信息工程学院校企实训

分组项目报告

|  |  |
| --- | --- |
| 项目名称 | 基于CNN的手写算式识别与计算 |
| 项目负责人 | 郭斌 |
| 专业 | 软件工程 |
| 任课教师 | 胡英铭 |
| 编制时间 | 2019年7月9日 |

西北农林科技大学信息工程学院

2019年6月制

项目成员及分工

（工作量总和为 100%）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 学号 | 姓名 | 工作内容 | 工作量（%） |
| 01 | 2016012931 | 武杰 | 项目分工与项目报告编写 |  |
| 02 | 2016012933 | 彭兆前 | Web前端设计与实现 |  |
| 03 | 2016012934 | 张胜君 | 图像处理 |  |
| 04 | 2016012935 | 郭斌 | CNN模型设计与训练 |  |

项目总打分表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指标内容 | 分 值 | 指标内涵及评估标准 | | | | 得 分 |
| A | B | C | D |
| 构思（20 分） | | | | | |  |
| 选题意义 | 10 | 意义重大 | 意义较大 | 意义一般，属于简单开发 | 无意义 |  |
| 技术路线的可行度 | 10 | 合理可行，具体且有创新 | 合理可行,具体 | 基本合理可行 | 不够合理 |  |
| 设计（20 分） | | | | | |  |
| 设计内容 | 10 | 内容非常丰富 | 内容较丰富 | 内容一般 | 内容欠缺 |  |
| 解决的关键技术问题 | 10 | 准确，范围合 适，重点突出 | 基本准确 | 部分关键 | 未抓住关键 |  |
| 实现（20 分） | | | | | |  |
| 项目完成的技术水平 | 10 | 难度很大，达到较高水平 | 难度较大，超出一般水平 | 难度一般，达 到普通水平 | 难度小，很容易实现 |  |
| 团队精神 | 10 | 团队合作精神 强 | 合作情况良好 | 合作情况一般 | 合作不好 |  |
| 运作（20 分） | | | | | |  |
| 应用程度 | 20 | 产业化或实际 应用 | 模拟应用 | 可以应用但未应用 | 不能应用 |  |
| 文字表达及文档制作水平（10 分） | | | | | |  |
| 文字表达 | 5 | 文字表达非常 好 | 文字表达较好 | 文字表达一般 | 文字表达差 意思不明了 |  |
| 文档制作 | 5 | 制作非常专业 化 | 制作良好 | 制作一般 | 制作效果差 |  |
| 报告质量及口头表达能力（10 分） | | | | | |  |
| 报告质量 | 5 | 报告非常完整 | 报告比较完整 | 完整程度一般 | 报告不完整 |  |
| 口头表达能力 | 5 | 整体效果很好 | 整体效果良好 | 整体效果一般 | 整体效果差 |  |
| 总分： | | | | | |  |
| 评语： | | | | | | |

个人期末项目成绩表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 学号 | 年级 | 专业 | 姓名 | 成绩 |
| 01 | 2016012931 | 2016 | 软件工程 | 武杰 |  |
| 02 | 2016012933 | 2016 | 软件工程 | 彭兆前 |  |
| 03 | 2016012934 | 2016 | 软件工程 | 张胜君 |  |
| 04 | 2016012935 | 2016 | 软件工程 | 郭斌 |  |

说明：

1. 第二阶段实训任务及报告完成情况＝个人期末项目成绩＝小组总成绩×小组人数×个人工作量＋教师调控（范围－10～＋10）
2. 个人学期总成绩＝劳动纪律和实训态度×20%＋第一阶段实训任务及报告完成情况×40%＋第二阶段实训任务及报告完成情况×40%

# 项目背景

当前主流搜题软件如小猿搜题、作业帮等通过用户拍照，对图片中的字符进行识别，在软件自身题库中进行搜索。用户在搜题过程中，部分题目是是对包含加减乘除的算式或方程进行求解计算，如果能对算式直接进行识别，并求解计算，将不需要搜索软件自身题库，不再受题库限制，软件直接为用户输出结果，为用户节省了获取结果的时间。

# 项目概述

实训中学习了利用CNN方法完成了对0-9手写数字的识别模型的构建，通过使用MINIST数据集完成了对模型的训练，经过测试，识别准确率达到90%以上。

本项目，在完成前期0-9手写数字识别的基础上，对程序功能进行了扩展。手写数字识别程序，可以识别0-9共计10个数字，并对识别率和识别结果进行输出。

项目通过对原有的MINIST数据集进行扩展，加入手写的“+”、“-”、“×”、“÷”、“（”、“）”等数据，扩展模型可识别的字符，使模型能够对通过Web前端输入的手写的“+”、“-”、“×”、“÷”、“（”、“）”等符号进行识别。通过OpenCV定位算式位置，通过像素对单个字符进行分割，并将每个字符补充至28\*28像素，对每个字符逐一识别，并将识别的字符连接成算式，最后通过python中的eval函数对算式字符进行转换，并对相应的计算结果进行输出。

