目录

[第一章 绪论 1](#_Toc476000699)

[1.1 文本识别的研究背景及意义 1](#_Toc476000700)

[1.2 自然场景文本识别难点分析 2](#_Toc476000701)

[1.3 自然场景文本识别的研究现状 3](#_Toc476000702)

[1.3.1 传统文字识别研究 3](#_Toc476000703)

[1.3.2 基于深度学习的文字识别 4](#_Toc476000704)

[1.3.3 端到端研究方法的兴起 4](#_Toc476000705)

[1.4 本文的研究目标与内容安排 4](#_Toc476000706)

[第二章 神经网络 6](#_Toc476000707)

[2.1 传统人工神经网络 6](#_Toc476000708)

[2.1.1 神经元 6](#_Toc476000709)

[2.1.2 常用激活函数 7](#_Toc476000710)

[2.1.3 前馈神经网络 8](#_Toc476000711)

[2.2 循环神经网络 10](#_Toc476000712)

[2.2.1 简单循环神经网络 10](#_Toc476000713)

[2.2.3 RNN反向传播算法 11](#_Toc476000714)

[2.2.4 LSTM 11](#_Toc476000715)

[2.3 卷积神经网络 12](#_Toc476000716)

[2.3.1 局部连接 13](#_Toc476000717)

[2.3.2 权值共享 13](#_Toc476000718)

[2.3.3 下采样 14](#_Toc476000719)

[第三章 基于RNN的文本识别方法 17](#_Toc476000720)

[3.1 问题分析 17](#_Toc476000721)

[3.2 数据集 17](#_Toc476000722)

[3.3 图像预处理 18](#_Toc476000723)

[3.4 方法概述 18](#_Toc476000724)

[3.4.1 CNN层提取特征 20](#_Toc476000725)

[3.5 Connectionist Temporal Classification 21](#_Toc476000726)

[第四章 自然场景文本识别实验与总结 22](#_Toc476000727)

[4.1 不同深度学习开发框架对比 22](#_Toc476000728)

[4.1.1 Caffe 22](#_Toc476000729)

[4.1.2 PyTorch 23](#_Toc476000730)

[4.1.3 MXNet 23](#_Toc476000731)

[4.1.4 TensorFlow 24](#_Toc476000732)

[4.2 基于GPU的训练 25](#_Toc476000733)

[4.3 Dropout 26](#_Toc476000734)

[4.4 实验环境 27](#_Toc476000735)

[4.4 实验结果 27](#_Toc476000736)

[第五章 全文总结与展望 30](#_Toc476000737)

[5.1 总结 30](#_Toc476000738)

[5.2 展望 30](#_Toc476000739)

[参考文献 31](#_Toc476000740)

# 第一章 绪论

## 1.1 文本识别的研究背景及意义

文字是人们重要的交流工具，如何让计算机来认识并处理人类的文字一直是计算机视觉与模式识别领域重要的课题。早在上个世纪初期光学字符识别（Optical Character Recognition, OCR）[1]技术就被提出，经过长期发展，OCR在工业领域的应用日臻成熟[3]。但是OCR技术要求文档图像具有清晰简单的背景，较高的分辨率，而自然场景中的图像往往没有鲜明的对比背景，光照变换非常大，分辨率较低等使得OCR技术在自然场景中几乎无法使用。

然而随着智能移动设备走入千家万户，在智能移动上直接理解并处理文字信息必然有着巨大的应用需求。相对于多媒体文档中的常见的亮度、色彩、形状等底层信息，文字能够直接传递高级语义信息。合理地利用多媒体文档上的文字信息，能够极大的推动以下领域的发展：

（1）图像与视频检索。移动互联网时代图像与视频文件越来越多，如何对海量的图像与视频进行检索是一个重要问题。基于内容的图像和视频检索[4][5]的研究目的是根据图像和视频的抽象信息，根据一定的相似性指标来进行检索。目前难点在于计算机对图像与视频的相似性判断是根据图像的底层视觉特征来进行的相似性判断。而人由于在生活中拥有大量的经验与知识，对图像的相似性判断是根据语义信息来进行的相似性判断，这就产生了人对图像所理解的“语义相似”与计算机理解的“视觉特征相似”的“语义鸿沟”问题。而根据图像或视频中所识别的文本信息，可以提供较为丰富的高层语义信息，进而可以根据这些语义信息来对图像或者视频进行检索，或者依据语义信息对图像或者视频进行分类。

（2）车牌识别。随着人们生活水平的提高，机动车数量也越来越多。高效的车辆管理不仅可以缓解交通压力，而且对公共安全有重要意义。车牌往往就是唯一确定车辆身份的标志。现在往往在重要路口都有摄像头监控过往车辆的车牌，记录车辆的行驶信息，可以对违反交通法规的车辆记录在案，或者对可疑车辆及时报警。总之，车牌识别系统[2] [6] [7]对车辆管理具有重要意义，是模式识别与计算机视觉领域重要课题之一。

（3）无人驾驶。随着传感器与自动导航技术的发展，自动驾驶汽车[8]的也开始走入人们的视线。2012年5月Google的第一辆自动驾驶汽车获得自动驾驶许可证。自动驾驶可以有效的辅助甚至代替人类操作机动车量，能够为人类节省不少的驾驶时间，并且可以极大的减少能源消耗，降低交通事故的发生，让乘坐车辆更加安全。高速行驶车辆在复杂路况下识别道路指示牌和建筑标志中的文字显得更加重要。

## 1.2 自然场景文本识别难点分析

近几十年来文字识别研究得到了空前的发展，一大批科研工作者致力于这方面的研究，但是目前仍然不存在一种在所有自然场景中适用的实际方法来识别文本。如图1-1所示，光照不均匀，分辨率过低，视角的多变性，拍摄设备的移动都会导致图像模糊，文字难以识别。传统应用于扫描文档的OCR引擎几乎无法正确的扫描识别出这样的文字。





图1-1 自然场景下的文本图像示例

自然场景下的文本图像与传统的扫描文档相比，具有以下的特点：

1. 图像背景更加复杂。扫描文档的图像背景往往为单一颜色，对文本识别影响较小。自然场景中的图像则可能包含车辆、行人、楼宇建筑、花草树木、广告牌与交通灯等与文字没有任何相似性的元素。同时，背景元素对文本的遮挡和相似色彩的背景或前景也加大了识别难度。背景的光照、颜色、亮度等不确定，变化毫无规律。因此自然场景文本识别需要适应复杂的场景，对算法鲁棒性提出更高的要求。

