学号：**2017012872**

西北农林科技大学
NORTHWEST A&F UNIVERSITY

2021 届本科生毕业设计

基于深度学习的复杂环境下车牌检测与识别方法研究

学 院 （ 系 ）：	机械与电子工程学院
专　　业：	电子信息工程
年 级 班 级：	2017 级 1 班
学 生 姓 名：	杨奕铭
指 导 教 师：	宋怀波
协助指导教师：	无
完 成 日 期：	2021 年 5 月

本科生毕业论文（设计）的独创性声明

本人声明：所呈交的本科毕业论文（设计）是我个人在导师指导下独立进行的研究工作及取得的研究结果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究结果，也不包含其他人和自己本人已获得西北农林科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文的致谢中作了明确的说明并表示了谢意。如违反此声明，一切后果与法律责任均由本人承担。

本科生签名：　　时间：　　年　　月　　日

关于本科生毕业论文（设计）知识产权的说明

本毕业论文(设计)的知识产权归属西北农林科技大学。本人同意西北农林科技大学保存或向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅。

本人保证，在毕业离开西北农林科技大学后，发表或者使用本毕业论文（设计）及其相关的工作成果时，将以西北农林科技大学为第一署名单位，否则，愿意按《中华人民共和国著作权法》等有关规定接受处理并承担法律责任。

任何收存和保管本论文各种版本的其他单位和个人(包括作者本人)未经本论文作者的导师同意，不得有对本论文进行复制、修改、发行、出租、改编等侵犯著作权的行为，否则，按违背《中华人民共和国著作权法》等有关规定处理并追究法律责任。

本科生签名：　　时间：　　年　　月　　日

指导教师签名：　　时间：　　年　　月　　日

目 录

缩略词

表 1 本文缩略词

Table 1 Acronyms in this paper

缩写	中英文全称
CCA	Connected Component Analysis 连通区域分析
VQ	Vector Quantization 矢量量化
SCW	Sliding Concentric Window 滑动同心窗口
WT	Wavelet Transform 小波变换
SIFT	Scale-Invariant Feature Transform 尺度不变特征变换描述子
SVM	Support Vector Machine 支持向量机
CNN	Convolutional Neural Networks 卷积神经网络
CCPD	Chinese City Parking Dataset 中国城市车牌数据集
RCNN	Regions with CNN features 具有 CNN 特征的区域
HMM	Hidden Markov Model 隐性马尔可夫模型
CTC	Connectionist Temporal Classification 连接时序分类
RNNs	Recurrent Neural Networks 双向递归神经网络
BRNN	Bidirectional RNN 双向循环神经网络
AP	Average Precision 平均精确度
FPS	Frames Per Second 每秒帧数
GLOPS	Giga Floating-point Operations Per Second 每秒 10 亿次的浮点运算数
IoU	Intersection over Union 交并比
FPN	Feature Pyramids 特征金字塔
PAN	Path Aggregation Network 路径聚合网络
NMS	Non Maximum Suppression 非极大值抑制
LSTM	Long-Short Term Memory 长短期记忆

基于深度学习的复杂环境下车牌检测与识别方法研究

作者：杨奕铭　　指导教师：宋怀波

摘 要：车牌检测与识别系统于路面监控、高速公路收费管理、智能泊车系统、交通违规记录等方面发挥着重要作用，能够有效治理城市交通、对车辆进行更有效的管控，在现实生活中具有重要意义。然而目前车牌检测识别的研究仍存在复杂环境条件下(如光照变化、角度倾斜、视野模糊)检测与识别困难的问题。因此，如何有效地检测与识别复杂环境条件下的车牌成为智慧交通管理中重要的研究方向。本研究基于深度学习方法，以 CCPD 数据集中复杂环境下的车牌作为研究对象，使用 YOLOV5 目标检测算法与 CRNN+CTC 的字符识别算法实现了复杂环境下的车牌检测与识别，并且围绕着上述研究内容设计了一套包含微信小程序前端、PyQt5 管理者前端、云开发数据库的智能停车系统，对于促进城市的智慧交通管理发挥着重要的作用。本研究具体研究内容与主要结论如下：

