

刘邦龙-毕业论文 - 终

【PDF报告-大学生版】

报告编号: bf3898c2c2994980

检测时间: 2018-06-07 23:27:45

检测字数: 23,271字

作者名称: 刘邦龙

所属单位: 南京晓庄学院 (教务处)

检测范围:

- ◎ 中文科技期刊论文全文数据库
- ◎ 中文主要报纸全文数据库
- ◎ 中国专利特色数据库
- ◎ 博士/硕士学位论文全文数据库
- ◎ 中国主要会议论文特色数据库
- ◎ 港澳台文献资源
- ◎ 外文特色文献数据全库
- ◎ 维普优先出版论文全文数据库
- ◎ 互联网数据资源/互联网文档资源
- ◎ 高校自建资源库
- ◎ 图书资源
- ◎ 古籍文献资源
- ◎ 个人自建资源库
- ◎ 年鉴资源
- ◎ IPUB原创作品

时间范围: 1989-01-01至2018-06-07

检测结论:

- 全文总相似比: 12.86% (总相似比=复写率+他引率+自引率)
- 自写率: 87.14% (原创内容占全文的比重)
- 复写率: 12.86% (相似或疑似重复内容占全文的比重, 含专业用语)
- 他引率: 0% (引用他人的部分占全文的比重, 请正确标注引用)
- 自引率: 0% (引用自己已发表部分占全文的比重, 请正确标注引用)
- 专业用语: 0.00% (公式定理、法律条文、行业用语等占全文的比重)

总相似片段: 61

期刊: 11 博硕: 20 外文: 0 综合: 0 自建库: 0 互联网: 30

颜色标注说明:

- 自写片段
- 复写片段 (相似或疑似重复)
- 引用片段
- 引用片段(自引)
- 专业用语 (公式定理、法律条文、行业用语等)

分 类 号: TP311

学校代码: 11460

学 号: 14131613

南京晓庄学院本科生毕业设计

基于卷积神经网络的图像识别研究

Realization of image recognition based on Convolution Neural Network.

所 属 学 院: 信息工程学院

学 生 姓 名: 刘邦龙

指 导 教 师: 王寅同

职 称: 讲 师

研究起止日期: 二一八年一月至二一八年六月

二一八年六月

学位论文独创性声明

本人郑重声明:

1. 坚持以“求实、创新”的科学精神从事研究工作。
2. 本论文是我个人在导师指导下进行的研究工作和取得的研究成果。
3. 本论文中除引文外, 所有实验、数据和有关材料均是真实的。
4. 本论文中除引文和致谢的内容外, 不包含其他人或其它机构已经发表或撰写过的研究成果。
5. 其他同志对本研究所做的贡献均已在论文中作了声明并表示了谢意。

作者签名: 刘邦龙

2018年 6月

摘要

图像识别技术, 利用计算机视觉去识别图像中的物体、地点、人物和动作的能力。计算机通常可以与照相机和和人工智能软件相结合使用机器视觉技术实现图像识别。图像识别技术被用于执行大量基于机器的视觉任务, 例如用元标签标注图像内容, 执行图像内容搜索和引导机器人, 自动驾驶汽车和事故回避系统。虽然人类和大多数的动物都能够很容易地识别物体, 但计算机却很难完成任务。图像识别技术需要深度机器学习为基础, 基中在图像识别中, 卷积神经网络的性能处理上是最好的, 图像识别算法可以利用比较网络模型, 从不同角度使用边缘检测或组件来实现图像识别。图像识别算法通常是在数以百万计的预先标记的标准图片集上进行训练, 并引导计算机学习。

本本文基于经典的卷积神经网络模型, 使用Tensorflow, Keras等工具将其用于手写数学和英文字母识别任务。主要任务如下:

1) 构建卷积神经网络模型并应用于图像识别问题:手写数字识别和手写英文字母识别。

2) 识别实验采用Flask网页框架和相关技术对手写数据进行预测。

3) 实验的结果验证了卷积神经网络应用于手写数字识别和手写英文字母识别问题上的可行性。

本文证明了卷积神经网络可以很好地应用于手写体数字识别和手写英文字母识别问题, 无需对神经网络模型进行大量的调整与修改, 取得了比较好的识别效果。

关键词: 卷积神经网络; 图像识别; 手写数字识别; 网站建设

Abstract

Image recognition technology, the ability to use computer vision to identify objects, places, people and actions in an image. Computers can usually be combined with cameras and artificial intelligence software to achieve image recognition using machine vision technology. Image recognition technology is used to perform a large number of tasks based on machine vision, for example use meta tags tagging image content, implement image content search and guide the robot, automatic driving and accident avoidance system. Although humans and most animals can recognize objects easily, computers are difficult to accomplish tasks. Image recognition technology requires depth based on machine learning, in the base in the image recognition, the convolution of the neural network is one of the best performance, image recognition algorithm can use the network model, using edge detection or components from different angles to achieve image recognition. Image recognition algorithms are usually trained on millions of pre-labeled standard sets of images and guide computer learning.

This paper, based on the classical convolution neural network model, uses Tensorflow, Keras and other tools to use it for handwritten mathematics and English letter recognition tasks. The main tasks are as follows:

1) build the convolution neural network model and apply it to image recognition problems: handwritten digit recognition and handwritten English letter recognition.

2) the identification experiment USES Flask web page framework and relevant technology to write data for prediction.

3) the experimental results verify the feasibility of convolution neural network in the recognition of handwritten numerals and handwritten English letters.

This article proves the convolutional neural network can be well applied to handwritten numeral recognition and handwritten English letters recognition problem, don't need to make a lot of neural network model to adjust and modify, better recognition effect is obtained.

