合这四层的特征，之后将上一阶段产生的候选框与融合的特征进行 ROI-Align 产生 $n \times 14 \times 14$ 维度的特征，其中 n 代表了候选框的个数。在得到每一个候选框的分割结果后，为了更加精准的进行尺度较大文字分割，我们利用反卷积的操作增加了特征的大小，得到更加精细的分割结果，文本分割处理的结果如图 3-5 底部所示。

3.2.4 损失函数

LSN 网络总共有两个输出，第一个输出是候选框的分类结果，第二个是每个候选框的分割结果。因此我们的损失函数可以表示为：

$$L = \lambda_1 L_{\text{cls}} + \lambda_2 L_{\text{segment}}, \quad (3.2)$$

其中 L_{cls} 和 L_{segment} 分别代表文本候选框的分类损失和分割损失，其中 L_{cls} 我们使用了交叉熵损失函数，而 L_{segment} 使用了 smooth L1 计算出。 λ_1 和 λ_2 是两个损失的参数，目的是为了平衡 L_{cls} 和 L_{segment} 之间的重要性。



图 3-5 局部分割网络的候选框和分割示意图

3.3 弯曲连接

在我们得到了每个候选框以及相应的分割结果后，如何将这些局部的结果合并成一个文字区域，变成现在要解决的问题。

在合成文字的区域时，第一种简单的方式是将所有的分割结果直接合并到原图中，然后在原图中利用寻找连通区域确定每一个文字的区域。这种直接找相邻区域的方法效果如图3-6，左边：原始图片以及标注结果，中间：通过直接寻找分割区域的方法，右边：通过弯曲连接的方法得到的结果。存在以下问题：

1. 在文字密集的区域，各个文字区域之间的间隔非常的小，分割的方法常常会有一些像素的差距，这将会导致两个文字区域合并成一个区域，导致检测结果不符合要求。
2. 在边缘区域容易产生误差，无法很准确的定位边缘的具体位置，该方法确定边缘采用的是采样的方法，如果采样的点不够精准，就容易出现一些文字区域丢失的情况，造成精度的损失。

针对以上的问题，在本节中，我们提出了利用分割结果进行曲线连接的方法。



图 3-6 分割合并的效果

3.3.1 分割区域连接

由于候选框只能大概的确定文字的位置，为了能够更加精准的拼接候选框，我们采用了分割结果作为我们的拼接依据。

首先，对于每个预测的分割结果，进行二值化处理，我们设置阈值 s_1 以定义文本的精细区域的位置。当像素预测得分高于 s_1 时，我们设置像素值 1，否则为 0。为了合并同一个文字区域内所有的候选框，我们设计了分割合并方法，伪代码显示在算法1中。我们使用队列 Q 存储彼此分开的文字区域的外接矩形框已经分割结果。从最开始遍历所有的候选框，计算每一个候选框与 Q 中所有的已经合并区域进行比较，为了提高算法效率，我们先利用候选框与 Q 中所有的区域进行是否相交的判断，如果两个区域不相交，则将新的区域加入到队列 Q 中。如果两个相交，这需要判断两个区域分割结果的重合度。如果这两个区域的重叠分割结果与较小区域分割结果的面积比例高于阈值 s_2 ，则合并这两个区域，然后重新判断队列 Q 中是否有其它需要合并的区域，否则就将这个区域的信息放入到队列 Q 中。重复最后两个步骤，直到合并所有区域。

3.3.2 多边形文字边缘生成

在分割区域连接后，我们获得一些文字区域。为了能够更加精准的预测出文字的区域，预测出文字区域两个端点准确的位置，我们设计了多边形文字边缘生成，整个算法分为两个部分。

第一部分：文字区域中心线生成。对于每一个文字区域，我们选取了 n 个正样例的像素点，并使用主曲线算法^[48]来回归曲线中心线，该算法不仅能够回归出任意形状文字的中心点，还能够确定出长边的位置。

第二部分：多边形边缘生成。从第一部分生成的中心线选择七个点，第一个点和最后一个点为中心线的两头，其余五个点为等间距选取。对于在中心线中相邻的每对点，我们使用两个点的中心点作为矩形中心，并利用分割结果以及寻找轮廓的方法找到局部区域的外接矩形，然后将四个角点记录为文字区域的边界点。通过重复上述步骤获得文字区域的多边

Algorithm 1 Mask Merging for a given image.**Require:** Predict mask set S ; threshold s_1, s_2 **Ensure:** Merged mask queue Q

```

1:  $Q = \{\}$ 
2: for each predict mask  $p \in S$  do
3:   if  $p_{i,j} > s_1$  then
4:      $p_{i,j} = 0$ 
5:   else
6:      $p_{i,j} = 1$ 
7:   end if
8:   overlap list  $L = \{\}$ 
9:   for each item mask  $m_i \in Q$  do
10:    merged mask  $m = m_i \cup p$ 
11:    overlap ratio  $r = area_m / \min(area_m, area_p)$ 
12:    if  $r > s_2$  then
13:      insert  $i$  into  $L$ 
14:    end if
15:   end for
16:    $m = p$ 
17:   for each  $i \in L$  do
18:      $m = m \cup Q_{L[i]}$ 
19:     delete  $Q_{L[i]}$ 
20:   end for
21:   insert  $m$  to  $Q$ 
22: end for
23: return  $Q$ 

```

形。算法的细节显示在算法2中。

3.4 字符检测数据集

字符检测的发展离不开背后数据的支持，近年来举办过一系列的关于字符的比赛，从而产生来许多知名的数据集。包括了 ICDAR2011^[49], ICDAR2013^[32], ICDAR2015^[33], SVT^[50], MSRA-TD500^[51], Total-Text^[34], SynthText^[52] 等数据集，下面是对一些训练集的大概介绍。

3.4.1 任意方向数据集

ICDAR2013 ^[32] 数据集来自 ICDAR2013 举办的比赛。其中有 229 个自然场景图像用于训练，233 个自然场景图像用于测试。此数据集中的所有文本实例都是水平对齐的。因此大多用来验证水平的字符检测算法。

ICDAR2015 ^[33] 是 ICDAR2015 比赛时所用的自然场景文本检测数据集，该数据集用于检测任意角度四边形文本行的训练和测试。它包含 1000 个用于训练的图像，500 个用于测

Algorithm 2 Text Polygon Generation.

Require: Merged mask set M for a given image

Ensure: Text polygon set P for the image

```

1:  $P = \{\}$ 
2: for each proposal mask  $m \in M$  do
3:   Polygon  $p = []$ 
4:   choose  $n$  positive pixels
5:   use principal curve to find curve center line
6:   choose 7 points  $p_i$  ( $i = 0...6$ ) to represent center line
7:   for  $i \in [0, 6]$  do
8:     generate circumscribed rectangle of the area between  $p_i$  and  $p_{i+1}$ 
9:     regard rectangle points as polygon points
10:    insert the left two points into  $p$ 
11:   end for
12:   insert the  $p$  into  $P$ 
13: end for
14: return  $P$ 
```

试的图像。标注的标准为单词级别的标注，一些太小或者模糊的文字标注为不关心字符安。近些年来字符检测算法都将 ICDAR2015 作为基准数据集。

MSRA-TD500 ^[51] 包含来 300 张训练数据以及 200 张测试数据，数据集是由室内和室外组成，室内主要是一些标志，门牌等，而室外主要是复杂背景下的广告牌或者导向牌。该数据集包含了中文，英文以及符号等。

SynthText ^[52] 数据集是一个利用自然场景图像和单词组成的数据集，主要运用于自然场景领域中的文本检测，如图3-7所示，能够比较自然的将文字组合到背景图片中。SynthText 默认的数据集由 80 万个图像组成，大约使用了 800 万个合成的单词。SynthText 通过合成的方式生成了大量的数据，能够很好的提升字符检测的性能。大多算法在训练模型时，会将网络在 SynthText 进行预训练，得到较好的初始参数。之后在到真实数据集上调优，可以得到更好的效果。