(1) 为了解决车牌图像获取困难、训练集中省份数据不均的问题，提出一种生成车牌数据集的方法并进行了验证，结果表明该方法生成的数据集可在车牌检测、车牌识别方面起到较好的辅助作用；

(2) 为了准确定位并提取车牌区域，采用深度学习网络 YOLOV5 对车牌区域进行检测，并且加入 Landmark 回归的方法以定位车牌的四个角点位置，为后续倾斜车牌矫正奠定了基础。网络选择方法在综合考虑平均精度、推理速度、参数量、权重大小等因素后，优选使用 YOLOV5S 为主要网络结构，经中国车牌数据集(Chinese City Parking Dataset, CCPD)训练后的测试结果与学者们的实验结果对比表明，本研究的车牌检测网络在 CCPD 数据集上的平均准确率为 99.8%，每帧处理速度为 10 ms，表明本研究使用的车牌检测网络效果具有检测速度快、检测精度高的特点，可在各种复杂环境下实现车牌的精确检测；

(3) 在识别车牌算法方面，考虑到车牌识别是一个特殊的字符识别实例，选择使用 CRNN+CTC 网络结构作为车牌识别网络，同时为了提升算法对倾斜车牌的识别效果，使用透射变换实现了倾斜车牌的矫正。使用 CCPD 数据集做训练后的测试结果与学者们的实验结果对比表明，本研究的车牌识别网络在 CCPD 数据集上的平均准确率为 98.8%，每帧处理速度为 15 ms，说明本研究的车牌识别网络在保证识别速度的同时，具备良好的识别效果，特别是对于倾斜车牌的识别效果有了进一步的提高；

(4) 使用微信小程序与 PyQt5 分别设计了用户前端与管理者前端，后端使用了微信云端数据库进行搭建，共同实现了注册与登录、车牌信息登记、个人信息查询、停车信息查询等功能。其功能完善，界面美观的优点，可以很好地满足用户与管理者两方的需求。

关键词：车牌；深度学习；目标检测；光学字符识别

License Plate Detection and Recognition in Complex Environment based on Deep Learning

Author: Yiming Yang Tutor: Huaibo Song

Abstract: License plate detection and recognition system plays an important role in road monitoring, highway toll management, intelligent parking system, traffic violation record and so on. It can effectively control urban traffic and control vehicles more effectively. It is of great significance in real life. However, at present, there are still some difficulties in license plate detection and recognition under complex environmental conditions (such as illumination change, angle tilt, blurred field of vision). Therefore, how to effectively detect and recognize license plates under complex environmental conditions has become an important research direction in intelligent traffic management. Based on the deep learning method, in this study, the license plates taken in the complex environment of CCPD data set were selected as the research object, and YOLOV5 target detection algorithm and CRNN+CTC character recognition algorithm were applied to realize license plate detection and recognition in complex environment, and a set of intelligent parking system including WeChat Mini Programs front-end, PyQt5 manager front-end and cloud development database around the above research content were designed, which plays an important role in promoting the intelligent traffic management of the city. The specific research work and conclusions of this study are as follows:

(1) In order to solve the difficult problem of obtaining license plate image, a method of generating license plate data set was proposed, and it was analyzed that the data set generated by this method can play an auxiliary role in license plate detection and license plate recognition. it was also verified that using the generated data set for license plate recognition training can alleviate the possible provincial inequality in the license plate training set.

(2) In order to accurately locate and extract the license plate area, the deep learning network YOLOV5 was used to detect the license plate area, and the Landmark regression method wa added to locate the four corners of the license plate, which lays a foundation for the subsequent tilted license plate correction. After considering the factors such as average accuracy, reasoning speed, number of parameters and weight, the network selection method preferably uses YOLOV5S as the main network structure. the test results trained by CCPD data sets were compared with the experimental results of scholars. the results showed that the average accuracy of the license plate detection network in CCPD data set was 99.8%, and the processing

speed of each frame was 10 ms. It showed that the license plate detection network effect of this study has the characteristics of fast detection speed, high detection accuracy and good detection effect for license plates in various complex environments.

