

哈爾濱工業大學

畢業設計（論文）

題 目 基於神經網絡的車牌
識別系統設計

專 業 測控技術與儀器
學 號 150220213
學 生 樓盈楓
指 導 教 師 張 虹
答 辯 日 期 2019 年 6 月

摘 要

现代的社会、交通不断发展，车辆越来越普及，车牌识别系统被广泛使用。近年来，神经网络的发展也越来越好。

本文设计了基于神经网络的车牌识别系统，以 Python 为编程语言，实现了车牌图像的预处理，包括灰度化、滤波、边缘检测、二值化以及闭运算；在此基础上进行车牌的定位，先由几何形状进行定位，再通过简单的卷积神经网络进行更加精确地定位；然后运用投影法进行字符的分割；再使用 Tensorflow 库建立卷积神经网络对车牌的字符进行训练识别，并根据训练的精确度以及训练时间调整网络参数，寻找最佳方案，测试之后准确率达到了 91%；最后设计简洁、形象的基于 Tkinter 的用户交互界面，让用户能够很快地使用本系统。

关键词：图像识别；车牌识别系统；卷积神经网络；

Abstract

In modern society, traffic continues to develop, vehicles become more and more popular, so license plate recognition system is widely used. Also, the development of neural network is getting better and better in recent years. .

In this paper, a license plate recognition system based on neural network is designed. Using Python as programming language, the pre-processing of license plate image is realized, including grayscale, filtering, edge detection, binarization and closed operation, on the basis of which license plate location is carried out. Firstly, the geometric shape is located, then the simple convolution neural network is used to locate the plates more accurately, then the projection method is used to segment the characters, and then the convolution neural network is established by Tensorflow library to train and recognize the characters of the license plate. According to the accuracy of the training and the training time, the network parameters are adjusted to find the best scheme, After the test, the accuracy rate reached 91%.. And finally the user interaction interface which is based on Tkinter is designed, so that users can use the system quickly.

Keywords: image recognition, license plate recognition, convolution neural network.

目 录

摘要	I
Abstract	II
 第 1 章 绪论	
1.1 课题背景及研究的目的和意义.....	1
1.1.1 课题背景.....	1
1.1.2 研究目的和意义.....	2
1.2 车牌识别系统及其相关理论的发展概况.....	2
1.2.1 国外现状分析.....	2
1.2.2 国内现状分析.....	3
1.3 本文的主要研究内容.....	3
 第 2 章 神经网络基本理论	
2.1 引言.....	5
2.2 卷积神经网络的结构.....	5
2.2.1 卷积层.....	5
2.2.2 池化层.....	6
2.2.3 全连接层.....	7
2.3 激活函数.....	8
2.3.1 Sigmoid 函数.....	8
2.3.2 ReLU 函数.....	9
2.4 损失函数.....	10
2.5 优化算法.....	10
2.5.1 随机梯度下降法.....	10
2.5.2 Adam 算法.....	11
2.6 本章小结.....	12
 第 3 章 车牌识别系统总体设计	
3.1 引言.....	13
3.2 图片预处理.....	13
3.2.1 灰度化.....	13
3.2.2 中值滤波.....	14

3.2.3 边缘检测.....	15
3.2.4 二值化.....	16
3.3 车牌定位.....	18
3.3.1 截取车牌区域.....	18
3.3.2 卷积神经网络在车牌区域检测中的应用.....	18
3.3.3 倾斜校正.....	19
3.4 车牌字符分割.....	20
3.4.1 车牌号码组成.....	20
3.4.2 车牌字符分割算法.....	20
3.5 车牌字符识别.....	21
3.6 本章小结.....	21
第 4 章 基于卷积神经网络的车牌字符识别系统设计	
4.1 引言.....	23
4.2 输入数据库.....	23
4.3 卷积神经网络结构的构建.....	24
4.4 卷积神经网络参数确定.....	25
4.4.1 学习率.....	25
4.4.2 批处理参数.....	26
4.4.3 隐层节点数.....	28
4.5 交互界面设计.....	30
4.5.1 Tkinter 概述.....	30
4.5.2 交互界面的设计.....	30
4.6 本章小结.....	33
 结论.....	 35
参考文献.....	36
致谢.....	38

第 1 章 绪 论

1.1 课题背景及研究的目的和意义

1.1.1 课题背景

现代社会已进入信息时代，随着计算机技术，通讯技术和计算机网络技术的发展，自动化的信息处理能力和水平不断提高，并在人们社会生活的各个领域中得到广泛应用，在这种情况下，图像识别技术、自动检测技术作为提供信息来源的重要方法，越来越受到人们的重视。而且由于我国的高速发展，车辆越来越多，随之而来的是高速公路、城市道路、停车场建设越来越多，对交通控制、安全管理的要求也日益提高，智能交通管理^[1]（Intelligent Transportation System, 简称 ITS）已成为当前交通管理发展的主要方向，而车牌识别技术作为智能交通系统的核心，有着举足轻重的作用，它在高速公路、城市道路和停车场等项目管理中占有无可取代的重要地位。

车牌识别系统（License Plate Recognition, 简称 LPR）具有广泛的应用范围，主要应用于：高速公路收费、监控管理；小区、停车场管理；城市道路监控、违章管理；车辆验证、统计及安全管理。

识别系统中最重要、最核心的算法是字符识别的算法，现在的科学研究有很多种，包括模式识别法、神经网络等，今年，神经网络的发展使人工智能技术取得了巨大的进步，在图像分类方面表现出巨大的优势，本文中将选择神经网络解决车牌识别中的字符识别问题。

神经网络即人工神经网络，是由人工神经元互联组成的网络，从仿生学的角度对人脑的神经系统进行模拟，使机器具有人脑的感知学习和推理的能力。神经网络模式识别方法是近些年兴起的模式识别领域的一个新的研究方向，具有很强的自学习性、自组织性、容错性、高度非线性、高度鲁棒性、联想记忆功能和推理意识功能^[2]等。近几年由于科学的发展，训练数据逐渐增多，使训练数据具有代表性，因此神经网络的发展也逐渐成为科技发展的必然趋势。

在本研究中，车牌识别的准确率和快速性是衡量交通监视系统好坏与否的重要标准，而且车牌的字符具有明显的畸变，而神经网络良好的容错能力和联想能力，因此成为本文讨论的识别方式的重点。

但是，目前在高质量的图像情况下，识别的结果有很高的准确率，但是对于一些复杂的图像，比如图像中有多个车牌，车牌区域图像大小变化，图像分辨率

低，光照变化大灯，传统的车牌识别技术准确率难以提高。基于车牌图像的车牌识别还远未被完美解决，仍然是重要的挑战性研究课题。

1.1.2 研究的目的是和意义

车牌识别是智能交通管理的一个重要组成部分，它在交通控制与监视中有多种应用，如不停车缴费、失窃车辆侦查、停车场管理等等，在通道网络入口和出口处分别安装车辆检测系统。通过对车牌的匹配和智能交通系统网络，对车辆行驶速度、流量进行估计，检测当前的交通拥挤程度，自动提供实时交通信息。它能从图像中自动提取车牌图像，自动分割字符图像，进而对字符进行识别。车辆牌照识别是计算机视觉与模式识别技术在智能交通领域应用的重要课题之一，若能用计算机实时地识别车牌，就可以在无需为车辆加装其他特殊装置的情况下实现对车辆的自动检测，从而给交通系统的自动管理提供极大的方便，因此车辆牌照自动识别系统的实现是推动交通管理计算机化的关键技术之一。

车牌自动识别系统能将输入的汽车图像通过处理识别，输出为几个字节大小的车牌字符串，无论在存储空间的占用上还是与管理数据库相连方面都有无可比拟的优越性。在大型停车场、交通部门的违章检测、高速公路及桥梁的收费站管理等方面，有着广泛的应用前景。车牌识别系统的成功开发大大加速智能交通管理的进程。

1.2 车牌识别系统及其相关理论的发展概况

1.2.1 国外现状及分析

国外对于车牌识别的研究比较早，在 20 世纪 70 年代就已有自动车牌检测系统用于检查被盗的情况，现在，已经达到很高的应用水平。在各种应用中有对静态图像处理系统，如 Huang MuHwang 的《APC_BASED CAR LICENSE PLATE READER》在收费站硬件设施较好的条件下，平均 0.7 秒内识别率达到 97%。

从 20 世纪 90 年代开始，国外对于车牌自动识别的研究逐渐成熟，其主要途径就是对车牌的图像进行分析，自动提取车牌信息，确定汽车牌号。国外现有一些产品：以色列 Hi-Tech 公司的 See/Car System 一系列的产品，新加坡 Optasia 公司的 VLPRS 产品等^[2]。但他们的系统也有很大的缺陷，而且不能识别车牌中的汉字，另外英国、加拿大、德国等西方发达国家都有适合本国车牌的识别系统。但基本上都是基于车辆探测器的系统，设备投资巨大。

尽管各国在技术上取得了许多进展，但基于图像处理和模式识别的定位算法在应对复杂背景变化时仍旧存在着许多缺点和不足。如定位上的偏差给字符分割

带来的较大困难，最终导致字符识别失败。并且由于各个国家文字不同，车牌的规格、字符大小也各不相同，因此不能直接使用别的国家研究成熟的识别算法，现在，部分国家采取制定新的车牌标准的办法为定位和识别提供了大大的方便，降低了算法复杂度，并使系统实时性得到了提升。

机器学习是人工智能的一个重要分支，近年来，深度学习包括卷积神经网络以及深度置信网络，取得了巨大的成功，比如由谷歌公司研发的人工智能机器人^[3]“AlphaGo”，在围棋比赛中连续两次击败了人类的最高水平选手，这个人工智能机器人是基于深度神经网络以及搜索树的，体现了人工智能的高速发展。

1.2.2 国内现状及分析

国内在 20 世纪 90 年代也开始了车牌识别的研究，目前国内比较成熟的产品主要是中科院自动化研究所汉王公司的“汉王眼”，除此之外的亚洲视觉科技有限公司、深圳市吉通电子有限公司、中国信息产业部下属的中智交通电子系统也都有自己的产品。各大高校的实验室也都做过类似的研究。通常处理时为了提高系统的识别率，都采用了一些硬件的探测器和其他的辅助设备。国内也有很多关于车牌自动识别的文章，但大多因为条件不同而使用范围有限，车牌矫正的效果都不够理想，较少采用自适应算法，因此在复杂背景下和存在多种颜色车辆牌照的情况下的定位和识别效果均不理想。

除了技术层面上的问题，我国的车牌照上有很多特殊的特点，客观上给定位和识别带来了较大的困难。比如牌照中有多个汉字，并且车牌有蓝、黄、黑、白等多种颜色，车辆牌照的车型有多种牌照格式，不是严格统一的，因此给识别造成了较大的困难。