3.4.2 任意形状数据集

CTW1500 ^[53] 包含 1500 张图像（1000 张训练集和 500 测试集），其中包括倾斜文本，水平文本和任意形状文本。CTW1500 的标注比较精细，利用 14 个点定位一个文字区域，其中上边界 7 个点，下边界 7 个点，第一个点为左上角，其余的点按照顺时针方向给出。该标注的标准是一行文字，有些原本应该分开的数据被合并成一个区域，标注的质量存在一些问题。不区分中英文。

Total-Text ^[34] 不仅包含水平和多向文本实例，还包含弯曲文本。该数据集由 1255 个训练图像和 300 个测试图像组成。图像通过具有 $2N$ 个顶点 ($N \in \{2, \dots, 15\}$) 的多边形在单词的级别上进行标注。整个数据集包括了中英文，但是数据集的要求是只检测英文字符，对于中文或者一些看不清楚的文字都视为忽略字段，不统计准确率。

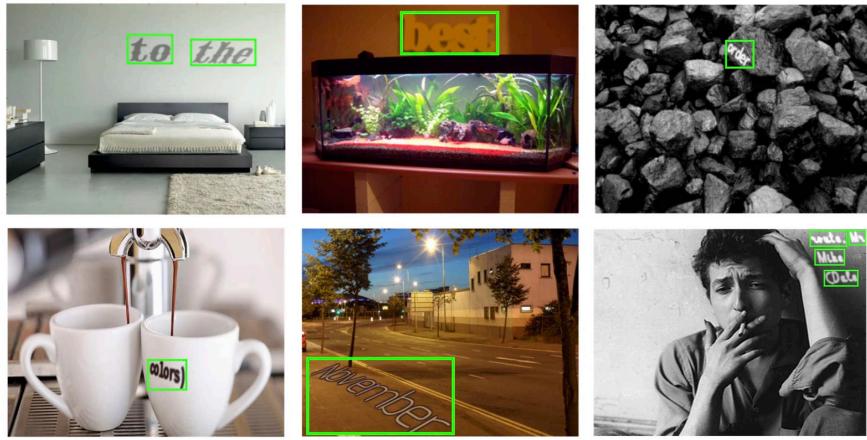


图 3-7 SynthText 数据集效果展示

3.4.3 评价指标

为了验证我们算法在弯曲字符检测的能力，我们使用最近提出的两个曲线文本检测数据集来评估我们的方法，分别是 CTW1500^[13] 和 Total-Text^[34] 数据集。两个数据集都有大量的各式各样形状的文字，包括了任意方向以及形状的文字，并且目前弯曲字符检测的算法大多都在这两个数据集上进行了验证，可以很好的验证算法的性能。

字符检测的指标通常是由三个数值组成，第一个是精度，第二个是召回率，第三个是 F-measure。假设一张图片上有 N 个目标文字区域，我们网络预测出了 M 个文字区域，其中 P 个区域是预测正确的，那三个衡量指标的计算公式如下：

$$\begin{aligned} recall &= P \div N \\ precision &= P \div M \\ F - measure &= (recall + precision) \div (recall \times precision) \end{aligned} \quad (3.3)$$

3.5 实验结果和比较

3.5.1 候选框标签生成

借鉴了 Faster-RCNN 中的 RPN 的思路，假设特征尺寸为 $W \times H$ ，我们的方法将生成 $W \times H \times 4$ 候选框。对于每个文本多边形的标签，我们定义了以下规则来判断候选框是否包含文字区域。

1. 候选框的中心点在文字区域内，并且候选框的高度不大于文字区域高度的 1.8 倍，CTW1500 数据集上文字区域高度通过计算 7 对定点距离的平均值得到，在 Total-Text 数据集上这是通过确定文字区域的两端线段，算出平均的高度。

2. 候选框有任意对角线上的两个点在文字区域的外侧。

当以上两个条件都满足的情况下，我们认为该候选框为正样本，否则认定为负样本。

表3-1 CTW1500数据集上不同弯曲字符检测算法的结果

| Method | Precision | Recall | F-measure |
|---------------------------|-----------|--------|-----------|
| CTD ^[13] | 74.3 | 65.2 | 69.5 |
| CTD+TLOC ^[13] | 77.4 | 69.8 | 73.4 |
| SLPR ^[14] | 80.1 | 70.1 | 74.8 |
| TextSnake ^[15] | 67.9 | 85.3 | 75.6 |
| LSN | 69.0 | 75.7 | 72.2 |
| LSN+CC | 83.2 | 78.8 | 80.8 |

表3-2 Total-Text数据集上不同弯曲字符检测算法的结果

| Method | Precision | Recall | F-measure |
|----------------------------------|-----------|--------|-----------|
| Total-Text ^[34] | 40.0 | 33.0 | 36.0 |
| Mask TextSpotter ^[53] | 69.0 | 55.0 | 61.3 |
| TextSnake ^[15] | 82.7 | 74.5 | 78.4 |
| LSN+CC | 82.4 | 76.9 | 79.5 |

3.5.2 数据增强

为了提高模型的泛化性，我们在训练时做了一些数据增强。在旋转问题上，我们对于输入的任意一张图片，我们随机的按照旋转 30° 的倍数进行调整，同时，为了能够提高数据文本比例的不同，我们随机将文本框的比例从 0.33 到 3 之间随机变换，并且随机进行左右反转。在这些之后，我们进行随机的剪切和模糊处理。在随机剪切上，为了保证切出来的图片都包含本来完整的文字区域，我们限定了随机裁剪的范围，例如固定好每次如果裁剪左边的话，就只能裁剪到最左边的文字区域的坐标的最小值。

3.5.3 模型训练

我们整个网络的实现使用的是 pytorch 深度学习框架，同时使用了 opencv 的图像处理库以及 numpy 等 python 库。我们使用 Adam 作为我们网络的优化器，与 SGD 相比，Adam 能找到更优的结果。在整个训练过程中，我们对于每一个特征层分别选取了 200 个正样本候选框作为 ROI-Align 的输入。在测试阶段，我们设置 $s_3 = 0.4$ 作为判断候选框是否为正样本的阈值，同时规定了每一层的候选框的正样本个数不能超过 2000 个。

3.5.4 实验结果和比较

在训练阶段，我们只用到了原本数据集的训练数据，并未加入其它的数据进行预先训练。所有的测试工具使用数据集自带的测试代码。IOU 的阈值为默认值 0.5。以下是在两个数据集上的检测效果。

CTW1500。为了验证算法的能力，我们在将算法与目前做弯曲字符检测的算法进行比较，方法包括 CTD 和 CTD+TLOC^[13] 利用 14 个边缘坐标点来表示弯曲文本的区域。SLPR^[14] 通过长与宽的回归来确定文字的区域，而 TextSnake^[15] 通过一系列的有序的，相互之间相交的圆来表示任意形状的字符。表3-1 展现了本文的方法和以上方法在精度，召回率以及 F-