(3) In the aspect of license plate recognition algorithm, considering that license plate recognition is a special character recognition example, the network structure of CRNN+CTC was selected as license plate recognition network, and transmission transformation was used to correct tilted license plate before recognition network. The comparison between the test results after training using CCPD data set and the experimental results of scholars showed that the average accuracy of the license plate recognition network in this study was 98.8% on the CCPD data set, and the processing speed per frame was 15 ms, indicating that the license plate recognition network in this study not only ensures the recognition speed, but also has a good recognition effect, especially for tilted license plate recognition.

(4) WeChat Mini Programs and PyQt5 were used to design the front end of the user and the front end of the manager, and the back end was built by Wechat cloud database, which jointly realizes the functions of registration and login, license plate information registration, personal information query, parking information query and so on. With the advantages of perfect function and beautiful interface, it can well met the needs of both users and managers.

Keywords: license plate; deep learning; object detection; optical character recognition

第一章 绪论

1.1 研究目的及意义

中国是世界上最大的发展中国家，同时也是世界上人口最多的国家，近年来中国社会上民用机动车保有量迅速增加，这得益于中国社会的生产力与消费能力的提升。据最新公开的我国公安部交通管理局 2020 年的统计数据表明，我国民用汽车保有量由 2003 年的 2400 万辆增长至 2020 年的 2.81 亿辆，年平均增长率 15.56%。由于汽车产业与多个生产制造领域息息相关且汇聚了多个领域的技术核心，因此汽车产业在国际层面上经常作为一个国家的制造业水平和科技生产能力的衡量标准，更是一个国家综合国力的直接体现(W. Chu 2011; P. Chaney et al. 1991)。随着我国汽车保有量的快速增长，一些城市可能会面临各类由汽车保有量上升而直接导致的交通问题，例如：道路拥堵、事故频发、空气污染等城市交通问题(陈璇璇 2008; C. Ding et al. 2017)。随着人们生活汽车使用频率的增加，汽车使用过程中也逐渐出现了一些较为恶劣的问题，如：交通肇事，无证驾驶，毒驾醉驾，噪声污染等问题(C. Lum et al. 2011; 盖盈盈 2017; 李媛 et al. 2021)。因此，为了解决这些车辆使用过程导致的棘手问题，各国都在积极研究如何更有效地管理和监控车辆，仅仅依靠交警等人力资源，将会带来高成本、低效率的问题。

使用智能交通设备来对车辆进行智能化管理可以提升交通监管的效率。在汽车保有量逐年上升的背景下，智能交通系统研究的热度逐步上升(C. Anagnostopoulos et al. 2006)。“智慧城市”的概念被学者们逐步具象化，“智能视频监控”和“智能交通疏导”成为目前研究的热点，为未来城市交通的智能化管理奠定基础。车牌检测与识别系统于路面监控、高速公路收费管理、智能泊车系统、交通违规记录等方面发挥着重要作用，能够有效治理城市交通，防止交通堵塞，在现实生活中具有重要意义(G. Liu et al. 2011; Y. Chiou et al. 2011; S. Kranthi et al. 2011; C. Anagnostopoulos et al. 2008)。

如何使用图像处理技术正确地在复杂的交通环境中检测与识别车牌，成为“智慧交通”促进车辆管理的重要一环。目前学术界有不少针对中国车牌识别的相关研究，仍存在的问题是大多数研究识别的车牌图像分辨率高较为清晰，在复杂环境下(角度倾斜、不良天气条件、光照不均、模糊)，其方法对车牌的检测与识别率将会大大降低(W. Wang et al 2020)。并且目前开源的多颜色中国车牌数据集较少，手动收集与标注的成本十分昂贵。使用深度学习对目标进行检测是学术界目前研究火热的领域，并且针对数据集较为缺失的情况现也有了一些解决思路。

因此，本研究基于深度学习技术，使用目标检测的方法检测到复杂环境条件(如光照变化、角度倾斜、视野模糊)下的车牌目标后进行车牌区域提取，对提取后的图像进行矫正再识别出车牌字符；对于车牌图像数据集手动收集标注困难且昂贵的问题，本研究将使用图像拼接方法生成一些扩充数据集来帮助解决；最后将车牌检测识别算法进行封装，

转化为一项实际的应用——设计一套包含管理者前端、用户前端、云端数据库的智能停车系统。综上，本研究对于多种复杂环境条件下提高中国车牌检测识别率，从而促进城市的智慧交通管理具有重要的研究意义。