因此结合国内外的技术水平和法律法规差别，研究一套适合我国交通运输先转搞得车牌定位与识别系统是一项很有意义并且具有挑战性的工作。

我国对 CNN 图像识别的研究也逐渐成熟，主要是根据图像的特征对图像进行分类，把有相同特征的归为一类，利用图像识别技术，可以帮助我们快速检索到所需信息，例如淘宝的淘立拍功能可以通过识别手机中的照片，快速匹配到具有相似属性的商品，除此之外，此项技术还广泛运用到二维码识别、条码识别以及人脸识别中区。

1.3 本文的主要研究内容

本课题的研究内容主要是针对车牌识别的预处理、车牌定位、车牌字符分割以及字符识别的神经网络这几大部分进行展开，研究各个部分的算法，其中最主要的算法是字符识别中使用的卷积神经网络算法，为了达到识别的目的，首先构

建卷积神经网络的结构，再对其进行训练，对网络中的参数，包括学习率、卷积核的大小和个数、批处理参数以及隐层节点数等进行优化，并保存训练完成的模型，进行字符识别，最后设计车牌识别系统的交互界面，使系统更加完整。

本文主要包括以下三部分内容：

- （1）神经网络基本理论；
- （2）车牌识别系统的各部分的设计以及算法实现；
- （3）卷积神经网络结构的构建以及参数的调整。

第 2 章 神经网络基本理论

2.1 引言

人工神经网络是受人脑组织的生理学知识启发而创立的，是 20 世纪 80 年代以来人工智能领域兴起的研究热点。它从信息处理角度对人脑神经元网络进行抽象，建立某种简单模型，按不同的连接方式组成不同的网络。在工程与学术界也常直接简称为神经网络或类神经网络。

神经网络是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的输出函数，称为激励函数。每两个节点间的连接都代表一个对于通过该连接信号的加权值，称之为权重，这相当于人工神经网络的记忆。网络的输出则依网络的连接方式，权重值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近，也可能是对一种逻辑策略的表达。可大规模的并行处理和分布式信息存储，良好的自适应能力。

神经网络是一种机器学习算法，其中卷积神经网络（Convolutional Neural Networks, CNN）是一类包含卷积计算且具有深度结构的前馈神经网络^[4]，是深度学习的代表算法之一。卷积神经网络起源于对大脑的视觉皮层的研究，现在经常被用于图像识别。卷积神经网络类似于生物神经网络的权值共享结构，可产生生物视觉中局部感受野效应，不仅降低了网络的模型复杂度，减少了权值数量，且对图像平移、比例缩放、倾斜等具有不变性，具有很好的特征提取性能，因此本文采用卷积神经网络对车牌的字符进行识别。

2.2 卷积神经网络的结构

卷积神经网络由卷积层、池化层、全连接层构成，如图 2-1 所示，其中卷积层和池化层可以循环多次。

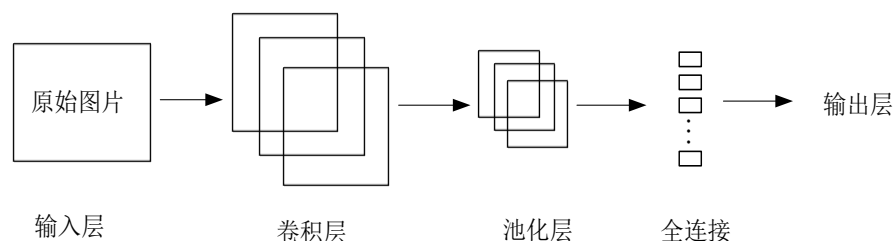


图 2-1 卷积神经网络基本结构

2.2.1 卷积层

卷积层是神经网络中最重要的部分。卷积层中的每一个节点的输入只是上一

层神经网络中的一小部分，即卷积核。卷积层试图将神经网络中的每一个卷积核进行更加深入的分析从而得到抽象程度更高的特征。一般来说，通过卷积层处理的节点矩阵会变得更深。卷积层是 CNN 卷积神经网络中最重要的部分。第一卷积层的神经元不会连接到输入图像中的每个像素，而是只与其接受野内的像素相连接。反过来，第二卷积层的每个神经元仅连接到位于第一层中的一个小矩阵内的神经元。这种结构允许网络集中在第一个隐含层的低阶特征中，然后在下一个隐含层中将他们组装成比较高阶的特征，即先对图片中的一个特征先进行局部感知，然后更高层次对局部进行综合操作。卷积层的计算公式为：

$$X_m^j = f(\sum_{i \in M_m^i} X_{m-1}^i * K_m^{ij} + b_m^j) \quad (2-1)$$

式中 X_m^j 表示 m 层第 j 个特征图， M_m^j 表示 m 层第 j 个特征图对应的输入特征集合， K_m^{ij} 表示 $m-1$ 层第 i 个特征图和第 j 特征图相连的卷积核， b_m^j 为 m 层第 j 个特征图的偏置。

若有一个 5×5 的图像，使用一个 3×3 的卷积核进行卷积，设置步长为 1，卷积核就从原图像的左上角的 3×3 区域进行卷积，然后就可以得到第一个输出的值，然后卷积核向右移动一步，进行卷积，可以的到第二个值，以此类推，最后可得到一个 3×3 的输出，卷积过程如图 2-2 所示。

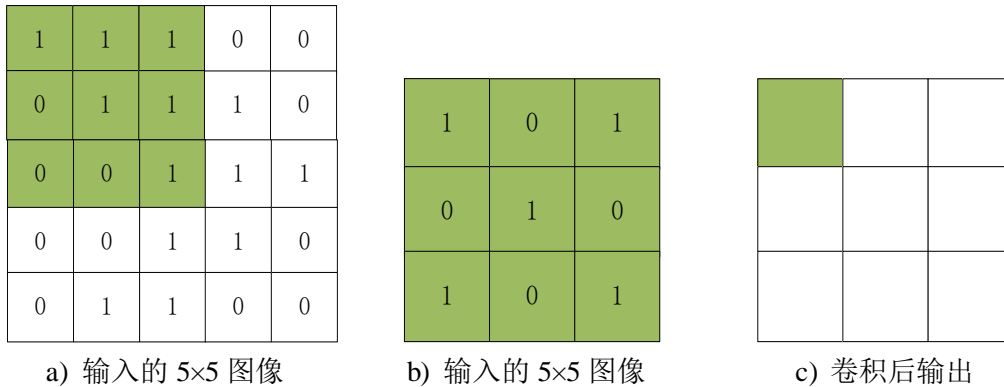


图 2-2 卷积过程

2.2.2 池化层

在训练期间，卷积层需要大量的内存，因为反向传播的方向传递需要在正向传递期间计算所有的中间值，因此我们需要用池化层，对输入图像进行二次采样以减小计算负载、内存利用率和参数数量，池化层是用来缩小矩阵的大小的，它不会改变输入矩阵的深度，可以认为是将一张分辨率较高的图片转化为分辨率较低的图片，即在保留图像特征的前提下降低维度。具体操作是划分卷积层得到的

小矩阵，每个小矩阵中用元素的平均值或者最大的元素代替，得到更小的新矩阵。

常用的池化方法有平均池化和最大池化。设置池化层为 2×2 ，步长为 2，首先在左上 2×2 的区域选最大值或者平均值，再向右平移两步，选出最大值或平均值，以此类推，可实现降维。平均池化会保留一些无效信息，而最大池化可以去除这些无效信息的干扰。

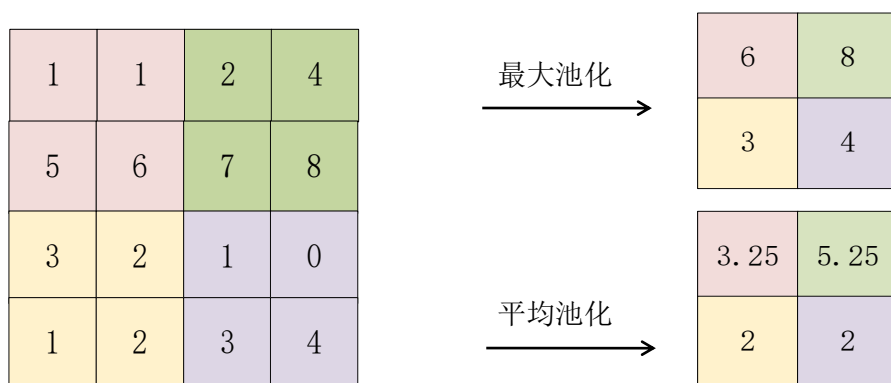


图 2-3 最大池化和平均池化过程示意图

2.2.3 全连接层

全连接层是在卷积神经网络的最后，在经过多轮卷积层和池化层处理之后，由 1 到 2 个全连接层来给出最后的分类结果。经过几轮的卷积层和池化层的处理之后，可以认为图像中的信息已被抽象成了信息含量更高的特征。我们可以将卷积层和池化层看成自动图像特征提取的过程。在特征提取完成之后，仍然需要使用全连接层来完成分类任务。接着进入全连接层，它的实质是 BP 神经网络，即误差后向传播神经网络，一般包含 3 个层：输入层、输出层以及隐含层。其结构图如图 2-4 所示。

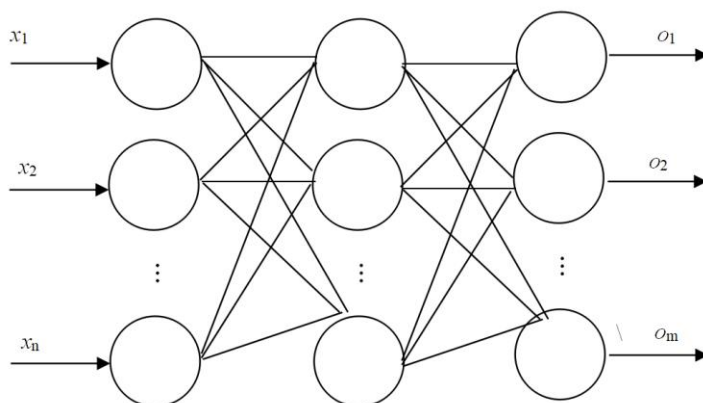


图 2-4 全连接层结构图

定义网络输入量为 (x_1, x_2, \dots, x_n) ，输出量为 (o_1, o_2, \dots, o_m) ，通过输入量数目

n 和输出量数目 m 决定隐含层神经元节点数量 l 。输入层和隐层节点之间的连接权值为 ω_{ij} ，隐层和输出层节点的连接权值为 ω_{jk} ，下面分析各层信号之间的关系。

