

**毕 业 设 计（论 文）**



**设计(论文)题目：** 基于深度学习的街景OCR文字检测

学生姓名： 刘运星 指导教师： 陶玉婷

二级学院： 软件工程 专　　业： 软件工程

班　　级： 15软件工程（J） 　　号： 1512001034

提交日期： 年 月 日 答辩日期： 年 月 日

目 录

[摘 要 III](#_Toc7993541)

[Abstract IV](#_Toc7993542)

[1 绪 论 1](#_Toc7993543)

[1.1研究背景及意义 1](#_Toc7993544)

[1.2国内外现状 1](#_Toc7993545)

[1.3评价标准与实验数据集 1](#_Toc7993546)

[1.4本文主要研究内容 2](#_Toc7993547)

[1.5本文设计目的以及组织结构 3](#_Toc7993548)

[1.6本章小结 3](#_Toc7993549)

[2 深度学习技术与OCR技术 4](#_Toc7993550)

[2.1光学文字识别技术简介 4](#_Toc7993551)

[2.2常见OCR技术 4](#_Toc7993552)

[2.3卷积神经网络 5](#_Toc7993553)

[2.4目标检测网络介绍 8](#_Toc7993554)

[2.5本章小结 9](#_Toc7993555)

[3 基于TextBoxes的自然场景文本检测模型 10](#_Toc7993556)

[3.1TextBoxes中的神经网络 10](#_Toc7993557)

[3.2TextBoxes中的NMS算法 11](#_Toc7993558)

[3.2本章小结 12](#_Toc7993559)

[4 对于TextBoxes模型的拓展 13](#_Toc7993560)

[4.1对于读取图片方式的改进 13](#_Toc7993561)

[4.2 TextBoxes中模型识别评价标准 14](#_Toc7993562)

[4.3对TextBoxes中MNS的改进 16](#_Toc7993563)

[4.4本章小结 18](#_Toc7993564)

[5 系统介绍与实现 19](#_Toc7993565)

[5.1系统需求分析 19](#_Toc7993566)

[5.2系统详细设计 19](#_Toc7993567)

[5.3系统实现 20](#_Toc7993568)

[5.4成果展示 23](#_Toc7993569)

[5.5本章小结 23](#_Toc7993570)

[6 总结与展望 25](#_Toc7993571)

[6.1设计总结 25](#_Toc7993572)

[6.2模型展望 25](#_Toc7993573)

[6.3应用系统展望 26](#_Toc7993574)

[参考文献 27](#_Toc7993575)

[致 谢 29](#_Toc7993576)

基于深度学习的街景OCR文字检测

# 摘 要

OCR（Optical Character Recognition）即光学文字识别技术，OCR技术所作的工作内容为将图片中的文字翻译成计算机可编辑的文字。传统文字识别方法是通过图像上的形状特征，来确定其形状然后通过已知的算法模型识别字符将文字分类，这种方法需要严格描述特征。而这种方法一般仅限于在背景简单光线条件较好的情况下使用，当背景变得复杂干扰性强的时候这种识别方式的效果就没有那么明显。在自然场景下的文字背景通常都是复杂的，传统的OCR检测算法无法很好的完成文字识别的过程。深度学习的方法区别于传统的做法，摒弃了传统的依赖于人工设计文字特征的建模识别方式，利用大量的数据集从中自动学习特征实现文字识别，能够很大程度上提高文字的识别率。

本文所研究的内容建立在模型TextBoxes之上主要研究通过深度学习的方式检测文本所在区域，而要达到文字识别的目的可以进一步在检测出的结果的基础之再次通过CRNN模型识别出检测到的文字结果。其中TextBoxes采用的是改进版本的SSD目标检测网络。并且本设计针对TextBoxes在GitHub上的源码进行了扩展，同时对于实验过程中TextBoxes存在的问题进行猜想分析并提出可行方案。

本设计以TextBoxes模型为检测模型开发了文字检测App，使用Python作为开发后台，具有Android端和网页端两端。

关键词：文字检测；深度学习；TextBoxes；系统

**Street View OCR Text Detection Based on Deep Learning**

# Abstract

Optical Character Recognition (OCR), or optical character recognition technology, is a process of recognizing and translating characters existing in pictures into computer words.The traditional method of character recognition is to determine the shape of an image by its shape features, and then to classify the characters by the known algorithm model.This method requires a strict description of features. However, this method is generally limited to the case of simple background light conditions, and when the background becomes more complex and disturbing, the effect of this recognition method is not so obvious.In natural scenes, the text background is usually complex, but the traditional OCR detection algorithm can not complete the process of text recognition very well.

The content of this paper is based on the model TextBoxes. It mainly studies how to detect the region of the text through in-depth learning. To achieve the purpose of text recognition, we can further train the results of the detection. TextBoxes uses an improved version of the SSD target detection network.TextBoxes model on Caffe is built to realize text detection. Python is used as Sever,and B/S mode is used to develop street scene text detection system ,The front end was realized by React.

**Key words:** Text detection; Deep Learning; TextBoxes; System

# 1 绪 论

1.1研究背景及意义

在生活中信息往往无处不在，一张街景图所蕴含的信息是巨大的。例如在路口的路标，告诉公路上的路牌，以及各大商场的花样百出的广告牌。识别街景上的文字图片不仅能够获得和解析的信息也是十分丰富的，使用街景文字识别后接下来还可以通过翻译程序将文字翻译成不同的语言。不仅如此，街景文字识别还可以与其他算法相结合，能够从街景中识别出违章车辆信息等。

近年来机器学习以及深度学习的浪潮越来越大，并且使用深度学习与机器学习算法在很多领域中都超过了传统的算法效率和准确率，在文字识别领域中亦然。传统的文字识别OCR十分依赖于文字的特征提取而在复杂的背景情况下这些特征就无法明显的区分和表达，在街景情况下认为设计和特征提取效果并不好而深度学习算法可以通过十分大规模的学习从而自动提取特征并识别文字。