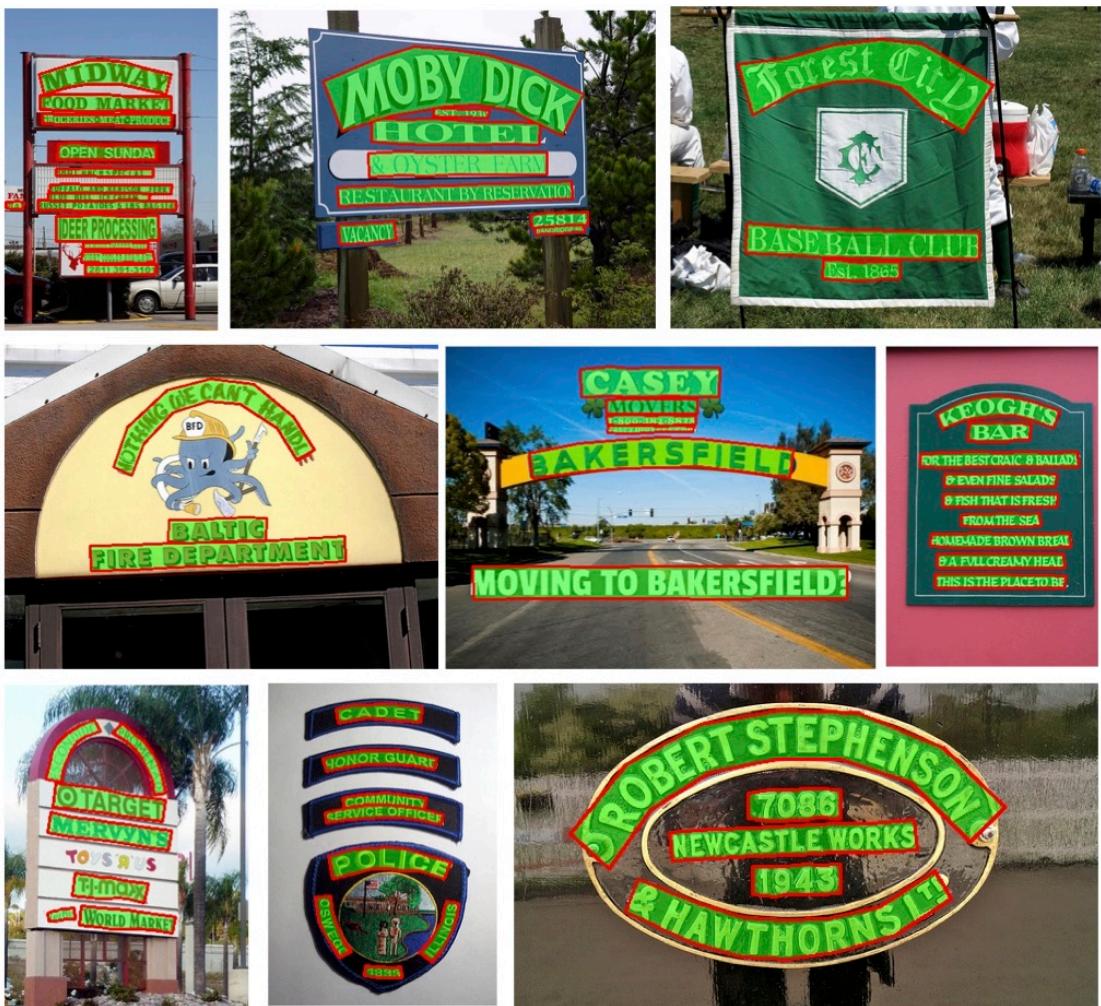


图 3-8 算法在 CTW1500 数据集上的效果展示

measure 上的差异，LSN+CC 表示的是完整的网络结果，局部分割网络与弯曲连接，LSN 代表了网络的输出没有包括弯曲连接。从实验结果可以看出，我们的弯曲字符检测算法在 CTW1500 数据集上达到世界领先的结果。同时，我们验证的曲线连接对算法的帮助，表中的 LSN 代表的是原始的算法，直接通过寻找区域的方式获取文字的结果，LSN+CC 是通过弯曲连接的方式产生文字区域。从结果中我们可以看到，弯曲连接不仅提高了检测模型的精度，同时也显著提高了模型的召回率。如图3-8展示了部分算法在 CTW1500 数据集上的效果。

Total-Text。表3-2展现了整体算法与其它算法之间的结果比较，比较的方法包括了 Total-Text^[34], Mask TextSpotter^[53], 以及 TextSnake^[15]。这些结果验证了通过局部分割和全局连接进行任意形状字符检测的有效性。一些在 total-text 数据集上的检测效果可以见图 3-9。可以看到我们的算法能够很好的拟合任意形状的字体。

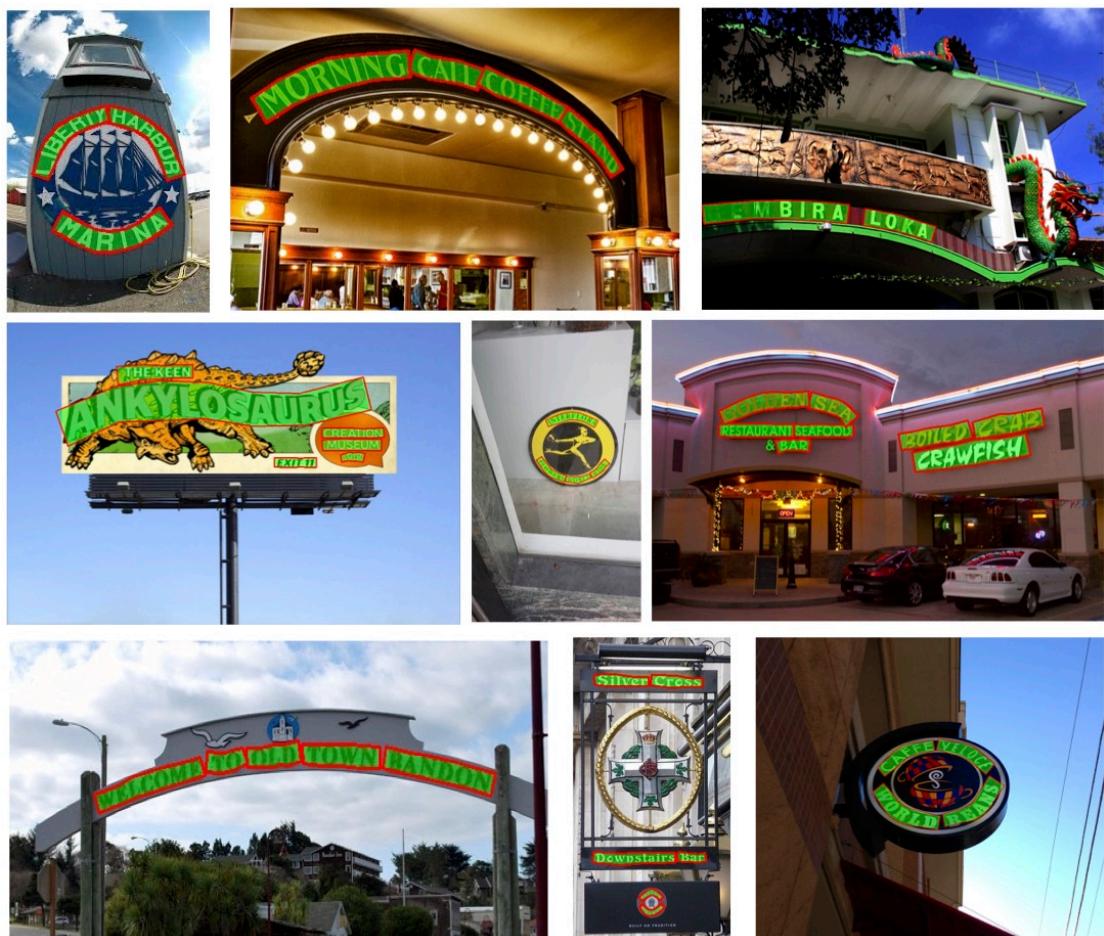


图 3-9 算法在 Total-Text 数据集上的效果展示

第4章 任意形状的字符识别

为了识别弯曲的文本，我们借鉴了 TPS-ResNet^[54] 的思路，采用了四个阶段的字符识别算法，即空间转化阶段、特征提取阶段、序列建模阶段和结果预测阶段。利用空间转化，将输入的任意形状的文字转化为工整的水平方向的文字。利用特征提取阶段，从图像中提取识别需要的特征，特征提取阶段是识别的重要步骤，将直接影响最终的识别结果。序列建模阶段，为上下文进行时序建模。结果预测阶段利用设计好的损失函数，进行网络优化，并且输出最终识别结果。为了提高识别架构的性能，我们在特征提取阶段对网络做了一些改进，使得网络能够自动调整特征的每个通道。

4.1 算法流程

四阶段算法分别由空间转化阶段、特征提取阶段、序列建模阶段和结果预测阶段四个阶段组成，如图4-1所示，四个阶段依次连接，形成端到端的架构。接下来我们将详细介绍一下四个阶段的作用以及相应的方法。