2. 字符畸变严重。扫描文档的图像中字体多为印刷体，几何形变较小。自然场景中文本图像多由相机拍摄，由于拍摄视角的多变、拍摄设备的差异、透视效果的不同都可能导致图像的畸变。自然场景文本图像为了增加艺术效果，也会增加人为扭曲，旋转，倒影等效果。这无疑增大了字符识别的难度。

3. 文档结构复杂。传统文档图像中文本往往布局固定，文字为水平或垂直分布。而自然场景图像中文本分布随意，字符可能出现在图像的任意位置。对于传统文本识别，往往需要首先对字符进行检测，确定文本的位置后提取文本信息再进行识别。

现在越来越多的研究将机器学习方面的算法引入到场景文本识别领域来，取得了很好的效果，但是距离实用化还有一段距离。为取得更好的识别效果，训练算法需要大量的专业领域知识来结合机器学习，对科研工作者提出较高的要求。

综上所述，随着智能移动终端设备以及移动互联网的逐渐普及，多媒体文档中文字识别会越来越重要，而目前的自然场景文字识别技术依然不能满足现实需要，所以展开对文字识别的研究显得非常迫切。

## 1.3 自然场景文本识别的研究现状

### 1.3.1 传统文字识别研究



##### 图 1-2 自然场景文本处理流程

传统的文本识别是通过对单个字符的检测然后对字符提取特征，将特征送入分类器训练以此得到合适的分类器，最终完成识别。如图所示，文字图像经过预处理后，检测到文本区域，提取出文本信息，之后提取人工设计的特征，送入分类器训练，经过大量训练后，得到识别结果。经典算法例如，Rodriguez[43]等人使用支持向量机（support vector machine, SVM）结合Fisher[48]向量处理的算法，通过提取Fisher特征，送入SVM训练来建模图像与文本的关系，最终来识别图像中的文本。同样类似的还有Bissacco[47]，他使用神经网络结合HOG特征来做字符分类器，对分割后的结果使用N元模型（N-gram）来做搜索，最终得到字符结果。

传统文字识别方法要求研究人员预先设计恰当的特征表达方法。但是自然场景文字极其复杂，设计能够表达文字信息的特征非常的困难，所以在自然场景文字识别中，传统的文字识别方法效果有限。

### 1.3.2 基于深度学习的文字识别

基于传统文字识别方法的局限，越来越多的研究开始利用深度学习的方式来进行识别。2013年，Alsharif[41]等人将卷积神经网络提取的特征送入隐马尔科夫（Hidden Markov Model, HMM）网络，来实现文字识别。Goodfellow [45]等人使用训练好的字符位置分类器来预测文本中字符数量，直接使用卷积神经网络进行字符识别，该方法对Google的街景门牌号码数据集和验证码数据集均取得了很好的效果。2016年，富士通研究所利用过分割技术结合卷积神经网络（Convolutional Neural Network, CNN）[24]首先分割出字符，然后对字符进行识别。完成了对自然场景文字的分割与识别。是目前自然场景文本识别领域效果最好的算法。

### 1.3.3 端到端研究方法的兴起

目前的方法还是有很多的问题：算法鲁棒性差，难以应对长字符串以及粘连字符；人工设计与提取特征往往不具有很好的通用性。虽然基于深度学习的方法取得了一定成绩，但是往往需要研究者有较高的专业背景知识，算法过多依靠研究者的经验来确定网络参数，技术过于复杂。

2015年，Miao[49] 使用循环神经网路网络识别语音信息。只需要足够多的标记样本，就可以完成对语音的识别。网络结构简单，训练过程无需过多人工干预。这种端到端（end to end）的方法逐渐成为人们的研究热点。该方法不需要研究者具有较强的背景知识就可以训练出很好的模型，取得很好的效果。

## 1.4 本文的研究目标与内容安排

本文针对上述的问题展开探索与研究，提出了基于深度学习的文本识别方法，以实现端到端的训练，无需专业领域的背景知识,简化了特征提取与分类器训练算法难度。

本文的主要研究任务如下：

1. 提出了结合CNN与RNN的新型网络结构，实现了端到端的文字识别。

2. 构建一个标准验证码数据集，对本文模型进行评估。

本文的内容与结构安排如下：

第二章：深度学习。对人工神经网络（Artificial Neural Networks, ANN），递归神经网络(Recurrent Neural Network, RNN)与卷积神经网络（Convolutional Neural Networks,CNN）的基本理论进行简要的介绍。

第三章：基于RNN的文本识别方法。结合CNN与RNN的新型网络结构来识别文本。

第三章：自然场景文本识别实验与总结。实验系统设计，平台搭建与基本配置情况。

第四章：全文总结与展望。本文算法的提高与不足进行总结，提出未来要开展的工作作出展望。

# 第二章 神经网络

## 2.1 传统人工神经网络

人工神经网络（Artificial Neural Networks, ANN）简称神经网络[13]，是模拟动物的神经网络的计算模型。这些计算模型通过模拟动物大脑的神经网络进行抽象计算，构建人工神经元，然后按照一定的拓扑结构来构建神经元之间的连接，来模拟动物的神经网络。1949年，生理学家D.O.Hebb提出改变神经元连接强度的Hebb规则，为神经网络的学习算法奠定了基础。这时的神经网络实只有一个隐含层，且受限于当时计算机硬件资源，并没有预期的效果[9][10]。1974年，哈佛大学的博士生Paul Werbos提出了著名的BP算法(back—propagating，反向传播法)[11]，但没有引起学界的关注。但在1986年，BP算法受到多伦多大学的Geoffrey Hinton等人[12]的关注，他们并在Nature上发表了一篇对这个算法的研究，这使得人工神经网络再次得到关注。

### 2.1.1 神经元

神经元是神经网络最简单的组成部分。人工神经元和感知器类似，模拟动物的神经元特性，接受一组输入并产生输出。下图是一个最简单的神经网络模型，它只包含一个神经元，如图所示：



##### 图2-1 神经元结构

这个神经元的输入为输出为为激活函数，为线性变换，为偏置项。

### 2.1.2 常用激活函数

线性模型的表达能力往往有限，为了增强神经网络的表达能力，我们需要使用连续非线性激活函数。下面介绍几种神经网络常见的激活函数。

#### logistic函数

logistic函数是一种sigmoid型函数，定义为：

（2.1）

logistic函数将输入映射到（0，1）范围内。函数的特点和动物神经元类似，对一些输入会产生兴奋，对另外一些输入会产生抑制。当输入越大时，函数输出越接近1；输入越小时，函数输出越接近0.

