学校代码： 10246
学　号： 16210240036

硕 士 学 位 论 文

（学术学位）

基于深度神经网络的多朝向场景字符识别

Multi-Oriented Scene Text Spotting based on Deep Neural Networks

院　　系：　计算机科学技术学院
专　　业：　计算机应用技术
姓　　名：　马建奇
指 导 教 师：　薛向阳　教授
完 成 日 期：　2019 年 2 月 25 日

指导小组成员名单

薛向阳 教授

李斌 研究员

目　录

摘 要

场景字符识别一直是计算机视觉社区中重要任务之一。字符识别任务是将字符区域从图片中定位出来，并将其中的字符转变为机器可以识别字符串的过程。实际的应用场景如车牌识别、路牌导航和文档扫描等。随着识别应用需求增加，越来越多的研究学者开始关注非限制场景下的字符识别。而在自然场景当中，由于字符朝向任意，现有的识别算法的识别性能受到很大限制。

深度神经网络包含了将原始输入从低级特征映射到高级特征和利用学习到的特征完成各种任务的特性。它能够自适应地从整体输入中根据任务学习和调整特征表示，相比手工设计的特征规则更为优越。因此，神经网络的模型有更强的泛化性，在计算机视觉领域有着很好的应用效果。

本文的研究工作主要是针对任意朝向的字符识别任务设计了一种朝向自适应的深度模型框架，使网络能够预测字符的阅读朝向，减轻字符朝向问题对识别效果的影响。该多任务可学习的深度模型能够独立完成检测到识别的整个识别流程。文章整体内容包括：

1.首先对当前字符检测和端到端识别任务的研究现状，了解现有方法的缺陷所在，具体阐述候选框检测算法的原理和本文对任意朝向字符的检测和识别模型设计思想。

2.在检测部分，本文基于通用目标检测算法 Faster R-CNN 提出了基于旋转候选框的字符检测模型，该模型不仅能够学习字符区域的中心坐标和长宽，还能学习字符区域的朝向角度。这种带角度的检测框相较传统的水平检测框具有更好的紧致性，能对后续的字符识别任务提供更紧致的字符区域，从而带来更好的识别效果。在 ICDAR2015、ICDAR2013 和 MSRA-TD500 标准数据集上的检测效果也充分说明了该算法产生检测结果的优越性。

3.文章在提出的检测算法基础上构造端到端的识别系统，将检测与识别任务联合成多任务架构，使得整个模型能够统一识别和检测任务，彼此促进增强。本文也从速度，多任务学习和识别效果三个方面进行试验。实验说明了联合的多任务识别系统相较检测和识别相互独立的方案更有效率，另外，在 ICDAR2015 数据集的端到端识别任务上的识别性能也说明了检测结果的紧致性对识别的影响。

关键词：神经网络；光学字符识别；多任务学习；多朝向字符检测

中图分类号：TP391

Abstract

Scene text spotting is a hot topic lasting for a few years in computer vision community. Text spotting aims to find the text region in a scene image and transfer the text region into word strings which computer can process. License plate number recognition and tracking and self-driving car guideboard navigation are two of scene text spotting applications. As practical demand increasing, text spotting in natural scene gets more and more attention from computer vision society. However, performance of scene text spotting methods is still limited due to its arbitrary orientation.

Deep neural network has the advantage of effectively extracting features from raw input and performing various tasks with particularly learned feature. Neural network can perform task-specific behaviors according to the training data and its output feature, which shows its superiority over manual feature engineering and better generalization on almost computer vision tasks.

In this thesis, we design a scene text spotting system which is aware of the text orientation by predicting the orientation angle of text instance. Thus we can alleviate suffer of text projective distortion on recognition by rectifying text orientation. Our multi-task model can train both text detection and recognition at the same time in a single model. Content of the thesis is listed as follows:

1. The thesis first introduces current research situation of text detection and text spotting system, points out the weaknesses and deficiencies of current algorithms. Next, we analysis proposal-based object detection in detail, thus leads to the insight of our proposed text spotting system.

2. In the detection part, we propose a novel multi-oriented text detection model based on Faster R-CNN, which can predict rotation proposals. This model can not only learn the position and scale of a text instance, but the orientation of the text instance. Obviously, inclined proposals have better tightness than traditional horizontal proposals. Thus, we can provide tighter input to later recognition task and bring better performance on recognition task. We also perform experiments on

ICDAR2015, ICDAR2013 and MSRA-TD500 datasets, whose performance on benchmarks shows the superiority and tight prediction our model provides.

3. Based on our detection model, we construct our text spotting system in form of multi-task model. Thus we can perform both text detection and recognition in a single model and the two tasks are jointly trained. We further make experiments on three aspects, namely, spotting speed, benefits of multi-task learning and superiority of using tighter proposal on recognition, to show the advantages of joint learning. Moreover, our results on ICDAR2015 text spotting benchmark also illustrate the benefits from tighter detection on recognition.

Key Words: Neural Network; Optical Character Recognition; Multi-Task Learning; Multi-Oriented Text Detection

CLC Number: TP391

第一章 绪论

1.1 研究背景及意义

光学字符识别任务在计算机视觉社区中拥有举足轻重的地位。字符作为一种人工的视觉信号，能够给他人和群体传递和分享信息。如果能够将像素表示的字符转变为机器可以识别的字符串，那么计算机就能够利用图像中的字符信号进行更高级的语义分析，完成更高级的语义任务，为后续计算机视觉相关的软件应用或研究任务提供便利，其中可能包括视觉分类[1][2]，视频分析[3][4]和移动应用[5]等。具体场景如停车场中的车牌识别系统，算法通过对车牌上的内容进行识别，智能系统就能够给车辆进行定位和鉴别，如果连接交通管理系统，甚至能够完成对车辆的追踪等高层语义任务。此外，场景中的路牌字符识别也能够为自动驾驶提供导航信息，为自动驾驶任务提供新导航信息。在这样的应用需求下，越来越多的研究学者开始关注非限制场景下的字符识别。而在自然场景当中，由于字符的形状不规则，由于相机视角透视变换的影响，字符的阅读朝向无法限制；加之字符区域模糊和曝光等因素的影响，自然场景当中的字符识别问题仍然无法很好的解决。

尽管当前有不少成熟的商业字符识别算法（如 ABBYY[6]）和系统（PhotoOCR[7]）应用在提供识别服务。但是这些商业系统也仅仅在文档，门牌号等限制场景下具有比较好的结果，但是在不受限制的自然场景当中，这些算法的性能和可靠性会剧烈下降。因为在不受限制的自然场景当中，字符区域的会产生较大的变化，曝光、遮挡、模糊，字符的透视变化等。在这样的场景中，字符在图片当中会产生不规则的形变和缺失，对算法识别的准确程度造成影响。而透视变换以及字符朝向的变化则是当前非限制场景当中一个最频繁出现的难题。

1.2 自然场景中任意朝向字符检测任务

字符检测任务是光学字符识别总任务中的子任务，也是字符识别任务的前置任务，检测结果的好坏对随后识别结果的影响至关重要。如前所述，字符检测任务需要算法将图片中的字符区域以坐标位置的形式预测出来。由于在实际的场景当中，图片当中的字符区域因拍摄角度的关系，所呈形状并不一定是水平或竖直的矩形形状，实际出现的形状总是会经过一定程度的透视变换从而呈现的形式通常是不规则的四边形。如果模型预测产生的字符区域不够紧致，则会对随后的识别模型预测的结果产生很大影响，通常会使得整个识别结果质量下滑，如下图（图 1-1）所示：

图 1-1 两种检测结果在字符识别算法上的结果

图 1-1 为两种模型预测的检测区域在识别算法上的识别结果，第一种算法预测的结果是水平矩形，而其中的字符呈现方式并不是水平的。第二种算法预测的是带字符区域朝向的矩形，可以预测出字符区域的朝向。明显看出后者预测出来的字符区域，明显比前者的预测结果要更为准确和紧致。从识别的结果来看，后者能够预测出准确的结果，而前者无法识别，体现了检测结果对识别任务的重要性。

1.2.1 评测指标

我们需要指标来对结果进行定量衡量，才能够准确表示算法的性能。几大标准数据集都通过评估检测框交并比的办法给出定量指标。

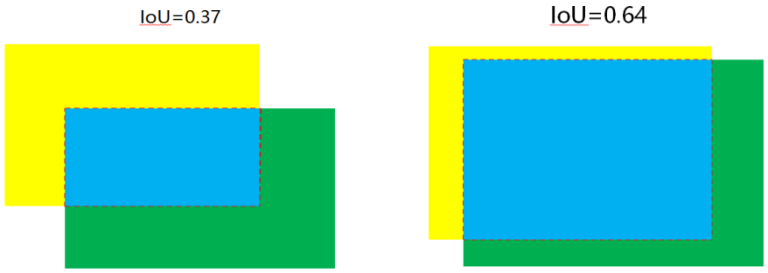


图 1-2 预测结果与标注区域交并比

计算两个矩形框的交并比的值，并与规定好的检测正样本阈值（IoU=0.5）作为对比，筛选出正确的检测结果与错误的检测结果。记真实标注区域为 GT，预测的结果为 P。则交并比的计算方式为：(GT∩P) / (GT∪P) 。计算好的值取值范围在（0, 1）。图 1-2 中蓝色区域为两结果交集的区域，黄色为算法预测候选框，绿色为真实的标注信息 GT。左边的交并比为 0.64，大于 0.5，因此是一个正确的预测结果；而图 1-2 右边的预测交并比为 0.37，小于 0.5，右边的检测结果是假阳性（False Positive, FP）结果。

（1）准确率

准确率的计算为所有正确的检测框（True Positive, TP）比上所有算法预测

出来的结果：

$$\text{precision} = \frac{TP}{TP + FP} \tag{1.1}$$

（2）召回率

召回率的计算为所有正确的检测框比上所有应该预测的正确结果，其中包括正确检测框与正确的不该检测的内容（True Negative, TN）：

$$\text{recall} = \frac{TP}{TP + TN} \tag{1.2}$$

（3）Fmeasure

Fmeasure 是准确率与召回率的均衡值，Fmeasure 高表明一个算法对准确率和召回率的兼顾较为到位，检测性能好。

$$\text{Fmeasure} = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{1.3}$$

一些公开数据集在评测时会根据检测结果自动将同一行上的多个检测框合并成为单独一行作为最后评比结果（如 ICDAR2013[24]和 ICDAR2015[25]），使得评测方式更为自由。

1.2.2 深度学习相关知识

随着深度神经网络逐渐在计算机视觉任务上的表现日新月异，它也逐渐被研究人员进行设计以用来完成各种视觉任务。而深度神经网络也是由基本的运算操作进行有规律的堆叠来完成的。那么其中几种最基本的操作层包括：卷积，池化，ReLU 非线性层以及批次归一化层等。

（1） 卷积（Convolution）

卷积操作是指将输入图像与核按位乘法，乘积的结果进行累加，随后填入输出图像对应的像素位置。一个简单的二维卷积操作计算可以表示为：

$$\mathrm{F}_{out(i,j)} = \sum_{m=i-\left\lfloor\frac{k}{2}\right\rfloor,x=0}^{i+\left\lfloor\frac{k}{2}\right\rfloor,k} \sum_{n=j-\left\lfloor\frac{k}{2}\right\rfloor,y=0}^{j+\left\lfloor\frac{k}{2}\right\rfloor,k} K(x,y) \cdot F_{in(m,n)} \tag{1.4}$$

其中$\mathrm{F}_{out(i,j)}$表示坐标为(i,j)位置上的输出结果，$F_{in(m,n)}$则表示输入位置(m,n)上的值。$K(x,y)$则表示卷积核第(x,y)位置上的参数值。经过按位乘法计算之后加和，得到最后的(i,j)位置上的输出结果。其简单图示则如图 1-3：

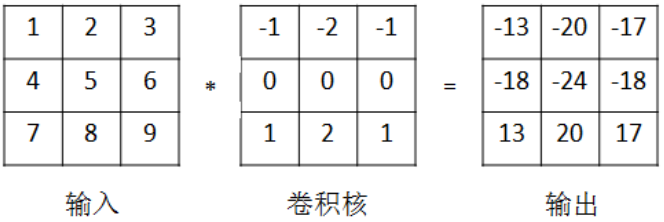


图 1-3 卷积操作图示

该图的输入为 3×3 的矩阵，卷积核为 3×3，则步长为 1 的计算结果如输出矩阵所示。因此，经过卷积核计算得到的图像特征能够在一定程度上扩大每个像素点的感受野，因此在特征图进行降采样的过程中能够获得图像的语义信息，在减少特征维度的过程中能够保持完整的图像语义信息。

（2） 池化（Pooling）

池化操作在神经网络结构当中主要充当降采样的角色，其池化操作又包含平均值池化（Mean Pooling）和最大值池化（Max Pooling）两种。对于一个简单的二维池化操作，其过程如图 1-4 所示：

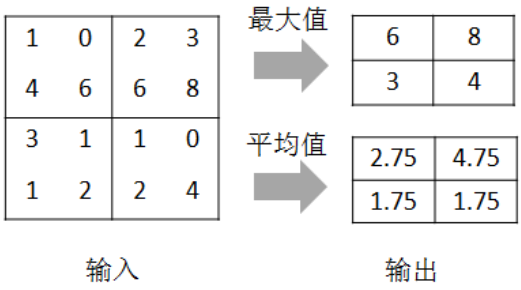


图 1-4 两种池化操作图示

上图表示的是窗口为 2×2 的池化操作过程，输入时特征图大小为 4×4，经过一次池化操作后分辨率减半，变为 2×2。最大值池化在特征降维的过程当中倾向于保留每个子区域当中的最大激活值，而平均值池化则是在子区域当中对所有值取平均。两种池化方式以不同的方式保留特征的信息。

（3） ReLU 层

ReLU（Rectified Linear Units）激活函数是神经网络中很经典的一个层组件。神经网络中主要采用 ReLU 操作来增加神经网络的非线性程度。该激活函数主要定义如下：

$$\mathrm{F}_{out} = \begin{cases} F_{in} & F_{in} \geq 0 \\ 0 & F_{in} < 0 \end{cases} \qquad (1.5)$$

若输入中某点特征的激活值大于 0，那么则保留该值作为输出；反之，则将该点的值置为 0。使用 ReLU 激活函数可以有效的将特征图稀疏化，减少噪声，神经网络模型可以进行更好的学习。

1.2.3 任务研究现状

由于时下的识别算法对自然图片中字符进行识别需要字符区域布满整张图片才能获得较好的效果[6][7]。因此我们需要对图片上的字符区域进行检测，将字符区域从原图上截取，然后再送入识别算法进行识别。那么识别效果的好坏，则取决于截取的字符图片对识别算法是否紧致。

传统图像处理方法中的图像字符检测算法过程主要基于连接组件（Connected-component）的方式，其中较为代表性的方法[9]首先将图片划分成若干个局部小窗口（sub-window），对每个窗口计算方向梯度直方图[8]特征，再通过简单的分类器对相应的特征描述子进行分类，筛选出字符的大致区域，再按照一定的置信度阈值对置信图进行二值化，就能够获得图像中字符区域所在的分割（segmentation）区域。由于粗的二值化图中字符区域过于碎片化，输出结果无法表示完整字符区域，因此粗二值化图还需要用条件随机场（Conditional Random Field，CRF）算法通过提高字符区域周边置信度的方式将图片中字符区域的可连接的部件组合成为完整的字符区域，从而将 HOG 特征分类产生的粗糙的二值化图矫正。连接组件系列算法的流程图如图 1-5。