在基于深度学习算法的OCR文字检测中一般将文本检测与文本识别分开进行，在文本识别过程中能够有效进行文本检测，有效减少文本识别的成本。所谓文本检测是指使用一定算法将图片中的文本区域找出，所谓文本识别的过程则是基于上述过程所检测到的所在位置进行识别。本文所研究的内容即是在基于神学习的文本检测算法中的文本检测环节。在深度学习的算法基础之上仍然存在着很多难点，例如文字扭曲，文字受到光线影响有较大的曝光等。本文基于文本检测算法实现了文本检测系统，即在Android端和网页端可以拍摄图片，上传到服务器之后，服务器处理之后反馈到系统前台。

1.2国内外现状

基于深度学习的自然场景文字识别既属于模式识别领域，也属于图像分析与文字识别领域，在国内和国外有很多优秀的学者都在这一领域中有不错的成绩。在本文研究的文本检测算法中，所使用的目标检测算法为基于SSD目标检测算法的改进，目前目标检测算法有多种，在例如R-CNN，Fast R-CNN等系列方法进行目标检测R-CNN[1]即Region-CNN是通过候选区域的方式和深度学习分类的方式，提取特征候选区域之后对相应区域进行深度学习方法再进行分类的的方案。除此之外还有包括本所所研究的算法的改进技术SSD[2]与YOLO[3]等。

1.3评价标准与实验数据集

本文的实验数据集一部分采用ICDAR13，此数据集有ground and truth，方便作为实验数据集和测试集使用，同样为了能够是识别效果尽量的好，本文拓展拓展数据集的方案包括以下方案，通过国内外地图提供的api，人工抓取图片，然后进行人工标注。在深度学习训练过程中需要大量数据集，而若要得到大量数据集则需要大量的人力并消耗巨大。另一种方案为采用人工合成使用Synthetic算法将文字自然的贴在图片之上，并且保存图片的位置能够为训练与测试提供有效的ground and truth，能够高效得到大量的具有标准数据的图片，同样可以将方案一与方案二结合，先通过api爬取大量图片，然后将对图片进行处理。

|  |
| --- |
|  |
| 图1 使用Synthetic方法合成的图片 |

方案三使用OpenCV将文字随机的贴在图片之上，并采用加噪处理给识别加大难度，这种方法虽然产生的图片效果并不美观，但是上述算法的成本不高，算法并不复杂而且能在相对较短的时间内得到大量的测试集。

|  |  |
| --- | --- |
|  |  |
| 图2 使用OpenCV合成（右图为原图） | |

本文使用的结果评价标准为类似于非极大抑制算法来评判是否正确，但是在非极大抑制算法的基础之上又进行了一些改进。首先要准备测试集，其次当文本检测之后，除了输出图片之外，将坐标信息输出，与ground and truth种的坐标进行比较，由于没有显著的特征，因此确定ground and truth中与输出结果中的哪一个是对应关系成为了难题。本文认为输出内容与ground and truth匹配度越高，则认为识别率越高，并且通过匹配数量，检测框识别过程中的置信度，目标丢失数量，目标错误识别数量等多方面评价一次的识别结果。

1.4本文主要研究内容

本文的主要研究内容包括OCR识别的一般过程即简单介绍OCR识别的传统算法以及概念，卷积神经网络的介绍即介绍神经网络相关知识，基于深度学习的OCR识别的过程即将深度神经网络与深度学习想结合所进行OCR识别的算法手段。框架TextBoxes的算法的研究，在研究TextBoxes算法的过程中针对不同的数据集进行测试，针对结果进行分析对比。针对TextBoxes算法提出几点不足之处，并针对不足之处提出改进方案。

本文开发了文本识别系统，其中文本识别系统前端包括Android端和网页端，Android端的功能包括上传图片之后得到文字结果反馈，调用本地摄像头拍摄照片得到反馈。网页端的功能只要包括上传图片，调用摄像头设备拍摄照片等手段上传照片并识别。

1.5本文设计目的以及组织结构

本设计的目的为研究深度学习算法相关内容，针对TextBoxes框架做出理论研究，并且设计出一个能够所见所得的手机app以及网页端，能够将上传的图片以及拍摄的图片进行文字检测，检测出文本所在的位置以及偏移量并且能够显示在图片之上。对于论文的目录和具体内容如下：

第一章绪论，本章节介绍了课题的研究背景，本课题主要研究为基于深度学习的OCR文字检测算法，然后叙述了OCR文字检测与识别的国内外现状包括深度学习的国内外现状，接下来叙述本课题的测试评价标准和本课题实验过过程中数据集来源和课题研究的目的以及设计内容。

第二章深度学习技术与OCR技术，介绍了OCR技术的概念，传统OCR技术的实现方式[4]与基于深度学习的OCR技术两者之间的区别，除此之外通过深度学习算法与传统算法的对比分析相对于传统OCR技术而言基于深度学习算法的文字检测算法的优势所在，同时也会基于传统OCR识别技术分析相对于深度学习算法而言传统OCR文字检测算法所存在的不足之处。。

第三章基于TEXTBOXES下的自然场景文本检测，本章内容主要包括对于TEXTBOXES模型的介绍，TEXTBOXES的组成部分网络结构算法分析，同时针对此算法进行实验，得出对比数据，对于非极大抑制算法的不足之处提出疑问并提出解决办法。

第四章系统介绍，对于本设计所设计的系统进行从功能方面，结构方面接口方面的叙述，详细讲述设计过程以及实现方式，并且对本系统的部署于测试进行介绍。

第五章系统实现包括介绍本系统使用的核心模型服务器系统的实现方式，客户端系统的具体实现方式，并且会将成品图贴出，展示成果。

第六章总结与展望，本章节对本设计包括检测模型与系统两部分进行评价，主要从不足之处和设想改进两方面着手，从客观角度分析模型的优点和需要改进之处，并根据需要改进之处设想改进方案，但是并未真正在本设计中实现。对于应用系统的展望包括对于系统目前所作功能从客观角度的分析评价和从功能角度希望丰富的功能但并未真正实现。