对于输出层，有：

$$o_k = f(net_k), \quad k = 1, 2, \Lambda, m \quad (2-2)$$

$$net_k = \sum_{j=0}^k w_{jk} y_j, \quad k = 1, 2, \Lambda, m \quad (2-3)$$

式中 ω_{jk} 是隐层和输出层节点之间的连接权值。

对于隐层，有：

$$y_i = f(net_i), \quad i = 1, 2, \Lambda, l \quad (2-4)$$

$$net_j = \sum_{i=0}^n \omega_{ij} x_i, \quad j = 1, 2, \Lambda, l \quad (2-5)$$

以上式子中， $f(x)$ 均为单极性 Sigmoid 函数：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2-6)$$

在正向传播完成之后，要进行反向传播。

当网络输出与期望输出不等时，存在输出误差 E ，定义如小：

$$E = \frac{1}{2} \sum_{k=1}^l (d_k - o_k)^2 \quad (2-7)$$

式中， o_k 是输出层的第 k 个神经元的输出， d_k 是第 k 个节点的期望输出。

这个误差是各层权值的函数，因此调整权值可以改变误差，原则是使误差不断减小，因此应使权值的调整量与误差的负梯度成正比，即：

$$\Delta \omega_{jk} = -\eta \frac{\partial E}{\partial \omega_{jk}} \quad j = 0, 1, 2, \Lambda, l; k = 1, 2, \Lambda, m \quad (2-8)$$

$$\Delta \omega_{ij} = -\eta \frac{\partial E}{\partial \omega_{ij}} \quad i = 0, 1, 2, \Lambda, n; j = 1, 2, \Lambda, l \quad (2-9)$$

式中，负号表示梯度下降，常数 $\eta \in (0, 1)$ 表示比例系数，在训练中反映学习速率。

2.3 激活函数

激活函数的主要作用是完成数据的非线性变换，解决线性模型的表达、分类能力不足的问题，将输入数据归一化，将输入数据映射到某个范围内。

2.3.1 Sigmoid 函数

Sigmoid 函数的特点是会把输出限定在 0~1 之间，使得数据在传递过程中不容

易发散。此函数的表达式为：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2-10)$$

Sigmoid 函数的单位阶跃响应图像如图 2-5 所示。

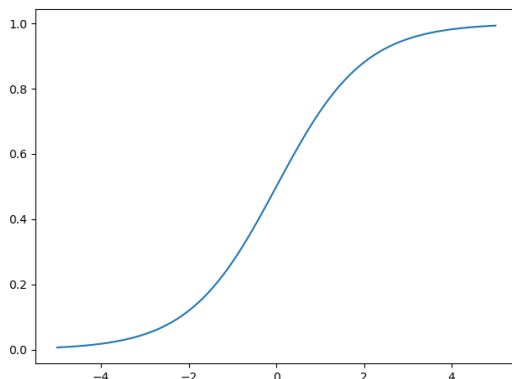


图 2-5 Sigmoid 函数的阶跃响应图像

但是 Sigmoid 函数的缺点是容易过饱和，丢失梯度。因为神经元的活跃度在 0 和 1 处是饱和的，梯度接近于 0，这样在反向传播时，很容易出现梯度消失的情况。

2.3.2 ReLU 函数

ReLU 函数的特点是当输入信号小于 0 时，输出为 0；当输入信号大于 0 时，输出等于输入。函数表达式为：

$$y = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2-11)$$

ReLU 函数单位阶跃响应曲线如图 2-6 所示。

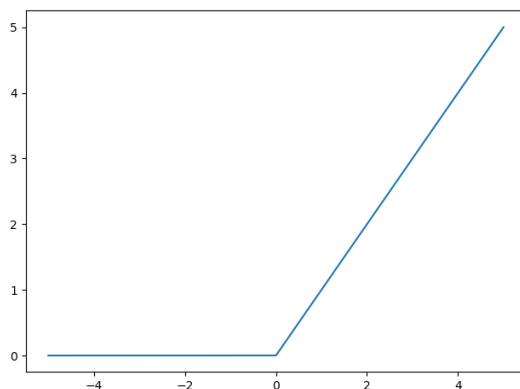


图 2-6 ReLU 激活函数响应曲线

ReLU 函数是部分线性的，不会出现过饱和现象，用此函数得到的随机梯度下降法的收敛速度比较快；并且 ReLU 只需要一个阈值就可以得到激活值，不需要像 Sigmoid 一样需要复杂的指数运算。但要注意在使用 ReLU 函数时，要设置一个较小的合适的学习率参数，避免 ReLU 神经元失去作用。

2.4 损失函数

损失函数是用来评估模型的预测值与真实值之间的差异程度。损失函数也是神经网络优化的目标函数，神经网络训练或者优化过程就是最小化损失函数的过程，损失函数越小，说明模型的预测值就越接近真实值，模型的准确性也就越好。

在神经网络图像分类问题中，经常使用交叉熵损失函数，其表达式为：

$$C(w, b) = -\frac{1}{n} \sum_x [y \ln a - (1 - y) \ln(1 - a)] \quad (2-12)$$

其中 C 表示交叉熵的值， w 表示权重， b 表示偏置， n 表示样本的总数， y 表示真实值， a 表示预测值。

若神经元的激活函数为 Sigmoid 函数 $f(x)$ ，用交叉熵对 w 求偏导：

$$\frac{\partial C}{\partial w} = -\left[\frac{y}{f(Z)} - \frac{(1-y)}{1-f(Z)}\right] \frac{\partial f}{\partial w} \quad (2-13)$$

式中， $Z = wx + b$ 。

可简化为：

$$\frac{\partial C}{\partial w} = (f(Z) - y)x \quad (2-14)$$

同理可得参数 b 的偏导为：

$$\frac{\partial C}{\partial b} = f(Z) - y \quad (2-15)$$

参数 w 、 b 的偏导数与期望输出和输出的差成正比关系，即差值越小，偏导数越小，收敛速度会缓；变差值越大，偏导数越大，收敛速度加快。因此交叉熵损失函数前期随期望输出和实际输出差别大而收敛较快，后期随差值的减小而放缓学习速度，表现出前快后慢的收敛现象。

2.5 优化算法

在调整模型更新权重和偏差参数的方式时，要选择合适的优化算法能让模型产生更好、更快的效果。优化算法是通过改善训练算法，来最小化损失函数的值。

2.5.1 随机梯度下降法

为了得到最优的权值，需要得到损失函数之后使用优化算法得到最小的函数

损失值。梯度下降算法（stochastic gradient descent, SGD）是卷积神经网络中使用比较广泛的优化算法，此算法每次随机选择一个样本来迭代更新一次，而不是针对所有的样本，这种算法明显降低了计算量，加快了训练速度。相关公式如下：

$$\begin{aligned} g(\varphi) &= \sum_{j=0}^n \varphi_j x_j \\ h(\varphi) &= \frac{1}{2m} \sum_{i=1}^m (y_i - g_{\varphi}(x^i))^2 \\ \varphi &= \varphi - \eta \nabla_{\varphi} h(\varphi) \end{aligned} \quad (2-16)$$

式中， φ 代表网络参数权值， ∇_{φ} 表示的是梯度， $h(\varphi)$ 代表损失函数， $g(\varphi)$ 代表目标函数， y_i 代表第 i 个样本的样本值， m 表示整个迭代进行的总次数， η 表示梯度下降中的步长即学习率。如果学习率过小，则会需要多次迭代才能找到最优解且会降低网络的收敛速度；但若增大学习率，会加快网络的训练速度，可能找不到最优解，因此 SGD 算法是否有效取决于是否选取合适的学习率。

2.5.2 Adam 算法

Adam（Adaptive Moment Estimation）算法^[3]是在梯度下降法的基础上融合了一种加速梯度下降的算法，Adam 在梯度下降法的基础上引入了平方梯度，并且可以为每个参数计算出不同的学习率，有自适应学习率的特点，会跟踪过去梯度的指数衰减平均值，也会跟踪过去梯度平方的指数衰减平均值^[5]，算法公式为：

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2-17)$$

$$\hat{s}_t = \frac{s_t}{1 - \beta_2^t} \quad (2-18)$$

式中， m_t 为一阶动量项， s_t 为二阶动量项， \hat{m} 、 \hat{s} 表示修正值。

最终参数更新的公式为：

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{s_t} + \varepsilon} \quad (2-19)$$

式中动量衰减超参数 β_1 通常设为 0.9，缩放衰减超参数通常设为 0.999， ε 为一个很小的常数。

Adam 算法是一个自适应学习速率算法，它需要对学习速率超参数进行微调。

2.5 本章小结

本章阐述了卷积神经网络的基本结构，包括卷积层、池化层以及全连接层，还详细介绍了卷积神经网络算法的理论基础，包括激活函数、损失函数以及优化算法。

第 3 章 车牌识别系统总体设计

3.1 引言

车牌识别系统由图片预处理、车牌定位、车牌字符分割、字符识别以及用户交互界面组成。整体设计如图 3-1 所示。

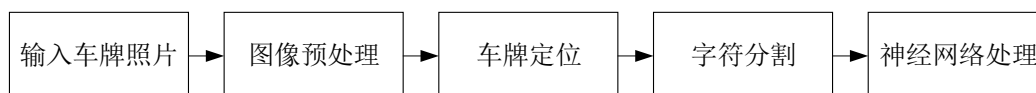


图 3-1 车牌识别系统整体设计

车牌预处理主要由灰度化、滤波、边缘检测、二值化以及闭运算五个步骤完成，找到类似车牌的大致位置，为之后车牌的精确定位做准备。

车牌定位时，根据长宽以及长宽比来判定是否为车牌并对车牌进行倾斜校正。

接着，先水平投影确定车牌上的字符区域，再垂直投影根据投影的宽度确定是否是一个字符。

最后，构建卷积神经网络进行字符识别，并选择概率最大的作为识别结果并输出。

本章主要详细阐述了上述部分的理论依据、具体算法以及处理结果。

3.2 图片预处理

车牌图片预处理的流程框图如图 3-2 所示，主要包括灰度化、滤波、边缘检测、二值化。

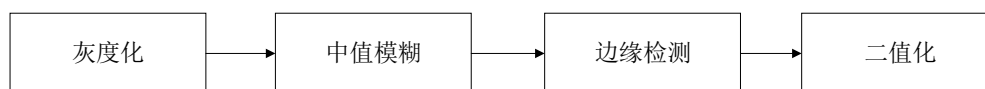


图 3-2 车牌图片预处理流程框图

3.2.1 灰度化

灰度处理是把车牌彩色图像转化为灰度图像的过程，目的使后续的图像的计算量变得少一点。在本设计采用 RGB 彩色模型^[5]计算出彩色图片的灰度值。彩色图像中的每个像素的颜色有 R 、 G 、 B 三个分量决定，而灰度图像是 R 、 G 、 B 三个分量相同的一种特殊的彩色图像，灰度图像的描述与彩色图像一样仍然反映了整幅图像的整体和局部的色度和亮度等级的分布和特征。

