



北京科技大学
University of Science and Technology Beijing

密级： 公开

本科生毕业设计(论文)

题 目： 仪表示数 OCR 模型与系统

作 者： XXX

学 号： XXXXXX

学 院： 计算机与通信工程学院

专 业： 计算机科学与技术

成 绩：

2019 年 5 月

本科生毕业设计(论文)

题目: 仪表示数 OCR 模型与系统

英文题目: A Method And A System Designed
For Automatic Meter Reading

学 院： 计算机与通信工程学院

班 级: XXXXX

学 生: XXX

学 号: XXXXXXX

指导教师: XXX 职称: XXX

指导教师: 职称:

声 明

本人郑重声明：所呈交的论文是本人在指导教师的指导下进行的研究工作及取得研究成果。论文在引用他人已经发表或撰写的研究成果时，已经作了明确的标识；除此之外，论文中不包括其他人已经发表或撰写的研究成果，均为独立完成。其它同志对本文所做的任何贡献均已在论文中做了明确的说明并表达了谢意。

学生签名：_____年__月__日

导师签名：_____年__月__日

毕 业 设 计 (论 文) 任 务 书

一、学生姓名: XXX

学号: XXXXXXXX

二、题目: 仪表示数 OCR 模型与系统

三、题目来源: 真实 ☒ 、 自拟 ☐

四、结业方式: 设计 ☒ 、 论文 ☐

五、主要内容:

针对人工读取仪表示数存在的成本费用高、工作效率低以及易出错等问题,利用图像处理技术,采用机器学习与深度学习的方法,建立一套仪表示数自动识别的系统。目前的多数系统采用网络摄像头对仪表进行拍摄,然后对传回的图像进行处理,分割得到单个数字进行识别,最后将单个数字识别的结果叠加得到示数。由于仪表所处环境不一,摄像头传回的图像可能存在遮挡、曝光等问题,在单字识别准确率很高的情况下出现精确度相比单字准确率较低的问题。本系统试图利用历史数据,减小仪表示数自动识别的误差,提高自动识别的精确度。

六、主要(技术)要求:

- 1、具备图像降噪,图像分割等预处理功能;
- 2、仪表示数识别正确率到 90%以上;
- 3、在完成模型的基础上能够提供 web 服务接口。

七、日程安排:

- 1--2 周: 收集课题相关资料,查阅相关文献,并完成选题报告;
- 3--5 周: 熟悉现有系统,利用 Tensorflow 等框架实现深度学习算法;
- 6--8 周: 建立模型,完成优化算法设计;
- 9--12 周: 进行系统设计,完成 web 应用的开发;
- 13--15 周: 总结研究成果,审阅和修改毕业论文,为毕业答辩做准备。

八、主要参考文献和书目:

- [1] 王锋,项导.基于卷积神经网络的数码仪表识别方法[J].机械设计与制造工程,2018,47(09):63-66.
- [2] 李明,丁琦,江洋,彭筱筱,傅时胜.基于图像处理的电能表数字识别方法[J].电子质量,2018(03):41-45.

- [3] Weicun Fan. Research on the recognition method of electric energy meter lead title based on the fuzzy image processing[A]. Information Engineering Research Institute, USA、 Technical Co-sponsored by Loughborough University, UK. Proceedings of 2014 2nd International Conference on Computer, Electrical, and Systems Sciences, and Engineering (CESSE 2014 V2)[C].Information Engineering Research Institute ,USA、 Technical Co-sponsored by Loughborough University, UK:智能信息技术应用学会,2014:9.
- [4] 赵诗宇.图像分割算法的分类与研究[J].科技风,2019(05):58.
- [5] D. Han and H. Kim, "A number recognition system with memory optimized convolutional neural network for smart metering devices," 2018 International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, HI, 2018, pp. 1-4.
- [6] 张苏沛,刘军,肖澳文,杜壮.基于卷积神经网络的验证码识别[J].武汉工程大学学报,2019,41(01):89-92.
- [7] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, et al. Beyond short snippets: Deep networks for video classification[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4694-4702.

指导教师签字:	年	月	日
学 生 签 字:	年	月	日
系(所)负责人章:	年	月	日

摘 要

针对人工读取各种仪表,如水表、电表、燃气表等,效率低而且成本高等问题,本文提出了一种自动识别仪表示数的方法,以减小此过程中的人工成本,提高识别效率。本文设计了识别模型并在此基础上实现了一个系统以对该方法进行了实际验证。

该识别方法首先借助 OpenCV 将采集到的表盘图像进行灰度化处理,然后利用基于直线探测的方法对表盘图片进行倾斜矫正。之后使用 Faster RCNN 网络模型替代传统的图像分割算法,使用目标检测的方法分割得到表盘示数区域图像。得到示数区域图像之后,将该图像在横向上平均切分为 6 份,得到 6 个单个数字图像,再将这 6 个单字图像交由 ResNet50 网络模型进行分类识别,然后按顺序组合分类结果,得到最终的示数识别结果。对于两个模型的验证,示数区域检测模型使用了平均 IoU 作为评价标准,而单数字识别模型则使用了数字识别的正确率作为评价指标。从两个模型的验证结果来看,两个模型都取得了令人满意的效果。

在系统实现上,系统包含了识别系统和 Web 系统两个部分。识别系统包含图像预处理、示数区域检测与单数字识别三个模块,其中示数区域检测被配置为 RPC 服务以提供更好的性能以及便利性。Web 系统在实时展示各表盘的识别结果的同时,还提供了识别错误预警以及人工修正的功能。

本系统的训练以及验证都在真实拍摄的表盘数据上进行,使用的数字集数量大且类型丰富,保证结果的真实可靠。系统的所有步骤均经过实验验证,且在单个表盘数据以及整体数据上都进行验证,从而更好地分析结果。实验结果表明,该方法适用范围广,识别准确率高,可以有效地解决仪表自动读数的问题,并运用在实际应用中。

关键词: 仪表识别, 神经网络, 图像处理

A Method And A System Designed For Automatic Meter Reading

Abstract

For the manual reading of various meters, such as water meters, electricity meters, gas meters, etc., having low efficiency and high cost, this paper proposes a method of automatic reading the number of meters to reduce the labor costs in this process and improve the recognition efficiency. In this paper, the recognition model is designed and a system is implemented to verify the method.

The method first performs grayscale processing, and tilt correction on the dial image by using a line detection method. Then, the Faster RCNN, instead of the traditional image segmentation algorithm, is used to segment and obtain the numerical region. Then the image is equally divided into 6 parts in the horizontal direction to obtain 6 single digital images, and they are subjected to classification by the ResNet50. Combine the classification results and the final recognition result is obtained. The numerical region detection model uses the average IoU as the evaluation standard, while the single digital recognition model uses the correct rate of digital recognition as the evaluation index. The verification results of the two models shows that both models have achieved satisfactory results.

The system contains two parts: the identification system and the Web system. The identification system includes three modules of image preprocessing, numerical region detection, and single digit recognition, where the numerical region detection is configured as an RPC service to provide better performance and convenience. The Web system displays the recognition results of each dial in real time, and also provides error warnings and manual corrections.

The training and verification of the system are carried out on the dial data from the real world. The number of dataset used is large and the types are rich, ensuring the authenticity of the results. All steps are validated on each single dial dataset as well as on the overall dataset to better analyze the results. The results show that the method has wide application range and high recognition accuracy, It can solve the problem of automatic meter reading and be used in applications.

Key Words: automatic meter reading, Neural Network, image processing

目 录

摘 要	I
Abstract	III
插图或附表清单	VII
1 引 言	1
1.1 研究背景	1
1.2 研究现状	1
1.2.1 仪表数字识别技术的发展与现状	2
1.2.2 Python 与 Web 应用框架 Django	5
1.3 研究意义	6
2 仪表识别模型	7
2.1 实验数据	7
2.1.1 数据集与数据集划分	7
2.1.2 原始数据	7
2.1.3 数据标注	8
2.2 图像预处理	8
2.2.1 灰度化与二值化	8
2.2.2 倾斜矫正	9
2.2.3 图像预处理的意义	9
2.3 示数区域检测模型	10
2.3.1 检测模型	10
2.3.2 训练集与测试集制作	11
2.3.3 模型训练结果	12
2.4 单字识别模型	15
2.4.1 网络模型	15
2.4.2 训练集与测试集制作	17
2.4.3 结果评估	18
3 仪表识别系统实现	21
3.1 开发环境	21
3.2 识别系统	22
3.2.1 示数区域检测服务	22
3.2.2 识别系统	24
3.2.3 系统评估	25

3.3 Web 系统	26
3.3.1 系统介绍	26
3.3.2 数据库设计	26
3.3.3 前后端设计与实现	27
4 仪表识别系统验证	31
5 结 论	36
参考文献	39
在学取得成果	41
致 谢	41

插图或附表清单

图 1-1 交点法（左）和交线法（右）示意图.....	2
图 1-2 识别流程示意图.....	3
图 1-3 粗网格特征提取示意图.....	3
图 1-4 BP 神经网络结构示意图.....	4
图 1-5 卷积神经网络结构示意图.....	4
图 1-6 Django 框架模块示意图.....	5
图 2-1 部分数据集倾斜矫正结果示例.....	9
图 2-2 示数区域标注示例.....	11
图 2-3 示数区域检测模型训练损失.....	12
图 2-4 示数区域检测模型训练过程中验证结果.....	13
图 2-5 IoU 计算示意图.....	13
图 2-6 测试集 M10L4XL 的检测结果显示.....	15
图 2-7 单数字识别网络结构.....	16
图 2-8 生成的单数字图片集.....	17
图 2-9 部分单数字图片示例.....	18
图 2-9 训练损失（a）和验证损失（b）.....	18
图 2-11 数字 2 和数字 3 的中间状态.....	20
图 2-12 数字 9 和数字 0 的中间状态.....	20
图 3-1 仪表识别系统结构示意图.....	21
图 3-2 系统 RPC 调用示数区域检测服务示意图.....	23
图 3-3 前端页面布局.....	28
图 3-4 单次识别结果展示示例.....	29
图 3-5 手动修正前结果展示.....	29
图 3-6 手动输入读数.....	30
图 3-7 手动修正之后结果展示.....	30
图 3-8 Web 系统整体界面.....	30
图 4-1 选择的验证表盘图片.....	31
图 4-2 识别结果展示在 Web 系统中.....	31
图 4-3 倾斜矫正结果.....	32
图 4-4 示数区域检测结果.....	32
图 4-5 数据库保存结果主要部分.....	32
图 4-6 修改过程.....	33
图 4-7 修改之后显示结果.....	33
图 4-8 修改之后数据库部分信息.....	33
图 4-9 测试集识别结果数据库内容.....	34
图 4-10 整个测试集识别结果 Web 展示.....	34
图 4-11 修改一个错误警告结果后的 Web 展示.....	34
表 2-1 示数区域检测 IoU 结果表.....	14
表 2-2 单数字识别正确率表.....	19
表 2-3 识别错误统计表.....	20
表 3-1 系统识别正确率与差 1 正确率.....	25
表 3-2 平均误差.....	26
表 3-3 数据表 dial 结构.....	26
表 3-4 数据表 recognition 结构.....	27