#### tanh函数

tanh函数也是一种sigmoid型函数，定义为：

（2.2）

tanh函数可以看作是放大并平移后的logistic函数：

（2.3）

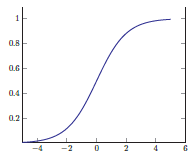
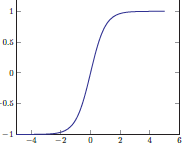
图2-2给出了logistic函数与tanh函数的形状。

#### 线性修正单元

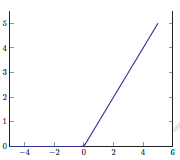
线性修正单元（rectified linear unit, ReLU）也称为rectifier函数[15][16]。在深层神经网络中普遍被使用。ReLU形状如图1-4所示，是一个“斜坡”函数，定义为：

(2.4)

采用ReLU函数的神经网络在计算上更加高效。Relu函数会使一部分神经元的输出为0，这样就造成了网络的稀疏性，并且减少了参数的相互依存关系，缓解了过拟合问题的发生。

a. logistic函数 b. tanh 函数



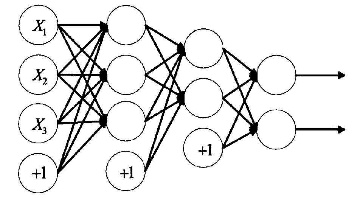
c. rectifier 函数

##### 图2-2

### 2.1.3 前馈神经网络

单层神经元通过训练只能得到一个线性分类面，其分类能力有限。动物大脑中神经细胞是相互连接在一起的，我们可以构建类似的神经网络结构，组成多层的人工神经网络。多层神经网络能够表示多个非线性分类面，分类能力大大增强。其中，前馈神经网络是最简单的多层神经网络[17]。

在前馈神经网络中，各神经元分属于不同的层。每一层的神经元接收上一层神经元输出的信号，并输出给下一层的神经元。整个网络中无反馈，信号从输入层向输出层单向流动，可以用一个有向无环图表示。如图1-3所示是一种广泛使用的前馈神经网络。它包含输入层 L1，两个隐藏层L2,L3,以及输出层L4。



layer1 layer2 layer3 layer4

##### 图1-5 三层神经网络

我们使用下面的记号来描述前馈神经网络。: 表示神经网络的层数；:表示第层神经元的个数；:表示层神经元的激活函数；:表示层到第层的权重矩阵；:表示层到第L层的偏置；:表示层神经元的输入；:表示层神经元的输出。我们可以得到前馈神经网络第层神经元的输入为：

（2.5）

第L层的神经元输出为：

（2.6）

公式合并为：

（2.7）

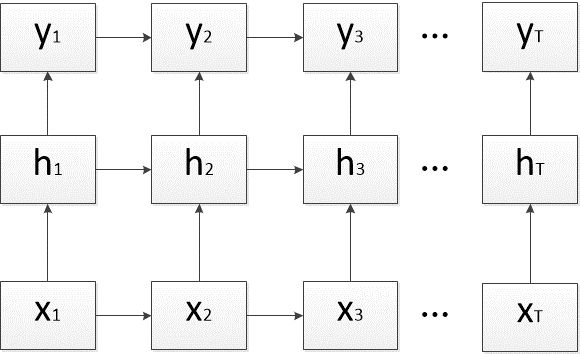
整个网络可以看作一个函数。

## 2.2 循环神经网络

### 2.2.1 简单循环神经网络

传统神经网络中每层之间的神经元是没有连接的，这种网络对于很多问题无能为力。循环神经网络（Recurrent Neural Network, RNN ）[18]，利用带自反馈的神经元可以处理任意长度的输入序列。这种网络更加适合处理与序列有关的任务。它已被广泛应用于语音识别，自动翻译，聊天机器人等任务上。

一个简单循环神经网络按时序展开之后如图1-6 所示：



##### 图1-6 循环神经网络按时序展开

这种网络在神经元之间既有内部的反馈连接，又有前馈连接。假设在时刻时，输入为, 隐层状态为。我们可以得到如下的表达形式：

（2.8）

为激活函数，和为线性变换函数，为偏置项。我们可以看到不仅和当前时刻的输入有关，也和上一个时刻的隐层状态有关。

### 2.2.3 RNN反向传播算法

循环神经网络的 参数训练可以通过随时间进行的反向传播算法[19] [20]来确定。假设循环神经网络在每个时刻t都有一个损失。整个序列损失为 。序列的损失关于的梯度为：

（2.9）

根据链式法则，展开可得

（2.10）

将项展开，

（2.11）

令 ƞ 。

将ƞ代入公式（2.10），可得

（2.12）

显然，当ƞ>1，时，，会出现梯度爆炸问题。而如果ƞ>1，时，，会出现梯度消失。因此，在实际中简单循环神经网络只能学习到短周期的时序依赖关系，即长期依赖问题[21]。

### 2.2.4 LSTM

为解决这个问题, Hochreiter 和 Schmidhuber 引入了门机制（Gating Mechanism）来控制信息的累计速度，并选择性遗忘以前的信息。这就是长短时记忆神经网络（Long Short-Term Memory Neural Network, LSTM）[22]。在时刻时候，记忆单元记录了到时刻为止的所有历史信息，受三个门单元控制：输入门，遗忘门和输出门。LSTM的更新方式为：

（2.13）

（2.14）

（2.15）

（2.16）

（2.17）

（2.18）

这里是logistic函数，是对角矩阵，是当前时刻的输入。遗忘门控制每一个内存单元需要遗忘的信息，输入门控制每一个内存单元记忆多少新的信息，输出门控制每一个内存单元输出多少信息。



图2-7 LSTM 结构图

LSTM网络可以学到长周期的历史信息。LSTM网络的计算机构可以用 图2-7来表示。

## 2.3 卷积神经网络

卷积神经网络（Convolutional Neural Networks，CNN）属于前馈神经网络。1962年Hubel和Wiesel对猫视觉系统的研究中，发现猫的大脑皮层的构造对于局部的视觉空间非常敏感，提出了感受野（receptive field）的感念。进而发展出了卷积神经网络（Convolutional Neural Networks，CNN）。卷积神经网络被广泛应用于计算机视觉领域，是目前机器学习领域的研究热点。卷积神经网络通过局部感受野、全值共享和下采样实现了对输入图像的位移变化、尺度变化、形变变化的不变性。对于卷积神经网络第L层神经元的输入有如下定义：

