单位代码 10006
学　　号 17373381
分 类 号 TP391

北京航空航天大學
BEIHANG UNIVERSITY

毕业设计(论文)

街景图像中英文文字检测与识别模型的研究与实现

院（系）名称　计算机学院
专 业 名 称　计算机科学与技术
学 生 姓 名　何逸宸
指 导 教 师　李舟军

2021 年 6 月

街景图像中英文文字检测与识别模型的研究与实现

何逸宸

北京航空航天大学

北京航空航天大学

本科生毕业设计（论文）任务书

I、毕业设计（论文）题目：

街景图像中英文文字检测与识别模型的研究与实现

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

1. 所使用的数据集：包括 Icdar 比赛从 2013 到 2019 的比赛数据集以及 COCO-Text 数据集，用于训练文字检测和文字识别模型。

2. 技术要求：设计一套完整的文字检测与识别任务处理流程，对模型进行改进，实现一个光学字符识别的实用工具，达到现有的主流水平。

III、毕业设计（论文）工作内容：

针对街景图像的特点，研究和分析文字检测领域的主流模型，设计一套完整的文字检测任务处理流程。同时将三种基于分割的文字检测方法进行拆分重组，分别优化，选出最优的组合进行应用。并利用薄板样条插值改进卷积循环神经网络(CRNN)，应用在文字检测算法之后，实现一个在线文字检测与识别系统。

IV、主要参考资料：

[1] Wang W，Xie E，Li X，et al. Shape robust text detection with progressive scale expansion network[C]. Proceedings of the IEEE Conference on Computer

Vision and Pattern Recognition. 2019: 9336-9345.

[2] Wang W，Xie E，Song X，et al. Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 8440-8449.

[3] Liao M, Wan Z, Yao C, et al. Real-time scene text detection with differentiable binarization[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34（07）: 11474-11481.

[4] 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.

计算机 学院（系） 计算机科学与技术 专业类 170614 班

学生 何逸宸

毕业设计（论文）时间： 2021 年 1 月 3 日至 2021 年 5 月 25 日

答辩时间： 2021 年 6 月 3 日

成 绩：

指导教师：

兼职教师或答疑教师（并指出所负责部分）：

系（教研室） 主任（签字）：

注：任务书应该附在已完成的毕业设计（论文）的首页。

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：何逸宸

签字：

时间：2021 年　6 月

街景图像中英文文字检测与识别模型的研究与实现

学　　生：何逸宸
指导老师：李舟军

摘要

光学字符识别技术是一种对图像中所包含的数据和文字进行定位检测与识别的技术，通过对版面布局和相关信息的分析，可获取该图像中的数据和文字信息。光学字符识别通常包含文字检测和文字识别两个步骤。文字检测用于在图像中定位文字区域，并提供其所在位置的坐标信息；文字识别进一步对该区域内的所有文字进行识别，并将其转化成计算机能够理解和处理的数据和文字信息。

本文针对街景图像的特点，深入研究和分析了目前文字检测和识别领域的主流模型，设计了一套完整的文字检测和识别任务的处理流程。本文将基于分割的文字检测模型的处理过程分为骨干网络、上采样分割头和分割输出三个阶段，并在每个阶段实现了若干可供选择的功能模块。具体而言，对于骨干网络，选择并复现了 Resnet50 模块和 Mobilenetv3 模块；对于上采样分割头，选择并复现了特征金字塔网络上采样分割头模块和特征金字塔增强与融合分割头模块，并分别对它们进行了改进与优化；对于分割输出，则选择并复现了渐进规模扩展网络模块、像素聚合网络模块和可微分二值化网络模块。在每个阶段任意选择一种模块，即可组合成一个不同的文字检测模型进行训练。通过训练结果的准确性、收敛速度、模型大小等方面的对比与分析，可以得到不同场景下最适用的模型。此外，本文同时考虑中英文字符的特点与区别，实现了一个中英文字符集上的文字识别算法，同时引入薄板样条插值对输入文字进行修复，以提高识别的准确性。

最后，本文设计并实现了一个文字检测与识别的在线系统，可提供面向通用场景下的中英文图像文字检测与识别服务。该系统可通过可视化交互界面接受用户的图像输入，系统后台经过上述的文字检测与识别模型以及后处理流程，给用户返回结构化的文字提取结果及其可视化呈现。

关键词：深度学习，光学字符识别，文字检测，文字识别

Research and implementation of the Chinese and English text detection and recognition model in scene images

Author: HE Yi-chen

Tutor: LI Zhou-jun

Abstract

Optical character recognition technology is a kind of technology to locate, detect and recognize the data and text contained in the image. Through the analysis of layout and related information, the data and text information in the image can be obtained. Optical character recognition usually includes two steps: text detection and text recognition. Text detection is used to locate the text area in the image and provide the coordinate information of its location. Text recognition further recognizes all characters in the region, and converts them into binary coded character information that can be understood and processed by computer.

According to the characteristics of street view images, this paper studies and analyzes the current mainstream models in the field of text detection and recognition, and designs a complete set of processing flow of text detection and recognition tasks. In this paper, the processing process of segmentation based text detection model is divided into three stages: backbone network, up sampling segmentation head and segmentation output, and several optional functional modules are implemented in each stage. Specifically, for the backbone network, Resnet50 module and Mobilenetv3 module are selected and reproduced; for the up sampling segmentation head, the up sampling segmentation head module and the feature pyramid enhancement and fusion segmentation head module are selected and reproduced, and they are improved and optimized respectively; and for the segmented output, the incremental scale expansion network module, the pixel aggregation network module and the differentiable binary network module are selected and reproduced. A different text detection model can be obtained by selecting any module in each stage. Through the comparison and analysis of the accuracy of training results, convergence speed, model size and other aspects, we can get the most suitable model in different scenarios. In addition, this paper also implements a character recognition algorithm on Chinese and English character set, which needs to consider the characteristics and

differences of Chinese and English characters at the same time. At the same time, thin plate spline interpolation is introduced to repair the input text.

Finally, this paper designs and implements an online system for text detection and recognition, which can provide Chinese and English image text detection and recognition services for common scenes. The system can accept the user's image input through the visual interactive interface. After the above text detection and recognition model and post-processing process, the system returns the structured text extraction results and visual presentation to the user.

Key words: Deep learning，OCR，Text detection，Text recognition

目　　录

1 绪论

1.1 课题研究背景

光学字符识别技术(OCR)是一种对图像中所包含的数据和文字进行定位检测与识别的技术，通过对版面布局和相关信息的分析，可获取该图像中的数据和文字信息。光学字符识别是信息抽取领域重要的研究内容之一。光学字符识别通常包含文字检测和文字识别两个步骤。文字检测用于在图像中定位文字区域，并提供其所在位置的坐标信息；文字识别进一步对该区域内的所有文字进行识别，并将其转化成计算机能够识别、理解和处理的二进制编码的文字信息。

光学字符识别技术不仅在深度学习的学术研究领域中具有重要的研究价值，而且同时也具有广阔的应用场景与应用需求。在光学字符识别的研究中，不仅需要应对文字种类和文字字体的差异，还要应对各种各样复杂背景的干扰。

近年来，随着深度学习技术的深入发展与应用，手工设计特征的研究手段正在逐渐被取代。得益于移动互联网中海量数据的积累，许多问题已经出现了一些规模较大的数据资源。而且图形处理器(GPU)技术的进步使得训练深度学习模型的人力物力和资源消耗大大降低，为深度学习技术的发展应用提供了坚实的基础与有力的支撑。同时文字检测和文字识别的精度和准确率也有了大幅度的提高。

街景图像由于其数据集易于获得和标注、数据量大、背景和文字复杂等特点，一直是该领域的持续关注焦点。以街景图像训练的模型在其他场景下都有不错的表现，因此本文主要关注街景图像。

1.2 国内外研究现状

1.2.1 文字检测

光学字符识别的第一个关键步骤就是文字检测，后续文字识别的准确率与整个系统的性能都与该部分密切相关。

基于回归的文字检测方法把文字区域看作一个目标物体进行目标检测，直接回归文字区域的边界框。区域文字生成网络(CTPN)[1]使用与目标检测模型中类似的方法来提取图像特征信息，对于潜在的文本区域，使用固定宽度的垂直默认检测框来进行检测，为了结合这些默认检测框的前后文信息来判断文字区域，把它们序列化为特征串作为双向

长短期记忆网络(LSTM)[3]的输入，最后用全连接层来分类或回归，并通过后处理将默认检测框合并成为文字检测框。旋转区域生成网络(RRPN)[4]对区域生成网络(RPN)[2]中的检测框增加了角度信息，可以识别出图像中具有一定旋转角度的文字区域。TextBoxes++模型[5]在单阶段的目标检测模型(SSD)[6]的基础上修改了文本检测的锚点和卷积核的尺度。并且使用了inception模块[7]来联结特征图，以扩大感受野。

基于分割的方法对图像中的每一个像素进行分类，将整个图像以像素为单位分割成文字区域与非文字区域，然后通过像素聚类等后处理方法，生成文字检测框。渐进规模扩展网络(PSENet)[10]为图像中的每个文本实例生成一系列嵌套的不同比例的核，并采用基于广度优先搜索(BFS)的渐进尺度扩展算法扩展较小的核到其紧邻的较大核逐渐得到完整的文本实例，比例不同的核与原始文本实例的形状几何相似，最小尺度的核边界彼此远离因此不容易混淆。像素聚合网络(PAN)[11]，改进了渐进规模扩展网络，删除了一系列嵌套的核，而是使用一个核和一个四维相似向量，并且使用一个可学习的像素聚合算法来重建完整的文本实例，引导文本像素聚类到正确的核。可微分二值化(DB)模型[12]是一个可以自适应地修改与预测二值化阈值的基于分割的网络，它不仅简化了后处理的操作，而且提高了文字检测的性能。可微分二值化模型将阈值图与概率图联合优化生成二值图，并对文字区域边界进行加强，从而完全区分前景和背景的像素。

1.2.2 文字识别

在深度学习方法中，文本识别问题通常被看作一个序列问题，通过编码器和解码器直接预测文本序列。受到语音识对齐方法的启发，卷积循环神经网络(CRNN) [14]在 2016 年被提出。使用卷积神经网络的卷积层和池化层提取图像的信息，所有图像在缩放到相同高度输入卷积层。卷积层的输出转化为序列化的特征串作为循环层的输入，即将卷积层的输出划分为宽度为 1 的列，从左到右按顺序将每一列转化为一个特征。将序列化信息输入深层双向长短期记忆网络中，生成标签信息，并通过连接时序分类(CTC)[15]计算损失。双向长短期记忆网络有很强的捕捉序列中上下文信息的能力且能操作任意长度的序列。而且堆叠多个双向长短期记忆网络形成深度双向长短期记忆网络，可以增强信息的提取能力。[31]采用的方法是使用深度卷积神经网络生成序列化信息，通过长短期记忆网络来处理上述特征得到最终结果。[32]将卷积神经网络替换成注意力机制的序列到序列结构，通过注意力机制可以识别输入中的关键特征进行分析。

1.3 本文研究内容

本文在对现有文字检测和识别相关工作调查和研究的基础上，分析了文字检测模型的各组成部分的优劣，找出最优组合进行训练，并改进现有文字识别网络，设计一套完整的文字检测与识别任务处理流程，实现了一个光学字符识别的实用工具，该工具可以实现上传图像并实时返回识别结果。

本文的主要研究内容如下：

（1）文字检测模型研究

针对街景图像的特点，分析深度学习时代下光学字符识别中文字检测领域的主流模型，熟练掌握基于分割的文字检测技术在光学字符识别领域的应用方法。本文将基于分割的文字检测模型的处理过程分为骨干网络、上采样分割头和分割输出三个阶段，并在每个阶段实现了若干可供选择的功能模块，做出相应的改进与优化。具体而言，对于骨干网络，选择并复现了 Resnet50 模块和 Mobilenetv3 模块；对于上采样分割头，选择并复现了特征金字塔网络上采样分割头模块和特征金字塔增强与融合分割头模块，并分别对它们进行了改进与优化；对于分割输出，则选择并复现了渐进规模扩展网络模块、像素聚合网络模块和可微分二值化网络模块。在每个阶段任意选择一种模块，即可组合成一个不同的文字检测模型进行训练。通过对每个阶段不同模块的选择，分别组合成不同的文字检测模型进行训练。对比分析各模型训练结果的准确性、收敛速度、模型大小，提出最优的模型组合。

（2）文字识别模型研究

该部分研究并实现了一个可以识别中英文字符集的文字识别算法。通过一系列前处理算法获得文字方向竖直的文本区域，同时在卷积层之前加入薄板样条插值，用于矫正输入像，以提高文字识别的准确性。

（3）文字检测与识别系统的实现

构建一个文字检测与识别的在线系统，提供面向通用场景下的中英文图像文字检测与识别服务，通过网页上的可视化交互界面接受用户的图像输入并保存到数据库，系统后台经过文字检测与识别模型以及后处理流程对输入图像进行处理，返回给用户文字提取结果的可视化呈现。

1.4 论文组织结构

论文的组织结构如下，总共包含五个章节：

第一章 绪论

本章主要包括该题目的研究背景和意义、重要性、国内外的研究现状以及本文的主要研究内容。

第二章 相关技术与原理

本章介绍了街景图像中英文文字检测与识别模型的相关研究工作。主要包括卷积神经网络和循环神经网络的构成、发展演进历程以及具体实现细节，以及用于处理分类问题的交叉熵损失函数、用于处理输出与标签值不等长的连接时序分类损失函数和用于评估集合相似度的 dice-coefficient 损失函数。最后对街景图像中英文文字检测与识别模型的框架流程进行了介绍。首先介绍了文字检测模型的上采样分割头和分割输出的算法流程中的关键步骤的实现方法和细节，然后介绍了使用的 CRNN 文字识别模型，重点介绍了该模型的三个层次以及连接时序分类损失的实现细节以及损失的计算。

第三章 文字检测与识别模型的实现与改进

本章是本文的主要内容，首先介绍了文字检测和识别网络的具体实现细节，以及对上采样分割头和文字识别模型的改进。然后介绍了整个文字检测和识别系统的实现流程，即使用统一的配置文件对系统进行训练、预测和验证。接着介绍了系统的主要功能和应用场景，结构设计和模块划分，包括用户交互模块、文本检测模块和文本识别模块，并详细介绍了各个模块的功能和设计以及实现要点。除此之外，本章还介绍了系统的开发环境和系统部署，包括前后端与深度模型开发过程中使用到的工具、框架、语言等。最后，本章进行了两种不同场景下的可视化效果展示，展示了在两种不同场景下的文本检测与识别效果。