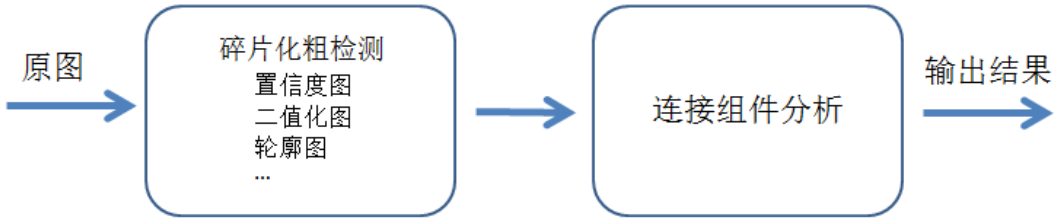


图 1-5 基于连接组件的字符检测算法流程

而在[9]中则提出通过字符色彩和像素梯度两者结合的方式提取出可能是字符的组件，随后通过临近字符和字符线（Text Line）归并的方式，考虑了字符线的弯折程度，确保字与字之间的朝向角保持一致，最后合成完整的检测结果，并且在任意朝向数据集上的实验也体现了连接组件思想在任意朝向字符检测上的优越性。

与通过颜色梯度发现字符区域组件不同的是，Canny 算子[10]通过对图片当中的轮廓进行检测，检测出来的结果通过笔划宽度变换[17]（Stroke Width Transform，SWT）提取出可能是字符轮廓的部分作为需要连接的组件，再进行归并和连接，最后组成检测结果。由于连接组件方式自底向上归并成为完整的检测结果，因此它的检测结果可以依据字符形状和朝向的变化而变化，易于表示不规则的字符区域，产生的检测结果自然也会更为紧致。在任意朝向的自然场景字符数据集上，该方法也取得了当时世界领先的检测精度。

连接组件的思想在神经网络模型设计中也在延续。随着全卷积网络[18]

（Fully-Convolutional Networks, FCN）在图像分割领域拔得头筹，它也被研究学者运用到了字符检测领域当中。由于 FCN 能够对图像中的物体进行像素级的分类，因此其输出结果能够表示物体较为精确的轮廓，对不规则的透视变换字符区域具有较好的适应性。Zhang [19]提出采用 FCN 对图像中的字符区域进行预测，产生粗的字符区域轮廓，再借助最大稳定极值区域[20]（Maximally Stable Extremal Regions, MSER）从粗字符区域中找到每个单字的位置，并在区域中以通过最多字符的中心线朝向为整个字符区域的朝向，从而确定整个带朝向的字符区域。（如图 1-2 所示）该方法借助 FCN 网络的像素级分类结果作为需要归并的组件，再通过中心线朝向对字符进行归并等方式取得最终检测结果，在透视变换的场景字符数据集 ICDAR2015 上取得世界领先的结果。

与自底向上的连接组件思想不同的是，在目标检测领域的算法主要采用自顶向下的构思对通用物体进行检测，通过预测物体的边界坐标直接产生完整的物体检测框，最早在 2014 年，研究学者设计的深度神经网络 AlexNet[11]在图像识别大赛 ImageNet 上战胜手工设计特征方法之后，深度学习（deep learning）在计算机视觉领域上逐渐占据主导地位，研究学者开始通过设计深度神经网络的方式让学习模型自动学习特征。因此 Ross 等人提出了利用富集层叠的深度特征进行物体检测[11]（Region-based Convolutional Neural Network, R-CNN），采用选择搜索算法[13]（Selective Search, SS）粗提取图片当中感兴趣区域（Region-of-Interest, RoI），随后将 RoI 送入神经网络中进行分类（classification）和回归（Regression）任务，将不够准确的区域进行矫正。R-CNN 也是第一个在检测数据集的检测任务上超过传统方法的算法，从而让图像任务进入了深度学习时代。

尽管 R-CNN 提出之后大大提高了图像检测的精度，效率的改进上仍然还有很大的空间。在 Ross 随后提出的研究工作 Fast R-CNN[14]也指出了 R-CNN 存在的几个问题：（1）模型训练需要太多步骤，不能结合成为一个整体；（2）训练耗时过长，浪费资源；（3）检测速度过慢。其原因在于，R-CNN 会对所有的 RoI 从头到尾重新做一次深度模型的前向传播。然而 RoI 之间有重叠的部分，因此所有的 RoI 会产生大量的重复计算。而 Fast-RCNN 提出的训练流程则以多任务学习的方式成为一个独立完整的模型进行训练。Fast R-CNN 算法检测一张图片，只需要做一次全局的特征计算，计算好的特征通过 RoI 池化操作（RoI Pooling）的方式，从中间层的全局特征中截取 RoI 区域对应部分的空间特征，随后每个截取的 RoI 特征都送入计算量较小的子网络(subnet)进行 RoI 区域的分类和定位任务。通过共用图片全局特征，Fast-RCNN 避免了 RoI 特征的重复计算，将检测速度从 R-CNN 的 47 秒/图加速到 0.32 秒/图，极大增加图像检测速度，同时保持了检测

精度的提升。

尽管 Fast-RCNN 已经将检测任务做成了完全用深度网络进行自学习的模型，但是在 RoI 提取部分仍然采用 SS 算法来完成。SS 算法提取 RoI 的过程相对于神经网络来说仍然慢了不少。因此在 Faster R-CNN[15]中针对提取 RoI 的部分做了优化，将 RoI 的提取以及 RoI 的分类和回归任务整合成为独立的卷积网络模型。

主干网络沿用了 Fast R-CNN 的网络架构，而候选框提取部分则用一个小型网络替代，称为区域候选框网络（Region Proposal Networks, RPN）。该网络主张在特征图的每个像素点构建多个锚点框。特征图上每个网点预先定义好不同长宽比和不同面积大小的锚点框，网点上的锚点框都以该网点中心坐标为框的中心坐标，得到共 9 种不同大小的锚点框。而区域候选框网络则通过学习真实标注框与锚点框之间的坐标与长宽差值来对锚点框进行矫正，为整个 Fast R-CNN 框架获得更为准确的 RoI。Faster R-CNN 采用区域候选框网络替代传统的 SS 算法，使得整个检测网络在速度和精度上有了更进一步的提升，在采用 ZF 网络[21]作为主干网络的条件下，Faster R-CNN 算法甚至达到了接近实时的检测速度。

Faster R-CNN 尽管在通用目标检测任务上取得了不错的成绩，但是在字符检测任务上效果仍然不佳，其原因在于字符行的长度范围相对通用物体更大，现有的锚点框尺寸无法覆盖太大的字符长度；加之，在第二步回归时只采用单层特征进行学习，会损失掉其他层有用的特征信息；针对这些不足之处，DeepText 算法[22]在 Faster R-CNN 的基础上针对字符本身不定长的特性对网络进行了改进。提出了 Inception-RPN 网络对原有的 RPN 网络进行加强，并在感兴趣区域采样的部分结合了 VGG16[23]主干网络中 conv4_3 层和 conv5_3 层产生的特征进行 RoI 池化，再以加权和的方式进行特征融合；在锚点框的设计上，增加了锚点框的尺寸个数以及长宽比个数，尽量覆盖数据集中字符区域大小及长宽的分布。最后实验也表明了这些增强确实使得检测网络在公开数据集上的检测性能得到了提升，在标准数据集 ICDAR2013[24]上也取得了比原 Faster R-CNN 高超过 10%精度的成绩，达到世界领先水平。

1.3 端到端识别（Text Spotting）任务

端到端识别任务主要是识别图中的字符，转变成计算机可以进行操作的字符串。然而大多数场景下带字符区域的图片，字符区域并不能充满整张图片。更困难的是，单张图片中可能包含多个字符区域，多个单词。因此要完成端到端识别的任务，我们需要先将字符从整图中定位出来，然后再进行识别。

1.3.1 评价指标

在进行端到端识别任务评测时衡量标准会从检测与识别两个方面进行考虑。那么，检测的准确度同样可以沿用计算交并比的方式来衡量，即，一个正样本检测结果与真实标注的交并比必须大于 0.5（如 1.2.1 所述）；并且，算法预测出的字符串必须完全与真实标注一致，才能够判定为一个正样本。总体的指标也是准确率、召回率和 F-measure。

而在最常用的标准数据集 ICDAR 系列中常用的一般为词准确率（word accuracy），预测序列与标注序列完全一致。而在词准确率标准下，根据是否提供语境字典又分为 3 类：

（1）强语境标准

强语境标准（Strong Contextual）指在进行标准集评测时，测试集会给每张图片准备好一个特定的字典合集（lexicon），包含的词数较少，通过比较预测序列与字典中每个单词，挑出最接近的单词即可，这种方式较为容易挑出正确的词汇。

（2）弱语境标准

弱语境标准（Weak Contextual）指在进行评测时，标准集会给一个较大的字典合集（一般会有数千个单词）供预测序列进行挑选。

（3）无语境标准

无语境标准（General）指一般的评测指标，直接用算法模型预测好的序列进行评估，该评估方式为三中标准中最难的一种。

在此之上，根据测试集的难度又分为端到端（End-to-end）评测和单词预测（Word Spotting）两种标准：第一种端到端（End-to-end）标准，所有的符号，数字和单词标注都参与识别评测；另一种单词预测标准则是只采用测试集中的纯字母单词进行评估。

本文中端到端的识别算法会进行以上标准的实验评估，我们在后续章节中会进行具体介绍。

1.3.2 字符序列预测背景知识

随着循环神经网络（Recurrent Neural Networks, RNN）在手写识别生的成功运用[37][38]，我们可以将字符识别部分看成一个不定长的序列标注问题。接下本文对序列识别任务常用的一些算法组件进行介绍。

(1)循环神经网络

其中 RNN 被用来对字符序列特征序列建模。其中循环神经网络的简单示意图

如下：

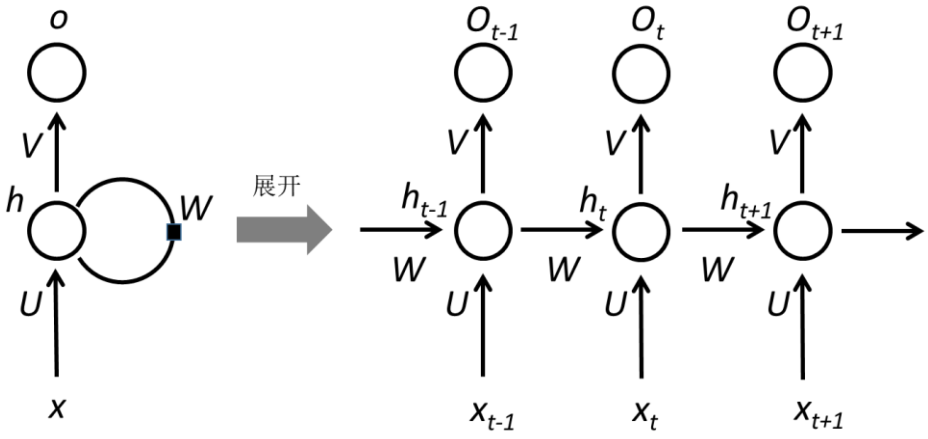


图 1-6 循环神经网络图示

图 1-6 展示的是一个简单结构的循环神经网络结构，在字符识别问题当中，x 为一定长度的输入特征序列，o 则对应各个位置上的预测字符概率。经过展开后，在序列的 t 时刻的计算则如图中右部所示。其中 W、U 和 V 为网络权重，x_t 为输入的特征序列在 t 时刻的输入。则在 t=0 时刻时的初始参数S_0为：

$$S_0 = 0 \tag{1.5}$$

因此，当 $t \geq 1$ 时，我们有：

$$\begin{aligned} \mathrm{h}_t &= Ux_t + Ws_{t-1} \\ \mathrm{s}_t &= f(\mathrm{h}_t) \\ \mathrm{o}_t &= g(V\mathrm{s}_t) \end{aligned} \tag{1.6}$$

其中f和g代表激活函数等一系列后续操作。其中h_t、s_t和o_t分别代表 t 时刻的隐层状态，隐层输出和最终输出。循环神经网络具有融合序列的时序特征的特点，任意时刻 t 预测的结果都会与先前 1 到 t-1 时刻的特征相关。正好符合语言前后关联的特性。因此采用循环神经网络来搭建字符识别网络，可以帮助模型学习语言中前后依赖的关系。

（2）连接时序分类器

连接时序分类器[35]（Connectionist Temporal Classification, CTC），最初用于语音分类中，其最主要的作用是对声波波形进行整流，使预测结果去除冗余的预测结果，并使得模型能够正确解析声波代表的语句。字符序列识别任务过程与序列标注任务相似，因此研究学者也将 CTC 分类器运用在识别任务当中。旨在让字符模型在 CTC 的帮助下正确解析图像中字符序列的类标。该解析方法通过前缀搜索解码的方式对整个字符串从头至尾进行分步解析，每一步都确保沿着最大概率的路径向下进行。那么，对于一个简单的序列“XY”，其解析方式如图 1-7 所示：

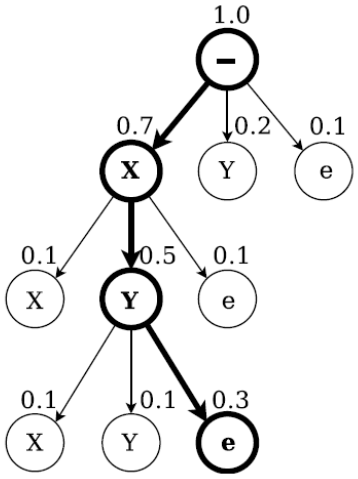


图 1-7 前序搜索解析方式

解析的起始状态为“-”，字母表为{X, Y}，每个字母被视为一个独立的状态，结束状态节点为“e”。解析的起始状态中，整个序列概率为1.0。路径的可选状态有{X, Y, e}三个，此时对应的 X 的概率为所有状态中最大，为 0.7，因此状态转移的第一步会选取 X 作为第一个转移状态，而第二部所有的概率总和为0.7，选取向下概率最大的 Y 作为序列的第二个时刻状态。再接着向下，此时结束状态的概率为最大，因此整个序列的解析结束。产生序列 XY 作为最终的预测结果。CTC 对序列的解析方式与之类似，但是更为复杂，我们接下来具体分析 CTC 对目标序列的解析过程。

设字母长度为l，则 CTC 填充序列的长度为$2l+1$。即在真实的字符间隙填充$l+1$个空白状态b，空白状态最后是可以被去掉的。当前字母到下一个状态的转移依据的是每个状态的概率按步解析。最后达到终止状态时，得到的每一步的状态都能够保证解析路径上的概率是最大的。因此对于前向推导给予的初始条件如下：

$$\begin{aligned}\alpha_1(1) &= \mathrm{y}_b^1 \\ \alpha_1(2) &= \mathrm{y}_{l_1}^1 \\ \alpha_1(\mathrm{s}) &= 0, \forall \mathrm{s} > 2\end{aligned} \tag{1.7}$$

我们定义第一个时刻（$t=1$）对应的第一个字符（$s=1$）为空格的概率为y_b^1，而第一个时刻（$t=1$）对应的第二个字符（$s=2$）为预测序列中第一个字符的概率为$\mathrm{y}_{l_1}^1$。第一个时刻的其他位上的字符概率为 0。由于每个时刻的字符概率都与上一个时刻的字符概率相关，因此其递推状态转移表示如下：

$$\alpha_t(\mathrm{s}) = \begin{cases} \overline{\alpha}_t(s) y_{l_s'}^t & if\ l_s' = b\ or\ l_{s-2}' = l_s' \\ \left(\overline{\alpha}_t(s) + \alpha_{t-1}(s-2)\right) & otherwise \end{cases} \tag{1.8}$$

其中：

$$\overline{\alpha}_t(s) \stackrel{\text{def}}{=} \alpha_{t-1}(s) + \alpha_{t-1}(s-1) \tag{1.9}$$

若当前字符为空格或是与前两个位置上的字符相同时，t时刻的字符概率只与上个时刻的当前字符概率和上个时刻的前一个字符相关；其他的时候则还与上个时刻的前两个字符相关。图 1-8 序列 CAT 的解析实例展示了从时刻 1 开始到时刻 T 的序列 CAT 解析实例。