用户也可以通过拍照的形式，在Web前端上传拍摄的算式图片，通过OpenCV获取到前端上传的图片，使用高斯过滤去除噪声，将图像灰度化，进行二值化处理，通过像素对单个字符进行分割，并将每个字符补充至28\*28像素，对每个字符逐一识别，并将识别的字符连接成算式，最后通过python中的eval函数对算式字符进行转换，并对相应的计算结果进行输出。

# 项目分析

1. 系统架构

本项目共分为四个子程序，分别为Web前端输入输出程序、图像处理程序、CNN图片识别程序、算式计算程序。

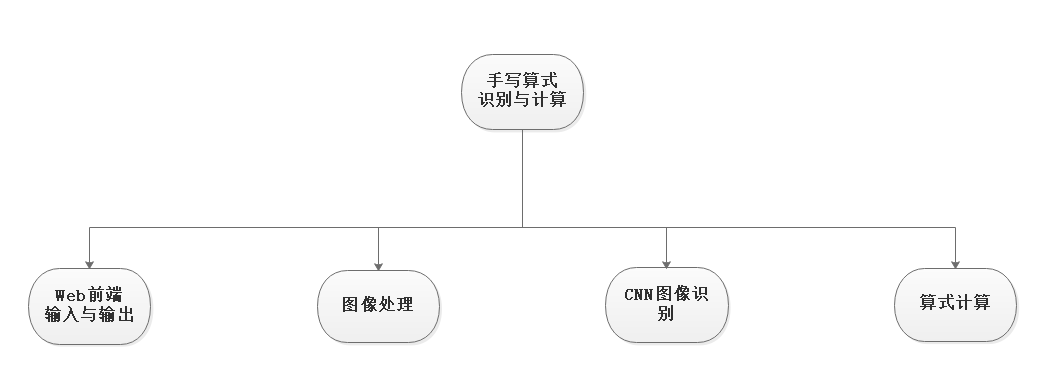


图 1 系统架构

1. 可行性分析
2. 问题定义

当前主流搜题软件如小猿搜题、作业帮等通过用户拍照，对图片中的字符进行识别，在软件自身题库中进行搜索。用户在搜题过程中，部分题目是是对包含加减乘除的算式或方程进行求解计算，如果能对算式直接进行识别，并求解计算，将不需要搜索软件自身题库，不再受题库限制，软件直接为用户输出结果，为用户节省了获取结果的时间。

同时，对这部分算式计算题目的存储需要耗费服务器大量存储空间，通过程序自身对算式实现计算功能，只需要将后台部署在服务器，便可以实现大部分算式的识别与计算，节省了服务器存储空间。

1. 技术可行性

本程序采用B/S模式开发。B/S模式是指在TCP/IP的支持下，以HTTP为传输协议，客户端通过Browser访问Web服务器以及与之相连接的后台程序的技术及体系结构。客户端的浏览器通过URL访问Web服务器，Web服务器请求应用服务器，并将获得的结果以HTML形式返回客户端浏览器。

Web端将为用户提供上传图片或手写算式、显示识别结果和计算结果的功能。用户可以上传算式照片或通过HTML5 CANVAS绘制手写算式传递到后台服务器，后台服务器根据上传的图像做出相应的处理。

后台服务器获取到图像后，若用户上传的是算式照片，则要对图像使用高斯过滤进行去噪，然后对照片灰度化、二值化，得到可进行像素分割的图像；若用户选择的是在HTML5 CANVAS绘制的手写算式，不需要进行去噪，照片经灰度化、二值化后，得到可进行像素分割的图像。分割后的照片进行像素填充后，每张图像变为28\*28像素，将这些图像数据合并成一个N\*784的数组，传递给卷积神经网络进行识别，最后将识别结果返回给Web前端，显示给用户。