第四章 实验设计与结果分析

本章首先介绍了实验数据集的来源和对应的评估任务，然后对各个算法的评估指标进行了详细的介绍，然后对模型效果进行了评估，并对基于分割的文字检测的三个模型进行了对比，分析了在不同环境下的最优组合。接着，通过实验证明了添加薄板样条插值模块对文字识别 CRNN 模型的检测效果有明显的改善。最后进行了消融实验，对模型的超参数选举做出了说明。

2 相关技术与原理

本章主要介绍了在文字检测与识别模型的实现与改进中使用到的相关理论与技术研究。主要围绕卷积神经网络、循环神经网络、损失函数、文字检测相关技术与文字识别相关技术五个方面进行介绍。

2.1 卷积神经网络

卷积神经网络是深度学习网络中最基本，最常见的网络之一。本节主要介绍卷积神经网络的基本组成以及经典卷积神经网络。

2.1.1 卷积神经网络基本构成

一个典型的卷积神经网络的结构如图 2.1 所示，其主要成分有以下四部分。

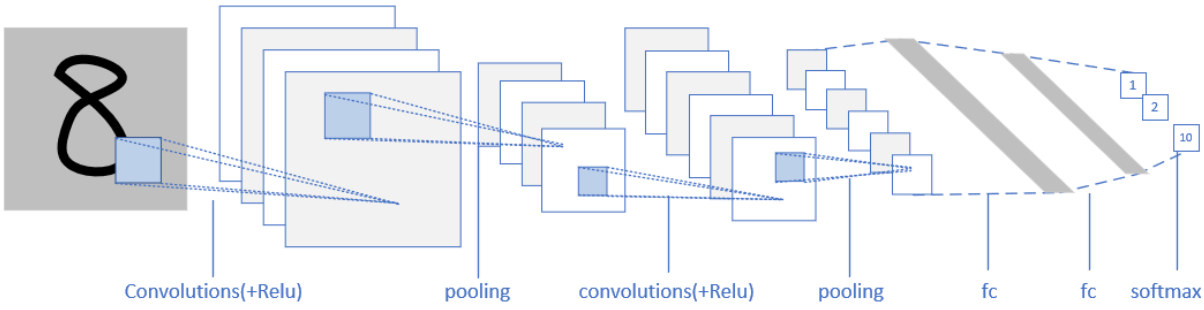

图 2.1 卷积神经网络的结构

1、卷积层

多个大小相同的卷积核构成一个卷积层，在输入图像上不断移动的卷积核可以计算得到特征图。卷积核大小、输出通道数、步长、边界填充等构成了卷积层的主要参数。在二维卷积中，卷积层输入和输出都是一个三维矩阵，即(通道数，长度，宽度)，卷积核的通道数与输入通道数相等。卷积操作是指卷积核在输入图像的特定位置进行逐元素的乘加操作生成二维输出矩阵的一个元素，卷积核在输入矩阵上按照步长不断的移动计算，就能生成了二维输出矩阵每个位置的元素。多个卷积核生成多个二维输出矩阵组合成为三维输出矩阵作为卷积层的输出。

具体来说，现有一卷积层，其卷积核大小为$(\mathrm{k}_h, \mathrm{k}_w)$，步长为$(\mathrm{s}_h, \mathrm{s}_w)$，边界填充为$(\mathrm{p}_h, \mathrm{p}_w)$，输出的通道数为$out_channels$。对于一个输入图像$(N, C_{in}, H_{in}, W_{in})$，经过该卷积层后，其输出为$(N, C_{out}, H_{out}, W_{out})$。其中，$C_{out} = out\_channels$，$H_{out} = \lfloor (H_{in} + 2 \times p_h - k_h)/s_h + 1 \rfloor$，$W_{out} = \lfloor (W_{in} + 2 \times p_w - k_w)/s_w + 1 \rfloor$。

图 2.2 是卷积神经网络中的卷积操作示例，输入为(1,3,5,5)，卷积核大小为(5,5)，步长为(2,2)，边界填充为(1,1)，输出通道数为 2，输出为(1,2,3,3)。

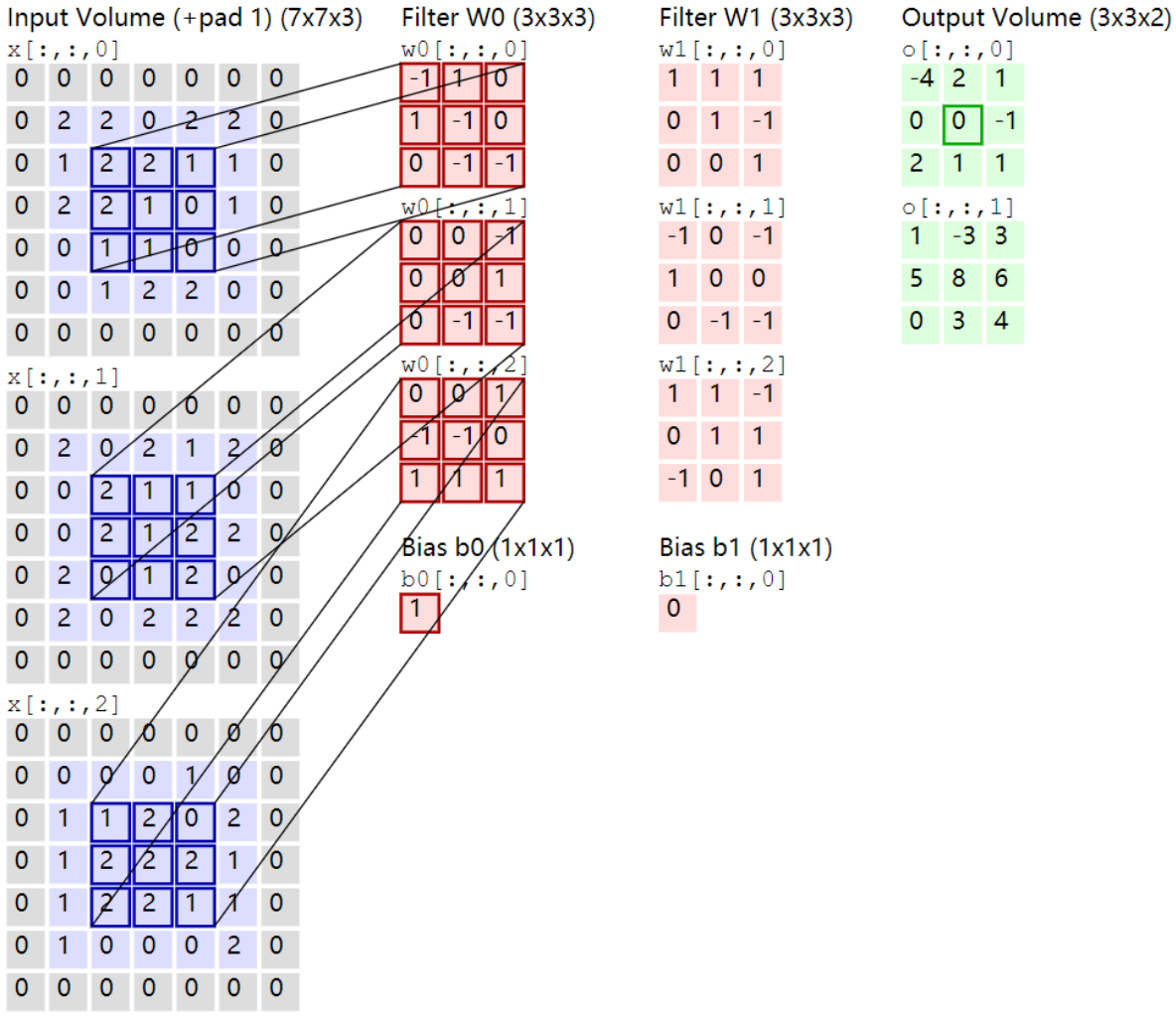

图 2.2 卷积神经网络中的卷积操作

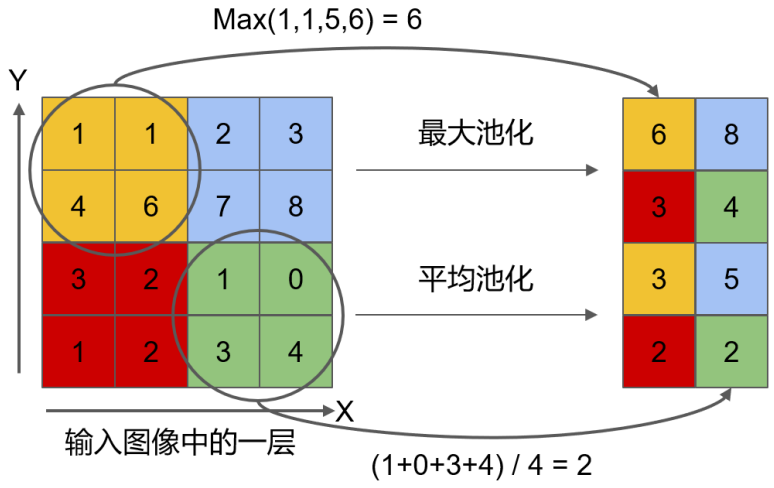

图 2.3 卷积神经网络中的池化操作

2、池化层

池化层是卷积神经网络中对输入数据进行压缩与提纯的层。池化层压缩输入特征使其尺度减小，降低计算复杂度，减少训练参数与计算时间的同时，对特征进行提纯，保留主要特征，防止过拟合。平均池化层和最大池化层是最常见的两种池化层，如图 2.3 所示。

3、激活层

激活层是用来增加卷积神经网络中非线性的层，一般在每个卷积层之后都会接激活层。卷积是一种线性操作，连续堆叠多个卷积层本质上还是线性操作，因此无法对非线性的结果进行拟合。激活层采用非线性函数对特征图中的每个元素进行非线性激活操作，从而提升了卷积神经网络的拟合能力。此外，使用激活层还可以提升卷积神经网络的收敛速度，加速卷积神经网络训练。图 2.4 为激活函数的图像，下面介绍几种常见的激活函数。

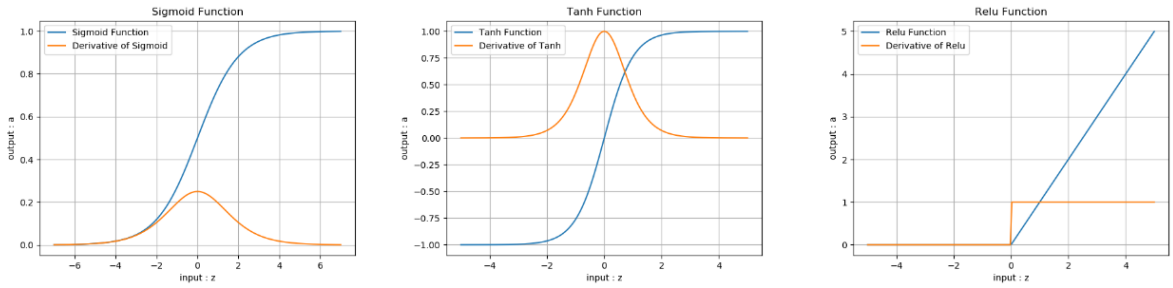

图 2.4 常用激活函数的图像

下面列举常用的激活函数及其倒数，Sigmoid 函数：

$$f(x) = \frac{1}{1+e^{-z}} = a \tag{2.1}$$

$$f'(x) = a(1-a) \tag{2.2}$$

Tanh 函数为：

$$f(x) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} = a \tag{2.3}$$

$$f'(x) = (1+a)(1-a) \tag{2.4}$$

ReLu 函数为：

$$f(x) = max(0,x) \tag{2.5}$$

$$f'(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{2.6}$$

4、全连接层

最终的分类操作由全连接层完成，全连接层的每个输出都与所有的输入相关联。假设全连接层的输入参数为a个，输出参数为b个，可以使用一个$(a，b)$维的权重矩阵和一个$(1，b)$维的偏移矩阵计算全连接层的结果。如图 2.5 所示。

其中$W_{x,y}$表示权重矩阵$(x，y)$处元素值，b_x为偏移矩阵x处元素值。

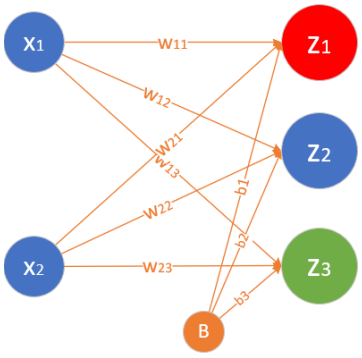

图 2.5 全连接层

2.1.2 经典卷积神经网络介绍

经典卷积神经网络主要包括 AlexNet[19]、VGGNet[20]、GoogleNet[21]和 Resnet[22]等。

AlexNet 引入了 ReLu 损失函数，引入了防止过拟合的方法 Dropout 和数据增强，同时引入局部归一化方法和多 GPU 多卡训练。但是仍然存在一些缺点，如前置卷积核大，参数多难以训练。

VGGNet 采用了3×3大小的卷积核和2×2大小的最大池化层，代替了5×5和7×7大小的卷积核。这一做法证明了堆叠小尺寸卷积层代替大尺寸卷积层可以使得感受野大小不变，而且多个小尺寸卷积核比一个大尺寸卷积核有更少的参数与更多的激活函数，因此拥有更强的拟合能力。同时该网络证明了增加深度在一定程度上可以提高网络性能。

GoogleNet 中使用的 Inception 模块使用3×3和5×5等不同尺寸的卷积核同时进行卷积运算，并使用1×1的卷积核对结果进行降维处理以减少计算成本。该模块类似于一个基础的神经元组件，内部有多个不同尺寸卷积核，使网络对于不同尺度的图像有更好的处理能力与适应性。

理论上网络的性能应该与堆叠的层数呈线性关系，层数越多网络的性能越好。但随着网络的加深，出现了错误率上升的“退化”问题。这不是由于过拟合产生的，而是随网络深度的加深，反向传播变得更加困难，出现了梯度爆炸和梯度消失的现象。

残差网络(ResNet)，引入了能够跳过一层或多层的捷径连接，因为捷径的存在使得网络的性能至少性能不会差于浅层网络，如图 2.6。该方法解决了堆叠卷积层带来的退化问题，使得卷积神经网络的层数大大加深达到上百层，并且大幅度提升了卷积神经网络的性能。