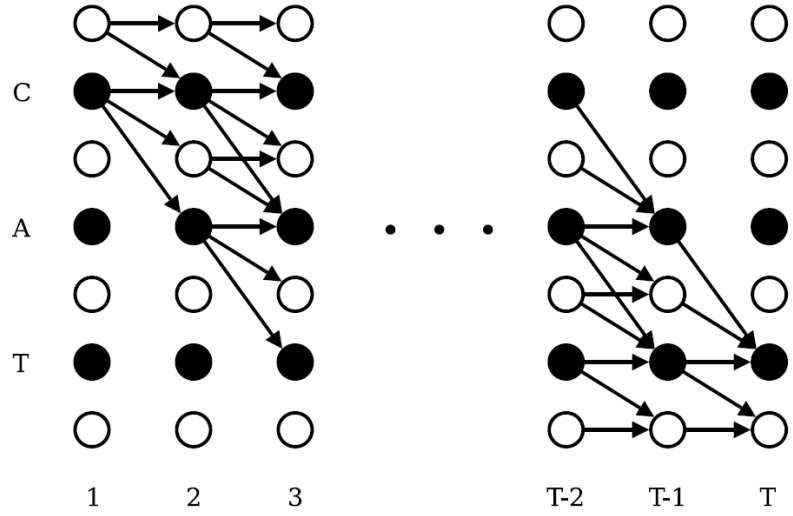


图 1-8 序列 CAT 的解析实例

1.3.3 任务研究现状

传统的端到端识别算法倾向于采用自底向上的方式组建整个识别系统，如非常有名的 PhotoOCR[7]识别系统。首先通过连接组件的方式将字符区域从图像中定位出来，随后再利用简单的神经感知机对手工特征进行建模，将字符从字符区域中一个个的识别出来，再通过定向搜索（Beam Search）的方式将字符序列组成最终的单词。整个系统当中的每个步骤被割裂开来，无法进行训练和优化。而另一类模型[32][39]则倾向于把字符区域看作一个整体，让模型对整个字符区域的特征进行词级别多分类任务。首先通过传统算法产生大量的字符区域候选框，然后再筛选出与真实标注框交并比大的候选框利用 CNN 进行坐标回归，将字符候选框进一步矫正，矫正之后的候选框内特征再进行 CNN 搭建分类器进行单词分类，建立起了基于 CNN 的端到端识别系统。而这样的多分类模型需要学习的单词类别过多，一般情况下的数据量难以使模型得到充分训练，而数据量过大则会耗费过多资源和时间。

而将字符当成序列来进行建模之后，我们可以利用 RNN 对语义连续的学习，以及 CTC 分类器对序列的准确匹配，模型就能高效的对字符进行学习和预测。

CRNN 模型以 CNN 作为底层的图像特征提取部分，经过降采样产生图像特征序列，CTC 用来对序列进行解码整流。在这样的设定下，RNN 编码的过程则变成一个学习训练数据字符串中语言模型的过程。由于 CNN, RNN 都可以进行反向传播，因此 CRNN 也将 CNN 与 RNN 进行串联，将两部分网络组成完整的系统进行训练，输入训练图片， CNN 产生图像特征序列，随后进入 RNN 对序列进行整理产生语义，最后产生完整的预测序列。端到端字符识别任务尽管需要拆分成两个任务进行，两个任务之间又存在着相互制约的关系，检测效果的好坏会影响到识别性能。若能对两个任务进行统一训练协调相互影响，这个统一模型应当能更好的协调检测与识别任务之间的关联，相互促进。因此，借由神经网络能够进行多任务协同训练的特性，Li[33]提出能够同时完成检测与识别的多任务学习框架。在 Faster R-CNN[15]框架的基础上，该模型的后端（RoI 池化层）的多任务分支除了完成字符区域的分类与坐标回归任务之外还增加了识别任务部分，形成了单一模型完成检测和识别的多任务框架思想。该模型首先对 RPN 网络部分改进，将原本的正方形的卷积核用两个不同尺寸的长方形卷积核替代，获得的特征再进行拼接，这样网络能够获得不同尺度下感知到的字符特征，对多尺度的字体具有更强的适应性。随后的特征池化层将候选框对应的特征从全局特征图中采集出来，执行后端的多任务。该模型在 ICDAR2013[23]端到端识别任务的数据集上也获得了最好成绩，但是由于产生的检测结果是水平候选框，对多朝向的字符内容的紧致性不足，因此在 ICDAR2015[25] 这类多朝向和透视变换场景下的数据集，在端到端识别任务上无法获得很好的效果。

1.4 本文研究内容及创新点

本篇文章的贡献点主要在于设计一种可以产生带朝向的字符候选框的算法，通过产生带朝向的字符区域候选框，使得模型产生更为紧致的有向字符检测结果。我们进一步对有向候选框是否能够真正对后端的识别任务有帮助进行了讨论，在检测模型的基础上增加了识别任务的分支，形成了检测与识别多任务统一训练的端到端识别算法框架，验证了有向检测框对识别任务的有效性。

1.5 论文章节安排

接下来的文章正文内容的主要安排：第二章主要叙述本篇文章中提出的有向字符检测算法的详细设计思路及实现细节；第三章主要叙述端到端识别系统的搭建思路以及统一训练的实现细节；第四章为总结部分。

第二章 有向字符检测算法

2.1 深度学习相关思想

尽管 Inception-RPN[22]加强了检测模型在字符检测任务上的检测效果，但是水平框检测结果仍然无法紧致的表示有非水平朝向的字符区域。因此通用目标检测算法在更复杂和字符朝向多变的 ICDAR2015[25]数据集上的检测精度会大幅下降。又在空间变换网络[26]（Spatial Transformer Networks，STN）中可以对物体姿态角度进行建模的思想下，基于 Faster R-CNN[15]针对有向字符设计了旋转区域候选框网络（Rotation Region Proposal Networks，RRPN），通过产生带朝向角度的候选框区域，提升了检测结果的紧致性。与连接组件方法自底向上的思想不同，模型沿用了目标检测自顶向下的思想，直接通过检测框来确定字符区域，简单快速，相对于先前用于透视变换字符检测的算法在速度和精度上都有提升。

2.2 深度网络模型搭建

接下来每个小节将详细介绍检测的流程和设计。文章分设计思路和数据集及实验 2 个部分介绍检测算法的实现过程。

2.2.1 设计思路

正如章节 1.2 所述，为了更加准确的检测不同尺寸的物体，RPN 采用了尺寸和长宽比两个参数来控制锚点。尺寸大小决定锚点的实际大小，长宽比则是控制锚点长和宽的比例。但是在自然场景的条件下，字符通常以一些不自然和任意朝向的形式出现的；由 RPN 生成的候选框仅能表示水平方向的候选框，不够健壮，

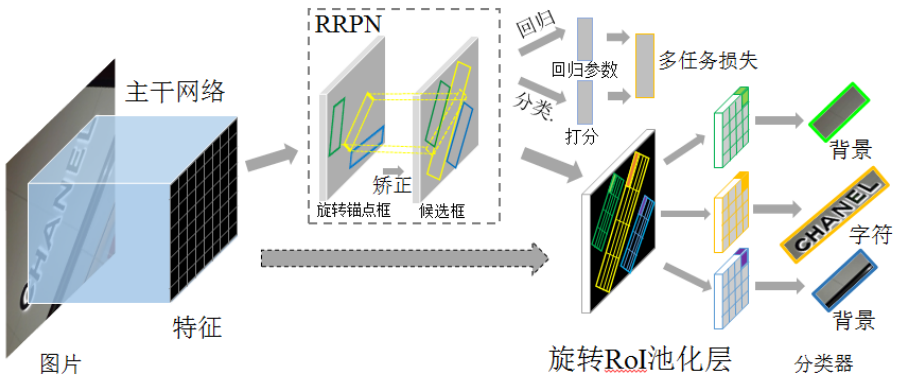


图 2-1 RRPN 算法流程图

对于朝向倾斜的字符区域无法形成足够紧致的字符框。为了让模型能够产生足够

紧致的候选框，笔者认为需要在网络的候选框生成过程中融入旋转的角度信息。

因此本文设计的模型整体框架如图 2-1 所示,模型以 VGG16[23]卷积网络作为图像特征的提取层，形成架构的主干（backbone）部分，产生出来的图像特征会送入随后的两个分支，即旋转区域候选框网络（Rotation Region Proposal Networks，RRPN）和 Fast R-CNN[14]网络。RRPN 网络为图像上的字符区域生成任意朝向的预测框，最后经过分类交叉熵损失和 smooth-L1[15]回归损失计算进行反馈学习。从RRPN网络产生的候选框则送入旋转RoI池化层(Rotated RoI Pooling)，计算出旋转候选框在特征图上的映射区域的最大值池化结果，即旋转候选框对应的旋转 RoI 特征，生成的特征送入后端多任务网络的后端的两层全连接层进行分类处理，产生候选框的类别和精确区域坐标。

2.2.2 网络模块细节

（1）输入数据处理

在数据处理环节，任意朝向的字符候选框表示为 5 元组 （$\mathrm{x,y,h,w,\theta}$）。其中的坐标 $\mathrm{(x,y)}$代表候选框的中心点，高h表示候选框短边，宽w则代表长边。朝向维度θ则是候选框长边的朝向。根据实际自然场景的字符朝向分布原则，我们将朝向的取值范围归一化到 $(-\pi/4, 3\pi/4)$ 区间中，以水平向右为角度零点，逆时针为正向，统一采用角度值表示。采用这组参数来表示旋转框有以下几个优点：方便不同旋转框之间朝向角差的计算，不需要通过坐标进行转换计算，减少中间误差；方便模型对角度的回归值进行建模和学习，朝向角度本身与坐标大小无关，因此分布稳定，便于网络的学习；相比于用四组点坐标表示旋转矩形，5 元组表示法能够很方便的计算出旋转后的真实标注，训练过程的旋转数据增强方式能够很容易实现。

对于一张尺寸为$I_H \times I_W$的输入图像，其中标注的字符区域框为$\mathrm{(x,y,h,w,\theta)}$。若我们以角度$\alpha \in [0, 2\pi)$对图像中心进行旋转，其中心坐标 (x', y')可以由以下矩阵运算实现：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T(\frac{I_W}{2}, \frac{I_H}{2}) R(\alpha) T(-\frac{I_W}{2}, -\frac{I_H}{2}) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

其中，T和R分别代表平移矩阵和旋转矩阵。两者的表示形式如下：

$$T(\delta_x, \delta_y) = \begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$R(\alpha)=\begin{bmatrix} cos\alpha & sin\alpha & 0 \\ -sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

旋转之后的标注框的宽w'与高h′不变，而旋转过后的朝向角θ'经过归一化返回$(-\pi/4, 3\pi/4)$区间中。

（2）旋转锚点框

根据实际场景中字符检测的应用和定义好的框架设计，本文采用新的策略来设计旋转锚点框(rotation anchor)。 首先，在原有设定的基础上，候选框的朝向维度上增加了朝向角度的参数来控制候选框最初生成时的朝向。我们主要选择了$-\pi/6$, 0, $\pi/6$, $\pi/3$, $\pi/2$ 和$2\pi/3$ 共 6 个朝向作为锚点框的最初朝向，选取策略同时权衡了框的数量和角度的覆盖程度。第二，由于字符的出现都是长条形，因而长宽比改为 1:2, 1:5 和 1:8，依次拟合短，中，长三种长度的字符行。 最后，锚点的尺寸则是继承了原本的设置，即 8, 16 和 32。选框策略的概况如图 2-2 所示。根据上述锚点框选择策略，每个旋转锚点框对应着 5 个参数 （x, y, h, w, θ）。在图像提取的特征图上，每个像素点总共产生 54 (6×3×3) 个锚点，相对应的，每个特征图点位上会在回归分支产生 270 (5×54) 个输出，分类器分支上产生 108 (2×54) 个输出。因此，对应一张宽 W 长 H 的特征图，RRPN 网络会生成 H × W × 54 个锚点框。

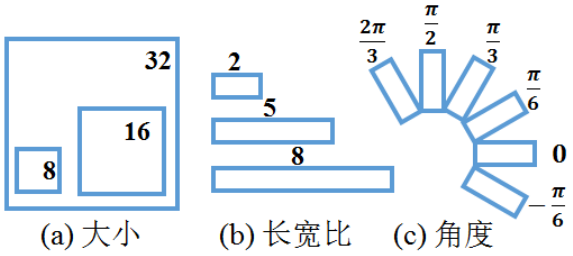


图 2-2 旋转锚点框选取策略

（3）旋转候选框（Rotation Proposals）学习策略

生成了旋转锚点框之后，适当的锚点框采样策略才能让整个网络更好的学习候选框的分类和回归。首先需要计算两个框的交并比 (Intersection over Union)。规定正样本为：(i) 和真实字符区域最高的交并比或与其交并比大于 0.7，并且 (ii)与真实字符框的朝向夹角不能大于$\pi/12$的样本；负样本:(i) 交并比小于 0.3,或(ii) 交并比大于 0.7 但是夹角大于$\pi/12$的样本。既不是正样本也不是负样本的锚点框不参加网络训练。因此，RRPN 的损失函数采用多任务损失函数的表示形式 (multi-task loss)，定义如下:

$$L(p,l,v^*,v) = L_{cls}(p,l) + \lambda L_{reg}(v^*,v) \tag{2.4}$$

其中l是候选框的分类指示信号。其中，l=1 则样本为字符，l=0 则样本为背景，参数 p=(p_0，p_1）是通过 softmax 函数计算出来的分类概率，v =(v_x，v_y，v_w，v_h，v_θ）代表真实标注数据的 5 个参数的数值，而 v^* =(v^*_x，v^*_y，v^*_w，v^*_h，v^*_θ）则是通过网络预测出来的候选框参数。公式的两项通过平衡参数λ来调节。而l类分类损失函数定义如下:

$$L_{cls}(p,l) = -logp_l \tag{2.5}$$

随后在候选框回归的损失项时，采用 smooth-L1 损失[15]函数来衡量字符区域和正样本锚点框之间的差距:

$$L_{reg}(v^*,v) = \sum_{i\in\{x,y,w,h,\theta\}} smooth_{L1}(v^* - v) \tag{2.6}$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & |x| \geq 1 \end{cases} \tag{2.7}$$

更进一步地，候选框与真实框的参数差值需要进行尺度归一化，以提高回归损失计算时的稳定性：

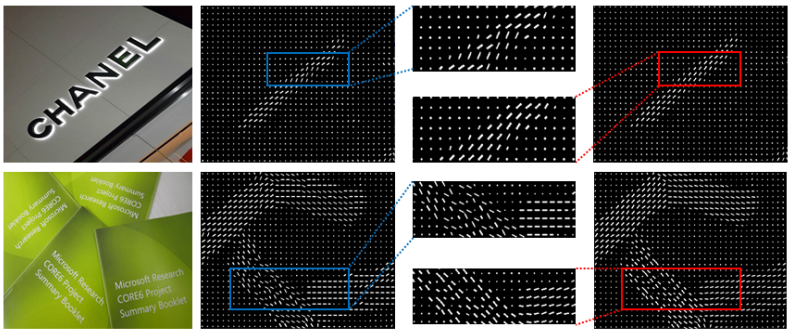
$$\begin{aligned} &v_x = \frac{x - x_a}{w_a}, v_y = \frac{y - y_a}{h_a}, \\ &v_h = log\frac{w}{w_a}, v_x = log\frac{h}{h_a}, v_\theta = \theta - \theta_a \end{aligned} \tag{2.8}$$

$$\begin{aligned} &v^*_x = \frac{x^* - x_a}{w_a}, v^*_y = \frac{y^* - y_a}{h_a}, \\ &v^*_h = log\frac{w^*}{w_a}, v^*_x = log\frac{h^*}{h_a}, v^*_\theta = \theta^* - \theta_a \end{aligned} \tag{2.9}$$