空间转化阶段：从检测的发展可以看出，文字区域的表示方法不断的改变，从水平的矩形框，到有角度的矩形框，到任意四边形再到多边形，可以看出文字排布的多样性。这些问题也影响这识别的准确率。目前解决任意形状字符识别的方法有两个大方向，第一个是输入图像不变，直接在识别网络中加入一个类似检测的小网络，通过这个小网络确定文字的具体位置。第二种方法是送入网络之前做图像的处理，将存在扭曲，倾斜等图片进行纠正。第一种方式的优点是流程简单，将文本形状问题交给网络解决，缺点是会增加识别网络的负担，后续的特征提取器需要学习图片弯曲，倾斜等特征。第二个方法的优点是网络不需要关心文本的形状，专注于识别图像上的文字，缺点是作为预处理的算法要足够的鲁棒，否则后续的网络无法更好的学习特征。综合两个方法的优缺点，我们选择了进行预处理的方法。为此，在空间转化阶段，我们采用了 STN^[31] 的一个变种 TPS (Thin Plate Spline)^[55] 对输入的图片进行处理。TPS 提供一系列的基础点，学习基础点与目标点之间的变换关系，从而图像变换与目标图像匹配。

特征提取阶段：在特征提取阶段，一般都采用了卷积神经网络来提取文字图像的特征，包括文字图片的字体、颜色、大小及背景等多种特征。由于卷积操作存在空间不变形等特点，特征图的每一列特征都代表着原图片的一部分。因此，每个特征向量对应于原始图像中的矩形区域，并且矩形区域与对应特征映射的从左到右列的顺序相同。目前的特征提取网络大多借鉴图像分类的网络。图像分类是指在给定图像的基础上，判断图像的类别。图像分类有一个著名的比赛 imageNet^[56]，imageNet 开始于 2009 年，每年都吸引了大量的参赛者。其中产生了大量优秀的特征提取网络，例如 VGG^[57] 网络，VGG 对 Alexnet^[58] 进行了改进，利用更小的卷积核达到更深的网络结构。ResNet 设计了残差结构，增加网络深度的同时避免了梯度消失问题。GoogleNet^[59] 使用了 Inception 的结构，得到了更加深层的结构。本文使用在图像分类任务中取得较高精度的 ResNet，利用 ResNet 的 50 层网络，提取图片的文

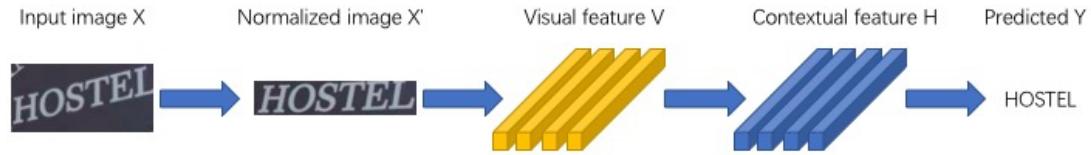


图 4-1 识别四阶段流程图示例

字特征。在 ResNet 的基础上，借鉴 SENet，我们将残差结构转变成 SEBlock，提高特征的质量，详见4.2。

序列建模阶段：在文字识别中，通常输入的图片都是包含一个完整的单词，因此利用好文字的上下文信息可以很好的提高文字的识别效果。循环神经网络在自然语言处理领域取得了很大的成功，其有效的利用了上下文的信息，在词性标注，句子语法检验取得了很大的成功。LSTM^[2]作为循环神经网络中的特殊类型，提出了利用三个“门”结构，解决了记忆长期问题的同时，解决了循环神经网络梯度消失和权重爆炸的问题，使得网络能够更好的选择需要结合的上下文信息。类似的结构还有门控循环单元（GRU）等。因此在序列建模阶段，为了更好的抽取文字的上下文，我们采用循环卷积神经网络（BiLSTM^[60]）来建立特征序列之间的关系，以此来提升识别的准确率。

预测阶段：输入的图片经过特征提取阶段和序列建模阶段，可以建立图像字符特征间的概率分布。然而通过序列建模后，由于上下文的信息，容易使时序信息出现较多的冗余。导致识别出许多重复的字符，因此如何区分是由于时序信息冗余产生的多余字符还是本身重复字符成为预测阶段需要解决的问题。CTC 把从序列建模阶段提取的特征重整，计算出最优的路径使得最终得到的识别结构概率最大。面对计算多条路径最优结果的时间问题上，CTC 采用了动态规划的思路来节省重复计算的次数从而提高时间效率。

4.2 特征提取改进

Squeeze-and-Excitation(SE) block^[30]是一个网络的基础结构，简称 SEBlock，SEBlock 的结构如图4.3所示，整个结构分为两个部分：

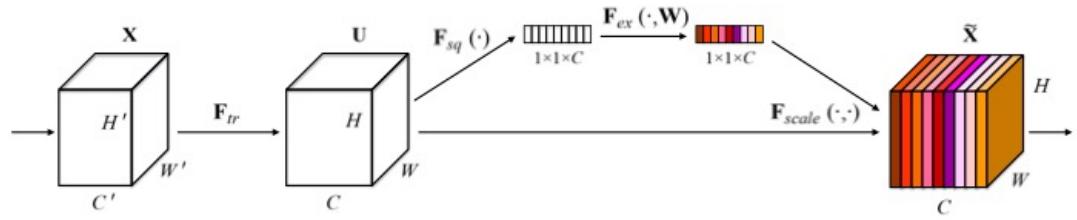
压缩部分：图中 F_{sq} 表示压缩操作，使用的是全局平均池化操作，将输入的中间特征层转化为一个 $1 \times 1 \times C$ 的特征向量，将全局的特征映射到低纬度。压缩部分可以表示成公式4.1：

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j). \quad (4.1)$$

其中 W 和 H 代表特征的大小， u_c 代表了特征的值。

提取部分：将经过 F_{sq} 操作后的特征向量输入到 $F_{ex}(W)$ 中，在特征向量的维度没有变的前提下，更新了特征向量的值。 $F_{ex}(W)$ 可表示为公式4.2，其中 δ 代表了 ReLU 激活函数。之后将输出的权重作用到特征 U 中，以 $F_{scale}(.,.)$ 表示，公式如4.3，其中 $F_{scale}(.,.)$ 代表了特征与权重的点乘。最终得到加权后的新的特征。

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(W_2 \delta(W_1 \mathbf{z})), \quad (4.2)$$

图 4-2 SEBlock 结构示例^[30]

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c, \quad (4.3)$$

SEBlock 能够在许多的网络中应用,如图4-3和4-4分别为SEBlock修改Inception和ResNet的基础卷积结果的详细情况。

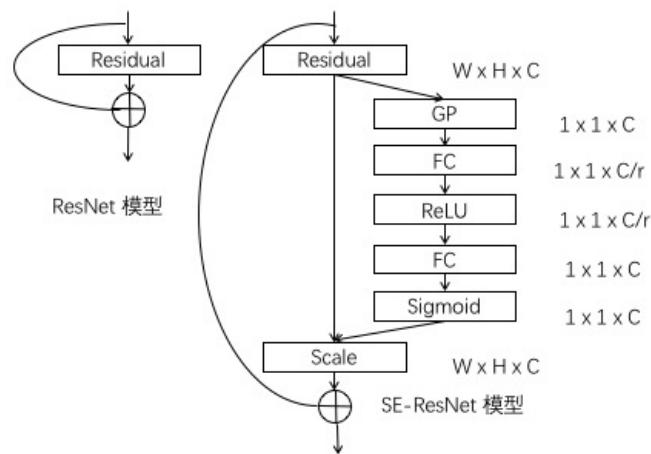


图 4-3 SEBlock 对 ResNet 的改进

SENet 为了解决不同特征之前的权重如何分配问题,核心思想在于通过网络根据损失去学习特征权重,重新编排特征的权重,使得对重要的特征的权重变大,无效或不重要的特征权重减小,以此训练模型达到更好的结果,这个流程可以表示成公式4.4。因此,我们利

