

Bank Card Recognition Method Based on Mobile Terminal

Anping Cai, Boyu Qiu, Xiaohui Tan, Shanglin Li*

School of Software and Communication Engineering, South Hunan University, Chenzhou Hunan
Email: *wmlsldd@163.com

Received: Apr. 2nd, 2020; accepted: Apr. 17th, 2020; published: Apr. 24th, 2020

Abstract

In view of the problem that the user has a high error in entering the bank card number on the mobile terminal, this paper proposes a bank card recognition method based on the mobile terminal. The method is mainly divided into three steps: bank card image acquisition and preprocessing, bank card number digital character segmentation, digital character recognition. The image preprocessing part mainly uses various edge detection algorithms and morphological operations to process, including the projection correction of the bank card and the acquisition of the card number area. After obtaining the digital area, the card number is extracted and the number is segmented. Finally, the convolutional neural network (CNN) is used to train the model to realize the final number recognition. Finally, through the development of the app based on Android client for many times, it is verified that this method can identify the bank card number better.

Keywords

Bank Card Recognition, Character Recognition, Character Segmentation, Convolutional Neural Network

基于移动端的银行卡识别方法

蔡安平, 邱博宇, 谭小辉, 李尚林*

湘南学院软件与通信工程学院, 湖南 郴州
Email: *wmlsldd@163.com

收稿日期: 2020年4月2日; 录用日期: 2020年4月17日; 发布日期: 2020年4月24日

摘要

针对用户在移动端进行银行卡卡号录入出错性高的问题, 本文提出了一种基于移动端的银行卡识别方法。

*通讯作者。

方法主要分为三步：银行卡图像获取及预处理，银行卡号数字字符分割，数字字符识别。其中图像预处理部分主要利用各类边缘检测算法、形态学操作进行处理，具体包括对银行卡进行投影矫正、获取卡号区域。获得数字区域后，再进行卡号提取、数字分割，最后使用卷积神经网络(CNN)训练模型，实现最终的数字识别。最后通过开发的基于Android客户端的App进行多次试验，验证了该方法能够较好地识别银行卡卡号。

关键词

银行卡识别，字符识别，分割字符，卷积神经网络

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近些年来，我国经济飞速发展，科技实力日益突飞猛进。手机移动端的使用给人们的生活带来了许多便利。日常生活中，移动端支付的方式越来越流行。手机端进行支付往往需要绑定银行卡。目前，在众多需要录入银行卡信息才能办理的业务仍然停留在纯手工录入的方式，流程繁琐又耗时，造成业务办理的等待时间长，流程效率低，顾客抱怨增多，运营成本也在不断增大。

近些年来，机器学习和人工智能的飞速发展，在计算机视觉领域取得了突破性进展，尤其是卷积神经网络技术的发展，使得部分图像识别算法达到甚至超过人工的能力[1]。在实际的项目开发中，OpenCV很好地吸纳了最新的技术，有效提高了开发效率和程序运行的可靠性，为识别银行卡提供了高效的开发工具[2]。

针对银行卡识别问题，一种经典的解决思路是光学字符识别(optical character recognition, OCR)技术。该技术在规范的文本文档扫描方面识别率已经达到了99%以上。然而，在自然影像场景中对银行卡识别的应用，仍然存在较大挑战[3]。OCR传统方法在应对复杂自然场景的文字识别方面显得力不从心[4]，目前银行卡识别由于自然场景复杂度高，存在一定的难度，部分银行卡难以定位辨析，识别率低。所以本文致力于结合OpenCV和深度学习技术，研究一种通用性和准确率高的银行卡识别方法，并移植到移动端。本文提出的银行卡识别算法具备简洁、快速、准确的优点，方便在移动端上进行实现，给用户实现快速的银行卡绑定提供便捷[5]。

2. 方法介绍

基于移动端，利用Android Studio、Python以及Tensorflow + Keras对银行卡识别方法进行研究。本软件的开发需要在CLE框架集成Python解释器，我们只需要把所需要Python模块编译成文件装载到apk中即可，并且对于移动端的深度学习框架Tensorflow Lite所使用的模型为TFLite模型，我们在PC端训练模型之后还需要转换成TFLite模型。根据移动端的简便性和Python的可移植性，本文致力于开发移动端的手持银行卡识别软件，具体流程如图1所示。本文将主要围绕以下方法进行详细研究：