1 引 言

1.1 研究背景

自 20 世纪 40 年代人类发明计算机以来,计算机始终朝着具有更高智能的方向发展着。致力于计算机研究的科学家们努力让计算机能够更好的模拟人类思维,使计算机能够具有人类的逻辑思维判断能力。时至今日,我们仍然在探索一个问题,即计算机是否能够“思考”^[1]。虽然对于这个问题,我们还没有办法回答,但是计算机的智能化已经走了相当一段路程。从 20 世纪 90 年代开始,机器学习的发展进入了一个高峰时期,各大机器学习算法争先恐后地出现理论和应用场景中,如计算机视觉、自然语言处理、数据挖掘等等。我所选择的课题便是属于计算机视觉这一范畴。

2016 年,AlphaGo 的出现使得深度学习广为人知,成为人工智能的代表之作。深度学习是属于机器学习的一个子集,是一种基于统计的机器学习方法。以 CNN、DBN、SAE 和 RNN 为核心的深度学习模型体系,在图像图形处理、语音语意识别、字符文本处理等多个专业领域得到了十分广泛的应用,并取得惊人的效果。比如使用 BP 神经网络可以有效识别手写数字^[2],使用经典卷积模型 LeNet-5 搭建的手写体阿拉伯数字识别系统在训练 1200 次之后就可以达到 97% 的识别效果^[3]。另外,在深度学习算法发展的同时,一些深度学习框架也相继出现,如 TensorFlow^[4]、Theano、Caffe、Torch 等等。这些深度学习框架的出现极大地降低了学习和使用深度学习算法的门槛,使得一般计算机工作者利用深度学习算法解决实际问题成为可能。

在以上描述的大背景下,计算机工作者们摩拳擦掌,迫不及待地将机器学习技术施展于实际问题,以提高生产效率及造福人类社会。仪表识别问题便处于这些问题的行列之中。传统方法是通过大量人工读取仪表示数并记录,而这种方式人工成本相当昂贵,而且主要存在效率低、易出错等问题。于是作为智能式电表的替代方案,出现了运用图像处理、深度学习等技术,利用网络摄像头拍摄仪表图片,上传到服务器进行识别示数的智能读取方案^[5,6]。在此方案中,仪表示数的识别是最为关键的处理过程,本课题就是研究在此过程中的一种可行的解决方法。

1.2 研究现状

本节第 1 小节阐述了有关仪表数字识别技术的发展与现状,包括传统识

别方法与基于深度学习进行识别的方法，其中识别的流程又分为先进行图像预处理再识别和端到端识别两种。第 2 小节介绍了一种利用历史数据进行校正的简单方法。第 3 小节简要介绍了基于 Python 以及 Django 的 Web 应用开发技术。

1.2.1 仪表数字识别技术的发展与现状

数字识别技术属于图像识别技术的一个方面，后者是计算机智能应用的重要组成部分之一。本节介绍的几种方法，其识别正确率基本都能达到 90% 以上，随着神经网络和深度学习的发展，数字识别正确率已经能达到 97% 以上，为本课题的研究提供了很大的参考价值。

文献中提到的穿线法^[7,8]是识别规整数字（如七段数字管数字）的一种简单有效的方法,如图 1-1 所示。该方法先通过人工标注目标区域，然后进行二值化（Thresholding）以及形态学腐蚀膨胀（Eroding and Dilating）操作，之后对图像进行字符分割，得到单个数字组成的序列，交由穿线法识别。穿线法利用了七段数字较为规整的特点，使用线条与各段数码管相交，通过统计交点（交点法）或交线（交线法或割线法）作为单个数字的特征。

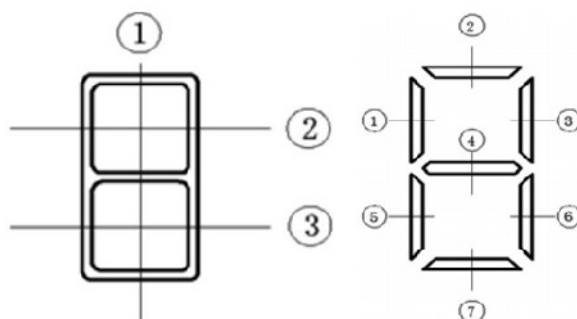


图 1-1 交点法（左）和交线法（右）示意图

总的来说这种方法比较简单，而且对数字的规整程度要求高，使用的范围比较小。本文引用此方法是因为一方面它能够说明处理仪表识别的一般流程，即图像预处理，字符分割^[9]，数字识别，另一方面它提供了一种特征提取的思路。



图 1-2 识别流程示意图

该方法在计算器数字识别问题上的准确率可以达到 100%。

除此之外，还有一些基于图像处理技术的识别方法^[10]，在一些较为简单的应用场景中能获得不错的效果。

在更加复杂的数字图像识别情况下，比如手写数字，上述方法并不能工作，而常常采用建立神经网络的方法进行识别。文献^[11]中提到一种基于向后传播神经网络的模型用以自动地识别机械式电表数字，在机械式电表数字的识别任务中，可以达到单字识别率为 99.9%，整表识别率为 98.5。

该方案处理流程与上述的流程相同，其预处理包括了三个部分，分别为图像数字区域搜索和数字图像二值化操作及图像倾斜矫正^[12]操作，再经数字分割后得到待识别的单数字图像。

得到单数字图像后，作者将字符进行了粗网格特征提取，即将字符图像点阵分为 5×7 共 35 个网格，分别计算单个小网格中黑色像素点的个数，得到一个 35 列的特征向量，并将它与 BP 神经网络的输入神经元连接。



图 1-3 粗网格特征提取示意图

作者采用了输入层 35 个神经元，隐藏层 16 个神经元，输出层 10 个神经元的三层 BP 神经网络结构。

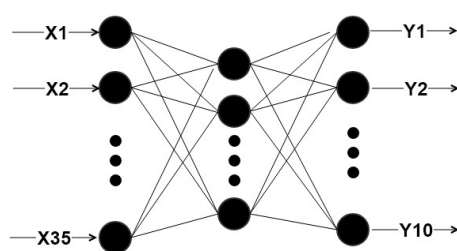


图 1-4 BP 神经网络结构示意图

BP 神经网络模型中的参数较多，计算量大，上一节的文献中虽然使用了粗网格特征提取方法减小了大量参数，但这样的抽象导致了原图像信息的丢失，利用只考虑了像素点的多少而没有考虑像素点之间的位置关系，最终模型的性能很大程度上取决于该特征提取方法的优劣。

卷积神经网络在图像识别中应用十分广泛^[13-15]，例如使用深度卷积自编码神经网络进行手写数字识别，在 MNIST 测试集^[16]正确率能达到 99.37%。

文献中^[17]提到利用卷积神经网络直接处理分割之后的单数字图像，不再人工进行特征提取，而是利用神经网络自动提取图像特征。

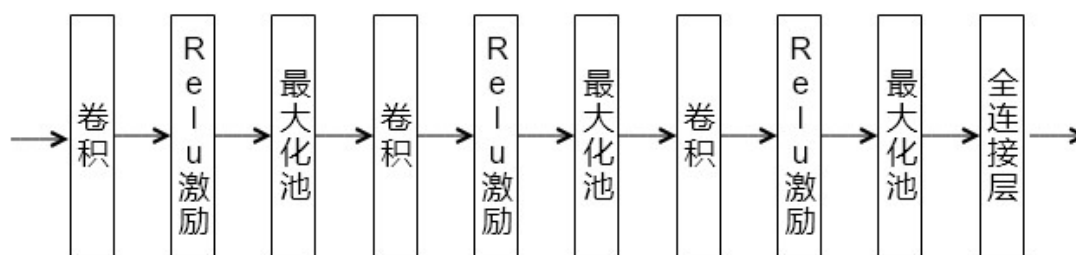


图 1-5 卷积神经网络结构示意图

以上提到的方法都是在字符分割得到单个数字的基础上进行识别，而当图像噪声较大，通过预处理之后可能仍然难以分割，或者得到错误的分割结果。

在网络验证码识别中，验证码生成器会在图像中添加大量噪声以及将字符进行堆叠从而干扰自动化识别程序的识别准确度。文献中^[18]作者基于 LeNet 构建了一个卷积神经网络模型，分层对图像提取从高级到低级的图像特征，选取 ReLU 作为激活函数，实现对验证码的识别，在测试集上识别准确率超过了 95%。

1.2.2 Python 与 Web 应用框架 Django

Django 是一个“强大，灵活，优雅，经充分测试”的开源 web 应用开发框架，使用“强大，干净，成熟，文档丰富”的 Python 语言编写^[19]。

Django 框架使用了比较常见的 Model-Template-View 模式，也被称为 MTV 模式。使用该模式的目的在于控制系统内多个组件的耦合关系，保证各个组件的设计不影响其他组件的运行^[20]。

M(Model)负责业务对象和数据库的关系映射(ORM)，T 表示模板(Template)，负责如何把页面渲染并且最终展示给用户，V 代表视图(View)，负责业务逻辑，协调调用 Model 和 Template。

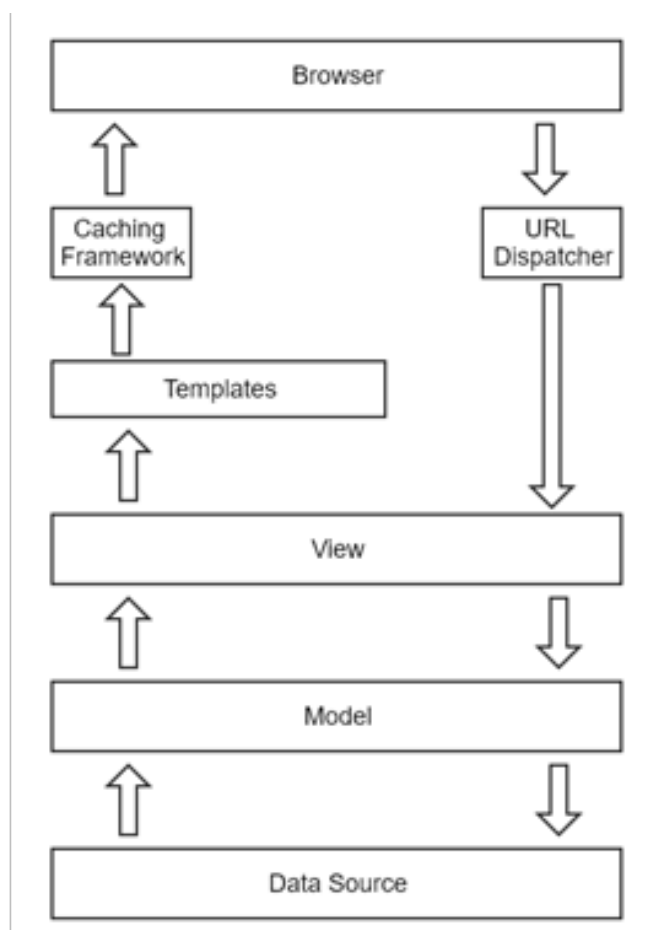


图 1-6 Django 框架模块示意图

随着网站开发技术的发展，前后端分离的设计方法已经成为主流技术路线，而设计符合 REST 规范的接口是实现前后端分离的有效手段^[21]。

1.3 研究意义

在上述的方案中，大多采用单个字符单独识别的方式，即先将仪表的图像利用传统图像处理技术定位表盘示数区域，然后进行分割，得到单个数字图像，然后利用机器学习的方法进行单数字分类识别。传统的图像分割方法具有泛化能力较弱的缺点，因此面对复杂的环境，如仪表盘规格不一，拍摄角度不一等情况，可能需要单独对图像分割算法进行调整。本文使用目标检测的方法，利用神经网络自动检测示数区域，提高示数区域检测的精确度。另一方面，本文尝试利用历史数据，减小数字 0 与数字 9 的识别错误，提高识别系统的准确率与精确度。

