

武汉理工大学毕业设计（论文）

基于 CNN 和 CUDA 的车牌识别算法优化与实现

学院（系）： 计算机科学与技术学院

专业班级： 软件工程 SY1301 班

学生姓名： 康康

指导教师： 袁景凌

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密口，在 年解密后适用本授权书

2、不保密口。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘 要

车牌识别系统(VLPR) 是计算机视频图像识别技术在车辆牌照识别中的一种应用,在商业和公共领域应用广泛。由于条件和车牌类型的差异, 车牌检测和识别仍然是一个具有挑战性的问题。

VLPR 系统一般包括三个模块: 车牌定位、字符分割和字符识别。本文首先从基本的人工神经网络开始, 介绍了卷积神经网络(CNN)的基础知识。然后分三个章节分别介绍了这三项技术。

(1) 在车牌定位中, 本文总结了常见的六种类型的算法: 基于边缘检测、彩色分割、小波变换、遗传算法、数学形态学、灰度图像纹理特征分析的车牌定位算法, 随后介绍了MSER 算法。

(2) 在字符分割中, 本文首先介绍了车牌图像的预处理操作, 包括倾斜扭正和旋转缩放操作。然后介绍了中国标准车牌的先验知识用来对字符进行切割, 进而介绍了切割方法和过程。

(3) 在字符识别中, 本文提出了一个共隐层的 CNN 网络模型来解决中文字符和字母数字特征间存在的差异, 并缓解因中文训练集过少带来的过拟合现象。最后分析了实验结果, 该模型比单独用两个 CNN 分别识别中文和字母数字在测试集上的相对错误率分别降低了 14.622%和 14.423%。

由于本文采用的算法大多适合并行计算, 本文对 CUDA C 编程环境进行了介绍, 最后分别介绍了如何运用 CUDA C 对算法的训练和预测过程进行并行优化。

关键字: 车牌识别 卷积神经网络 共享隐层 CUDA

Abstract

License plate recognition system (VLPR) is an application of computer video image recognition technology in vehicle license plate recognition, which is widely used in commercial and public fields. Licensing and identification are still a challenging issue due to differences in conditions and license plate types.

VLPR system generally includes three modules: license plate positioning, character segmentation and character recognition. This paper begins with the basic artificial neural network, and introduces the basic knowledge of convolution neural network (CNN). And then divided into three chapters were introduced in the three technologies.

(1) In the license plate location, this paper summarizes the six common types of algorithms: based on edge detection, color segmentation, wavelet transform, genetic algorithm, mathematical morphology, gray image texture feature analysis of the license plate positioning algorithm, MSER algorithm.

(2) In the character segmentation, this paper first introduces the pretreatment operation of the license plate image, including the tilt twist and the rotation scaling operation. And then introduced the Chinese standard license plate a priori knowledge used to cut the characters, and then introduced the cutting method and process.

(3) In character recognition, this paper proposes a CNN network model of co-hidden layer to solve the differences between Chinese characters and alphanumeric features, and alleviate the over-fitting phenomenon caused by too few Chinese training sets. Finally, the experimental results are analyzed, which reduces the relative error rate of Chinese and alphanumeric characters in the test set by 14.622% and 14.423% respectively with two CNNs.

As the algorithm used in this paper is mostly suitable for parallel computing, this paper introduces the CUDA C programming environment, and finally introduces how to use CUDA C to optimize the training and prediction process of the algorithm.

Keyword: VLPR CNN share-hidden-layer CUDA

目 录

摘 要	I
Abstract	II
目 录	III
第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 研究路线及工作内容	2
1.4 论文主要结构	3
第 2 章 卷积神经网络基础知识	5
2.1 深度学习背景	5
2.2 人工神经网络	6
2.3 梯度下降和反向传播	8
2.3.1 梯度下降算法简介	8
2.3.2 反向传播算法	8
2.4 卷积神经网络	10
2.4.1 卷积神经网络结构	10
2.4.2 常用 CNN 网络介绍	11
2.5 本章小结	11
第 3 章 车牌定位算法	12
3.1 常见的车牌定位算法	12
3.1.1 基于边缘检测的车牌定位算法	12
3.1.2 基于彩色分割的车牌定位算法	12
3.1.3 基于小波变换的车牌定位方法	13
3.1.4 基于遗传算法的车牌定位	13
3.1.5 基于数学形态学的车牌定位方法	14
3.1.6 基于灰度图像纹理特征分析的车牌定位方法	14
3.2 MSER 算法	15
3.3 本章小结	16
第 4 章 字符切割算法	17

4.1 切割预处理	17
4.1.1 偏斜扭正	17
4.1.2 旋转操作和调整大小	17
4.2 字符切割技术	17
4.2.2 中文车牌切割先验知识介绍	17
4.2.2 字符切割过程	18
4.3 本章小结	19
第 5 章 基于共隐层 CNN 的字符识别算法	20
5.1 模型网络结构	20
5.2 实验结果与分析	21
5.3 本章小结	22
第 6 章 CUDA 编程介绍	22
6.1 CUDA 背景	22
6.2 CUDA 编程简介	22
6.2.1 Hello, World!	22
6.2.2 核函数调用	23
6.2.2 参数传递	24
6.3 CUDA 在算法中的应用	25
6.3.1 CUDA 在训练中的应用	25
6.3.2 CUDA 在预测中的应用	26
6.4 本章小结	26
第 7 章 总结与展望	26
7.1 论文总结	26
7.2 工作展望	26
致谢	27
参考文献	27

第 1 章 绪论

1.1 研究背景及意义

车牌识别系统(Vehicle License Plate Recognition, VLPR) 是计算机视频图像识别技术在车辆牌照识别中的一种应用。

车牌识别技术要求能够将运动中的汽车牌照从复杂背景中提取并识别出来, 通过车牌提取、图像预处理、特征提取、车牌字符识别等技术, 识别车辆牌号、颜色等信息, 目前最新的技术水平为字母和数字的识别率可达到 99.7%, 汉字的识别率可达到 99%。车牌识别在高速公路车辆管理中得到广泛应用, 电子收费(ETC)系统中, 也是结合 DSRC 技术识别车辆身份的主要手段。

车牌识别技术结合电子不停车收费系统(ETC)识别车辆, 过往车辆通过道口时无须停车, 即能够实现车辆身份自动识别、自动收费。在车场管理中, 为提高出入口车辆通行效率, 车牌识别针对无需收停车费的车辆(如月卡车、内部免费通行车辆), 建设无人值守的快速通道, 免取卡、不停车的出入体验, 正改变出入停车场的管理模式。

1.2 国内外研究现状

车牌检测与识别技术作为一个集商业与公共于一身的项目, 无论是学术界还是商业界对它的重视程度相当高。从运输和通行违规到事故监测, 自动检测和车牌识别的能力是世界各地执法机构的关键工具之一。由于条件和车牌类型的差异, 车牌检测和识别仍然是一个具有挑战性的问题。大多数现有解决方案本质上是限制性的, 即对于特定类型的车牌模板, 以特定分辨率为特定视角进行固定照相机的工作。这意味着, 例如, 为静态摄像机设计的车牌系统在移动摄像机表现不佳。另外, 由于处理速度的瓶颈, 大多数技术严重依赖启发式方法。

随着深度 CNN 的出现^[1]和增强^[2], 以及更快的处理硬件的出现, 我们需要重新审视这个问题。

1.3 研究路线及工作内容

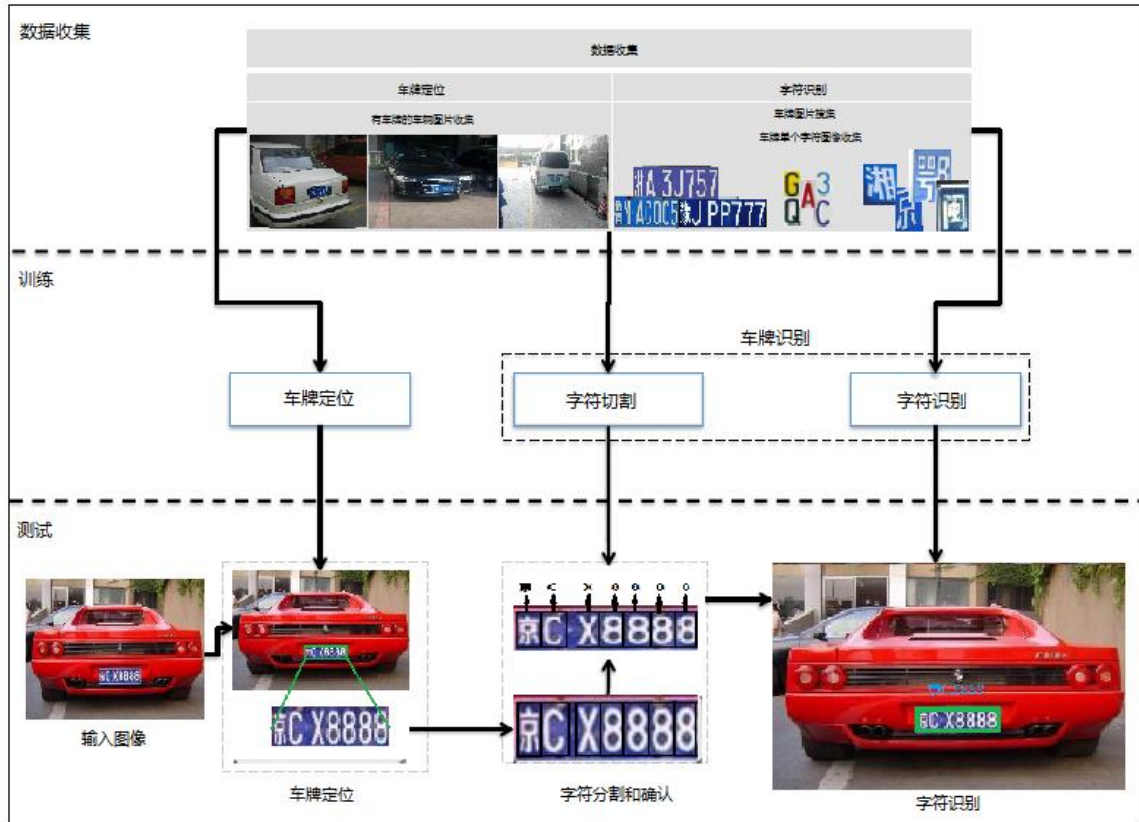


图 1.1 研究路线

(1) 图 1.1 展示了本文的研究路线，研究路线主要分为三个阶段：收集数据、训练网络及测试。首先，将收集的数据进行清理、组织和适当地注释（以半自动的方式），然后进行 CNN 的训练。每个阶段分为三个模块：车牌检测、字符切割及字符识别。1) 车牌检测器用于发现和裁剪车牌;2) 字符分割用于分离每个字符;3) 使用字符检测模块去掉非字符元素;4) 通过识别系统识别每个剩余字符，以获得正牌号码。