2.1. 边缘检测

OpenCV提供了多种边缘检测算法，主要有Sobel和Canny算子，Canny算子实际上是对Sobel的优化，并且经过大量实验之后，发现Canny算子能够最大化去除噪声并且保留边缘信息，所以在这里选

用 Canny [6]算子来进行边缘检测。由于彩色图片存在占用空间大、计算量大得特点，需要把彩色图像进行灰度化[7]，灰度化通常采用四种方式:最大值法、平均值法、分量法、加权平均值法。本文采用的是加权平均值法，通常采用公式： $Gray = 0.299R + 0.587G + 0.114B$ 。

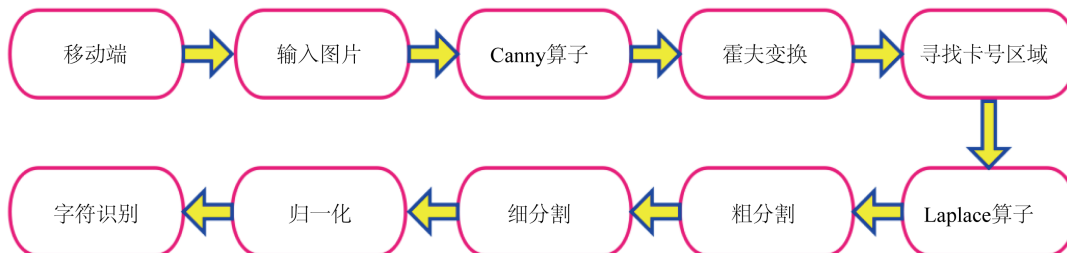


Figure 1. Programing flowchart
图 1. 流程图

对于灰度化的银行卡需要进行高斯模糊[8]，分别从 X 和 Y 两维的方向进行卷积。卷积的意义在于获取原始图像中像模版特征的性质。计算梯度值和方向；非极大值抑制，是边缘检测中的一个重要步骤，通俗意义上是在寻找像素点的局部最大值。沿着梯度方向，比较前后梯度值。Canny 算法需要用一个高阈值和一个低阈值和判断强弱边缘点。Canny 算法对凸字银行卡处理效果如图 2、图 3 所示。



Figure 2. Original image
图 2. 原始图像



Figure 3. Result of canny edge detector
图 3. Canny 算法处理效果

2.2. 图像矫正

目前对于图像的倾斜矫正的算法很多，比如基于 Radon 变换的倾斜矫正算法以及基于霍夫变换的倾斜矫正算法，还有许多改进的倾斜矫正算法。本文是利用基于霍夫变换的倾斜矫正算法[9]。霍夫变换是用来找出物件中的特征，根据霍夫变换寻找银行的边缘直线，从而获取水平、垂直方向银行卡的边缘直线。

通过找到银行卡的水平垂直方向的焦点对目标的银行卡进行投影变换，这里输入原始图像如图 4 所示，得到矫正的银行卡，如图 5 所示。

2.3. 寻找数字区域

在之前的工作中已经得到了二值化的图像，我们所需要的做的就是从繁乱的背景中提取出目标的数字区域。OpenCV 中有许多封装好的算法，本文利用 cvFindContours ()得到粗略的卡号区域。根据像素点在中间区域比较集中，故对于中间区域在水平和垂直方向根据像素点的统一值设置一个阈值，经过大量

实验得出阈值设置为水平和垂直总像素点平均值的一半，根据该阈值为找出数字区域，如图 6 所示。

得到卡号区域后，需要进一步进行二值化，消除噪声，输入二值图像，得到数字区域的图像，如图 7 所示。



Figure 4. Original image

图 4. 原始图像



Figure 5. Image after correction

图 5. 矫正之后的图像



Figure 6. Area of card number

图 6. 获取数字区域的图像

6217 0029 4000 0410 265

Figure 7. Binaryzation and noise remove

图 7. 二值化、去除噪音后的图像

2.4. 卡号切割

具体做法是利用拉普拉斯算子[10]获取图像梯度。再通过对比消除背景拿到数字区域。

对定位后的图像进行粗分割，经过投影后会存在波峰和波谷，数字与数字之间的间隙会导致波谷的产生，把明显的峰谷的区域切割，从而就会得到好几块用数字的整体区域的图，图 4 所对应的垂直投影图如图 8 所示。粗分割会存在数字的间隙不够明显，再接着在对银行卡的整体数字区域的图进行细分割，在加强阈值后分割成单个字符[11]。经过二次分割后能够得到较为精确的字符，那么这样就全部统一成单个字符了，如图 9 所示。