1.2 国内外研究现状及发展趋势

1.2.1 传统车牌提取方法

（1）基于边界或边缘信息的车牌提取

由于车牌通常为一个长宽比固定的矩形形状，因此可以搜索图像中所有可能的矩形区域来寻找车牌区域。由于车牌与车身之间存在着明显的轮廓过渡，车牌的边界在图像中可以由边缘来表示。水平边缘检测与垂直边缘检测进行共同检测时，车牌区域可被视为一个完整的矩形，这样就能检测出车牌位置的候选区域，因此边缘检测方法常用来寻找这些矩形。H. Bai 等(2004)使用了边缘提取和形态学相结合的方法来提取可能是车牌的矩形，再进行层次分类选出优先级最高的候选区域，在 9825 幅灰度和彩色图像上，能达到 99.6%的检测精度。C. Busch 等(1998)应用了垂直和水平的 Sobel 算子来分割车牌区域，因为它比其他滤波器例如 Laplace 算子，对噪声更加不敏感的特性，并且 Sobel 算子还可以平滑输入的边缘，因此在交通视频流中能取到一定的分割效果。

经过后续的研究发现，车牌垂直边缘的大小被认为是一种鲁棒的提取特征，而使用水平边缘只会导致汽车保险杠引起的误差(S. Wang et al. 2003)。M. Sarfraz 等 (2003) 对图像特征中的垂直边缘进行匹配，得到一些候选矩形，根据先验知识选择与车牌长宽比相同的矩形作为候选矩形，在不同光照条件下，该方法的检测效果为 96.2%。D. Zheng 等(2003)发现，如果提取垂直边缘并去除背景边缘即可从边缘图像中提取出车牌区域。1165 幅图像的检测率在 99%左右，一幅 384×288 的图像总处理时间为 47.9 ms。A. Al-Ghaili 等(2008)构建了一种车牌垂直边缘的检测算法，这种算法比传统的算子例如 Sobel 算子处理速度更快、精度更高。H. Lee 等(2004)提出了基于特征块的方法，车牌的位置区域就是边缘响应强度高的特征块区域，因为块处理减少了对车牌边界处边缘特征的依赖性，因此应用于车牌边界不清楚的图像有较好的效果，准确率为 92.5%。V. Kamat 等(1995)使用了 Hough 变换实现了对车牌边界线的检测，其优点是可以检测到有一定倾角的直线，缺点是 Hough 变换的计算比较消耗内存和时间。边缘定位算法具有准确率高的有点，但是普遍存在的问题有以下几点问题：

(1) 受噪声影响，区域特征容易发生黏连导致形态学分析失效，从而提高误检率；

(2) 仅仅判断图像中垂直和水平的直线形成的矩形是否为车牌区域，会忽略车牌视角倾斜的情况；

(3) 受光照条件影响较大，整体鲁棒性不足。

（2）基于图像全局信息的车牌提取

连通区域分析(Connected Component Analysis, CCA)是二值图像处理过程中的重要分割技术，使用该方法扫描二值图像，并根据像素之间的连通性将符合条件的像素标记为一类。再结合上车牌面积和长宽比等先验知识，可用来进行车牌提取(N. Bellas et al. 2006)。H. Wu 等(2006)将 CCA 应用于低分辨率车牌分割之中，在 4 小时以上视频进行测试的结果表明，其正确率可达到 96.62%。M. Chacon 等(2003)将车牌图像转化为二值图像，应用边缘轮廓检测算法检测连通对象。与车牌具有相似长宽比等几何特征的矩形区域被认为是车牌区域的候选区域，其准确率为 85%，但是在图像质量较差的情况下，该算法可能会因为轮廓失真，导致分割失败。K. Miyamoto 等(1991)使用二维交叉相关来寻找车牌区域。与预先存储的车牌模板进行二维交叉关联，对整个图像进行定位寻找车牌的候选区域，用与车牌模板的相关性方法来提取车牌与车牌在图像中的位置无关，因此适合多车牌区域的提取，但是二维交叉相关是很耗时的，在车牌位置跟踪时不适用。对图像全局信息进行检索并提取出车牌位置为多车牌信息提取提供了很好的借鉴，但其存在受图像质量影响较大、计算量大的缺点。