(2) 本文主要工作内容包含以下几点：

- ① 车牌检测部分采用 Maximally Stable Extremal Regions (MSER) 算法^[3]。车牌检测是车牌识别的最重要部分，其质量直接影响车牌识别的结果。在自然光线下，车牌定位需要克服不均匀的照明、车牌倾斜角度、复杂的背景条件等。本文中，车牌定位使用的方法主要是基于 MSER 算法和多重分类。
- ② 根据拥有的中国车牌分割识别的知识，本文提出了一套关于车牌分割的算法，即采用垂直投影方法进行字符分割。
- ③ 在字符识别中，卷积神经网络 (CNN) 的性能优于其他大多数模型。然而，为了保证

令人满意的表现，CNN 通常需要大量的样本。由于中文和字母数字的差异，识别两类的最常用方法是分别使用两个独立的 CNN。在本文中，为了解决汉字短缺的问题，我们实现了一个 CNN 模型，分别为汉语和字母数字字符共享隐藏层和两个不同的 softmax 层。

1.4 论文主要结构

第一章 绪论。介绍了论文研究背景、意义及国内外研究现状，综合阐述了论文的主要工作和组织结构。

第二章 卷积神经网络基础知识。本章从基本的 BP 神经网络开始，进而介绍了卷积神经网络以及梯度下降算法。

第三章 车牌定位算法。本章从常用的车牌定位算法出发，进一步介绍了本文中要使用的 MSER 算法。

第四章 字符切割算法。本章从字符切割所需要的预处理算法出发，首先介绍了中国标准牌照的先验知识，进一步根据这些知识提出了适用于中国的字符切割算法。

第五章 基于共隐层 CNN 的字符识别算法。本章介绍了共隐层 CNN 的网络结构和实验结果分析。

第六章 CUDA 编程介绍。本章从 CUDA C 的背景和并行计算发展的历史出发，用几个示例程序介绍了 CUDA C 的编程特点。从而进一步介绍了在车牌识别的一系列过程中，那些地方可以利用 CUDA C 进行优化加速。

第七章 总结与展望。

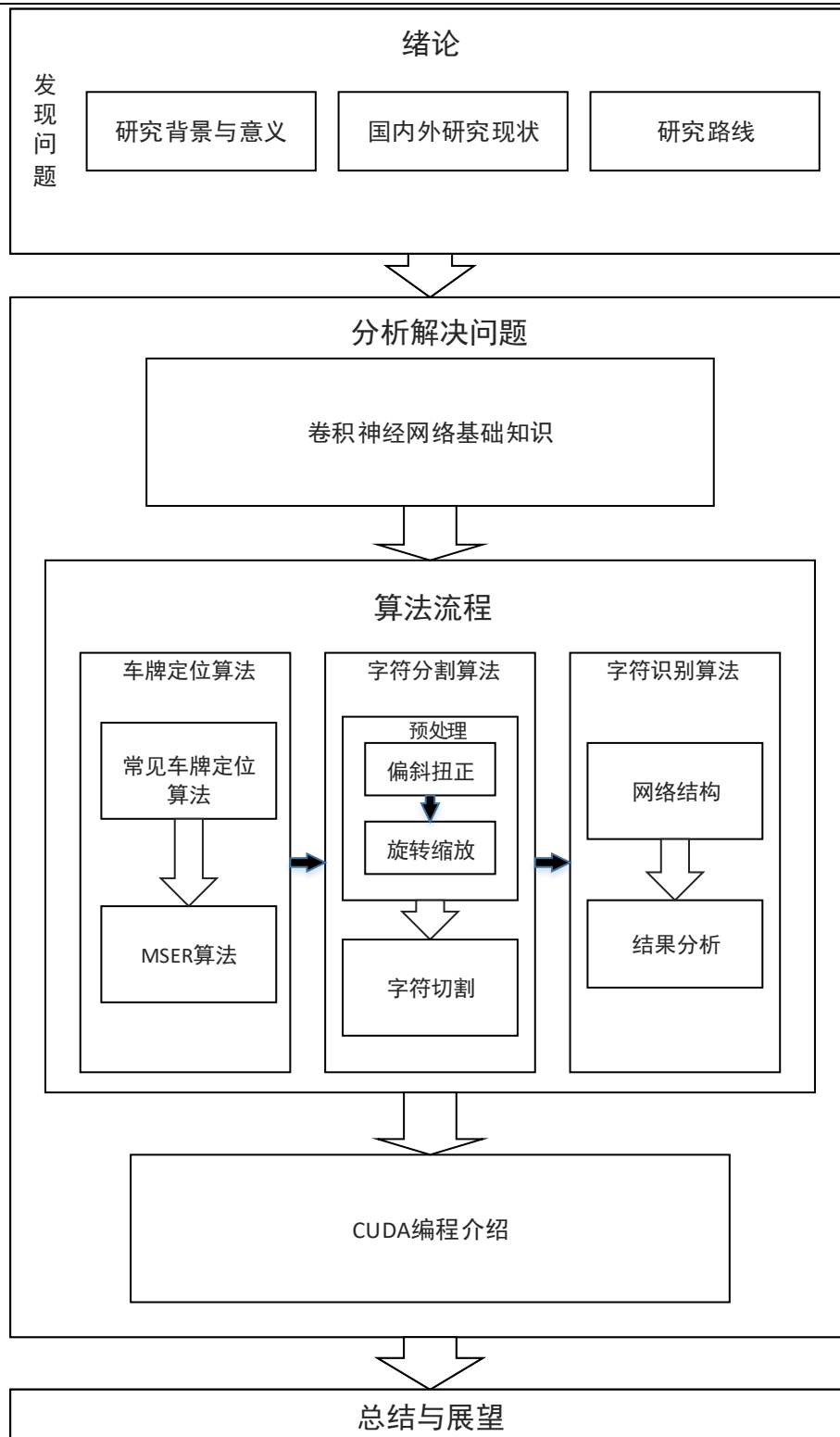


图 1.2 论文主要组织结构

第 2 章 卷积神经网络基础知识

2.1 深度学习背景

其实 CNN 并不是近几年才出现的。早在 1985 年，Rumelhart 和 Hinton 等人就提出了反向传播（Back Propagation，BP）算法^[4]，使得神经网络的训练变得简单可行。

几年后，LeCun 利用 BP 算法来训练多层神经网络用于识别手写邮政编码^[5]，这个工作就是 CNN 的开山之作，如图 2.1 所示，多处用到了 5*5 的卷积核，但在这篇文章中 LeCun 只是说把 5*5 的相邻区域作为感受区域，并未提及卷积或卷积神经网络。

1998 年的 LeNet5^[6]标注着 CNN 的真正面世，但是这个模型在后来的一段时间并未能得到学术界的重视，主要原因是训练的计算量巨大，而且其他算法（如 SVM）也能达到类似的效果甚至超过。随着计算机计算能力的提升尤其是 GPU 并行计算的出现和流行，使得更深的网络成为可能，使深层的神经网络的学习能力得以彰显出来。

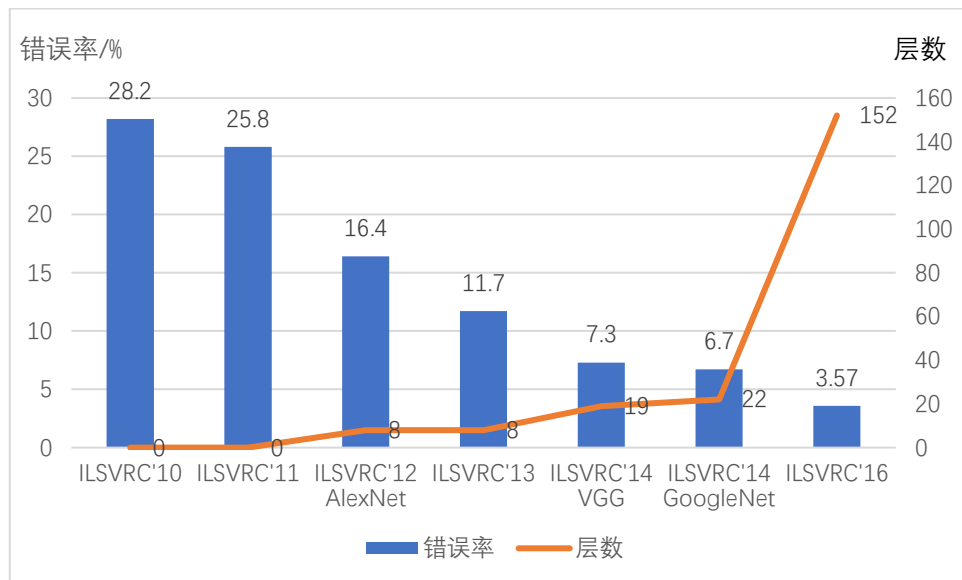


图 2.1 历年 ILSVRC 的 Top5 错误率和网络层数变化

ImageNet 大规模视觉识别大赛(ImageNet Large Scale Visual Recognition Competition, ILSVRC)是全球最为权威的计算机视觉比赛，该赛事的目标是对 1000 类图像进行分类，比赛的训练数据集数据达 126 万张图像，验证集 5 万张，测试集有 10 万张。2012 年，Hinton 的学生 Alex Krizhevsky 一举摘下了 ILSVRC 2012 冠军，效果大幅度超过传统的 SVM 方法，使 TOP5 的错误率首次降到 20% 以下。这个深度学习模型就是后来大名鼎鼎的 AlexNet 模型。从此，深度学习一发不可收拾，ILSVRC 每年都不断被深度学习也就是 CNN 刷榜，如图 2.1 所示，随着模型变得越来越深，Top-5 的错误率也越来越低，2015 年降到了 3.5% 附近，而在同样的 ImageNet 数据集上，人眼对其分类的辨识错误率大概在 5.1%，也就是目前的 CNN 对图像分类的能力已经超过了人眼。