需要注意的是，做图像分割需要选择合适的方法，银行卡通常是左右型的，我这里选择的是进行垂直分割。对于上下型的，则做水平分割即可。在其他的分割实验中，有时候对于复杂的图形，单单选择

其中一种分割可能是不足的，有时候需要考虑采用水平和垂直分割的结合使用。

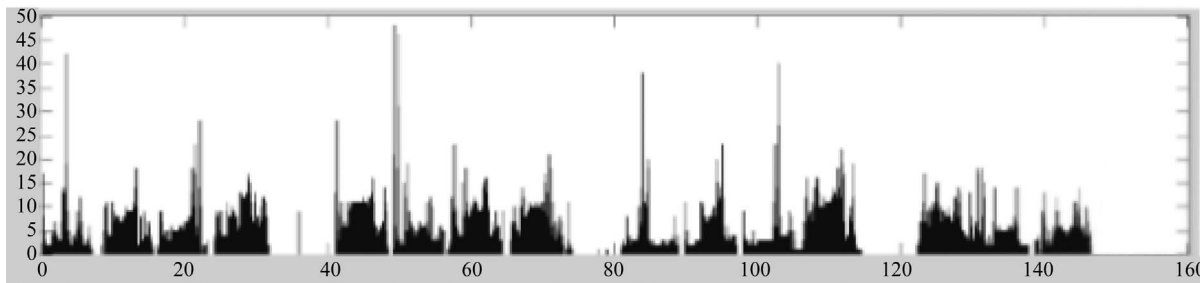


Figure 8. Vertical projection map

图 8. 垂直投影图



Figure 9. Image of single number

图 9. 切割成单个字符的图像

2.5. 归一化

在进行字符识别之前，由于字符图片的大小不一致，不方便进行字符识别，需要对图片进行处理，得到与模型训练集图片大小一致的字符图片。OpenCV 中提供 ReSize 函数可直接对图片进行大小调整，这里统一调整为 60 * 40，实现字符图片大小一致，从而方便后续的字符识别。

2.6. 字符识别

银行卡卡号识别的方法有：模版匹配识别法、神经网络识别法、特征统计匹配识别法。文中所选用的是神经网络识别。神经网络法是把待处理图像输入网络，由网络自动实现特征提取直至识别。卷积神经网络是通过多个二维平面组成的神经网络，这种网络互连较多，待处理信息量大。这种方法无需特征提取，由网络自动识别字符，抗干扰性能好，识别率高[12] [13]。这里是通过实验证明在采取神经网络识别的方法相对来说是合适的，识别率和可操作性都比较强。

在模型训练完成后，就可对前步骤切割好的单个字符图像进行预测。

模型的训练过程和识别如图 10、图 11 所示。

最后我们在训练好模型之后，只需要把训练好的模型转换成 TFLite 模型交给移动端使用即可。

3. 实验部分

银行卡需要找到银行卡区域，在对银行卡进行定位之后，银行卡由于人工拍照，会存在一定角度的倾斜和不完整性的误差，给银行卡的定位造成一定的困难。在定位之前，需要对银行卡进行矫正，再做下面的工作。需要说明对图像的预处理和识别，由于移动端屏幕和图片展示的方便性不高，这里采用 PC 端的方式展示。利用移动设备对银行卡进行拍照，这里选用的银行卡都是完整的，并且拍照也是完整的。定位银行卡操作如下：