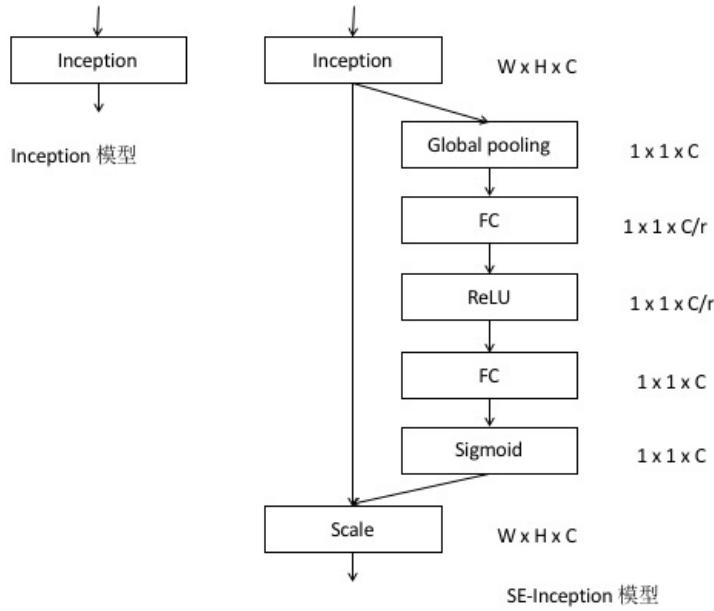


图 4-4 SEBlock 对 Inception 的改进

用 SENet 的思想，将 SEBlock 加入的特征提取阶段，希望能够将识别任务相关的特征通道权重加大，无关的特征通道减小，以此来优化特征提取阶段，最终提高识别的效果。

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s. \quad (4.4)$$

4.3 识别算法网络结构

在特征提取的网络结构中，我们以 ResNet 作为基础的模型，通过 SEBlock 结构增强不同通道的特征。该结构总共有 29 个层，网络结构的细节见表格4-1，每层的简单描述如下：在卷积层中， c 代表了输入特征的通道数， k 代表了卷积核的大小。在池化层中， k 代表了池化核的大小， p 是边缘留白的大小， s 代表了池化操作的步长。在模块层中， c 代表了输入特征的通道数， k 代表了池化核的大小， p 是边缘留白的大小， CA 代表了全连接操作， SA 代表了空间注意机制的卷积操作。整个网络输出的特征是 512×26 。压缩模型的细节可以见表4-2。 k 代表了一维卷积的大小， c 是输出特征的通道数和卷积扩张的大小。

4.4 字符识别数据集

ICDAR2003 (IC03) 包含 1,156 个用于训练的图像和 1,110 个用于测试的图像。在此数据集中，忽略所有太短（少于 3 个字符）的单词或包含非字母数字字符的单词，最终数据集包含 867 个图像。已使用两种不同版本的数据集进行评估：具有 860 和 867 图像的版本。867 数据集在 860 图像数据集上增加了 7 个文字框。

ICDAR2013 (IC13) 包含 ICDAR03 的大部分图像。它包含 848 个用于训练的图像和 1,095 个用于评估的图像，其中在删除非字母数字字符后留下 1,015 个图像。目前有两个不同的版本用于评估：857 和 1,015 图像。857 图像集是 1,015 集的子集，其中删除了短于 3 个字符的单词。

ICDAR2015 (IC15) 包含 4,468 张用于训练的图像和 2,077 张用于评估的图像。ICDRAR2015 数据集的难度比较大，图像模糊，旋转，分辨率低。目前有两个不同的版本用于评估：1,811 和 2,077 图像。

Street View Text (SVT) 包含其测试集中的 647 个裁剪图像和 257 个火车集中的图像，这些图像是从 Google 街景中以较低的图像分辨率收集的。

IIIT 5k-Words (IIIT5k) 来自谷歌图像搜索，包括 5000 张图像，包括自然场景图像和数字。IIIT 包括 2,000 张用于训练的图像和 3,000 张用于评估的图像。

SVT Perspective (SP) 包含 645 张评估图片，这些图片是从 Google 街景中收集的。由于存在不同角度的拍摄，存在许多透视图像。

CUTE80 (CT) 包含 288 个裁剪图像用于评估，许多图像都是弯曲的。

其中 *IIIT5k, SVT, IC03, IC13* 为较工整的识别图像，而 *IC15, SVTP, CUTE* 为不规则的识别图像，如图4-5所示，我们将自己改进后的网络在以上数据集上都做了相应的实验。



图 4-5 字符识别数据集展示

4.5 实验结果与比较

为了验证我们识别算法的性能，我们将模型分别在 *IC03, IC13, IC15, SVT, IIIT5k, SP* 和 *CT* 字符识别数据集上做了实验，并且比较了多个字符识别方法，包括 CRNN, GR-CNN, Rosetta 等。实验结果如表4-3。

从表中可以看出，我们的算法在许多数据集上已经超越了之前的算法，在 *SVT3000* 数据集合上，我们比当前最优的算法 TPS-ResNet 要高 0.4 个百分点。*IIIT* 数据上，精度最高的为 SSFL 算法，该算法精度达到了最高的 89.4，在这个数据上我们的方法有所欠缺，低于 SSFL 算法 2.2 个百分点。在 *IC03* (647) 数据集上，精度最高的是 TPS-ResNet，达到了 94.9，我们的算法要优 0.5 个百分点。而 *IC03* (860) 数据集上，最优的算法为 SSFL，和我们的算法精度相同，都达到了 94.7。对于 *IC13* 数据，在 867 上，我们低于最优算法 SSFL 的 94.0，只达到了 93.3 的精度。在 857 上，我们低于最优算法 EP2.2 个百分点。在 *IC15* 上，我们都优于最优的算法，达到了 78.5 和 72.46。在 *SP2077* 上，我们的算法达到了 80.7 的精度，超过最优算法 TPS-ResNet1.5 个百分点。在 *CT645* 数据集上，我们要比最优算法 AON 低 3.5 个百分点。

为了能够更加直观的查看网络输出的识别结果，我们分别在 *IIIT* 数据集和 *ICDAR13* 数据集上挑选了一些识别正确和错误的具有代表性的图片，分别挑选了 14 个片段，如图4-6, 4-7, 4-8和4-9所示。

在 *IIIT* 数据集上，从图4-6中我们可以看出，四阶段算法能够正确识别任意的弯曲文字，例如图中的 *broad* 和 *prospect* 等图片。并且能够很好的抗拒倾斜的文字，如图中的 *everyone*

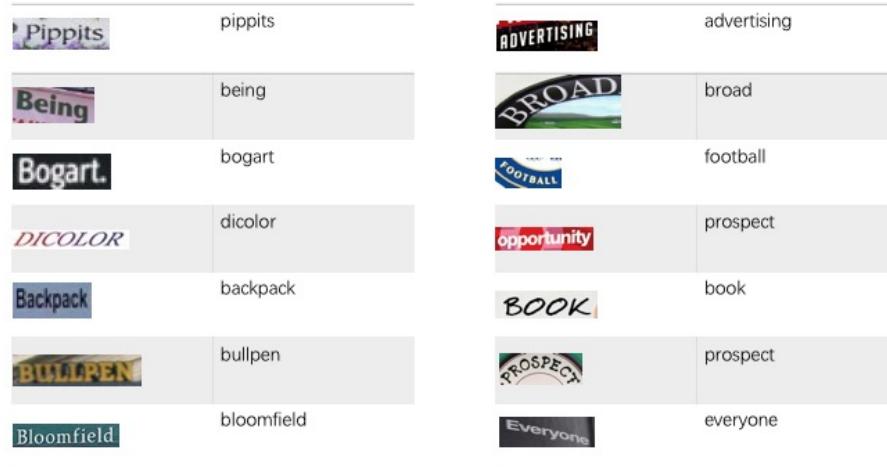


图 4-6 IIIT 数据集识别正确的样例展示

和 being。同时面对复杂艺术自己的文字识别上，我们同样达到了较好的识别结果，例如图中的 bullpen。

从错误的图4-6中分析，该模型在识别形近字上存在问题，容易将干扰识别成文字，例如”识别成 p, ! 识别成了 l。空间变换阶段也存在问题，将 THE 变换错误，导致识别成了 can。其中错误的情况较多发生在各种艺术字体上，对于连笔的艺术字体的识别效果不佳，例如图中的 menuboam 和 grisberg 等。

在 ICDAR2013 数据集上，从图4-8中可以看出，大多正确的识别都是较为工整的输入图像，代表在场景限定为识别形变较少的文字时，我们的算法能有很好的表现。

从错误图4-9可以看出，在模糊文字识别上，该模型还存在缺陷，例如图4-9左列第三行和第四行的例子。在遮挡问题上，模型还无法很好的解决，无法自动的补全被遮挡的部分，如图4-9的左列第一张图和右列的最后一张图。说明在序列建模阶段的语义特征还不够鲁棒。