本程序将Python Web技术与当前热门的人工智能相结合，利用此前所学的内容，能够为用户提供所需的基本功能。

1. 需求分析
2. 结构化分析
3. 顶层数据流图

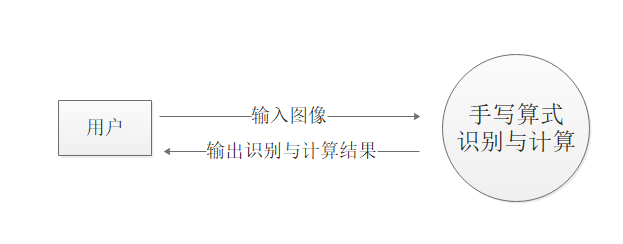


图 2 顶层数据流图

1. 0层数据流图：

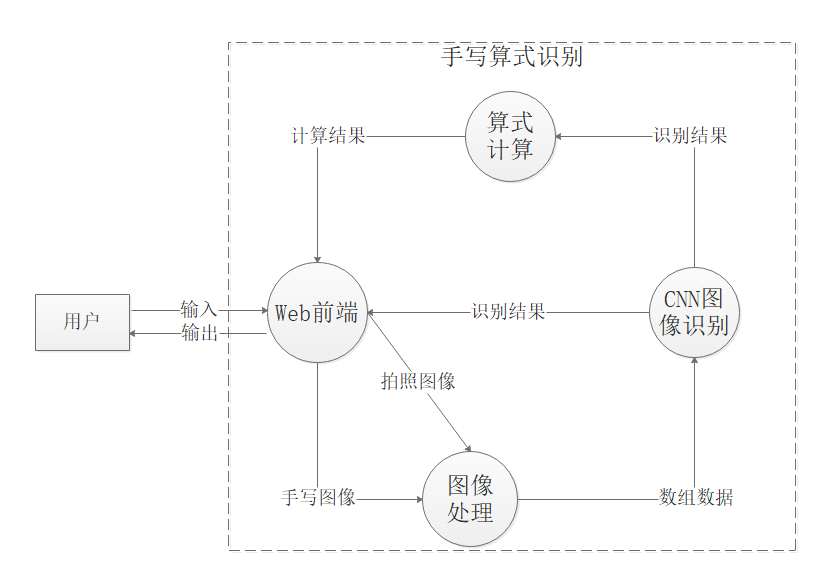


图 3 0层数据流图

1. 1层数据流图：

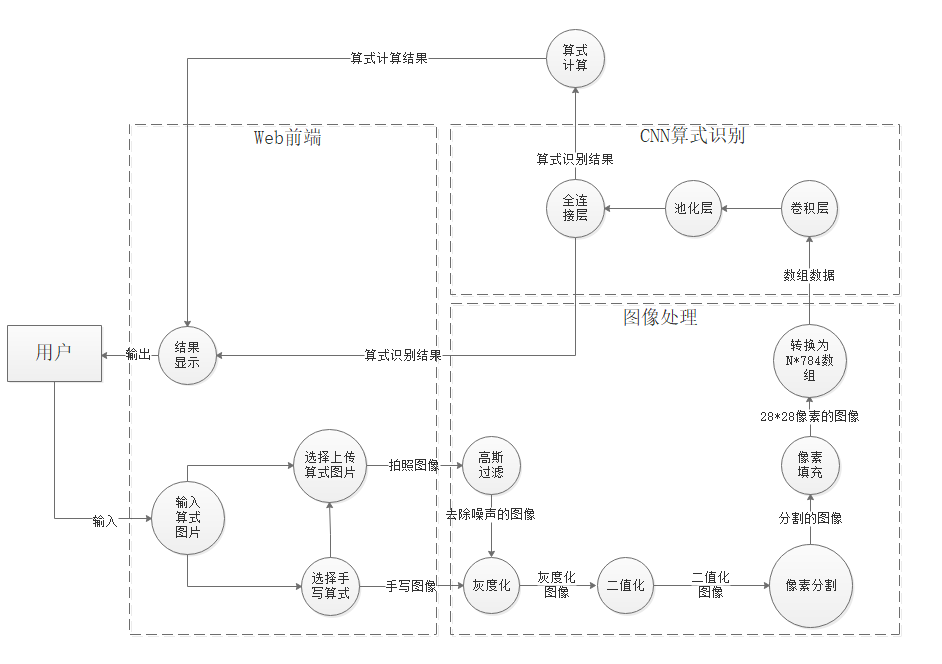


图 4 1层数据流图

1. 功能性需求
2. 为用户提供可交互的可视化的Web前端窗口。
3. 从Web前端获取用户通过鼠标绘制的算式图片。
4. 从Web前端获取用户上传的本地算式图片。
5. 对Web前端获取的算式图片按字符进行分割，获取到分割后的多张图片。
6. 对分割后的每张图片进行处理，生成28\*28像素图片。
7. 对生成的每张图片进行识别。
8. 将每张图片的识别结果存储在内存中。
9. 利用内存中的图片识别结果，进行算式的计算。
10. 将算式识别结果返回给用户。
11. 将算式计算结果返回给用户。
12. 非功能性需求
13. Web前端的美观性。
14. Web前端对用户的错误操作给予提示。
15. 对CNN模型进行训练时，具备数量巨大的训练数据。
16. 对CNN模型进行训练时，具有高效的训练效率。
17. 设计约束
18. 打开该程序的浏览器需支持Html5 Canvas。
19. 需要具备Python运行环境
20. 项目规划
21. 项目开发计划
22. 项目对硬件、软件的资源需求