(1) 数字识别和分割。主要通过对银行进行矫正，然后定位，分割出数字区域，最后再进一步的对数字区域中的数字进行一一切割，提取出单个的数字字符。

```

Run: train_digit x demo x
11520/12693 [=====>...] - ETA: 0s - loss: 0.0300 - acc: 0.9947
11776/12693 [=====>...] - ETA: 0s - loss: 0.0293 - acc: 0.9948
12032/12693 [=====>...] - ETA: 0s - loss: 0.0287 - acc: 0.9949
12288/12693 [=====>...] - ETA: 0s - loss: 0.0281 - acc: 0.9950
12544/12693 [=====>...] - ETA: 0s - loss: 0.0275 - acc: 0.9951
12693/12693 [=====] - 4s 340us/step - loss: 0.0272 - acc: 0.9952 - val_loss: 0.0049 - val_acc: 0.9979

 32/1411 [.....] - ETA: 1s
192/1411 [==>.....] - ETA: 0s
352/1411 [=====>.....] - ETA: 0s
512/1411 [=====>.....] - ETA: 0s
672/1411 [=====>.....] - ETA: 0s
832/1411 [=====>.....] - ETA: 0s
992/1411 [=====>.....] - ETA: 0s
1152/1411 [=====>.....] - ETA: 0s
1312/1411 [=====>...] - ETA: 0s
1411/1411 [=====] - 1s 385us/step
Test loss: 0.004948801260234029
Test accuracy: 0.9978738483345145

Process finished with exit code 0

```

Figure 10. Process of training

图 10. 卡号训练图



Figure 11. Card num recognition

图 11. 卡号识别图

银行卡卡号识别的方法有：模版匹配识别法、神经网络识别法、特征统计匹配识别法。

基于特征匹配效果好，但是算法复杂，耗时长，难以实现实时性，由于移动设备普遍的处理能力不足，一般都由服务器处理这些图像，再将预测结果返回至 app，此方法需要搭建一个网络良好且算力充足的服务器，所以不使用。

Tesseract 识别法局限性较大，其原理是计算数据集图像与被预测图像的匹配程度，选择概率最大的结果作为输出，算法简单，耗时短，但由于银行卡随着时间推进会进行不断更新，其卡面样设计式不计其数，这种方式需要实时的更新银行卡数据集才能准确预测图像。

本次实验对于机器学习和 Tesseract 进行了测试，从各个步骤的时间以及识别准确率上面进行的统计。Tesseract 着重对一种字体进行识别，这里选取的是 OCR-A 字体，以及普通的喷涂型和凹凸亮色的银行卡进行识别。OCR-A 字体的银行卡由于特殊收集数量不多，这里选用了 100 张清晰的银行卡，为了保证数量的一致，其他的喷涂型和凹凸亮色型均为 100 张。同时为了对比两种识别方式的抗干扰能力，对之前的 100 张图片加入了干扰因素，如调整角度，增添环境光等操作。识别时间如表 1 所示，识别时间均为平均值，识别准确率如表 2 所示，识别准确率均为平均值。

Table 1. Identification time comparison table

表 1. 识别时间对比表

识别方式	检测轮廓	切割单个字符	识别	平均总用时
Tesseract 匹配	17.9 ms	19.9 ms	5.0 ms	42.89 ms
深度学习神经网络	138.72 ms	174.96 ms	10.4 ms	321.7 ms

Table 2. Comparison table of recognition accuracy

表 2. 识别准确率对比表

识别方式	OCR-A	普通喷涂型	凹凸亮色型	平均识别准确率
Tesseract 匹配	99.9%	0%	0%	33.3%
深度学习神经网络	93.1%	93%	91.9%	92.7%
Tesseract (干扰)	64.8%	0%	0%	21.6%
深度学习(干扰)	92.4%	92.9%	91.1%	92.1%

通过实验结果可以得出结论，Tesseract 的识别速度虽然优于机器学习，但是只能针对某一类特定的字体或者某一类特定的银行卡模板。因此，Tesseract 这种识别方式的通用性太低，而且对于多种不同字体的银行卡进行识别时，需要相当大的工程量，而且扩展性很差。此外，这类算法限制条件较多，易受光照，拍摄角度等因素影响，鲁棒性也很差。深度学习不依赖特定的模板和字体，同时能适应各种复杂的光线、角度变化，鲁棒性较强，同时可以进行数据增强扩大训练集，进一步获得更高的抗干扰能力与识别准确率。综上所述，本文方法比 Tesseract 代表的经典识别算法更好。

(2) 模型训练。这里主要包括数据增强。

本实验基于 TensorFlow + keras 深度学习框架，对银行卡卡号区域截取的逐行的单个的数字进行预测。

在该研究方法中选用的训练集来自于 900 张银行卡卡号的单个数字图片，由于该训练集数量不足，容易出现过拟合的问题，所以我们使用数据增强来防止过拟合现象。本文方法通过经典的图像处理操作从原始图像中获额外的训练数据，具体包括翻转，旋转，缩放，加噪，平移等操作，将一张图片增加到 16 张，最后的数据集中，单个数字的训练图片达到 1000 张以上，如图 12 所示。