最后包括致谢，感谢为我耐心解答疑问的老师，以及帮助我完成课题的同学。

1.6本章小结

本章为绪论部分，主要从研究背景以及意义，国内外的研究现状数据评价标准和实验数据集准备等方面概述了本课题的深度学习模型相关信息，同时介绍了本课题将实现的识别系统，本课题实现的系统包括Android前端与网页前端，他们可以共用同一个后台。除此之外对于本论文的总体结构进行了概述。

2 深度学习技术与OCR技术

本章主要讲述了目前所常见的OCR技术，以及基于深度学习方式的文字识别算法。在介绍深度学习算法的章节主要介绍了卷积神经网络的基础概念。

2.1光学文字识别技术简介

OCR（Optical Charater Recognition）即光学文字识别技术，是Tausheck于1929年取得的专利[5]技术，如今OCR由于其应用的广泛性，使其在各个领域都取得了广泛的应用。并且OCR技术在计算机图形学，机器学习，数字图像处理等与人工智能领域中的模式识别与深度学习中一直是非常重要的话题。经过了几十年的发展，无论是国内还是国外都有较为成熟的OCR产品，各大物联网公司也有较为成熟的OCR模型并且提供接口API使其他用户可以使用OCR产品。

2.2常见OCR技术

目前比较常见的OCR识别方式有以下几种：基于统计特征字符的识别技术与基于结构字符的识别技术和基于深度学习的字符识别技术。OCR一般的处理过程如下图所示：

|  |
| --- |
|  |
| 图3 常见OCR处理流程 |

2.2.1统计特征字符识别技术

统计特征字符识别的特征向量一般是那些相对稳定的，共有的并且有较好的分类性能的特征。常用的统计特征包括字符二位平面特征、字符在水平或者垂直方向的投影直方图特征、矩特征和字符经过频域变换或者其他形变后的特征等[16]。庞大的字符统计特征在经过学习的过程、提取的过程、分类的过程之后形成关于字符的原型信息，构成文字识别的基础模板信息。这些算法根据其不足，也不断有学者提出新的思路。

2.2.2结构字符识别技术

在实际生活中，更多的是解决相近字符的识别和与印刷字体相差很大的手写字符的识别。基于结构字符识别的技术由此发展而来。这种技术第一步是提取字符的结构，这些结构又被称作为被提取字符的模式、部件、基元[16]。所被提取的基元按照某种顺序排列起来就构成了字符。识别过程是建立在提取基元的过程之上，利用词法分析、图匹配和树匹配和知识推理的方法分析字符结构。与统计识别方法相比，字符结构的识别技术的应用场景更偏向于识别字符形变较大的或者字型相近的字符，例如识别手写体是识别。但是也由于要对字符结构进行精确的标数，所以结构字符识别技术要消耗大量的存储和计算机资源，并且算法在实现上更加的复杂，识别速度也比较慢。

2.2.3基于深度学习的OCR技术

基于深度学习的OCR技术是使用深度学习技术力图训练出高效的识别方式。虽然现在传统的OCR识别技术越来越成熟，然而在一些情况之下仍然存在着巨大的问题。相比于文档中的字符，街景中的背景更加复杂，场景的字体的颜色，排列以及纹理等都会成为巨大的敢让影响OCR识别的结果。除此之外由于图片的拍摄角度，拍摄光线等问题也会使文字产生不同程度的形变。因此街景文字识别存在着文字本身以及文字背景的干扰，其识别难度远远高于普通的文档中的文字识别。随着深度学习算法的兴起,计算机视觉与模式识别领域的多项重要问题都取得了重大的突破，深度学习技术在物体检测，机器翻译，语言翻译等多项领域都取得了重大突破。其中深度学习模型的核心是深度学习网络，它是一种有十分强大的学习能力的机器模型。深层神经网络的本质是一个拟合函数，这种拟合函数具有高度高度非线性、高度参数化的通用的特点。这种神经网络在经过大量的训练后可以得到良好的预测能力。深度神经网络之所以能够达到如此良好的效果主要因为深度学习能够主要学习特征，并且具有强大的学习能力，并且这种能力可以通过自然场景图片的大量训练中而来。相比于其他识别方式来说，利用神经网络可以讲识别模式设计为端到端的形式，即只需要输入图片和输入图片的数据的标注即刻，优化算法能够自动学习所有的网络参数，并不需要手工设计和调整，从而降低了系统开发的人力成本。

|  |  |
| --- | --- |
|  |  |
| 图4 文档文字识别与街景文字识别 | |

2.3卷积神经网络

卷积神经网络[6](Convolutional neural network, CNN)是一种用来处理具有类似网格结构的数据神经网络，例如图像数据可以看作为二位像素网格等。它是一种经典的前馈神经网络[[1]](#footnote-1)，主要是受到生物学中的神经科学启发而产生的。卷积神经网络的名字表明了卷积神经网络使用了卷积这种运算，卷积网络中的主要运算时卷积运算。卷积神经网络也包括正向传播和反向传播的过程。正向传播是指从输入层输入数据，经过若干层的卷积运算再从输出层输出的过程。反向传播实际上是使用误差后向传播(Error Back Propagation, EBP)算法和梯度下降算法重新对网络层次调整权重。通过比较输出的数据结果和标注的数据结果从而计算得到误差。实际上调整参数和权重就是训练学习的主要过程。

卷积神经网络的结构一般由卷积层(Convolution)、下采样层(Pooling)、全连接层(Fully-Connection)构成。下采样层一般再卷积层之后，与卷积层交替出现，最后连接的是全连接层。在卷积神经网络中通常采用局部连接、权值共享和空间或时间相关的下采样方法从而获得很好的平移、缩放和扭曲不变性，使得提取的特征更具有区分性。

2.3.1卷积运算

通常情况下卷积运算是指对两个实变函数[[2]](#footnote-2)的一种数学运算。考虑以下情形，一个物体的位置关于时间都有唯一的位置输出x（t），其中位置输出通过设备测量得到。其中x代表位置，t代表时间。这意味着在任何时间都能得知物体在t时的位置。在物体远离的过程中，然而测量设备在距离物体越远的情况下越容易收到干扰，导致直接通过输出无法得到精确的位置。为了能够得物体位置的精确估计则需要对测量结果进行平均。显然时间越相近的测量结果越精确，所以采用一种加权平均的方法对于最近的测量结果赋予更高的权重。考虑采用加权函数w(a)来实现。其中a表示的当前测量时间与当时结果的测量时间的时间间隔。因此可以得到估计函数如下：