2.2 人工神经网络

人工神经网络由多个神经元模型构成（如图 2.2），这种由许多神经元组成的网络具有可并行计算的分布结构。每个神经元具有单一输出，并且能够与其它神经元连接。每条连线上都有一个权重，代表上一个神经元对下一个神经元的影响程度。令来自其它处理单元(神经元) i 的信息为 x_i ，它们与本处理单元的互相作用强度为 w_i ， $i = 0, 1, \dots, n - 1$ 。那么本神经元的输入为 $\sum_{i=0}^{n-1} w_i x_i$ ，而神经元的输出为： $y = f(\sum_{i=0}^{n-1} w_i x_i)$ 。

式中， x_i 为第 i 个元素的输入， w_i 为第 i 个元素与本处理单元的互联权重。 f 称为激发函数(activation function)。它决定节点(神经元)的输出，提供了神经网络的非线性特性。

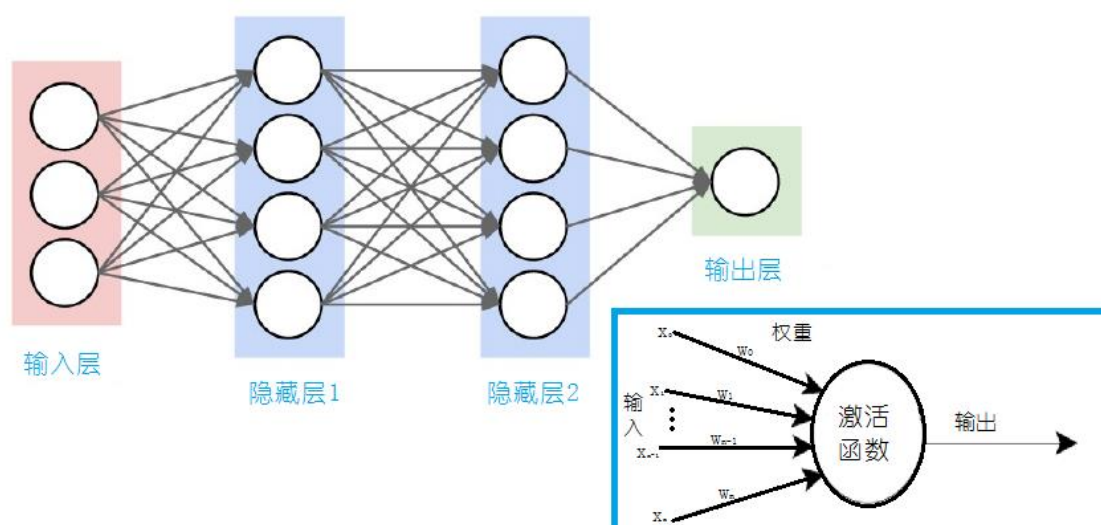


图 2.2 人工神经网络结构示意图

常用的激活函数有阈值函数、sigmoid 函数、ReLU（改进版 Leaky ReLU）、双曲正切函数，这些函数都有两个共同的特点：（1）其函数值都在某个 x_0 处发生急剧的变化；（2）其导数都能用函数值容易的表示出来。

（1） sigmoid 函数

$$y = \frac{1}{1 + e^{-x}}, y' = \frac{-e^{-x}}{(1 + e^{-x})^2} = \frac{1}{1 + e^{-x}} \cdot \left(1 - \frac{1}{1 + e^{-x}}\right) = y(1 - y) \quad (1)$$

（2） 阈值函数

$$y = \begin{cases} 1, & x \geq \theta \\ 0, & \text{otherwise} \end{cases}, y' = 0 \quad (2)$$

（3） ReLU

$$y = \begin{cases} x, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}, y' = \begin{cases} 1, & x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

(4) Leaky ReLU

$$y = \begin{cases} x, & x \geq 0 \\ \alpha x, & \text{otherwise} \end{cases}, y' = \begin{cases} 1, & x \geq 0 \\ \alpha, & \text{otherwise} \end{cases} \quad (4)$$

(5) 双曲正切函数

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}, y' = \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2} = 1 - y^2 \quad (5)$$

图 2.3 是上述激励函数的图像，我们可以发现，这些图像都在某个值附近发生了突然性的变化。

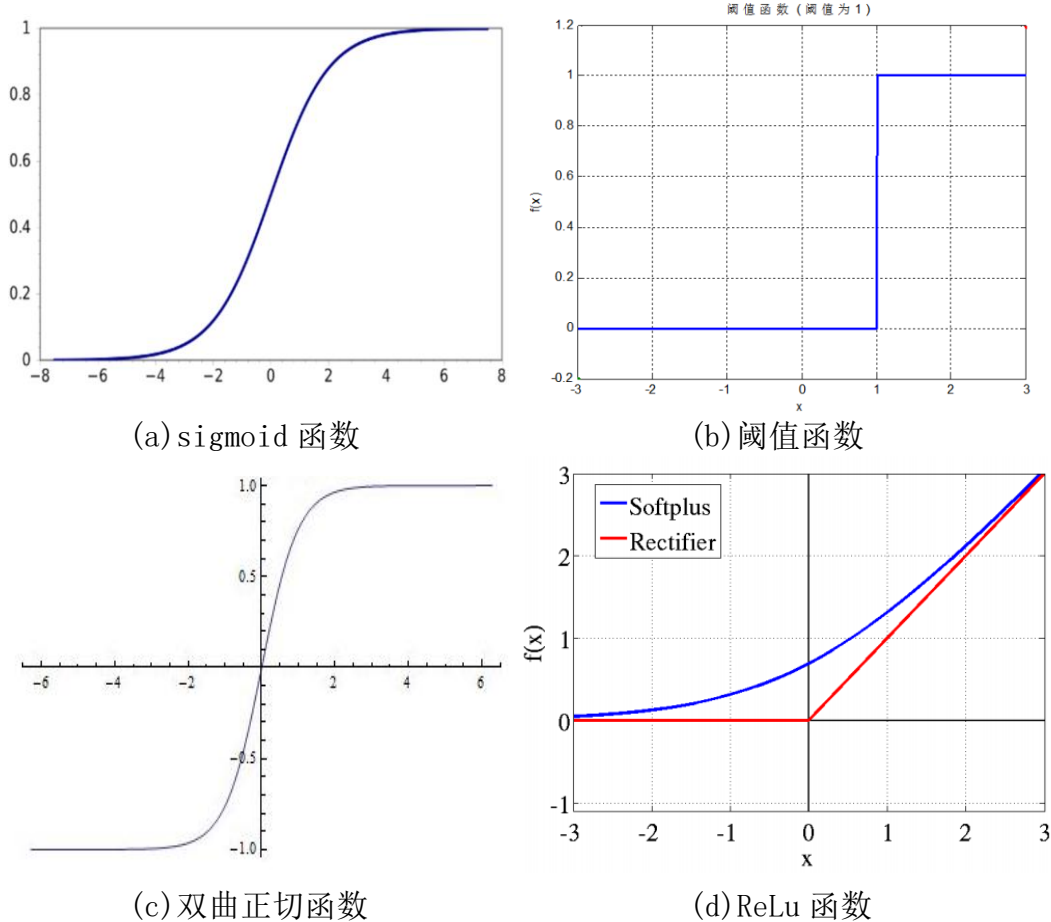


图 2.3 常见激励函数

2.3 梯度下降和反向传播

2.3.1 梯度下降算法简介

顾名思义，梯度下降法的计算过程就是沿梯度下降的方向求解极小值（也可以沿梯度上升方向求解极大值）。

其迭代公式为 $a_{k+1} = a_k + \rho_k \bar{s}^{(k)}$ ，其中 $\bar{s}^{(k)}$ 代表梯度负方向， ρ_k 表示梯度方向上的搜索步长。梯度方向我们可以通过对函数求导得到，步长的确定比较麻烦，太大了的话可能会发散，太小收敛速度又太慢。一般确定步长的方法是由线性搜索算法来确定，即把下一个点的坐标看做是 a_{k+1} 的函数，然后求满足 $f(a_{k+1})$ 的最小值的即可。

因为一般情况下，梯度向量为 0 的话说明是到了一个极值点，此时梯度的幅值也为 0。而采用梯度下降算法进行最优化求解时，算法迭代的终止条件是梯度向量的幅值接近 0 即可，可以设置个非常小的常数阈值。

2.3.2 反向传播算法

人工神经网络为学习实数值和向量值函数提供了一种实际可行的方案，连续的和离散的属性都可以使用，并且对训练数据中的噪声具有很好的健壮性。反向传播算法是网络学习算法中最常见的算法。反向传播算法是一种具有很强学习能力的系统，结构比较简单，且易于编程。

鲁梅尔哈特(Rumelhart)和麦克莱兰(Meclelland)于 1985 年发明并发展了 BP 网络学习算法，实现了明斯基的多层网络设想。BP 网络不仅含有输入节点和输出节点，而且含有一层或多层隐(层)节点。输入信号先向前传递到隐藏节点，经过作用后，再把隐藏节点的输出信息传递到输出节点，最后给出输出结果。节点的激发函数选用 Sigmoid 函数。