（2.31）

这里为第L层滤波器，代表L-1层神经元的输入，代表偏置矩阵。f为激活函数。通常会使用多组滤波器来得到多组输出，以此来增强卷积层的表达能力。

### 2.3.1 局部连接

全连接网络中，层与层神经元结点之间是全连接的。而在卷积神经网络中，每一层的神经元节点只和相邻层较近的神经元节点相互连接。图2-8

1. 全连接网络 b. 卷积神经网络

图2-8 全连接与局部连接方式示意图

在普通前馈神经网络中，每层神经元节点之间均相互连接。卷积神经网络中，最底层神经元只与它最相近的三个神经元连接。这样大大降低了神经网络的参数量。

### 2.3.2 权值共享

卷积层中每一个滤波器对输入数据进行卷机操作，提取图像特征。每个滤波器共享相同的参数。



图2-9 权值共享示意图

在上图中，L层有三个神经元，与特定神经元相连接的连接线的权重是共享的。上图展示部分有三组共享权值的滤波器。共享权值使得训练时神经网络要学习的参数数量大大降低。滤波器用来提取局部区域的特征，相当于一个特征提取器。

### 2.3.3 下采样

卷积层共享权重虽然可以减少神经元之间连接的个数，但是经过卷积后的特征维数仍然很高。在卷积层后进行池化（Pooling）操作，可以降低特征维数，避免过拟合。设经过第L层卷积后的特征为，我们将分为K个区域加上下采样函数后的神经元定义为：

（2.32）

其中，和分别为待训练的权重与偏置项。

下采样（Subsampling）函数一般取区域中的最大值（Maximum Pooling）或者平均值（Average Pooling）。

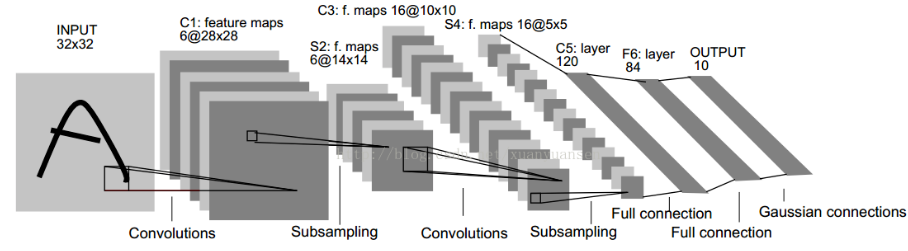
(2.33)

(2.34)

池化操作使得神经元对于较小的形态学改变能够保持不变性，并且可以使得感受野更大。

2.3.4 LeNet-5示例

1989年LeCun提出了经典的卷积神经网络模型LeNet-5[36],之后被美国多家银行用来识别支票上面的手写体数字。



##### 图2-10 LeNet-5 网络结构

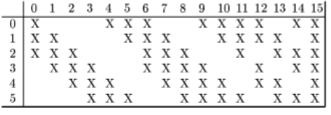
LeNet-5一共有8层。第一层为输入层，输入图像为3232像素的黑白图片。

C1层为卷积层。卷积核为55，一共6个卷积核。输出为6组2828的feature map.该层神经元个数为64704。每个feature map包括55的卷积核参数，以及一个偏置项，所以待训练参数为（55+1）6=156。

S2层为下采样层。对C1层卷积得到的6组feature map中22邻域进行下采样，取平均值。该层神经元个数为1414。根据公式2.32可知，可训练参数为6（1+1）=12个。

C3层同样为卷积层。上一层S2有6组feature map，C3层将上一层不同的feature map进行组合，再进行卷积操作。表2-1所示为S2层与C3层的连接关系。

##### 表 2-1



由表2-1可知，C3层第一个特征映射依赖于S2层前三个feature map，C3层第二个特征映射依赖于S2层第二、三、四个feature map，依此类推。C3层一共有60个卷积核，大小为55。C3层可训练参数为5516=1516。

S4层又是下采样层。对C3层卷积得到的16组feature map中22邻域进行下采样。该层神经元为55。可训练参数为16（4+1）= 2000个。

C5层为卷积层。每个feature map与上一层的全部feature map相连接。一共有12016=1920个卷积核，卷积核大小为55。可训练参数为192025+120 = 48120个。

F6层为全连接层。该层将上一层的feature map全部连接，输出为84个神经元。可训练参数个数为84（120+1）= 10164个。

最后一层为输出层。输出层由10个径向基函数（Radial Basis Function， RBF）组成，输出结果为10个数字的概率值。

# 第三章 基于RNN的文本识别方法

## 3.1 问题分析

计算机对文本的识别是通过将图像信息转化为计算机可表示的信息并进行处理的过程。由于自然场景的文字及其复杂，现在的识别方法没有很好的鲁棒性，并且需要较深的专业领域知识，通用性较差。 同样的问题，在物体分类、图像分割等其他计算机视觉领域也存在。随着深度学习理论的发展，卷积神经网络在计算机视觉领域取得了不错的成绩。卷积神经网络直接输入图像像素信息，不需要人工设计特征，同时也大大减少了数据预处理的工作量。

虽然基于卷积神经网络[23]的自然场景文字识别取得了一定的成果，但是仍然存在一些不足。卷积神经网络针对单字的识别虽然较高，但是对于复杂的自然场景下的文本信息，却依然需要配合传统的字符定位与分割方法[24]，不能实现端到端的训练。而且当图像所含文本较长时，识别率会明显下降。

我们借鉴深度学习在语音识别上取得的成就[26]，将文本识别任务作为特殊的“语音信息”进行处理[25]。本章将会结合CNN与RNN，提出一种新型网络结构，用于文本识别，以期能够实现端到端的自然场景文本识别。

## 3.2 数据集

本文使用了两种数据集，根据python验证码库captcha生成的验证码图片以及被广泛使用的Google 街景门牌号数据集（SVHN）[34]。

C:\Users\a\AppData\Local\Microsoft\Windows\INetCacheContent.Word\16_6fd9.bmpC:\Users\a\AppData\Local\Microsoft\Windows\INetCacheContent.Word\28.png