（3）基于纹理特征的车牌提取

基于纹理特征的方法依赖于车牌中字符的存在，因为字符颜色与车牌背景颜色之间的灰度级别发生显著变化产生的颜色过渡，会导致车牌区域为边缘密度较高的区域。

R. Zunino 等(2000)提出了基于矢量量化(Vector Quantization, VQ)的方法定位图像中的文本信息，使用 VQ 方法增强图像局部区域，在局部区域得到可用的提示信息，可以充分利用提示信息来提高定位性能。所述方法通过相应地训练编码和定位模块来实现车牌的定位，实验结果表明，对于不同质量的图像，检测率达到 98%，但是易受图像中出现的其他文本信息的干扰。C. Anagnostopoulos 等(2006)提出了滑动同心窗口(Sliding Concentric Window, SCW)的方法，利用了车牌位置纹理的不规则性，得出纹理突变的位置是车牌的候选位置，使用 1334 幅图像进行测试的结果表明，其分割的准确率为 96.5%。K. Deb 等(2009)提出了一种基于 SCW 和直方图的车牌检测方法，在不同光照和天气条件下的检测成功率为 82.5%。

Gabor 滤波器是图像纹理分析的主要方法之一，该滤波器的优点是可以在不限方向和尺度上对纹理进行分析(H. Caner et al. 2008)。当应用于以固定和角度获取的车牌图像时，F. Kahraman(2003)等使用 Gabor 滤波器进行车牌的定位，成功率是 98%，但该算法的计算较为费时。C. Hsieh 等(2005)采用基于小波变换(Wavelet Transform, WT)的车牌提取方法。在 WT 方法中，有四个子带。其中子带 High-Low 包含垂直边缘信息，Low-High 包含水平边缘信息。水平边缘的最大变化是通过扫描 Low-High 图像来确定的，并通过参考线来进行识别。垂直边缘水平投影到这条线以下，以根据最大投影确定位置，检测正确率为 92.4%。H. Zhang 等(2006)首先训练了一个使用全局统计特征的分类器，超过

70%的背景区域可以被排除，然后利用 AdaBoost 学习算法根据选定的局部 Haar-like 特征建立其他分类器，结合上述两者，得到一个级联分类器，该系统具有低复杂性，不受车牌的亮度、颜色、大小和位置影响的特点，最终检测成功率为 93.5%。基于纹理的方法具有即使车牌边缘不明显或有较大形变时也能检测出车牌的优点，但是该计算方法在图像有较多边缘细节时的计算较为复杂。

（4）基于颜色特征的车牌提取

车牌颜色信息往往较为固定，因此基于颜色特征的车牌提取方法就是从图像中定位它们的颜色来实现提取，其本质特征是车牌中的底板颜色和子符颜色的组合是唯一的。E. Lee 等(2005)将 RGB 图像转换成 HSL 颜色模型之后，使用神经网络来对每个像素的颜色进行分类，对神经网络推理出的与车牌颜色相同的候选区域进行垂直和水平投影，以确定最高颜色密度区域即车牌区域。S. Chang 等(2004)，由于中国台湾地区车牌仅使用了四种颜色(绿色、黑色、白色和红色)，存在三种颜色组合，因此通过神经网络训练了一个彩色边缘检测器仅聚焦于三种边缘(黑色-白色、红色-白色和绿色-白色)，使用了从不同场景和不同条件下拍摄的 1088 幅图像的测试结果表明，车牌定位准确率为 97.9%。利用颜色特征提取车牌具有能够检测到倾斜和变形车牌的优点，但其主要不足如下：

(1) 在不同的照明条件下，RGB 颜色空间对色彩的不够准确；
(2) HLS 颜色空间对噪声较为敏感；
(3) 使用彩色投影的方法有可能会检测错误，特别是当图像的某些部分具有相同的车牌颜色时，如交通警示牌、广告牌、车身等会导致误识别率增加。