反向传播(back-propagation, BP)算法是一种估计单个权重变化引起整个网络性能变化值的较为简单的方法。由于 BP 算法过程包含从输出节点开始，反向地向第一隐藏层(即最接近输入层的隐藏层)传播由总误差引起的权重修正，所以称为"反向传播"。反向传播特性与所求解问题的性质和所作细节选择有极为密切的关系。

对于由一系列确定的单元互连形成的多层网络，反向传播算法可用来学习这个多层网络的权值。它采用梯度下降方法试图最小化网络输出值和目标值之间的误差平方，因为我们要考虑多个输出单元的网络，而不是像以前只考虑单个单元，所以我们要重新计算误差 E，以便对所有网络输出的误差求和：

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2 \quad (6)$$

其中， $Outputs$ 是网络输出单元的集合， t_{kd} 和 o_{kd} 是与训练样例 d 和第 k 个输出单元的相关输出值。

反向传播算法的一个迷人特性是：它能够在网络内部的隐藏层发现有用的中间表示：

1.训练样例仅包含网络输入和输出，权值调节的过程可以自由地设置权值，来定义任何隐藏单元表示，这些隐藏单元表示在使误差 E 达到最小时最有效。

2.引导反向传播算法定义新的隐藏层特征，这些特征在输入中没有明确表示出来，但能捕捉输入实例中与学习目标函数最相关的特征

反向传播训练神经元的算法如下：

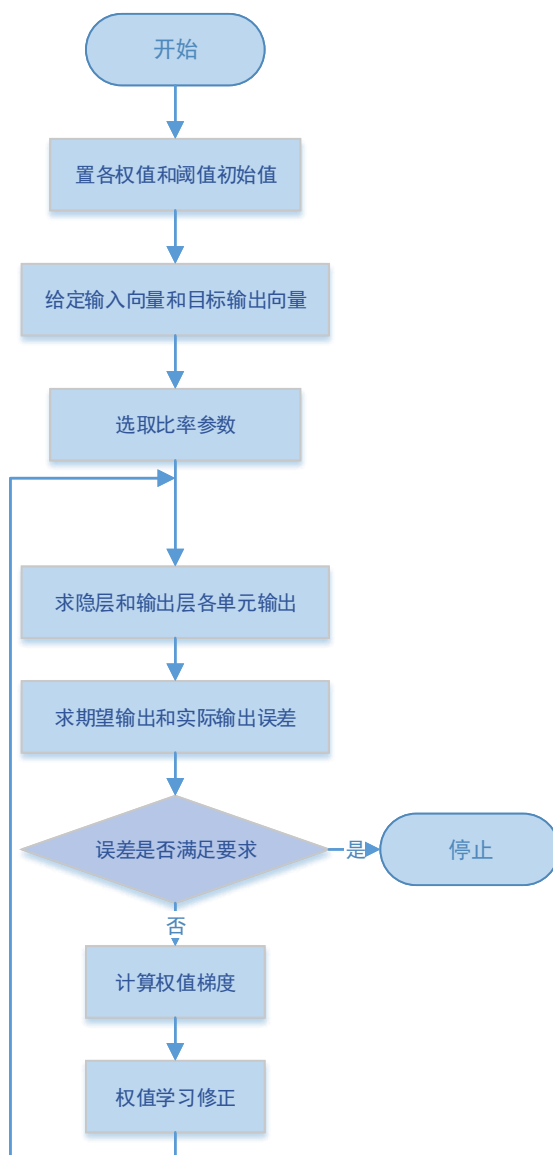


图 2.4 反向传播算法流程图

2.4 卷积神经网络

2.4.1 卷积神经网络结构

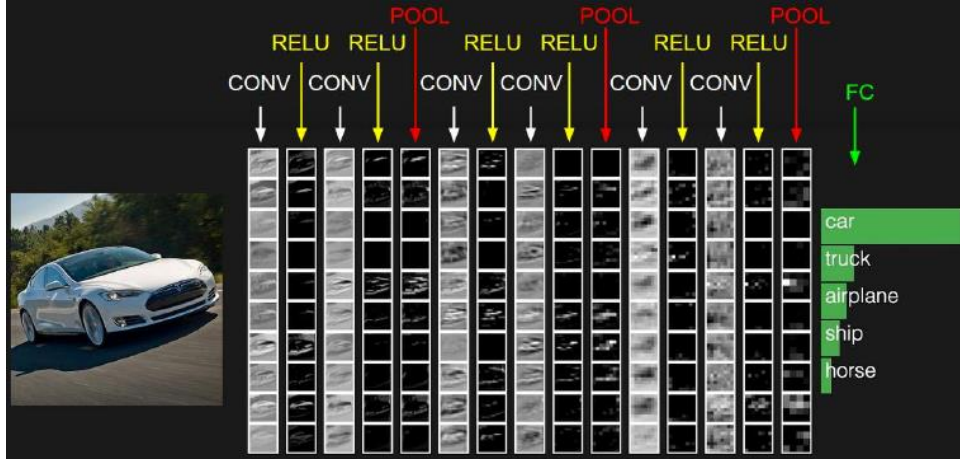


图 2.5 卷积神经网络结构示意图

（1）数据输入层（Input Layer）

数据输入层的工作是对图片进行预处理，一般处于网络的最前端。常见的预处理方法有去均值、归一化、PCA 白化。

- 取均值：把输入数据各个维度都中性化到 0。
- 归一化：把输入数据的幅度规约到同一范围。
- PCA：用主成分分析降维。
- 白化：对数据每个特征轴上进行归一化。

（2）卷积计算层（CONV Layer）

对于一副图像来说，其实后一层的每个节点没必要和上一层的所有节点相连，如果后一层一个单元和前一层所有连接，权重参数会异常的多。而我们的图像在视觉机制上表现为：图像上的某一部分的像素所表示的意义其实只和它周围的像素领域有关，这也是图像的空间特性。那么我们何不将上面每个节点连接范围缩小呢？这也叫做是局部感知。

当然 CNN 减少参数的方式可不止这些，CNN 减少参数的另一个核心方法是参数共享。所谓卷积，其实就是一种滤波方式。这在信号处理、图像处理上经常用到。最简单的在图像里面，一副图像和一个给定图像或者模版的进行卷积，卷积就是对应位置数据相乘然后再相加，如公式 7。

$$x_{i,j}^l = \sum_k^m \sum_q^n x_{i-k,j-q}^{l-1} h_{[m/2]-k,[n/2]-q} \quad (7)$$

其中， x^{l-1} 为原图像， h 为卷积核。

这种卷积运算实质上是将全连接的参数进行了共享。每个神经元连接数据窗的权重是固定的；每个神经元只关注某一种特性。这样便大大减少了网络参数，例如经典

网络 AlexNet 如果采用全连接参数将会有一亿多，而实际数字是 3.5 万。

（3）ReLU 激励层（ReLU Layer）

和人工神经网络的激励函数一样，是将卷积层的输出结果利用激励函数做非线性的映射。

（4）池化层（pooling Layer）

池化层一般夹在连续的卷积层中间，对数据进行下采样，用于压缩数据和参数的量，并减小过拟合。常用的池化方法有 Max Pooling 和 Average pooling。

（5）全连接层（FC Layer）

全连接层就是普通人工神经网络的层级结构，即两层之间所有神经元都有权重连接。通常处于神经网络的尾部，由于其参数过于巨大，近几年的 CNN 多采用 1×1 的小卷积核所构成的卷积层作为最后一层。

（6）BN 层（Batch Normalization Layer）

BN 层是对进行过多次处理后的数据进行处理使得数据重新符合正态分布且将数据归一化。由于本篇论文没有涉及此层，将不多赘述。

一般 CNN 的层次结构依次为：

- 输入层
- [[卷积层->激励层]*N->池化层?]*M
- [全连接层->激励层]*K
- 全连接层或小卷积层

2.4.2 常用 CNN 网络介绍

- LeNet，最早用于数字识别的 CNN。
- AlexNet，ILSVRC2012 远远超过第二名的 CNN，比 LeNet 更深，用多层小卷积层叠加替换掉大卷积层。
- ZF Net，ILSVRC2013 比赛冠军。
- GoogleNet，ILSVRC2014 比赛冠军。
- VGGNet，ILSVRC2014 比赛中用到的模型，图像识别率略差于 GoogleNet，但是在很多图像迁移学习问题（比如目标检测）上效果很好。
- ResNet，ILSVRC2014 比赛冠军，采用了残差学习以适应更深层次的 CNN 训练。

从此 CNN 的层数与日俱增。

2.5 本章小结

本章首先介绍了深度学习的背景知识，然后介绍了人工神经网络的基本概念，同时在对人工神经网络的理解的基础上，介绍了求解神经网络的一般性方法。介绍了反向传播算法和梯度下降算法。最后进一步介绍了卷积神经网络和常见的 CNN 结构。

第 3 章 车牌定位算法

3.1 常见的车牌定位算法

根据车牌的不同特征，可以采用不同的定位方法。目前车牌定位的方法很多，最常见的定位技术主要有基于边缘检测的方法、基于彩色分割的方法、基于小波变换的方法、基于遗传算法的方法、基于数学形态学的车牌定位和基于灰度图像纹理特征分析的方法等，在此对几种常用的定位算法进行简单的介绍。

3.1.1 基于边缘检测的车牌定位算法

边缘是指其周围像素灰度有阶跃变化的那些像素的集合。边缘的两侧分属于两个区域，每个区域的灰度均匀一致，而这两个区域的灰度在特征上存在一定的差异。边缘检测的任务是精确地定位边缘和抑制噪声。检测的方法有多种，例如 Roberts 边缘算子、Prewitt 算子、Sobel 算子以及拉普拉斯边缘检测。这些方法正是利用边缘处灰度变化剧烈这一特点来检测图像的边缘。各算子对不同边缘类型的敏感程度不同，产生的效果也不同，经过大量实验分析可知，Roberts 边缘算子是一种利用局部方差算子寻找边缘的算子，定位比较精确；Prewitt 算子和 Sobel 算子对噪声有一定的抑制能力，但不能完全排除伪边缘；拉普拉斯算子是二阶微分算子，对图像中的阶跃型边缘点定位准确且具有旋转不变性，但容易丢失一部分边缘的方向信息，同时抗噪能力较差。针对不同的环境和要求，选择合适的算子来对图像进行边缘检测才能达到好的效果。具体定位流程如图 3.1 所示。