也可以被表示成:

其中w是一个概率密度函数，\*表示卷积运算。卷积的第一个参数通常叫做输入即上面例子中的函数x，函数w通常被叫做核函数。

|  |
| --- |
|  |
| 图5 二维卷积运算 |

在实际中我们的计算机在处理数据时，数据会被离散化，上述例子中的要求测量机器在每个时刻都将位置信息反馈时不切实际的。因此就可以将卷积离散化。图5为二维离散数据经过卷积运算的过程，其中卷积核为（x,y,z,w），经过卷积运算之后的输出为a~i这9个数据。其中

其他的运算以此类推。

2.3.2 全链接与局部链接

卷积神经网络中局部链接是一个比较重要的特征。传统的神经网络中往往采用的是全链接的方式。然而图像本身的统计特性决定了其一个区域的统计特性极大可能与其他区域的统计特性相同，所以可以利用这一特征不适用全连接的方式而是使用局部链接的方式能够以较少的代价更好的抽象出图像的局部特征。局部链接能够明显的减少网络中层与层的链接数，减少网络运算的复杂度。假设图像的尺寸是200\*200像素，如果使用全连接假设网络中有106个隐藏单元，那么使用全连接则需要进行200\*200\*106=4\*1010个链接；假设局部链接中网络使用10\*10的像素，那么一个局部连接只需要10\*10\*106=4\*108个链接。可见采用局部链接可以极大的缓解计算机压力。并且当图片很大时如果采用全连接的方式无疑会给计算机造成很大的压力。

|  |  |
| --- | --- |
|  |  |
| 图6 全连接方式 | 图7 局部链接方式 |

2.3.3权值共享

权值共享通过使模型共享参数以达到减少参数的目的。在传统的神经网络中，当计算一层的输出时权重矩阵的每个元素仅仅使用一次之后就不再使用。而使用的权值共享机制之后只需要使用一个参数集合即可，这虽然没有改变向前传播的效率但是减少了参数的存储。

|  |
| --- |
|  |
| 图8 不使用权值共享[[3]](#footnote-3) |

|  |
| --- |
|  |
| 图9 使用权值共享与局部链接 |

2.4目标检测网络介绍

在计算机视觉领域中，目标检测[7]几乎是十分基础的任务。目标检测的主要任务是在给定的图片中精确的找出物体所在的位置,并标注出物体的类别。目标检测网络既要识别出物体的位置又要识别出物体是什么。然而物体的尺寸，图片拍摄的角度都可能为识别在造成很大的困难。自从在ILSVRC 2012挑战赛，AlexNext获得了冠军之后，使用CNN进行分类成为了主流。目前结合深度学习与神经网络的目标检测算法主要包括R-CNN[8],Fast R-CNN[9]，YOLO[10]，SSD[11]等算法。

2.4.1基于滑动窗口的目标检测算法

基于滑动窗口的目标检测算法的思路大概是使用不同的检测框，在图中依次滑动然后将截取的图片通过CNN网络识别并给出评分。最终如果评分高于阀值那么可以认定这是所需要的目标。这种检测算法需要消耗大量的计算机资源。并且识别速度很慢。

2.4.2深度学习分类算法

只是基于滑动窗口的目标检测算法显然不是一个好的选择，那么可以考虑预先找出图中目标可能出现的位置，利用图像中的纹理，边缘等信息可以简化选取滑动窗口的过程。主要步骤是输入测试图像之后利用搜索算法将图片中提取可能出现物体位置的候选框

2.5本章小结

本章内容详细的介绍了深度学习技术与OCR技术，介绍了几种传统的OCR技术和使用深度学习技术的OCR技术做对比从而显示出使用深度学习方式的OCR技术的优势。本文所研究的内容是基于深度学习的文字检测，如果将文字看作一个目标那么检测文字实际上就是目标检测网络的一类，检测对象是文字。在目标检测算法中主要使用卷积神经网络作为基础，其他算法几乎都是基于卷积神经网络然后再做扩展。

3 基于TextBoxes的自然场景文本检测模型

TextBoxes模型[12]是一个文本检测算法模型，基于文本目标的检测以往可以分为三大类包括基于字符的检测，基于单词的检测和基于文本行的检测。其中基于字符的目标检测是首先检测出单个字符，然后将字符组成单词。基于单词的检测则是直接检测出单词与一般对象检测算法一致。基于文本行的检测则是检测文本行之后然后再将单词进行拆分。TextBoxes是基于单词文本检测是一个端到端的网络，即只需要输入则可以得到输出。TextBoxes包含两个部分，一部分是目标检测网络，TextBoxes的目标检测网络来源于目标检测网络SSD的一个发展。剩下的部分是当TextBoxes检测出多个文本框时则需要采用非极大抑制算法对多个文本框进行抑制。

|  |
| --- |
|  |
| 图10 textboxes整体结构[12] |

3.1TextBoxes中的神经网络

VGG-16[13]架构是Simonyan和zyserman2014年提出是架构，TextBoxes中的网络结构则继承了这个架构。VGG架构主要包括有卷积层与池化层和全连接层还有预测层。

|  |
| --- |
|  |
| 图11 VGG16 |

TextBoxes模型相对于VGG16来说保留了其中的从conv1\_1到conv4\_3，并将VGG16中的最后两个全链接层通过下采样的方式转换成卷积层。除此之外还增加了一些额外的的卷积层和池化层。TextBoxes中有多个text-box 输出层，这些输出在一些卷积层之后。在这些输出之后会通过nms作为抑制。在TextBoxes中每一层都预测一个盒子的位置和边界，同时还输出置信度。除此之外还预测位置的偏移，从而得到x，y，w，h。由于文本具有比较大的纵横比,因此TextBoxse网络[14]中选用长卷积核。