在完成识别模型的基础上，本文还将设计和实现一个 Web 应用，对识别结果进行展示和管理，同时在系统中提供对识别结果的错误预警以及人工修正。

2 仪表识别模型

在本文实现的系统中，需要两个模型，一个用于检测示数区域，另一个进行单字识别。每原始图片经预处理之后先交与检测模型检测示数区域，然后根据返回结果切割得到 6 张单字图片，这 6 张单字图片经单字识别模型后得到 6 个识别结果，将其按从左到右的顺序进行组合就可能得到最终的识别结果。下面先介绍本文所使用的数据集以及预处理过程，然后将详细介绍这两个神经网络模型的训练与评估。

2.1 实验数据

2.1.1 数据集与数据集划分

本文使用的数据均来自真实世界的拍摄图片，包含了 37 个仪表在一段时间内拍摄的共 11703 张图片，这 37 个仪表又包含了 5 种不同类型的仪表。拍摄的每张图片均为彩色 RGB 图片，宽 320 像素，高 240 像素。由于拍摄角度问题，数据集中的图片存在不同程度的倾斜。

为了对模型进行训练和验证，本文将数据集分为训练集和测试集，其中训练集包含 29 个仪表的共 9290 张图片，测试集包含 8 个仪表的共 2413 张图片。训练集和测试集均分别覆盖了 3 种不同的仪表类型。

本文使用的数据集数量较大，种类较多，且来自真实拍摄的图片，这些都支持了最终结果的可靠性。

2.1.2 原始数据

本文使用的原始图片为大小为 320×240 的真实拍摄图片，虽然仪表的类型存在不同，但都有一些相同的特征。例如，每种表盘的示数区域均为以黑色为背景的长条矩形，其中共含有 6 位数字，每位数字占表盘示数区域的宽度相同，且除最后一位外，前景色均为白色。

各类表盘的不同之处在于：a. 其示数区域宽高不同，边框也不同；b. 表盘示数区域之外的背景在内容及样式上各有不同；c. 有些表盘最后一位数字背景为红色，表示小数位，而有的表盘最后一位数字与示数区域背景为相同的黑色，表示整数最低位。

2.1.3 数据标注

本文使用的图片数据的标注均保存在与该图片文件具有相同文件名称的 XML 文件中，收纳在图片文件目录下的 ‘/baocun’ 目录中。

图片标注主要包含该图片的两个属性，一个属性为名称，另一个属性为示数区域坐标。名称属性与前文提到的图片文件名称不同，它表示的是该图片中表盘 6 位数字的真实读数。示数区域坐标属性指出了表盘示数区域在图片中的具体位置，由 $xmin$, $xmax$, $ymin$, $ymax$ 四个部分组成，分别表示矩形左上角 x 坐标，右下角 x 坐标，左上角 y 坐标，右下角 y 坐标。

2.2 图像预处理

图像预处理是整个识别流程的第一个步骤，主要包含图像的灰度化、二值化以及倾斜矫正。这些图片处理技术除了在预处理阶段使用，在整个实验的其他部分，如数据集制作，模型验证等部分也频繁使用。

2.2.1 灰度化与二值化

由网络摄像头采集的图片为彩色 RGB 图像，由 OpenCV^[22] 读取为形状为 (320, 240, 3) 的 Numpy 数组。

对 RGB 图片进行灰度化有几种常用的方法，公式 (2-1) 取每个像素的 B 分量为灰度像素的值，公式 (2-2) 取 B, G, R 三个分量的平均值作为灰度像素的值，公式 (2-3) 将 B, G, R 三个分量按一定的比例求加权平均作为灰度像素的值。本文采用第三种方法对图片进行二值化。

$$Gray = B \quad \text{公式 (2-1)}$$

$$Gray = \lceil (B + G + R) / 3 \rceil \quad \text{公式 (2-2)}$$

$$Gray = \lceil 0.114 \times B + 0.587G + 0.299R \rceil \quad \text{公式 (2-3)}$$

图片二值化是指将每个像素点用两个值来表示，通常这两个值选用 0 和 255。一般选定一个阈值 t ，然后用 0 表示原灰度图中该点像素值低于阈值，255 表示原灰度图中该点像素值高于阈值，如公式 (2-4)。

$$Bin = \begin{cases} 0, & Gray < t \\ 255, & Gray \geq t \end{cases} \quad \text{公式 (2-4)}$$

2.2.2 倾斜矫正

在采集到的仪表数据中，存在由于摄像设备固定时出现倾斜导致仪表图像出现倾斜的情况，需要先将图片进行倾斜矫正，得到表盘示数区域横向水平的图片，才能输入到系统中进行后续的工作。

在图像处理中，常见的倾斜矫正方法有基于轮廓提取技术以有基于直线探测技术等方法。在本系统的倾斜矫正任务中，虽然示数框的轮廓相当清晰且规则，但是仪表盘上的内容复杂多样，基于轮廓提取技术的方法在实际应用中出现诸多干扰，不利于矫正任务。而这些内容虽然复杂，但都是以直线方式组合的，如示数框为直线，上方标识文字以及下方的型号说明等等文字都按直线对齐排列，因此采用直线探测的方法进行倾斜矫正比较合理且容易取得比较好的矫正效果。

基于直线探测技术的倾斜矫正方法在文本矫正任务中比较常见，它的主要步骤为：

- (1) 选择合适的参数，用霍夫线变换探测出图像中的所有直线。
- (2) 为了排除干扰，计算出每条直线的倾斜角，找出其中出现次数最多的倾斜角作为最终的倾斜角。
- (3) 根据倾斜角旋转矫正。

在实际应用中，本文使用了 OpenCV 提供的霍夫直线检测方法，然后根据上面的步骤进行处理，最终在所有测试图片上都获得了很好的矫正效果，如图 2-1 所示。



图 2-1 部分数据集倾斜矫正结果示例

2.2.3 图像预处理的意义

在将图像输入到识别系统之前，本文对图像进行了灰度化和倾斜矫正。对图像进行灰度化处理，主要是为了减少输入数据量的大小，因为经灰度处理后图像大小变为原图像的 1/3，这对后续神经网络的训练与验证提供了方便。另外，进行灰度化处理还以去除颜色对识别系统的干扰，比如不同表盘

最后一位可能为红色也可能为黑色，但灰度化后差异就减小了很多。图像矫正主要使示数区域矩形框在图片中保持横向水平，一方面提高示数区域识别模型平均 IOU，另一方面由于单字分割采用垂直切割，所以对于单字分割的效果也有所改善。

图像二值化操作并没有直接作用于输入图片，而是作为倾斜矫正的中间步骤发挥作用。

2.3 示数区域检测模型

本节介绍了使用 Faster RCNN 神经网络模型对示数区域进行目标检测，并对训练结果进行了评估。

2.3.1 检测模型

多数文献中通过仪表表盘具有的某些特征来检测示数区域，但是当表盘存在较大差异时，使用同一种特征的检测算法就很难正确检测出所有的示数区域。本文使用目标检测模型 Faster RCNN^[23,24]，将示数区域作为待检测的目标，让网络模型自动学习 4 种不同表盘示数区域的特征。

目标检测是指从一张图像中找到感兴趣的区域，并对此区域打上标签。Faster RCNN 是 2016 年出现的一种目标检测框架，它在原有的目标检测框架的基础上做了改进，使用区域推荐网络提高了区域推荐的速度和质量。

Faster RCNN 框架有 2 个主要步骤，第一个是候选框生成，得到一定数量的可能含有目标的方框，另一个是目标分类，对这些候选框进行具体类别的分类，打上标签。

在实际应用中，本文在 Faster RCNN 框架的一个开源实现的基础上进行修改，以适应本文中示数区域识别的具体任务。在图像特征提取网络上，本文选择了 VGG16，同时选择了 PASCAL_VOC 作为训练集的数据集格式。本次目标检测任务中，需要检测的目标只有一个类型，即示数区域，另外加上背景类，总共两个类别，即“__background__”和“dial”两类，背景类固定在所有类别的最前面。在配置文件中设置与本次工作环境相对应用的显卡，同时由于本次使用的显卡显存较小，只有 8G 显存，默认设置在训练中会出现显存不足的问题，所以设置框架输入图片的最大边不超过 350，这样训练过程所占用的显存将大大减少，而本文中使用的训练图片大小统一为 320×240，这样的设置足够对训练集进行训练。

2.3.2 训练集与测试集制作

训练该模型时需要制作 PASCAL_VOC 格式的数据集，此种格式的数据集在目标检测领域十分常见。此格式主要内容包括三个文件夹，分别为 Annotations, ImageSet 和 JPEGImages。其中 Annotations 文件夹中存放图像标注，使用 xml 文件格式，每张图片都有一个标注文件，且与图片文件名相同；JPEGImages 文件夹中存放待训练的图片；ImageSet 文件夹中主要考虑 Main 文件里，里面存有 4 个文件，test.txt, train.txt, trainval.txt, val.txt，分别指定了测试集，训练集，验证与训练集以及训练集。这 4 个文件中每行代表一个训练或验证或测试数据，用图片的文件名表示。

在生成 PASCAL_VOC 数据集之前，先要对每张图片示数区域的位置进行标注，如图 2-2。由于在本文使用的数据集中，每一个表盘的拍摄位置是固定的，所以每个表盘只需要标注一次，总共需要为 37 张图片就行标注即可。数据标注程序遍历这 37 个表盘数据各自的第一张图，首先对其进行倾斜矫正，然后将图片显示出来，利用 OpenCV 的鼠标事件以及绘图功能，让用户框选出示数区域，然后将示数区域的左上角以及右下角坐标保存下来。

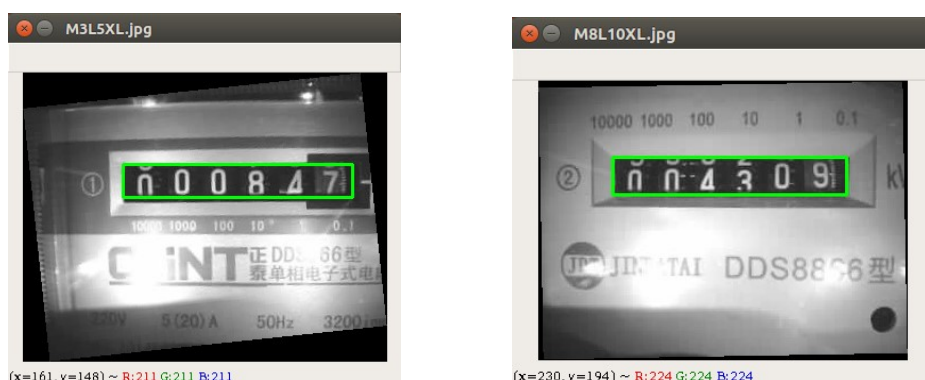


图 2-2 示数区域标注示例

得到示数区域标注后，即可按 PASCAL_VOC 格式制作训练集与测试集。为了让神经网络正确学习到示数区域的特征，而排除位置的干扰，需要将图片进行平移。图片进行平移的范围与图片的大小以及示数区域的位置有关：

$$\begin{cases} -xmin < \Delta x < d - xmax \\ -ymin < \Delta y < h - ymax \end{cases} \quad \text{公式 (2-5)}$$

$$\begin{cases} xmin' = xmin + \Delta x \\ ymin' = ymin + \Delta y \\ xmax' = xmax + \Delta x \\ ymax' = ymax + \Delta y \end{cases} \quad \text{公式 (2-6)}$$