图 3.1 基于边缘检测的车牌定位流程

该方法的定位准确率较高、反应时间快、能有效去掉噪声，适合于包含多个车牌的图像，在多车牌图像的情况下定位速度也很快。但是对车牌严重褪色的情况，由于检测不到字符笔画的边缘会导致定位失败，在有外界干扰以及车牌倾斜时，定位后的区域比车牌稍大。

3.1.2 基于彩色分割的车牌定位算法

基于彩色分割的车牌定位算法由彩色分割和目标定位等模块组成，采用多层感知器网络对彩色图像进行分割，然后通过投影法分割出潜在的车牌区域。在进行彩色分割时采用神经网络模型，一般图像采用 RGB 三原色，但由于 RGB 三原色中两点的欧氏距离不成线性比例。为了更好地进行彩色分割，一般将 RGB 模式的彩色图像转化为 HSI 模式，即色调、饱和度和亮度，然后对输出图像的饱和度作调整。为了减少计算量，将彩色图像下采样后再进行模式转化。同时，为了减少光照条件对图像分割产生的影响，采用对

数方法进行彩色饱和度调整。然后对模式转化后的彩色图像进行彩色神经网络分割，最后根据车牌底色及长宽比等先验知识，采用投影法分割出合理的车牌区域。具体定位流程如图 3.2 所示。

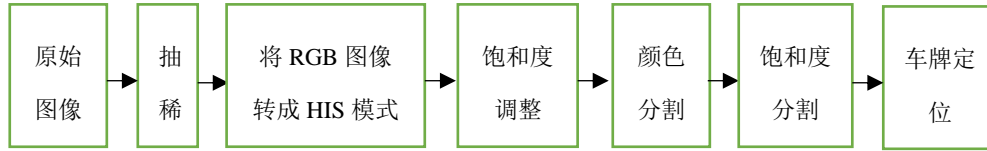


图 3.2 基于彩色分割的车牌定位算法流程

该定位算法正确率较高，但由于采用了神经网络计算法，当区域颜色与附近颜色相似时，计算速度较慢。当获取的彩色图像质量较高时，尤其是车牌区域颜色与附近颜色差别较大时，准确率将有所下降。

3.1.3 基于小波变换的车牌定位方法

小波分析是一种应用于图像处理的重要分析工具，具有“显微镜”的特性。小波分析的多分辨率特性使得小波分解系数在不同方向的高频子波系数具有不同特性，因此利用方向小波能够反映出图像在不同分辨率上沿任一方向变化的情形。小波分析的多尺度分解特性更加符合人类的视觉机制。

小波变换的基本思想是将原始信号经过伸缩、平移等运算分解为一系列具有不同空间分辨率、不同频率特性和方向特性的子带信号，这些子带信号具有良好的时频特性，通过利用这些特性可以实现对信号的时域、频域的局部分析。具体定位流程如图 3.3 所示。

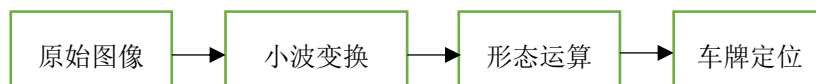


图 3.3 基于小波变换的车牌定位流程

目前利用小波分析的车牌定位算法大多是利用小波变换与其他多种方法相结合来实现更准确、快速的定位。例如基于小波分析和数学形态学的车牌定位方法，该方法通过小波多尺度分解提取出纹理清晰且具有不同空间分辨率、不同方向的边缘子图，然后利用车牌目标区域具有水平方向低频、垂直方向高频的特点实现子图提取，最后用数学形态学方法对小波分解后的细节图像进行一系列的形态运算，进一步消除无用信息和噪声，以确定车牌位置。该方法在噪声较小的情况下定位效果好，分割精度高；其缺点是速度较慢，且在噪声较大时误定位机率也随之增大。

3.1.4 基于遗传算法的车牌定位

基于遗传算法的车牌定位方法利用遗传算法对图像进行优化搜索，结合区域特征矢量构造适应度函数，最终寻找车牌区域的最佳定位参量。车牌定位是寻找一个符合“车牌区域特征”最佳区域的过程，本质上就是从参量空间寻找最优定位参量的问题，而寻找参量空间的最优解正是遗传算法所擅长的。但是在实时系统中，车牌定位速度受遗传算法中迭代次数的影响很大。具体定位流程如图 3.4 所示。

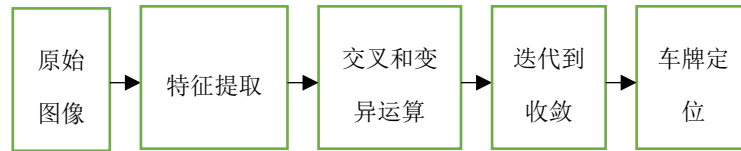


图 3.4 基于遗传算法的车牌定位流程

3.1.5 基于数学形态学的车牌定位方法

数学形态学图像处理的基本思想是利用一个结构元素来探测一个图像，看是否能将这个结构元素很好地放在图像内部，同时验证填放元素的方法是否有效。腐蚀、膨胀、开启和关闭是数学形态学的基本运算。具体定位流程如图 3.5 所示。

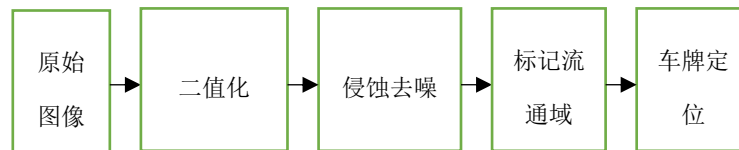


图 3.5 基于数学形态学的车牌定位流程

基于数学形态学的车牌区域定位方法不能精确确定车牌左右边界的位置，所以必须结合其他定位方法进行精确定位。例如基于数学形态学和边缘特征的车牌定位方法，这种方法先对车牌图像进行预处理，然后基于垂直方向结构元素的腐蚀运算进行滤波，再用闭合运算来填补车牌区域内细小孔洞，进而增强车牌区，使车牌区成为一个连通区域，最后利用字符边缘的特征对车牌进行准确的定位。该方法将数学形态学运算与数字图像的特征相结合，有效改进了传统的车牌定位方法，提高了车牌定位的速度和准确度。

3.1.6 基于灰度图像纹理特征分析的车牌定位方法

传统的纹理特征分析定位算法大多基于灰度图像来分析的，因此该算法需要对图像进行预处理，将彩色图像转换为灰度图像，然后进行行扫描，找出图像中每一行所含有的车牌线段，记录下它们的起始坐标和长度，如果有连续若干行均存在不少于一个的车牌线段，且行数大于某一确定的阈值，则认为在行的方向上找到了车牌一个候选区域，并确定了该候选区域的起始行和高度；在已找到的可能存在车牌的区域做列扫描，以确定该车牌

候选区域的起始行和高度以及起始列坐标和长度，由此确定一个车牌区域；继续在其他可能存在车牌的区域寻找，直至找到所有的车牌候选区域。



图 3.6 基于灰度图像纹理特征分析的车牌定位流程

该算法对于牌照倾斜或变形以及光照不均、偏弱或偏强有很好的效果，但对噪声敏感，对于背景复杂的图像可以结合垂直投影的方法来得到真正的车牌区域，可以有效地解决背景复杂的车牌定位。

车牌定位技术是车牌识别系统中的一个重要环节，在定位的精度、计算速度和适用的可靠性方面还需要进一步改进和提高。目前，还没有一种算法能够实现对于任意背景、位置和光照条件下的汽车图像进行车牌定位。

3.2 MSER 算法

最稳定极值区域（maximally stable extremal region，MSER）^[7]。下面是 MSER 提取过程：

- 1.使用一系列灰度阈值对图像进行二值化处理，得到一系列二值图像；
- 2.对于每个阈值得到的二值图像，得到相应的黑色区域与白色区域；
- 3.在比较宽的灰度阈值范围内保持形状稳定的区域就是 MSER，判断标准为 dA/dt ；

David Nister 等人于 2008 年在对 J.Matas 提出的 MSER 算法进行了改进^[8]，使其在时间复杂度方面比原算法要小得多，算法思路如下：

- ① 像素点排序→②极值区域生成→③稳定区域判定→④区域拟合→⑤区域归一化



图 3.7 原始图像



图 3.8 MSER 分步操作结果

如图 3.8，先使用 MSER 算法提取出图像的区域，然后对这些区域进行形态学闭操作，使其成较为完整的区域，再对这些区域求外接矩形，根据外接矩形的参数去判断是否是车牌区域。

3.3 本章小结

本章分别介绍了基于边缘检测、彩色分割、小波变换、遗传算法、数学形态学、图像纹理特征分析地车牌定位方法，进而介绍了本文采用的 MSER 算法。在介绍这些算法时，着重介绍了这些算法的适应情况及面临的问题。

第4章 字符切割算法

4.1 切割预处理

通过上一步车牌定位我们得到了车牌的区域，进而得到了只包含车牌的图像。在进行下一步之前我们首先对得到的车牌图像进行下列预处理。

4.1.1 偏斜扭正

解决了车牌的定位问题以后，下面的问题是：在定位以后，我们如何把偏斜过来的车牌扭正呢？这个过程叫做偏斜扭转过程。我们在这里采用仿射变换来进行偏斜扭转，根据实际四边形的形状对所得区域进行仿射变换得到车牌的正视图。