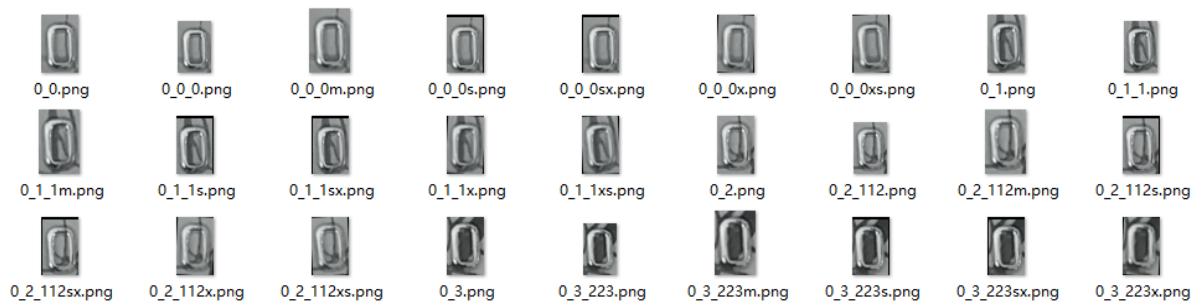


Figure 12. Digital image enhancement

图 12. 数据增强图

(3) 模型的移植。

TensorFlow lite 是 Google I/O 2017 大会上的其中一个重要宣布，使用 TFLite 模型，开发者可以在移动设备上部署人工智能。

对原本训练好的 keras 模型的 3 个文件，分别是 meta, ckpt, checkpoint 文件进行 freeze 操作合并，得到一个变量值和运算值结合的文件，这是将变量值固定在图中的操作，方便直接转换 TFLite 模型，生成 pb 文件。

使用 toco 指令将固化模型 pb 文件输出为 TFLite 模型，放入安卓工程即可使用。

(4) 字符识别。

在安卓端加载 TFLite 解释器 org.tensorflow.lite.interpreter，使用训练好的 TFLite 模型对预处理步骤切割出的单个数字图像进行循环预测，最后整合输出卡号至 app 的 textbox 文本框。

4. 结果与讨论

该研究方法的实验平台为 Windows10 64 bit 操作系统，硬件为 Intel Core I5-6200U 2.40 GHZCPU, 8 G 内存和 NVIDIA 920 M 图形处理器，深度学习框架采用 Tensorflow + Keras [14]。Android Studio 采用的是 3.5 的版本，移动端使用的深度学习框架为 Tensorflow Lite。本文中对于移动端的操作流程如图 13 所示。

根据实验结果我们选择使用耗时较短精度较高的神经网络识别法。银行卡识别的结果如图 14 所示，第一部分为加载的银行卡图片，能够通过导入或者拍照获取，第二部分是进行银行卡号定位得到的图像，第三部分是识别的卡号部分。在这里采用移动端进行实验的方法能够增强用户体验感以及能够提高实用性和可操作性，并且移动端的使用更加符合用户的要求。Android Studio 是一款实用性很强的移动端编译平台[15]，可以直接导入 OpenCV。经过大量的实验发现，对于喷涂型的准确率可以达到 93%左右，而对于亮色凹凸类型的在 92%左右，这里的识别成功的正确定义是以整张银行卡的卡号为标准，后者准确率低的原因主要是因为亮色凹凸类型受背景的影响较大，某些卡号甚至和背景几乎融为一体，造成了定位和分割的误差出现。综合两种类型的银行卡，造成误差出现的共同原因主要是因为银行卡的背景图像复杂，在一些情况下，对银行卡卡号的细分割难以做到完全正确，甚至有些时候出现漏检的情况。因此在细分割的部分还需要进一步的优化。