*a. Captcha生成的验证码图片 b. SVHN数据*

##### 图4-4 数据集

Captcha生成的验证码噪声、背景图与字符数均可以由用户决定，我们这里生成长度为4个与3个字符的验证码图片。SVHN数据集是真实世界的街道门牌号数字识别数据集。SVHN有10类数字，长度不等。其中73257幅图片用于训练，26032幅图片用于测试，531131幅图片是较容易的图片用于额外的补充数据。

## 3.3 图像预处理

自然场景图像虽然大部分是彩色图，但是实验发现图像转换为灰度图并不影响识别精度。为了减小计算量，原始图像可以按照灰度图来读取。为了减少光照的影响，需要对图像进行灰度值归一化。

为了后续神经网络处理方便，图像的输入需要保持固定大小。这里将图像设为约60 像素。图像预处理示意图如下所示：

C:\Users\a\AppData\Local\Microsoft\Windows\INetCacheContent.Word\13.png C:\Users\a\AppData\Local\Microsoft\Windows\INetCacheContent.Word\13_42.bmp

图3-1 图像预处理

## 3.4 方法概述

本文提出基于改进的RNN为基础框架。整个识别流程如图3-2所示：

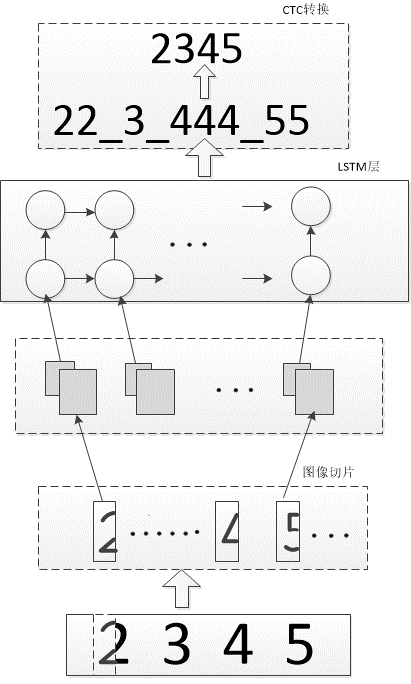


图3-2 文本识别流程图

训练集中图片归一化为6060像素，使用滑动窗经过图像，得到若干组图像切片。将切片送入卷积层提取特征，得到切片的若干组特征，将切片特征经过全连接网络后送入LSTM网络。网络输出经过CTC层转换后得到最终结果。

### 3.4.1 CNN层提取特征

图像完成预处理后，使用卷积层来对图像切片提取特征。这里设置卷积核大小分别为33、55，步长均为1。卷积后使用Relu函数进行激活，再经过池化层。CNN层的输入为一批图片，由2000的张量组成，分别代表批大小、切片高度、切片宽度、切片通道数。

早期神经网络中人们通常使用Sigmoid以及Tanh函数作为神经元的激活函数。实验表明，数据在远离原点的区域变化不大，这会导致反向求导的时候梯度过快消失，使得训练变得困难。之后，因为ReLU函数更加接近生物神经细胞的激活模式，并且因为近似线性函数的优点，开始被应用于卷积神经网络。本文使用ReLU函数作为激活层的激活函数。

为减少计算量，图像切片经过卷积层、激活层后再次进行池化操作。本文使用最大化池化（Max pooling）:使用22的滑动窗进行池化，步长为2，取滑动窗中的最大值输出。

3.4.2 LSTM网络设计



图3-3

卷积层提取图像切片特征后，将特征数据送入LSTM网络。本文中每个RNN结构包含100个LSTM单元。特征数据首先经过一次先行转换，得到1的向量，以适应LSTM单元结构，然后送入下一层的网络。

## 3.5 Connectionist Temporal Classification

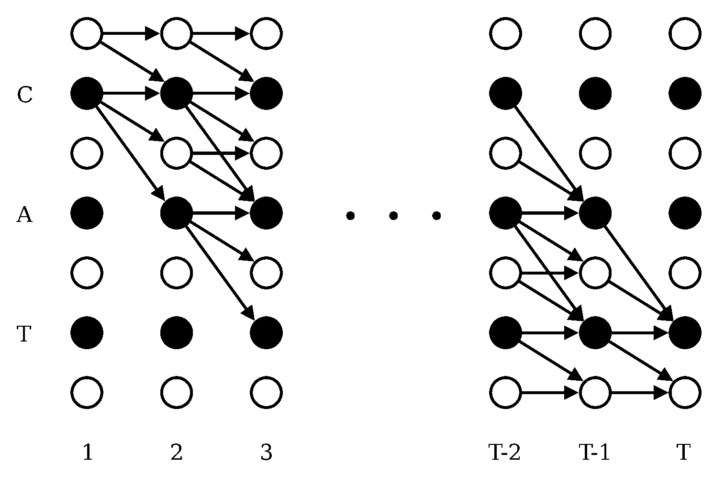


图3-4 CTC路径搜索示意图

在LSTM网络中，如果序列的长度为T，则会产生T个输出。这就需要将LSTM的输出解码为正确的类别。Connectionist Temporal Classification (CTC)[27]技术用来实现LSTM网络输出与数据标签对齐，计算LSTM网络训练过程中标签与LSTM网络输出之间的Loss。图3-4 是CTC对LSTM网络输出到数据标签的路径搜索示意图。

通过图3-4可以发现，CTC在数据标签中增加空白标签来分割标签之间的间隔，最后会把空白符号和预测出的重复符号消除。假设输入，标签为，数据集为，损失函数为。CTC技术使得模型从输入端输入数据后可以在输出端直接取得识别结果，实现了端到端训练算法。

# 第四章 自然场景文本识别实验与总结

深度学习的训练要求计算机有较高的计算能力。2012年ImageNet竞赛第一名AlexNet模型[28][29][30]，使用两块GTX 580 GPU,训练时间长达六天，2014年ImageNet竞赛第一名 Oxford VGGNet模型[31]，使用4块Titan Black GPU，训练时间更是长达3周。另外，神经网络所含的参数众多，如神经网络层数、神经元数量、规则化系数、激活函数等。训练时超参数如学习率，批尺寸（batch size）大小等也会明显影响收敛速度。

本章尝试使用一些策略来缓解这些问题，并探索不同的参数对实验的影响。

## 4.1 不同深度学习开发框架对比

深度学习技术近年来发展迅速，很多互联网企业与研究机构也都推出了自己的深度学习开发平台，将深度学习理论迅速转化为代码应用到开发实践当中，大大促进了深度学习领域的发展。