Keywords: CNN; Image recognition; Handwritten digital recognition; website construction

目录

摘要 I

Abstract II

目录 IV

第1章 绪论 5

1.1 开发背景和意义 5

1.2 图像识别技术现状 6

1.3 论文的研究内容 7

第2章 相关研究技术 8

2.1 Python语言 8

2.1.1 Python简介 8

2.1.2 Python版本选择 9

2.2 卷积神经网络 9

2.2.1 简介 9

2.2.2 基本概念 10

2.2.3 基本结构 11

2.2.4 第三方集成库 13

2.3 网站搭建技术 14

2.3.1 简介 14

2.3.2 网站结构 15

第3章 网站搭建 16

3.1 服务器与配置 16

3.2 网站环境搭建 16

3.2.1 Python3.6 16

3.2.2 虚拟环境搭建 16

3.2.3 Flask网站框架安装 17

3.4 网站工作流程 17

3.5 网站运行结果 18

第4章 手写数字/英文字母图像识别实现 19

4.1 手写数字/英文字母的数据准备 19

4.2 手写识别应用程序实现 20

4.2.1 加载数据并训练数据（返回模型并估模型） 20

4.2.2 构建和训练卷积神经网络 23

4.2.3 评估模型 24

4.2.4 网页端识别手写数字/英文字母 24

第5章 总结与展望 29

参考文献 30

致谢 31

1 绪论

1.1 开发背景和意义

近些年，伴随着人工智能理论和图像识别技术的发展，基于卷积神经网络的图像识别技术逐渐开始新兴起来。图像识别，简单来说，就是利用现代计算机技术采集需要的对象，以图像的数据为基础，对图像进行对象识别，让计算机或其他机器来模仿人的视觉感知，能够自动完成对某一信息的处理能力，以识别不同模式的目标和对象的技术，代替人完成图像分类及辨别的任务。就当前而言，图像识别的主要应用：光学字符的识别（如手写数字体、信封上邮政编码、汉字、英文字母、条形码识别、二维码、汽车牌照等），以及生物特征识别（如光学指纹、人脸面部特征、红膜等等）。虽然现今图像识别研究已取得了较大的成果，并且成功的应用于各种领域。但是依然有几个迫切问题需要去解决：

第一个问题便是存储在图像本身中的大量数据，不单是图像储存的容量，还有图像存储的信息，非二维的对称信息，比如像图像背后的信息。这使得在识别图像过程中要耗费大量的计算力和很长的时间。其次便是图像在处理之前图像存储时，图像本身可能会旋转、位移、图像发生畸变、尺寸大小的变化，还有外部环境干扰的时，以及机器设备本身的误差，部分图像本身或多或少有些许噪声。随着图像识别在越来越重要的领域中的应用，便要求图像识别系统在有较高的处理运算速度和识别速度的前提下，并且能让图像识别的

准确率提高。

人工智能技术的发展,为解决上述问题,提供了一种新的路径,大规模分布式存储的人工神经网络自适应并行处理操作,以满足对大量数据实时处理的需求。卷积神经网络(Convolutional Neural Networks, 简称CNN),因为其具有卷积运算操作(convolution operators),所以称为卷积神经网络,但有别于其他的神经网络模型(如,误差反向传播算法error Back Propagation,简称BP网络算法、玻尔兹曼机BM: Boltzmann machines、支持向量机算法SVM: Support vector machines、神经图灵机NTM: Neural Turing machines等)。卷积神经网络主要的领域应用于图像识别的相关任务上,诸如,图像分类(image classification)图像语义分割(image semantic sementation)、图像检索(image retrieval)、物体检测(object detection)等计算机视觉问题。还有卷积神经网络去自然语言的处理(natural language processing)中的文本进行分类等工作,软件工程数据挖掘(software mining)中的软件BUG问题预测等问题都可以使用卷积神经网络进行处理,并取得了较传统的算法甚至其他神经网络算法更优的预测效果。

1.2 图像识别技术现状

随着计算机信息技术,移动互联网、智能手机和社交网络的不断发展,图像识别这一技术在社会各个领域的应用越来越广泛和普及,并且已经逐渐开始渗入到了我们的日常生活当中。

与文本相比,图片数据可以为用户提供更生动、更容易理解、更有趣、更艺术化的信息。其次,从图片来源的角度来看,智能手机为我们提供了方便的拍摄和截图,帮助我们更快地收集和记录信息。然而,随着图像成为互联网的主要信息载体,问题出现了。当信息通过书面记录或存储成文字时,我们可以通过关键词搜索,容易找到和编辑内容,但当信息被图像记录时,我们不能检索图像的内容,这时就需要计算机来完成图像识别的任务了。就目前而言,我图像识别技术是排在世界第一我国的图像识别其本身具有一定的优势,加之我国的人口众多,上网的用户总体基数大,对于处理图像存储和识别等问题,都有较好的方法和经验。现在图像存储的像素点高,处理精度高、特别是数字图像,在图像分析的基础上研究图像中目标的性质。但是,在实际发展过程中,该技术还是存在着一定的问题。

本文所探讨的是基于卷积神经网络的手写数字/字母识别研究,是在传统计算机图像识别方法的基础上融合了当今卷积神经网络算法的一种图像识别方法,并且成功的将该卷积神经网络模型运用到了网页端的手写体程序识别上。

1.3 论文的研究内容

本论文研究的内容是:基于卷积神经网络(Convolution Neural Network),利用公用数据集MNIST和EMNIST,进行学习迭代,以达到高效率,高准确度识别目标。论文的基本框架:

第一章结论,该章分析了本论文研究的背景和意义,现今国内外图像识别的技术现状及论文的基本架构。

第二章相关研究技术,本章针对论文中所实现的卷积神经神经网络图像识别中所用到的技术,开发工具,背景技术和开发语言进行简单的介绍。

第三章网站搭建,简单介绍与网站相关的背景技术,并通过Flask网站微框架,实现利用卷积神经网络算法识别手写数字图像在其上面运行。

第四章手写数字/英文字母图像识别实现,通过基于经典卷积神经网络算法,对手写数字/英文字母(区分大小写)的识别。

第五章总结,对论文的完成情况进行总结和说明,表述其开发过程中所遇到的问题,不足之处与心得感悟,今后未来的展望。

2 相关研究技术

2.1 Python语言

2.1.1 Python简介

Python 是面向对象的解释型高级计算机程序设计语言,具有非常丰富和强大的库以及第三方的库文件。一种高级的、多用途的、解释的、交互式的和面向对象的编程语言。Python易于学习,通用脚本语言Python是一种编程语言,它支持结构化和功能性方法,并且具有数据结构的构建、可移植性和可扩展性。它也是一种可扩展的语言。解释意味着它在运行时通过解释器进行处理。在执行之前不需要编译程序。它是用来给数据库交互的能力,构建的应用程序可以运行在Windows环境中,建立的CGI脚本可以运行从Web浏览器等。到处都是用于从商业网站在线游戏,从简单的转换脚本到复杂网络更新银行和其他金融机构的例程。