其中 $xmin$, $xmax$, $ymin$, $ymax$ 为示数区域的坐标点, w 为图片的宽, h 为图片的高, $\Delta x, \Delta y$ 分别表示横向偏移与纵向偏移距离, $xmin'$, $xmax'$, $ymin'$, $ymax'$ 表示平移之后示数区域的坐标点。

实际处理过程中, Δx 与 Δy 的取值范围较大, 如果全部遍历生成的训练过大, 在训练过程中容易产生训练过度, 训练时间过长, 因此随机选取了部分 Δx 与 Δy 。另外, 由于每个表盘的不同图片相差并不是很大, 也不需要全部加入到训练集中, 也通过随机选取的方式选取了部分图片加入到训练集中。随机选取图片的比例设置为 0.015, Δx 与 Δy 的选取比例设置为, 最终将训练集大小控制在 0.003。

在训练集中还需要进一步划分出训练集, 验证集和测试集, 三者的比例分别为设置为 0.7, 0.2 和 0.1。

2.3.3 模型训练结果

将生成的训练集按要求命名并放在要求的目录下, 即可运行模型训练。模型训练进行了 70000 次迭代, 在使用 NVIDIA GTX1080Ti 显卡的服务器上训练了 12 小时左右, 最终训练损失为 0.1212, 验证损失为 0.1215, 如图 2-3 所示。

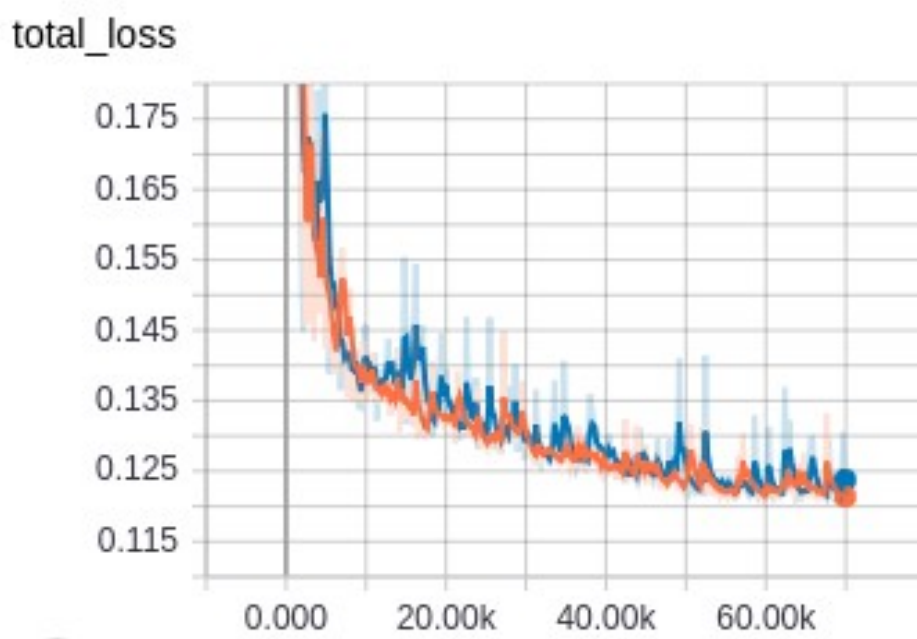


图 2-3 示数区域检测模型训练损失

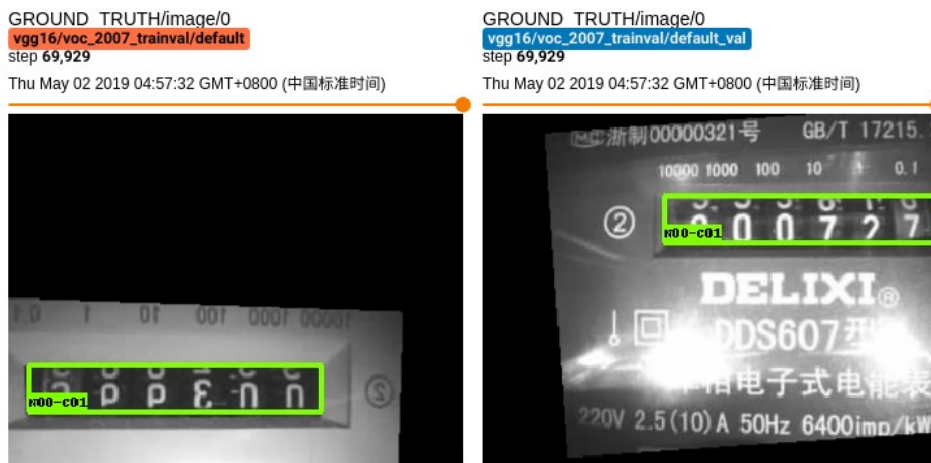


图 2-4 示数区域检测模型训练过程中验证结果

如图 2-4，从模型在训练过程中对验证集的示数区域检测结果来看，模型达到了不错的检测效果。

下面介绍使用 IoU 计算模型性能 mAP 对模型的效果进行严格的评价。



图 2-5 IoU 计算示意图

如图 2-5 所示，假设上面的方框表示该图片本身标注的示数区域，下面的方框表示使用模型进行检测得到的示数区域，那么有公式 (2-7)：

$$IoU = \frac{A_i}{A_u} \quad \text{公式 (2-7)}$$

其中 A_i 表示图中深色部分的面积，即两个方框的交集的面积， A_u 表示图中深色部分与浅色部分面积的总合，即两个方框的并集的面积。

$$\begin{aligned} w &= \max(xmin_L, xmin_D) - \min(xmax_L, xmax_D) \\ h &= \min(ymax_L, ymax_D) - \max(ymin_L, ymin_D) \quad \text{公式 (2-8)} \\ A_i &= w \times h \end{aligned}$$

A_i 计算如公式公式 (2-8)，其中带下标 L 表示标注框，带下标 D 表示检测框。

$$\begin{aligned} A_L &= (xmax_L - xmin_L) \times (ymax_L - ymin_L) \\ A_D &= (xmax_D - xmin_D) \times (ymax_D - ymin_D) \quad \text{公式 (2-9)} \\ A_u &= A_L + A_D - A_i \end{aligned}$$

A_u 计算如公式公式 (2-9)，其中 A_L 表示标注框的面积， A_D 检测框的面积。

模型的输出形状为 (N, 5)，其中 N 表示模型检测到目标的个数，5 表示结果向量的长度为 5。结果向量格式为 {xmin, ymin, xmax, ymax, score}，其中 score 表示该检测结果的置信度。在测试时，模型选择了所有 score 大于阈值 0.8 的检测，而在实际使用中本文只选取了 score 最高的一项作为检测结果，并返回其方框坐标。

得到检测方框坐标后使用公式公式 (2-7) 计算该图片的 IoU，然后以表盘为单位计算每个表盘数据上的平均 IoU，再以整个测试集为单位，计算整个测试集上的平均 IoU。计算结果如下表：

表 2-1 示数区域检测 IoU 结果表

数据集	平均 IoU
M10L10XL	0.9003
M10L4XL	0.8101
M1L8XL	0.9022
M2L10XL	0.9596
M2L7XL	0.8693
M3L1XL	0.8691
M3L10XL	0.9492
M8L10XL	0.9250
整体	0.8982

由计算公式可知，平均 IoU 是一个小于 1 的数，而且数值越大，表示检测结果与标注结果越接近。一般而言，当这个数字大于 0.5 的时候，就可以认为此目标检测是一个不错的结果。由于本文中检测的目标只有一个分类，而且特征比较明显，本身又是较为规则的矩形区域，所以检测结果的平均 IoU 相对比较大，每个表盘测试集上都达到 0.8 以上，整体平均 IoU 达到 0.8982，说明目标检测的结果相当不错。

从表中可以发现，测试集 M10L4XL 的平均 IoU 最低，只有 0.8101，但是从图 2-6 中的检测结果来看，这样的检测结果已经足够用于后面的其他步骤了。因此，该示数区域检测模型效果达到了预期，可以用于本文的识别系统。

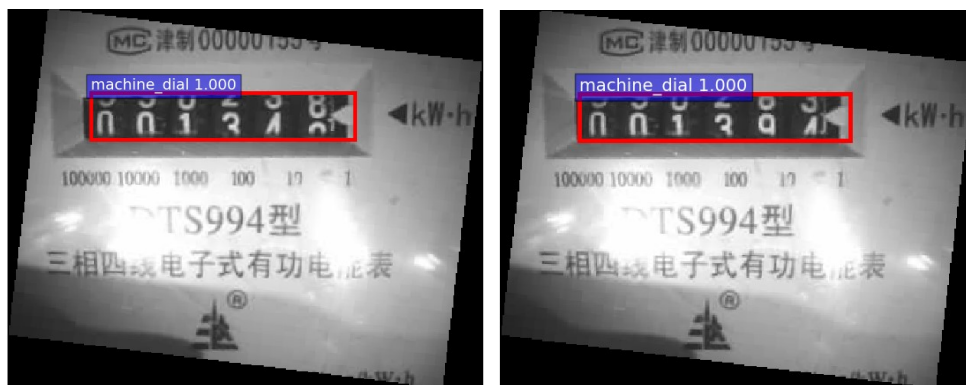


图 2-6 测试集 M10L4XL 的检测结果显示例

2.4 单字识别模型

原始图片经示数区域检测后得到示数区域坐标，切割即可得到示数区域图像。将此图像横向平均切分了 6 份，即可得到该表盘的 6 个数字图像。

对于得到的 6 个单数字图像，本文使用 ResNet50 神经网络进行分类，具体说明如下。

2.4.1 网络模型

ResNet^[25,26]在传统的卷积神经网络中加入了一种新的残差块结构，它的输出与输入相同，即作用为绕过卷积网络，直接将输入连接到输出。当训练达到一定程度时，其中一些残差块就可能变为恒等网络即输出等于输入，避免了网络深度对训练带来的负面影响。从某种角度来看，ResNet 相当于多种网络结构的并行，在训练过程，网络会通过将部分网络变为恒等映射的方式自动选择最优的结构。

在实际应用中，本文采用了 ResNet50，即深度为 50 层残差网络，包含 49 层卷积层以及 1 层全连接层。在残差网络之后，首先加上了一层 Flatten 层，将残差网络的输出在一维方向上展开，以适应上层的网络输入。

接着，在网络中加入了一层 Dropout 层，并设置丢弃比例为 0.2。Dropout 层的作用是随机丢弃一定比例的训练结果，即将一定比例的神经单元的输出设置为 0。通过这种机制，可以使神经网络不过分依赖于局部特征，增强网络的泛化能力，解决神经网络容易发生的过拟合问题。

另一种降低过拟合影响的方法是在网络中添加正则化层，它在损失函数中加入惩罚项，使得参数更加复杂的网络的损失更大，从而在训练中尽可能地使得网络的结构更加简单。这样一来，网络就会选择更加有用的特征，而

减小无用特征的影响，提高网络的泛化能力。在本文中使用了 $L_1 = 0.001$ ， $L_2 = 0.001$ 的 L1 与 L2 正则。

该网络输入为单个数字图像的灰度 Numpy 数组，形状为 (X, 64, 64, 1)，输出为长度为 10 的一维向量，形状为 (X, 10) 即分类结果的独热码表示，其中 X 为训练集的大小。例如，分类结果为 0 即对应向量 {1, 0, 0, 0, 0, 0, 0, 0, 0, 0}，分类结果为 1 即对应向量 {0, 1, 0, 0, 0, 0, 0, 0, 0, 0}，以此类推。与此输入输出相对应地，网络的最后一层为 10 个神经元的全连接层，使用 softmax 激活函数。