（5）基于字符特征的车牌提取

基于字符特征的车牌提取方法是通过检查图像中是否存在包含于车牌字符库中的特定字符从而实现车牌区域的定位。如果找到特定字符，则提取它们的区域作为车牌区域。J. Matas 等(2005)试图找到图像中可能存在字符的区域，而不是直接使用车牌的特征属性。使用神经网络对区域进行判断分类，如果找到了字符状区域的组合，则确定为车牌区域的候选区域，经测试检测率为 98%。F. Alegria 等(2006)将与字符具有相同纵横比并且超过 30 个像素的二值图像区域进行标记。使用 Hough 变换来检测被标记图像上下两条直线，如果两条直线在一定倾斜角度内可以被认为是平行，且它们中间的区域可以确认为车牌的候选区域。W. Ho 等(2009)使用 36 个 AdaBoost 分类器作为第一级分类器获得潜在的字符区域，然后将其传递给由基于尺度不变特征变换描述子(Scale-Invariant Feature Transform, SIFT)的支持向量机(Support Vector Machine, SVM)构建的第二级分类器以进一步消除非字符区域，最终检测率为 90.1%。这些从二值图像中检测字符的方法非常耗时，并且当图像中存在其他文本时，容易出现误检测的情况。

1.2.2 基于深度学习的车牌提取方法

深度学习允许由多个处理层组成的计算模型来学习具有多个抽象级别的数据表示，通过反向传播算法，深度学习网络能从大型数据集中的学习到统一的参数计算结构，并在前一个处理层的基础上改变内部参数获得一个能表示前者的全新处理层(Y. LeCun et al. 2015)。

目前深度学习方法在车牌提取中的应用是车牌识别中的研究热点，随着深度学习的发展，学者们不断发掘新的基于深度学习的车牌提取的方法，并取得了明显优于传统方法的良好结果。H. Li 等(2018)使用级联框架来提取车牌，以滑动窗口的方式在图像时采用卷积神经网络(Convolutional Neural Networks, CNN)分类器来检测文本并生成文本显著性图，通过使用游程平滑算法和 CCA 在每个尺度上独立生成候选包围盒。然后，生成的盒子通过几何约束进行过滤，并通过车牌的边缘特征进行细化。最后，使用辨别车牌和非车牌的 CNN 分类器来验证剩余的包围盒，模型的准确率达到 97%左右，召回率达到 0.95 以上。L. Xie 等(2018)提出了一种基于 CNN 的多方向车牌网络结构——MD-YOLO。类似于 YOLO，每幅输入图像被分成规则的 7×7 网格单元，并且汽车牌照中心所在的单元被用于检测牌照，并且为每个单元预测 2 个包围盒和置信度得分。与 YOLO 不同的是，MD-YOLO 引入了角度信息，并指导模型回归和确定给定汽车牌照图像的旋转角度，提出了角度偏差惩罚因子来逼近预测值与标签值的交比，检测模型在 GPU GTX980 上处理时间为 5ms，准确率可达 99%以上。由于大多数当前的车牌检测和识别方法都是利用作者制作的数据集进行测试，数据量往往不高，通常不具代表性。因此，Z. Xu 等(2018)收集并开源了一个拥有 30 余万幅带有中国汽车牌照的中国车牌数据集 CCPD，并且提出了一种车牌提取端到端训练的网络结构 RPnet，在 CCPD 上进行评估，准确率最高可达 98.5%。

1.2.3 车牌字符识别方法研究现状

字符识别是将车牌图像中的字符图案转换成字符串。字符识别的目标是准确地输出所有字符，包括每个字符的正确分类，并防止丢失和冗余字符的发生。与文本信息提取不同，车牌字符识别在字体和字号上不会有太大的差异。

车牌识别的整体流程可以简述为将前一阶段提取的车牌作为输入，输出字符序列。在传统的车牌识别过程中，字符分割对车牌识别的精度有较大影响。如果没有正确分割，即使有一个强大的识别器可以处理各种缩放、不同字体规格和字体旋转，车牌也可能被错误识别。因此，一些研究人员考虑如何有效地分割。还有一些学者为了避免字符分割困难(Y. Liu 2018)，结合具有 CNN 特征的区域 (Regions with CNN features, RCNN)，提出了一系列的序列标注方法。根据字符识别是否分段，可以将识别算法分为基于分段的识别算法和无分段的识别算法。