Python是一种编程语言，可以在许多不同的平台上应用。它附带了标准库，包括几乎所有的Internet协议、操作系统接口、测试和字符串处理。字符串处理中的正则表达式，计算文件之间的差异，Unicode。Internet协议—HTTP、SMTP、FTP、XML-RPC、IMAP、POP、CGI编程。软件工程中的单元测试、日志记录、剖析和解析Python代码。操作系统接口中的系统调用、文件系统和TCP/IP套接字。

与其他编程语言不同，没有几行代码能胜任这份工作。不需要花括号，但需要适当的缩进。不需要python的变量数据类型声明将处理它。Python代码不需要有类，您可以通过在函数或方法中声明它们来获得功能。创建模块并导入它以重用代码。它被用来支持有限的多重继承形式，非常容易学习。它支持其他编程语言所具有的(Object-Oriented Programming System，面向对象的程序设计系统)概念。它提供了易于适应和工作的内置功能。

Python的特点：面向对象的、免费的、可移植的、强大的、可伸缩的、可溶混的、易于使用的、容易学习、稳定的、支持GUI编程。

2.1.2 Python版本选择

Python有Python2.X和Python3.X。之所以选择Python3的，主要原因是本论文中的手写识别程序和网站的相关第三方库完美的支持Python3，而Python2的一些功能包和库文件等都已经不再更新，属于被遗弃状态。一些大的社区论坛GitHub和CSDN上的开发者们更愿意使用Python3，而不是Python2。

2.2 卷积神经网络

2.2.1 简介

在机器学习中，卷积神经网络(Constitutional Neural Networks, CNN)是一种类神经网络，是基于多层深度神经网络而建立起来的一套学习算法，在图像识别和分类等领域已经被证明非常有效了。卷积神经网络与一般的神经网络有着不同的结构。常规的神经网络通过一系列隐藏的层来转换输入。每一层都由一组神经元组成，每一层都与前一层的所有神经元完全相连。最后，还有最后一个完全连接的层——输出层——表示预测。卷积神经网络有点不同。首先，这些层是在三维空间中组织的:宽度、高度和深度。此外，一层的神经元不连接到下一层的所有神经元，但只连接到它的一小部分。最后，最终的输出将被简化为一个概率分数的向量，沿着深度维度组织。

卷积神经网络受到大脑的启发，神经元之间的连通性模式类似于动物视觉皮层的组织。个体皮层神经元对刺激的反应只在被称为接收场的视觉区域的受限区域。不同神经元的接受域部分重叠，从而覆盖整个视野。卷积神经网络允许计算机看到，换句话说，通过将原始图像通过图层转换成一个类的分数来识别图像。CNN受到视觉皮层的启发。每次我们看到一些东西，一系列的神经元层就会被激活，每一层都能检测出一系列的特征，比如线条、边缘。高层次的层将检测更复杂的特征，以便识别我们所看到的。

2.2.2 基本概念

卷积神经网络是由一个输入层，一个输出层以及多个隐藏层组成的。一般隐藏层包含以下几层：

卷积层(Convolutional layer)，是卷积神经网络中的基础操作，卷积层将卷积运算应用于输入，将结果传递给下一层。卷积模拟了单个神经元对视觉刺激的反应。每个卷积神经只处理其接收域中的数据。

池化层(Pooling layer)，通常使用的池化操作的为均值池化(average-pooling)和最大值池化(max-pooling)，本文的卷积神经网络便使用的是最大值池化。池化层的作用是可以使输入的特征值不性(feature invariant)，还能达到图像的特征降维。卷积网络可能包括局部或全局汇聚层，它将神经元簇的输出与下一层的单个神经元相结合。例如，最大池化使用的最大值来自前一层的每一个神经元簇。另一个例子是平均池化，它使用前一层的每个神经元的平均值。

全连接层(Fully-Connected layer)，把每一层神经元与另一层的每个神经元连接起来，在整个卷积神经网络中起到了“分类器”的作用。而本文中所中最后两层采用的两个全连接层，第一个全连接层用来连接前面卷积层过来的输入数据，第二个全连接层是用来分类输出的。

卷积神经网络各层应用实例：图2.1

图2.1 卷积神经网络各层应用实例

2.2.3 基本结构

卷积神经网络基本结构：通过由不同的层组成，其中最基础的是卷积层。这些层通过神经网络层将输入转换为输出。通常使用几种

不同类型的层。如图2.2

图2.2 卷积神经网络各层

卷积层(Convolutions): 卷积层是CNN的核心组成部分。该层的参数由一组可学习的过滤器(或内核)组成,它们有一个小的接受域,但可以通过输入卷的全部深度进行扩展。在转发过程中,每个筛选器都在输入卷的宽度和高度上进行卷积,计算过滤器和输入之间的点积,并生成该过滤器的二维激活映射。因此,当网络在输入的某个空间位置检测到某种特定类型的特征时,它就会学习过滤器。在深度维度上叠加所有过滤器的激活映射,形成卷积层的全部输出量。因此,输出卷中的每一个条目都可以被解释为一个神经元的输出,该神经元在输入中观察一个小区域,并在相同的激活图中与神经元共享参数。

局部感知(Local connectivity): 在处理像图像这种维数据做为输入时, 将每个神经元连接到前一层中的所有神经元是不现实的,因为这样的神经网络结构是没有考虑数据存储的空间结构。卷积网络利用相邻层神经元之间的局部连接模式,利用空间局部相关性:每个神经元只连接到输入量的一小部分。这种连通性的程度是一个被称为神经元接收场的超参数。连接在空间中是局部的(沿宽度和高度),但始终沿着输入卷的整个深度进行扩展。这样的体系结构确保了所学习的过滤器对空间局部输入模式产生最强烈的响应。

图2.3卷积层(蓝色)的神经元,连接到它们的接收层(红色)

权值共享(Spatial arrangement): Spatial arrangement在卷积层中使用权值共享方案来控制参数的数量。它依赖于一个合理假设:如果补片特征对于在某个空间位置计算,则在其它位置计算也是有用的。换句话说,将一个二维的深度切片表示为深度切片,我们在每个深度切片中限制神经元使用相同的权重和偏差。由于单个深度切片中的所有神经元都具有相同的参数, 所以在CONV层的每个深度切片上的前向传递可以被计算为神经元的权重与输入量的卷积(因此名称为卷积层)。因此,通常将权重集称为筛选器(或内核),它与输入进行卷积。这个卷积的结果是一个激活映射,每个不同的过滤器的激活映射集合一起叠加在深度维度上,以产生输出量。参数共享有助于CNN架构的翻译不变性。有时,参数共享假设可能没有意义。在CNN的输入图像有一些特定的中心结构的情况下尤其如此,我们期望在不同的空间位置上可以学习到完全不同的特性。一个实际的例子是,当输入是以图像为中心的人脸时:我们可能会期望在图像的不同部分中学习不同的眼睛特定或特定毛发的特征。在这种情况下,放松参数共享方案是很常见的,而只是将其称为局部连接的层。