利用 RGB 彩色模型算法建立亮度与 R 、 G 、 B 三个颜色分量的关系，以亮度值表达图像的灰度值：

$$I = 0.3 \times R + 0.59 \times G + 0.11 \times B \quad (3-1)$$

式中 I 为亮度值， R 、 G 、 B 为彩色图像的三个颜色分量值。

计算出灰度值后即可进行灰度化，图片灰度化处理之后的图像如图 3-3 所示。



图 3-3 图片灰度化处理与原图对比

3.2.2 中值滤波

图像信息在采集过程中往往受到各种噪声源的干扰，这些噪声在图像上表现为一些孤立像素点，通常噪声点像素灰度与其临近的灰度有显著不同。通常，一般的图像处理后的图像仍然带有后续所不希望夹带的孤立像素点，这种干扰或孤立像素点如不经过滤波处理，会对以后的图像区域分割、分析和判断带来影响。

对受到噪声污染的图像可以采用线性滤波的方法来处理，但是很多线性滤波有低通性，在去噪声的同时也使得边缘模糊了，中值滤波在某些情况下可以做到既去除噪声又保护图像的边缘^[6]，他是一种非线性的去噪声的方法。

中值滤波的原理是把数字图像中的一点的值用该点的一个区域的各个点的值的中值代替。设有一个一维序列 X_1, X_2, Λ, X_n ，取窗口长度为 n (n 为奇数)，对此序列进行中值滤波，就是从输入序列中连续抽出 n 个数， $X_{i-v}, \Lambda, X_i, \Lambda, X_{i+v}$ ，其中 X_i 为窗口的中心值， $v = (n-1)/2$ ，再将这 n 个点的数值按其数值大小排列，取其序号的正中间的那个数为滤波输出为：

$$Y_i = \text{Med}\{X_{i-v}, \Lambda, X_i, \Lambda, X_{i+v}\} \quad (3-2)$$

式中 $i \in Z$ 且 $v = (n-1)/2$,

对数字图像进行中值滤波，实质就是对二维序列的中值滤波，滤波窗口也是二维的，用一个滑动窗口在图像上进行扫描，把窗口内包含的图像像素按灰度级按照升序排列起来，窗口的中心像素灰度就是灰度值居中的像素灰度。

本设计中采用 3×3 的矩形窗口为中值滤波窗口，结果表明中值滤波后，孤立噪声点的干扰被很好地消除了，并且可以将边界信息很好地保护起来，如图 3-4 所示，为消除噪声点后的车牌照片。



图 3-4 中值滤波后的车牌照片

3.2.3 边缘检测

图像边缘检测是检测图像的不连续性，即灰度级具有突变地方，既是一个区域的终结，也是另一个区域的开始。通常使用边缘检测局部算子法来提取边缘，即考察图像的每个像素点在某个区域内灰度的变化，利用边缘临近一阶或者二阶方向导数变化规律检测边缘。

边缘检测算子检查每个像素的领域并对灰度变化率进行量化，通常还包括了方向的确定。在本设计中采用 Sobel 算子进行检测。

Sobel 算子是典型的基于一阶导数的边缘检测算子，由于该算子引入了类似局部平均的运算，因此对噪声具有平滑作用，能很好地消除噪声的影响。

Sobel 算子包含两组 3×3 的矩阵，分别为横向及纵向模板，将之与图像作平面卷积，即可分别得出横向及纵向的亮度差分近似值。实际使用中，常用如下两个模板来检测图像边缘：

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3-3)$$

$$\mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3-4)$$

式中 \mathbf{G}_x 为检测水平边缘的横向模板， \mathbf{G}_y 为检测垂直边缘的纵向模板。

图像中的每个像素点都用这两个模板做卷积，可得到边缘幅度图像，如图 3-5 所示。



图 3-5 边缘检测后的幅度图像

3.2.4 二值化

在经过灰度化、滤波以及边缘检测后的图像中，依然有很多噪点影响识别的结果，为了减少无关部分对识别的影响，需要对图像进行二值化。

基本原理是将一幅图像的所有像素按照阈值来分别用 0 和 1 表示。为了关键在于寻找到一个适合的门限阈值，对于该阈值的点被判为白色像素，小于该阈值的点被判为黑色像素。阈值选高了，虽然背景上的内容少了，但是车牌字符上的信息会出现丢失；阈值选低了，背景上的信息又基本得不到消除，会给车牌定位带来一定影响。经过以上处理，图像中的许多无效信息已经被删除，同时图像的对比度也得了拉伸，因此图像二值化的重点在于快速、理想的找到门限阈值。

本设计采用 Otsu 算法^[4]，这是一种比较成熟、简单的门限阈值获取方法。

灰度值为 i 的点的概率为：

$$p_r(r_q) = \frac{n_q}{n}, q = 0, 1, 2, \dots, L-1 \quad (3-5)$$

式中， n 是图像中的像素总数， n_q 是灰度级为 r_q 的像素数目， L 是图像中所有可能的灰度级数。假设已经选定了一个阈值 k ， C_0 是一组灰度级为 $[0, 1, \dots, k-1]$ 的像素， C_1 是一组灰度级为 $[k, k+1, \dots, L-1]$ 的像素。Otsu 算法选择最大化类间方差 σ_B^2 的阈值 k ，类间方差定义为：

$$\sigma_B^2 = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \quad (3-6)$$

式中， $w_0 = \sum_{q=0}^{k-1} p_q(r_q)$ ， $w_1 = \sum_{q=k}^{L-1} p_q(r_q)$ ， $\mu_0 = \sum_{q=0}^{k-1} qp_q(r_q) / w_0$ ， $\mu_1 = \sum_{q=k}^{L-1} qp_q(r_q) / w_1$ ，

$$\mu_T = \sum_{q=0}^{L-1} qp_q(r_q)。$$

一幅输入图像的直方图按照以方法计算即可得到其最大化 σ_b^2 的阈值，得到合适的二值化效果，如图 3-6 所示。



图 3-6 二值化后的图像

此种方法通过自动选取阈值进行二值化，计算简单，不受图像亮度的影响。最后，进行闭运算，先膨胀运算，再腐蚀运算，将两个细微连接的图块封闭在一起，如图 3-7 所示。



图 3-7 闭运算后的图像

至此，图片预处理的过程全部完成，在经过了车牌灰度化、中值滤波、边缘检测、二值化以及闭运算几个步骤之后，已经初步选出车牌的大概位置，为之后的车牌精确定位做好了准备。

3.3 车牌定位

车辆牌照的定位就是从一幅随即图像中找出一块具有某种特征的区域，该区

域包含了汽车牌照。这种特征就是车牌照本身区别与图像其他部分的特征。

3.3.1 截取车牌区域

车牌定位主要分为获取车牌轮廓、获取轮廓外接矩形，然后通过外接矩形的长、宽、长宽比三个值排除一部分非车牌的轮廓，最后即可选出所需要的车牌部分。车牌定位的实现流程框图^[6]如图 3-8 所示。

我国的车牌长度均为 440mm，宽度均为 140mm，长宽比约为 3.14^[7]，故若得到的矩形轮廓符合三项要求，这个类似车牌的区域就是所需要车牌区域。

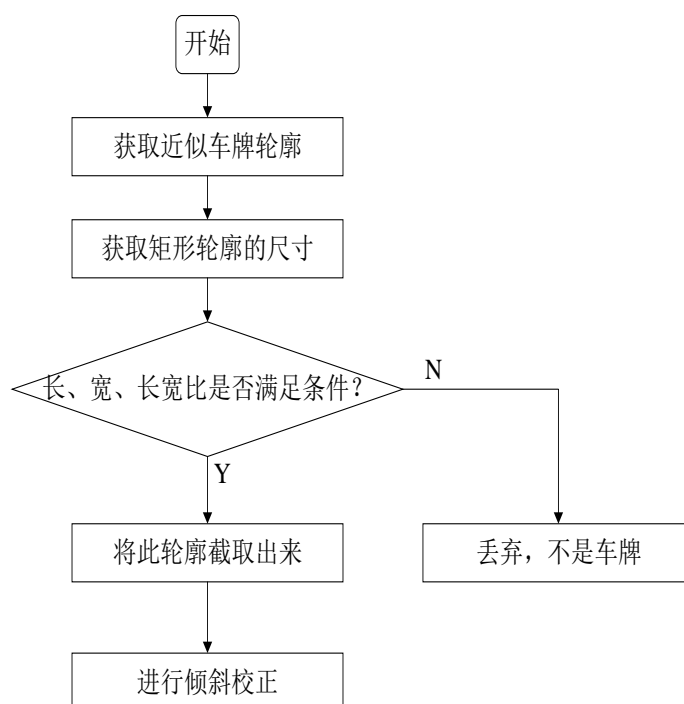
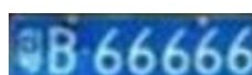


图 3-8 截取车牌区域流程框图

3.3.2 卷积神经网络在车牌区域检测中的应用

在进行车牌的定位实验中，发现有少数车牌图像可能由于周围环境较复杂，只根据矩形框的长、宽以及长宽比，并不能把所有非车牌区域排除掉，如图 3-9 所示，图 a)为选出的车牌区域，图 b)为未排除的非车牌区域。



a) 车牌区域



b) 未排除掉的非车牌区域

图 3-9 未排除掉所有非车牌区域的情况

为解决上述问题，构建一个比较简单的卷积神经网络^[8]，对整个车牌图片进行训练。数据库为 1916 个车牌照片以及 3978 个非车牌区域，共 5894 个图像，大小

为 136×36 ，设置两层卷积的卷积神经网络，选择 3×3 的卷积核，第一层输出 $136 \times 36 \times 32$ ，在经过 2×2 的池化层，输出为 $68 \times 18 \times 32$ ，再进入下一层卷积，输出为 $68 \times 18 \times 64$ ，经过池化，输出 $34 \times 9 \times 64$ ，再进入全连接层，选用 ReLU 激活函数、交叉熵损失函数以及 Adam 优化算法进行训练，设置学习率初始值为 0.001，批处理参数为 100，进行训练，并保存模型，最后可以基本排除所有非车牌区域。

3.3.3 倾斜校正

在选出车牌后，为了给之后的字符分割做准备，需要对截取出来的车牌进行倾斜校正。

车牌图像的倾斜主要有水平倾斜和竖直倾斜。水平倾斜是指车牌的上、下边缘与水平方向产生了一个倾斜角度，可能是由于照片拍摄的角度等问题造成拍摄出的车牌照片在水平方向产生倾斜。要矫正这种倾斜，只需检测出车牌的上、下边缘与水平方向产生的倾斜夹角再消除这个夹角即可。竖直倾斜是指车牌的左、右两边缘或者车牌的字符与竖直方向产生了倾斜角度，这种倾斜可能会对之后的字符分割产生影响，若想消除这种竖直倾斜，只需要检测出左右两边缘的倾斜角度再进行校正即可。