不同于SSD算法在TextBoxes中为默认框定义了6个纵横比，分别为1，2，3，5，7和10。然是这种长的纵横比使得默认框在水平方向上比较稠密，而在垂直方向上比较稀疏从而导致匹配不佳。因此为每个盒子都设置了纵横比。此外区别于SSD网络使用3\*3的滤波器， TextBoxes采用1\*5的滤波器，更适合宽高比比较大的单词。

3.2TextBoxes中的NMS算法

在图9中当在TextBoxes Layer之后还有需要进行非极大抑制操作。由于TextBoxes的每一个 boxe-layer都会输出一个盒子，因此需要采用非极大抑制算法抑制非最优结果，将最优结果找出。该算法在物体检测算法中应用广泛，主要通过检测结果的置信度与交叠作为标准实现去除冗余的识别框。

|  |
| --- |
|  |
| 图12 非极大抑制算法流程图 |

在TextBoxes中非极大抑制算法的思想是首先在所有被识别出来的文本框中选择置信度最高的文本框，将选择的文本框标记为a与其他的未被标记的文本框计算面积的重叠率，或者叫做面积交叠率（IOU）。将计算出的交叠率与阀值进行比较。如果交叠率大于所设置的阀值，那么认为检测到的是同一物体同时将这一文本框标记为被处理过，假设标记为b，如果交叠率小于所设置的阀值那么则认为两者所检测到的是不同的物体，此时这个文本框不做标记并且等待下一次被抑制。重复上述的过程，直到所有的文本开框都已经被标记完成。其中所有标记为a的文本框则是去除冗余之后选择的最优结果。在非极大抑制算法中阀值并不是一个统一的标准是可以人为设置的，并且阀值的高低对非极大抑制算法也具有很大的影响。如果阀值设置过高那么会出现冗余去除不够的结果，如果阀值设置的过低则会出现抑制过头的结果。图12展示了非极大抑制算法中交叠率的示意图，其中交叠率的公式可以表示为:

|  |
| --- |
|  |
| 图13 交叠率计算示意图 |

3.2本章小结

本章主要介绍了TextBoxes算法的相关网络结构以及用到的主要算法。本文中关于TextBoxes的介绍主要参考发表于AAAI上的论文。TextBoxes是一个文本检测算法，其网络结构与目标检测网络SSD算法类似。TextBoxes主要由两部分组成，一部分是检测网络，另一部分是非极大抑制算法。本章节针对这两部分首先介绍了TextBoxes的神经网络结构，分析了TextBoxes网络与SSD网络的不同之处然后详细讲述了非极大抑制算法的算法原理。

4 对于TextBoxes模型的拓展

4.1对于读取图片方式的改进

TextBoxes在GitHub[[4]](#footnote-4)上的代码中的demo代码中，仅针对一张图识别并且地址为写死的地址，本设计在研究此模型时针对这一缺陷进行了改进。改进后的demo可以从过参数的形式读取图片存放的目录地址和图片存放的目录地址，目录中可以包含子文件夹。当图片被检测完毕之后将图片按照原来的存储节后放入目的地址即可。

以上改进主要根据目录的树存储的形式特点，以队列作为数据结构通过广度优先搜索来实现文件读取。首先将识别程序封装成函数，然后再循环调用则可以达到改进的效果。具体步骤如下：

（1）运行程序时，携带识别地址参数，如果不携带参数则使用程序中默认的参数。

（2）维护一个目录队列初始放入图片的初始目录，在数据集目录中逐个读取目录中的文件和目录，如果读取的是文件那么将图片地址作为输入调用识别程序。识别完成之后根据图片地址和输出目录地址将图片放入相应的目录。

（3）如果读取的是目录，那么将目录地址存放人目录队列中。

（4）当读取完本目录中的所有文件和目录时，将本目录地址从对于维护的队列做出队操作。

（5）判断队列中是否还有目录，如果还有目录那么读取这个目录并且重复以上操作，如果对列为空那么说明已经读取完成，程序运行完毕。

|  |
| --- |
|  |
| 图14 Linux中目录结构示例图 |

图13展示了在Linux系统中在目录dataset/中存在两个文件夹和一个图片文件并且在set1中包含一个目录dir和一个图片img2，dataset/set2中包含若干张图片。

使用本设计的文件读取算法图片的读取顺序为img1.png，set1/img2.png，set2/img2.png，以及set2中其他图片。

|  |
| --- |
|  |
| 图15 流程图 |

上述描述的读取过程可以使用伪代码的形式体现：

Queue.push(origin\_path);

Path = origin\_path

While Queue.size() do

Read element in path:

If element is dictionary:

Queue.push(elment.path)

else element is image

do textboxes and output file

Queue.pop()

其中Queue代表队列，push操作代表入队操作，pop操作代表出队操作。

4.2 TextBoxes中模型识别评价标准

由于目标检测与文字识别不同，目标检测所得到的最终结果是以偏移量的形式展示，通过偏移量可以计算出识别框的坐标位置。一个四边形的要表示顶点坐标需要四个二维坐标表示，一共是八个数据，但是实际上由于长方形关于中心点中心对称，所以只需要两组坐标就可以表示出来。TextBoxes将这两组数据称之为Xmin，Xmax，Ymin，Ymax。

|  |
| --- |
|  |
| 图16 xmin，xmax，ymin，ymax示意图 |

上述数据可以对通过对TextBoxes的检测程序进行改造而得到。由于检测得出的数据是坐标形式，标准形式也是坐标形式且仅仅含有坐标，那么确定模型检测而出的数据与真实数据的对应关系成了评价的难点。一种方案是根据上述数据而计算交叠图形的坐标，计算公式如下：