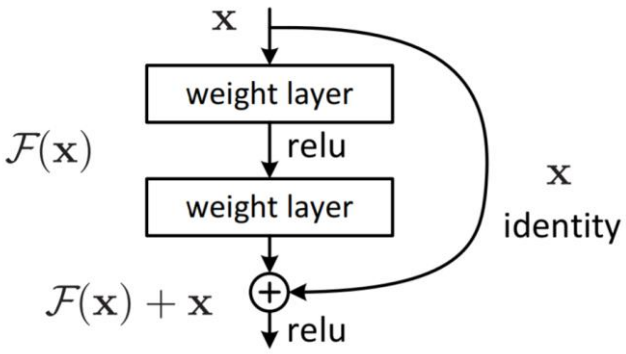

图 2.6 残差网络中的残差块

2.2 循环神经网络

循环神经网络是前馈神经网络的一种扩展，用于学习动态序列数据的非线性特征的。本节主要介绍传统的循环神经网络模型以及长短期记忆网络模型。

2.2.1 简介

循环神经网络(RNN)的特点是网络中的连接沿着时序关系形成一个有向图，并且可以使用内部缓存的隐藏状态处理输入序列，这使得它常被用于处理输入序列长度可变，且输入序列各元素之前存在较强的依赖关系的任务中，如语音辨识、文字识别、自然语言处理等任务。使用循环神经网络能够很好的对输入序列中的上下文关系建模，通过循环神经网络中的隐藏层能有效学习输入数据序列的特征表征。

2.2.2 长短期记忆网络

长短期记忆网络(LSTM)[3]也是一种用于处理序列输入的网络，它是循环神经网络的一种变形，因为结构简单高效、性能好等特点而被大量使用。传统的循环神经网络对近期输入敏感，在更新长期记忆时会出现梯度消失和爆炸。现在增加一个细胞状态(cell state)，如图 2.7 所示，可以对长期记忆进行有效地保留。长短期记忆网络设计了门控(gate)结构，控制信息的保留和丢弃。其中，遗忘门控制上一时刻的隐藏状态有多少保留到当前时刻；输入门控制当前输入有多少保存到隐藏层状态；输出门控制当前隐藏状态有多少输出到当前时刻的输出。长短期记忆网络的推导以及解决梯度爆炸与消失的推导见附

录 A。

长短期记忆网络的创新之处在于通过三个门来控制传输状态，通过隐藏层学习输入序列的历史信息，并且在学习的过程中通过遗忘门适当的遗忘不重要的历史信息，对于很多输入序列较长需要长期记忆的任务来说有很大提升。在实际使用长短期记忆网络的过程中，经常会堆叠多个长短期记忆网络来提高网络的深度，进一步提升网络的学习能力。在有的任务中，还会使用双向长短期记忆网络，对输入数据进行反向处理作为网络输入，使得长短期记忆网络在处理序列信息时不仅能参考到上文的信息，同时也能参考到下文的信息，从而进一步提升泛化能力。

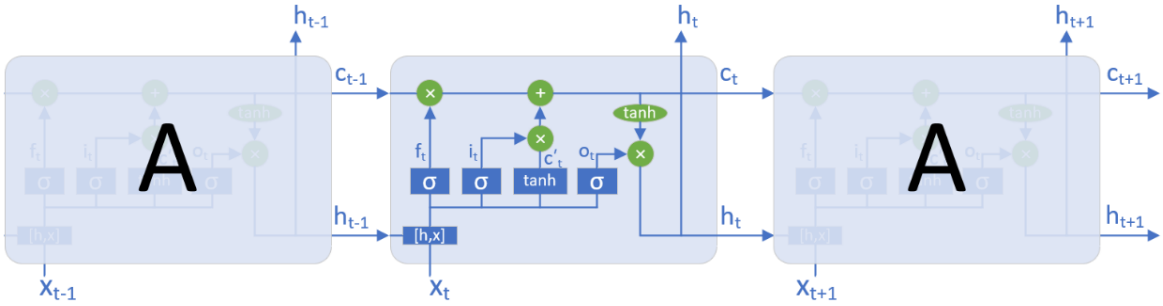

图 2.7 长短期记忆网络结构

2.3 损失函数

用于计算神经网络的前向计算结果与真实标签值之间的差距的函数称为损失函数，它可以使得参数的优化可以向着损失减小的方向进行。下面将介绍几种本文用到的损失函数。

2.3.1 交叉熵损失函数

交叉熵(Cross Entropy)的概念来自于是 Shannon 信息论中，用于度量两个概率分布之间的差异。

$$H(p,q) = \sum_i p_i \cdot ln\frac{1}{q_i} = -\sum_i p_i \cdot lnq_i \tag{2.7}$$

交叉熵可在神经网络中作为损失函数，对于单一分类问题，样本的真实概率是一个零一向量，所属分类的维度值是 1，其余地方都是 0。在实际使用时，常常先使用 Softmax 函数对模型预测值进行概率归一化操作。以单一分类问题为例，交叉熵函数可以用以下公式表示：

$$loss(x, class) = -\log\left(\frac{e^{x_{class}}}{\sum_j e^{x_j}}\right) \tag{2.8}$$

其中，x代表神经网络的输出向量，长度与总类别数相同；$class$代表输入的标签，表明当前样本属于哪个类；$\frac{e^{x_{class}}}{\sum_j e^{x_j}}$是 Softmax 函数，对$x$进行概率归一化操作，并取出属于$class$类的概率。在实际使用的过程中，还可以采用对不同的类别加不同的权重系数，挖掘难例等方法以解决类别之间不均衡的情况。

2.3.2 连接时序分类损失函数

连接时序分类损失函数(CTC)[15]是一种专门针对序列预测问题所提出的损失函数，在语音识别和文字识别中有着广泛的应用。连接时序网络分类函数实际上是一种神经网络和损失函数的结合，主要用来处理循环神经网络的序列输出。在语音识别和文字识别这类序列识别问题中，循环神经网络输出的序列长度往往与真实值序列的长度不相同，这导致一般的交叉熵损失函数、平均平方误差损失函数等损失函数不好对长度不同的两个输入进行计算。连接时序分类损失函数能够很好的解决输出与真实值序列长度不一样的问题，因此被广泛的应用在许多序列识别问题中。

连接时序分类定义了如何从网络定长输出到不定长序列的转化。假设整个网络的输入向量序列x的长度是T，y是网络的输出序列向量，长度也为T；π是一个输出序列，则输出序列关于输入的条件概率可以用以下公式表示：

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L^T \tag{2.9}$$

其中，$p(\pi|x)$表示输出序列关于输入序列的条件概率；$y_{\pi_t}^t$表示输出特征向量序列的第t个向量被分类成π_t的概率；L表示序列输出每个位置的空间，例如在文字识别问题中就是所有文字的集合；L^T表示长度为T的，每个字符都在T中的任意序列。在这个公式中，序列π的长度与输入序列x和输出序列y的长度一致且一一对应，其出现的概率等于y每个位置出现指定字符的概率之积。

在此基础上，连接时序分类提出了一种多对一的映射函数，通过引入空格符特殊符号，将多个不同的序列π映射成同样的序列。具体的做法是首先在字符集L中加入一种特殊字符空白符，然后将序列π中连续相同的字符映射成一个字符，如图 2.8 所示。

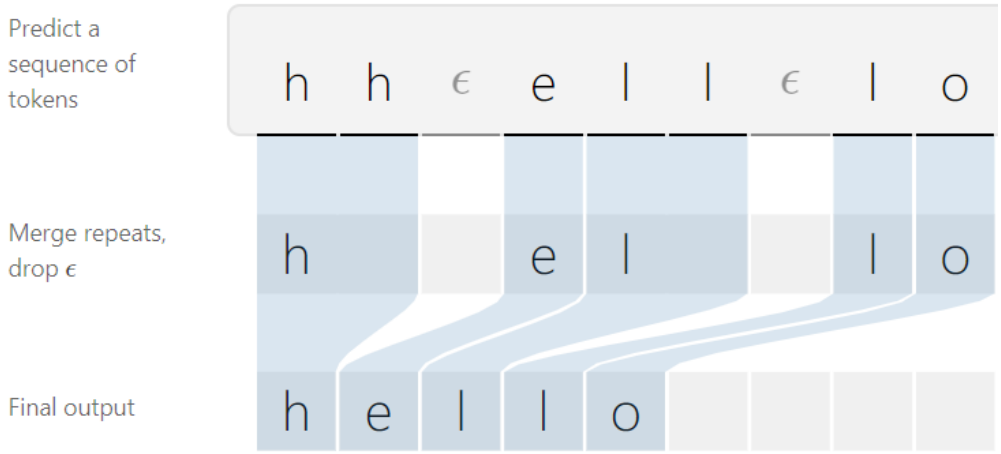

图 2.8 CTC 映射函数

假设映射关系用B表示，则$B(a_ab_) = B(_aa__abb) = aab$，其中_代表空白符。映射时，先将连续相同字符合并，然后再去除空白符。通过引入空白符，就可以表示原本存在连续字符的映射。此时，针对输出序列长度小于输入序列长度的情况下的条件概率，就可以通过映射函数表示，公式如下：

$$p(l|x) = \sum_{\pi \in B^{-1}(l)} p(x) \tag{2.10}$$

其中，$p(l|x)$表示长度小于输入长度的条件概率；$p(l|x)$由公式(2.20)计算得来；$\pi \in B^{-1}(l)$是一个集合，由全部长度与输入长度相同且经过映射可以变成l的序列组成。简单来说，通过引入空白符和映射函数的操作，可以将定长的输出向量转换成一个不定长的序列，并求得关于输入的条件概率。有了定长输入对应不定长输出序列的条件概率就可以计算对应真实值序列的概率，并针对真实值序列的条件概率计算损失值，在此基础上构造损失函数。

在训练的过程中，$\pi \in B^{-1}(l)$这个集合的大小往往非常大，呈指数量级增长，通过枚举计算概率总和算法复杂度太高，难以使用。为了解决这个问题，连接时序分类算是函数提出了一种基于动态规划的方法计算$p(l|x)$，对查找路径进行剪枝加速处理。算法的核心思想是根据时序关系提前合并一些相同的路径，只保留那些合法的路径，并用中间变量记录存储。经过动态规划算法优化的算法，在计算$p(l|x)$时可以达到$O(T^2)$的时间复杂度，非常高效。

本文的文字识别算法中采用了连接时序分类损失函数，将连接时序分类损失函数接在循环神经网络输出之后进行损失函数计算。

2.3.3 dice-coefficient 损失函数

Dice 系数通常用于度量集合的相似度，以及计算两个样本的相似度：

$$dice = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2.11}$$

$|X \cap Y|$是X和Y之间的交集，$|X|$和$|Y|$分别表示X和Y的元素个数。

Dice Loss[26]可以写作：

$$L_{dice} = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \tag{2.12}$$

2.4 文字检测相关技术

文字检测模型可以划分为骨干网络，上采样分割头和分割输出三个阶段，骨干网络使用的 Resnet50 模型已在 2.1 节有过详细的介绍，本节主要介绍两种现有的上采样分割头和三种现有主流分割输出模型。

2.4.1 特征金字塔网络上采样分割头

特征金字塔网络上采样分割头[28]首先对高层结果进行扩大二倍的上采样，与比其低一级的结果进行合并，通过卷积层后上采样到$[l/4, w/4]$大小。最终将四个同一尺度的中间结果$[b, inner, l/4, w/4]$拼接成$[b, inner \times 4, l/4, w/4]$，然后通过卷积层生成特征图$[b, inner, l/4, w/4]$，具体过程如图 2.9 所示。

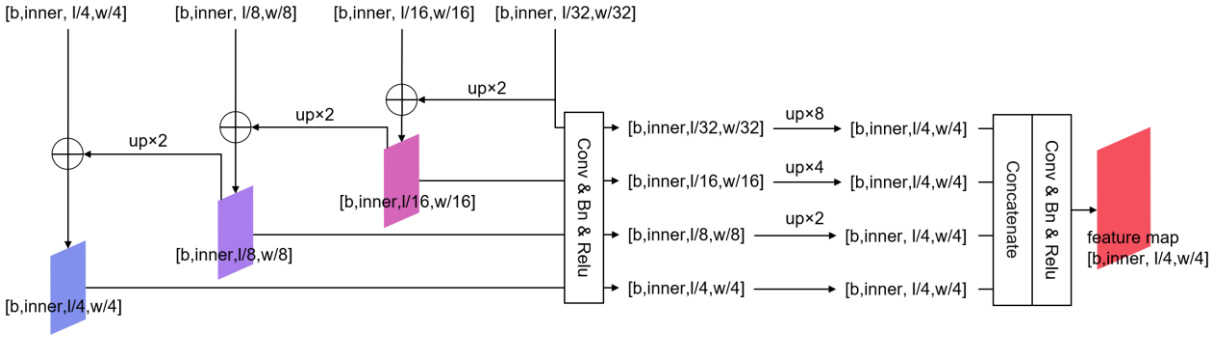

图 2.9 特征金字塔网络上采样分割头

2.4.2 特征金字塔增强与融合分割头

与上述使用传统的卷积神经网络作为分割头的模型不同，特征金字塔增强与融合分割头采用了轻量化的卷积操作。具体而言，在一个特征金字塔增强模块中，对于输入先进行一系列上采样操作，然后进行一系列下采样操作，输出结果与输入的尺度相同。其中卷积部分使用了 Depth-wise 卷积和 Point 卷积[29]，用于减少参数数量和获取更大的感

受野，具体过程如图 2.10 所示。

与传统的特征金字塔网络相比，特征金字塔增强模块的优点是可级联以及计算成本低。为了增强网络的特征提取能力，可以使用多个特征金字塔增强模块，将每个模块的输出结果输入给特征融合模块，生成特征图。