池化层(Pooling Layer):池化,这是一种非线性下采样。有几个非线性函数来实现池化,通常使用的池化操作为平均值池化和最大值池化,本文手写体识别应用采用的是最大值池化(max-pooling)。如果使用的卷积层相同, 但操作不相同。池化层就不需要包含其需要的学习参数。使用时仅需指定池化类型(平均值或最大值等)、池化操作的核大小和池化操作的步长等超参数即可。池化层在输入每个深度切片上是独立运行的,并在存储空间上动态调整其大小。最常见的形式是一个大小为2x2的滤波器,本文手写体识别应用采用的是3x3的滤波器。在输入每个矩阵上沿着高度与宽度,每个滤波器的步幅为2。在这种情况下,如果每个矩阵的最大操作超过4个数字。深度和维度将保持不变。池化函数除了最大值池化之外,还有平均值池化和L2范数池化。平均值池化通常已经不怎么使用了,一般分类问题中使用最多的是就是最大值池化,在实践中效果更好。

图2.4 使用2x2过滤器最大池化(步幅为2)

全连接层(Fully connected layer):最后,在经过卷积层和池化层之后,神经网络中高层部分的分类操作则要通过全连接层来完成。一个完全连接的层将前一层中的所有神经元(无论是完全连接的、共享的还是卷积的)连接到它所拥有的每一个神经元。全连接层不再是空间位置的了(可视化为成一维的),因此在完全连接的层之后就不会有卷积层。

2.2.4 第三方集成库

Anaconda 是一个python发行版,用于数据科学和机器学习相关应用(大规模数据处理,预测分析,科学计算)。具有安装和包管理工具,提供了大量的软件包和商业支持。同时也是一个环境管理器,提供了创建不同python环境的工具,每个环境都拥有自己独立的个性化设置,在数据科学平台上提供了更大的优势。

Keras 是一个用Python编写的开源神经网络库,用于深度学习。它能够在TensorFlow, Microsoft Cognitive Toolkit, Theano 或MXNet上运行。它的开发是为了使深度学习模型的实现尽可能快速和容易的进行研究和开发。它在Python 2.7或3.5上运行,并且可以在给定底层框架的gpu或cpu上无缝地执行。它是在麻省理工学院许可下发布的。Keras是由谷歌工程师Francois Chollet开发和维

护的，使用了四个指导原则：

1. 模块化：一个模型可以被理解为一个序列或一个图。深度学习模型的所有关注点都是可以任意组合的离散组件。
2. 极简主义：该库提供的内容仅够实现一个结果，没有不必要的装饰和可读性最大化。
3. 可扩展性：新组件在框架中有意添加和使用，目的是让研究人员尝试并探索新的想法。
4. Python：没有使用自定义文件格式的独立模型文件。一切都是原生Python文件。

Tensorflow 是一个开源免费软件库，用于跨一系列任务的数据流编程，专注于谷歌创建的机器学习。它是一个符号数学库，也用于机器学习应用，如神经网络等。TensorFlow最初作为Apache 2.0开源许可的一部分发布，最初由谷歌Brain Team的工程师和研究人员开发，主要用于内部使用。TensorFlow被认为是闭源应用程序DistBelief的继任者，目前谷歌用于研究和生产目的。TensorFlow被认为是专注于深度学习的框架的第一个重要实现。

Scikit-learn 是Python编程语言的免费软件机器学习库。它提供了大量常用算法的有效版本。Scikit-learn的特点是一个干净的、统一的、流线型的API，以及非常有用的、完整的在线文档。这种一致的好处是，理解Scikit-learn的基本用法和语法，切换到新的模型或算法就非常简单了。

NumPy 是一个Python包，表示的是“Numerical Python”。它是科学计算的核心库，包含一个强大的n维数组对象，为集成C、C++等提供了工具。它在线性代数、随机数能力等方面也很有用。

2.3 网站搭建技术

2.3.1 简介

服务器：本文手写体识别应用程序的服务器是采用阿里云CentOS7 x64位

域名地址：http:// banglong.site:5000，IP地址：101.132.172.41

Flask 是用Python编写的基于Werkzeug工具包和Jinja2模板引擎的微型网站框架。Flask的独特之处在于它是一个自定义的“微框架”，这意味着它背后的理念是为应用程序提供一个功能强大、简洁的核心，剩下的就交给开发人员了。Flask被认为是更轻量级的，更注重体验。Flask应用程序可以是单个Python文件，本文的手写体识别应用就是一独立的Python文件。

requests 是一个Python HTTP库，在Apache2许可证下发布。该项目的目标是使HTTP请求更简单，更人性化。它的口号是：“Requests: HTTP for Humans”。

AJAX即“Asynchronous Javascript And XML”（异步JavaScript和XML）。AJAX是在XML、HTML、CSS和Javascript的协同下创建更好、更快和更具交互性的web应用程序的一种新技术。Ajax使用XHTML处理内容，CSS处理表示，文档对象模型和JavaScript处理动态内容显示。传统的web应用程序使用同步请求向服务器发送和接收信息。这意味着您要填写一个表单，点击submit，然后使用来自服务器的新信息指向一个新页面。而使用AJAX时，单击submit，JavaScript将向服务器发出请求，解释结果，并更新当前屏幕。在最纯粹的意义上，用户永远不会知道任何东西甚至被传输到服务器。XML通常用作接收服务器数据的格式，尽管可以使用任何格式，包括纯文本。AJAX是一种独立于web服务器软件的web浏览器技术。当客户端程序从后台服务器请求信息时，用户可以继续使用应用程序。直观自然的用户交互。不需要单击，鼠标移动是一个足够的事件触发器。AJAX是由数据驱动的，而不是页面驱动的。