我们首先介绍当前最为常用的几大深度学习开发框架，并比较各个框架的优点与适用场景。

### 4.1.1 Caffe



##### 图4-11 Caffe标志

Caffe（Convolutional Architecture for Fast Feature Embedding）最早是由伯克利大学视觉与学习中心开发的基于C++/CUDA实现的卷积神经网络工具。2013年开源后迅速在深度学习开发人员中普及开，是深度学习领域最为流行的开发框架。

Caffe利用了MKL, OpenBLAS, cuBLAS等计算库，支持CUDA加速；提供了一整套工具集，可以用于模型训练、预测、微调、数据预处理，以及自动测试；完全开源，代码组织良好，可读性强。

但是Caffe不支持递归神经网络。用户如果自己定义新的网络架构，需要改动源代码并配合ProtoBuf描述。同时为了支持GPU，需要自己手动实现一遍CUDA版的forward、backward、gradient update。目前Caffe只支持单机多卡的并行计算，不支持多几多卡的分布式计算。

### 4.1.2 PyTorch



##### 图4-12 PyTorch标志

PyTorch[38]是Facebook AI实验室领衔开发深度学习工具包，源自较早的torch工具包。支持大部分的机器学习算法，是一个轻巧的框架。集成了各种计算加速库，如Intel MKL、CuDNN和NCCL来优化速度。PyTorch基于python，底层使用C/CUDA扩展模块实现，可已在GPU加速基础上实现张量；支持自动求导。

PyTorch使用反向模式自动微分的技术，可以零延迟地改变网络的行为。相对于Tensorflow、Theano、Caffe等框架，后者需要事先构建一个神经网络，然后进行使用。PyTorch是目前动态神经网络最快的实现技术。使得定义神经网络获得最高的灵活性及速度。

### 4.1.3 MXNet

MXNet[39]是分布式机器学习社区（Distributed Machine Learning Common, DMLC）开发的一个同时兼顾效率和灵活性的深度学习框架，支持多机多卡的分布式运行。MXNet支持众多的语言绑定（C++/python/R/Go），并且支持混合式符号编程和命令式编程。其核心是一个动态的依赖调度，能够自动并行符号和命令的操作，提供了两种编程接口：N维数组（ndarray）接口，类似于Matlab或Python中的numpy或PyTorch中的tensor；符号（Symbolic）接口，可以快速的构建一个神经网络，实现自动求导。



图 4-13 MXNet标志

### 4.1.4 TensorFlow

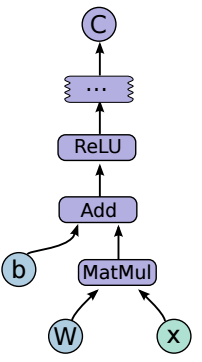


##### 图 4-14 TensorFlow标志

2015年Google推出了人工智能学习系统TensorFlow[40]。并在2017年2月16日发布了TensorFlow1.0正式版。TensorFlow是分布式大规模机器学习框架，并且支持手机等移动设备的移植。随着TensorFlow的发展，以及对CuDNN等加速库的支持，TensorFlow成为目前最为流行的深度学习框架之一。

TensorFlow已经被广泛应用于工业界，如小米、京东、Uber等。TensorFlow支持多种神经网络模型，可以轻松自定义网络结构。支持自动求导。采用数据流计算，其表达的数据流计算可以由有向图来表示。如何4-15所示，每个节点有一个或多个输入和零个或多个输出，表示一种操作的实例化。图中的叶子节点通常为常量或者变量，非叶子节点为一种操作，箭头所示为张量的流动方向。

除了上述的深度学习框架，还有一些框架可以尝试，这里不再一一列举。本文综合考虑多种框架，从易用性以及稳定性上考虑，选择TensorFlow作为本文的实验平台。



##### 图4-15 Tensorflow 计算图

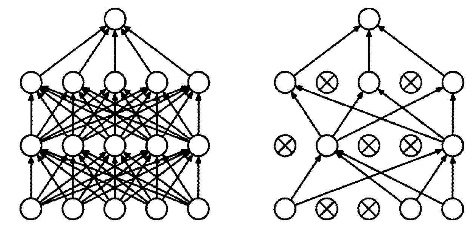
## 4.2 基于GPU的训练

GPU相对CPU有更多的计算核心，每个核拥有相对较小的缓存，少而简单的数字逻辑运算但与。这使得GPU更加适合处理相对简单而重复的计算任务。目前主流的深度学习研究工作越来越多的使用GPU来对训练过程进行加速。训练过程中采用 批随机梯度下降中，每批样本的数量更具GPU显存大小进行合理设置，尽可能充分利用GPU显存。

本实验使用Nvidia GTX 980ti GPU，深度学习框架Tensorflow基于cudnn加速。为减少CPU与GPU通讯耗时，一次性将全部训练数据读入内存，每一轮次迭代由内存送入GPU数据。这样大大增加了GPU利用率，减少资源浪费，加速训练过程。

## 4.3 Dropout

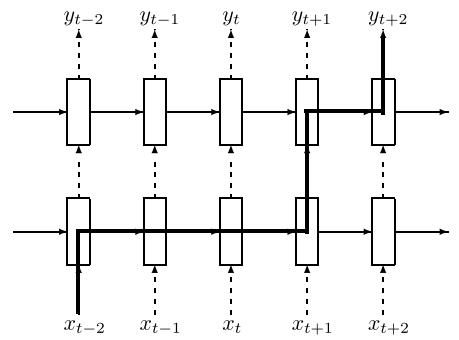
Dropout[32] 作为一种防止过拟合的方法，广泛被用在神经网络训练中。训练时，随机对全连接层直接连接移除。



a. 不进行dropout的全连接网络 b. dropout的全连接网络

##### 图 4-2 dropout示例

Wojciech Zaremba在2015年将dropout应用于RNN网络[33]，取得了不错的效果。图4-3是RNN中dropout应用示例：



##### 图4-3 dropout在RNN中应用

图中虚线代表应用dropout的连接。单步的层间连接进行一定概率的dropout，序列的循环连接不使用dropout。

## 4.4 实验环境

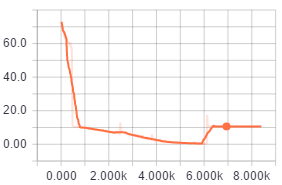
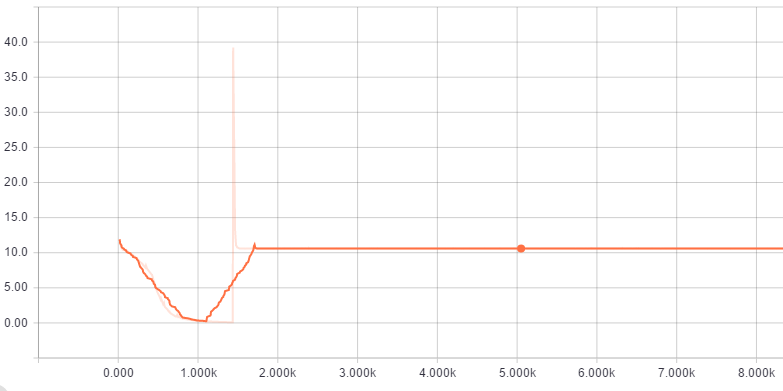
实验使用python进行数据预处理，使用tensorflow框架[35]进行神经网络训练。具体实验环境如表 4.1 所示。