本设计中主要用到了图像的仿射变换这一算法，图像可以看做是由横竖行的像素点组成，就可以通过建立坐标系，给每个像素点定一个坐标。设输入图像中，像素点的坐标为 (x, y) ；输出图像中，像素点的坐标为 (x^1, y^1) ，为了仿射变换，需要引入其次坐标，即用三维向量表示二维向量，用矩阵乘法表示变换。仿射变换可以统一为：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^1 \\ y^1 \\ 1 \end{bmatrix} \quad (3-7)$$

当矩阵中 $a_{11}, a_{12}, a_{21}, a_{22}$ 取不同的值，对应着不同的类型，包括平移、旋转、剪切以及缩放。

完成车牌部分的截取以及倾斜校正后，可得到合适的车牌部分，如图 3-10 所示。



图 3-10 定位之后得到的车牌

3.4 车牌字符分割

字符分割的任务是把多字符图像中的每个字符从整个图像中分割出来，使其成为单个字符。字符分割是一个承上启下的环节，因为这些字符可用于下一步的字符识别。在实际设计中，车牌的字符情况可能会有比较大的干扰或者噪声，可能会让字符变得模糊或者粘连，因此字符分割的算法影响了之后字符识别的准确度。

3.4.1 车牌号码组成

我国的车牌的轮廓长度均为 440mm，宽度为 140mm，共有 7 个字符，第一个字符为省份的简称，第二位为城市的字母代号，在第二个和第三个字符之间有一个空格，接着为五位英文字母或者阿拉伯数字。单个字符的同意宽度为 45mm，高度为 90mm，第二个字符和第三个字符之间的空格间距为 34mm，其余字符的间距为 12mm，字符的总长度为 409mm^[9]。车牌的字体均为规则的、统一印刷的字体，其长、宽以及长宽比都是按照严格标准制造的，给字符的分割与识别带来很大的方便，可以很精确地对每个字符进行切割，得到单个字符。

3.4.2 车牌字符分割算法

本设计采用投影法对车牌的字符进行分割，这是对每一列的像素进行统计而得出符合字符间空隙的一种方法。可直接利用投影法进行切分，得到投影图。根据投影图的明显峰谷特征，其中波峰对应字符区域，波谷则对应字符间的空隙。这样可确定每个字符的开始和结束为止，得到分割结果。

首先进行水平投影，将二值化的车牌图片水平投影到 Y 轴，得到连续投影最长的一段作为字符区域，因为车牌四周有白色的边缘，这里需要把水平方向上的连续白线过滤掉。这时可以得到车牌字符紧贴四周边缘的车牌图像，如图 3-11 所示。



图 3-11 水平投影之后的车牌

然后进行垂直投影，将二值化的车牌图片垂直投影到 X 轴，得到每个字符的宽度，分割后的字符宽度必须达到全局宽度才能算作一个字符，并且排除第二个和第三个字符之间的间隔。

字符分割完成之后，由于切割出来的字符可能有大有小，因此需要将每个字符都归一化成相同的 20×20 的尺寸，最终的结果如图 3-12 所示。



图 3-12 字符分割后的结果

3.5 车牌字符识别算法

在字符识别中，字形是模式的表现形式，字符识别的实质是自行识别。对于字符识别来说，识别过程是：从输入的待识别字符（模式样本）点阵图形（字形）中提取描述该字符的特征。车牌的字符由有限的汉字、英文字母和数字组成，本文采用卷积神经网络对车牌的字符进行识别。

卷积神经网络结构主要包括输入层、卷积层、池化层、全连接层以及输出层^[10]。

输入层是整个神经网络的输入，在处理图像的卷积神经网络中，它一般代表了一张图片的像素矩阵。这一层可以处理多维数据，在处理彩色图片时，三维矩阵的长和宽代表了图像的大小，而三维矩阵的深度代表图像的色彩通道。

卷积层是神经网络中最重要的部分。在这一层需要确定卷积核的大小、移动过的步长以及输出的特征参数的个数。

在训练期间，卷积层需要大量的内存，因为反向传播的方向传递需要在正向传递期间计算所有的中间值，因此我们需要用池化层，来缩小矩阵的大小的，达到降维的作用。

全连接层是在卷积神经网络的最后，给出最后的分类结果，在特征提取完成之后，仍然需要使用全连接层来完成分类任务。在本设计中，由于输出为 0~9，大写英文字母 A~Z 以及 31 个省份的中文简称共 65 个类别，故输出节点数即为 65 个。

训练时，选择 ReLU 激活函数，交叉熵损失函数计算损失值，以及 Adam 优化算法来进行训练。

在经过神经网络的多轮训练学习后，字符的特征都已经保存在了训练好的模型中，在运行系统时，只需要调用保存好的训练模型即可输出识别结果。

但是在训练过程中，有许多参数，包括卷积核大小、批处理尺寸、学习率、卷积层数、隐层节点数^[11]等需要经过多次训练实验，找寻最佳方案并对参数的改变对实验结果的影响进行对比，最后选择识别率恰当、训练速度较快的网络为最终的训练模型。

3.6 本章小结

车牌识别技术在现实生活中逐渐得以广泛应用，成为促进人类交通发展的重

要研究课题之一。本章主要阐述了车牌识别系统所需要的步骤以及每一个步骤所需要用到的详细算法。总体的算法步骤如图 3-13 所示。

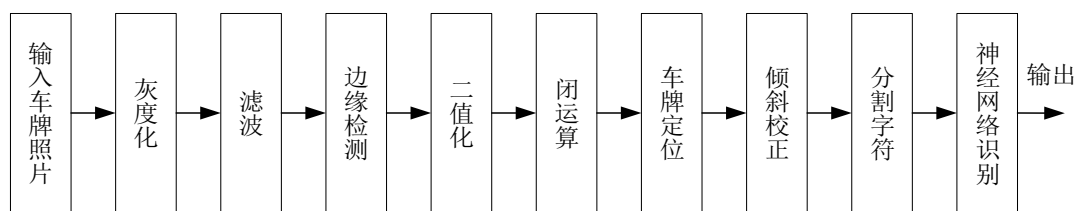


图 3-13 车牌识别系统整体框图

首先，在图像预处理方面，经过灰度化、中值滤波、Sobel 算子对车牌进行边缘检测、二值化、闭运算，找到车牌的大致位置已经，图像的预处理已经完成，接下来需要进行进一步的精确定位。然后需要进行车牌的定位，

车牌定位，首先选取窗口，采用形态学定位的方法，可以排除大部分车牌，但是少数车牌因为周围环境比较复杂，仅凭形态学定位不能全部排除，构建一个卷积神经网络，基本可以排除所有非车牌的区域，为了之后字符分割，需要对车牌进行倾斜校正。

接着，在截取出来并且已经倾斜校正的车牌图片上进行字符分割，本文使用的是投影法，先水平投影，截掉车牌的边缘部分，只剩下含有车牌字符的图像。再进行垂直投影，根据严格规定统一的字符的长宽以及车牌是由七个字符组成和一个空格间隙，根据垂直投影的宽度进行字符的分割，可得到七个字符的图片，输入到最后的字符识别网络中去。

最后，搭建卷积神经网络进行训练，并更改参数选择最佳模型，然后进行训练并保存，最后调用此模型，对车牌的字符进行识别并输出最后结果。

第4章 基于卷积神经网络的车牌字符识别系统设计

4.1 引言

目前车牌字符识别有许多方法，如：基于结构特征和灰度特征的方法、基于模板匹配和神经网络^[12]的方法、基于深度信念网络的方法，本文中选择使用基于卷积神经网络的方法进行训练，然后对所需要识别的车牌字符进行预测。

神经网络的输入层接受待识别样本的特征向量，经过各隐层神经元的运算，输出层的输出就是识别结果，输出层的每个神经元代表一类，代表了输出结果。

构建卷积神经网络，首先要搭建主要的结构，包括输入层、卷积层、池化层、全连接层以及输出层，搭建好结构之后，为了得到最优模型，需要对神经网络中的超参数进行微调，包括隐含层的个数、隐含层的神经元数、激活函数、学习率等，接着就要运行神经网络，其工作过程主要有两个阶段组成^[13]：

- （1）学习期，对神经网络的连接权值进行修改；
- （2）工作期，对于给定的输入进行计算，最终得到识别结果。

4.2 输入数据库

卷积神经网络算法需要大量的数据才能使其正常工作，因为网络需要先从这些数据集中提取特征参数，然后进行训练学习，才能有效地、精确地识别出所需要的内容。

因此，数据集需要有足够多地数据，让神经网络能够在训练过程中逐步提高准确率，但是需要指出的是，由于获得额外的训练数据并不总是一件轻而易举或者物美价廉的事情，所以中小型数据集依旧非常普遍；其次，数据集需要有多样化的特征，尽可能的包括更多的相关特征，较少的无关特征，系统才能够完成学习，这样会让数据集中的数据具有代表性，实现泛化，否则，即便样本数量非常大，也很难提高精确度；最后，数据集需要采用错误、异常值或者噪声较少的，这样会节省清理这些异常的、低质量的数据的时间^[14]。

基于以上数据集的特征及要求，本文的数据库含有 16035 张图片，共分为数字 0~9，大写英文字母 A~Z 以及 31 个省份的中文简称，共 61 个类别，每个类别的图片约为 240 张左右，图片的大小为 20×20，数据库的部分数据如图 4-1 所示。

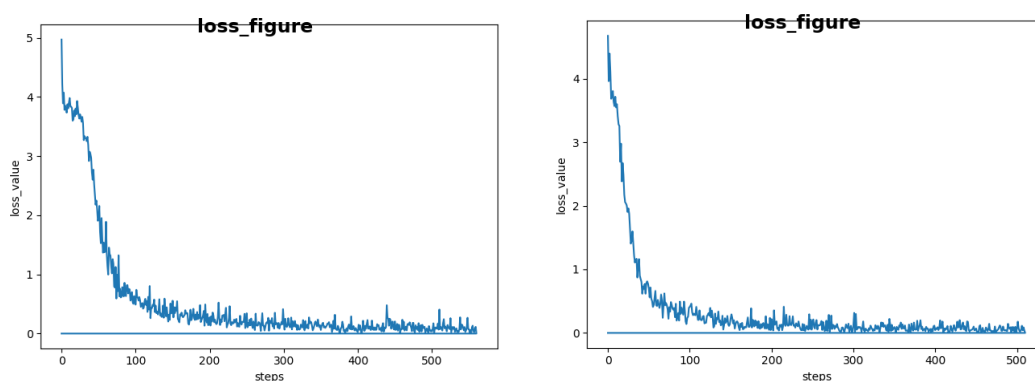


图 4-1 数据集中的部分数据

4.3 卷积神经网络结构的构建

输入数据库之后，数据库中的数据就要进入卷积网络进行训练。卷积神经网络主要包括卷积层、池化层和全连接层。本设计中神经网络的结构用 TensorFlow 库进行搭建。