网络在训练时使用的损失函数为 categorical_crossentropy，该损失函数常与 softmax 激活函数配合用于解决分类问题。为了避免错误最优的模型，配置了较小的学习率， $lr = 0.0001$ 。较小的学习率会导致参数更新的幅度较小，从而学习速度较慢，因此配置了较大的训练轮数，epoch 设置为 200，保证能够训练得到最优的模型。

在进行训练之前，把训练图片数据的 Numpy 数组除以 255，从而使得每个像素的值都在 0 到 1 之间。这样做的目的在于减小网络参数的大小，有利于梯度下降，使网络收敛得更快。同时对数据集进行随机打乱处理，否则网络会按顺序先输入一定数量的数字 0 图片，然后是数字 1 图片，以此类推，这样会导致网络学习的效率降低，最终神经网络学习到的特征也很有限。最终的网络结构如图 2-7 所示。

Layer (type)	Output Shape	Param #
resnet50 (Model)	(None, 2, 2, 2048)	23581440
flatten_3 (Flatten)	(None, 8192)	0
dropout_2 (Dropout)	(None, 8192)	0
dense_2 (Dense)	(None, 16)	131088
dense_3 (Dense)	(None, 10)	170
Total params: 23,712,698		
Trainable params: 23,659,578		
Non-trainable params: 53,120		

图 2-7 单数字识别网络结构

2.4.2 训练集与测试集制作

训练数据来自第 2 章介绍数据集划分出的训练集，经示数区域检测模型检测得到读数区域，然后平均将此区域平均分割 6 份得到单字数字图像。这样得到的单字训练集大小是原始训练集大小的 6 倍，即 5 万左右。这样大小的训练集对于本次数字图像分类任务来说过大，容易使得在训练过程中难以学习，同时训练时间过长，所以实际在制作训练过程中，通过一定的比例选取最终加入至训练集的图片，最终将训练集大小控制在 1.5 万左右。

由示数区域分割而得的单字图像，因为分割方式为平均分割，得到的单字图像中，数字在整个图像中的位置相对固定，不具备比较好的泛化能力。因此，在制作训练集时，将分割得到的单字图像随机在一定范围内进行位置的偏移，这样训练出来的模型可以很好的应对由其它示数区域切割而来的位置不一的单字图像。具体说一说如何偏移的，包括一些边界处理。

单字图像虽然是平均切割而来，每个示数区域切分的 6 张图片保证大小是相同的，但是对于示数检测来说，检测到的示数区域本身的大小是不尽相同的。为了方便将训练数据输入到神经网络当中，本文将所有单字图像缩放到 64×64 大小。

训练集处理得当后，保存为 Numpy 文件，方便重复加载及训练。

测试集的制作过程与训练集类似，但是没有随机选取以及位置偏移这两个步骤。

最终生成 4 个文件：train_data.npy，train_label.npy，test_data.npy，test_label.npy，分别对应训练集数据，训练集标签，测试集数据以及测试集标签。生成的数据如图 2-8、图 2-9 所示。

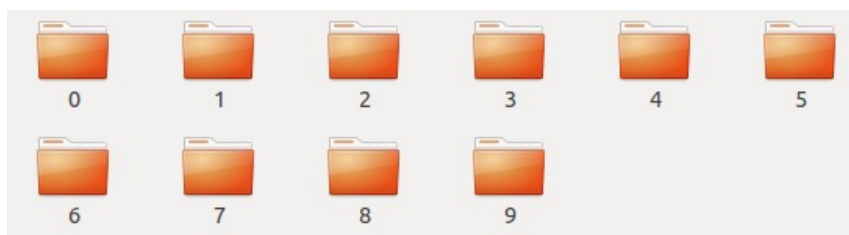


图 2-8 生成的单数字图片集

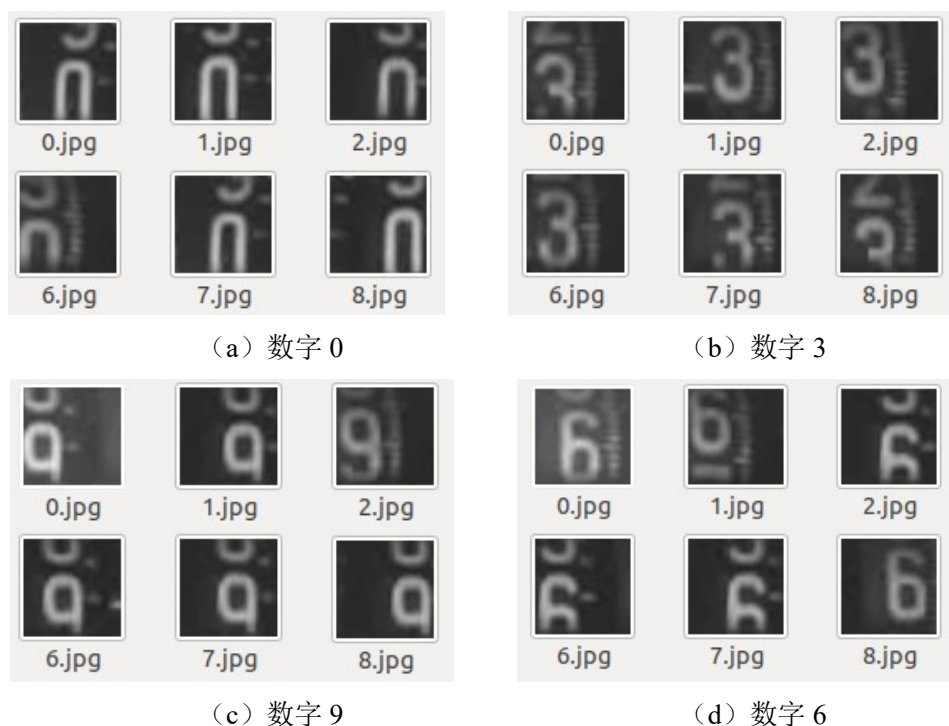


图 2-9 部分单数字图片示例

2.4.3 结果评估

网络在总共训练 200 个 epoch，在使用 NVIDIA GTX1080Ti 显卡的服务器上训练了 5 小时左右，最终训练损失为 $5.0288e-3$ ，验证损失为 0.04292。

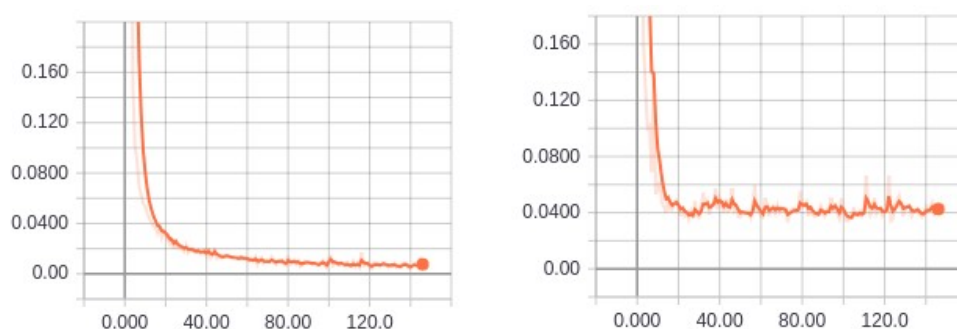


图 2-10 训练损失 (a) 和验证损失 (b)

将训练模型在测试集上进行验证，计算每个数字的识别正确率以及 10 个数字整体的识别正确率，结果如表 2-2 所示。可以看到在 14478 个单数字图片中，整体识别正确率达到 99.23%，而每个数字的识别正确率都在 97% 以上。由表中可以得知，虽然数字 7 和数字 9 的识别正确率相对较低，但那是因为测试集中这两种数字的数量本身占比就比较小，因此计算识别正确率的时候分母较小导致正确率相对较低。因而，从整体角度来看，这个单字识

别模型的识别结果达到了令人满意的效果，可以用于本文的识别系统。

表 2-2 单数字识别正确率表

数字	识别正确率	测试数量
0	0.9985	5872
1	0.9894	1800
2	0.9889	1264
3	0.9918	1097
4	0.9940	1336
5	0.9886	881
6	0.9818	769
7	0.9778	406
8	0.9865	668
9	0.9714	385
整体	0.9923	14478

分析了单数字识别的正确率，下面再分析一下识别错误的统计情况。如表 2-3 所示，表格的列标识表示模型识别的结果，表格的行标识表示该单数字图片的标注读数，例如表格的第 0 行第 9 列为 7，表示标注为 0 但是识别为 9 的情况出现了 7 次，而表格的第 2 行第 3 列为 12，表示标注为 2 但是识别为 3 的情况出现了 12 次，以此类推。仔细分析这张表格可以发现，两个斜对角线以及第 0 行第 9 列、第 9 行第 0 列中（表格中用粗体标注的部分）识别错误的情况比较多，而其它部分的错误情况基本为零。

这是由于，如第 2.1 节所说，在本文使用的数据集中，存在两个数字的中间状态图片，在这样的情况下，识别为上一个数字和下一个数字实际上都可以算作是识别正确的。如图 2-11，识别为数字 2 或者数字 3 都可以算作是识别正确。但是，同样的情况对于数字 0 和数字 9 数字来说，如图 2-12，就不能算作是识别正确，因此这会带来较大的误差。实际上数字 0 和数字 9 的错误可以通过基于历史数据的方法进行修正，在第 3.2.2 节会详细介绍。

表 2-3 识别错误统计表

识别 标注	0	1	2	3	4	5	6	7	8	9
0	0	2	0	0	0	0	0	0	0	7
1	13	0	6	0	0	0	0	0	0	0
2	0	2	0	12	0	0	0	0	0	0
3	1	0	4	0	4	0	0	0	0	0
4	0	0	0	4	0	4	0	0	0	0
5	0	0	0	0	9	0	1	0	0	0
6	0	0	0	0	0	9	0	5	0	0
7	0	0	0	0	0	0	3	0	6	0
8	0	0	0	0	0	0	0	3	0	6
9	8	0	0	0	0	0	0	0	3	0

图 2-11 数字 2 和数字 3 的中间状态

图 2-12 数字 9 和数字 0 的中间状态

3 仪表识别系统实现

为了验证上述模型与方法的可行性，本文还设计并实现了一个仪表识别系统，如图 3-1。该系统接收一张含有仪表表盘的图片，经过预处理后，利用上述两个模型进行检测和识别，得到仪表的示数，并将相关的信息保存到数据库中，最后利用 Web 系统将结果展示到网页上，同时该 Web 系统还提供识别结果错误警告以及人工修正等功能。

该系统的最前端应该是一个图片采集系统，由固定在真实表盘上的网络摄像头、光补偿以及单片机或微处理器构成，主要定时拍摄表盘图片，并经由 4G 或 WI-FI 网络上传到本系统，再由本系统进行后续的识别。上述采集系统虽实现简单但硬件制件较复杂，而且本文设计此系统主要是为了验证文中识别方法和模型的有效性，故并没有将此采集系统整合在本系统之中，而是使用已拍摄好的表盘图片进行验证。在实际应用中，可以通过专门的硬件设计制造商定制采集相关硬件，然后编写图像采集与上传程序，与本系统完成对接。