图 4.1 偏斜扭正示意图

4.1.2 旋转操作和调整大小

旋转操作是提高后面的字符识别率的关键环节，是将偏斜的车牌调整至水平的操作。可以看出，没有旋转操作的车牌是倾斜，加大了后续车牌判断与字符识别的难度。因此最好需要对车牌进行旋转。在角度判定阈值内的车牌矩形，我们会根据它偏转的角度进行一个旋转，保证最后得到的矩形是水平的。如图4.2所示。

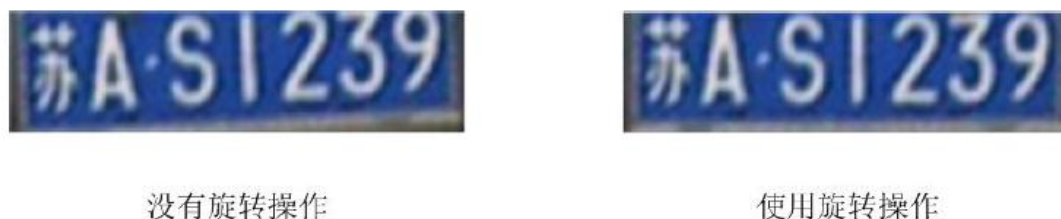


图 4.2 旋转操作示意图

最后，在我们把这些候选车牌导入后续步骤之前，需要对图像进行缩放确保他们的尺寸一致。

4.2 字符切割技术

4.2.2 中文车牌切割先验知识介绍

为了使计算机拥有中国车牌分割和识别的知识，本文提出了一套关于中文车牌分割的算法。为了准确提取车牌中的字符，必须清楚中国标准牌照的结构和格式。

中国标准牌照格式：A1 A2·A3 A4 A5 A6 A7，A1 是汉字，省，市，自治区的缩写；A2 是英文字母；A3 A4 A5 A6 A7 分别代表数字或英文字母；A2 和 A3 之间有点分隔符。如图 4.3 所示。

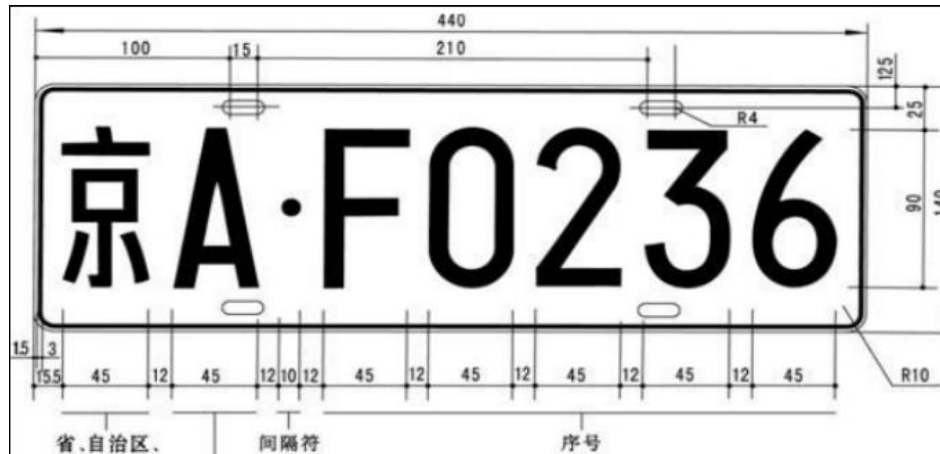


图 4.3 中国标准牌照格式示例

如图 4.3 所示，车牌轮廓尺寸为 140mm * 440mm；字符尺寸为 90mm * 45mm；差距为 90mm * 12mm；字符和板外轮廓之间的边界距离为 25mm；左右边界的字符与平板外轮廓之间的距离为 15.5mm；第二和第三个字符之间的距离为 34mm；点直径为 10mm。使用这些先验知识可以帮助我们有效地分割车牌。^[9]

4.2.2 字符切割过程

字符分割是指将整个车牌矩形图像区域分割成小区域；每个地区都有独立的字符。本文采用垂直投影方法进行字符分割。垂直投影方法的步骤如下。

在找到车牌位置后，根据图 4.3 的车牌图像标准格式：每个字符的宽度是一定的，第二个和第三个字符之间的空间大于其他字符之间的空格。因此，在执行偏斜扭正之后，每个字符的宽度被定义为分割段的长度，图像的左端和右端用作边界。

首先将图像灰度按相似性分为两类，使其变为二值图像。车牌的二值图像是垂直投影的，每列白点的统计信息保存在数组中，J 是相应列的值。分析投影值，字符区域通常对应于占据位置的峰，字符显示在凹槽之间；图 19 示出了数组的图形。

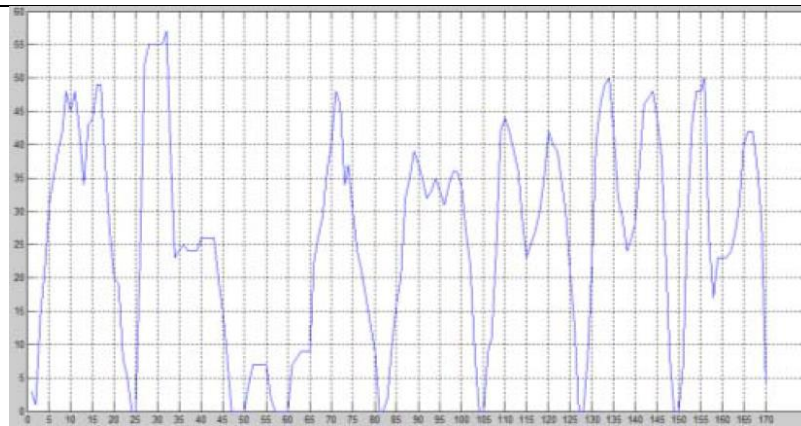


图 4.4 白点统计数组图示

根据给出阈值，得到存储在起始点和终点坐标的每个峰。根据上述步骤，字符分割结果如图 4.5 所示。



图 4.5 字符切割结果图示

4.3 本章小结

本章首先对字符切割之前所需要做的预处理操作进行了介绍，包括偏斜扭正、旋转操作和调整大小。然后介绍了中国标准牌照相关的先验知识，然后介绍了根据此先验知识对车牌图像进行字符切割的方法及过程。接下来，我们将着重介绍关键的基于共享隐层 CNN 的字符识别算法。

第5章 基于共隐层 CNN 的字符识别算法

在字符识别中，卷积神经网络（CNN）的性能优于其他大多数模型。然而，为了保证令人满意的表现，CNN 通常需要大量的样本。由于中文和字母数字的差异，识别两类的最常用方法是分别使用两个独立的 CNN。由于我们汉字数据较少，本文为了解决汉字短缺的问题，我们实现了一个汉语和字母数字字符共享隐藏层的 CNN 模型，此模型含有两个不同的 softmax 输出层。为了避免训练过程中的过拟合问题，采用了早期停止规则。

5.1 模型网络结构

参考 J. Bai 的方法，我们使用的共享隐藏层 CNN（SHL-CNN）模型的概述如图 5.1 所示。在这个模型中，所有的卷积层和池化层都是在识别中文和字母数字任务之间共享的。这些隐藏层能提取出中文和字母数字的共同特征。底层 softmax 层不共享，作为两个面向任务的分类器。与 J.Bai 的网络不同，我们没有采用归一化层。^[10]

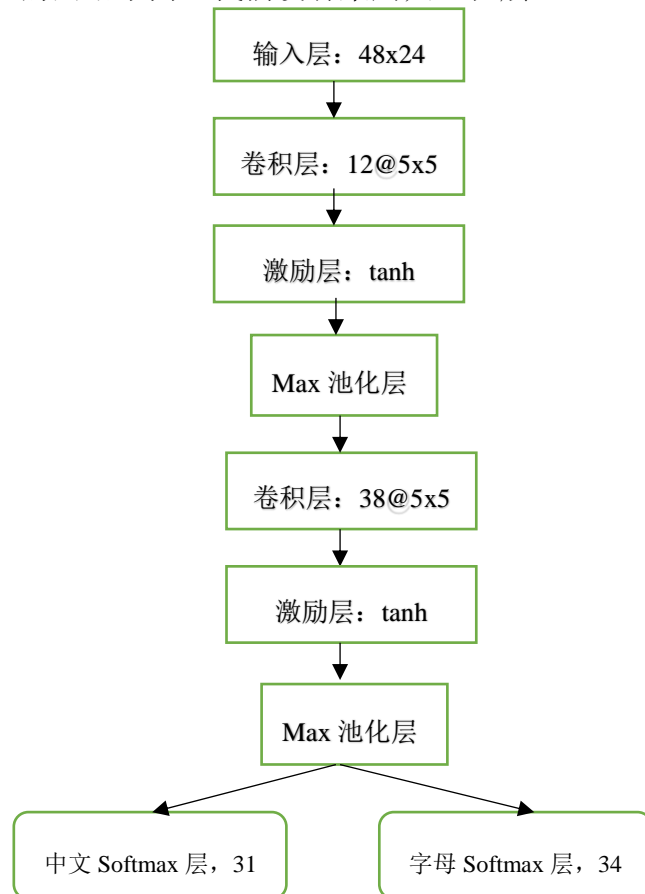


图 5.1 共隐层 CNN 网络结构

如图 5.1 所示，我们首先将输入图像的尺寸都缩放为 48×24 。在两个卷积层中，我们统一使用 5×5 的卷积核进行操作，两层的神经元个数分别为 12 和 38。池化层采用 Max Pooling，激励层采用双曲正切函数，在网络的尾端，分别加上两个 FC 层并采用 softmax 函数进行归一化，输出分别对应 31 个汉字省份和 34 个英文数字字符。

5.2 实验结果与分析

表 5.1 实验结果

数据	模型		绝对优化率	相对优化率
	单独的 CNN	共隐层 CNN		
中文	4.254%	3.632%	0.622%	14.622%
字母数字	1.560%	1.335%	0.225%	14.423%