表4.1 实验环境

|  |  |
| --- | --- |
| *CPU* | *Intel core i7-4790K* |
| *GPU* | *NVIDIA GTX980ti* |
| *内存* | *16GB* |
| *操作系统* | *Ubuntu 14.04* |
| *开发语言* | *Python* |
| *GPU开发库* | *CUDA 5.1* |
| *图像处理库* | *Opencv 3.1* |
| *深度学习框架* | *Tensorflow 0.9* |

## 4.4 实验结果

本文首先对验证码数据集进行训练。针对验证码数据集，分别使用了RNN模型与本文提出的CNN结合RNN的新模型训练，均迭代8400次。损失值下降情况如图4-5所示。

a. RNN网络 b. CNN结合RNN网络

##### 图4-5 验证码数据集训练Loss曲线

可以明显看出，本文提出的CNN结合RNN网络较单纯RNN网络收敛速度明显加快。

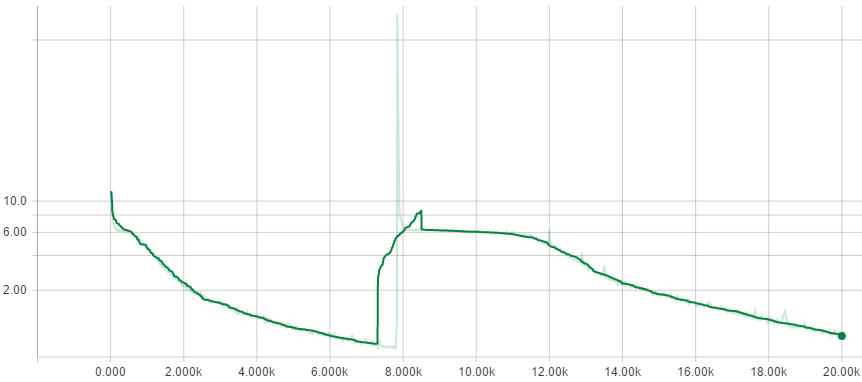
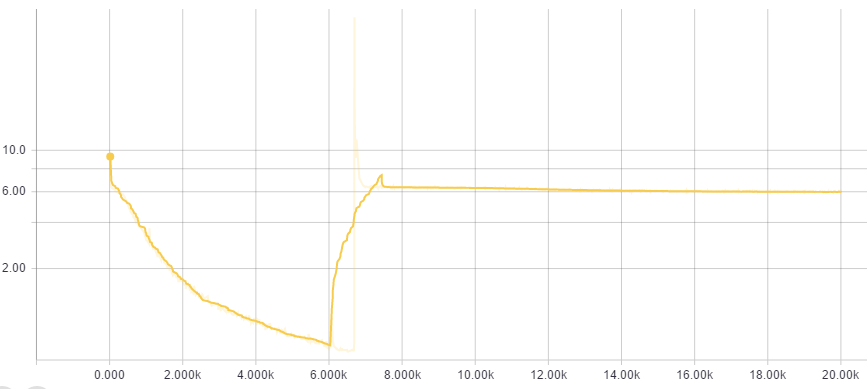
本文继续针对SVHN数据集进行训练。分析图像归一化后不同的大小、卷积核尺寸、RNN序列长度对结果的影响。

该模型训练耗时21个小时。三组实验的精度大致接近，损失值下降情况如图4-5所示。

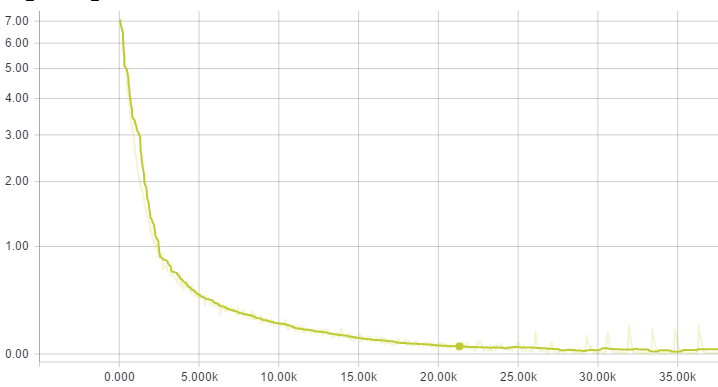
对验证码的实验表明本文的CNN结合RNN要好于单纯的RNN网络模型结果。对SVHN的实验结果发现对于图像不同的归一化尺寸，不同的图像切片大小都对结果有一定影响。较小的切片对应更长的切片序列，结果要好于宽切片的图像序列。

表4.2 实验结果对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | 实验一 | 实验二 | 实验三 |
| 图像大小 | 40×81像素 | 40×81像素 | 40×39像素 |
| 滑动窗大小，步长 | 3×81像素 3像素 | 5×81像素 4像素 | 3×81像素 3像素 |
| RNN:time\_step | 27 | 20 | 13 |
| 精度 | 0.90 | 0.88 | 0.93 |

a. 实验一 b. 实验二



c. 实验三

##### 4.5 结果分析

# 第五章 全文总结与展望

## 5.1 总结

近年来，深度学习技术成为模式识别领域的研究热点，并取得了惊人的成绩。文本识别作为模式识别与计算机视觉领域经典研究方向，已经取得了长足的发展。本文提出的使用CNN结合RNN的方法可以将一幅图像直接作为输入，将特征提取与分类器识别结合，实现了端到端的训练。