2.3.2 网站结构

本文所的网站程序结构如下图2.5网站结构，手写体识别应用运行在Flask服务端。其执行的顺序为：用户浏览器打开网址：banglong.site:5000，请求网页数据，并输入手写字符。Flask服务器端将用户画布里输入的图像转成base64格式的png，经过转置处理操作后再传给手写体识别应用。手写体识别应用拿Flask传过来的图像后，跟训练过的模型进行比较，得到预测结果传回Flask。Flask接收到返回预测结果再传回到网页端。

图2.5 网站结构

3 网站搭建

3.1 服务器与配置

本文手写识别程序的服务器为：CentOS7 x64，Python的版为：Python3.6.5。

Keras (2.0.4): 机器学习框架, 高层神经网络API。

Tensorflow (1.8.0): Google机器学习开源软件库。

Flask(0.12.2): 基于Python的微型网站框架。

urllib3 (1.22): Python HTTP库, 客户端程序访问服务器手写体识别应用时会使用到该库。

3.2 网站环境搭建

本文手写体识别程序网站服务器运行环境为: CentOS7 x64位, Python 3.6.5, Flask 0.12.2。

3.2.1 Python3.6

在CentOS7服务器中Python的默认版本是Python2.7的, 所以需要安装Python3.6。在CentOS7上使用源码安装的方法, 以下是安装步骤:

(1)从官网下载Python3.6的安装包

```
[rootCentOS7 ] wget https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tar.xz
--2018-06-04 22:19:51-- https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tar.xz
Resolving www.python.org (www.python.org) 151.101.72.223, 2a04:4e42:11::223
Connecting to www.python.org (www.python.org)|151.101.72.223|:443 connected.
HTTP request sent, awaiting response 200 OK
Length: 17049912 (16M) [application/octet-stream]
Saving to: 'Python-3.6.5.tar.xz'
```

(2)解压并安装Python3.6

解压:

```
[rootCentOS7 ] xz -d Python-3.6.5.tar.xz
[rootCentOS7 ] tar xvf Python-3.6.5.tar
```

配置(configure)、编译(make)、安装(make install):

```
[rootCentOS7 Python-3.6.5] ./configure --prefix=/usr/local/python3
[rootCentOS7 Python-3.6.5] make make install
```

其中配置时, -prefix=/usr/local/python3选项是设置Python3的安装路径为/usr/local/python3。

(3)测试Python3是否安装成功

```
[rootCentOS7 Python-3.6.5] python
Python 2.7.5 (default, Apr 11 2018, 07:36:10)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

3.2.2 虚拟环境搭建

安装完Python3后, 还需要通过Python3为本文的手写体识别程序设置Python3的虚拟环境。本文采用的是Python3.5原生的venv模块来创建Python3的虚拟环境, 其提供了与CentOS系统中的Python2的隔离操作, 每一个所创建的虚拟环境都拥有自己的Python版本和独立的Python包。手写体识别程序单独使用一个虚拟环境, 不会影响到其他项目的环境, 也不会影响到CentOS服务器中的Python2与Python3, 实现了Python2与Python3共存。

3.2.3 Flask网站框架安装

通过使用上一小节中Python3创建的venv虚拟环境。

(1)激活虚拟环境并查看Python版本

```
[rootCentOS ] source ENV/bin/activate
```

(ENV) [rootCentOS7] python

Python 3.6.5 (default, May 26 2018, 15:22:34)

[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux

Type "help", "copyright", "redits" or "license" for more information.

(2)安装Flask 0.12.2

(ENV) [rootCentOS7] pip install Flask==0.12.2

(3)测试Flask是否安装成功

首先创建并写入以下代码：

(ENV) [rootCentOS7] cat hello.py

from flask import Flask 导入Flask

app = Flask(name)

app.route('/') app.route是flask中的路由

def helloworld():

return 'Hello Flask!'

if name == 'main':

app.run() 函数run() 启动本地服务器

然后运行程序：

(ENV) [rootCentOS7] python hello.py

Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

测试结果：

[rootCentOS7] curl 127.0.0.1:5000

Hello Flask!

屏幕上成功输出Hello Flask!则Flask安装完成。

3.4 网站工作流程

Flask网站基本运行流程如下图：

图3.1 网站运行流程

如上图所示，Flask网站是由服务器，客户端和应用程序，客户端与服务器之间通过是通过TCP/IP协议中的HTTP互相访问的。而应用程序是要搭建在服务器之上的，因此，几乎所有的Python Web框架都是采用WSGI(Web Server Gateway Interface)服务器网关接口来实现应用程序与服务器之间通信。本文的手写体识别程序就是由Flask创建的app。应用程序要跑起来，就必须通过服务器。这里的服务器可以是Apache、nginx、gunicorn等。

WSGI规定了应用程序与服务器之间通过所用到的接口。服务器调用应用程序时，会传两个参数至应用程序：environ 这是要请求的信息，startresponse是应用程序在处理运算完后调用的函数，参数是状态码，响应头部与错误信息。

本文服务器端所采用的是在服务器端Flask + gunicorn 搭建起来的。

3.5 网站运行结果

使用浏览器打开网址：http://banglong.site:500，其终端使用电脑和手机访问时结果如图3.1和图3.2所示，便已成功运行手写体识别应用程序。

图3.1 图3.2

电脑端访问 手机端访问

4 手写数字/英文字母图像识别实现

通过查阅资料，发现手写数字/英文字母图像识别是属于图像识别学科下的分支，是图像处理与模式识别研究领域的重要应用之一。但由于手写数字/英文字母的随意性很大，是由人为书写的，如字体的笔画粗细，字体大小，字的倾斜角度等等因素都有可能直接影响到字符的识别准确率，所以手写体数字/英文字母识别是一个有挑战性的课题。在过去的数十年中，许多研究者和前辈提出了许多传统识别方法，并且取得了一定的成效。而且数字/英文字母识别在大规模数据统计如全国的身份证上的姓名，性别，民族，出生日期、全国人口普查、公司的财务与税务、邮件和信封上的分拣等应用领域都有着广阔的应用前景。

不同于传统的图像识别，本文所采用的手写数字/英文字母图像识别，是基于经典的卷积神经网络算法，通过训练标准的样本数据，以达到可以识别手写体的应用。

两次实验分别实现了手写数字和手写英文字母识别，并对手写数字/英文字母识别的方法进行了简要介绍和分析。本文所实现的手写数字/字母识别程序，基于CNN（卷积神经网络）算法，并使用了5层神经网络，运行在Flask网站构架平台下，具有手写数字/英文字母图像读取、特征提取及识别功能。

4.1 手写数字/英文字母的数据准备

手写数字MNIST的数据集来自于<http://yann.lecun.com/exdb/mnist/>，由Google实验室的Corinna Cortes和纽约大学柯朗研究所的Yann LeCun共同建立的一个手写数字数据库，训练库中共有60,000张手写数字图像，测试库中有10,000张。

—— mnist

—— t10k-images-idx3-ubyte.gz: test set images (1648877 bytes)

—— t10k-labels-idx1-ubyte.gz: test set labels (4542 bytes)

—— train-images-idx3-ubyte.gz: training set images (9912422 bytes)

—— train-labels-idx1-ubyte.gz: training set labels (28881 bytes)

EMNIST数据集来自<https://www.westernsydney.edu.au/bens/home/reproducibleresearch/emnist>，由悉尼大学Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik共同建立。MNIST数据集已经成为学习、分类和计算机视觉系统的标准基准。促进其广泛采用的是任务的可理解和直观的性质，其相对较小的大小和存储要求以及数据库本身的可访问性和易用性。

EMNIST数据库来自一个更大的数据集，称为NIST特殊数据库19，它包含数字、大写和小写的手写字母。

4.2 手写识别应用程序实现

4.2.1 加载数据并训练数据（返回模型并估模型）

数据集文件：

图4.1数据集文件

emnist-byclass.mat 英文字母EMNIST数据集文件已分类

mnist.mat 手写数字数据集文件

首先查看emnist中的数据，先打印里面图像的维度。

print(trainingimages.shape)，输出 (697932, 28, 28, 1)，共97932张训练图像，其维度为一个黑白通道的28x28的像素的图像，以矩阵方式存储在内存中。

再查看训练图中的一张图像的例子，先打第23号图像和对应的编号：

print(display(trainingimages[23]), traininglabels[23])，其输出见下图4.2：

，。

图4.2 数字3储存方式和对应的编号

该数据的标签是数字3，图像的数据维度为28x28的矩阵，矩阵上的每个数值代表的是每个像素点的灰度值。

关键代码：

加载mat格式数据集