其中x，x_a和x^* 分别代表网络预测的候选框，锚点框和真实标签的中心横坐标；y，h，w和θ则同上。因此最后能够逐个计算v =(v_x，v_y，v_w，v_h，v_θ)与v^* =（v^*_x，v^*_y，v^*_w，v^*_h，v^*_θ）之间的回归差值归一化的损失。

如前所述，所有旋转锚点框的朝向被固定在区间$(-\pi/4, 3\pi/4)$中，因此 6 个不同朝向的锚点能够拟合夹角在 $\pi/12$ 之内的候选框，每个不同朝向的锚点都有其固定的拟合区间，该区间在本文中称为拟合域（fit domain)。当一个真实标注落角度值在某一个旋转锚点框的拟合域中时，该锚点框很可能成为这个真实标注对应的训练正样本。由此，6 种不同朝向锚点的拟合域将整个真实标注的朝

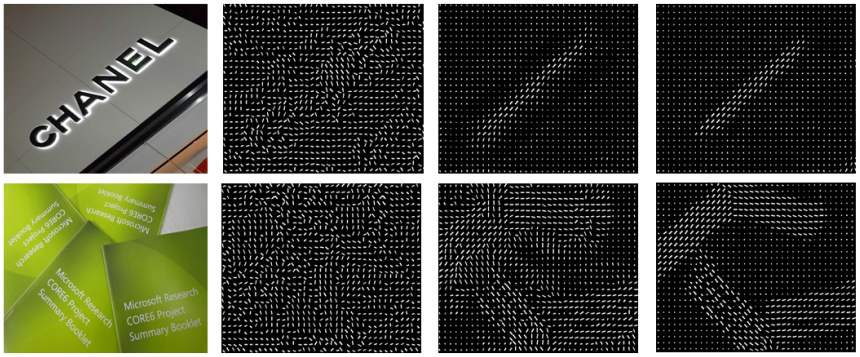
向区间$(-\pi/4, 3\pi/4)$完全包括，并将区间分割成 6 等分。由此，在$(-\pi/4, 3\pi/4)$区间内的任何真实标注都能被至少一个旋转锚点框拟合。图 2-3 展示了角度回归的效果。可以看到经过角度回归的纠正之后，字符区域内相近的点的朝向基本一致。



(a) 输入图像　(b) 无回归矫正　(c) 有回归矫正

图 2-3 角度回归效果的可视化

我们对训练过程中的候选框朝向进行了可视化，进一步了解网络模型对字符区域朝向的学习能力，如图 2-4 所示。对于一张输入图像，RRPN 网络在不同训练轮数产生的特征图，白线代表每个点上对字符响应最大的锚点，线的长短表示响应的程度。图中能够看到随着训练轮数的增加响应的区域越来越集中到字符区域周围，对字符区域的响应也越来越大。



(a)　(b)　(c)　(d)

图 2-4 角度回归效果的可视化 (a) 输入图像; (b)网络未经过训练 (c) 训练 10000 轮的学习情况;(d) 训练 150000 轮的学习情况.

（4）候选框的精确学习

斜交矩形交并比 有向候选框的朝向任意，因此之前采用水平候选框交并比算法用在任意朝向的候选框上会导致计算的不精确，导致正负样本的学习出现不必要的误差。因此算法中笔者选择三角分割对斜交矩形区域的凸多边形进行切分，分别计算三角形子区域的面积之后再逐个求和，图 2-5 展示了几何计算的步骤，算法 2-1 列出了计算的具体细节。

算法 2-1 两两斜交矩形交并比

输入：旋转矩形 $\boldsymbol{R_1, R_2, \ldots R_n}$

输出：n 个旋转矩形之间两两的交并比矩阵 $\boldsymbol{IoU \in R^{n\times n}}$

for $< \boldsymbol{R_i, R_j} >$ *do*

点集 $\boldsymbol{PSet}$ *赋值为空*

将 $\boldsymbol{R_i}$ *与* $\boldsymbol{R_j}$ *相交的交点加入* $\boldsymbol{PSet}$

将 $\boldsymbol{R_i}$ *在* $\boldsymbol{R_j}$ *内的顶点加入* $\boldsymbol{PSet}$

讲 $\boldsymbol{R_i}$ *在* $\boldsymbol{R_j}$ *内的顶点加入* $\boldsymbol{PSet}$

将 $\boldsymbol{PSet}$ *中所有顶点按逆时针序排列*

通过三角剖分计算相交区域面积 $\boldsymbol{Area(I)}$

$$\boldsymbol{IoU[i,j]} \leftarrow \frac{\boldsymbol{Area(I)}}{(\boldsymbol{Area(R_i)} + \boldsymbol{Area(R_j)} - \boldsymbol{Area(I)})}$$

end for

首先算法接收 n 个旋转矩形作为输入，通过遍历的方式选取任意两个旋转矩形$\mathbf{R_i}$和$\mathbf{R_j}$，首先计算出$\mathbf{R_i}$与$\mathbf{R_j}$相交的交点，加入点集**PSet**中；第二步，将$\mathbf{R_i}$在$\mathbf{R_j}$内的顶点加入**PSet**；第三步，将$\mathbf{R_j}$在$\mathbf{R_i}$内部的顶点加入**PSet**。我们由此获得了斜交区域的所有顶点;第四步，将所有点按逆时针顺序排列，将点归组成两两不重叠的三角形区域（三角剖分）。通过计算每个三角形的面积最后加和，最后得到了两个斜交矩形的相交区域面积。如图 2-5(a)所示，相交区域有四个顶点{N,M,H.B}，按照逆时针排列得到{N,H,M.B}，随后将该区域分割成三角形{△ NMH,△ HMB}，因此斜交矩形的面积可以由△ NMH与△ HMB的面积加和得到。

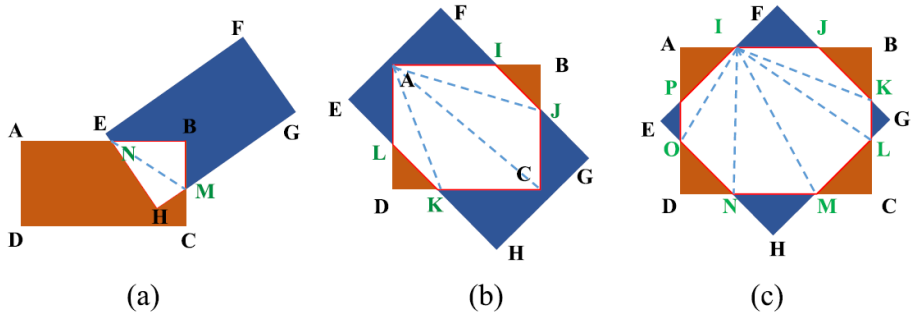


图 2-5 斜交矩形间的相交区域情形 （a）为 4 个顶点的相交区域（b）6 个顶点的相交区域（c）8 个顶点的相交区域

斜交矩形非极大值抑制（Skew-NMS）传统的 NMS 只考虑交并比大小(交并比大于 0.7 的两个候选框只留下置信度最大的)，但是对于带朝向的候选框来说是不够的。如果取极端的情况，两个候选框长宽比为 8:1，夹角为$\pi/12$，它们的交并比最大只能达到 0.31，但是在人为规定中它们仍然是正样本，这样的样本就不能够被抑制掉。因此，本文新提出的 Skew-NMS 分为两步：(i) 交并比大于 0.7 的多个候选框只保留最大交并比的候选框；(ii) 如果所有相交的候选框群彼此

交并比都在 [0.3, 0.7]中，保持夹角和真实标注最小的候选框(夹角最小的候选框必须小于$\pi/12$)。

（5）旋转感受野池化层(RRoI Pooling Layer)

Fast R-CNN 架构中，RoI 池化层为每个候选框采样一个固定尺寸的特征图，每个特征向量随后被送到之后的全卷积网络进行分类和回归的计算，输出由网络预测出的物体区域坐标和物体分类。相比于之前对每个候选框区域都进行特征提取计算，它对每一张图像只进行一次特征提取，候选框从特征图上的相应区域提取特征，由于特征计算次数的减少，物体检测的速度得到了加速。而 RoI 池化层采用最大值池化将候选框的特征转换为$\mathrm{h}_r \times \mathrm{w}_r$的小特征图，其中高$\mathrm{h}_r$和宽$\mathrm{w}_r$则成为 RoI 层的超参数，与 RoI 本身无关。

传统的 RoI 池化层[14]只接收水平候选框的 RoI 作为输入，无法处理有向候选框，因此我们提出采用 RRoI 池化层来适应 RRPN 网络产生的候选框到 RRoI 变换的计算。我们首先对 RRoI 的超参数进行设置，每个旋转候选框生成高H_r，宽W_r的 RRoI。因此每个高h，宽w的候选框区域会被切分成$\mathrm{H}_r \times \mathrm{W}_r$个等面积的区域(如图 2-6 所示)；

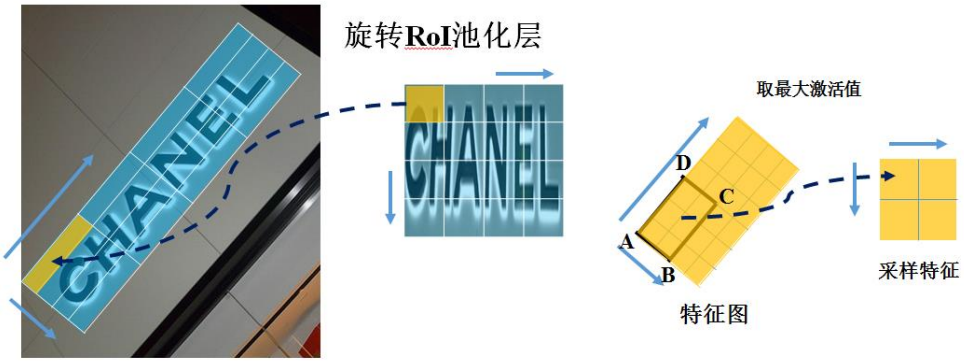


图 2-6 旋转 RoI 池化层

每个子区域与候选框有相同的朝向，而在特征图上子区域的位置和它的四个顶点(A, B, C, 和 D)。四个点的坐标可以通过旋转变换计算出来，被组织起来以界定子区域的边界。每个子区域内的特征值由最大值池化来决定，随后填入 RRoI 矩阵中的相应位置。最后，候选框被转换为 RRoI，送入网络随后部分的分类器进行分类。其算法展示如算法 2-2 所示。

（6）其他

主干回归任务的后端分支设计与 Faster R-CNN 相似，为两层全连接层，其多任务的学习与采样策略与 RRPN 子网络相同，正负样本比例都为 1:3，分类损失的设计都采用交叉熵损失，回归部分则用 smooth-L1 损失来评估候选框生成的好坏。整体 RRPN 的网络框架对锚点框两步回归修正，从而输出最终的修正的检测结果。

算法 2-2 旋转感兴趣区域池化

输入：旋转候选框 (x, y, h, w, θ)，*池化尺寸* (H_r, W_r)，*主干特征* $InFeatMap$ *以及缩放尺寸 SS*

输出：采样后的特征 $OutFeatMap$

$Grid_w, Grid_h \leftarrow \frac{w}{W_r}, \frac{h}{H_r}$

for $< i, j > \in \{0, \ldots, Hr - 1\} \times \{0, \ldots, Wr - 1\}$ *do*

$L, T \leftarrow x - \frac{w}{2} + jGrid_w,\ y - \frac{h}{2} + iGrid_h$

$L_{rotate} \leftarrow (L - x)\cos\theta + (T - y)\sin\theta + x$

$T_{rotate} \leftarrow (T - y)\cos\theta + (L - x)\sin\theta + y$

$max\_value \leftarrow 0$

for $< k, l > \in \{0, \ldots, \lfloor Grid_h \cdot SS - 1 \rfloor\} \times \{0, \ldots, \lfloor Grid_w \cdot SS - 1 \rfloor\}$ *do*

$P_x \leftarrow \lfloor L_{rotate} \cdot SS + l\cos\theta + k\sin\theta + 0.5 \rfloor$

$P_y \leftarrow \lfloor L_{rotate} \cdot SS - l\sin\theta + k\cos\theta + 0.5 \rfloor$

if $InFeatMap > max\_value$ *then*

$max\_value \leftarrow InFeatMap[P_y, P_x]$

end if

end for

$OutFeatMap[i, j] \leftarrow max\_value$

end for

2.2.3 训练流程

模型训练实验的数据集，本文采用 MSRA-TD500[40]的训练集 (300 张图像)，以及 HUST-TR400[41]数据集中的 400 张图像样本参加训练。对于 ICDAR2015[25]上的实验，ICDAR2015 以及之前所有 ICDAR 系列竞赛的数据集都参加训练。而字符朝向的学习，如果单纯借助现有的数据集，模型仍然无法得到充分训练。因此，训练过程中，本文对图片样本进行了一定的数据增强，将图片进行一定角度的旋转之后再送入网络，以少量数据让模型对朝向信息得到尽可能充分的学习。所有的效果评估指标都以准确率、召回率 和 F-measure 进行定量分析。ICDAR2013[24]所采用的训练集与 ICDAR2015[25]相同。

模型初始值采用经过 ImageNet[28]分类数据集预训练过的权重值作为基础实验的初始权重。权重更新的学习率设定为前 200,000 轮训练保持10^{-3}，随后的 100,000 轮保持10^{-4}，权重衰减保持 5×10^{-4}不变，冲量值则设定为 0.9。由于产生的旋转候选框是原 Faster-RCNN 的 6 倍，因此，为了减少计算量，在进行 IoU 计算时，过滤掉一部分超过图像边界的候选框。因此该模型无论在训练环节

还是测试环节都还能够保持与原框架相当的速度。

2.2.4 实验用检测数据集

（1）ICDAR2013（水平字符数据集）

ICDAR2013[24]又称 Focused Scene Text Challenge，数据集在 2013 年发布，作为 2013 年的 Robust Reading Competition 的标准数据集。图片多为水平字符区域，区域较为居中。数据集中 229 张图片组成训练集，233 张图片组成测试集。数据的标注为每个字符区域由左、上、右和下四个边界横纵坐标组成。

（2）ICDAR2015（多朝向字符数据集）

ICDAR2015[25]又称随手拍场景字符挑战（Incidental Scene Text Challenge），数据集在 2015 年作为当年的鲁棒性阅读比赛（Robust Reading Competition）的标准数据集发布。图片中的字符区域相较 ICDAR2013 出现的更为不规则，出现的字符区域形状多呈任意四边形（因为相机视角问题而产生的透视变换）。整个数据集样本数量也远大于 ICDAR2013。训练集样本数量为 1000，测试集样本数量为 500。示意图如下，因为出现的字符区域并不是规整的长方形，标注的形式与 ICDAR2013 有所区别，标注的信息为字符区域不规则四边形的四个顶点横纵坐标（x1, y1, x2, y2, x3, y3, x4, y4），所给的字符内容若标志为“###”，则该字符区域的检测与识别不参与评测。

（3）MSRA-TD500（多朝向字符数据集）

MSRA-TD500 数据集[40]由微软亚洲研究院的研究学者 Cong Yao 发布，数据集内容主要针对多朝向字进行标注。数据集包括 300 个训练样本，以及 200 个测试样本。标注的格式包括任意字符朝向的中心点(x,y)，任意朝向的矩形框的长宽，最后一个参数是弧度制的角度 θ 。

2.3 实验及结论

在本章节来本文将通过多方面的对比实验来详细了解 RRPN 模型的训练细节和检测性能。

2.3.1 数据增强实验

细节的改进和数据增强相关的实验本文选择在 MSRA-TD500 标准数据集上展开，采用 MSRA-TD500 训练集中的 300 张图片对基础框架进行训练;长边规定为 1000 像素。测试结果为准确率 57.4%，召回率 54.5%和 F-measure 55.9%。 这个结果已经远远高于原本的 Faster-RCNN 模型 38.7%, 30.4% 和 34.0%的成绩。旋