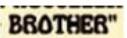
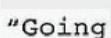
| | | | |
|---|----------|---|----------|
|  | brotheri |  | outtage |
|  | boardl |  | yearl |
|  | daarvan |  | can |
|  | valent |  | fl |
|  | pultys |  | cottags |
|  | pgoing |  | osmosisl |
|  | menuboam |  | grisberg |

图 4-7 IIIT 数据集识别错误的样例展示

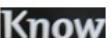
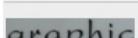
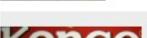
| | | | |
|---|-------|---|---------|
|  | know |  | here |
|  | kills |  | graphic |
|  | key |  | glass |
|  | kenco |  | fun |
|  | hot |  | gallery |
|  | ice |  | get |
|  | hose |  | germ |

图 4-8 ICDAR13 数据集识别正确的样例展示

表 4-1 特征提取网络配置

| Layers | Configurations | Output |
|--------|--|-----------------|
| Input | corrected image | 100×32 |
| Conv1 | $c: 32 k: 3 \times 3$ | 100×32 |
| Conv2 | $c: 64 k: 3 \times 3$ | 100×32 |
| Pool1 | $k: 2 \times 2 s: 2 \times 2$ | 50×16 |
| Block1 | $\begin{bmatrix} c : 128, k : 3 \times 3 \\ c : 128, k : 3 \times 3 \\ CA : [8, 128] \\ SA : [k = 7, p = 3] \end{bmatrix} \times 1$ | 50×16 |
| Conv3 | $c: 128 k: 3 \times 3$ | 50×16 |
| Pool2 | $k: 2 \times 2 s: 2 \times 2$ | 25×8 |
| Block2 | $\begin{bmatrix} c : 256, k : 3 \times 3 \\ c : 256, k : 3 \times 3 \\ CA : [16, 256] \\ SA : [k = 7, p = 3] \end{bmatrix} \times 2$ | 25×8 |
| Conv4 | $c: 256 k: 3 \times 3$ | 25×8 |
| Pool3 | $k: 2 \times 2 s: 1 \times 2 p: 1 \times 0$ | 26×4 |
| Block3 | $\begin{bmatrix} c : 512, k : 3 \times 3 \\ c : 512, k : 3 \times 3 \\ CA : [32, 512] \\ SA : [k = 7, p = 3] \end{bmatrix} \times 5$ | 26×4 |
| Conv5 | $c: 512 k: 3 \times 3$ | 26×4 |
| Block4 | $\begin{bmatrix} c : 512, k : 3 \times 3 \\ c : 512, k : 3 \times 3 \\ CA : [32, 512] \\ SA : [k = 7, p = 3] \end{bmatrix} \times 3$ | 26×4 |
| Conv6 | $c: 512 k: 2 \times 2 s: 1 \times 2 p: 1 \times 0$ | 27×2 |
| Conv7 | $c: 512 k: 2 \times 2 s: 1 \times 1 p: 0 \times 0$ | 26×1 |

表 4-2 特征压缩配置

| layers | Configurations |
|--------|--------------------------------|
| layer1 | $k:3 c:256 d:1$ dropout=0.3 |
| layer2 | $k:3 c:256 d:2$ dropout=0.3 |
| layer3 | $k:3 c:256 d:4$ dropout=0.3 |
| layer4 | $k:3 c:256 d:8$ dropout=0.3 |

| Model | Year | IIIT | SVT | IC03 | IC13 | IC15 | SP | CT | Time | params ms/image | |
|----------------------------|------|------|------|------|------|------|------|------|-------|--------------------|------|
| | | | 3000 | 647 | 860 | 867 | 857 | 1015 | 1811 | | |
| CRNN ^[61] | 2015 | 78.2 | 80.8 | 89.4 | — | — | 86.7 | — | — | — | 160 |
| RARE ^[62] | 2016 | 81.9 | 81.9 | 90.1 | — | — | 88.6 | — | — | — | 71.8 |
| R2AM ^[19] | 2016 | 78.4 | 80.7 | 88.7 | — | — | 90.0 | — | — | — | 2.2 |
| STAR-Net ^[20] | 2016 | 83.3 | 83.6 | 89.9 | — | — | 89.1 | — | — | — | 73.5 |
| GRCNN ^[21] | 2017 | 80.8 | 81.5 | 91.2 | — | — | — | — | — | — | — |
| ATR ^[22] | 2017 | — | — | — | — | — | — | — | — | — | 75.8 |
| FAN ^[23] | 2017 | 87.4 | 85.9 | — | 94.2 | — | 93.3 | 70.6 | — | — | 69.3 |
| Char-Net ^[24] | 2018 | 83.6 | 84.4 | 91.5 | — | 90.8 | — | — | 60.0 | 73.5 | — |
| AON ^[25] | 2018 | 87.0 | 82.8 | — | 91.5 | — | — | — | 68.2 | 73.0 | 76.8 |
| EP ^[26] | 2018 | 88.3 | 87.5 | — | 94.6 | — | 94.4 | 73.9 | — | — | — |
| Rosetta ^[63] | 2018 | — | — | — | — | — | — | — | — | — | — |
| SSFL ^[64] | 2018 | 89.4 | 87.1 | — | 94.7 | 94.0 | — | — | — | 73.9 | 62.5 |
| TPS-ResNet ^[54] | 2019 | 87.9 | 87.5 | 94.9 | 94.4 | 93.6 | 92.3 | 77.6 | 71.8 | 79.2 | 74.0 |
| Ours | | 87.2 | 87.9 | 95.4 | 94.7 | 93.3 | 92.2 | 78.5 | 72.46 | 80.7 | 73.3 |
| | | | | | | | | | | | 27.6 |
| | | | | | | | | | | | 49.8 |

表 4-3 识别在多个数据集上比较的结果

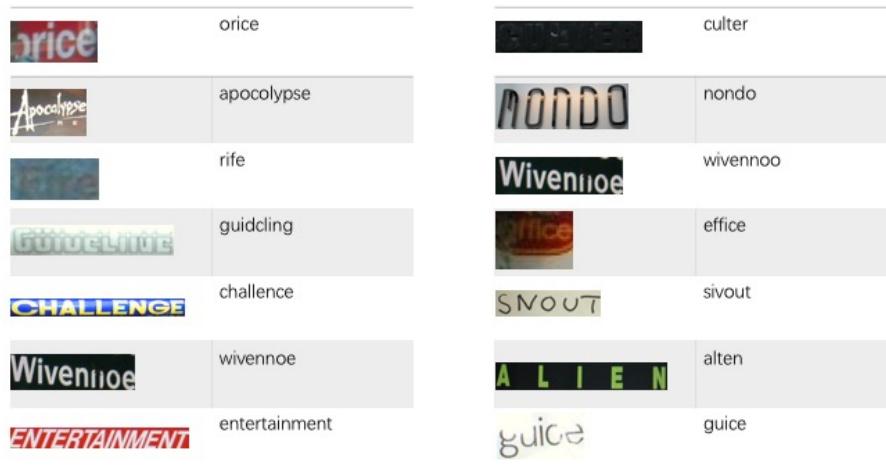


图 4-9 ICDAR13 数据集识别错误的样例展示

第 5 章 总结和展望

5.1 总结

文字作为人类记录传达信息的重要工具，在许多场景中会出现。识别图像上的文字，有助于我们很好的理解图片的含义，为后续任务提供支持。字符检测作为提取文字信息的第一步，一直都是研究的热点。字符检测的发展从最初的检测打印资料上水平上的文字，到自然场景中任意的方向的文字，再到任意形状的文字。需求不断的提高，也导致方法从最初的图像特征方式，到候选框方式，再到分割的方式。字符识别作为提取文字信息的另一个步骤，如何识别出任意形状的文字成为研究的热点。本文为了解决检测识别任意形状的文字，有以下几点贡献：