特征融合模块对于特征金字塔增强模块的结果进行逐元素相加，然后进行上采样生成特征图。

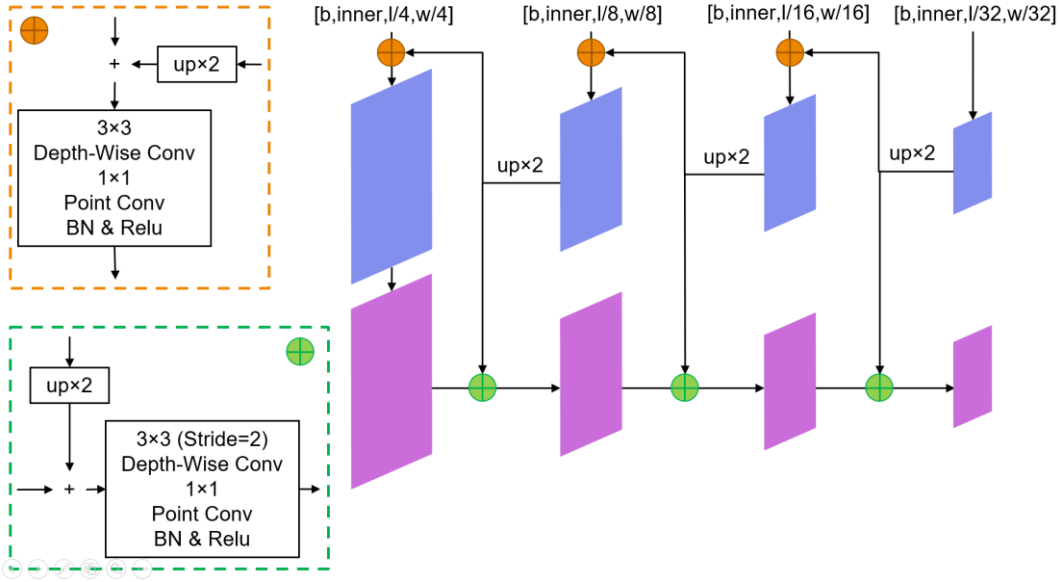

图 2.10 特征金字塔增强模块

2.4.3 渐进规模扩展网络

渐进规模扩展网络(PSENet)[10]的特点是为每个文本实例生成一系列嵌套的不同比例的核，核的作用是区分相邻或重叠的文字区域，并采用基于广度优先搜索(BFS)的渐进尺度扩展算法扩展较小的核到其紧邻的较大核逐渐得到完整的文本实例，比例不同的核与原始文本实例的形状几何相似。最小尺度的核边界彼此远离因此不容易混淆，恢复后的完整的文本实例可以覆盖文本实例的完整区域。

图 2.11 显示了渐进规模扩展网络的结构图，对于骨干网络和上采样分割头生成的特征图，将其投影到n个分支中以产生n个分割结果$\{S_1, S_2, \ldots, S_n\}$。每一个$S_i$都是某一比例下所有文本实例的一个分割覆盖。

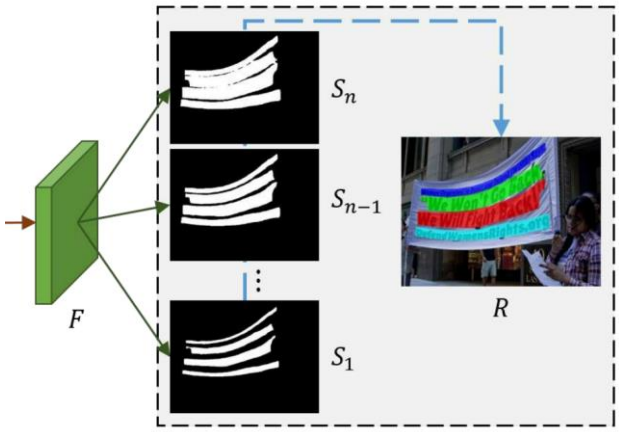

图 2.11 渐进规模扩展网络[10]

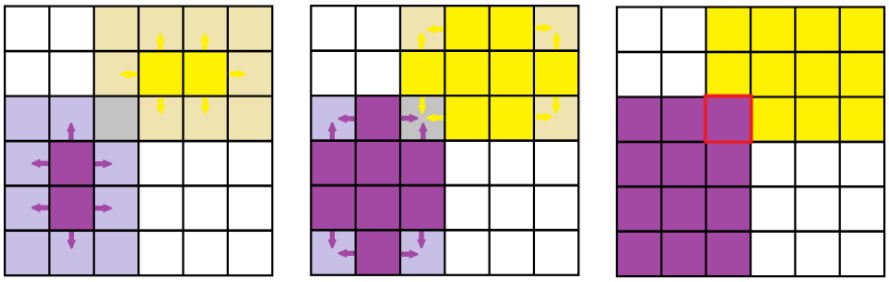

图 2.12 渐进尺度扩展算法

渐进尺度扩展算法的示例如图 2.12，在该示例中深色区域表示S_i，其周围的浅色区域表示S_{i+1}，灰色区域表示两个核S_{i+1}的重合部分。对红色框圈出的冲突像素，采用先到先得的处理原则。

图 2.13 标签生成示例

对于产生的不同大小的分割结果，训练时需要不同大小的标签与之对应。生成标签的方法是使用原始标注的文本框进行不同比例的缩小。图 2.13 在为标签生成的示例，其中(a)为原始文本实例区域，它对应于最大的分割标签。

对于其他大小的标签，使用 Vatti 裁剪算法[30]获得。即对于一个缩放比例r_i，可计算

一个收缩距离d_i：

$$d_i = \frac{Area(p_n) \times (1 - r_i^2)}{Perimeter(p_n)} \tag{2.13}$$

其中$Area$为标注框的面积，$Perimeter$为标注框的周长，r_i的计算方法为：

$$r_i = 1 - \frac{(1-m) \times (n-i)}{n-1} \tag{2.14}$$

其中m为最小的核的缩放比例，取值范围为(0,1]，对于两个超参数m和n，可确定一系列收缩率$r_1, r_2 \ldots\ldots r_n$，它们从$m$增加到 1，本文中$m$取 0.4，$n$取 7。

最后需要计算预测值与标签的差距以进行反向传播，由于存在收缩文字区域，因此需要分别计算完整文字区域的损失L_c以及收缩文字区域的损失L_s。根据街景图像的特点可知，文本区域通常只占整个图像的极小一部分，所以如果使用二进制交叉熵[13]会使得负例的数量大大增加。受到[23]的启发，本文使用 Dice-coefficient 损失函数：

$$D(S_i, G_i) = \frac{2 \sum_{x,y}(S_{i,x,y} \times G_{i,x,y})}{\sum_{x,y} S_{i,x,y}^2 + \sum_{x,y} G_{i,x,y}^2} \tag{2.15}$$

其中$S_{i,x,y}$表示输出S_i在(x,y)处的像素值，$G_{i,x,y}$表示标签G_i在(x,y)处的像素值。

在训练时对L_c使用在线难例挖掘(OHEM)[24]，根据输入样本的损失进行筛选，筛选出难例生成 0/1 掩码M。L_c的计算方法为：

$$L_c = 1 - D(S_n \cdot M, G_n \cdot M) \tag{2.16}$$

在计算L_s时，可忽略完整文本区域以外的其他区域，因此同样生成一个掩码 W：

$$W = \begin{cases} 1, S_{n,x,y}\text{为文字区域} \\ 0, \text{其他} \end{cases} \tag{2.17}$$

L_s的计算方法为：

$$L_s = 1 - \frac{\sum_{i=1}^{n-1} D(S_i \cdot W, G_i \cdot W)}{n-1} \tag{2.18}$$

总损失函数定义为：

$$L = \lambda L_c + (1-\lambda) L_s \tag{2.19}$$

λ平衡了L_c和L_s之间的重要性，本文中λ取 0.7。

2.4.4 像素聚合网络

像素聚合网络(PAN)[11]为每个文字区域生成一个完整文本区域预测(图 2.14 Text Region)，一个核(图 2.14 Kernel)和一个四维的相似向量(图 2.14 Similarity Vector)。核可

以很好地区分文本实例，但它不是完整的文本实例。因此使用一个可学习的像素聚合算法来重建完整的文本实例，引导文本像素聚类到正确的核。

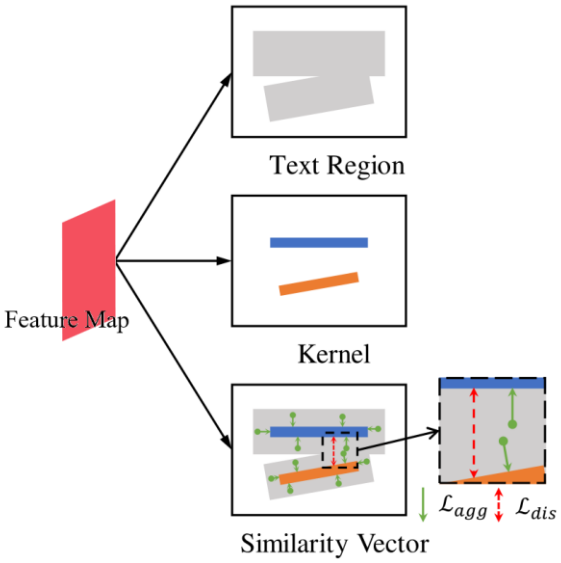

图 2.14 像素聚合网络分割输出结果[11]

对于相似向量，希望同一文本实例的文本像素和核之间的距离应该很小，使用L_{agg}损失来实现这个规则：

$$L_{agg} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|T_i|}\sum_{p\in T_i}\ln\left(D(p,K_i)+1\right) \tag{2.20}$$

$$D(p,K_i) = \max\left(\|F(p)-G(K_i)\|-\delta_{agg},\ 0\right)^2 \tag{2.21}$$

$$G(K_i) = \sum_{q\in K_i}\frac{\mathrm{F(q)}}{|K_i|} \tag{2.22}$$

其中N是文本实例的数量，T_i是第i个文本实例，$D(p,K_i)$定义了p和K_i之间的距离，δ_{agg}是一个用于过滤简单样本的超参数，在本文中设为 0.5。$F(\cdot)$是像素p的相似向量(四维)，$G(\cdot)$用于计算核的相似向量。

此外，不同文本实例的内核应该保持足够的距离，以保持聚类中心的区分度，使用L_{dis}损失来实现这个规则：

$$L_{dis} = \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{\substack{i=1\\ j\neq i}}^{N}\ln\left(\mathrm{D}\left(K_i,K_j\right)+1\right) \tag{2.23}$$

$$D(K_i, K_j) = \max\left(\delta_{dis} - \left\|G(K_i) - G(K_j)\right\|, 0\right)^2 \tag{2.24}$$

δ_{dis}定义了两个核之间最大距离，在本文中设为 3。

对于完整文本区域和核，采用与渐进规模扩展网络类似的损失函数L_{tex}和L_{ker}：

$$L_{tex} = 1 - \frac{2\sum_i P_{tex}(i) G_{tex}(i)}{\sum_i P_{tex}(i)^2 + \sum_i G_{tex}(i)^2} \tag{2.25}$$

$$L_{ker} = 1 - \frac{2\sum_i P_{ker}(i) G_{ker}(i)}{\sum_i P_{ker}(i)^2 + \sum_i G_{ker}(i)^2} \tag{2.26}$$

其中$P_{tex}(i)$表示完整文本区域的分割结果，$G_{tex}(i)$表示完整文本区域的标签值，同样的，$P_{ker}(i)$表示核的分割结果，$G_{ker}(i)$表示核的标签值。核的生成方法与渐进规模扩展网络相同。

总损失函数定义为：

$$L = L_{tex} + \alpha L_{ker} + \beta(L_{agg} + L_{dis}) \tag{2.27}$$

α和β平衡了四个损失函数之间的重要性，本文中α取 0.5，β取 0.25。

2.4.5 可微分二值化网络

可微分二值化(DB)[12]网络是一个可以自适应地二值化阈值的网络，它不仅简化了后处理的操作，而且提高了文字检测的性能。可微分二值化模型将阈值图与概率图联合优化生成二值图，并对文字区域边界进行加强，从而完全区分前景和背景的像素。其模型的整体结构如图 2.15 所示，特征图预测概率图(P)和阈值图(T)，进而预测出近似二值图($\hat{B}$)。

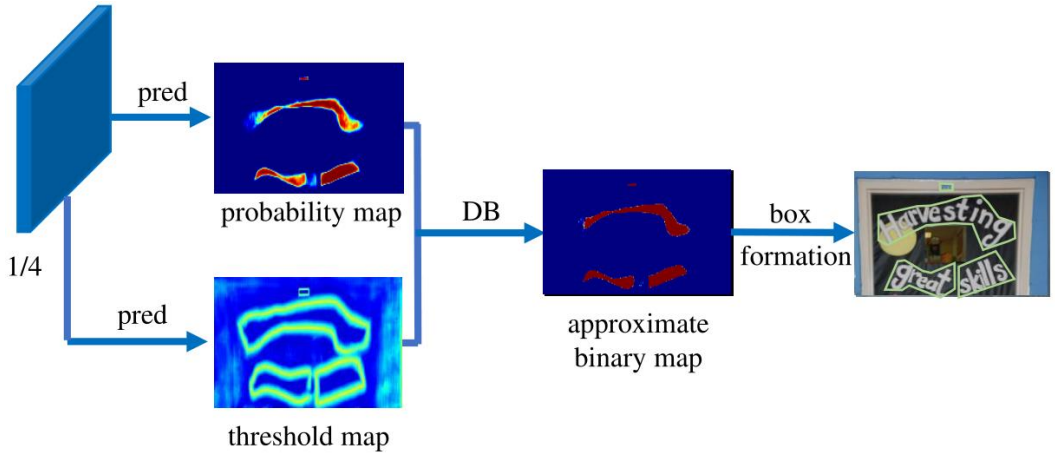

图 2.15 可微分二值化模型网络结构[12]

传统的二值化方法是对于生成的阈值图P，生成一个二值图B，其中 1 代表文字区域，0 代表非文字区域。

$$B_{i,j} = \begin{cases} 1, & P_{i,j} \geq t \\ 0, & \text{其他} \end{cases} \tag{2.28}$$

其中t是预先定义的固定阈值。

然而标准的二值化是不可微的，在训练阶段不能进行优化，因此本文使用近似的二值化函数：

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}} \tag{2.29}$$

其中，T是从网络中学习的自适应阈值图，k表示放大因子，通常设为 50。

在反向传播中，以交叉熵损失函数为例，设可微分二值化方程为$f(x) = \frac{1}{1+e^{-kx}}$，其中$x = P_{i,j} - T_{i,j}$。用$l_+$表示正例，$l_-$表示负例：

$$l_+ = -\log \frac{1}{1 + e^{-kx}} \tag{2.30}$$

$$l_- = -\log(1 - \frac{1}{1 + e^{-kx}}) \tag{2.31}$$

损失为：

$$\frac{\partial l_+}{\partial x} = -kf(x)e^{-kx} \tag{2.32}$$

$$\frac{\partial l_-}{\partial x} = kf(x) \tag{2.33}$$

因此，可微分二值化网络可在反向传播中通过梯度下降对概率图和阈值图进行修改。

阈值图从外观上类似于[25]中的文本边界图，然而阈值图的动机和用途不同于文本边界图，它突出显示文本边界区域。

概率图和二值图使用相同的标签，其生成方法受到渐进规模扩展网络的启发，正例区域为将标注框G收缩为G_s，其偏移D的计算方法为：

$$D = \frac{A(1 - r^2)}{L} \tag{2.34}$$

其中A为标注框的面积，L为标注框的周长，r为缩放比例，本文中取 0.4。

使用同样的方法可以为阈值图生成标签，首先， 使用同样的偏移将标注框G扩张为G_d，G_s和G_d之间的区域被视为文本区域的边界G_b。将所求的G_b框内所有像素到G框距离，除以偏移量D进行归一化，用 1 减去归一化结果，再将[0, 1]区间映射到[0.3, 0.7]，生成