转候选框方法和水平候选框方法在数据集上的测试效果的可视化对比如图 2-7。旋转候选框与水平候选框相比，检测区域内的背景部分更少，说明旋转回归策略对字符检测是更为紧致的。

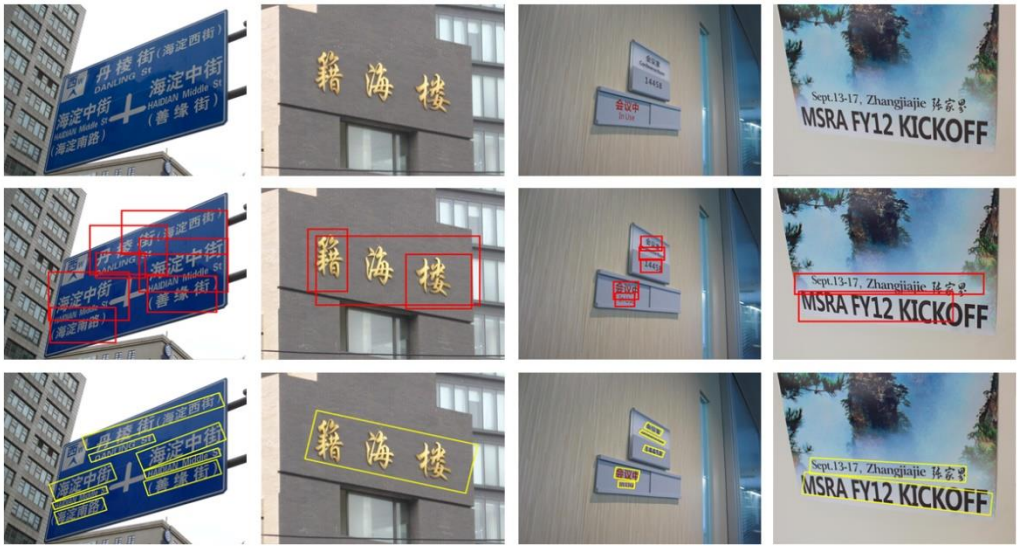
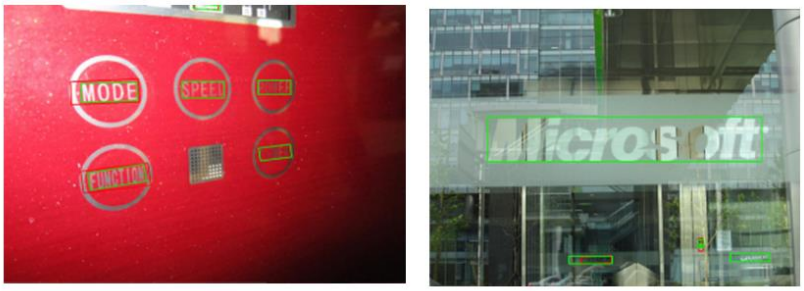


图 2-7 RRPN 基准实验结果可视化；第一行为输入原图；第二行为水平检测算法的结果展示；第三行为 RRPN 输出结果可视化

表 2-1 不同实验策略和参数设置在 MSRA-TD500 数据集上的结果；其中 P、R 和 F 分别代表准确度、召回率和 F-measure。ΔF 则是在基准实验上的提升程度。其中实验的策略包括：a. 字符区域上下文信息；b. 增加训练样本数量；c. 越界框的保留比例；d. 扰动输入图片尺度

a	b	c	d	P	R	F	ΔF
Faster R-CNN[15]				38.7%	30.4%	34.0%	
RRPN 基准实验				57.4%	54.5%	55.9%	
√				65.6%	58.4%	61.8%	5.9%
	√			63.3%	58.5%	60.8%	4.9%
		√		63.1%	55.4%	59.0%	3.1%
√	√	√		68.4%	58.9%	63.3%	7.4%
√	√	√	√	**71.8%**	**67.0%**	**69.3%**	13.4%

对基础实验做进一步分析时发现：(i)图中模糊或受到不均匀光照的字很难被检测到;(ii)很小的字符很难被正确检测，这个问题导致测试结果的召回率较低;(iii)测试集中字符区域的长度不受限制，在统计中长宽比甚至超过 10:1，这类的字符区域由于初始锚点框过短，无法被正确检测，本来应该单一的检测结果被分成几个候选框，而所有的候选框因为不满足评测的要求而成为了错误的检测结果(一些错误的检测结果如图 2-8)。改善基础实验效果的一系列的策略以及测试结果如表 2-1。



复杂背景

小字符

图 2-8 检测当中的困难情况（其中红色框为检测结果，绿色框为真实标注）

（1）字符区域上下文

在通用物体识别领域，将上下文(物体周围区域)加入模型的检测过程，会增强模型的检测效果[27]，因此笔者希望探究是否这种方法可以对字符检测任务起效。保持旋转候选框的中心点坐标和朝向不变，在数据的预处理步骤同时以 1.X 倍增大旋转候选框的长和宽。在测试环节，我们将候选框尺寸还原。如表 2-2 所示，扩大的倍数的所有实验对 F-measure 都有提升。出现这种现象的原因可能是由于候选框变大之后，字符区域的上下文信息能够被捕捉到，因此字符区域的位置和朝向能够被更准确的预测出来。

表 2-2 标注框扩大不同倍数的实验结果

扩大倍数	准确率	召回率	F-measure
1.0	57.4%	54.5%	55.9%
1.2	59.3%	57.0%	58.1%
1.4	**65.6%**	**58.4%**	**61.8%**
1.6	63.8%	56.8%	60.1%

（2）扩大训练集

训练的数据容量在原有的训练集中增加了 HUST-TR400 数据集[29]作为训练数据的补充，整个训练集扩充到 700 张图片。模型的训练效果有了显著的提升，F-measure 达到了 60.8%，表明模型得到了更充分的训练。

（3）扩展图像边界

对候选框的过滤策略过于严格，会导致大多数超过图像边界的旋转锚点被过滤。而夹带斜朝向的旋转锚点，原本在图像边界之内，但是经过上下文区域策略

训练之后，产生的候选框可能会超出图像边界而被过滤掉，因此我们对图像进行边界的扩展，用以保留更多有效的候选框样本，增加检测的准确率。实验显示，图像的每条边增加 0.25 倍的情况时，检测的效果最好。最后的策略则是将扩大训练集，上下文策略和扩展图像边界三者结合，实验效果的 F-measure 进一步的提升到了 63.3%。

（4）尺寸扰动

采用了前面叙述的训练策略之后，模型仍然难以检测出一部分小字符样本。因此我们对输入训练的图像尺寸进行了扩大，以增强检测系统的健壮性。图像输入模型时长边会被重设为小于 1300 像素的某一个值，并保持图像的长宽比不变。在之前的基础上采用了尺寸扰动策略之后，实验的 F-measure 提升到了 69.3%，与没有采用策略时的结果提升了 6%。

2.3.2 检测器在不同数据集上的性能

（1）MSRA-TD500

我们采用对比实验中的最佳设置参数在 MSRA-TD500 数据集上进行评测与检测性能调研后发现，由于数据集中是对字符行进行检测，由于中文字符行相对于英文标注框更长，导致现有锚点框的长宽比无法完美覆盖。因此最后的检测结果中会出现截断的情况。因此我们需要对截断的检测框进行合并，成完整的字符行检测框。其算法流程大致如算法 2-3 所示。对于一个未处理的候选框集合$\{\boldsymbol{P_1},\ldots,\boldsymbol{P_N}\}$，算法会从第一个框开始两两遍历候选框，在遍历中的任意一对候选框$<\boldsymbol{P_i},\boldsymbol{P_j}>$，其中存在$\boldsymbol{i}<\boldsymbol{j}$的偏序关系。我们对这一组框分别设置一个有效位$\boldsymbol{Valid}[\boldsymbol{i}]$和$\boldsymbol{Valid}[\boldsymbol{j}]$,其初始值为 1。若在遍历过程中$\boldsymbol{Valid}[\boldsymbol{i}]$或$\boldsymbol{Valid}[\boldsymbol{j}]$都变为了 0，则说明其中有框已经合并完毕并丢弃。若两个框有效位都为 1。则对这对框的几何位置关系进行判断，满足合并的条件共有两个：两个框中心连线的距离$\boldsymbol{Dis}$小于两者的平均宽度；$\boldsymbol{P_i}$的朝向角与两者中心连线的斜率差不超过阈值$\boldsymbol{T}$（实验中我们设$\boldsymbol{T}=\boldsymbol{10}$）。合并时，我们将两个短框合成为一个长框，以两框的$\boldsymbol{x},\boldsymbol{y},\boldsymbol{h},\boldsymbol{\theta}$计算平均作为合并框的对应参数，而宽度$\boldsymbol{w}$则相加作为合并框的宽度。

合并断框之后，模型的检测结果的 Fmeasure 从 69%（表 2-1）提升到了 74%（表 2-3 左栏）。最终的实验结果如表 2-3 左栏所示。本文提出的算法与效果最好的算法相差在 2 个百分点以内，但是在速度上比效果最好的算法快了两倍，检测耗时从0.6秒加速到了0.3秒，体现了检测算法比分割算法在速度上的优越性。

算法 2-3 字符框连接算法

输入：候选框 $\boldsymbol{P_1,\ldots,P_N}$*；* $\boldsymbol{P_k=(x_k,y_k,h_k,w_k,\theta_k)}$
输出：融合后的框集合 $\boldsymbol{PSet}$
角度阈值 $\boldsymbol{T\leftarrow 10}$
if $\boldsymbol{N==1}$ *then*
    $\boldsymbol{PSet\leftarrow P_1}$
end if
for $\boldsymbol{k\in\{1,\ldots,N\}}$ *do*
    $\boldsymbol{Valid[k]\leftarrow 1}$
end for
for each pair $\boldsymbol{<P_i,P_j>\ (i<j)}$ *do*
    if $\boldsymbol{Valid[i]==0}$ *or* $\boldsymbol{Valid[j]==0}$ *then*
        continue
    end if
    平均宽度 $\boldsymbol{Width\leftarrow\frac{w_i+w_j}{2}}$
    中心距离 $\boldsymbol{Dis\leftarrow\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}$
    中心连线斜率 $\boldsymbol{Grad\leftarrow\left|arctan\frac{(x_i-x_j)}{(y_i-y_j)}\right|}$
    if $\boldsymbol{Dis<Width}$ *and* $\boldsymbol{|Grad-\theta_i|<T}$ *then*
        $\boldsymbol{P_i\leftarrow\frac{x_i+x_j}{2},\frac{y_i+y_j}{2},\frac{h_i+h_j}{2},w_i+w_j,\frac{\theta_i+\theta_j}{2}}$
        $\boldsymbol{Valid[j]==0}$
    end if
end for
$\boldsymbol{PSet\leftarrow\{P_k|Valid[k]==1\}}$

（2）ICDAR2015

在 ICDAR2015 数据集上的实验设置也采用 MSRA-TD500 数据集上最好的参数集，训练集采用数据集提供的所有训练样本，得到了准确率为 45.32%、召回率为 72.56%和 F-measure 为 55.87%的结果。由于 MSRA-TD500 与 ICDAR 系列数据集所提供的检测标注存在差别（MSRA-TD500 数据集提供行级别的标注，而 ICDAR2015 数据集提供的是单词级别的标注），因此最终的检测效果不够好。除此之外，训练样本中存在很多模糊不清面积很小的训练样本，这些样本会使得检测器出现过多像字却不是字的误检结果；并且，输入的图片过小，检测器就更难发现较小的字符区域；更重要的一点，相比于一些算法[30][31]，检测器由于训练集过小没有得到足够充分的训练。

首先，为了减轻小字符所造成的性能下降，我们在训练过程中将输入图片在保证长宽比的同时，将图片长边在不大于 1700 像素的条件下进行尺寸随机扰动。让模型在训练的过程中能接触到尺寸变化更大的检测样本，增加检测器的鲁棒性。与此同时，在进行测试时保证图片长边为 1700 像素，进一步减小小样本出现的比例。

其次，基准实验中测得的准确率相较召回率过低，造成这个问题的原因是误检结果过多造成。而在调研 ICDAR2015 数据集时发现训练集中对无法辨别的字符区域也进行了标注，能够识别文字的标注信息中有具体的文字信息，而无法辨别的文字区域的文字标注为“###”。如果无法辨别的文字样本过多，则会干扰检测器。对此，笔者按比例随机移除 ICDAR2015 数据集中的“###”样本进行了对比实验（如图 2-9 所示），从曲线的走向可以看出，移除 0%~80%的“###”样本对召回率影响并不明显，而准确率稳步上升。这个现象证明了前文所述的猜想，字符模糊的样本会影响检测器的准确率，从而降低整体的检测性能。

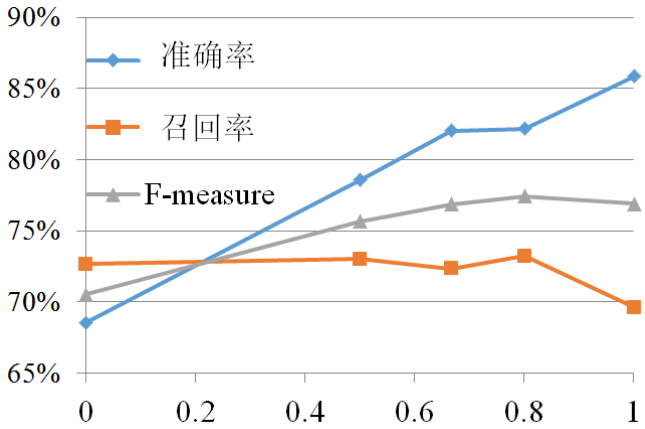


图 2-9 一定比例“###”样本移除后的各种指标折线趋势，纵坐标为三种曲线的百分比数值，横坐标为移除的样本占总数的比例。

表 2-3 RRPN 方法在各个标准数据集上与其他方法的比较，P 代表准确率，R 代表召回率，F 则代表 F-measure

MSRA-TD500					ICDAR2015				ICDAR2013			
方法	P	R	F	时间(秒)	方法	P	R	F	方法	P	R	F
Yin 错误!未找到引用源。	71	61	65	0.8	CTPN[45]	74	52	61	FRCNN[15]	75	71	73
Kang 错误!未找到引用源。	71	62	66	-	Yao 错误!未找到引用源。	72	59	65	Gupta[49]	92	76	83
Yin[44]	81	63	71	1.4	DMPNet[46]	68	73	71	Yao 错误!未找到引用源。	89	80	84
Zhang[19]	**83**	67	74	2.1	TextSpotter[47]	65	79	71	DeepText[22]	85	81	85
Yao 错误!未找到引用源。	77	**75**	**76**	0.6	OrientedText[48]	77	75	76	CTPN[45]	93	83	88
RRPN	82	68	74	0.3	RRPN	82	73	77	RRPN	90	72	80
RRPN*	82	69	75	0.3	RRPN*	**84**	77	**80**	RRPN*	**95**	**88**	**91**

因此采用去除 80%的“###”样本之后的训练集进行训练，这样能够获得最好的

评测结果（如表 2-3 中栏的 RRPN 方法），得到的结果比当前的最好方法提高了一个百分点，F-measure 达到了 77%。

（3）ICDAR2013

为了验证 RRPN 模型的适应性，我们在水平检测数据集 ICDAR2013 上也进行了实验。我们将 ICDAR2015 上最好结果的模型在 ICDAR2013 数据集上进行测试，测试的旋转矩形会转换成为外接的水平矩形框进行评测。评测结果的 F-measure 也达到了 80%，在相同训练集的情况下比表 2-3 右栏中显示的 Faster R-CNN[15] 算法还要高出 7%，这也成功的证明了有向检测框比水平检测框更能准确定位字符区域。