1. 本文提出了一种新的任意形状的文字表示方法，利用一系列的局部分割结果形成任意形状的文字区域。该表示方法灵活性高，不受文字形状，比例等因素影响，并且拥有更加精准的文字区域，为后期的字符识别提供便利。
2. 本文提出了一种局部分割的方法。该方法首先利用图像特征生成大量的水平正方形候选框，为了更好地适应不同大小的文字，本文在生成候选框时利用了不同维度的特征，候选框的生成只当作分类问题，没有回归的问题。之后将候选框在融合好的特征图上进行特征提取，最后将所提取到的特征送入到分割网络中，每一个候选框形成功分割结果，当作文字精准的区域。
3. 我们设计了弯曲连接算法，在得到候选框信息以及分割结果后，利用分割结果进行候选框的合并，形成文字区域。之后利用合并之后的分割结果生成文字区域的中心线，最终利用中心线求出文字的外边框，最终生成检测结果。我们在 CTW1500 和 Total-Text 数据集上与当前最优方法进行了对比实验，体现了所提出的 LSN 和曲线连接对于任意形状文本检测任务的有效性。
4. 本文使用了空间转化阶段、特征提取阶段、序列建模阶段和结果预测阶段四个阶段的字符识别模型，该架构能好很好的适应任意形状的字体，在多种数据集上都实现了对其他算法的超越。
5. 我们针对特征提取阶段，使用的 SEBlock 的结构，调整特征通道的权重，增强有利于识别结果的特征通道，减弱无用的特征，以此来增强特征提取阶段的性能。
6. 我们在诸多字符识别的数据集上进行实验对比，和当前最优的识别算法分别做了对比，实验结果显示在大多的数据集上，我们的方法具有世界领先的识别精度。

5.2 展望

深度学习的不断发展，出现了许许多多优秀的基础网络。同时，通用目标检测的迅速发展，给字符检测带来了许多新的灵感；自然语言处理、语音识别等领域的发展，给字符识别提供了许多想法。

在检测方面，LSN 网络和 CC 算法能够解决大部分弯曲文本检测的问题，但是该算法在 CTW1500 和 Total-Text 数据集上还存在一些问题，首先是速度问题，LSN 的后处理需要比较各个分割结果之间的连接关系，需要消耗大量的时间。之后的工作准备优化合并文字区域的后处理，提升性能；其次显存问题，LSN 网络生成了比较密集候选框，因此送到后期分割网络时需要比较大的显存去支撑，后期考虑加上非极大值抑制或者全局分割的方式解决现存问题；还有是检测失败问题，LSN 网络在相邻文字区域总是容易相连，尤其在 total-text 上比较明显，近期出现了一些文章关于分割问题，考虑借鉴他们的思路解决文字相连问题。同时在生成中心线是有时点的顺序会出现一些问题，导致生成的检测结果失败。后期考虑稳定性大的中线回归算法。希望之后能够借鉴其它论文的思路，不断晚上检测算法。

在识别方面，四阶段的架构在一些数据集上的精度并为达到最优，但是同样存在一些明显的问题。首先是形近字问题，如何更好的区分形近字，减少形近字之间的判断错误，尤其是标点符号和文字的区分。一种思路是提取更好的特征进行区分，因此后期将尝试不同的特征提取网络。其次是遮挡问题，如果局部去观察文字是无法正确捕捉特征。后期考虑加入语言模型，利用语义的特征补齐被遮挡的部分，提高识别精度。

最后，目前检测和识别算法还是分为两个阶段，并没有真正做到端到端的识别检测，希望在后期能够更好的结合检测和识别，在提高效率的同时，节省资源占用，提高整体的精度。

参考文献

- [1] TIAN Z, HUANG W, HE T, et al. Detecting text in natural image with connectionist text proposal network[C]//European Conference on Computer Vision (ECCV). [S.l.: s.n.], 2016: 56-72.
- [2] ZHOU X, YAO C, WEN H, et al. East: an efficient and accurate scene text detector[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2017: 2642-2651.
- [3] ZHONG Z, JIN L, ZHANG S, et al. Deeptext: A unified framework for text proposal generation and text detection in natural images[J]. arXiv preprint arXiv:1605.07314, 2016.
- [4] LIAO M, SHI B, BAI X, et al. Textboxes: A fast text detector with a single deep neural network [C]//AAAI Conference on Artificial Intelligence (AAAI). [S.l.: s.n.], 2017.
- [5] MA J, SHAO W, YE H, et al. Arbitrary-oriented scene text detection via rotation proposals [J]. IEEE Transactions on Multimedia, 2018, 20(11):3111-3122.
- [6] LIU Y, JIN L. Deep matching prior network: Toward tighter multi-oriented text detection[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2017: 3454-3461.
- [7] SHI B, BAI X, BELONGIE S. Detecting oriented text in natural images by linking segments [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2017: 3482-3490.
- [8] JIANG Y, ZHU X, WANG X, et al. R2cnn: Rotational region cnn for orientation robust scene text detection[J]. arXiv preprint arXiv:1706.09579, 2017.
- [9] LYU P, YAO C, WU W, et al. Multi-oriented scene text detection via corner localization and region segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2018: 7553-7563.
- [10] HU H, ZHANG C, LUO Y, et al. Wordsup: Exploiting word annotations for character based text detection[C]//Proceedings of the IEEE International Conference on Computer Vision. [S.l.: s.n.], 2017: 4940-4949.
- [11] LIAO M, SHI B, BAI X. Textboxes++: A single-shot oriented scene text detector[J]. IEEE transactions on image processing, 2018, 27(8):3676-3690.

- [12] LIU X, LIANG D, YAN S, et al. Fots: Fast oriented text spotting with a unified network[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2018: 5676-5685.
- [13] LIU Y, JIN L, ZHANG S, et al. Detecting curve text in the wild: New dataset and new solution [J]. arXiv:1712.02170, 2017.
- [14] ZHU Y, DU J. Sliding line point regression for shape robust scene text detection[C]//2018 24th International Conference on Pattern Recognition (ICPR). [S.l.]: IEEE, 2018: 3735-3740.
- [15] LONG S, RUAN J, ZHANG W, et al. Textsnake: A flexible representation for detecting text of arbitrary shapes[C]//European Conference on Computer Vision (ECCV). [S.l.: s.n.], 2018: 20-36.
- [16] LI X, WANG W, HOU W, et al. Shape robust text detection with progressive scale expansion network[J]. arXiv:1806.02559, 2018.
- [17] XU Y, WANG Y, ZHOU W, et al. Textfield: Learning a deep direction field for irregular scene text detection[J]. arXiv preprint arXiv:1812.01393, 2018.
- [18] SHI B, WANG X, LYU P, et al. Robust scene text recognition with automatic rectification[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016: 4168-4176.
- [19] LEE C Y, OSINDERO S. Recursive recurrent nets with attention modeling for ocr in the wild [C]//CVPR. [S.l.: s.n.], 2016: 2231-2239.
- [20] LIU W, CHEN C, WONG K Y K, et al. Star-net: A spatial attention residue network for scene text recognition.[C]//BMVC: volume 2. [S.l.: s.n.], 2016.
- [21] WANG J, HU X. Gated recurrent convolution neural network for ocr[C]//NIPS. [S.l.: s.n.], 2017: 334-343.
- [22] YANG X, HE D, ZHOU Z, et al. Learning to read irregular text with attention mechanisms [C]//IJCAI. [S.l.: s.n.], 2017.
- [23] CHENG Z, BAI F, XU Y, et al. Focusing attention: Towards accurate text recognition in natural images[C]//ICCV. [S.l.: s.n.], 2017: 5086-5094.
- [24] LIU W, CHEN C, WONG K Y K. Char-net: A character-aware neural network for distorted scene text recognition.[C]//AAAI. [S.l.: s.n.], 2018.
- [25] CHENG Z, XU Y, BAI F, et al. Aon: Towards arbitrarily-oriented text recognition[C]//CVPR. [S.l.: s.n.], 2018: 5571-5579.
- [26] BAI F, CHENG Z, NIU Y, et al. Edit probability for scene text recognition[C]//CVPR. [S.l.: s.n.], 2018.