由表 1 可知，通过对我们两个小型车牌字符数据库进行培训和测试，该模型在中文字符和字母字符上分别比常规 CNN 模型分别降低了 14.622% 和 14.423% 的测试集错误率。图 22 和图 23 表述了各中文和字母数字在测试集上的识别错误率分布。

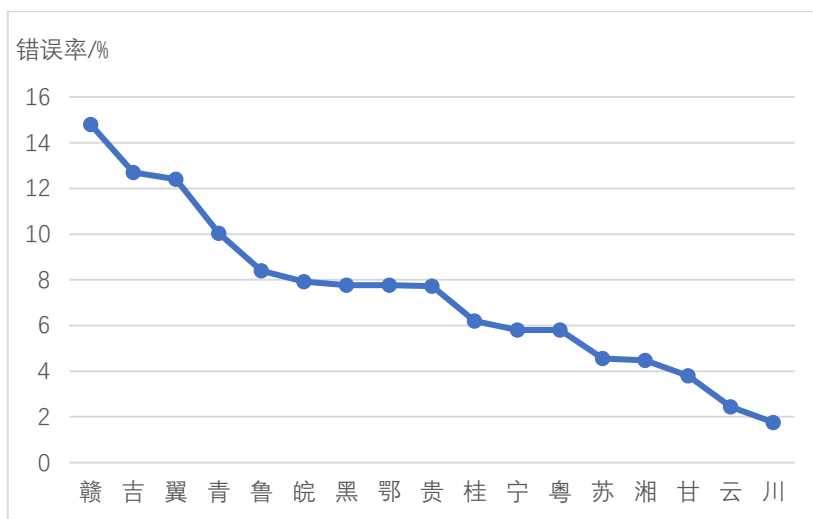


图 5.2 中文识别错误率

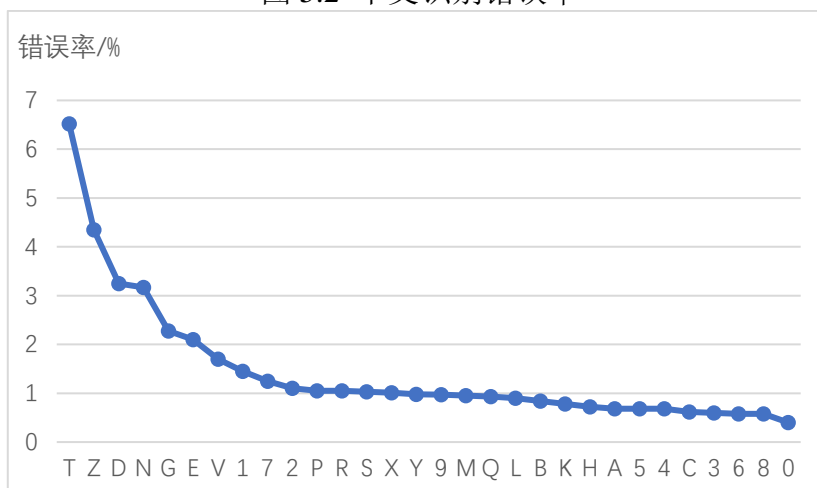


图 5.3 字母数字识别错误率

5.3 本章小结

本章首先介绍了共隐层 CNN 的网络结构，然后对实验的过程和结果进行了分析。该模型在中文字符和字母字符上分别比常规 CNN 模型分别降低了 14.622% 和 14.423% 的测试集错误率。

第 6 章 CUDA 编程介绍

6.1 CUDA 背景

近年来，计算机行业界正在迅速迈进并行计算时代。到 2010 年，几乎所有的计算机都采用了多核处理器。并行计算已经不再是超算的专属技术。随着时间的推移，越来越多的并行计算平台和技术即将出现。近年来计算机制造商在集成电路元器件中存在的各种限制，已经无法在现有的架构上通过提高处理器时钟频率来提高性能。随着计算机功耗与发热的急剧升高以及晶体管的大小已经接近 10 纳米，研究人员和制造商开始寻求其他的方式。在此背景下，多核革命的发生使得个人计算机上开始发生重大的转变。

与 CPU 传统的流水线数据处理方式相比，图像处理器 (Graphics Processing Unit, GPU) 上执行通用计算还是一个新的概念。然而，在图形处理器上执行计算并非新概念。由于 GPU 有着很高的计算吞吐量，因此 GPU 计算有着非常光明的应用前景。然而，这种编程模型对于开发人员来说具有很大的局限性。首先，该模型具有严格的资源限制，因为程序只能以颜色值和纹理单元等形式来输入数据。其次，程序员在将计算结果写入内存的方式以及位置上同样存在着严格的限制，如果在算法中需要将数据写入到内存的分散位置，将无法在 GPU 上运行。而且，GPU 无法处理浮点数据，这导致大多数科学计算都无法使用 GPU。此外，没有合适的调试工具也阻碍了 GPU 计算的发展。

在此背景下，NVIDIA 于 2006 年推出 CUDA (Compute Unified Devices Architecture)，可以利用其推出的 GPU 进行通用计算，将并行计算从大型集群扩展到了普通显卡，使得只需要一台带有显卡的笔记本就能跑较大规模的并行处理程序。

6.2 CUDA 编程简介

6.2.1 Hello, World!

我们通过一段简单的示例来学习 CUDA C，因此来看第一个 CUDA C 示例。首先给出的是一 "Hello World!" 示例。

```
#include "iostream"

int main(void){
    printf ("Hello, world\n") ;
    return 0;
}
```

当着到这段代码时，你肯定发现这和普通的 C 语言一模一样。是的，CUDA C 与熟悉的 C 语言很大程度上没有任何区别。

6.2.2 核函数调用

现在，我们在示例程序中添加一些代码，这些代码比最初的程序看上去会陌生一些。

```
#include <iostream>

__global__ void kernel( void ) {
}

int main( void) {
    kernel<<<1,1>>>>();
    printf("Hello,world\n");
    return 0;
}
```

这个程序与最初的程序相比，多了两个值得注意的地方：

- 一个空的函数 `kernel()`，并带有修饰符 `__global__`。
- 对这个空函数的调用，并且带有修饰字符 `<<<1,1>>>`。

在上一节中看到，代码默认是由系统的标准 C 编译器来编译的。例如，在 Linux 操作系统上用 GNU gcc 来编译主机代码，而在 Windows 系统上用 Microsoft Visual C 来编译主机代码。NVIDIA 工具只是将代码交给主机编译器，它表现出的行为就好像 CUDA 不存在一样。

现住，我们看到了 CUDA C 为标准 C 增加的 `__global__` 修饰符。这个修饰符将告诉编译器，函数应该编译为在设备而不是主机上运行。在这个简单的示例中，函数 `kernel()` 将被交给编译设备代码的编译器，而 `main()` 函数将被交给主机编译器（和上一个例了一样）。

那么，`kernel()` 的调用究竟代表着了什么含义，并且为什么必须加上尖括号和两个数值？注意，这正是使用 CUDA C 的地方。

我们已经看到，CUDA C 需要通过某种语法方法将一个函数标记为设备代码。这并没

有什么特别之处，只是一种简单的表示方法，表示将主机代码发送到一个编译器，而将设备代码发送到另一个编译器。事实上，这里的关键在于如何在主机代码中调用设备代码。CUDA C 的优势之一在于，它提供了与 C 在语言级别上的集成，因此这个设备函数调用看上去非常像主机函数调用。

6.2.2 参数传递

前面提到过可以将参数传递给核函数，现在就来看一个示例。

```
#include <iostream>

__global__ void add( int a , int b , int *c ) {
    *c = a + b;
}

int b, int *c){
int main( void ) {
    int c;
    int *dev_c;
    HANDLE_ERROR( cudaMalloc( (void**)&dev_c , sizeof( int ) ) );
    add<<<1,1>>>( 2 , 7 , dev_c );
    HANDLE_ERROR( cudaMemcpy( &c,
                                dev_c,
                                sizeof(int),
                                cudaMemcpyDeviceToHost ) );

    printf(" 2 + 7 = %d\n",c);
    cudaFree( dev_c );
    return 0;
}
```

需要注意的地方在于通 `cudaMalloc()`来分配内存。这个函数调用的行为非常类似于标准的 C 函数 `malloc()`，但该函数的作用是告诉 CUDA 运行时在设备上分配内存。第一个参数是一个指针，指向用于保存新分配内存地址的变量，第二个参数是分配内存的大小。除了分配内存的指针不是作为函数的返回值外，这个函数的行为与 `malloc()`是相同的，并且返回类型为 `void*`。要释放 `cudaMalloc()`分配的内存，需要调用 `CudaFree()`，这个函数行为和 C 中 `free()`一致。

在主机代码中可以通过调用 `cudaMemcpy()`来访问设备上的内存。这个函数调用的行为类似于标准 C 中的 `memcpy()`，只不过多了一个参数来指定设备内存指针究竟是源指针还

是目标指针。在这个示例中，注意 `cudaMemcpy()` 的最后一个参数为 `cudaMemcpyDeviceToHost`，这个参数将告诉运行时源指针是一个设备指针，而目标指针是一个主机指针。

6.3 CUDA 在算法中的应用

6.3.1 CUDA 在训练中的应用

在字符识别共隐层 CNN 的训练中，网络只在特定层（如输入层、全连接层）与其他层有全面的连接，而其他较为独立的直线连接关系即可作为模型的可并行部分。将模型的可并行部分拆分到多个 GPU 上，同时利用多个 GPU 的计算能力各执行子模型的计算，可以大大加快模型的单次前向-后向训练时间。