首先应确定神经网络的层数，搭建了两层卷积和三层卷积的卷积神经网络^[15]进行对比，图 4-2 显示了在其他参数（学习率、批处理参数等）都相同的情况下，每一轮选出 80% 的数据进行训练，20% 进行测试，并设置训练轮数大于 500 轮且测试集精确率达到 0.98 即停止训练，得出两层网络和三层网络的误差函数曲线，此曲线表明了预测的结果和正确结果之间的误差。



a) 两层卷积的网络 loss 曲线

b) 三层卷积的网络 loss 曲线

图 4-2 三层神经网络和两层神经网络的比较

由图 4-2 曲线可以看出，三层卷积网络在训练大约 400 轮的时候，训练集 loss 的值逐渐趋于不变，开始收敛，最后的值为 0.08615，而两层卷积网络，在训练大约 200 轮时，就开始收敛，最后的 loss 值为 0.05223，由此可以看出，在此设计中，三层网络和两层网络均可在一定轮数内收敛，且都可以达到一定的准确率，而两层网络明显要比三层网络的训练速度要快，因此，选择设计两层卷积网络。卷积网络的结构如图 4-3 所示。

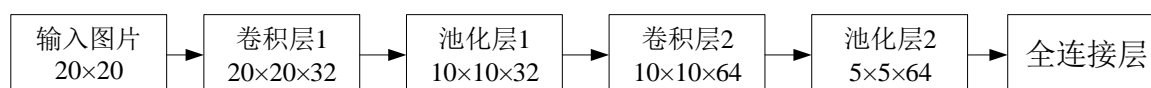


图 4-3 卷积网络结构图

首先搭建卷积层，选用 3×3 的卷积核，输出为 $20 \times 20 \times 32$ ，设置卷积核在各个维度移动的步长为 1，第一个卷积层结束后，对应 32 个输出。然后进入池化层，选用 2×2 的最大池化层，设置输入图像各个维度的池化尺寸，会输出为 $10 \times 10 \times 32$ 的图像。

有时当模型参数太多，训练样本又太少时，可能会出现过拟合现象^[16]，即在

训练数据上损失函数较少，预测准确率较高；但在测试数据上损失函数较大，预测准确率较低，为了避免这种现象，设置一个 Dropout 层，这个层的作用是在每个训练批次中，忽略一半的特征，可以明显地减少过拟合现象。

接着进入第二层卷积层和池化层，会输出 64 个 5×5 的图像。然后由于全连接层接收一维数据，所以需要将图像压平为 $5 \times 5 \times 64$ ，隐层节点数设为 1024，输出为数字 0~9，大写字母 A~Z 以及各省份的中文简称共 65 个类别。

4.4 卷积神经网络参数确定

基本的神经网络框架已经搭建完成，接下来就是进一步优化训练算法，不断调整网络中的各个参数，包括学习率、隐含层节点数、批处理参数以及卷积核的大小，以继续提高训练的正确率。

4.4.1 学习率

学习率（learning_rate）即每次参数更新的幅度^[17]，决定了在一个小批量中的权重在梯度方向要移动多远。学习率若很低，训练会变得更加可靠，但是优化会耗费较长的时间，因为朝向损失函数最小值的每个补偿很小；如果学习率很高，训练可能根本不会收敛，甚至会发散^[18]。权重的该变量可能非常大，使得优化越过最小值。因此，如何选择合适的学习率是一个非常重要的问题。

由于选用的是 Adam 优化算法，可以自适应地调节学习率，因此需要设定一个恰当的学习率初始值，应当从较大的学习率开始，因为在开始时，初始的随机权重远离最优值。因此就需要设置一个最合适的初始值，能够使损失函数最优，且不损失训练速度。设置初始学习率分别为 0.01，0.001，0.0001，损失函数图像分别如图 4-4，图 4-5，图 4-6 所示，训练结果的比较如表 4-1。

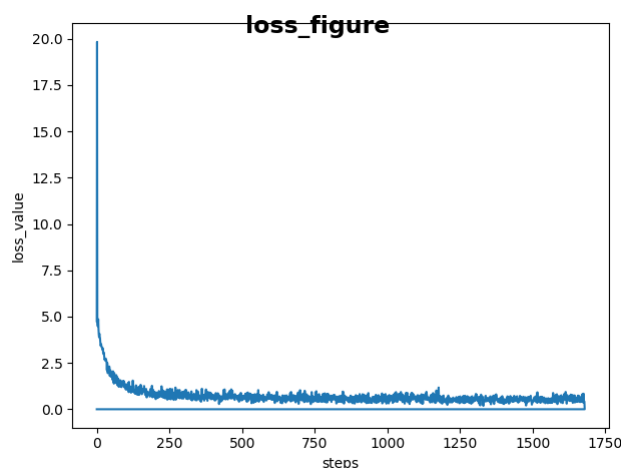


图 4-4 学习率为 0.01 时的损失函数

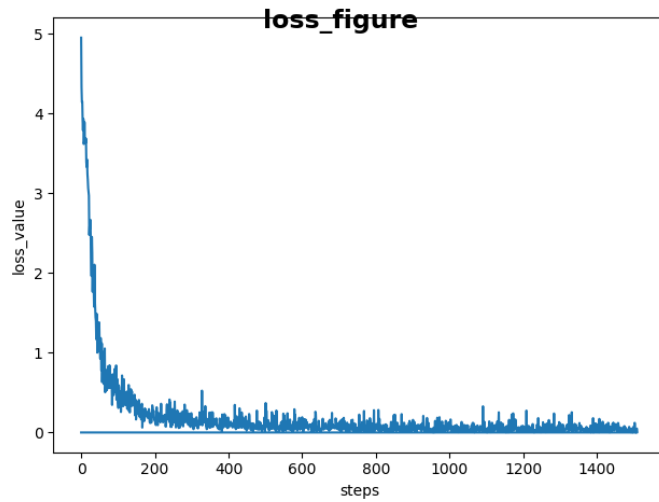


图 4-5 学习率为 0.001 时的损失函数

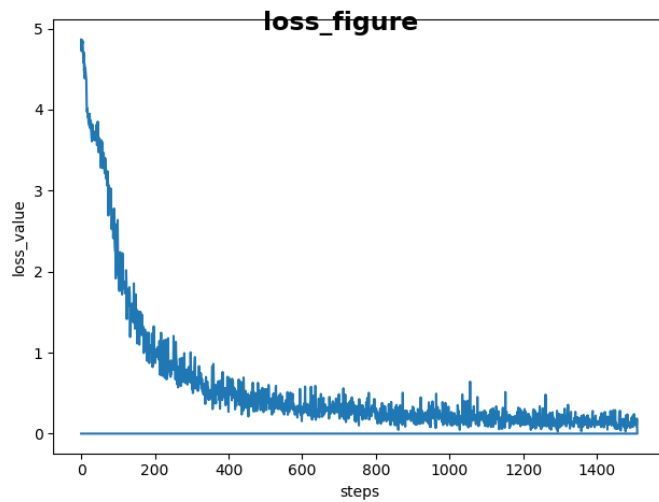


图 4-6 学习率为 0.0001 时的损失函数

表 4-1 改变学习率的训练结果

学习率	测试集准确率	损失函数	训练次数
0.01		不收敛	
0.001	99%	0.0493	1510
0.0001	99%	0.0877	1520

由以上实验可知，学习率的初始值会影响训练的效果，当学习率初始值为 0.01 时，损失函数 loss 值一直非常大，网络不收敛；当学习率初始值过小为 0.0001，虽然网络更加可靠，但是训练速度明显变慢。因此，学习率需要选择一个适中的 0.001 为最终的训练参数。

4.4.2 批处理参数

批处理参数（Batch size）为一次训练所选取的样本数，批处理参数的大小影响模型的优化程度和速度，同时也会直接影响到 GPU 内存的使用情况^[19]。如果不

设置批处理参数，意味着网络在训练时，需要一次性把所有的数据输入网络中，然后计算它们的梯度进行反向传播，若是一个大型的数据，若一次性把所有数据输入给网络，可能会引起内存的爆炸，因此就需要设置批处理参数来解决这一问题。

若设置了合适的批处理参数，可以通过并行化提高内存的利用率，尽量让 GPU 满载运行，提高训练速度，但是，每一次训练处理的数据少了，提取的特征也少了，因此如果需要达到相同的识别精度，需要更多轮的训练，并且可以让梯度下降的方向更加准确。本文依次设置批处理参数为 10、100、1000，其他参数均相同，分别得出训练损失函数如图 4-7、图 4-8、图 4-9 所示。训练的结果对比见表 4-2 所示。