阈值图的标签。生成结果如图 2.16 所示，其中最暗处值为 0.3，最亮处值为 0.7。

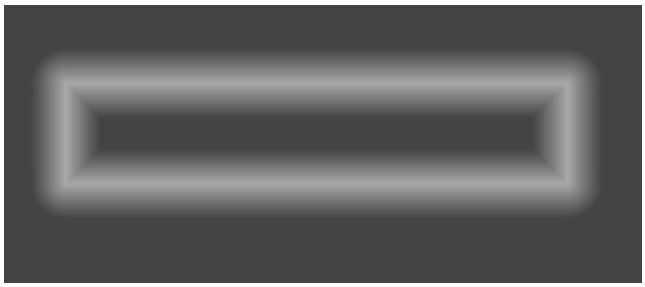

图 2.16 阈值图的标签

对于概率图和二值图，使用交叉熵损失函数，同时使用与渐进规模扩展网络相同的难例挖掘策略获取难负例。

$$L_p = L_b = \sum_{i \in S_l} y_i \log x_i + (1 - y_i) \log(1 - x_i) \tag{2.35}$$

S_l是使用难负例，正负样例比为 1:3 的采样集合。

对于阈值图的损失，计算阈值图和标签的扩张文本区域G_d内像素的 L1 距离：

$$L_t = \sum_{i \in G_d} |y_i^* - x_i^*| \tag{2.36}$$

总损失函数定义为：

$$L = L_p + \alpha \times L_b + \beta \times L_t \tag{2.37}$$

α和β用于平衡不同损失之间的重要程度，本文中α取 1，β取 10。

在预测阶段，可以使用概率图或二值图来生成文字框，为了提高效率，可以使用概率图，此样就可以移除阈值图分支，提高计算速度。预测时首先使用固定阈值 0.2 来二值化概率图，然后从二值图中获取连通的缩小的文字区域，最后对收缩的文字区域进行反向扩张，偏移D'的计算方法为：

$$D = \frac{A' \times r'}{L'} \tag{2.38}$$

其中A'和L'分别表示为收缩文字区域的面积和周长，r'为扩张率，本文取 1.5。

2.5 文字识别相关技术

本节将介绍现有主流文字识别模型：卷积循环神经网络(CRNN)[14]，该网络由卷积层，循环层和转录层三部分组成，如图 2.17 所示。

其中卷积层用于从输入图像中提取序列特征，所有图像在缩放到相同高度输入卷积层。卷积层的输出转化为序列化的特征串作为循环层的输入，即将卷积层的输出划分为

宽度为 1 的列，从左到右按顺序将每一列转化为一个特征。具体的，模型采用$VGG16$网络[20]，因为文字区域通常为水平长条状，长度通常为高度的数倍，因此网络中靠后的池化层采用2×1的大小，实验证明该卷积层能更好地来提取文字特征。具体的特征提取示例如图 2.18 所示。

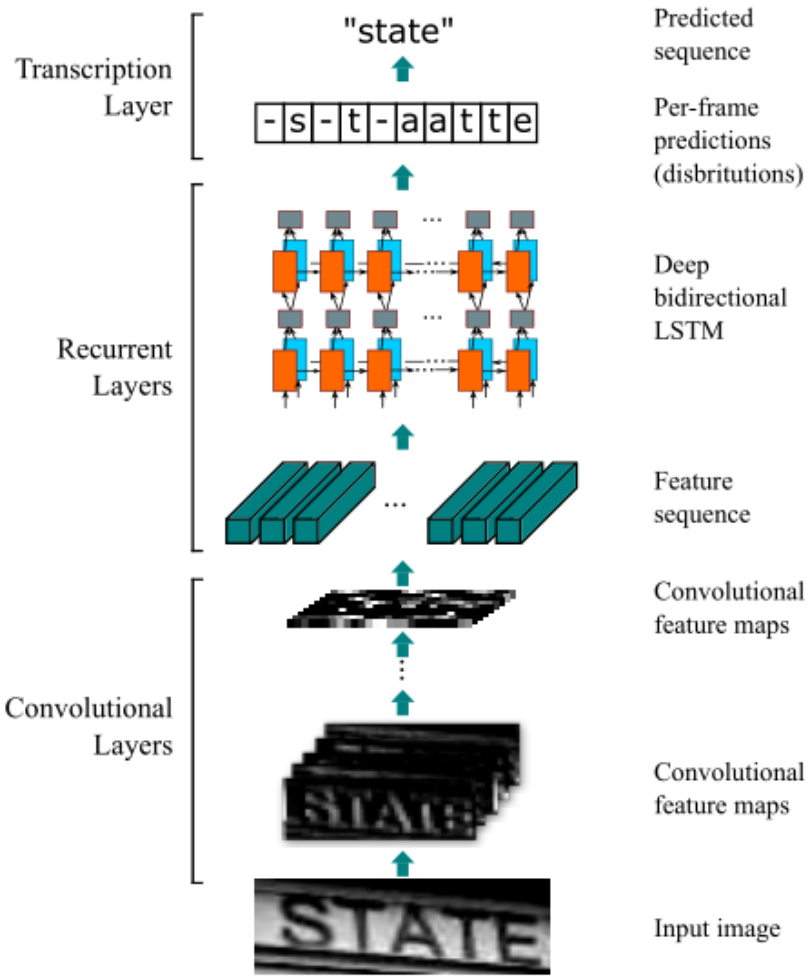

图 2.17 卷积循环神经网络示意图[20]

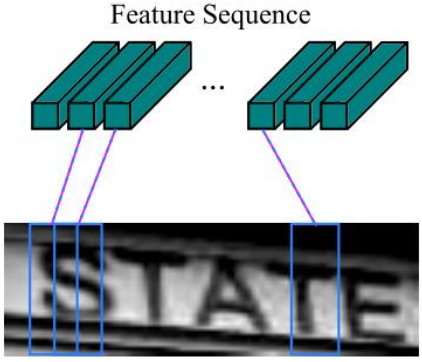

图 2.18 卷积层的序列特征提取示意图[20]

循环层使用深度双向长短期记忆网络[3]，因为它有很强的捕捉序列中上下文信息的能力且能操作任意长度的序列。双向长短期记忆网络不仅可以考虑到序列的上文信息，也可以考虑到序列的下文信息。而且堆叠多个双向长短期记忆网络形成深度双向长短期记忆网络，可以增强信息的提取能力。在循环层底部，特征序列转换成特征图，使误差可以反向传播到卷积层。转录层使用连接时序损失分类函数，其具体推导过程见附录B。

2.6 本章总结

本章简要介绍了街景图像中英文文字检测与识别模型的相关研究工作。主要包括卷积神经网络和循环神经网络的构成、发展演进历程以及具体实现细节，以及用于处理分类问题的交叉熵损失函数、用于处理输出与标签值不等长的连接时序分类损失函数和用于评估集合相似度的 dice-coefficient 损失函数。最后对街景图像中英文文字检测与识别模型的流程进行了介绍。将基于分割的文字检测模型分为骨干网络、上采样分割头和分割输出三个部分分别进行介绍，包括算法流程中的关键模块的实现方法和细节。文字识别模型使用了 CRNN 模型，重点介绍了该模型的三个层次以及连接时序分类损失的实现细节以及损失的计算。

3 文字检测与识别模型的实现与改进

光学字符识别是信息抽取领域的重要研究内容之一，本章分析了文字检测和文字识别的模型分析以及具体实现的细节。在上一章介绍的文字检测和识别模型的基础上进行改进和创新，以降低模型的参数数量，提高模型的准确率。基于上述研究成果，构建了一个面向多场景中英文的在线图像检测与识别的系统，并详细说明了整体系统的结构和功能设计，以及各个模块的设计与实现。

3.1 文字检测模型实现

光学字符识别的第一个关键步骤就是文字检测，该过程通过提取和分析输入图像的特征信息，输出图像中文字区域的位置坐标信息。文字检测部分的准确率会对整个系统的准确率产生至关重要的影响。

文字检测模型主要可分为数据加载模块，模型训练模块，以及后处理等模块，其中模型部分又可以划分为骨干网络，上采样分割头和分割输出三个阶段。具体介绍如下：

3.1.1 数据加载模块

数据加载部分包括将输入图像文件整理为索引文件方便加载与训练以及对输入图像进行旋转，翻转等前处理来增强模型的泛化能力，防止过拟合。

生成索引文件时，首先将训练集和测试集的图像和标签(真实值)分别放在两个文件夹下，然后建立索引文件用于确定上述两个文件夹中内容的对应关系，索引文件以键值对的方式存储，以输入图像为键，标签文件为值。最后还实现了划分数据集的功能，能根据需要将数据集的训练集划分出验证集。

对数据进行前处理时，可以对图像大小进行调整，即构建 scale 列表，如 scale＝[0.8, 0.9, 1.0, 1.1, 1.2]，在列表中随机选择一个数值对图像进行缩放；可以对图像进行旋转，即在[-10°, 10°]之间随机选择一个角度对图像进行旋转，旋转后图像角落处会出现黑色区域；可以对图像进行翻转，即有 50%的概率对图像进行翻转；可以对图像进行随机裁剪，即在保留有文字的前提下对图像进行裁剪。上述变换在执行的同时需要对标签进行相同的变换。

3.1.2 骨干网络

骨干网络用于提取图像中的特征信息，因此使用特征金字塔网络，对于一批输入图

像，通过下采样处理生成不同尺度的特征图，以此来获取输入图像的不同尺度的特征信息。经过实验研究以及测试发现，较优的骨干网络有 Resnet[22]，Mobilenetv3[27]等。其中 Resnet 引入了能够跳过一层或多层的捷径连接，可以大大加深网络的深度，也拥有较多的参数，有较强的图像特征提取能力，而 Mobilenetv3 在模型相对轻量化的同时保持了不错的性能，适合在移动端等对于模型大小有较高要求的环境里使用。本文的后续研究主要使用的是 Resnet50 模型。

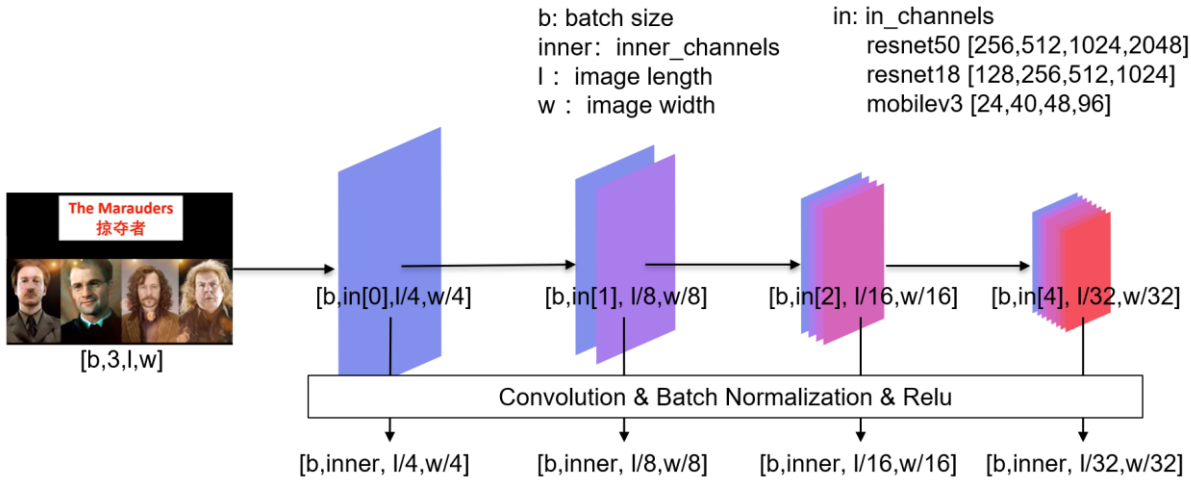

图 3.1 骨干网络特征金字塔结构图

骨干网络的结构图如图 3.1 所示，其中，b是批大小，l是输入图像的长度，w是输入图像的宽度，本文中输入图像的长宽均设置为 640。$inner$为中间层的通道数，对于 Resnet50 取 256，Resnet18 取 128，mobilenetv3 取 96。in表示一个长度为四的数组，对应为下采样过长中每一步对应结果的通道数。

3.1.3 上采样分割头

在下采样过程中，生成了不同尺度的中间结果，这些结果对预测文字区域都有作用。其中低层结果具有更高的分辨率，对小的文字区域更敏感；而高层结果具有更强的语义信息，对大的文字区域更敏感，所以需要通过将低级纹理特征和高级语义特征连接起来，融合在特征图中，这种融合有助于后续的预测工作。本条主要对 2.4 节介绍的两个分割头做出改进改进与创新，具体介绍如下：

1、改进特征金字塔网络上采样分割头

中间层数的多少一定程度上决定了模型处理信息的能力强弱，但是数量过多的中间层不仅会增加模型参数的数量，而且可能会造成过拟合的现象。图 2.9 所示的特征金字塔网络上采样分割头的中间结果的通道数均为$inner$，合并后通道数变为$inner \times 4$，这

样会使得卷积层参数量增加很多，改进后的特征金字塔网络上采样分割头中间结果的通道数均为$inner/4$，合并后通道数直接是$inner$，使得卷积层参数量显著减少，但仍能保留不错的性能，具体过程如图 3.2 所示。

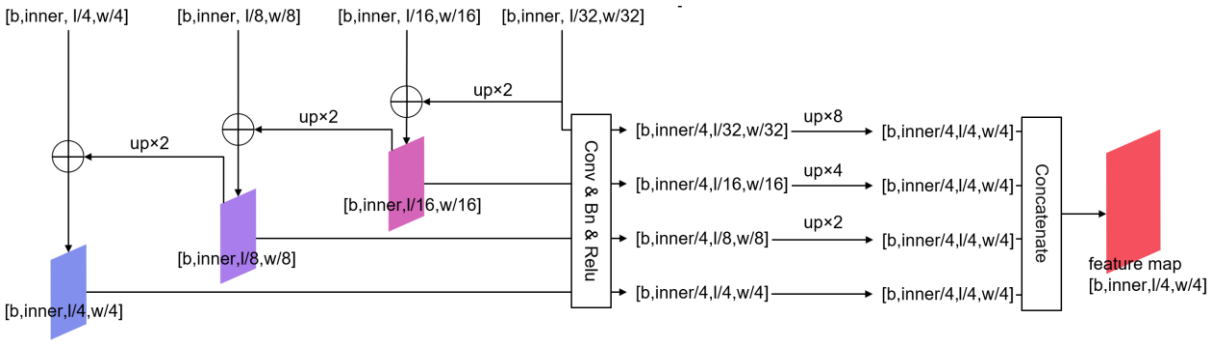

图 3.2 改进特征金字塔网络上采样分割头

2、特征融合模块的改进

2.4.2 条所示的特征金字塔增强与融合分割头的特征融合模块对于特征金字塔增强模块的结果进行逐元素相加，然后进行上采样生成特征图。该方法直接将不同的特征图进行相加，可能会造成特征的丢失。