```
mat = loadmat(matfilepath)
```

加载训练数据样本

```
if max == None:
```

```
max = len(mat['dataset'][0][0][0][0][0][0]) 求出数据的数据集中图像个数
```

```
trainingimages = mat['dataset'][0][0][0][0][0][0][:max].reshape(max, height, width, 1) 原始图像
```

```
traininglabels = mat['dataset'][0][0][0][0][0][1][:max] 数据的标签
```

加载测试数据样本

```
if max == None:
```

```
max = len(mat['dataset'][0][0][1][0][0][0]) 求出数据的数据集中图像个数
```

```
testingimages = mat['dataset'][0][0][1][0][0][0][:max].reshape(max, height, width, 1) 原始图像
```

```
testinglabels = mat['dataset'][0][0][1][0][0][1][:max] 数据的标签
```

其中数字和英文字母所对应的特征值数量是不同的，数字的特征值数量为10个。映射的格式为：字典：{key:value}，其中key代表的序号，value是对应的ASCII值，如序号为3的对应的ASCII为51，代表的是数字3。{0: 48, 1: 49, 2: 50, 3: 51, 4: 52, 5: 53, 6: 54, 7: 55, 8: 56, 9: 57}

英文字母有62个为大写A-Z的序号10至35(26个)，小写a-z序号36至61(26个)，数字的序号0至9（10个）。映射的方式和上面的数字是一样的。例如序号为5对应的ASCII值为53，代表的是数字5。序号为28的对应的ASCII值为82，代表的是大写字母：R。

{0: 48, 1: 49, 2: 50, 3: 51, 4: 52, 5: 53, 6: 54, 7: 55, 8: 56, 9: 57, 10: 65, 11: 66, 12: 67, 13: 68, 14: 69, 15: 70, 16: 71, 17: 72, 18: 73, 19: 74, 20: 75, 21: 76, 22: 77, 23: 78, 24: 79, 25: 80, 26: 81, 27: 82, 28: 83, 29: 84, 30: 85, 31: 86, 32: 87, 33: 88, 34: 89, 35: 90, 36: 97, 37: 98, 38: 99, 39: 100, 40: 101, 41: 102, 42: 103, 43: 104, 44: 105, 45: 106, 46: 107, 47: 108, 48: 109, 49: 110, 50: 111, 51: 112, 52: 113, 53: 114, 54: 115, 55: 116, 56: 117, 57: 118, 58: 119, 59: 120, 60: 121, 61: 122}

本文所采用的卷积神经网络架构：共9层，分别是1输入层、2层卷积层、1池化层、1 Dropout层、1 Flatten层、1全连接层、1 Dropout层、1全连接层。

图4.3 卷积神经网络构建

对图4.3卷积神经网络构建中的各层说明：

(1)输入层：预处理后的2828灰度图像数据，无权重和偏置；

(2)卷积层：卷积核大小33，输出特征图数量32，卷积核种类132=32，输出特征图大小2626

Tensor("conv2d1/Relu:0", shape=(?, 26, 26, 32), dtype=float32)

(3)卷积层：卷积核大小33，输出下特征图数量32，卷积核种类32，输出特征图大小2424

Tensor("conv2d2/Relu:0", shape=(?, 24, 24, 32), dtype=float32)

(4)池化层：池化核大小22，将使图片在两个维度上（竖直，水平）均变为原长的一半，输出下采样数量1212。

Tensor("maxpooling2d1/MaxPool:0", shape=(?, 12, 12, 32), dtype=float32)

(5)Dropout层：设置神经元随机失活概率为0.25，输出1212

Tensor("dropout1/cond/Merge:0", shape=(?, 12, 12, 32), dtype=float32)

(6)Flatten层：将输入的图像数据压成一维数据

Tensor("flatten1/Reshape:0", shape=(?, ?), dtype=float32)

(7)全连接层：卷积核大小11，输出512（自己设置的）

Tensor("dense1/Relu:0", shape=(?, 512), dtype=float32)

(8)Dropout层：设置神经元随机失活概率为0.25，输出512

Tensor("dropout2/cond/Merge:0", shape=(?, 512), dtype=float32)

(9)全连接层：输出特征图数量62（手写英文字母，手写数字为：10）

Tensor("dense2/Softmax:0", shape=(?, 62), dtype=float32)

4.2.2 构建和训练卷积神经网络

构建顺序模型