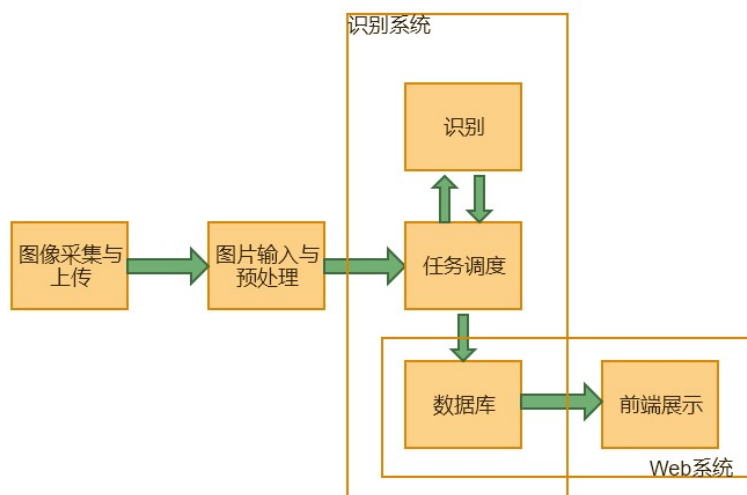


图 3-1 仪表识别系统结构示意图

3.1 开发环境

本文中仪表识别系统的实现主要使用 Python 作为编程语言，它语法简洁，逻辑清晰。它还具有十分丰富且功能强大的第三方类库，可以快速地构建出功能强大的系统。另外，在机器学习和深度学习方面，Python 也是一门非常合适的编程语言。本文中使用的 TensorFlow 以及 Keras 深度学习框架即是在 Python 语言上的深度学习框架。文中主要使用的 Python 版本有 Python2.7

与 Python3.6。

对于图片处理方面，本文使用了 OpenCV 提供的 Python 接口，主要用到了其图像输入输出方法、颜色空间转换方法、直线及轮廓检测以及图像显示方法等等方法。OpenCV 功能强大，具有丰富的图像处理函数库，对于本文中需要完成的任务来说绰绰有余，而对以后的改进和优化也完全能够胜任。

对于 Web 系统开发，本文基于 Django Web 框架进行开发。Django 是 Python 生态中的一个开源的 Web 开发框架，从 2005 年发展至今，具有设计优美，稳定可靠，开发便捷以及文档丰富等等优秀的特点。

本系统使用了开源的 MYSQL 数据库，它是目前使用最为广泛的开源的关系型数据库，具有体积小，速度快等特点。

本系统前端页面使用 Vue.js 进行开发，前后端逻辑分离，前端使用 AJAX 异步请求与后端 RESTful 接口进行通信，将得到的数据展示在网页上。

本系统在 Ubuntu16.04 上进行开发，也适用于其它与之兼容的系统。另外，由于 Python 本身具有跨平台的特性，本文中的系统只需做少许改动就可以运行在 Windows，MAC 等主流平台上。

3.2 识别系统

识别系统主要对输入的含有表盘数据的图片先进行一系列数据图像处理操作，然后交由示数区域检测模型和单数识别模型进行识别，最终得到识别结果，并将结果保存到数据库当中。

3.2.1 示数区域检测服务

示数区域检测服务主要用到了上文中介绍的示数区域检测模型。分为以下 3 个步骤：

(1) 将输入的图片进行倾斜矫正，得到示数区域方框在横向水平的图片，并进行灰度化处理，交由示数区域检测模型进行检测。

(2) 示数区域检测模型检测示数区域得到方框区域。

(3) 从模型输出中提取 score 最高方框坐标，去除 score 并返回。

示数区域检测服务基于 RPC (Remote Procedure Call 远程方法调用)^[27]，接收一个图片路径，返回该图片示数区域的检测坐标。

远程过程调用包含一个客户端和一个服务端，服务端通常通过一个或多个方法为客户端提供服务，而客户端接入服务时其操作更像是在调用服务端的一个方法，因此叫作远程过程调用，如图 3-2。在本系统中，示数区域检

测模型放在服务端，通过方法 `get_pos` 提供示数区域检测服务，客户端通过远程过程调用使用此方法得到示数区域方框的坐标，然后再进行后续的识别以及其它任务。

在实际搭建系统的过程中，本文采用了 Python3.6 中提供的 XML-RPC，即 XML 远程过程调用。XML-RPC 使用 HTTP 协议在客户端与服务端传递数据，使用 XML 格式为要传递的数据进行编码。

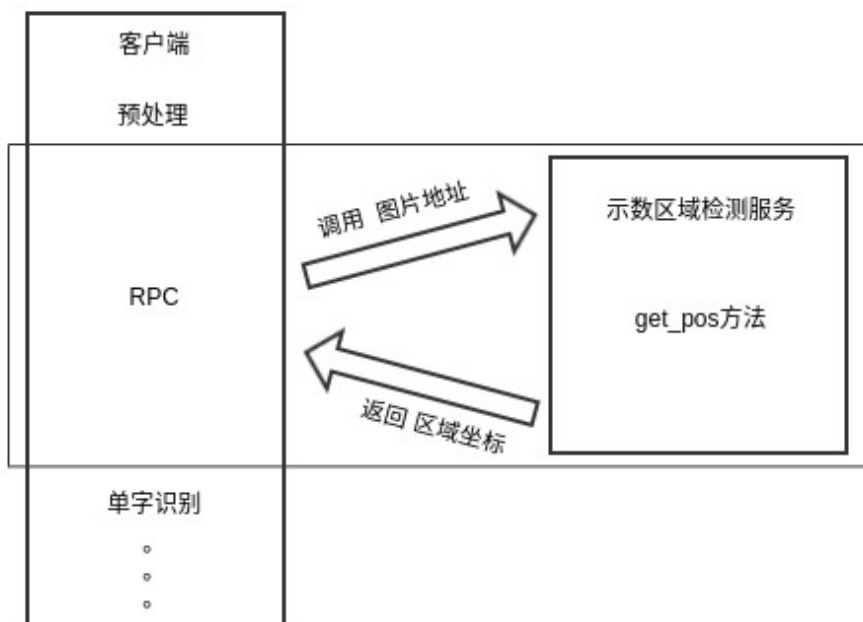


图 3-2 系统 RPC 调用示数区域检测服务示意图

在本系统中使用了两个神经网络模型分别进行示数区域检测和单数识别，这两个模型都运行在 GPU 上。虽然可以让两个模型同时运行在一个 GPU 上共享 GPU 的显存，但为了提高整个系统的速度，有必要让两个模型分别运行在不同的显卡上。对于只含有一个显卡的服务器来说，使用远程过程调用就可以让两个模型运行在不同的服务器上，从而分别利用各自的显卡。除此之外，由于两个模型代码使用的 Python 版本以及 TensorFlow 版本均不相同，虽然可以使用虚拟环境进行管理，但是独立运行在不同的服务器上更加方便。综上，使用 RPC 的两个模型的运行互不干扰，更加方便的同时也提高了系统的性能。

在客户端调用 RPC 服务时只提供了一个图片地址，因此服务端需要保存好表盘图片数据，并且按照一定的目录结构存放。这样一来，图片数据就需要在两台服务器上各存储一遍。之所以选择这样的方式，一方面是由于 Python 的 XML-RPC 原生只支持传递基本数据类型，如数字类型、字符类型等，如

果需要传图片的话需要自己打包成参数类从而传递图片的 Numpy 数组,不太方便。另一方面,通过网络传送图片数据比从硬盘中读取图片要慢得多,因此在服务端与客户端各保存一份图片数据相当于用硬盘空间的占用换取系统运行速度的提升。

3.2.2 识别系统

识别系统包含了两个模型以及数据库操作,系统的输入含有待识别表盘读数的图片,输出为存入数据库当中的基本信息、识别结果以及部分中间结果。

该系统首先将图片进行倾斜矫正,得到示数区域横向水平的图片 I_h , 然后对其进行灰度化 I_g 。

接着该系统将图片地址作为参数,远程调用示数区域检测服务的接口方法,得到该图片示数区域的坐标,并根据此坐标从图片 I_g 中将示数区域裁剪出来,得到只包含示数区域的图片 I_d 。

将 I_d 横向平均分为 6 份,得到 6 张图片,每张图片包含一个待识别的表盘数字。将这 6 张图片分别缩放到 64×64 大小,然后按在表盘中从左到右的顺序组合成形状为 (6, 64, 64, 1) 的 Numpy 数组,其中最后一个维度值为 1 表示通道数只有 1 个,即灰度图像。将此 Numpy 输入单字识别模型进行单字识别,得到形状为 (6, 10) 的输出,即 6 个以独热编码的识别结果,分别取每个独热编码最大值所在的下标,即可得到每位数字的识别读数,再依次组合起来,便可以得到最终的识别结果。

接着识别结果将连同其它的一些信息,如表盘名称、图片地址及示数区域坐标等,一起存入数据库当中,供 Web 系统访问。

在对系统识别结果的简要分析当中发现,在最后一位数字的识别当中,数字 0 和数字 9 容易发生识别错误的情况,这对于整个系统精确度的影响比较大。考虑到系统对每个表盘都进行了连续的采样与识别,系统在当前一次的识别中可以利用上一次的识别结果进行一个验证。比如在此问题上,若系统的识别结果 R_t 中最后一位为数字 0,系统上一次对同一表盘的识别结果为 R_l ,而 R_t 小于 R_l ,由于同一表盘的识别是连续递增的,这说明系统错误地将最后一位的数字 9 识别成了数字 0,即可将其校正过来。

再进一步考虑,由于系统采样间隔时间较短,远短于结算的间隔时间,因此,系统识别的结果不会立即用于结算,这给了系统对先前识别结果一次

校正的机会。比如在最后一位数字 0 与数字 9 的识别问题中, 如果系统发现当前识别结果 R_i 最后一位不为数字 0 或者数字 9, 上一次识别结果 R_l 中最后一位数字识别为 9, 而 R_i 小于 R_l , 由于同一表盘的识别是连续递增的, 这说明系统错误地将上次识别中最后一位数字 0 识别成了数字 9, 也可以将上次识别结果校正过来。

通过上面两种考虑, 利用历史数据以及同一表盘读数具有连续递增的特点这一规律, 就可以将系统最后一位数字识别对 0 与 9 的混淆这一问题解决, 提高系统的整体精确度。

3.2.3 系统评估

如表 3-1 所示, 系统在单个表盘测试集上的正确率均达到 91%以上, 最低的识别正确率为 91.36%, 其对应的表盘为 M3L1XL。在 2413 张测试图片中, 整体识别正确率达到 94.70%。

除了统计识别正确率, 由于在单数字识别系统中, 中间状态的数字识别为上一个数字或下一个数字都可以算作是识别正确, 因此在最终的识别结果中, 如果识别结果和标注数据相差为 1, 也可以算作是识别正确, 因此还统计了差 1 正确率, 即将结果差 1 视为识别正确的正确率。可以看到, 差 1 正确率在每个表盘测试集上都达到了 99%以上。

比较表盘 M3L1XL 测试集, 可以发现其正确率最低但差 1 正确率相对较高, 说明在这个测试集当中, 最后一位处理中间状态的图片数据较多, 因此出现了这样的结果。