（4） 综合模型结果

针对不同的数据集最后整合所有的训练集对检测器进行统一训练，结果相较单一数据集上训练的结果有了进一步的提升，更是在 ICDAR2013 和 ICDA2015 两个数据集上比现有方法高出了 3 个百分点足以证明有向框算法 RRPN 模型的优越性（如RRPN*方法）。

2.3.3 检测结果可视化及讨论

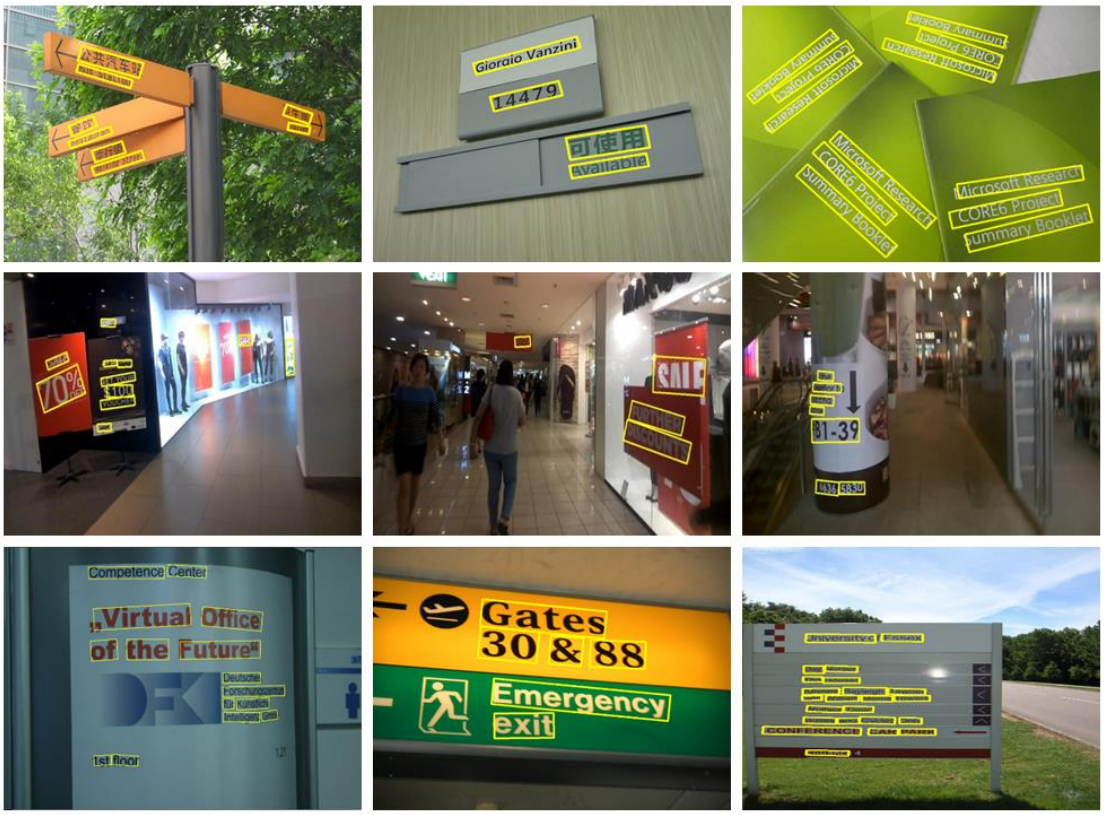


图 2-10 各大标准数据集的检测结果；按行排列依次是 MSRA-TD500、ICDAR2015 和 ICDAR2013 数据集的检测结果

为了更详细的了解模型的检测性能，笔者对训练模型的得到的检测结果可视

化，如图 2-10 所示，第一行为 MSRA-TD500 数据集的检测结果，检测器针对平面旋转具有很好的适应性，旋转候选框在视觉上给字符区域提供了比水平候选框更为紧致的检测结果。而 ICDAR2015[24]数据集中的样本的难点不仅仅有透视变换，还有样本大小变化大的情况，但是从第二行的检测结果可以看出，针对透视变换的字符区域旋转候选框也能够给出字符的阅读朝向。不同尺寸大小的字符区域，模型的检测结果也具有很强的适应性。而第三行 ICDAR2013[23]数据集也同样说明了本文提出的方法具有更强的适应性，尽管是水平检测数据集，ICDAR2013 中也存在着很多透视变换和平面旋转的字符区域，RRPN 模型的检测结果甚至比水平的标注框更为紧致。

尽管检测器已经在标准数据集上获得了世界领先的性能，但是在结果分析的过程中我们仍然发现了一些难以检测的字符情况，图 2-11 中展示了一部分表现较差的检测结果。由于预先设置的锚点框过于稀疏，无法覆盖所有大小尺寸的字符区域，再加上特征图缩放过于严重（是原图大小的 0.0625 倍），导致检测器对小字符区域不够敏感，因此表现较差（小字符图片的召回率只有 0.54）。同时，图片样本中也存在颜色光照难以和背景区分的困难情形，图片中字符的质量较差也是导致检测器性能变化的原因之一（两张图片总共的召回率只有 0.63）。

小字符 召回率=0.54　　字符质量 召回率=0.63

图 2-11 难以检测的字符情况

从 MSRA-TD500 的数据集检测结果来看，我们的检测结果与现有方法相比也存在一些不足。这主要归咎于数据集在不同语言上的标注风格问题。根据前文叙述，我们在进行检测的过程中针对超长的检测结果。我们的检测器产生了许多断裂框。其原因可能是在锚点框参数的设置上覆盖不全，而另一点则是在数据学习上。MSRA-TD500 数据集中包括了中文和英文两种语言场景。而两种语言的标注风格存在区别：英文采用的标注方式为按词标注（字符框长宽变化比较小）；而中文采用的标注方式为按行标注（字符框长宽比变化较大）。这两种标注的差别导致了字符框的长度变化过大，模型无法有效的对框的长宽进行学习。后续模型可以从分类上，根据不同语种进行多语种分类的改进，从而更有效的区分不同长宽分布的标注方式，改善检测效果。

2.4 小结

本章节主要介绍了旋转框检测方法 RRPN 模型的具体设计细节以及实验效果。检测性能在各大标准数据集上都体现出了它的优越性。相对于图像分割方法，RRPN 模型不仅继承了检测模型较快的速度，通过单一候选框自顶向下定义确定字符区域，也与连接组件方法有着本质上的区别。本文提出的检测方法在设计上较为统一和简洁。在第二阶段回归的网络通过 RRoI 层对特征进行自适应的池化，使得后端的多任务网络能够得到更好的学习，使得本文的方法达到了世界领先水平。而后端的多任务网络也对端到端识别的网络提供了统一的模型设计思路。随后可以在后端的多任务网络上多加字符识别的分支，从而实现能够统一检测与识别端到端训练的深度模型，模型设计的细节会在第三章作具体介绍。

第三章 端到端有向字符识别模型

3.1 端到端字符识别模型介绍

在章节 1.3 介绍中指出，深度神经网络模型出现之前的字符识别系统中的每一步都是分开独立完成的，从检测到识别无法进行统一训练和测试。直到深度神经网络模型出现后，Jaderberg 等[32]提出采用深度卷积网络对检测出的图片区域进行建模，最后输出字符序列，端到端识别任务才开始普遍采用卷积网络进行建模。

而 Fast R-CNN[14]出现后，研究学者发现 Fast R-CNN 后端的多任务网络除了可以利用主干网络的特征进行字符检测之外还能同时进行识别任务[33][34]，这样的多任务思想则开始逐渐统一检测和识别任务。由于本文提出的 RRPN 检测器与 Faster R-CNN 相似，都是二阶段（Two-Stage）回归的检测器，因此同样可以以多任务学习（Multi-Task Learning）的方式，在 RRPN 检测器的基础上增加识别分支，从而使得整个端到端识别系统能同时对检测与识别进行训练。除此之外，模型设计中还对 RRoI 池化的特征做了进一步的加强，RRPN 检测器只采用单层特征用于多任务学习，而对识别任务来说是不够鲁棒的。这是因为识别模型在对字符进行分类的过程中对轮廓的辨别十分重要，仅仅采用高层低分辨率的特征进行识别部分的学习得到的效果比较差，因此模型还采用特征金字塔池化（Pyramid Feature Pooling）的方式整合高低层的特征，加强识别的学习效果。

3.2 模型设计

在原本 RRPN 基础上，本文在 RRoI 池化层之后增加了进行识别任务的网络分支。主要由卷积层和循环神经网络构成。由此，文章提出的端到端识别网络在后端的多任务学习部分会学习三个任务：字符区域分类、字符区域坐标回归，以及字符区域内序列的预测（如表 3-1）。那么本章节将具体介绍识别分支的设计细节以及对特征样本的学习过程。

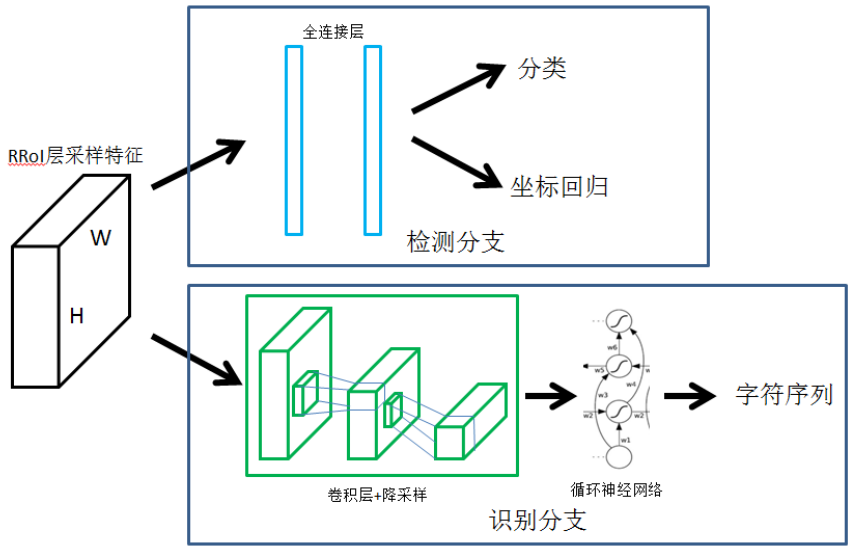


图 3-1 端到端字符识别系统多任务端设计

3.2.1 识别分支网络

主干网络（Backbone）中获得的特征通过 RRoI 池化层采样，将特征图中不同大小的区域采样为长宽固定的特征。因此不同分辨率的特征图模型可以通过该层将对应的字符区域采样成相同大小，对 VGG16[23]主干网络的 Conv1_3，Conv2_3，Conv3_3，Conv4_3 层输出经过 RRoI 池化层产生的特征，以 RRoI 候选框为范围分别以缩放倍数为 0.5，0.25，0.125，0.0625 的缩放比例在特征图上进行相同长宽的采样。四层采样好的特征通过在通道（Channel）维度上连接（Concatenation）的方式进行特征融合。这样的特征结合方式，我们定义为特征金字塔池化（Feature Pyramid Pooling），图 3-2 详细展示了操作细节及参数。通过这样的特征融合，我们能够将主干网络中的高低层特征都充分利用，增强后端多任务识别分支的识别性能。

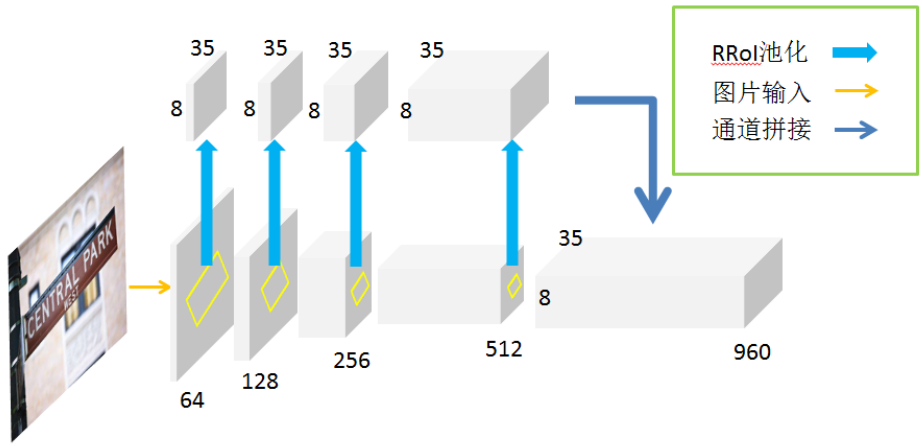


图 3-2 金字塔特征池化操作细节图示

在网络中，RRoI 特征的宽 W 和高 H 的参数分别设置为 35 和 8。随后经过 3 组相同的网络层组合，每个组合中有两组卷积核为 3×3 的卷积层（Conv）、批归一化层（BatchNorm）、非线性层（ReLU）的组合，最后通过最大池化层按高度为 2，宽度为 1 的参数设置进行降采样，将特征高度减半，由于特征高度为 8，经过 3 次降采样之后变为 1。送入随后的单层长短期记忆(Long Short-Term Memory, LSTM）层。由于字符之间往往存在前后的依赖关系，例如英文单词中会频繁出现“ing”、“ious”等固定组合，所以在网络中加入 LSTM 层可以对字符前后之间的关系进行建模，增加识别序列预测的鲁棒性。因此，LSTM 对于前面特征的输入会产生 35 个时间步（Time Stamp），LSTM 层的输出经过最后一层全连接层产生字符的概率，其输出维度为|S|，其中|S|代表需要预测的字符类别数，因此每个 RRoI 特征最后得出的序列总长度为 35。而根据经验统计，绝大部分数据样本的序列长度都小于 35，因此长度为 35 的总序列可以完整预测出所有包含结果的序列。

表 3-1 识别分支网络设计细节，其中 conv_bn_relu 代表卷积，批次归一化和非线性激活层组合，max_pool 代表最大池化层，bi-lstm 代表双向长短时记忆层，fc 为全连接层。

层类别	层参数 [窗口尺寸，步长]	层输出通道数
conv_bn_relu	[(3, 3), (1, 1)]	256
conv_bn_relu	[(3, 3), (1, 1)]	256
max_pool	[(2, 1), (2, 1)]	256
conv_bn_relu	[(3, 3), (1, 1)]	256
conv_bn_relu	[(3, 3), (1, 1)]	256
max_pool	[(2, 1), (2, 1)]	256
conv_bn_relu	[(3, 3), (1, 1)]	256
conv_bn_relu	[(3, 3), (1, 1)]	256
max_pool	[(2, 1), (2, 1)]	256
bi_lstm		256
fc		|S|

3.2.2 数据处理与学习策略

通过统计，我们将数据集中出现的所有字符整合成为字典，统计结果共出现了 99 个不同字符。另再增加一个空白类“-”用来表示非字符的位置。由此，最后预测序列会包含正常的 99 个字符与空白类，只要将空白类去掉即可得到标准的预测序列，因此，识别分支的分类器会产生 100 类分类概率，选取最大概率的分类即是网络预测的该位置上最可能的字符。

在加入了识别分支之后，整个端到端系统的监督信息（Supervision）除了字符区域的位置坐标信息之外，还有与坐标信息对应的标注序列。RRPN 的第一步坐标回归部分保持不变，分类与回归的损失计算与 RRPN 模型一致；而在第二步多任务学习的过程中，RRPN 产生的第一步旋转候选框经过与标注框计算交并比，挑选出大于 0.7 的候选框作为正样本，小于 0.3 的候选框为负样本作为后端第二次检测任务的学习样本。如图 3-3 所示，对于识别分支，只需要挑出其中的正样本进行学习即可，因为足够大交并比的候选框才可能基本覆盖字符区域中的所有文字，从而在识别分支进行训练的时候不会学习到过于困难的样本。