```
model = Sequential()
```

''' 输入图像数据的shape, 默认数据为(28, 28, 1) 第一个卷积层需接受一个输入图像数据的参数, 而后面各层可以自动推导出中间图像数据的shape '''

首先, 第一个卷积层, 构造32个滤波器, 每个滤波器覆盖范围是 3x3

滤波器的挪动步长为1, 其图像四周补一圈0, 并用relu进行非线性变换

```
model.add(Convolution2D(nbfilters, kernelsize, strides=(1, 1), padding='valid', inputshape=inputshape,
activation='relu'))
```

```
model.add(Convolution2D(nbfilters, kernelsize, activation='relu'))
```

添加一层 MaxPooling, 在2x2的格子中取最大值

```
model.add(MaxPooling2D(poolsize=poolsize))
```

设立Dropout层, 将Dropout的概率设为0.25

```
model.add(Dropout(0.25))
```

神经元随机失活

把当前层节点展平, 拉成一维数据, 才能全连接

```
model.add(Flatten())
```

全连接层1

```
model.add(Dense(512, activation='relu'))
```

```
model.add(Dropout(0.5))
```

随机失活

全连接层2, Softmax评分

```
model.add(Dense(nbcasses, activation='softmax'))
```

'''

最后编译模型, 并设置模型的学习过程

1. loss损失函数: 目标函数, 可为预定义的损失函数

2. optimizer优化器: 参数可指定为预定义的优化器名字, 如rmsprop、adagrad或一个Optimizer类对象

3. 指标列表: 如果是分类问题, 则一般该列表参数为metrics=['accuracy']

'''

```
model.compile(loss='categorical_crossentropy',
```

```
optimizer='adadelta',
```

```
metrics=['accuracy'])
```

4.2.3 评估模型

手写数字模型评估:

score = model.evaluate(xtest, ytest) 模型评估, 基于测试样本评价模型的准确度

D:\Program Files\Anaconda3\python.exe" I:/B毕设/EMNIST/training.py -f ./matlab/emnist-mnist.mat

Using TensorFlow backend.

Train on 60000 samples, validate on 10000 samples

60000/60000[=====] - 139s 2ms/step - loss: 0.0184 -acc: 0.994 -valloss:0.0.250 - valacc: 0.9919

Test score: 0.025195021106

Test accuracy:0.9919

60000 样本数据, 10000测试数据。迭代10次, 准确率达到99.19。

手写英文字母模型评估:

D:\Program Files\Anaconda3\python.exe" I:/B毕设/EMNIST/training.py -f ./matlab/emnist-byclass.mat

Using TensorFlow backend.

Train on 697932 samples, validate on 116322 samples

Epoch 1/10

697932样本数据, 116322测试数据。迭代10次,

4.2.4 网页端识别手写数字/英文字母

一、加载训练过的模型

载入 YAML 和创建模型

```
yamlfile = open('%s/model.yaml' % bindir, 'r')
```

```
loadedmodelyaml = yamlfile.read()
```

```
yamlfile.close()
```

```
model = modelfromyaml(loadedmodelyaml)
```

将已训练过的模型加载

```
model.loadweights('%s/model.h5' % bindir)
```

二、网页端访问

首先运行网站应用程序:

```
(ENV) [rootCentOS7 EMNIST-master] python server.py --host 0.0.0.0
```

Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)

浏览器输入网址: http://banglong.site:5000, 如下图4.3

图4.3 手写体图像识别

三、预测结果

网页端的画布图像使用Ajax传到Flask (JS代码), 见图4.4

图4.4 网页端绘制手写图像

```
("predictbutton").click(function() { //单击识别按钮
```

```
var canvasObj = document.getElementById("canvas");
```

```
var img = canvasObj.toDataURL('image/png');
```

//canvas提供了一个重要的方法toDataURL(), 这个方法能把画布里的图案转变成base64编码格式的png, 然后返回 Data URL数据。

```
.ajax({
```

```
type: "POST",
```

```
url: "/predict/",
```

```
data: img,
```

```
success: function(data) {
```

```
('result').text('预测: ' + data.prediction);
```

```
('confidence').text('相似度: ' + data.confidence + '%');
```



```
}
});
});
```

解析通过Ajax使用POST过来的图像数据

```
def parseImage(imgData):
```

将画布base64位格式的图像保存成output.png

```
imgstr = re.search(b'base64, (. )', imgData).group(1) 使用正则匹配
```

```
with open('output.png', 'wb') as output:
```

```
output.write(base64.decodebytes(imgstr))
```

从网页记录绘图并保存成output.png图像，见下图4.4

```
parseImage(request.getdata())
```

图4.5 网页端输入的图像保存成output.png

以每8位黑白模式（L）读取分析图像

```
x = imread('output.png', mode='L')
```

x = np.invert(x) 将图像数据转置，原图像是白底黑色，转为黑底白色

将已转置过的图像保存成resized.png见图4.6

```
imsave('resized.png', x)
```

图4.6 output.png转置成黑底白字的图像

```
x = imresize(x, (28, 28))
```

原始图像为280280，8bits重构图像数据维度成一维的2828，1通道，用于神经网络

```
x = x.reshape(1, 28, 28, 1)
```

```
x = x.astype('float32')
```

统一格式

```
x /= 255
```

从模型中预测结果

```
out = model.predict(x) 输出out
```

```
[[ 8.99958536e-07  3.87457334e-07  1.23956556e-06  9.16228890e-04
```

```
1.73078922e-06  9.634920e-01  2.03575291e-05  4.30803411e-08
```

```
1.17835616e-05  3.39387945e-04]]
```

生成Json格式的相似度最大的结果并返回给网页端，见图4.7

```
response = {'prediction': chr(mapping[(int(np.argmax(out, axis=1)[0]))]),
```