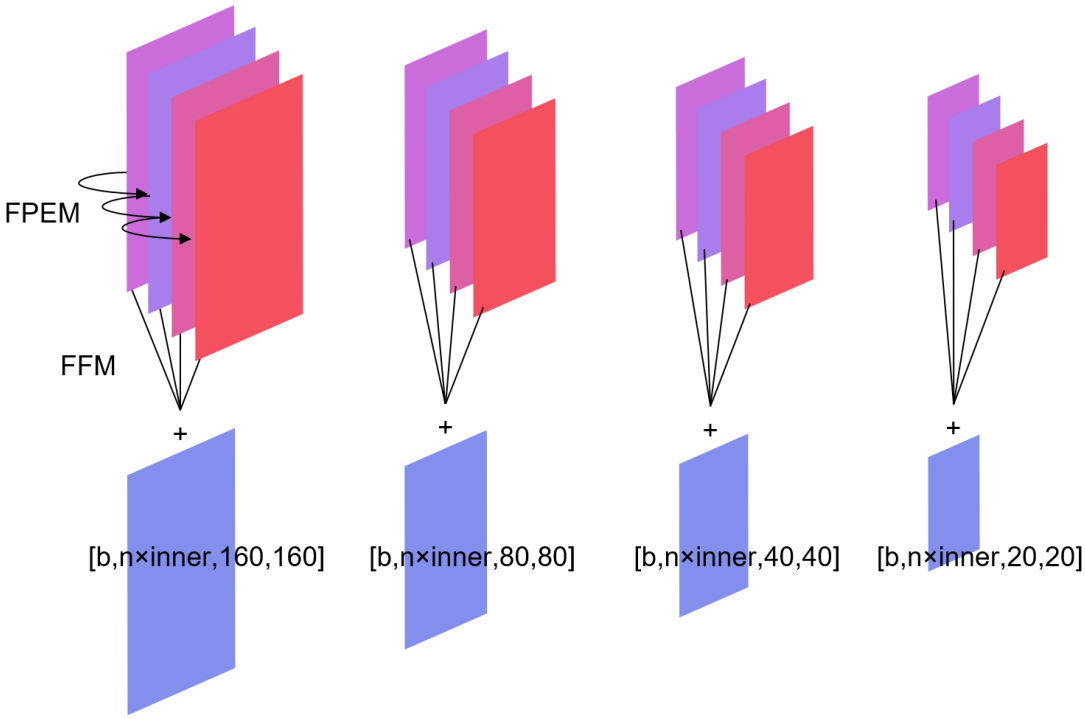

图 3.3 特征融合模块

改进的特征融合模块使用传统的特征拼接方法，然后通过卷积网络来降低通道数，生成最终的特征图，具体过程如图 3.3 所示。

3.1.4 分割输出

通过骨干网络和上采样分割头之后，模型获得了一个特征图，该特征图包含了丰富的语义信息，分割输出的功能就是处理特征图并通过二值化方法生成图像的文字区域和非文字区域。

对于一个大小为$(b, inner, l/4, w/4)$的特征图，在分割输出模块通过卷积操作和上采样操作生成大小为(b, n, l, w)的输出结果。对于渐进规模扩展网络，由于需要生成七个相互嵌套的核，因此n为 7，其中$(b, 1, l, w)$为完整的文字区域预测，$(b, 6, l, w)$为完整文字区域的一系列不同缩放比例的核。对于像素聚合网络，由于需要生成一个完整文本区域预测、一个核和一个四维的相似向量，因此n为 6，其中一个$(b, 1, l, w)$为完整的文字区域预测，另一个$(b, 1, l, w)$为核的预测，$(b, 4, l, w)$为四维相似向量。对于可微分二值化网络，由于需要生成概率图和阈值图，因此n为 2，其中一个$(b, 1, l, w)$为概率图的预测，另一个$(b, 1, l, w)$为二值图的预测。

3.1.5 后处理模块

后处理模块的作用是将分割输出的文本区域像素进行聚类分析，并生成相应的文字检测框。具体步骤为在分割输出的基础上寻找文字区域的连通区域，然后使用广度优先算法对于像素间欧氏距离小于特定阈值的像素进行聚类，最终生成文字检测框。

3.2 文字识别相关技术

文字识别模型对文字检测生成的区域内的所有文字进行识别，并将其转化成计算机能够理解和处理的文字信息。文字检测模型主要可分为数据加载模块，模型训练模块，以及后处理等模块，具体介绍如下：

3.2.1 数据加载模块

数据加在模块与 3.1.1 条中的介绍类似，加载用于训练的数据，其中训练数据分为两类：一类是只提供原始图像以及文字包围框的坐标和框中文字内容信息，需要自行裁剪用于训练。另一种是直接提供裁剪好的图像，可以直接进行训练。训练前需要对图像进行模糊，对比度及颜色调整等操作以提高模型的泛化能力，防止过拟合的发生。

3.2.2 模型训练模块

卷积循环神经网络主要由卷积层，循环层和转录层三部分构成。

卷积层用于从输入图像中提取序列特征，所有图像在缩放到相同高度输入卷积层。

卷积层的输出转化为序列化的特征串作为循环层的输入，即将卷积层的输出划分为宽度为 1 的列，从左到右按顺序将每一列转化为一个特征。

在本文的实现过程中，对 CRNN 实现了如下的改进与创新：将 VGG16 网络替换成了 Resnet34 网络，进一步提高了卷积层的信息提取能力。循环层使用深度双向长短期记忆网络[3]，因为它有很强的捕捉序列中上下文信息的能力且能操作任意长度的序列。还尝试在卷积层之前加入薄板样条插值模块进行训练，薄板样条插值用于改善输入图像中文本的形状，如图 3.4 所示。该模块通过定位网络寻找文字区域的边界，然后生成网格，在使用薄板样条插值修正输入文本。该模块不需要进行单独训练，可以在端到端的文字识别过程中进行联合优化。实验证明使用该方法准确性有一定的提高。

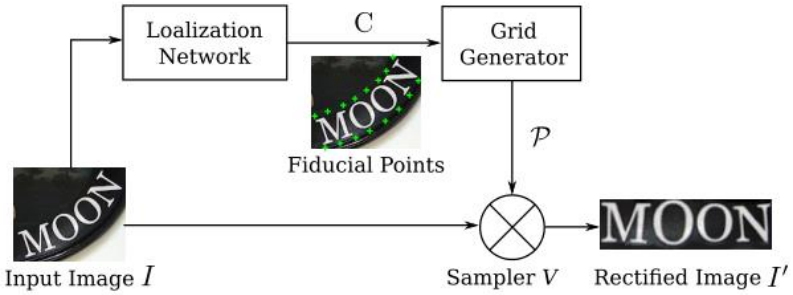

图 3.4 薄板样条插值模块[34]

最后通过连接时序分类函数转录出最终结果，连接时序分类函数采用贪心策略，搜索每个时刻的输出概率作为最终输出，而不考虑全局最优解。实验证明该贪心策略能大大降低运算时间，同时达到较高的准确率。

3.2.3 后处理模块

转录层生成的预测序列中包含重复字符以及空白字符，因此后处理模块用于将其转化成正常的文字序列。后处理模块首先将生成序列中的所有的相邻重复字符只保留一个，将剩余的删去，然后取出所有的空白字符得到最终的预测序列。

3.3 文字检测与识别系统的实现

本文中的文字检测与识别模型通过一个配置文件进行训练与预测，配置文件的主要内容包括基础配置(训练模型使用的 GPU 编号、是否进行预训练、图像裁剪大小、epoch 数量、开始验证的 epoch 数和模型文件存储路径等)、模块选择、损失函数、优化器选择、训练集和测试集等内容。通过配置文件可以方便地对模型进行训练，验证与预测。

文字检测与识别系统主要包含的系统功能有用户交互功能和文字检测与文字识别

功能。用户交互功能，负责与用户进行交互，通过可视化界面接受用户的输入以及传递给用户输出。文字检测与文字识别功能，负责对用户输入的图像进行文字检测与文字识别以及后处理，通过调用后端的文字检测模型与文字识别模型对用户输入进行处理并返回给前端结果。

3.3.1 系统模块设计

本系统主要包含三个主要模块，分别是：用户交互模块、文字检测模块和文字识别模块。

1、用户交互模块

用户交互模块的作用是为用户与系统进行交互提供一个图形化界面。用户交互模块的主要功能有两个，一方面是接受用户的输入上传到后台系统，并交付给文字检测模块和文字识别模块进行处理；另一方面是接受后台处理的结果将其传递给用户，显示在图形化界面上。

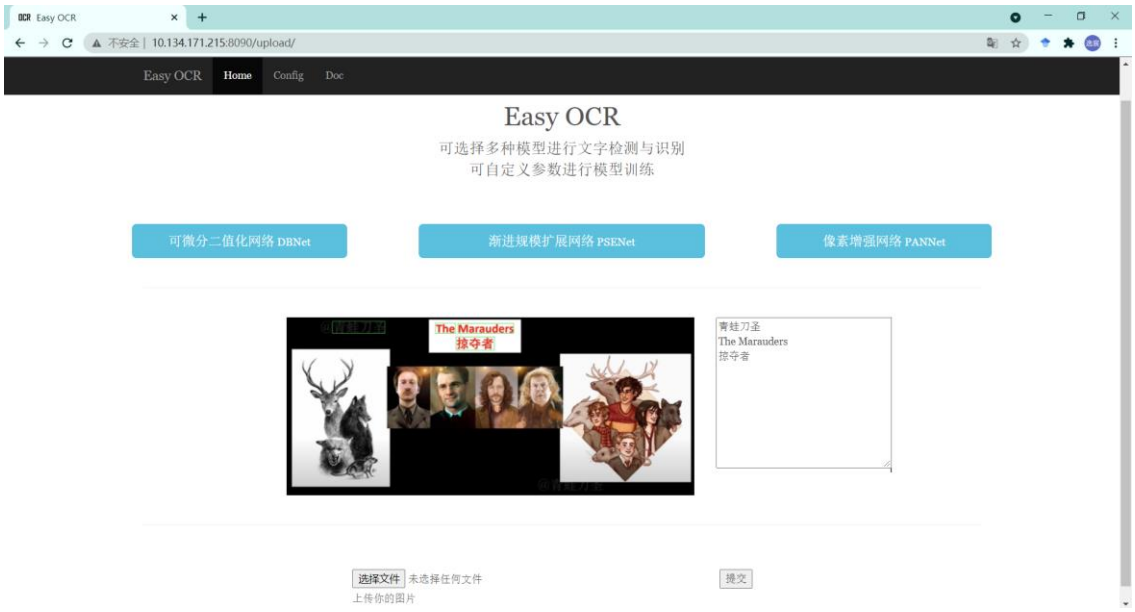

图 3.5 用户交互界面

图 3.5 是用户交互模块的效果界面展示图。用户交互界面上，用户点击选择文件按钮会弹出系统对话框选择文件，选择好需要上传的图案文件后点击提交，文件就会上传到后端服务器，随后文字检测的结果显示在左边图像上，文字识别的结果显示在右边的方框中。

2、文字检测模块

文字检测模块的主要功能是对用户上传的图像进行文字检测处理。文字检测模块对

图像的处理过程主要分为三个步骤，前处理、文字检测和后处理。前处理阶段主要是对输入图像进行图像像素变换，使得输入图像符合文字检测模型的输入标准，包括对像素进行减均值、除方差、归一化、通道变换步骤。文字检测阶段主要是调用文字检测模型对经过预处理的图像进行文字检测，得到文字检测框结果。后处理阶段主要是提取文字检测框结果，由于预测出的文字检测框通常是四边形形式，需要将四边形的四个顶点坐标映射到矩形框的四个顶点，并对图像进行旋转、伸缩变化。

3、文字识别模块

文字识别模块的主要功能是对文字检测出的文字区域进行文字识别处理。文字识别模块接收来自于文字检测模块处理的文字检测框结果进行处理，文字识别模块主要包含前处理、文字识别、后处理三个步骤。前处理步骤主要是对输入文字检测框图像进行灰度化、减均值、除方差、归一化等操作，将图像处理成文字识别模型需要的输入形式。文字识别步骤主要是调用文字识别模型进行处理，得到预测结果。后处理步骤主要是对文字识别模型的预测结果进行后处理，包括文字合并和根据不同场景下的规则对文字进行修正等。最终文字识别模块得到的是文字检测框图像对应的文字数据。

3.3.2 开发环境和系统部署

本文在设计与实现系统时的系统开发环境如表 3.1 所示， 后端开发语言主要以 Python 为主，深度学习开发工具基于 PyTorch。

表 3.1 系统开发环境

后端开发语言：python	前端开发语言：HTML、CSS3、JavaScript
前后台框架：Django	前端框架：Bootstrap
深度学习工具：pytorch	服务器操作系统：Ubuntu 18.04

所有系统都运行在 Linux 系统服务器上，服务端后台和数据库交互使用 Python 语言，深度学习模型部分使用 PyTorch 训练，以服务形式部署供服务器后端调用。

3.4 系统效果展示

本节主要对不同场景下的图像文字检测与识别结果进行展示。图 3.6 中英文混杂的密集文本场景，图中绿色的框是文本检测框，右边方框中是文字识别的结果，在该场景下文字较为集中，本系统依旧能够很好的进行处理。

图 3.6 系统效果展示 1

图 3.7 系统效果展示 2

图 3.7 是纯英文的稀疏文本场景，在该场景下文字的大小较大，容易与其他图案混淆，本系统依旧能够很好的进行处理。

3.5 本章小结

本章首先介绍了文字检测和识别网络的具体实现细节，以及对上采样分割头和文字识别模型的改进。然后介绍了整个文字检测和识别系统的实现流程，即使用统一的配置文件对系统进行训练、预测和验证。接着介绍了系统的主要功能和应用场景，结构设计和模块划分，包括用户交互模块、文本检测模块和文本识别模块，并详细介绍了各个模块的功能和设计以及实现要点。除此之外，本章还介绍了系统的开发环境和系统部署，包括前后端与深度模型开发过程中使用到的工具、框架、语言等。最后，本章进行了两种不同场景下的可视化效果展示，展示了在两种不同场景下的文本检测与识别效果。