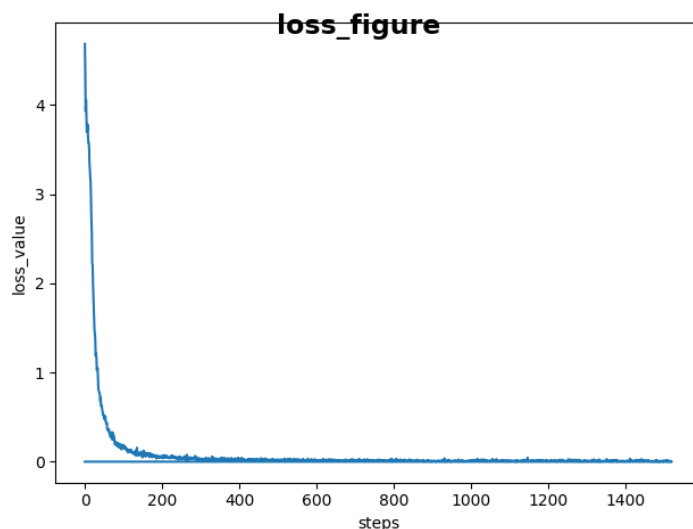


图 4-7 Batch Size 为 1000 时的损失函数

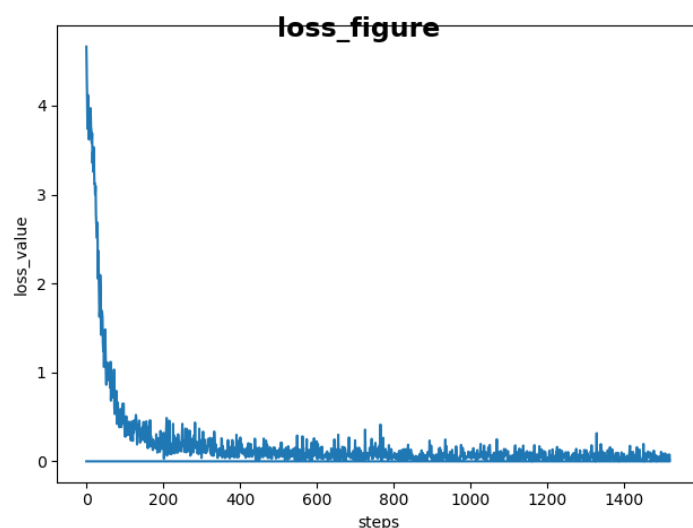


图 4-8 Batch Size 为 100 时的损失函数

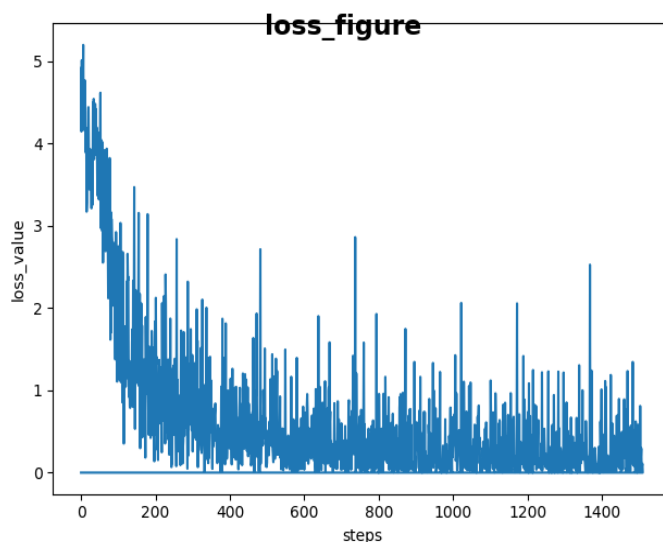


图 4-9 Batch Size 为 10 时的损失函数

表 4-2 改变批处理参数的训练结果

Batch Size	损失函数	训练时间
1000	0.3750	最长
100	0.0493	适中
10	0.1083	最短

由以上实验可知，当批处理参数很小的时候，比如设为 10 时，下降梯度不稳定，经常变来变去，网络性能很差；当批处理参数逐渐增大到一个合适的值时，梯度变得非常准确，训练的速度、精确度都能达到合适的要求；如批处理参数继续增大时，达到一定程度后，内存容量可能会增大，每一轮的迭代次数减少，要想达到相同的精度，就需要花费更多的时间，从而参数修正也就显得更加缓慢。因此，本设计选择一个适中的 100 为最终批处理参数。

4.4.3 隐层节点数

在设计全连接网络时，隐层在网络中起着极其重要的作用，隐层神经元的个数影响着网络是否能收敛、收敛速度的快慢、记忆能力及归纳和泛化能力^[19]。

一般情况下，隐含层神经元的个数是根据网络收敛性能的好坏来确定的，有经验公式：

$$s = \sqrt{0.43mn + 0.12n^2 + 2.54m + 0.77n + 0.51} \quad (4-1)$$

式中， m 为输入层神经元个数， n 为输出层神经元个数。

根据公式可计算出隐层节点数约为 40 个，但是此经验公式仅仅只能作为参考，具体在设计网络结构的时候，还需要设置不同的节点数，找出最合适的节点数。

当隐层节点数太少时，损失函数的值很大，不容易收敛；但隐含层节点数太多时，局部最小点越多，网络容易训练过度，导致网络不仅记住了学习样本一般特征而且记住了个别特征（如随机噪声）导致网络容错性降低。因此需要选择隐层节点数适中的网络。

设置节点数分别为 40, 100 和 200，观察损失函数图像以及分析训练结果，选出最佳的节点数，损失图像分别为图 4-10，图 4-11 和图 4-12。训练结果对比见表 4-3。

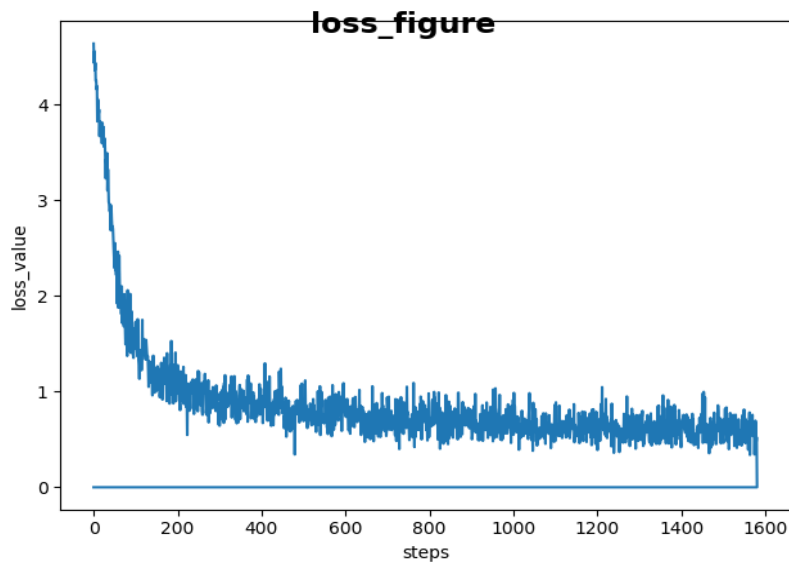


图 4-10 隐层节点数为 40 的损失函数图像

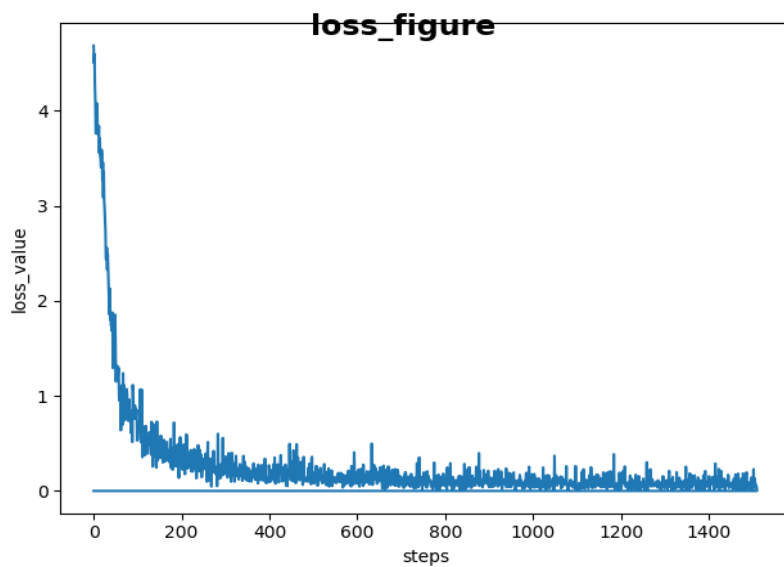


图 4-11 隐层节点数为 120 的损失函数图像

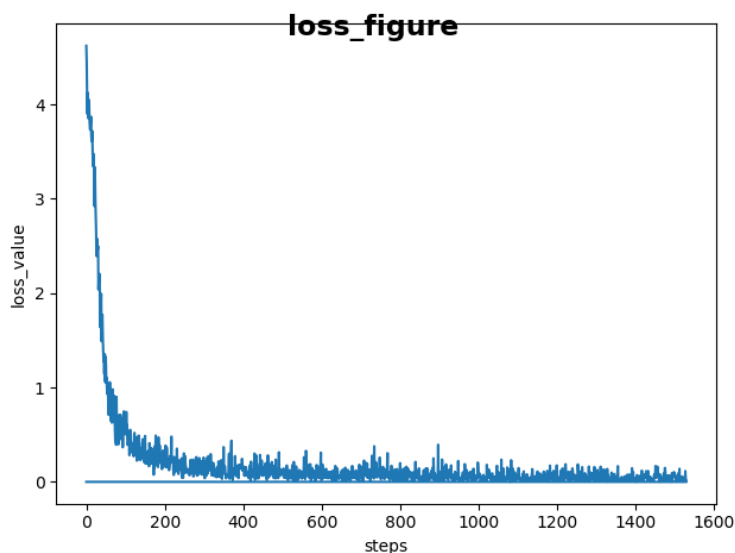


图 4-12 隐层节点数为 200 的损失函数图像

表 4-3 改变隐层节点数的训练结果

隐层节点数	损失函数	训练次数
40	0.5223	1580
120	0.0402	1510
200	0.0227	1530

由以上实验可知，当节点数为 40 时，损失函数很难收敛，应该增加节点数，当增加到 120 时，损失函数已经收敛，训练速度也比较快，当继续增加到 200 时，网络的损失函数值和训练次数与时间与 120 时相差不大，因此，为了避免过拟合现象发生，本设计选 120 为最终的隐层节点数。

4.5 交互界面设计

4.5.1 Tkinter 概述

Tkinter 模块是 Python 的标准 Tk GUI 工具包的接口，可实现本地窗口风格，并可以良好地运行在绝大多数平台中，

Tkinter 设计有一些主要控件，包括标签(Label)、按键(Button)、输入框(Entry)、文本(Text)、多选框(Checkbutton)、菜单栏(Menu)、画布(Canvas)等。在本设计中，主要用到标签(Label)、按键(Button)、输入框(Entry)、文本(Text)、画布(Canvas)这些控件。

4.5.2 交互界面设计

基于 Tkinter 的用户交互界面，使用本身具有的控件设计具有本地窗口风格的

界面，界面的风格简洁明了，清晰大方，可以让用户很快地能够使用，也比较完整地展示了图像的处理过程，感觉非常直观，并为了美观进行不断地修改，让页面的字体等更加地协调有序。

图 4-13 显示了交互界面的使用流程图。

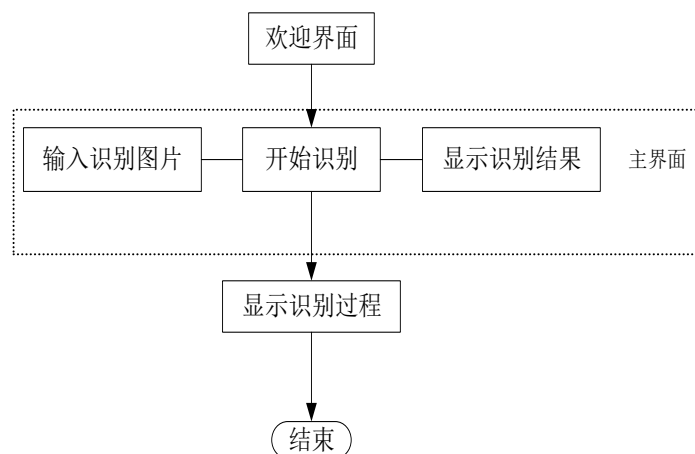


图 4-13 交互界面使用流程图

首先，点击运行，进入欢迎界面，将图片保存为.gif 格式，使其在画布上显示，显示设计的题目以及设计人和指导老师，在图片下面有“开始”按钮。启动界面如图 4-14 所示。