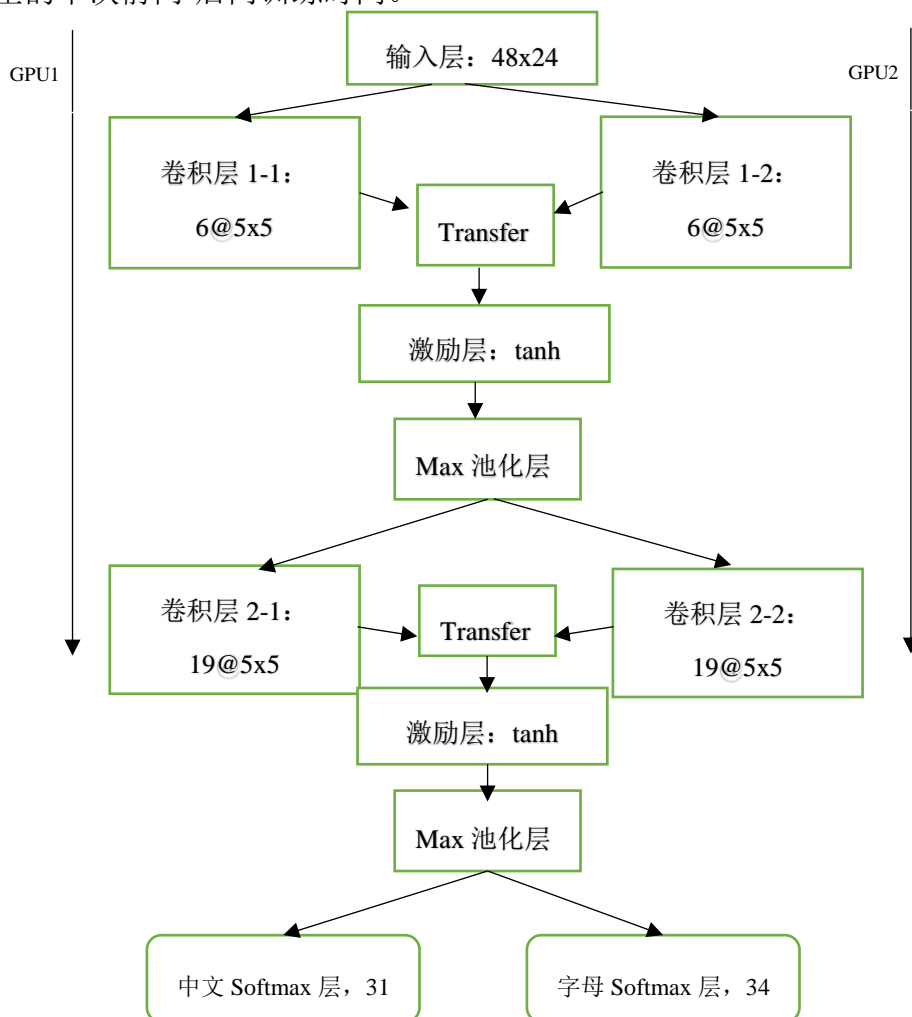


图 6.1 两个 GPU 网络结构示意图

如图 6.1 所示，我们将每个卷积层分别拆分为两个卷积层让两个 GPU 各自独立运算，然后通过 transfer 单元将两者计算的结果合并。

6.3.2 CUDA 在预测中的应用

在车牌定位中，除一些图像处理步骤可以分区计算外，多个疑似车牌区域的判别是可以并行计算的；在字符分割过程中，统计白点数目也是可以并行计算的，但由于计算量较小，实际中未采用；在字符识别的前向计算中，我们仍可以分割网络进行并行处理。

6.4 本章小结

本章中，我们首先介绍了并行计算和 GPU 的背景知识。然后从示例代码出发，介绍了 CUDA C 程序的基本构成和思想，这个过程比你想象的要更轻松。从本质上来说，CUDA C 只是对标准 C 进行了语言级的扩展，通过增加一些修饰符使我们可以指定哪些代码在哪个设备上运行。在函数前面添加关键字 `__global__` 将告诉编译器把该函数放在 GPU 上运行。为了使用 GPU 的专门内存，我们还介绍了与 C 的 `malloc()`/`memcpy()` 和 `free()` 等 API 相对应的 CUDA API。最后针对本文，介绍了 GPU 并行计算 CUDA 在车牌识别中的运用。

第 7 章 总结与展望

7.1 论文总结

车牌定位和识别是目前研究的热点与重点，他主要包括三个任务：车牌的定位、字符分割和识别。依据不同的实用环境，同一个算法可能有不同的效果，目前没有一个能在所有环境下都达到完美效果的算法。在车牌定位中，本文总结了常见的六种类型的算法：基于边缘检测、彩色分割、小波变换、遗传算法、数学形态学、灰度图像纹理特征分析的车牌定位算法，随后介绍了 MSER 算法；在字符分割中，首先介绍了车牌图像的预处理操作，然后介绍了中国标准车牌的先验知识，最后介绍了切割方法和过程；在字符识别中，本文提出了一个共隐层的 CNN 来解决中文字符和字母数字识别间的差异，并分析了实验结果，比单独用两个 CNN 错误率分别降低了 14.622% 和 14.423%。最后本文介绍了该识别算法如何在 CUDA C 环境中进行并行优化。

7.2 工作展望

在本文的研究中，所提出的算法虽然在实验结果上基本符合预期的设想，基于共隐层 CNN 算法使字符识别的错误率大大的降低，且缓解了因中文字符的缺失带来的过拟合问题。但因为自己知识水平和能力的有限，一些算法设计在后续还可以作如下方面的改进：

（1）在车牌定位过程中，可以考虑适当降低 MSER 算法的判定阈值，增加可能是车牌的区域，然后通过车牌判定，进而判断是否为车牌。

（2）在字符切割过程中，可以考虑适当降低白值阈值使字符范围适度扩大，然后加入字符判定单元，判断是否为真实的字符。也可以参考文献^[19]，采用 CNN 去做字符切割。

致谢

四年时间一晃而过，毕业论文完成之时也就意味着我的大学四年生活即将画上句点。

首先，我由衷的感谢我的指导老师袁景凌教授，从毕业设计最初的定题到资料的收集与具体设计，她都给了我耐心的指导和无私的帮助。从袁老师给我们上课时的那一种专注和认真，我就知道她是一个知识渊博的学者，同时对待学生的提问，她总能和蔼耐心的解答。在论文的创作及完成阶段，她真给了我莫大的鼓励与帮助，在这里再次的感谢袁老师，祝她工作顺利、生活幸福美满。

其次，我要感谢在我本科阶段所有教授我知识的老师们，是他们让我走进知识的殿堂，教会我学习及生活的技能。感谢实验室师兄陈旻骋在我论文写作阶段对我的悉心指导以及室友和同学在生活给予我的帮助与照顾。

最后，我要感谢生我养育我的父母，是他们培育我长大，教我做人的道理。在这一路的学习生涯中始终伴我左右，给我关怀与鼓励。你们永远的健康快乐就是我最大的心愿。

康康

2016 年 5 月于武汉

参考文献

- [1] Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25. (2012) 1097{1105
- [2] Deghan, A., Masood, S.Z., Shu, G., Ortiz., E.G.: View independent vehicle make, model and color recognition using convolutional neural network. In: arXiv:1702.01721. (2017)
- [3] Hong T, Gopalakrishnam A K. License plate extraction and recognition of a Thai vehicle based on MSER and BPNN[C]. Knowledge and Smart Technology (KST), 2015 7th International Conference on. IEEE, 2015: 48-53.
- [4] DE Rumelhart, GE Hinton, RJ Williams, Learning internal representations by error propagation. 1985 –

DTIC Document.

- [5] Y. LeCun , B. Boser , J. S. Denker , D. Henderson , R. E. Howard , W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural Computation, vol. 1, no. 4, pp. 541-551, 1989.
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [7] Matas J, Chum O, Urban M, et al. Robust wide-baseline stereo from maximally stable extremal regions[J]. Image and vision computing, 2004, 22(10): 761–767.
- [8] Nistér D, Stewénus H. Linear time maximally stable extremal regions[J]. Computer Vision–ECCV 2008, 2008: 183-196.
- [9] Wang N, Zhu X, Zhang J. License Plate Segmentation and Recognition of Chinese Vehicle Based on BPNN[C]、 Computational Intelligence and Security (CIS), 2016 12th International Conference on. IEEE, 2016: 403-406.
- [10] J. Bai, Z. Chen, B. Feng, and B. Xu, "Image character recognition using deep convolutional neural network learned from different languages," International Conference on Image Processing, DOI. 10.1109/ICIP.2014.7025518, pp. 2560-2564, 2014.
- [11] Hsieh J W, Yu S H, Chen Y S. Morphology-based license plate detection from complex scenes[C]//Pattern Recognition, 2002. Proceedings. 16th International Conference on. IEEE, 2002, 3: 176-179.
- [12] Arth C, Limberger F, Bischof H. Real-time license plate recognition on an embedded DSP-platform[C]//Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007: 1-8.
- [13] Yuan Y, Zou W, Zhao Y, et al. A robust and efficient approach to license plate detection[J]. IEEE Transactions on Image Processing, 2017, 26(3): 1102-1114.
- [14] Du S, Ibrahim M, Shehata M, et al. Automatic license plate recognition (ALPR): A state-of-the-art review[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2013, 23(2): 311-325.
- [15] Li H, Shen C. Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs[J]. arXiv preprint arXiv:1601.05610, 2016.
- [16] Chang S L, Chen L S, Chung Y C, et al. Automatic license plate recognition[J]. IEEE transactions on intelligent transportation systems, 2004, 5(1): 42-53.
- [17] Anagnostopoulos C N E, Anagnostopoulos I E, Psoroulas I D, et al. License plate recognition from still images and video sequences: A survey[J]. IEEE Transactions on intelligent transportation systems, 2008, 9(3): 377-391.

- [18] Gou C, Wang K, Yao Y, et al. Vehicle license plate recognition based on extremal regions and restricted boltzmann machines[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1096-1107.
- [19] Cheang T K, Chong Y S, Tay Y H. Segmentation-free Vehicle License Plate Recognition using ConvNet-RNN[J]. arXiv preprint arXiv:1701.06439, 2017.
- [20] Hsu G S, Chen J C, Chung Y Z. Application-oriented license plate recognition[J]. IEEE Transactions on Vehicular technology, 2013, 62(2): 552-561.