- [27] LIU Y, WANG Z, JIN H, et al. Synthetically supervised feature learning for scene text recognition[C]//Proceedings of the European Conference on Computer Vision (ECCV). [S.l.: s.n.], 2018: 435-451.
- [28] HE K, GKIOXARI G, DOLLÁR P, et al. Mask r-cnn[C]//International Conference on Computer Vision (ICCV). [S.l.: s.n.], 2017: 2980-2988.
- [29] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016: 770-778.
- [30] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2018: 7132-7141.
- [31] JADERBERG M, SIMONYAN K, ZISSERMAN A, et al. Spatial transformer networks[C]//NIPS. [S.l.: s.n.], 2015: 2017-2025.
- [32] KARATZAS D, SHAFAIT F, UCHIDA S, et al. Icdar 2013 robust reading competition[C]//ICDAR. [S.l.: s.n.], 2013.
- [33] KARATZAS D, GOMEZ-BIGORDA L, NICOLAOU A, et al. Icdar 2015 competition on robust reading[C]//ICDAR. [S.l.: s.n.], 2015: 1156-1160.
- [34] CH'NG C K, CHAN C S. Total-text: A comprehensive dataset for scene text detection and recognition[C]//ICDAR. [S.l.: s.n.], 2017: 935-942.
- [35] SHI B, BAI X, YAO C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(11):2298-2304.
- [36] WERNER G. Text detection in natural scenes with stroke width transform[J]. Ben Gurion University, 2013.
- [37] CHEN H, TSAI S S, SCHROTH G, et al. Robust text detection in natural images with edge-enhanced maximally stable extremal regions[C]//2011 18th IEEE International Conference on Image Processing. [S.l.]: IEEE, 2011: 2609-2612.
- [38] UIJLINGS J R, VAN DE SANDE K E, GEVERS T, et al. Selective search for object recognition[J]. International journal of computer vision, 2013, 104(2):154-171.
- [39] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2014: 580-587.
- [40] GIRSHICK R. Fast r-cnn[C]//International Conference on Computer Vision (ICCV). [S.l.: s.n.], 2015.

- [41] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//NIPS. [S.l.: s.n.], 2015: 91-99.
- [42] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector[C]//European Conference on Computer Vision (ECCV). [S.l.: s.n.], 2016.
- [43] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016.
- [44] NEUMANN L, MATAS J. Real-time scene text localization and recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2012: 3538-3545.
- [45] YAO C, BAI X, SHI B, et al. Strokelets: A learned multi-scale representation for scene text recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2014: 4042-4049.
- [46] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2015: 3431-3440.
- [47] LIN T Y, DOLLÁR P, GIRSHICK R B, et al. Feature pyramid networks for object detection [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2017: 936-944.
- [48] HASTIE T, STUETZLE W. Principal curves[J]. Journal of the American Statistical Association, 1989, 84(406):502-516.
- [49] SHAHAB A, SHAFAIT F, DENGEL A. Icdar 2011 robust reading competition challenge 2: Reading text in scene images[C]//ICDAR. [S.l.: s.n.], 2011: 1491-1496.
- [50] WANG K, BELONGIE S. Word spotting in the wild[C]//European Conference on Computer Vision (ECCV). [S.l.: s.n.], 2010: 591-604.
- [51] YAO C, BAI X, LIU W, et al. Detecting texts of arbitrary orientations in natural images[C]// IEEE Conference on Computer Vision & Pattern Recognition. [S.l.: s.n.], 2012: 1083-1090.
- [52] GUPTA A, VEDALDI A, ZISSERMAN A. Synthetic data for text localisation in natural images[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2016: 2315-2324.
- [53] LYU P, LIAO M, YAO C, et al. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes[C]//European Conference on Computer Vision (ECCV). [S.l.: s.n.], 2018.

- [54] BAEK J, KIM G, LEE J, et al. What is wrong with scene text recognition model comparisons? dataset and model analysis[J]. arXiv preprint arXiv:1904.01906, 2019.
- [55] BOOKSTEIN F L. Principal warps: Thin-plate splines and the decomposition of deformations [C]//TPAMI: volume 11. [S.l.]: IEEE, 1989: 567-585.
- [56] DENG J, DONG W, SOCHER R, et al. ImageNet: A large-scale hierarchical image database [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2009.
- [57] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//ICLR. [S.l.: s.n.], 2015.
- [58] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//NIPS. [S.l.: s.n.], 2012: 1097-1105.
- [59] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2015.
- [60] GRAVES A, LIWICKI M, FERNÁNDEZ S, et al. A novel connectionist system for unconstrained handwriting recognition[C]//TPAMI: volume 31. [S.l.]: IEEE Computer Society, 2009: 855-868.
- [61] SHI B, BAI X, YAO C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[C]//TPAMI: volume 39. [S.l.]: IEEE, 2017: 2298-2304.
- [62] SHI B, WANG X, LYU P, et al. Robust scene text recognition with automatic rectification[C]// CVPR. [S.l.: s.n.], 2016: 4168-4176.
- [63] BORISYUK F, GORDO A, SIVAKUMAR V. Rosetta: Large scale system for text detection and recognition in images[C]//KDD. [S.l.: s.n.], 2018: 71-79.
- [64] LIU Y, WANG Z, JIN H, et al. Synthetically supervised feature learning for scene text recognition[C]//ECCV. [S.l.: s.n.], 2018.

致谢

时间像一阵风，在指尖敲击键盘的时候，悄悄流失。一晃眼，研究生生涯已经快要结束了，两年的求学时间仿佛就是一瞬，刚入学的情形都还历历在目。

两年前，很荣幸能够成为复旦的一份子，也很荣幸能够成为薛向阳教授的学生。在攻读计算机视觉硕士学位期间，我学习到了很多当下最前沿的技术，从最开始对人工智能的一无所知，后来慢慢的接触神经网络，看了许多学术界经典的论文，慢慢地真正开始学习、研究人工智能的相关知识；随后开始自己尝试搭建网络，在显卡上跑一些实验，也有了一些成果；接着，自己选择了字符检测方向进行深度的研究，在这期间，看了很多字符检测方面的文章，也从各类会议上的文章中获得了许多灵感。并且，也动手实现了某些文章中的内容，获益匪浅。经过前期的打基础后，开始试着实现自己的想法，整个过程使我得到了很多的锻炼。

除了进入到人工智能的这个领域，我还锻炼了自己的工程能力。接触到了一线的工程技术，认识了一些志同道合的同事，一起钻研，共同进步。两年下来，自己的工程能力精进不少。

感谢薛向阳教授两年多来的教诲，是您带领我走进的计算机视觉这个领域，从全局角度给了我许多宏观上的指导，教会我要有严谨的科研态度。让我喜欢上了计算机视觉，感谢您两年多来学习与生活上的帮助。让我了解了许多关于做科研、做项目的方法。

感谢李斌老师的教导，让我了解了计算机视觉和机器学习方向的算法，扩宽了我的眼界。感谢郑莹斌博士和叶浩博士的教导，引领我从入门到逐渐熟悉计算机视觉相关的算法。

感谢我的同学孔昱、马建奇、王丽，从他们身上学到了做科学研究，做落地项目的方法和精神，同时也感谢他们两年来的照顾，很荣幸在研究生生涯能够遇见你们。

感谢复旦大学两年来的培养，为我的学习生活等方方面面提供了很大的帮助，同时为我们学生提供了很多学习的机会，让我们不断的接触世界一流的技术，不断的成长，愿母校越来越好。

虽然学习的历程暂时告一段落，但是求知的道路依然很长。在今后的工作、学习和生活中，我将继续努力学习与计算机视觉相关方面的知识，将理论与实践相结合，做有价值的输出，为了更美好的未来而奋斗。

复旦大学

学位论文独创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。论文中除特别标注的内容外，不包含任何其他个人或机构已经发表或撰写过的研究成果。对本研究做出重要贡献的个人和集体，均已在论文中作了明确的声明并表示了谢意。本声明的法律结果由本人承担。

作者签名：_____ 日期：_____

复旦大学

学位论文使用授权声明

本人完全了解复旦大学有关收藏和利用博士、硕士学位论文的规定，即：学校有权收藏、使用并向国家有关部门或机构送交论文的印刷本和电子版本；允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。涉密学位论文在解密后遵守此规定。

作者签名：_____ 导师签名：_____ 日期：_____