表 3-1 系统识别正确率与差 1 正确率

数据集	正确率	差 1 正确率	数据集大小
M10L10XL	0.9669	0.9934	302
M10L4XL	0.9269	0.9900	301
M1L8XL	0.9468	0.9934	301
M2L10XL	0.9475	0.9967	305
M2L7XL	0.9502	0.9934	301
M3L1XL	0.9136	0.9934	301
M3L10XL	0.9668	0.9900	301
M8L10XL	0.9568	0.9934	301
整体	0.9470	0.9930	2413

除了正确率, 本文还在整个测试集上计算了平均误差, 如表 3-2 所示。

进行数字 0 和数字 9 的修改之前，系统在测试集上的平均误差约为 0.1550，而进行修正之后的平均误差减小为 0.1326。实际上，平均误差的主要来源是差 1 错误，所以数字 0 和数字 9 修正对于平均误差的影响没有太大，但是从结果上来看，此修正的确提高了系统识别的精确度。

表 3-2 平均误差

	0、9 修正前	0、9 修正后
平均误差	0.1550	0.1326

3.3 Web 系统

3.3.1 系统介绍

除识别系统之外，本系统还包括一个 Web 系统，用于识别结果的展示与校正。该系统从数据库中读取识别系统的识别结果，向用户展示在前端页面上，并提供错误预警和校正功能。

此 Web 系统使用 Django 作为后端框架，Vue.js 作为前端开发框架，采用前后端分离的开发模式，前端通过异步请求向后端 RESTful 接口请求数据并实时显示。

3.3.2 数据库设计

本系统选择了著名的开源关系型数据库 MYSQL，使用 Django REST framework 对数据库进行各项所需要的操作。

本系统主要需要两张数据表，一张数据表用于存储所有的表盘，每个表盘占一行数据，另一张数据表用于存储所有的识别结果，每个识别结果占一行数据。具体的表字段如表 3-3，表 3-4 所示：

表 3-3 数据表 dial 结构

编号	字段名	字段类型	字段说明
1	id	INTEGER	该表盘的唯一标识，自动增加
2	name	CHAR(255)	该表盘的名称，用于前端展示时区分
3	warnings	INTEGER	该表盘识别结果警告个数

表 3-4 数据表 recognition 结构

编号	字段名	字段类型	字段说明
1	id	INTEGER	本次识别的唯一标识, 自动增加
2	dial	INTEGER	被识别表盘的 ID
3	image_path	CHAR(255)	识别图片所有路径
4	result	CHAR(255)	识别结果
5	angle	CHAR(255)	旋转角度
6	crop_pos	CHAR(255)	示数区域坐标
7	warning	INTEGER	结果是否有错误警告
8	create_date	DATETIME	识别时间, 自动生成

表 3-3 中, id 为主键, 是表盘的唯一标识, 自增整型; name 为表盘的名称, 在测试中为测试数据集所在的文件夹名; warnings 为整数, 表示该表盘警告个数。

表 3-4 中, id 为主键, 是识别的唯一标识, 自增整型; dial 为本次识别图像的表盘 id; image_path 为识别图片的路径, 在系统中另有一图片服务器, 此处使用的是此图片在图片服务器中的 http 地址; result 为本次识别的识别结果, 以字符串形式存储; angle 为倾斜矫正时对图片的旋转角度, 用字符串表示, 以度为单位; crop_pos 为示数区域模型检测的示数区域的坐标, 以字符串表示的左上角坐标以及右下角坐标; warning 为 bool 值, 表示识别结果是否有警告; create_date 为本次识别的识别时间, 在插入数据时由数据库自动生成。

在前端页面展示时通过 recognition 表的 dial 字段可以找到本次识别的表盘名称。通过 image_path 字段可以得到识别图片的路径, 将图片显示在页面上。通过 angle 和 crop_pos 两个字段可以得到识别过程中的旋转、区域检测结果以及分割结果, 并显示在前端页面上。

3.3.3 前后端设计与实现

前端页面主要使用了 Vue.js 与 Element UI 以及 JQuery 进行实现, 其中 Vue.js 主要利用其响应式的特点, 负责数据的更新显示, JQuery 框架主要利用其 Ajax 方法向后端异步请求数据, 实现前后端分离, 而 Element UI 主要利用其布局以及其它页面元素组件, 达到页面美化的目的。

根据本系统所设计的功能，将前端页面划分为 3 个区域：头部区域简要显示系统信息，左部侧边栏显示所有的表盘，中间显示详细的识别结果以及对识别结果的校正操作。布局实现使用了 Element UI 的 container 组件，包含 header，aside 以及 main 三个子组件。布局效果如图 3-3 所示：



图 3-3 前端页面布局

页面左边使用列表展示当前系统所识别过的表盘名称，点击表盘名称，将在右边主界面按识别时间分页展示此表盘的所有识别结果。当点击左侧某一项表盘名称时，页面会以该表盘的 ID 为参数向后端发起 Ajax 请求，同时还会带上页码以及每页大小等信息，后端收到这些信息之后，会根据这些信息从数据库中读取部分相应的识别结果。这些识别结果经 Django 的 REST framework 框架序列化成 JSON 格式之后，返回给前端页面。前端页面收到返回数据，将其解析为 JavaScript 对象，并将其赋值给 Vue 的一个变量。Vue 是一个响应式的框架，这意味着，当它绑定的一个变量的值发生变化的时候，Vue 会自动地对页面重新进行渲染。本页面使用了 Vue 的 for 循环，遍历识别结果组成的数组变量，将识别结果按顺序显示在主界面。每一个识别结果的显示使用一个带有 header 的 card 组件实现，分为三个部分。最左边的部分是本次识别的原始图片，可以在此区域看到拍摄过程是否出现问题。中间的部分是经示数区域检测以及平均分割后的结果，在此区域可以查看示数区域检测以及分割是否出现问题。最右边的部分是本次识别的结果，为 6 位数字，可以查看识别结果是否正确。三个部分如图 3-4 所示：



图 3-4 单次识别结果展示示例

除了查看识别结果，系统还提供手动修正的功能，用以对识别可能有误的结果进行人工调整。由于系统对于每个表盘是进行连续识别的，所有识别结果都保存在数据库中，而用户对于每次采集间隔时间内仪表示数的增量都会有一个最大的预期，如果两次识别结果的差值超出了这个预期，说明很有可能是系统识别出现了错误，此时可以进行人工干预，手动修改本次识别结果。具体修改方法是点击右上角的手动修正按钮，在弹出的输入框中输入正确的示数，然后点击确定，正确的读数以及本次识别的 ID 就会作为参数，通过 Ajax 请求传给后端，后端将会修改数据库中的 `correct` 字段，同时前端页面将更新显示结果。修改过程如图 3-5，图 3-6，图 3-7 所示。

前端整体界面如图 3-8 所示。



图 3-5 手动修正前结果展示

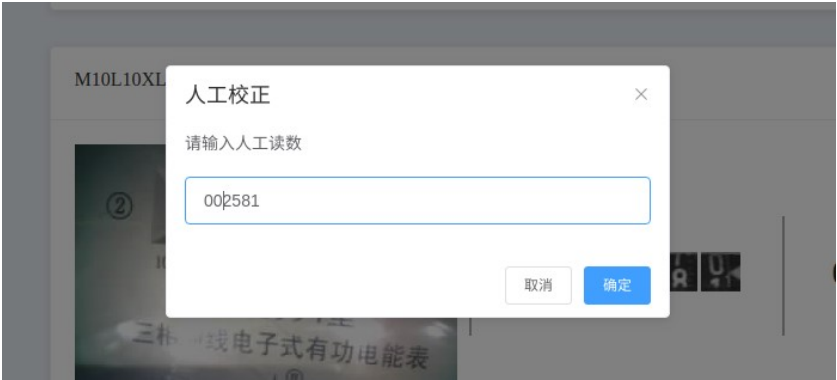


图 3-6 手动输入读数



图 3-7 手动修正之后结果展示



图 3-8 Web 系统整体界面

4 仪表识别系统验证

前文已经对识别系统在测试集上进行了验证，本章主要通过对一张图片进行识别的流程，验证整个系统的可用性。

首先，随机从测试集中选择一张图片进行本次验证。选择的图片如图 4-1 所示：



图 4-1 选择的验证表盘图片

将图片输入到识别系统当中，得到的结果如图 4-2 所示，可以看到系统正确识别了此图中的表盘示数并且显示在了 Web 系统中。可以看到该图片来自表盘数据集 M2L10XL，其原始图片显示在结果最左边，中间展示了其经过倾斜矫正、区域识别以及单字分割后的结果，最右边该图片的正确示数 001124。



图 4-2 识别结果展示在 Web 系统中

再看一下该图片识别过程的中间结果。如图 4-3 为倾斜矫正的结果，可以看到图片被顺时针旋转了一定角度，使用表盘示数区域水平。如图 4-4 为示数区域检测结果，可以看到示数区域检测模型检测正确检测到了示数区域，并且检测结果十分准确。如图 4-5 为数据库中对此次识别结果保存的部分信息截图，可以看到，系统保存了该图片的地址，识别结果，旋转角度以及示数区域坐标等信息。



图 4-3 倾斜矫正结果



图 4-4 示数区域检测结果

image_path	result	angle	crop_pos
http://127.0.0.1:8081/M2L10XL/2018-11-23-13-31-01.jpg	001124	-1.9778...	{'xmin': 72, 'ymin': 67, 'conf': 1, 'ymax': 104, 'xmax': 268}

图 4-5 数据库保存结果主要部分

为了验证手工修正功能，假设本次识别结果不正确，而正确的结果为 001125。点击识别结果右上角的手动修正按钮，在弹出的输入框中输入 001125 并点击确定。修改过程和修改结果如下面的几张图所示。如图 4-6 所示，在

输入框中手动输入正确的示数。如图 4-7 所示，修改完成之后 Web 页面立即更新识别结果为人工修正后的示数。如图 4-8 所示，修改完成之后，数据库中的 correct 字段被修改成正确的示数，而 result 仍然是原来识别的结果。



图 4-6 修改过程

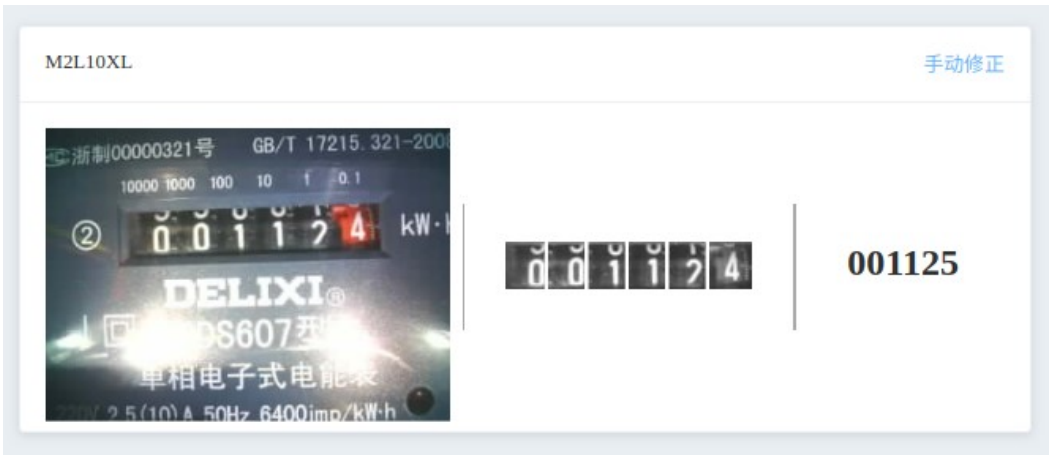


图 4-7 修改之后显示结果

image_path	result	correct
http://127.0.0.1:8081/M2L10XL/2018-11-23-13-31-01.jpg	001124	001125