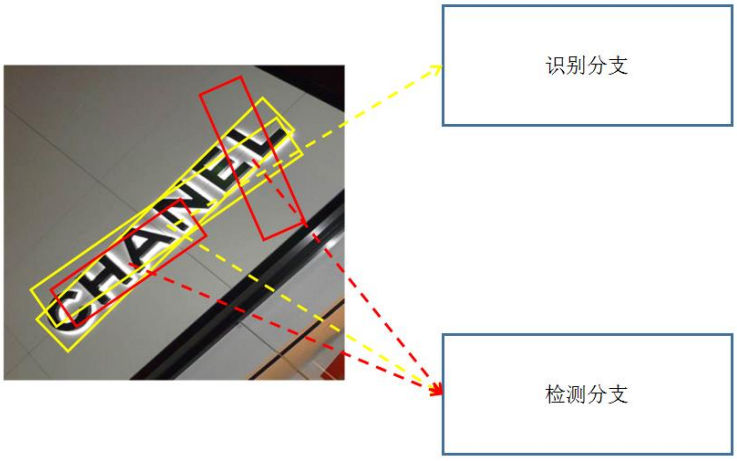


图 3-3 识别分支与检测分支的样本挑选，红色框为交并比小于 0.3 的候选框；黄色框为交并比大于 0.7 的候选框

对于字符区域中字符序列预测的准确程度本文主要采用连接时序分类（Connectionist Temporal Classification, CTC）损失[35]来衡量。连接时序分类所产生的概率能够不受字符所在具体位置的影响。因此在进行训练时，监督信息只需要提供标注序列就能够让模型预测精确的序列，而忽略序列中每个字符应该处在什么位置，也减少了标注的工作量。

CTC 的分类条件概率可以简略地用以下式子表示：对于一个长度为 T 的输入概率序列 $y = \{y_1, \ldots, y_T, y_t \in R^{|L|}\}$，L为数据集中出现的所有字符集合，多加入一个空白符“blank”用于填充序列中的非字符位置。由此定义 $\mathcal{B}$ 是一个长度为 T 的序列 π 上的序列到序列的映射函数。例如，映射函数 $\mathcal{B}$ 会将预测序列“--g-o-oo--d”映射到真实序列“good”。映射的条件概率则定义为序列 π 所有可能的到真实标注序列 l 的映射 $\mathcal{B}$ 概率和：

$$p(l|y) = \sum_{\pi:\mathcal{B}(\pi)=l} p(\pi|y) \tag{3.1}$$

其中序列π的条件概率被定义为$p(\pi|y)=\prod_{t=1}^{T}y_{\pi_t}^{t}$，$y_{\pi_t}^{t}$则是t时刻选取字符$\pi_t$的概率，最后得到的概率通过负对数计算获得最终的损失，以此评估预测序列与真实序列的差距。而识别部分通过优化损失从而对样本的字符序列进行学习。

3.2.3 系统损失函数设计

端到端识别模型的目标函数对分类、坐标回归和识别三个任务进行同时优化。对于检测任务部分的目标函数由章节2.2.2所述可以表示如下：

$$L_{Det}=\frac{1}{N}\sum_{k=1}^{N}L_{cls}(p_k,p_k^*)+\frac{1}{N_+}\sum_{k=1}^{N_+}L_{reg}(d_k,d_k^*) \tag{3.2}$$

其中N代表所有采样的样本数，在训练时主要设定批次数为128，其中包含的正负样本数量比例最多不大于1:3。而N_+则表示正样本的数量，N_+的数量不大于32。字符二分类部分损失L_{cls}主要优化交叉熵损失（式2.2），而坐标回归部分主要采用smooth-L1损失(如式2.3),而端到端系统的识别部分损失函数L_{rec}定义如下：

$$L_{rec}\left(s^{(k)},Y^{(k)}\right)=-logp(s^{(k)}|Y^{(k)}) \tag{3.3}$$

针对每个候选框对应的池化特征得到的预测序列$Y^{(k)}$，它得到标注序列的条件概率为$p(s^{(k)}|Y^{(k)})$，那么整个系统将会对该样本的预测序列负对数条件概率进行优化。至此，整个端到端系统的目标函数以多任务损失L_{E2E}的形式进行列出：

$$\begin{aligned}L_{E2E}=&\frac{1}{N}\sum_{k=1}^{N}L_{cls}(p_k,p_k^*)+\frac{1}{N_+}\sum_{k=1}^{N_+}L_{reg}(d_k,d_k^*)\\&+\frac{1}{N_+}\sum_{k=1}^{N_+}L_{rec}(s^{(k)},Y^{(k)})\end{aligned} \tag{3.4}$$

其中p_k，d_k和$Y^{(k)}$分别代表第i个字符候选框内字符特征预测出的类别，字符坐标位置和字符序列，而p_k^*，d_k^*和$s^{(k)}$则分别代表标注类别概率，标注真实坐标与锚点框的差值和标注序列。识别项中只采集正样本进行训练，因此用于识别训练的总样本数为N_+。

3.3 训练数据集与模型训练设置

本章节笔者将具体介绍整个识别系统的训练集组织方式以及训练细节。

3.3.1 组织训练集

模型训练过程中采用 Syn800K[36]生成数据集，ICDAR2013[24]与 ICDAR2015[25]进行训练。其中，Syn800K 数据集为人工合成的字符端到端识别数据集，数据集通过泊松融合的方式将字符串的像素图与手动采集的背景图进行合成。该合成数据集提供字符区域的四点坐标信息以及字符区域内的字符序列，总共有 80 万训练数据。

3.3.2 数据增强

（1）尺寸扰动

在原有训练集的基础上，还需要对输入图片在尺寸上进行扰动，扩张训练数据的多样性。每张图片在输入时都在$[(720,1280),(900,1600),(1080,1920)]$中随机选取一组高宽组合进行缩放。在缩放的过程中保持原图的长宽比例，保证图片不会变形，不足的部分采用像素 0 进行补足填充。

（2）旋转

在 RRPN 检测器的训练当中本文也进行过对图片旋转的数据增强方式（任意旋转$[0,360]$中任意角度），但是由于识别部分对字符行的朝向有要求，因此在识别时，对图片的旋转角度不能过大，保证字符区域的初始最左端与最右端不能颠倒。为了简便，我们将旋转角度的随机范围调整至$[-30,30]$。每张输入图片都会在该区间内随机挑选一个角度进行旋转。

3.3.3 训练策略

由于 Syn800K 数据量与 ICDAR2013 和 ICDAR2015 相差过大，因此，在进行训练的过程中，我们会按三个数据集样本按等比例进行采样，来保证样本的平衡。训练时将最大训练轮数设为 35 万轮，其中前 20 万轮以 0.001 的学习率进行训练，用以保证模型较快的收敛速度，也避免训练过程过早进入局部最优，剩下的 15 万轮以 0.0001 的学习率进行微调，其他参数与第 2 章中训练 RRPN 检测器的设置一致。

3.4 对比实验以及性能分析

本小节将对识别系统的算法各方面性能进行详细分析，笔者将从识别速度、多任务性能以及识别系统的识别性能对比三个方面进行对比，立体评价端到端识别模型的整体性能。整个识别系统的训练流程在 NVIDIA GTX 1080Ti 型号显卡上完成。

3.4.1 测试速度比较

对于端到端训练的识别模型来说，图片中的字符序列和坐标位置可以同时得出，相比于两项任务分开完成来说，减少了特征的重复计算，在运行速度则理所当然会上比两项任务分开进行要高。对此我们也进行了定量测速。为了平衡性能和速度，端到端的识别模型中，模型会选取 RRPN 预测得到的字符分类打分在前 600 的候选框内字符区域进行识别。而分开两个任务的实验本文采用两个不同算法分开实现，检测部分采用 RRPN 对字符区域进行检测，识别部分则采用 CNN-RNN 模型。同时对打分前 600 候选框中的字符区域进行识别，程序从原图上在候选框对应的区域截取，送入 CRNN 模型进行识别。完成此次对比实验。输入检测器的图片尺寸我们保持在宽为 1700 像素，高为 1000 像素。则最后的测速结果如表 3-2 所示：

表 3-2 端到端识别系统识别速度与识别检测任务分别进行的时间效率对比

算法	检测部分（秒）	识别部分（秒）	总共时间（秒）
端到端识别系统	-	-	0.48
RRPN+CRNN_B1	0.37	4.53	4.90
RRPN+CRNN_B128	0.37	1.26	1.63

笔者提出的端到端多任务有向字符识别系统只需要 0.48 秒就能完成整张图中所有字符区域的预测，而在完成检测之后单张图顺序输入识别网络，则需要 4.90 秒才能完成整张图的识别过程（如表 3-1 中的 RRPN+CRNN_B1 方法），为了加快识别部分的速度，我们将识别部分的输入以批次为 128 输入 CRNN 进行并行，最后的总耗时仍然需要 1.63 秒（如表 3-1 中 RRPN+CRNN_B128 方法）。仍然是端到端识别系统的 3 倍多。因此在识别效率上，端到端识别系统相较分开进行的算法更高。

3.4.2 识别分支对检测器性能的改善

我们认为字符的检测任务与识别任务本质上应当是有相关而非独立的，因此在端到端系统当中将两个任务放在一起训练，识别分支应当能够增强检测器的分支的检测性能。如表 3-3 中的前两行的实验结果也可以看出，我们对识别分支与检测分支进行联合训练之后获得的检测性能相较单独训练在 F-measure 上有了 1.52%的提升。

如第二章节中所述，RRPN 检测器最后输出的结果会过滤掉字符分类打分过低的字符候选框。而在端到端识别系统中，我们除了能够获得候选框区域的分类打分之外，还能获得字符区域中每个字符的打分。相对于笼统的两分类打分（判断是否是字符），识别分支的打分能够更细致的区分候选框区域中的内容是什么字

符，这个打分则可以更细致的区分出一些像字符的误检区域，让最后的检测结果更为准确，减少误检结果。其字符序列打分计算方式如公式 3.4：

$$S_{seq} = \frac{1}{|l_+|}\sum_{l_i \in l_+}^{l} p(l_i) \tag{3.4}$$

由于识别分支得到的是整个预测序列的概率组合，并不是单一可比较的概率分数。因此对长度为L概率序列$l = (l_1, \ldots, l_L)$中非空白符，即$l_i \in l_+$的字符概率$p(l_i)$进行了平均，获得了该序列的打分 S_{seq}。结合检测分支的字符分类打分，我们利用两个阈值进行联合筛选：若一个候选框区域满足检测打分在 0.8 以上且字符序列打分在 0.8 以上，模型才将这个候选框加入最后的输出结果。如表 3-2 中最后一行显示的测试结果，经过双阈值的筛选，检测结果的准确率从 72%提升到了 78%，而召回率没有明显的变化。说明双阈值的筛选能够有效的区分出某些误检样本，两种打分结果的可视化效果如图 3-4（a）所示：

(a)

	情形	分类打分	序列打分
简单字符样本	1	0.98	0.97
	2	0.99	0.99
误检测样本	3	0.84	0.51
	4	0.80	0.54
难字符样本	5	0.89	0.97
	6	0.61	0.78

(b)

图 3-4 序列打分对检测结果的优化和矫正。（a）各种情形下检测结果展示；（b）每种情形的两种打分统计

对于某些难以分辨的误检，其检测分类打分可以很高，如图 3-4（b）所示情形

3 和 4 是误检样本，然而检测模型的分类打分仍然很高，分别有 0.8 和 0.84，但是序列打分却很低，只有 0.51 和 0.54。因此对序列打分设置合理阈值即可有效抑制误检样本。而对于难样本字符，字符序列打分也可以保持较高水平，从而增加难样本的检测置信度。如图 3-4 中的情形 5、6，两种情形的样本分别遭遇了模糊和极端形变，导致分类打分偏低，而序列打分高于分类打分，因此相对于分类打分，按照序列打分合理设置阈值能够得到更高的准确率。

表 3-3 端到端训练对检测器的性能改善，实验在 ICDAR2015 上进行

方法	召回率	准确率	F-measure
只训练检测器	61.96%	71.94%	66.58%
联合训练	**64.08%**	72.65%	68.10%
联合训练 +双阈值筛选	63.75%	**78.41%**	**70.32%**

3.4.3 ICDAR2015 数据集上的识别性能比较

为了体现多任务模型在有向字符识别上的优越性，我们选择在有向数据集 ICDAR2015 上进行端到端识别任务性能的检验。如表 3-4 所示，

表 3-4 ICDAR2015 数据集端到端识别各个算法的精度（F-measure）

方法	端到端评估（End-to-End）			单词识别（Word Spotting）		
	强语境	弱语境	全字典	强语境	弱语境	全字典
TextSpotter[52]	0.35	0.20	0.16	0.37	0.21	0.16
Stradvision[25]	0.44	-	-	0.46		
TextProposals + DictNet[50][51]	0.53	0.50	0.47	0.56	0.52	0.50
Deep TextSpotter[34]	0.54	0.51	0.47	0.58	0.53	0.51
本文算法	**0.57**	**0.53**	**0.50**	**0.64**	**0.57**	**0.55**

本文提出的端到端识别算法在 ICDAR2015 上相较之前的候选框方法取得了更好的成绩，这得益于 RRPN 检测器提供的有向候选框，让识别部分得到的更为紧致的字符区域，送入识别分支之后再经过金字塔特征池化，对主干特征进行了更好的利用，增强了最后识别分支的识别结果。在两方面的因素下，本文方法超越了先前端到端识别的方法。TextProposals+DictNet[50][51]检测方法尽管针对字符的特性做了优化，但是最后检测的结果仍然只采用水平候选框，导致字符区域的紧致性不足，字符识别部分的算法性能仍然受到影响。Deep TextSpotter[34]方法虽然也采用了有向候选框产生更为紧致的字符区域，但是由于该方法在识别上对主

干网络的特征利用不够充分，且产生的候选框只经过一次矫正，检测效果较差，因此对字符序列的识别性能有限。因此文章提出的模型相较上述两种候选框方法，具有更好的识别性能。

3.4.4 识别结果可视化及分析

为了更为直观的展示端到端识别系统的算法性能，我们对最终的识别结果也进行了可视化，如图 3-5 所示。尽管 ICDAR2015 数据集存在很多小字符以及透视变换等检测识别难点，模型采用基于有向候选框和多层特征利用的方式也能够达到比较好的识别效果，但是仍然有很大的空间可以提升。

图 3-5 ICDAR2015 端到端识别结果可视化

识别结果可视化分析发现，我们的识别系统仍然存在局限性（如图 3-6 所示）。算法仍然存在对小字符区域仍然难以辨识的问题，由于字符区域过小因而造成了字符模糊，加之图像特征的分辨率不足以让识别系统进行分辨，导致了误识别的情况；其次，在检测过程中产生的误检框，将多条字符区域并做一个字符区域，多行字被当成单行进行识别，也是一个导致误识别的重要原因。

图 3-6 误识别样例

3.4.5 中文场景讨论

由于当下的场景字符识别大多数研究都在英文和拉丁字母作为所有字符集的设定下展开。总共类别数在 37 类左右（可能包括一些特殊符号的扩充等）比较少，且只能适应英文场景。对于中文场景下的端到端字符识别，目前仍是一个及其困难的任务场景，目前也没有十分规范的数据集来对中文场景的识别任务跟进这项任务。最近提出的多语境数据集 ICDAR-MLT[53]上有少部分的中文数据样本可以供我们进行中文场景下的实验。

ICDAR-MLT 数据集中的数据样本主要取自不同语言环境下的街景拍摄，其中字符的形状大多为不规则的四边形，拍摄的文字区域都有不同程度的透视变换。其中包括的语言有中文，英文，法文，孟加拉文等共 9 种语言。其训练集数量为 7200 张，验证集为 1800 张。我们选取其中 800 张中文场景图片作为训练集，进行训练。经过统计，我们总共需要预测的类别数为 4746。其中包括英文字母、数字、中文字符和部分特殊符号。我们在进行试验时保持与在 ICDAR2015 数据集上相同的参数设置。识别效果展示如图 3-7 所示。