图 4-14 系统启动界面

接着，是系统的主界面，由一个输入框，“开始识别”按钮，“显示车牌图片”，“显示处理过程”三个按钮以及一个画布组成，如图 4-15 所示。点击车牌识别按钮后，会显示出车牌的字符以及原始车牌的照片。

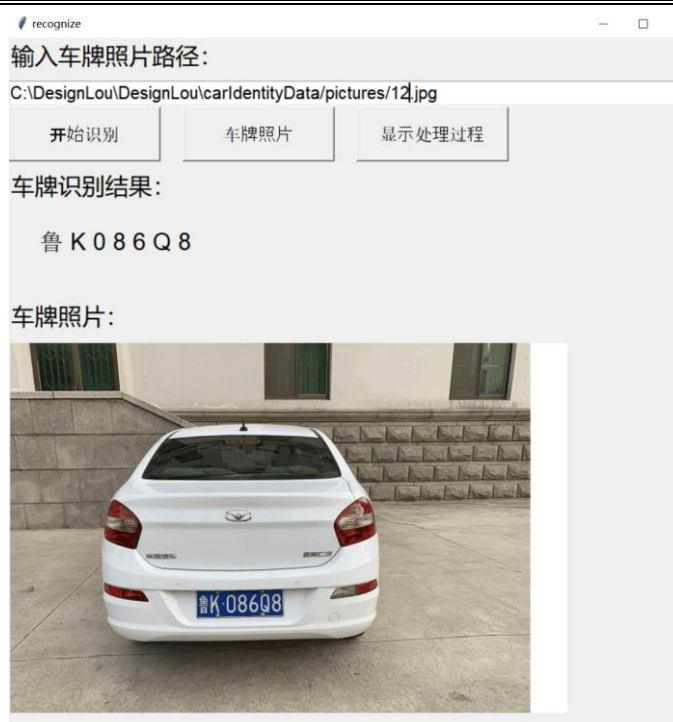


图 4-15 系统主界面

然后，设计了显示图像的处理过程的界面，如图 4-16 和图 4-17 所示，图 4-16 显示了预处理中灰度化、均值模糊、边缘检测的过程，并且和原图形成了对比。图 4-17 显示了二值化和闭运算的结果以及通过车牌定位选出的仅含有车牌照照片，最后可关闭界面结束程序。



图 4-16 图像处理过程



图 4-17 显示图像处理过程

首先，用户在图 4-15 的输入框中输入所需要识别的照片路径，输入框使用控件 Entry，即可获取图像的路径

点击“开始识别”按钮和“车牌照片”按钮，整个系统开始工作，会在相应的位置显示车牌的识别结果，在画布上显示原来车辆的照片。

点击“显示处理过程”按钮，即会显示灰度化、均值模糊、边缘检测的过程，并与车牌原图进行对比。

再点击“Next”，会打开第二个处理过程的显示窗口，会显示二值化和闭运算的结果以及选出的车牌照片。

最后点击关闭窗口，就会停止运行程序。

4.6 本章小结

对车牌字符的识别是系统中最核心也是最重要的一部分，本设计选用卷积神经网络算法对车牌字符进行识别。卷积神经网络的识别流程框图如图 4-18 所示。

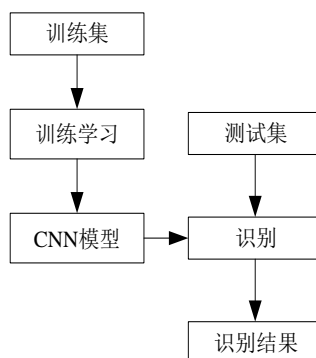


图 4-18 卷积神经网络的识别流程

首先，寻找合适的数据库，要注意数据的数量和质量，要有足够的数据量，并且要有一定的代表性，数据的质量会直接关系到网络的训练质量，会影响到最后的识别质量。

然后，利用 TensorFlow 库搭建初步的神经网络结构，并将两层网络和三层网络进行对比，考虑训练速度和训练精确度等问题，最后选择两层卷积神经网络的结构。

接着，调整神经网络中的其他参数，包括学习率，批处理参数以及全连接层的隐层节点数，比较训练集损失函数图像，测试集的精准确率以及训练速度等因素，注意数据选择过大或者过小对网络的影响，选择最优的模型，确定了卷积网络的模型：两层卷积，选择 3×3 的卷积核， 2×2 的池化层，全连接层的隐层节点数为 120，学习率初始值为 0.001，批处理参数为 100。

最后，设计交互界面，使系统更加完整。

结 论

本文比较详细地、系统地阐述了车牌识别系统的整体设计过程，也对各个过程所用到的具体算法进行了详细地分析与研究，结合当前科技的发展形势，将神经网络算法运用到车牌识别的系统中去，希望能达到更好的识别效果。得到如下结论：

（1）在图像预处理方面，经过灰度化、中值滤波、Sobel 算子对车牌进行边缘检测、二值化、闭运算，找到车牌的大致位置已经，图像的预处理已经完成，接下来需要进行进一步的精确定位。

（2）车牌定位，首先选取窗口，采用形态学定位的方法，可以排除大部分车牌，但是少数车牌因为周围环境比较复杂，仅凭形态学定位不能全部排除，构建一个卷积神经网络，基本可以排除所有非车牌的区域，为了之后字符分割，需要对车牌进行倾斜校正。

（3）车牌字符分割采用了投影法，先水平投影找出车牌中只含有字符的部分，再垂直投影，根据波峰波谷找出字符。

（4）最后是车牌的识别系统，由于输出分类包含了字母、数字以及中文字符共 65 类，因网络输出较多，故选择卷积神经网络进行训练，从构建神经网络的结构开始，对网络的各个参数，包括网络的层数，学习率，批处理参数以及隐层节点数进行调整，比较误差曲线与训练的速度、质量，选择最佳的方案。最后在学校附近采集了若干张车牌图片进行测试识别，均可以识别，但在用网络上的车辆照片验证时，发现有少数不够清晰的照片、或者倾斜角度太大的照片无法识别，故识别的准确率可达到 91%。

（5）虽然本设计系统达到了一定的识别率，但是测试的车牌比较少，而且仅对小型车的车牌进行了测试，因此实验需要进行，测试的准确率有待进一步验证。

本文所使用的神经网络、图像处理与识别的内容是当今发展迅速的学科，是研究的热点，是以后常用、有价值的方法以及算法，需要更深入的学习以及研究才能提高识别率，作出真正实用的系统。

参考文献

- [1] 章为川. 基于神经网络的车牌识别系统的研究与设计. 西南交通大学, 2006-05-01.
- [2] 李罡. 基于人工神经网络的不同颜色车牌识别系统的研究. 北京工业大学, 2009-06-01.
- [3] 刘丰. 机器学习和卷积神经网络在车牌识别的研究和应用. 华南理工大学, 2017-06-05.
- [4] 杨观赐, 杨静, 李少波, 胡建军. 基于 Dropout 与 ADAM 优化器的改进 CNN 算法. 华中科技大学学报, 2018,46 (7): 122-127
- [5] 陈炳权, 刘宏立, 孟凡斌. 数字图像处理技术的现状及其发展方向[J]. 吉首大学学报 (自然科学版), 2009, 30(1):63-70.
- [6] 林明旺. 基于卷积神经网络的图像分类研究. 湖南科技大学. 2017-12-01.
- [7] 李文举. 基于边缘颜色的车牌定位新方法[J]. 计算机学报, 2004, 27(2):204-208.
- [8] 章品正, 王健弘. 一种应用机器学习的车牌定位方法[J]. 应用科学学报, 29(2):147-152.
- [9] 徐雪丽, 乌旭东, 白媛. 车牌定位及字符切分算法的研究与 MATLAB 仿真. 延安大学学报, 2018,37(3):46-50
- [10] 董峻妃. 基于卷积神经网络的车牌区域检测和车牌字符识别研究. 西华师范大学, 2018-04-01
- [11] Jason Axelrod. The rise of automatic license plate recognition[J]. The American City & County, 2019.
- [12] 王敏, 黄心汉, 魏武, 李炜. 一种模板匹配和神经网络的车牌字符识别方法[J]. 华中科技大学学报, 2001, 29(3):48-50.
- [13] 董峻妃, 郑伯川, 杨泽静. 基于卷积神经网络的车牌字符识别[J]. 计算机应用, 2017,37 (7): 20014-2018
- [14] 苏延平. 基于图像处理的车牌识别技术的研究与实现[D]. 长春: 吉林大学, 2016:29-31.
- [15] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.
- [16] 姜桂艳, 常安德, 牛世峰. 基于车牌识别数据的交通拥堵识别方法[J]. 哈尔滨工业大学学报, 2011,43(4):131-135.
- [17] Nakama T. Theoretical analysis of batch and on-line training for gradient descent learning in neural networks[J]. Neurocomputing 2009, 73(1-3):151-159.

- [18] 朱宏吉. 基于深度神经网络的车牌字符识别算法研究与实现[D], 杭州: 浙江大学数字技术及仪器研究所, 2015:1-77.
- [19] He K, Zhang X, Ren S, Spatial pyramid pooling in deep convolutional networks for visual networks for visual recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9): 1904-1916.
- [20] Charl Coetzee,. PC Based Number Plate Recognition Systems. IEEE International Symposium on Industrial Electronics. 1998: 605-610

致 谢

时光荏苒，四年的大学生活即将结束，四年中我收获了知识、收获了友情、收获了成长。

在此，首先要感谢我的导师张虹老师，您在生活上无微不至的关怀我，在学业上孜孜不倦的教诲我，您渊博的知识和严谨的作风也无时无刻不感染着我，在您的帮助下我收获了许多，成长了许多。感谢您在论文撰写的过程中，您为我提供了技术上的支持、方法上的指导、精神上的鞭策，为我论文的顺利完成给予的极大帮助，您辛苦了。同时，感谢学院的各位老师，在本人论文写作过程中给予的关怀、指导和帮助，也要感谢在大学期间为我授课的各位老师，让我丰富了专业相关领域的知识，在此由衷的说一声：谢谢老师！

感谢我的室友、我的同学、我的朋友，我们的友谊凝固在这朝夕相处的日日夜夜，感谢你们对我的帮助，协助我完成了论文。

感谢即将对我的论文进行批评指正的老师，你们的意见对我完善专业知识将起到无可替代的帮助。谢谢各位老师！