采取的方案为以标准数据为基准，向模型检测所得出的数据中寻找交叠率最高的一组数据，当一组数据被匹配后，标记不被其他数据匹配。若这组数据超过给定的阈值，那么视为对应数据，并且计算交叠率为准确度。如果，那么则视为两个图形没有交叠。将标准数据为基准能够体现出当图片中遗漏检测的情况，因为当一组数据与模型输出数据对比时都不能找到对应数据的时候，那么就认为这组数据被遗失，模型并未检测到这组数组。但是这种对比方式无法体现出模型识别中出现的识别错误的情况。因此出了将标准数据作为基准向模型中对比之外，还需要考虑模型输出数据与标准数据一一对比之后，模型数据中剩余的向量。那么模型检测的准确率就可以综合这三个因素，其中准确率可以占重要比重，而多检测和少检测的情况中少检测所占的比重大于多检测所占的比重。

|  |
| --- |
|  |
| 图17 多检测现象 |

4.3对TextBoxes中MNS的改进

在TextBoxes模型之中，由于存在多个text-layer，并且每一层都会由一个相应的输出，所以如果不做非极大抑制那么对于一个单词会输出多个文本框。非极大抑制的作用就是去除冗余。但是非极大抑制算法仍然会出现抑制错误的情况。

|  |  |
| --- | --- |
|  |  |
| 图18 将被识别的部分作为结果 | |

比如一个比较长的单词，模型在识别这个单词的同时还由可能会对单词的其中一部分做出检测，那么如果被检测到的这一部分与整体单词做IOU，如果IOU低于阀值，那么被识别出的不完整的单词就会被视作一个新的单词而不被抑制。但是在实际场景中，单词与单词之间几乎不会出现重叠的情况。因为在实际场景中尤其是街景中文字存在的目的往往是为了能够有效的传达信息，因此文字重叠的可能性小之又小。因此对于这一问题，可以采用以下方案进行改进。

4.3.1通过降低阀值的方式

由于在街景图片中文字几乎都是分开散布的因此可以将阀值设置的尽量低，即只要两个检测文本框之间有微小的相交那么就认为这是同一个文本框并且进行抑制。通过阅读TextBoxes代码可知TextBoxes中设置的阀值为0.3，那么尝试将阀值以步长0.1下降，分别再进行实验测试。

|  |
| --- |
|  |
| 图19 TextBoxes部分代码 |

使用以下图片进行测试，并且使用前面章节所提到评价算法计算正确率之后所得到的结果，可以看出通过降低阀值的方式可以使检测正确率得到提升。

表4-1 不同nms阀值下数据集检测正确率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nms阀值 | 0 | 0.1 | 0.2 | 0.3 |
| 检测正确率 | 70% | 80.2% | 80.3% | 79.9% |

4.3.2 对NMS算法进行修改

除了通过降低阀值的方法，还可以通过改进NMS算法的方式来实现对于MNS的优化，观察图17中的两个错误检测例子，可以发现被识别错误的文本框几乎包含在识别正确的文本框之中，那么考虑另外一个因素即被包含的情况。本文将这个指标乘为包含率，即对于文本框A和文本框B，A与B的交集比上A本身的面积那么就是A相对于B的包含率称之为(ICR，Inclusion Rate)，公式如下：

则可以将包含率作为抑制的标准在判断交叠面积是否大于阀值时同时判断包含率是否大于阀值。具体步骤如下：首先参照原本的NMS算法，在判断一个文本框不需要被抑制时，即文本框A与文本框B的交叠率小于阀值，此时还要进行判断两者的包含率，判断文本框B的包含率。即计算A与B相交的面积比上B的面积，如果此时值接近于1或者高于所设置阀值，那么就认为B是被A包含着是需要被抑制的文本框。再与上述相同的测试集的情况下仍然使用前面章节所提到的评价标准对比加入判断包含率与原本算法的识别准确率。

表4-2 加入包含率与使用原算法的对比

|  |  |  |
| --- | --- | --- |
| 数据集 | 使用改进算法（阀值0.3） | 使用NMS（阀值0.3） |
| Icdar13 | 79.9% | 81.2% |

4.4本章小结

本章内容主要针对TextBoxes在GitHub上的源码进行改进，虽然都是微小的改进但仍然能为识别过程提供便利。本章首先对TextBoxes模型的demo程序的文件读取方式进行改进，然后对于测试结果的判断提供了评判的方法，另外对于使用TextBoxes模型而其检测出结果仍然存在问题，通过分析问题产生的原因提出了初步解决方案。

5 系统介绍与实现

5.1系统需求分析

5.1.1 功能性需求分析

本系统包含的功能简单。在本基于深度学习的OCR系统之后，用户可以选择拍照上传图片然后进行识别，同样也可以从系统相册中选择图片进行识别。图片上传之后可以将识别完成的图片展示到界面中。本系统包含Android端应用和网页端应用。图片传回到Android前端之后要求可以通过点击方式识别检测后的图片。

图片上传模块，图片上传包括从相册上传和拍照上传两个部分，其中选择图片之后支持图片预览功能。

图片文字检测模块，图片文字检测主要由系统后端完成，系统将图片上传至服务端服务端调用TextBoxes模块，使用TextBoxes检测出图中的文字所在位置并将结果反馈至用户。

图片结果显示模块，图片检测完成后将检测完成的图片放到页面上，分上下两块对比显示。

以上需求同样适用于Android端和网页端，。

|  |
| --- |
|  |
| 图20 系统用例图 |

5.1.2 非功能性需求分析

要求本系统运行流畅，界面友好。其中检测效率不能低于1min每张。

5.2系统详细设计

本章主要介绍图片检测系统的详细设计模块包括接口设计以及UI设计。本系统采用前后端分离的形式进行开发。

5.2.1前端详细设计

系统前端包括Android前端与网页前端。其中Android前端接口包括上传图片接口。上传图片调用。其中Android端的前端UI图设计如下。

|  |  |
| --- | --- |
|  |  |
| 图21 界面1 | 图22 界面2 |

前端界面十分简洁仅有一个界面。分为上下两个部分，其中点击上部分选择图片之后会出现两个选择。分别为从手机相册选择和使用相机拍摄。点击选择拍摄可以拍摄图片后进行文字检测，点击从相册中选择之后可以从手机相册中选择图片之后再进行识别。其中网页端设计与Android端设计大致相同，除此之外网页端前端支持图片拖拽上传，图片批量上传等功能，识别时为了能够即使反馈识别信息，且由于过程持续时间过长所以采用websocket，当识别完成时主动向前端发送消息。