图 3-7 中文识别结果展示

尽管模型能够准确的识别出某些简单轮廓的字符，但是大多数复杂结构的中文字仍然不够准确，模型在识别的过程中会容易将要识别的字符与其中文形近字相混淆，导致识别结果错误。因此在中文的复杂场景中的识别仍有待改善。其中一部分原因与数据集的数据偏差存在着一定的关系。经过数据集统计，我们发现训练的数据存在一定的偏斜，如图 3-8 所示。

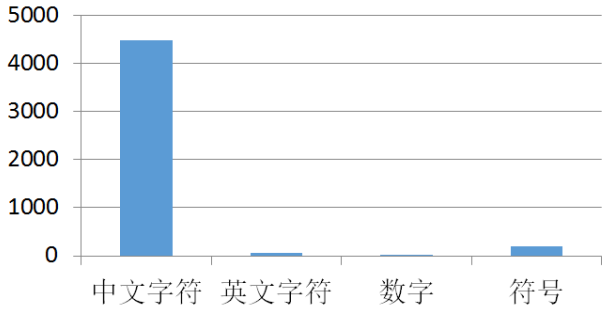


图 3-8 字符类别统计

其中中文字符类别为 4485 个，占到总类别数的 90%以上，而英文数字和其他符号仅占很小比例。这样的类别分布可能会导致最大概率的类标概率降低，不利于最终结果的预测。我们再对数据集中的字符频率进行统计，并对频率最高的前 100 个字符按降序排列进行可视化，如图 3-9 所示：

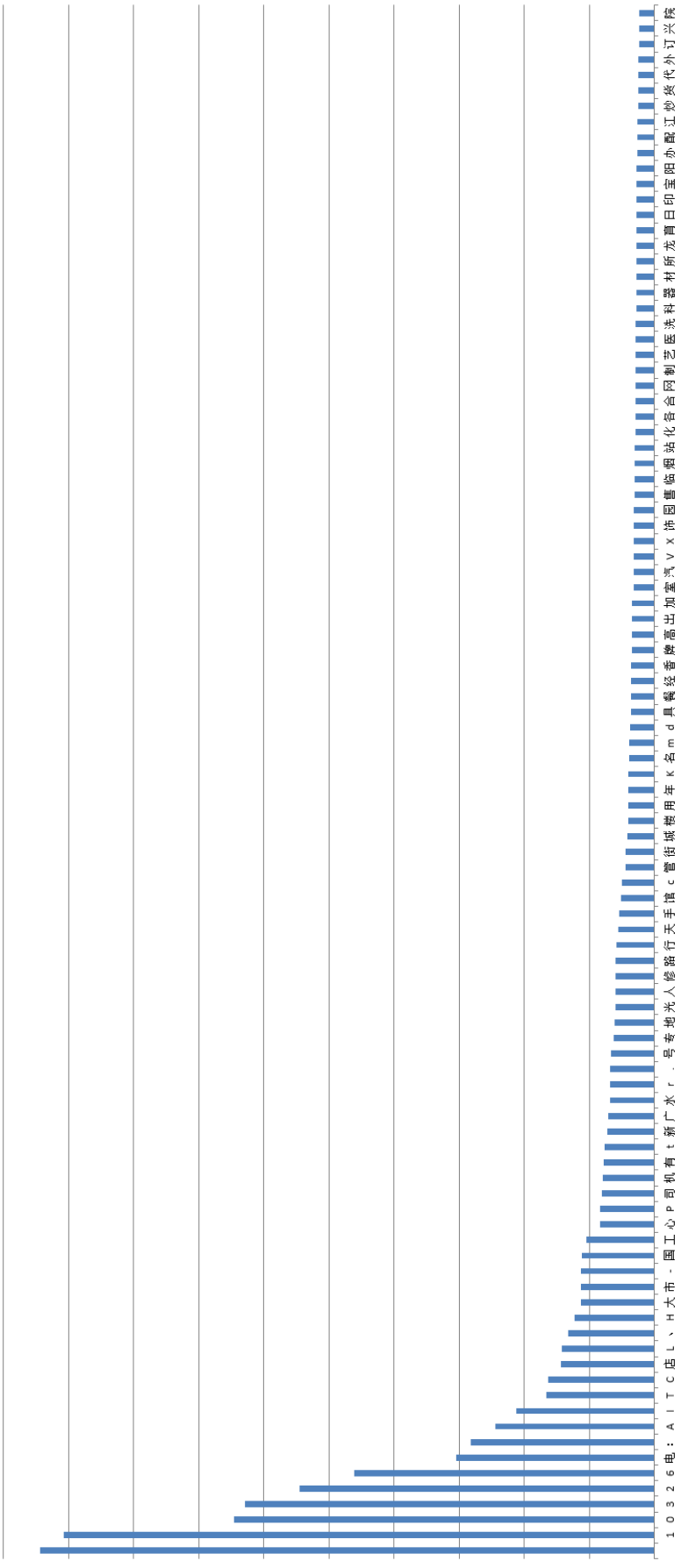


图 3-9 字符频率前 100 的字符统计结果

统计图里前 100 结果的样本占到了整个数据集的 80%以上，而类别数仅仅是所有类别的 4%。说明数据集出现了严重的数据偏斜。在现有的中文场景数据集中，“长尾”现象普遍存在，样本分布的偏差导致模型对中文学习较差。因此，我们可以通过生成数据的方式去补足出现频率较低的字符类别，让数据集分布更为均衡。此外，输入识别部分网络的特征分辨率仅为原来的 1/16 倍。在这样的分辨率下，文字特征能够表征的字符轮廓信息有限，区分性较差，因此容易与形近字混淆。由此来看，模型的识别部分网络设计仍然有很大的改善空间。

3.5 小结

本章节主要介绍了整个端到端可训练的识别系统的模型搭建以及训练实验。由于字符识别和字符检测任务应当是相互关联的，因此我们以多任务训练的方式将两个任务进行了联合训练。最终的实验结果也证明了字符识别网络分支对检测分支具有促进作用。与此同时，联合识别与检测两个任务可以减少网络特征的重复计算，加快识别的速度，在 ICDAR2015 标准数据集上的识别效果也证明了本文提出的端到端算法的优越性。但是识别模型在中文数据集上的表现仍然不足，除了数据集本身存在着“长尾”效应之外，模型对字符轮廓不敏感的问题仍亟待解决。

第四章 总结

4.1 文章总结

本文针对字符检测任务提出了一种有向检测框来检测字符区域的算法，有向检测框相对于水平检测框对字符区域具有更紧致的表示。我们针对有向检测框设计了能够感知字符区域朝向的 RRPN 检测模型，RRPN 模型在检测字符区域时能够将字符区域的可能朝向进行预测。预测的朝向能够将透视变换的字符进行矫正，从而使得识别算法能够更好的对字符进行识别。RRPN 检测算法在 MSRA-TD500,ICDAR2013 和 ICDAR2015 标准数据集上取得的优越性能已经初步证明了有向候选框检测结果的优越性。然而这并不能完全说明紧致的检测框真正对识别有效，因此我们对此在 RRPN 后端搭建了字符识别分支，将网络设计成为端到端可训练的识别系统，并且在 ICDAR2015 任意朝向数据集上取得了最好成绩，从识别任务上进一步证明了有向候选框确实能够改善后端字符识别算法的性能。

4.2 研究未来展望

虽然本文提出了紧致的字符候选框可以加强识别性能，但是从识别结果来看，现有的算法仍然面临很多难以解决的问题，例如对小字符区域不敏感，过于严重的仿射变换字符难以识别等。我们未来将进一步进行的工作将会从下面几点展开：

1．端到端识别算法在推理速度上相较分开识别在速度上有提升，但是仍然需要消耗很多计算资源，在一般 CPU 上仍然无法达到实时，加快识别的速度，减少算法的计算资源仍然是一个值得研究的内容。

2．从图像中获得文字信息之后，我们可以将这些文字信息用于更高级的图像逻辑推理任务当中，增强计算机对图像信息的获取能力，增加推理的信息来源，让计算机的逻辑推理算法更为鲁棒。

3. 中文字符也是字符识别中一个重要的任务，中文字符相比英文字符，轮廓更为复杂，类别更为繁多。如何让模型有效区分更为复杂的轮廓也是一个值得探究的方向。

参考文献

[1] Karaoglu S, Tao R, Gevers T, et al. Words matter: Scene text for image classification and retrieval[J]. IEEE Transactions on Multimedia, 2016, 19(5): 1063-1076.

[2] Bai X, Yang M, Lyu P, et al. Integrating scene text and visual appearance for fine-grained image classification[J]. IEEE Access, 2018, 6: 66322-66335.

[3] Yin X C, Zuo Z Y, Tian S, et al. Text detection, tracking and recognition in video: a comprehensive survey[J]. IEEE Transactions on Image Processing, 2016, 25(6): 2752-2773.

[4] Liu X, Wang W. Robustly extracting captions in videos based on stroke-like edges and spatio-temporal analysis[J]. IEEE transactions on multimedia, 2011, 14(2): 482-489.

[5] Bouman K L, Abdollahian G, Boutin M, et al. A low complexity sign detection and text localization method for mobile applications[J]. IEEE Transactions on multimedia, 2011, 13(5): 922-934.

[6] Goel V, Mishra A, Alahari K, et al. Whole is greater than sum of parts: Recognizing scene text words[C]//2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013: 398-402.

[7] Bissacco A, Cummins M, Netzer Y, et al. Photoocr: Reading text in uncontrolled conditions[C]//Proceedings of the IEEE International Conference on Computer Vision. 2013: 785-792.

[8] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//international Conference on computer vision & Pattern Recognition (CVPR'05). IEEE Computer Society, 2005, 1: 886--893.

[9] Pan Y F, Hou X, Liu C L. Text localization in natural scene images based on conditional random field[C]//2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009: 6-10.

[10] Canny J. A computational approach to edge detection[M]//Readings in computer vision. Morgan Kaufmann, 1987: 184-203.

[11] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[12] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[13] Uijlings J R R, Van De Sande K E A, Gevers T, et al. Selective search for object recognition[J]. International journal of computer vision, 2013, 104(2): 154-171.

[14] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[15] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.

[16] Yi C, Tian Y L. Text string detection from natural scenes by structure-based partition and grouping[J]. IEEE Transactions on Image Processing, 2011, 20(9): 2594-2605.

[17] Epshtein B, Ofek E, Wexler Y. Detecting text in natural scenes with stroke width transform[C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010: 2963-2970.

[18] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

[19] Zhang Z, Zhang C, Shen W, et al. Multi-oriented text detection with fully convolutional networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4159-4167.

[20] Matas J, Chum O, Urban M, et al. Robust wide-baseline stereo from maximally stable extremal regions[J]. Image and vision computing, 2004, 22(10): 761-767.

[21] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]//European conference on computer vision. springer, Cham, 2014: 818-833.

[22] 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.

[23] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[24] Karatzas D, Shafait F, Uchida S, et al. ICDAR 2013 robust reading competition[C]//2013 12th International Conference on Document Analysis and

Recognition. IEEE, 2013: 1484-1493.

[25] Karatzas D, Gomez-Bigorda L, Nicolaou A, et al. ICDAR 2015 competition on robust reading[C]//2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015: 1156-1160.

[26] Jaderberg M, Simonyan K, Zisserman A. Spatial transformer networks[C]//Advances in neural information processing systems. 2015: 2017-2025..

[27] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[28] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009: 248-255.

[29] Yao C, Bai X, Liu W. A unified framework for multioriented text detection and recognition[J]. IEEE Transactions on Image Processing, 2014, 23(11): 4737-4749.

[30] Zhang Z, Zhang C, Shen W, et al. Multi-oriented text detection with fully convolutional networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4159-4167.

[31] Yao C, Bai X, Sang N, et al. Scene text detection via holistic, multi-channel prediction[J]. arXiv preprint arXiv:1606.09002, 2016.

[32] Jaderberg M, Vedaldi A, Zisserman A. Deep features for text spotting[C]//European conference on computer vision. Springer, Cham, 2014: 512-528.

[33] Li H, Wang P, Shen C. Towards end-to-end text spotting with convolutional recurrent neural networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 5238-5246.

[34] Busta M, Neumann L, Matas J. Deep textspotter: An end-to-end trainable scene text localization and recognition framework[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2204-2212.

[35] Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 369-376.

[36] 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.

[37] Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 369-376.

[38] He P, Huang W, Qiao Y, et al. Reading scene text in deep convolutional sequences[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[39] Jaderberg M, Simonyan K, Vedaldi A, et al. Reading text in the wild with convolutional neural networks[J]. International Journal of Computer Vision, 2016, 116(1): 1-20.

[40] Yao C, Bai X, Liu W, et al. Detecting texts of arbitrary orientations in natural images[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012: 1083-1090.

[41] Yao C, Bai X, Liu W. A unified framework for multioriented text detection and recognition[J]. IEEE Transactions on Image Processing, 2014, 23(11): 4737-4749.

[42] Yin X C, Yin X, Huang K, et al. Robust text detection in natural scene images[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 36(5): 970-983.

[43] Kang L, Li Y, Doermann D. Orientation robust text line detection in natural images[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 4034-4041.

[44] Yin X C, Pei W Y, Zhang J, et al. Multi-orientation scene text detection with adaptive clustering[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1930-1937.

[45] Tian Z, Huang W, He T, et al. Detecting text in natural image with connectionist text proposal network[C]//European conference on computer vision. Springer, Cham, 2016: 56-72.

[46] Liu Y, Jin L. Deep matching prior network: Toward tighter multi-oriented text detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1962-1969.

[47] Qin S, Manduchi R. Cascaded segmentation-detection networks for word-level text spotting[C]//2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, 1: 1275-1282.

[48] Shi B, Bai X, Belongie S. Detecting oriented text in natural images by linking segments[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2550-2558.

[49] Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2315-2324..

[50] Gómez L, Karatzas D. Textproposals: a text-specific selective search algorithm for word spotting in the wild[J]. Pattern Recognition, 2017, 70: 60-74.

[51] Jaderberg M, Simonyan K, Vedaldi A, et al. Reading text in the wild with convolutional neural networks[J]. International Journal of Computer Vision, 2016, 116(1): 1-20.

[52] Neumann L, Matas J. Real-time lexicon-free scene text localization and recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 38(9): 1872-1885.

[53] Bušta M, Patel Y, Matas J. E2E-MLT-an unconstrained end-to-end method for multi-language scene text[J]. arXiv preprint arXiv:1801.09919, 2018.

致谢

感谢父母在我研究生期间对我无微不至的关怀以及对我研究生学业的大力支持，他们对我的支持是我继续研究和学习最好的动力。同时也感谢我的导师薛向阳教授在我的研究生期间对我研究方向的大力支持和学业上的帮助，他给我们提供了非常充足的实验资源供我们进行科学研究，是我们能够成就科学成果的基础；在平时组会过程中对同学们的教导，一直激励着我不畏困难努力前行。感谢小组的李斌研究员对我们的殷切指导，在我们遇到难题时总能站在我们的角度分析问题，对我们亦师亦友的和蔼态度让我们对学习和科研产生了热爱。感谢实验室的兄弟姐妹们，没有他们的对我鼓励和友善，也不会成就今天开朗的我。

感谢我的师兄郑莹斌博士和邵蔚元对我的悉心教导和大力支持，他们对研究领域读到和领先的见解影响着我科研的思路和作风，让我以严谨的态度对待每一个实验细节。

感谢答辩委员会各位老师和工作人员的劳动，以及对我们毕业答辩的大力支持。没有他们的敬业精神，我的答辩过程无法顺利进行。

最后，对在我研究生期间所有帮助过我的人表示衷心的感谢和祝福。

研究生期间发表论文

1. Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, & Xiangyang Xue (2018). Arbitrary-oriented scene text detection via rotation proposals[J]. IEEE Transactions on Multimedia, 2018, 20(11): 3111-3122.

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：马建奇　　日期：2019.5.29

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：马建奇　　导师签名：[illegible]　　日期：2019.5.29