4 实验设计与结果分析

本章针对前文提出的算法与系统设计并进行了五项实验，对选用的数据集和评价指标进行了详细的介绍，并对实验结果进行了分析与讨论。

4.1 数据集介绍

文字检测的数据集给定一张图像和图像中文字包围框的顶点坐标作为标签，如图 4.1 所示。

854,454,899,454,899,465,854,465
787,335,865,335,865,365,787,365
787,301,903,301,903,335,787,335
820,285,871,285,871,298,820,298
873,283,904,283,904,300,873,300
786,282,819,282,819,299,786,299
944,254,971,254,971,261,944,261
667,156,691,156,691,170,667,170

图 4.1 文字检测数据集介绍

文字识别的数据集分为两种，一种是直接提供上述文字包围框的坐标以及框中文字内容，需要自行裁剪用于训练。另一种是直接提供裁剪好的图像，可以直接进行训练。

表 4.1 模型训练用到的数据集

数据集	语言	训练集	测试集
Icdar2013 focused scene text	英语	229	233
Icdar2015 incidental scene text	英语	1000	500
COCO Text	英语	15124	
Icdar2017 reading Chinese in wild	汉语	8034	
Icdar2019 large-scale street view text	汉语	30000	
Icdar2019 Reading text on signboard	汉语	20000	

本文使用到的数据集见表 4.1。其中 Icdar2013 聚焦场景文本的重点是在真实场景中阅读文字。“聚焦文本”，是指文字内容清晰的图像。这是文字阅读和文本翻译应用程序的典型场景，其中用户明确地将相机聚焦于感兴趣的文字内容。同时，这是目前大多数方法和数据集所处理的典型场景。Icdar2015 附带场景文本是与 Icdar2013 类似的真实场景图像。但与 Icdar2013 基于聚焦于文本内容的图像(捕捉良好的水平英文文本，通常位

于图像中心)相反，Icdar2015 是用户没有采取任何特定的事先动作来改善文字在图像中的位置或质量的附带场景文本，通常使用 google 眼镜等可穿戴设备进行随时抓拍。COCO Text[8]是一种用于自然图像文本检测和识别的新型大尺度数据集。根据易读性(可辨认文本和不可辨认文本)、类别(机器打印文本和手写文本)和文字类型等对图像进行分类。

Icdar2017 reading Chinese in wild、Icdar2019 large-scale street view text 和 Icdar2019 Reading text on signboard 为三个街景图像中文数据集，与之前英文数据集的作用类似，但提供了更多的图像。

4.2 评价指标

4.2.1 文字检测

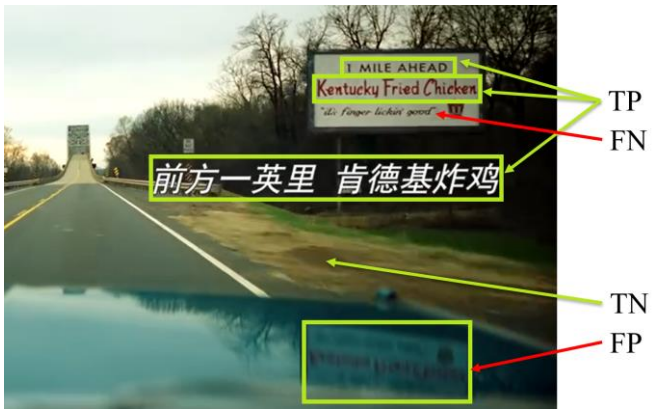

图 4.2 四种识别结果

文字检测的性能评价采用与图像分类类似的评价指标，即计算精确度和召回率来计算 F1 值。定义四种识别结果：真阳性(TP，True Positive)，预测出文字区域实际标注也是文字区域；假阴性(FN，False Negtive)，预测出非文字区域实际标注是文字区域；真阴性(TN，True Negtive)，预测出非文字区域实际标注也是非文字区域；假阳性(FP，False Positive)，预测出文字区域实际标注是非文字区域。如图 4.2 所示

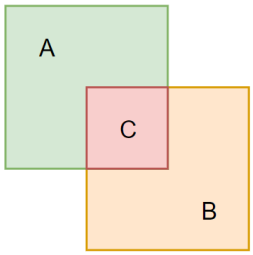

图 4.3 交并比示例图

判断一个预测框的识别结果时，采用交并比(IoU)作为指标，交并比大于 0.5 的被作为 TP。交并比是指两个集合的交集与并集的比值，图 4.3 中的交并比可表示为：

$$IoU = \frac{A \cap B}{A + B - A \cap B} \tag{4.1}$$

精确度(Precision)是指在识别出来的图像中，真阳性所占比例：

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

召回率(Recall)是指数据集中所有文字区域中，被正确识别出的比例：

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

在本例中精确度和召回率都为 75%。

4.2.2 文字识别

文字识别使用编辑距离衡量两个串之间的差异，朴素编辑距离的计算方法需要消耗巨大的时间空间资源，因此可以采用动态规划的方法解决这个问题。对于两个串A和B，A_i和B_j分别表示A和B的第i和第j个字符。将$A_{1..i}$修改为$B_{1..j}$可以在将$A_{1..i}$修改为$B_{1..j-1}$的基础上插入一个B_j，这是插入操作；可以在将$A_{1..i-1}$修改为$B_{1..j}$的基础上删除一个A_i，这是删除操作；还可以在将$A_{1..i-1}$修改为$B_{1..j-1}$的基础上将A_i修改为B_j(如果A_i和B_j相等则无需修改)，这是修改操作。状态转移方程为：

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j)， & if\ \min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases} \tag{4.4}$$

4.3 结果分析

4.3.1 原始文字检测模型训练结果

表 4.2 原始文字检测模型训练结果

模型	语言	验证集	precision	recall	F1 score
DBNet	英	icdar2015 Ist	0.8537(0.882)	0.7900(0.827)	0.8207(0.854)
PSENet			0.8270(0.869)	0.7929(0.845)	0.8101(0.857)
PAN			0.8370(0.840)	0.7468(0.819)	0.7893(0.829)
DBNet	中&英	icdar2019 ReCTS	0.8522	0.7968	0.8235
PSENet			0.8430	0.6750	0 .7497
PAN			0.8042	0.7438	0.7728

表 4.2 展示了复现的三种基于分割的文字检测模型的训练与测试结果，其中括号中的数据为论文中的验证结果，其他数据为本文的验证结果。可以看到对于纯英文的街景图像文字检测和中英文的街景图像文字检测，DBNet 都具有最优的结果。

4.3.2 模块化文字检测模型的组合结果

本节研究了基于分割的文字检测模型各阶段不同模块的特点以及组合的训练结果。其中主要研究上采样分割头和分割输出的异同，因此骨干网络统一采用 Resnet50；上采样分割头有三种选择：特征金字塔网络上采样分割头(FPN_Head)、改进特征金字塔网络上采样分割头(IMP_FPN_Head)和特征金字塔增强与融合分割头(FPEM_FFM_Head)；分割输出有三种选择：渐进规模扩展网络(PSE_Segout)、像素聚合网络(PAN_Segout)和可微分二值化网络(DB_Segout)。

改进特征金字塔网络上采样分割头通过先使用卷积来减少通道数在进行拼接的方法，使得分割头的大小减少了 27M。改进后的特征融合模块参数增加了 25M。改进前后的特征金字塔网络上采样分割头存在一定的性能差异，但是改进后的特征金字塔增强与融合分割头与改进之前的性能相近，因此本文最终只保留了改进前的特征金字塔增强与融合分割头。总的来说 IMP_FPN_Head 的参数最少，FPN_Head 的参数最多。同时可从训练网络时的平均占用显存以及模型的大小得出类似的结论，具体数据见表 4.3。

表 4. 3 上采样分割头参数数量比较

模块组合	占用显存大小 MB	模型大小 MB
Resnet50+FPN_Head+PSE_Segout	12945	219
Resnet50+IMP_FPN_Head+PSE_Segout	8090	192
Resnet50+TRAD_FFM_Head+PSE_Segout	14080	211
Resnet50+FPEM_FFM_Head+PSE_Segout	12215	196

实验证明改进后的特征融合模块性能改进效果并不明显，但大大增大了计算量，模型大小和运算速度。受到[33]中位置信息嵌入词向量的方式是直接相加的启发，猜想使用逐元素相加而不是拼接的方法可以有效地对特征进行提取和融合。

使用同样的方法比较分割输出的参数数量，得出三个分割输出网络的参数数量相近，可微分二值化网络的参数最多，结果见表 4.4。

表 4. 4 分割输出参数数量比较

模块组合	占用显存大小 MB	模型大小 MB
Resnet50+FPEM_FFM_Head+PSE_Segout	12215	196
Resnet50+FPEM_FFM_Head+PAN_Segout	11479	196
Resnet50+FPEM_FFM_Head+DB_Segout	13785	199

对于模型收敛速度以及准确率的实验结果见表 4.5。渐进规模扩展网络和像素聚合

网络的训练收敛速度较快，以 4000 张训练图像作为参照，最佳的 epoch 数为 600 左右，而可微分二值化网络需要 1200 个 epoch。改进特征金字塔网络上采样分割头由于其参数较少，收敛速率略优于其余两者。在测试集上 Resnet50+FPEM_FFM_Head+DB_Segout 获得了最优的表现；Resnet18+IMP_FPN_Head+PAN_Segout 拥有最快的收敛速度以及最少的参数数量。

表 4.5 模型收敛速度以及准确率

分割输出	训练 epoch 数	上采样分割头		
		FPN_Head	FPEM_FFM_Head	IMP_FPN_Head
PSE_Segout	300	0.8659	0.8425	0.9135
	600	**0.9223**	**0.9263**	**0.9266**
	900	0.9335	0.9334	0.9257
	1200	0.9457	0.9307	0.9362
PAN_Segout	300	0.8822	0.8597	0.8997
	600	**0.9106**	**0.9125**	**0.9174**
	900	0.9263	0.9236	0.9306
	1200	0.9503	0.9435	0.9459
DB_Segout	300	0.7337	0.7401	0.7476
	600	0.8452	0.8538	0.8698
	900	0.8992	0.9003	0.8964
	1200	**0.9306**	**0.9308**	**0.9295**

4.3.3 文字识别模型训练结果

薄板样条插值可以改善图片中文字的扭曲程度，使用 icdar2013 验证集对使用传统 CRNN 模型进行验证时，准确率为 90.44%，在卷积层之前加入薄板样条插值后，准确率达到了 91.15%，具体结果见表 4.6。

表 4.6 文字识别模型测试效果

方法名	CRNN	CRNN+Tps
精确度	90.44%	91.15%

4.3.4 渐进规模扩展网络消融研究

渐进规模扩展网络中存在两个超参数 m(最大收缩比例)和 n(嵌套核的数量)，该超参数的取值影响着整个系统的性能。其中超参数 n 的选取与测试结果见表 4.7。

表 4.7 超参数 n 的选取与对应的 F1 值

n	2	3	4	5	6	7	8	9	10
F1	80.63	80.84	80.95	80.98	81.00	81.01	81.01	81.01	81.01

超参数 m 的选取与测试结果见表 4.8。

表 4.8 超参数 m 的选取与对应的 F1 值

m	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
F1	76.56	78.72	80.78	81.01	80.90	80.83	80.55	79.67	76.64

图 4.4 展示了 m 和 n 选取与 F1 值关系的折线图，本文选取 n=7，m=0.4。其中模型的参数数量随着 n 的增大而增大，当 n≥7 时 F1 值几乎不再随着 n 的增大而增大，因此 n 取 7 最佳。

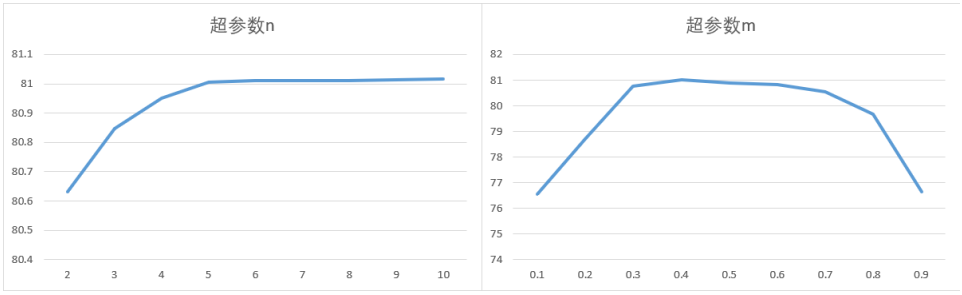

图 4.4 渐进规模扩展网络 m 和 n 的消融实验

4.3.5 像素聚合网络的消融研究

像素聚合网络中存在超参数 n(相似向量的维数)该超参数的取值影响着整个系统的训练收敛速度，模型大小以及准确度。其中超参数 n 的选取与测试结果见表 4.9。

表 4.9 超参数 n 的选取与对应 F1 值

n	2	3	4	5	6	7
F1	57.37	72.86	78.93	78.93	78.94	78.94

图 4.5 展示了 n 选取与 F1 值关系的折线图，本文选取 n=4。模型的参数数量随着 n 的增大而增大，当 n≥4 时 F1 值几乎不再随着 n 的增大而增大，因此 n 取 4 最佳。

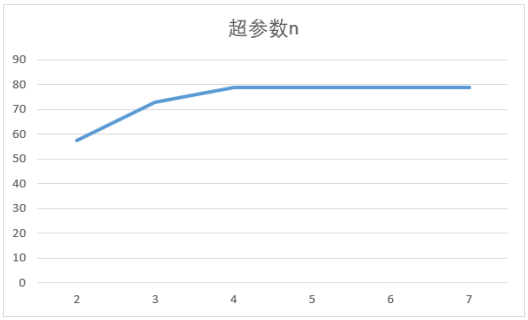

图 4.5 像素聚合网络 n 的消融实验

4.4 本章小结

本章首先介绍了文字检测与识别领域常用的数据集以及评价指标，然后进行了实验介绍以及结果分析。Resnet18+IMP_FPN_Head+PAN_Segout 拥有最快的收敛速度以及最

少的参数数量，适合在移动端等部署使用；Resnet50+FPEM_FFM_Head+DB_Segout 拥有最高的准确率，但训练收敛速度较慢，参数量大，适合在服务器端部署使用。接着，通过实验证明了添加薄板样条插值模块对文字识别 CRNN 模型的检测效果有明显的改善。最后通过消融实验分析对模型的超参数选作了说明。