本文的工作主要包含一下几个方面：

（1）本文总结和概括了人工神经网络，深度学习和RNN的发展历史，分析了深度学习对人工神经网络的影响。

（2）本文介绍了RNN的算法思想和结构，并推导了RNN的训练过程。

（3）提出了基于CNN和RNN的文本识别方法。利用CNN提取图像的高层语义特征，结合RNN捕获图像全局序列信息。实验表明该方法有很好的表达能力。

## 5.2 展望

深度学习经过近些年的发展，在人脸检测、物体检测、人脸识别、文本识别等领域被广泛应用。随着GPU、分布式计算等技术的发展，以及大数据的便捷获取，深度学习会在更多领域得到应用。基于自然场景的文本识别是一个复杂的问题，现有的技术仍然不能够被大规模商业化应用。以后可能会在一下几个方面对算法进行改进。

（1）提升算法的鲁棒性。本文数据集相对现实场景的数据仍然是较小的。而更大规模的数据集往往更能训练出泛化能力更强的模型。

（2）本文中算法时间复杂度和空间复杂度都较高。训练时使用显存占用高达5.8GB，训练时间长达21小时。未来的研究会尝试减小计算量，使用预训练的CNN来提取特征，以期能够有更好的效果。

# 参考文献

1. HerbertF Schantz. History of OCR,Optical Character Recognition. Recognition Technologies Users Association, 1982.
2. Shyang-Lih Chang,Li-Shien Chen,Yun-Chung Chung,and Sei-Wan Chen,Automatic License plate recognition. IEEE Transactions on Intelligent Transportation Systems,5(1):42-53,2004
3. Gustav Tatwchek. Reading machine,December 31 1935. US Patent 2,026,329.
4. Behzad Shahraray and David C Gibbon. Automated authoring of hypermedia documents of video programs. In ACM International Conference on Multimedia, pages 401-409,1995
5. Toshio Sato, Takeo Kanade, Ellen K Hughes, and Michael A Smith. Video ocr for digital news archive. In IEEE International Workshop on Content-Based Access of Image and Video Database, pages 52-60,1998.
6. 赵志宏，杨绍普，马增强．基于CNNLeNet-5的车牌字符识别研究[J]．系统仿真学报， 2010，03：638．641．
7. Rodolfo Zunino and Stefano Rovetta. Vector quantization for license-plate location and image coding. IEEE Transactions on Industrial Electronics,47(1):159-167,2000.
8. Ramos S, Gehrig S, Pinggera P, et al. Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling[J]. arXiv preprint arXiv:1612.06573, 2016.
9. F.Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. Psychological review,vol 65,pp.386-404,1958.
10. W.McCulloch and W.Pitts. A logical calculus of the ideas immanent in nervous activity,bulletin of Mathematical biophysics,vol.10,pp. 115-133,1943.
11. Werbos P J. Beyond regression: new tools for prediction and analysis in the behavioral sciences[D]. Boston: Harvard University,1974.
12. Rumelhart D, Hinton G, Williams R. Learning representations by back-propagating errors[J]. Nature,1986,323:533-536.
13. Hopfield J J. Artificial neural networks[J]. IEEE Circuits and Devices Magazine, 1988, 4(5): 3-10.
14. Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]//Aistats. 2011, 15(106): 275.
15. Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th international conference on machine learning (ICML-10). 2010: 807-814.
16. Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: A deep learning approach[C]//Proceedings of the 28th international conference on machine learning (ICML-11). 2011: 513-520.
17. 徐春晖, 徐向东. 前馈型神经网络新学习算法的研究[J]. 清华大学学报: 自然科学版, 1999, 39(3): 1-3.
18. Williams R J, Zipser D. A learning algorithm for continually running fully recurrent neural networks[J]. Neural computation, 1989, 1(2): 270-280.
19. Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560.
20. Williams R J, Peng J. An efficient gradient-based algorithm for on-line training of recurrent network trajectories[J]. Neural computation, 1990, 2(4): 490-501.
21. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE transactions on neural networks, 1994, 5(2): 157-166.
22. Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
23. Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex[J].The journal of physiology,1962,160(1):106.
24. Chen L, Wang S, Fan W, et al. Cascading Training for Relaxation CNN on Handwritten Character Recognition[C]//Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on. IEEE, 2016: 162-167.
25. Graves A. Supervised sequence labelling[M]//Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 2012: 5-13.
26. Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013: 6645-6649.
27. 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.
28. Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.
29. Bhaskar J, Patel A. Image Classification using Convolutional Neural Network[J].
30. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
31. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
32. Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
33. Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.
34. Goodfellow I J, Bulatov Y, Ibarz J, et al. Multi-digit number recognition from street view imagery using deep convolutional neural networks[J]. arXiv preprint arXiv:1312.6082, 2013.
35. Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[J]. arXiv preprint arXiv:1603.04467, 2016.
36. LeCun, Yann, et al. "Comparison of learning algorithms for handwritten digit recognition." International conference on artificial neural networks. Vol. 60. 1995.
37. Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014.
38. Mitra, Bhaskar, Fernando Diaz, and Nick Craswell. "Luandri: a Clean Lua Interface to the Indri Search Engine." arXiv preprint arXiv:1702.05042 (2017).
39. Chen, Tianqi, et al. "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems." arXiv preprint arXiv:1512.01274 (2015).
40. Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).
41. Mishra, Anand, Karteek Alahari, and C. V. Jawahar. "Scene text recognition using higher order language priors." BMVC 2012-23rd British Machine Vision Conference. BMVA, 2012.
42. Zimmermann, Grant R., Joseph Lehar, and Curtis T. Keith. "Multi-target therapeutics: when the whole is greater than the sum of the parts." Drug discovery today 12.1 (2007): 34-42.
43. Rodriguez-Serrano, Jose A., Florent Perronnin, and France Meylan. "Label embedding for text recognition." Proceedings of the British Machine Vision Conference. 2013.
44. Almazán, Jon, et al. "Word spotting and recognition with embedded attributes." IEEE Transactions on Pattern Analysis and Machine Intelligence 36.12 (2014): 2552-2566.
45. Matan, Ofer, et al. "Multi-digit recognition using a space displacement neural network." NIPS. 1991.
46. Jaderberg, Max, et al. "Reading text in the wild with convolutional neural networks." International Journal of Computer Vision 116.1 (2016): 1-20.
47. Bissacco, Alessandro, et al. "Photoocr: Reading text in uncontrolled conditions." Proceedings of the IEEE International Conference on Computer Vision. 2013.
48. Perronnin, Florent, et al. "Large-scale image retrieval with compressed fisher vectors." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.
49. Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding[C]//Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE, 2015: 167-174.