5.2.2 服务端设计

服务端使用Python的web的Flask框架[15]，调用TextBxoes模块。服务端为前端提供识别功能，服务端能够接收图片并将图片地址作为参数传给TextBoxes模块程序，TextBoxes程序提供识别功能，识别后返回图片存放路径，服务端通过图片存放路径从服务器中取出图片后以文件地址的形式返还给前端，因为结果放在了nginx图片服务器中，因此返回图片地址结果即可以通过网络路径读取到图片。

5.3系统实现

本章节将详细讲述本系统的实现过程，包括实现前端所用到的技术所使用的框架以及在实施过程中遇到的困难等。本章节将以前后端分开的形式以及总体的形式讲述前后端实现的过程以及前后端的通信问题。

|  |
| --- |
|  |
| 图23 时序图 |

5.3.1 Android前端实现

|  |
| --- |
|  |
| 图24 uni-app 功能框架图[[5]](#footnote-5) |

Android实现采用的框架是uni-app，uni-app是一个使用了Vue.js开发的跨平台应用前端框架，所谓跨平台意味着本系统在iOS，Android系统甚至H5中都可以很好的编译运行。同时uni-app还支持微信小程序，支付宝小程序，百度小程序，头条小程序等。使用uni-app的优势包括跨端数量更多,一套代码可以支持iOS，Android，小程序，H5d等多个平台；在跨平台的同时通过条件编译和平台特有的API调用可以优雅的为所需要的平台编写个性化的代码；基于通用的前端技术学习成本低，采用vue语法+微信小程序api无需额外的学习成本。

前端在采用这种技术的情况下仍然有需要解决的问题。在前后端交互的过程中会出现跨域问题。因为采用的uni-app实际上是基于Vue.js的前端框架即页面实际上可以认为是运行在浏览器上的因此要解决跨域问题。跨域问题是由于浏览器的同源策略而导致的，其策略主要内容为限制两个来自于当不同源的资源需要进行交互时，这种机制能够隔离潜在恶意文件。在跨域资源共享问题中CORS起到重要的作用，其全称为“跨域资源共享”。解决同源问题有很多方案，本设计使用的方案是在后端服务器中设置回复体头字段。

除了解决跨域问之外还要解决编码问题，服务端返还给客户端的路径是经过Unicode编码的，而服务端若想成功读取文件则需要经过解码。

5.3.2 网页端前端实现

网页端实现采用react框架实现，组件库使用ant-design的组件库，为开发节约成本，使能在短时间内开发出一套界面优美的前端界面。

根据系统需求，使用ant-design的图片上传组件，在图片上传之前判断上传文件是否为图片格式。除此之外前端调用系统摄像头，拍摄图片后调用后端接口识别。网页版前端可采取批量上传的模式，再从批量上上传的图片库中选择图片进行识别。

5.3.3 服务端实现

|  |
| --- |
|  |
| 图23 后端功能介绍 |

本系统的服务端采用Python开发，在本系统中实现Web端采用Flask框架。Flask框架使一个轻量级的Web应用框架。其优点为简单易学。

Flask框架的作者为Armin Ronacher，其自带开发应用服务器和debugger，采用Resutful风格。为了调用文字检测模块，将TextBoxes打包封装成Python模块，并在后端中引用。将TextBoxes改装成可以将图片当作参数读取的方式进行，服务端返回数据时返回图片在nginx图片服务器中。

本设计采用的nginx为最轻量级的nginx-light，符合本系统简洁的风格。为了满足Android端与网页端两端，由于网页端与Android端存在不同的需求，因此后端针对不同的前端可能会书写不同的接口。

5.4成果展示

本设计Android端展示，图片上传功能与识别功能展示

|  |  |
| --- | --- |
| C:\Users\hp\Documents\Tencent Files\1144042776\FileRecv\MobileFile\Cache_-5d93ff1ae13c5448..jpg |  |
| 图24 识别功能 | 图25 结果预览 |

系统网页端展示

|  |
| --- |
|  |
| 图 26 上传图片界面 |

|  |
| --- |
|  |
| 图 27 图片预览样式 |

支持多文件上传

|  |
| --- |
|  |
| 图 27 图片多文件上传 |

5.5本章小结

本章节分为五个小结，主要内容关于本设计的文字识别系统的介绍与实现。其中前第一节介绍了本系统的需求功能，即系统功能介绍。第二节为详细设计，主要包括前端UI设计以及后端接口设计。后面两小节分别为具体系统实现与系统测试。系统实现章节主要详细介绍了实现系统时所使用的工具以及系统实现时遇见的难点。

# 6 总结与展望

6.1设计总结

本设计主要研究了基于深度学习的OCR算法，首先介绍了常见OCR的使用场景，并且介绍了几种常用的OCR算法。介绍几种常用的OCR算法的主要目的在于将传统的OCR算法与基于深度学习的OCR算法做对比总而凸显出使用深度学习算法进行OCR识别的优势。除了上述所标数之外，本设计还详细讲述了在OCR系统中，使用卷积神经网络的具体原理。包括卷积神经网络的基础概念。使用深度学习算法进行OCR文字检测其优势主要在于不需要人工提取特征，通过神经网络学习即可，并且在比较复杂的背景之下仍然能够展现出良好的识别效率。

其次本系统针对深度学习模型TextBoxes模型进行介绍主要从TextBoxes的算法模型以及非极大抑制算法方面介绍基于深度学习的OCR文字检测，详细介绍了TextBoxes使用的深度学习算法以及非极大抑制算法。并且通过反复实验发现了TextBoxes模型存在一些不足之处针对这些不足之处提出了个人见解以及改进。

本设计还包括一个拍照识别的系统，本系统能够通过拍照和上传图片的方式调用上述TextBoxes模型进行图片文字检测。本设计所设计的系统前端包括App系统以及网页端系统。后端采用使用Python语言。其中前端采用React框架采用Ant Design组件库，不仅简化了前端系统开发也使前端界面具有较好的外观。