总结与展望

当今社会是一个信息量爆炸的社会，如何从海量的信息中获取有用的信息进行分析研究是当前研究的重点问题之一。而从图像中抽取信息也是研究的重点问题之一。使用光学字符识别技术从包含数据和文字在内的各种图像档案信息进行数据分析和识别处理，获得数据和文字以及版面信息对于信息的抽取与过滤有着重要的帮助作用。光学字符识别通常包含文字检测和文字识别两个步骤。文字检测用于在图像中定位文字区域，并提供其所在位置的信息；而文字识别则进一步地对图像中该区域内的所有文字都进行识别，将其转化成计算机能够理解和处置的各种文字信息。

光学字符识别不仅在计算机视觉领域中具有重要的研究价值，而且同时也具有广阔的应用场景与应用需求。在光学字符识别的研究中，不仅需要应对文字种类和文字字体的差异，还要应对各种各样复杂背景的干扰。

本文针对街景图像中英文文字检测与识别问题提出了一个完整的算法，算法包括两个子任务：文字检测和文字识别。文字检测部分对比分析了三个基于分割的文字检测模型，并对上采样分割头进行了改进。结果表明 Resnet18+IMP_FPN_Head+PAN_Segout 拥有最快的收敛速度以及最少的参数数量，适合在移动端等部署使用；Resnet50+FPEM_FFM_Head+DB_Segout 拥有最高的准确率，但训练收敛速度较慢，参数量大，适合在服务器端部署使用。文字识别部分采用 CRNN 模型，同时引入薄板样条插值模块，提高了文字识别的准确性。

在未来的研究中，街景图像的文字检测与识别向票据卡证等具体场景转变时可以采用迁移学习的方法，在一定程度上减少对数据及规模的要求与依赖。现有训练方法依赖于大量的标注数据进行，在今后的工作中，可以尝试半监督学习的方法，以减少人力资源，同时需要提高模型的响应速度，减小模型大小，对一些中文单字，特殊形状文本的泛化性能需要提高。

致谢

大学生活就这样不知不觉地走过四年，四年间我经历了北航的春夏秋冬，日出日落，也见证了自己的一点一滴的进步与成长。知行合一，德才兼备的校训时时刻刻鞭策着我，仰望星空，脚踏实地的愿望也时时刻刻鼓励着我前行。四年之前，我怀着期待与憧憬踏入校园，四年后的今天我交上了本科的最后一份答卷，释然且毫无遗憾。

四年时间里感谢老师同学们给予我的帮助，感谢家人们在我背后无声的支持与鼓励。尤其感谢我的指导老师李舟军教授，在我保研后加入实验室，李老师为我提供了良好的学习环境，经常给我提供前沿的研究成果与学习资料，并督促我认真学习研究，使得我能在短短时间内深入了解光学字符识别领域的相关内容。

感谢我的家人们为我提供源源不断的支持，让我没有后顾之忧地勇敢向前；感谢我的朋友们愿意与我一起分享喜悦与悲伤；感谢我的同学们与我一起交流学习，对我的学习生活提供了莫大的帮助；感谢所有帮助过我的人，相见即是缘，你们都是我生命中的贵人！

逝者如斯夫，不舍昼夜。大学生活转眼就结束了，但我学到的知识、思想与技能会伴随我一直走下去。书到用时方恨少，写论文时我深切感受到自己的水平还有待提高；纸上得来终觉浅，将想法付诸实践是使我进步的不竭动力源泉。生命不息，学习不止，我还有许多需要学习与改进的地方，我会怀揣理想，一步一个脚印地继续走下去！

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. Springer, Cham, 2016: 56-72.

[2] 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.

[3] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[4] 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.

[5] 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.

[6] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]. European conference on computer vision. Springer, Cham, 2016: 21-37.

[7] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9

[8] Veit A, Matera T, Neumann L, et al. Coco-text: Dataset and benchmark for text detection and recognition in natural images[J]. arXiv preprint arXiv:1601.07140, 2016.

[9] 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.

[10] Wang W, Xie E, Li X, et al. Shape robust text detection with progressive scale expansion network[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 9336-9345.

[11] Wang W, Xie E, Song X, et al. Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network[C]. Proceedings of the IEEE International Conference on Computer Vision. 2019: 8440-8449.

[12] Liao M, Wan Z, Yao C, et al. Real-time scene text detection with differentiable binarization[C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2020,

34(07): 11474-11481.

[13] De Boer P T, Kroese D P, Mannor S, et al. A tutorial on the cross-entropy method[J]. Annals of operations research, 2005, 134(1): 19-67.

[14] 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.

[15] 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. 2006: 369-376.

[16] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[17] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation[R]. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[18] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.

[19] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

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

[21] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[22] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[23] Milletari F, Navab N, Ahmadi S A. V-net: Fully convolutional neural networks for volumetric medical image segmentation[C]. 2016 fourth international conference on 3D vision (3DV). IEEE, 2016: 565-571.

[24] Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard example mining[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 761-769.

[25] Xue C, Lu S, Zhan F. Accurate scene text detection through border semantics awareness and bootstrapping[C]. Proceedings of the European conference on computer vision (ECCV). 2018: 355-372.

[26] Milletari F, Navab N, Ahmadi S A. V-net: Fully convolutional neural networks for volumetric medical image segmentation[C]. 2016 fourth international conference on 3D vision (3DV). IEEE, 2016: 565-571.

[27] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1314-1324.

[28] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[29] Chollet F. Xception: Deep learning with depthwise separable convolutions[C] . Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.

[30] Vatti B R. A generic solution to polygon clipping[J]. Communications of the ACM, 1992, 35(7): 56-63.

[31] He P, Huang W, Qiao Y, et al. Reading scene text in deep convolutional sequences[C]. Proceedings of the AAAI conference on artificial intelligence. 2016, 30(1).

[32] Lee C Y, Osindero S. Recursive recurrent nets with attention modeling for ocr in the wild[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2231-2239.

[33] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. arXiv preprint arXiv:1706.03762, 2017.

[34] Shi B, Wang X, Lyu P, et al. Robust scene text recognition with automatic rectification[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 4168-4176.

附录 A 长短期记忆网络的推导

遗忘门的输出为f_t，采用 sigmoid 激活函数，将输出映射到[0,1]区间。上一时刻细胞状态c_{t-1}。通过遗忘门时，结果与f_t相乘，显然，乘数为 0 的信息被全部丢弃，为 1 的被全部保留。这样就决定了上一细胞状态c_{t-1}有多少能进入当前状态c_t。遗忘门f_t的公式如下：

$$f_t = \sigma\left(h_{t-1} \cdot W_f + x_t \cdot U_f + b_f\right) \tag{A1}$$

其中，σ为 sigmoid 激活函数，h_{t-1}为上一时刻的隐层状态。x_t为当前时刻的输入。

输入门i_t决定输入信息有哪些被保留，输入信息包含当前时刻输入和上一时刻隐层输出两部分，存入即时细胞状态$\tilde{c}_t$中。输入门依然采用 sigmoid 激活函数，$\tilde{c}_t$通过输入门时进行信息过滤。输入门i_t的公式如下：

$$i_t = \sigma(h_{t-1} \cdot W_i + x_t \cdot U_i + b_i) \tag{A2}$$

即时细胞状态$\tilde{c}_t$的公式如下:

$$\tilde{c}_t = tanh(h_{t-1} \cdot W_c + x_t \cdot U_c + b_c) \tag{A3}$$

上一时刻保留的信息，加上当前输入保留的信息，构成了当前时刻的细胞状态c_t。当前细胞状态c_t的公式如下：

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{A4}$$

其中，符号·表示矩阵乘积，∘表示 Hadamard 乘积，即元素乘积。最后，需要确定输出信息。输出门o_t决定h_{t-1}和x_t中哪些信息将被输出，公式如下：

$$o_t = \sigma(h_{t-1} \cdot W_o + x_t \cdot U_o + b_o) \tag{A5}$$

细胞状态c_t通过 tanh 激活函数压缩到 (-1, 1) 区间，通过输出门，得到当前时刻的隐藏状态h_t作为输出，公式如下：

$$h_t = o_t \circ tanh(c_t) \tag{A6}$$

最后，时刻t的预测输出为：

$$a_t = \sigma(h_t \cdot V + b) \tag{A7}$$

若循环神经网络的状态更新方程为：

$$S_t = tanh(W_t S_{t-1} + W_x X_t + b_1) \tag{A8}$$

那么在反向传播中，存在这样的链式求导项：

$$\prod_{j=k+1}^{t}\frac{\partial S_j}{\partial S_{j-1}}=\prod_{j=k+1}^{t}tanh'W_S \tag{A9}$$

其中如果$tanh'W_S$小于 1，则结果趋于 0，出现梯度消失现象；相反的如果$tanh'W_S$大于 1，则结果会趋于无穷，出现梯度爆炸现象。对于式$(A4)$进行链式求导：

$$\prod_{j=k+1}^{t}\frac{\partial c_j}{\partial c_{j-1}}=\prod_{j=k+1}^{t}tanh'f_t \tag{A10}$$

因为f_t的结果为大都非 0 即 1，因此式$(A10)$的结果$\prod_{j=k+1}^{t}tanh'f_t\approx 0|1$，这就解决了在更新长期记忆时梯度消失和爆炸的问题。

附录 B 连接时序分类损失函数的推导

对于一个循环层输出序列$y=(y^1,\dots,y^T)$，其中T为长短期记忆网络的输出个数，y^t为：$y^t=(y_1^t,\dots,y_n^t)$，n 字典大小，y_i^t服从概率分布$\sum_k y_k^t=1$。

因为不是所有y^t都有字符与之对应，所以引入空白符号$-$。对于英文字母集合$L=\{a,b,c,\dots,x,y,z\}$，插入空白符之后集合变为$L'=L\cup\{-\}$。

定义β变换，用于将L'序列转化为 L 序列：$\beta:L'^T\to L^{\leq T}$。举例说明，当$T=12$时：

$$\beta(\pi_1)=\beta(--haap-pp--y)=happy$$

$$\beta(\pi_2)=\beta(hha-ppp-pyy-)=happy$$

对长短期记忆网络的输出进行β变换，即可得到输出结果。β是一个多对单映射。

使用连接时序分类损失函数进行预测时，对于循环层预测的结果x，输出为l的概率为：

$$p(l|x)=\sum_{\pi\in\beta^{-1}(l)}p(\pi|x) \tag{B1}$$

其中，$\pi\in\beta^{-1}(l)$代表所有经过β变换后是l的路径π。对于任意一条路径π有：

$$p(\pi|x)=\prod_{t=1}^{T}y_{\pi_t}^t \tag{B2}$$

对于上述$\pi_1=(--haap-pp--y)$路径来说：

$$p(\pi_1|x)=y_-^1\cdot y_-^2\cdot y_h^3\cdot y_a^4\cdot y_a^5\cdot y_p^6\cdot y_-^7\cdot y_p^8\cdot y_p^9\cdot y_-^{10}\cdot y_-^{11}\cdot y_y^{12} \tag{B3}$$

实际情况中$\pi\in\beta^{-1}(l)$有非常多种可能性，因此需要一种快速的计算方法。要计算$p(l|x)$，定义路径l'为在路径l每两个元素之间以及头尾插入$-$。即

$$l=happy$$

$$l'=-h-a-p-p-y-$$

定义所有经β变换结果是l且在t时刻结果为l_k(记为$\pi_t=l_k$)的路径集合为$\{\pi|\pi\in\beta^{-1}(l),\pi_t=l_k\}$

$$\frac{\partial p(l|x)}{\partial y_k^t}=\frac{\partial\sum_{\pi\epsilon\beta^{-1}(l)}p(\pi|x)}{\partial y_k^t}=\frac{\partial\sum_{\pi\epsilon\beta^{-1}(l),\pi_t=l_k}p(\pi|x)}{\partial y_k^t} \tag{B4}$$

即与$\frac{\partial p(l|x)}{\partial y_k^t}$有关的路径$t$时刻都是$l_k$。

$$\sum_{\beta^{-1}(l),\pi_t=l_k} p(\pi|x) = \text{forward} \cdot y_k^t \cdot \text{backward} \tag{B5}$$

定义前向递推概率和forward $= \alpha_t(s)$。

$$\alpha_t(s) = \sum_{\pi\in\beta(\pi_{1:t})=l_{1:s}} \prod_{t\prime=1}^{t} y_{\pi_{t\prime}}^{t\prime} \tag{B6}$$

由β变换的特点可以得到$\alpha_t(s)$的递推公式：

$$\alpha_t(l'_k) = \left(\alpha_{t-1}(l'_k) + \alpha_{t-1}(l'_{k-1}) + \alpha_{t-1}(-)\right) \cdot y_{l\prime_k}^t \tag{B7}$$

同理定义反向递推概率和backward $= \beta_t(s)$，

$$\beta_t(s) = \sum_{\pi\in\beta(\pi_{t:T})=l_{s:|l|}} \prod_{t\prime=t}^{T} y_{\pi_{t\prime}}^{t\prime} \tag{B8}$$

$$\beta_t(l'_k) = \left(\beta_{t+1}(l'_k) + \beta_{t+1}(l'_{k+1}) + \beta_{t+1}(-)\right) \cdot y_{l\prime_k}^t \tag{B9}$$

因此

$$\alpha_t(l'_k)\beta_t(l'_k) = \sum_{\pi\in\beta^{-1}(l),\pi_t=l\prime_k} y_{l\prime_k}^t \prod_{t=1}^{T} y_{\pi_t}^t \tag{B10}$$

$$\alpha_t(l_k)\beta_t(l_k) = \sum_{\pi\in\beta^{-1}(l),\pi_t=l_k} y_{l_k}^t \prod_{t=1}^{T} y_{\pi_t}^t \tag{B11}$$

$p(l|x)$与forward和backword递推公式之间的关系是：

$$p(l|x) = \sum_{\pi\in\beta^{-1}(l),\pi_t=l_k} \frac{\alpha_t(l_k)\beta_t(l_k)}{y_{l_k}^t} \tag{B12}$$

在训练时，通过梯度$\frac{\partial p(l|x)}{\partial \omega}$调整参数$\omega$，使得对于输入样本为$\pi \in \beta^{-1}(z)$时有$p(l|x)$取得最大。

$$\frac{\partial p(l|x)}{\partial y_{l_k}^t} = \frac{\partial \sum_{\pi\in\beta^{-1}(l),\pi_t=l_k} \frac{\alpha_t(l_k)\beta_t(l_k)}{y_{l_k}^t}}{\partial y_{l_k}^t} = \frac{\sum_{\pi\in\beta^{-1}(l),\pi_t=l_k} \alpha_t(l_k)\beta_t(l_k)}{(y_{l_k}^t)^2} \tag{B13}$$

上式中的$\alpha_t(l_k)\beta_t(l_k)$是通过地推计算的常数，因此该公式可以快速计算出梯度进行反向传播。