```
'confidence': str(max(out[0]) 100)[:6]}
```

```
return jsonify(response)
```

图4.7 手写识别结果

四、数字手写体识别结果：

图4.3 手写数字识别

五、英文字母识别结果：

图4.4 手写英文字母识别

5 总结与展望

目录, 人工智能已潜移默化的渗入到我们生活的各个方面, 图像识别技术已被广泛运用于指纹, 人脸, 车辆牌照, 手写体数字、邮政编码、汽车牌照、汉字、条形码识别等领域。

本文使用Keras作为卷积神经网络框架, 后端采用Tensorflow进行网络卷积运算, 数据集采用MNIST (YannLeCun建立的手写数字数据库集)和EMNIST(Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik共同建立的数字, 小写和大写字母数据集)。识别实验采用Flask网站框架和相关神经网络技术对手写数据进行预测。

作为刚接触到图像识别的初学者, 自知本身还需多多学习, 文本中的实验还有提高的空间, 其卷积神经网络算法还要进一步完善, 而且数字和英文字母有时不易区分, 如写了一个数字0, 计算机并不知道这是数字0还是字母大写的o, 所以文字识别要结合单词和语意进行预测才能达到较好的期望值。作为使用网站做平台实现神经网络, 后期可以采用多并发的, 高性能的nginx服务。

脚踏实地, 认真勤奋, 求真务实的学习态度, 面对困难, 积极应对, 并寻求老师与同学的帮助。我认为在撰写本文时, 我收获到了有关论文写作的规则和框架技巧。通过本次论文的设计, 学到了不仅仅是对卷积神经网络的单一方面的认识与了解, 更多的是思路, 对问题处理的方法。指导老师学识渊博, 低调的人为处事与谦虚认真的态度, 我非常敬指导我的老师。对于未来的道路, 充满无限的可能性, 作为刚要走入社会的毕业生, 深知专业知识的重要性, 在写本文过程中, 我学习到了新的知识与技能, 并增长了见识, 在大学四年所学的专业知识仍然有限, 像计算机这种专业, 每天都有新的技术更新, 旧的就会被时代所淘汰, 社会是需要自学能力强的, 今后, 自己的会朝着所喜欢的事情去奋斗, 去努力不断的充实自己。

参考文献

- [1] 孔斌;人类视觉与计算机视觉的比较[J];自然杂志;2002年01期
- [2] 邹明福;钮兴显;刘昌平;白洪亮;;联机手写英文识别[J];计算机研究与发展;2006年01期
- [3]] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278 - 2323, 1998.
- [4] 户保田;基于深度神经网络的文本表示及其应用[D];哈尔滨工业大学;2016年
- [5] 冯子勇;基于深度学习的图像特征学习和分类方法的研究及应用[D];华南理工大学;2016年
- [6] 王强. 基于CNN的字符识别方法研究[D]. 天津师范大学, 2014.
- [7] 李卫. 深度学习在图像识别中的研究及应用[D]. 武汉理工大学, 2014.
- [8] 刘荣荣. 基于卷积神经网络的手写数字识别软件的设计与实现[D]. 内蒙古大学, 2015.
- [9] 王瑞. 基于卷积神经网络的图像识别[D]. 河南大学, 2015.
- [10] 李明威. 图像分类中的卷积神经网络方法研究[D]. 南京邮电大学, 2016.
- [11] 王斐. 卷积神经网络在手绘草图识别中的应用研究[D]. 安徽大学, 2016.
- [12] 江帆, 刘辉, 王彬, 等. 基于CNN-GRNN模型的图像识别[J]. 计算机工程, 2017, (4).doi:10.3969/j.issn.1000-3428.2017.04.044.
- [13] 段萌. 基于卷积神经网络的图像识别方法研究[D]. 郑州大学, 2017.
- [14] 吴正文;卷积神经网络在图像分类中的应用研究[D];电子科技大学;2015年
- [15] 何西麟;基于深度学习的手写体字符识别研究与实现[D];中山大学;2015年
- [16] 陈先昌;基于卷积神经网络的深度学习算法与应用研究[D];浙江工商大学;2014年
- [17] 冯方向;基于深度学习的跨模态检索研究[D];北京邮电大学;2015年
- [18] 张清辰;面向大数据特征学习的深度计算模型研究[D];大连理工大学;2015年

致谢

本次毕业设计的基于卷积神经网络的图像识别程序和论文都已经完成, 首先要向我的指导老师王寅同和专业课老师们表示衷心的感谢。论文的顺利完成, 要归功于指导老师与专业老师们的悉心指导, 正是他们的不断的督促、指导以及关怀, 我才能按时完成。我从尊敬的导师身上, 我学习了的不仅仅是扎实与深广的专业知识与技能, 更学会的是做人与处理的道理, 在此我要向我的导师致以最衷心的感谢和深深的敬意。

其次，我要感谢与我相处一起学习的同学们，遇到问题时我都会与大家一起讨论，分析问题的原因，找出解决的办法，正是因为由于你们的存在，我才能顺利的完成基于卷积神经网络的图像识别研究的论文。我还要感谢论文中涉及的学者，从他们的研究成果中我得到了帮助和启发，否则我将难以完成本文。

最后，我要感谢学院的所有的老师们，感激四年来你们对我的关怀和教育，如果不是你们这四年来不断的授业传道解惑，我不会成长的像今天这样的优秀，是你们让我获取了人生最宝贵的财富，领悟了知识的力量，感悟到了人生的真谛，真的很感谢你们。

大学的时光即将结束，感谢这四年来在我生活中出现的所有人，是你们让我的大学生活充满了色彩，是你们让我体会了人生的酸甜苦辣，感谢大家。

Flask服务端

输入图像数据

请求网页数据，手写数据

手写体识别应用

客户端浏览器

返回预测结果

返回预测结果

WSGI

客户端

手写体识别

服务器

应用接口

请求

响应

调用函数信息

1. 输入图像(维度: 28281)

卷积核个数: 32

卷积核大小: (3, 3)

输出:2626, 32

2. 卷积层

卷积核个数: 32

卷积核大小: (3, 3)

输出:2424, 32

3. 卷积层

4. 池化层

池化核大小: (2, 2)

输出:1212, 32

rate: 0.25 训练时每次更新参数按概率随机断开输入单元，防止过拟合。

5.Dropout

6.Flatten

输出: 一维数据

7.全连接层

8. Dropout

输出512

rate: 0.5

输出:

手写数字: 10个特征值

手写英文: 62个特征值

9. 全连接层

PAGE

• 声明:

报告编号系送检论文检测报告在本系统中的唯一编号。

本报告为维普论文检测系统算法自动生成, 仅对您所选择比对资源范围内检验结果负责, 仅供参考。

客服热线: 400-607-5550 | 客服QQ: 4006075550 | 客服邮箱: vpcs@cqvip.com

唯一官方网站: <http://vpcs.cqvip.com>



关注微信公众号