6.2模型展望

本设计采用的是搭建在Caffe上的TextBoxes模型，此模型在ICDAR13数据集中检测率可以达到80%，但仍然有不足之处。

|  |
| --- |
|  |
| 图28 中文检测结果 |

（1）本模型对于中文识别支持不友好。本模型基于的是单词识别所以在设计其网络模型时采用的是较大的纵横比。而中文的展现方式与英文有很大的不同，在自然场景中中文的词组虽然也与英文类似有较好的纵横比但中文的一段话中字与字之间没有较大的空隙或者说没有较为明显的分隔，因此本模型的检测算法不能较好的对中文检测。

如图28所示，在中文文字情境下如高速公路牌上的文字，并没有被完美检测。以上说法只是其中之一，实际上最主要的原因可能是因为在此模型的数据集上并没有对中文有良好的支持和训练，因此要解决此问题则需要大量的中文街景图片进行训练。

（2）由于模型的default box 在检测过程中只考虑了上下左右的偏移而没有考虑到旋转，因此文字检测都是以水平正方形的形式展现。但是在自然场景中不乏旋转排列的情况此时TextBoxes模型就无法很好的检测。

|  |
| --- |
|  |
| 图29 对于弯曲文本仍然使用水平框检测 |

以上两点可以通过修改模型以及丰富训练集再次训练的方式进行改进。

6.3应用系统展望

本设计包含的应用系统为两套，即APP端以及网页端，这两套应用使用同一个后台。目前此两套系统的展望包括丰富系统功能。目前APP端仅仅支持拍照检测的功能。在实习应用中仅仅检测文字实际上并没有十分大的实际用处，在检测到文字之后的识别与翻译才是使系统具有使用价值的关键。因此本应用系统在调用以上模型之后再使用CRNN的文字识别系统，调用开源API将文字进行翻译，最终再显示再用户界面之上。

对于网页端设计目前仅仅支持上传检测功能，为了丰富网页端功能将提供训练功能，即用户上传图片以及标注，系统调用训练程序对模型进行训练。

# 参考文献

[1] Weiying Xie,Yunsong Li,Xiuping Jia. Deep convolutional networks with residual learning for accurate spectral-spatial denoising[J]. Neurocomputing,2018.

[2] Jing Zhang,Long He,Manoj Karkee,Qin Zhang,Xin Zhang,Zongmei Gao. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN)[J]. Computers and Electronics in Agriculture,2018,155.

[3] Si-qi ZHAO,Peng LIU,Jie YANG,Song-bin LI. A Fast Vehicle Detection Method for Road Monitoring[P]. 2nd International Conference on Computer, Mechatronics and Electronic Engineering (CMEE 2017),2017.

[4] 宋毅,崔平远,居鹤华.一种图像匹配中SSD和NCC算法的改进[J].计算机工程与应用,2006(02):42-44.

[5] 孙羽菲. 低质量文本图像OCR技术的研究[D].中国科学院研究生院（计算技术研究所）,2005.

[6] 荆涛,王仲.光学字符识别技术与展望[J].计算机工程,2003(02):1-2+80.

[7] Olufunminiyi, O.E.,Onifade,Olufade, F.W.. Image-based distance detection for non-zooming stationary camera towards detecting intrusive target[P]. Computer Applications and Information Systems (WCCAIS), 2014 World Congress on,2014.

[8] Laiho, M., Paasio, A., Halonen, K.. Improved cell core for a mixed-mode polynomial CNN[P]. Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on,2004.

[9] 周晓彦,王珂,李凌燕.基于深度学习的目标检测算法综述[J].电子测量技术,2017,40(11):89-93.

[10] Brad Walker.Comparison of the Birds Point-New Madrid Floodway, Mississippi River and the Yolo Bypass, Sacramento River[J].Journal of Earth Science,2016,27(01):47-54.

[11] 杨俊. 基于卷积神经网络的目标检测研究[D].兰州理工大学,2018.

[12] Liao M , Shi B , Bai X , et al. TextBoxes: A Fast Text Detector with a Single Deep Neural Network[J]. 2016.

[13] Paul R. Classifying cooking object's state using a tuned VGG convolutional neural network[J]. 2018.

[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 & Machine Intelligence, 2015, 39(11):2298-2304.

[15] 马旭,王淑丽.基于Flask框架的展示型网站的设计与实现[J].数字技术与应用,2018,36(11):137-138.

[16]荆涛,王仲.光学字符识别技术与展望[J].计算机工程,2003(02):1-2+80.

# 致 谢

本设计的顺利完成离不开老师和同学们的帮助。在完成本设计的过程中存在着很多困难，但也同时在老师的耐心指导之下才得以克服，在同学的帮助下我能够顺利的完成本设计的实验，并在实验过程中通过老师指导的方向丰富本设计因此收获良多。在此，我对指导我完成毕业设计论文的老师以及帮助我的同学们致以最衷心的感谢。

于此同时还要感谢学校图书馆提供的大量书籍以及学校提供的资料网站，在完成本设计过程中需要进行大量知识理论的学习和补充，学校提供的便利使我在完成本设计的过程中尤其在资料查阅方面十分方便。

最后要感谢本文所引用的文献的专家学者们，他们的著作在我完成毕业设计中起到了重要的参考作用，也因为站在他们的肩膀之上，本设计才能顺利完成。

衷心的感谢所有人的帮助。

1. 前馈神经网络：每个神经元只与前一层神经元相连，接收前一层的输出并传输给下一层，各层间没有反馈。 [↑](#footnote-ref-1)
2. 实变函数：以实数作为变量的函数。 [↑](#footnote-ref-2)
3. 红色虚线箭头表示某一特殊参数。 [↑](#footnote-ref-3)
4. TextBoxes在 GitHub地址: https://github.com/MhLiao/TextBoxes [↑](#footnote-ref-4)
5. 图片引用自：https://uniapp.dcloud.io/ [↑](#footnote-ref-5)