图 4-8 修改之后数据库部分信息

下面对整个测试集进行验证，将所以图片输入到系统中，一段时间后查看识别结果。图 4-9 是数据库中存入的数据，可以看到存入了 2413 条结果。

#	id	dial	image_path	result	angle	crop_pos
1	727	1	http://127.0.0.1:8081/M10L10XL/2018-11-24-07-26-01...	002579	4.96601...	{'xmin': 84, 'ymin': 60, 'conf': 1, 'ymax': 92, 'xmax': 273}
2	728	1	http://127.0.0.1:8081/M10L10XL/2018-11-23-18-36-01...	002525	4.96601...	{'xmin': 85, 'ymin': 59, 'conf': 1, 'ymax': 92, 'xmax': 280}
3	729	1	http://127.0.0.1:8081/M10L10XL/2018-11-24-05-21-01...	002570	4.96601...	{'xmin': 84, 'ymin': 59, 'conf': 1, 'ymax': 92, 'xmax': 279}
4	730	1	http://127.0.0.1:8081/M10L10XL/2018-11-24-09-01-01...	002586	4.96601...	{'xmin': 86, 'ymin': 59, 'conf': 1, 'ymax': 92, 'xmax': 275}
5	731	1	http://127.0.0.1:8081/M10L10XL/2018-11-24-03-41-01...	002563	4.96601...	{'xmin': 85, 'ymin': 60, 'conf': 1, 'ymax': 92, 'xmax': 278}

monitor_recongnition 7

Action Output

#	Time	Action	Message	Duration / Fetch
6	20:47:53	SELECT * FROM monitor.monitor_recongnition LIMIT 0, 5...	0 row(s) returned	0.0051 sec / 0.0000...
7	21:04:27	SELECT * FROM monitor.monitor_recongnition LIMIT 0, 5...	2413 row(s) returned	0.013 sec / 1.904 sec

图 4-9 测试集识别结果数据库内容



图 4-10 整个测试集识别结果 Web 展示



图 4-11 修改一个错误警告结果后的 Web 展示

从图 4-10 可以看到，将错误警告的阈值设置为 10，即当前示数与上一次读数的差大于 10 即认为有可能是系统识别错误，用红色背景以及表盘名称

左上角数字提示错误警告以及具体警告的数量。可以看到在主界面中，选择的表盘 M10L10XL 有一个错误警告，且警告的识别结果已经置顶且用红色背景标识。此图片正确的读数应该是 002539，而系统错误的识别为了 002549。点击右上角的手动修正按钮，即可进行人工修正。修正之后的结果展示如图 4-11 所示，可以看到左侧警告标记已经消失，主界面也不再置顶之前警告的识别结果。

5 结 论

本文针对人工读取仪表示数效率低，统计不便等缺点，设计并实现了一个仪表自动识别系统，在测试集上正确率达到 94.70%，差一正确率达到 99.30%，平均误差为 0.1326。

本系统包括识别系统和 Web 系统两个部分，其中识别系统又包含示数区域检测模型和单字识别模型两个模型。示数区域检测模型使用 Faster RCNN 神经网络模型代替传统的区域检测方法，在测试集上平均 IoU 达到 0.8982。单字识别模型使用 ResNet 神经网络模型，在测试集上单字识别正确率达到 99.23%。两个模型与两个系统分工合作，相互支持，形成一个从识别到校正的完整系统。

在两个模型的交互上，系统使用了 RPC 模式交换数据和结果，以提高系统的性能，减小系统的耦合。两个系统之间通过数据库传递识别结果，互不干扰，交接融洽。

系统在使用单字识别模型的基础上，为了提高系统识别的精确度，尽可能减少数字 0 与数字 9 的识别错误，利用历史数据，一方面根据上次识别结果修正当前识别结果，另一方面利用当前识别结果校正上次识别结果，减小了系统在测试集上的平均误差。

虽然系统的识别正确率比较高，但是针对可能出错的情况，系统还提供了错误预警以及人工修正的功能，使用系统更加具有实用性。

本系统各个部分都单独做了验证，最后对于整个系统也做了验证。验证既在测试集的单个表盘数据上进行，也在整个测试集上进行并统计分析。各项验证说明本系统在仪表自动识别任务上是可应用的，并且在识别正确率与精确度上都达到了预期的效果。

本系统还存在一些不足和值得改进的地方，比如：

(1) 系统训练和测试使用的数据集虽然包含的表盘种类和数量都比较多，但是整体来看类型还是比较单一，比如都是 6 位数字，示数区域背景为黑色方框等等。系统目前能处理的仪表也只能是含有上述特征的仪表，而对于上述之外的仪表完全不具有处理能力。将更多的表盘数据加入到训练集中，并针对差异较大的仪表使用不同特征进行区分，各自处理，可以提高本系统的处理能力。

(2) 系统在切割单数字时利用数字在示数区域中平均分布的特点，采用简单的平均分割方法，偶尔可能会出现由于示数区域检测不准确而导致切割

数字左右有少部分残缺的情况。采用基于轮廓检测的方法可以得到每个数字更精确的位置信息，使用数字切割更加准确，对于识别系统的准确率与精确度也可能有提高。

（3）系统在对图像的预处理时，仅进行了灰度和倾斜矫正处理，对于噪声较大的图像不具备降噪的功能。在系统中加入图像降噪功能，可以使系统对于光照、清晰度不同的图像有较好的兼容性。

在以后对系统的完善中，将首先针对上述不足以及相应的改进办法进行实践，进一步提高系统的准确率和精确率，同时扩大系统的处理能力范围，增加系统的抗干扰能力。

参考文献

- [1] 刘瑞祯, 于仕琪. OpenCV 教程——基础篇[M]. 北京航空航天大学出版社, 2007.6.
- [2] 嵇晶晶, 陈雅倩, 王季宇, 骆金秋, 常恒. 车牌识别中倾斜牌照的快速矫正[J]. 科技经济导刊, 2018, 26(35): 68.
- [3] 谢奇芳, 姚国清, 张猛. 基于 Faster R-CNN 的高分辨率图像目标检测技术[J/OL]. 国土资源遥感, 2019(02): 38-43[2019-05-25].
- [4] 张汇, 杜煜, 宁淑荣, 张永华, 杨硕, 杜晨. 基于 Faster RCNN 的行人检测方法[J]. 传感器与微系统, 2019, 38(02): 147-149+153.
- [5] 尼加提·卡斯木, 师庆东, 刘素红, 比拉力·依明, 李浩. 基于 VGGNet 和 ResNet 模型的沙漠腹地绿洲植物群落自动分类方法[J/OL]. 农业机械学报: 1-19[2019-05-29].
- [6] 王恒, 李霞, 刘晓芳, 徐文龙. 基于 ResNet50 网络的乳腺癌病理图像分类研究[J]. 中国计量大学学报, 2019, 30(01): 72-77.
- [7] 赵晓辉. XMLRPC 应用实例[J]. 中文信息, 2003(11): 58-59+49.
- [8] François Chollet. Deep Learning with Python[M]. 张亮., 北京: 人民邮电出版社, 2018.
- [9] 张黎, 刘争鸣, 唐军. 基于 BP 神经网络的手写数字识别方法的实现[J]. 自动化与仪器仪表, 2015(06): 169-170.
- [10] 吕红. 基于卷积神经网络的手写数字识别系统的设计[J]. 智能计算机与应用, 2019, 9(02): 54-56+62.
- [11] 邢艳芳, 段红秀, 何光威. TensorFlow 在图像识别系统中的应用[J/OL]. 计算机技术与发展, 2019(05): 1-5[2019-03-20].
- [12] 王锋, 项导. 基于卷积神经网络的数码仪表识别方法[J]. 机械设计与制造工程, 2018, 47(09): 63-66.
- [13] Gallo I, Zamberletti A, Noce L. Robust angle invariant GAS meter reading[C]//2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA). IEEE, 2015: 1-7.
- [14] 卓磊, 周律, 杨丽红. 基于穿线法的计算器数字识别[J]. 软件工程, 2018, 21(12): 1-3.
- [15] 王舒憬, 杜皎洁. 基于 OPENCV 的数字万用表数字识别方法[J]. 自动化与仪器仪表, 2014(01): 62-63.

- [16] 赵诗宇.图像分割算法的分类与研究[J].科技风,2019(05):58.
- [17] 李明,丁琦,江洋,彭筱筱,傅时胜.基于图像处理的电能表数字识别方法[J].电子质量,2018(03):41-45.
- [18] 胡立夫,齐胜男,张海军.基于神经网络的电表数字识别技术研究[J].沈阳航空航天大学学报,2011,28(02):43-45.
- [19] FAN W, LI Y. Research on the recognition method of electric energy meter lead title based on the fuzzy image processing, Singapore, F, 2014 [C].
- [20] D. Han and H. Kim, "A number recognition system with memory optimized convolutional neural network for smart metering devices," 2018 International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, HI, 2018, pp. 1-4.
- [21] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, et al. Beyond short snippets: Deep networks for video classification[C]. // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4694-4702.
- [22] 曾文献,孟庆林,郭兆坤.基于深度卷积自编码神经网络的手写数字识别研究[J/OL].计算机应用研究:1-6[2019-03-20].
- [23] 陈晓冰,彭浩滇,欧德炜.基于深度学习的机械式电表行码识别方法[J].数字通信世界,2018(10):227-228.
- [24] 张苏沛,刘军,肖澳文,杜壮.基于卷积神经网络的验证码识别[J].武汉工程大学学报,2019,41(01):89-92.
- [25] Forcier, Jeff, Paul Bissex, and Wesley J. Chun. Python web development with Django[M]. Boston:Addison-Wesley Professional, 2008.
- [26] 周君,贾昆霖,蓝机满,宋艳.基于 Django 框架的智能图书推荐系统[J].电子科技,2018,31(12):78-80+85.
- [27] 张云飞,杨明光.基于 Django 的 RESTful 通用程序接口研究与实践[J].电脑知识与技术,2018,14(28):30-32.

在学取得成果

一、 在学期间所获的奖励

无

二、 在学期间发表的论文

无

三、 在学期间取得的科技成果

无

致 谢

转眼到了大学四年的最后一个阶段，在毕设工作以及毕业论文即将完成之际，我诚挚地感谢在我大学四年尤其是毕设阶段中对我给予了学业上的帮助和指导的人。

首先我要感谢我的指导老师 XXX 老师。XXX 老师是一位极富责任心的老师，在我进行毕设的整个过程中，包括选题、下达任务书、毕设实验以及最终的论文撰写，他都给予了悉心的指导和及时的帮助。XXX 老师每周组织的组会让我对当前进度有一个整体的把握的同时，对目前存在的问题以及下一步的工作都了然于胸，这篇论文能够完成都得益于此。

其次，我要感谢在毕设过程中所有帮助过我的学长学姐，尤其是 XXX 学长。XXXX 学长在我毕设期间给我提供了不少参考资料，在组会上提出的建议也对我的思路有很大的启发。

另外，我要感谢 XXXX 实验室为我提供的实验数据、服务器以及实验过程中使用到的其它设备，正是这些软硬件设施支持了我毕设的顺利进行。

最后，感谢每一位参与毕设答辩和评审的老师，感谢你们的辛勤工作！再次向以上提到的人和组织表示最诚挚的感谢！