硬件资源需求：

CPU：i7-6700

RAM: 16GB

ROM: 20GB

软件资源需求：

IDE： pycharm-professional-2019.1.3

Python集成开发环境：Anaconda3

相关开发框架或包：

表格 1 开发框架和包

|  |  |
| --- | --- |
| **Package** | **Version** |
| Django | 2.2.2 |
| h5py | 2.9.0 |
| Keras-Applications | 1.0.8 |
| Keras-Preprocessing | 1.1.0 |
| kiwisolver | 1.1.0 |
| matplotlib | 3.1.0 |
| numpy | 1.16.4 |
| opencv-python | 4.1.0.25 |
| Pillow | 6.0.0 |
| tensorflow | 1.14.0 |
| tqdm | 4.32.2 |

1. 项目开发模型

程序分模块实现，符合增量模型的特点。运用增量模型的软件开发过程是递增式的过程。本项目中，对图片的分割和识别为核心功能部分。开发时， Web前端、图像处理、CNN算式识别模块同时进行开发，图像处理初步需实现对图像的灰度化、二值化和图像分割功能，CNN算式识别需实现对各类字符的识别，后期结合Web前端，将三个模块进行合并，逐步完善程序功能，以达到预期的需求。

1. 项目进度表

|  |  |  |
| --- | --- | --- |
| 日期 | 工作内容 | 完成情况 |
| 2019/7/1 | 小组分工，需求分析。 | 完成小组分工，初步完成可行性分析、需求分析。 |
| 2019/7/2 | 制定项目计划，查询各项分工所需的资料，编写项目报告中相应部分。 | 对程序进行模块设计，进一步完善需求分析，制定项目计划，并查阅数据集扩展、图像处理等有关内容的资料。 |
| 2019/7/3 | 按照分工开发各模块，实现各模块的基本功能。 | 初步确定各模块的开发技术手段和实现方法，开始对各模块进行开发。 |
| 2019/7/4 | 按照分工开发各模块，实现各模块的基本功能。 | 各模块功能完善，漏洞修复。 |
| 2019/7/5 | 按照分工开发各模块，完善、增强各模块的功能。 | 各模块开发基本完成。 |
| 2019/7/8 | 各模块整合，测试程序整体功能。 | 模块整合完成，对整合过程中出现的接口问题进行调整、修改、完善。 |
| 2019/7/9 | 测试程序整体功能性，对程序进行完善修改。 |  |
| 2019/7/10 | 测试程序整体功能性，对程序进行完善修改。 |  |
| 2019/7/11 | 测试程序整体功能性，对程序进行完善修改。  准备项目答辩内容，进行PPT制作。 |  |
| 2019/7/12 | 项目答辩，程序改进。 |  |

# 项目实现

1. Web前端

运用Django来实现网站的框架，生成一个网站app，在settings.py文件中进行网站的设置，在static文件夹中存放静态文件，在template文件夹中存放网页模板，在urls.py文件中进行路由配置，在views.py文件中编写逻辑、处理用户发送的请求；运用jQuery，layer来实现对网页元素的操作及前后端的对接，实现动态网页；用boostrap来进行网页样式的设计，美化按钮等页面元素。

1. 图像处理

运用opencv对图片进行图像处理，用opencv的GaussianBlur函数对图片进行高斯过滤去除噪点，用opencv的cvtColor函数将图片转换为灰度图，用opencv的threshold函数将图片二值化，用opencv的findContours函数获取到手写算式的笔迹，再用opencv的boundingRect函数将每一部分笔迹框起来，然后对这些方框进行处理，将方框列表进行分行，提取除号的操作，然后将方框中的内容截取下来，添加空白边框使字符居中，将其尺寸用opencv的resize函数设为28\*28，再将这些图片转化成n\*784的数组传给卷积神经网络。

1. CNN算式识别

首先对mnist数据集进行扩充，将搜集到的手写加减乘除和括号的图片与mnist数据集进行合并，运用h5py将合并后的数据集压缩保存下来；然后使用两层卷积层，两层池化层，两层全连接层的卷积神经网络（卷积层使用5\*5的卷积核进行卷积，激活函数使用relu激活函数，池化层使用最大池化，全连接层每层有1024个神经元，激活函数使用relu激活函数，为了防止过拟合在第一层全连接层使用dropout函数，使用交叉熵来计算损失函数，使用AdamOptimizer函数来降低损失）对拓展后的数据集进行训练，使神经网络能够识别数字、加减乘除及括号，并且最后的识别准确率达到了99.42%。

# 项目测试

# 结果分析

# 总结

# 致谢

# 参考文献