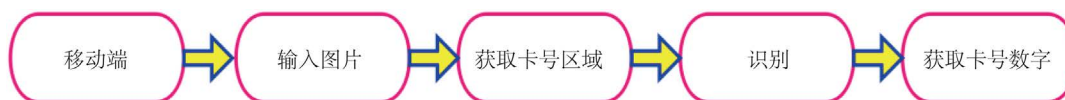


Figure 13. Operational process

图 13. 移动端操作图

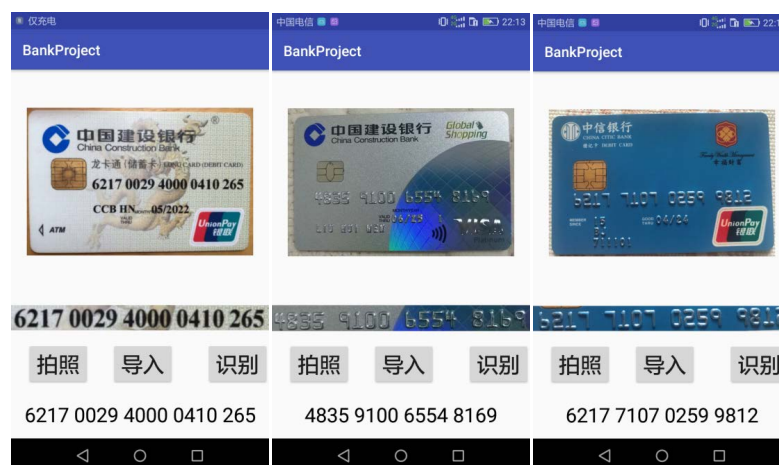


Figure 14. Bank card recognition on mobile terminal

图 14. 移动端识别结果图

5. 结论

在该研究中,提出了一种利用移动端拍照加载银行卡图片,利用 OpenCV 和卷积神经网络对图片进行预处理和识别的方法。在本方法中能够有效地通过 OpenCV 封装好的图像处理算子对银行卡进行处理,利用 CNN 进行识别,并且简单有效。经过移动端的多次实验,能够取得预想的效果,证明该研究方法是有效的并且识别准确率是足够高的,以及识别范围具有通用性,能够很好地应用于移动端。同时,本文提供的方法也存在一定的局限性,分割和识别的时间等步骤仍然需要继续提高。

基金项目

湘南学院自然科学基金(062, 4311803)。

参考文献

- [1] 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述 [J]. 数据采集与处理, 2016, 31(1): 1-17.
- [2] 张曦元. 基于 OpenCV 的智能车牌检测系统[J]. 电脑知识与技术: 学术版, 2019(5): 225-226.
- [3] Zhang, H., Zhao, K., Song, Y.-Z., et al. (2013) Text Extraction from Natural Scene Image: A Survey. *Neurocomputing*, **122**, 310-323. <https://doi.org/10.1016/j.neucom.2013.05.037>
- [4] 朱怀中. 基于 Android 的手机 OCR 识别技术设计与实现[J]. 电子科技, 2012, 25(9): 45-48.
- [5] 何朔. 移动支付的沿革与发展探究[J]. 中国信用卡, 2008(14): 47-53.
- [6] 文章, 张欣, 周昌顺, 等. 一种基于 Canny 的边缘检测改进算法[J]. 通信技术, 2017, 50(10): 2236-2240.
- [7] 游达章, 张建钢, 甘勇. 位图图像灰度化的方法及编程实现[J]. 广西科技大学学报, 2004, 15(1): 23-26.
- [8] 丁怡心, 廖勇毅. 高斯模糊算法优化及实现[J]. 现代计算机(专业版), 2010(8): 76-77+100.
- [9] 孙丰荣, 刘积仁. 快速霍夫变换算法[J]. 计算机学报, 2001, 24(10): 1102-1109.
- [10] 王卜堂, 杨善林. 基于 Gauss-Laplace 算子的灰度图像边缘检测[J]. 计算机工程与应用, 2003, 39(26): 132-134.
- [11] 王磊. 基于 MATLAB 和 LabVIEW 的车牌识别系统[J]. 现代农业研究, 2018(12): 129-130.
- [12] 洪洋, 葛振华, 王纪凯, 等. 基于深度卷积神经网络的验证码识别[C]//第 19 届中国系统仿真技术及其应用学术年会.
- [13] 李翌昕, 邹亚君, 马尽文. 基于特征提取和机器学习的文档区块图像分类算法[J]. 信号处理, 2019, 35(5): 747-757.
- [14] 来学伟. TensorFlow 读取数据在简单图像识别中的应用[J]. 现代信息科技, 2019, 3(12): 98-99.
- [15] 胡清华. 基于 Android 平台的软件开发方法研究[J]. 数码世界, 2018(1): 25.