（1）基于字符分割的识别方法

在分割方面，传统算法大致可以分为五类：连通分量分析、投影分析、字符先验知识、字符轮廓及其组合。随着深度学习的发展，目标检测模型也被用来提取字符。车牌分割成单个字符后由各类识别算法实现字符识别。

Y. Liu 等(2018)实现了连通分量分析和项目分析相结合的车牌字符分割。设计了两个简单的递归神经网络——空间 CNN 神经网络和递归神经网络，一个用于汉字识别，另一个用于数字和字母识别，对于过度曝光的车牌，无法直接进行二值化处理，因此提出了一种灰度转换算法，其基本思想是抑制像素点的灰度，提高图像的对比度。最后，使用广度优先搜索算法获取连通分量，并利用连通分量确定是否存在缺失或冗余字符。基于 2189 幅图像，分割率和识别率分别为 96.58%和 98.09%。M. Dong 等(2017)提出了一种由并行空间转换网络和共享八个识别器组成的创新结构。校正后的车牌被输入到七个平行的无监督字符组，每个字符组隐含地执行字符分割。最后，七个识别器识别每个段，第一个识别器为汉字单独训练，其余六个识别器共享权重。该模型的最终识别准确率为 89.05%。分割后再识别的算法虽然能取到一定的效果，但是分割时又会受到光照和噪声等影响，因此分割效果对识别率造成了较大影响。

（2）基于无分段字符的识别方法

无分段算法的核心思想是将车牌字符的识别问题转化为字符序列的标注问题。与基于字符分割的算法不同，无分割算法利用输入车牌图像的全局信息。学者们提出最早实现序列标注的框架是隐性马尔可夫模型(Hidden Markov Model, HMM)及其混合方法 HMM-RNN，但其使用的预分割和后处理操作严重限制了实用性。因此，A. Graves 等(2006)提出了连接时序分类(Connectionist Temporal Classification, CTC)模型来解决无分割序列的标注问题。

H. Li 等(2017)利用双向递归神经网络(Recurrent Neural Networks, RNNs)的 CTC 损失来标记序列特征，并针对传统 RNN 训练中突然出现的梯度异常的现象，使用了长短期记忆网络，最后，采用 CTC 层进行序列解码，在 AOLP 数据集上，平均识别率为 91.83%，执行速度为 400ms。J. Zhuang 等(2018)将 YOLO-VOC 网络用于车牌分割和字符识别，实现巴西车牌的识别，该模型能够正确分割 99%的字符，识别率为 93%。J. Wang 等(2018)利用双向循环神经网络(Bidirectional RNN, BRNN)和 CTC 对车牌字符进行识别，利用空间变换网络对倾斜变形的车牌进行调整，CNN 提取校正后的车牌字符序列特征。生成的特征被集成为 BRNN 的输入，最后序列标签由 CTC 实现，经测试识别率为 96.62%，处理时间为 17.53ms。

1.3 研究思路

鉴于上述研究现状，目前车牌检测识别的研究仍存在复杂环境条件下(如光照变化、角度倾斜、视野模糊)检测与识别困难的问题，深度学习对复杂环境条件下的目标检测有着比传统方法更好的性能，因此本研究拟采用深度学习方法尝试在复杂环境条件下更好地检测识别出车牌。针对中国车牌的研究，虽然有公开可评估的数据集 CCPD，但使用该数据集进行车牌的研究仍较少。此外较少学者能关注到中国蓝牌、黄牌、绿牌三种颜色车牌的共同研究，这是因为存在收集数据集困难的问题。本研究也将在 CCPD 上评估方法的性能，并且研究一项可行的中国车牌数据集扩充方法。

针对上述中国车牌的检测识别中仍存在的问题和解决思路，本研究课题的技术路线如图 1-1 所示，车牌检测与识别系统具体流程为：首先在基于 PyQt5 的管理者的前端输入一幅车辆图像，若存在车牌则通过基于 YOLOV5+Landmark 的车牌检测模块将其位置和四个角点推理出来，车牌矫正模块通过对车牌四个角点进行透视变换将车牌矫正平整后送入至车牌识别模块，扩充数据集用于车牌检测网络与车牌识别网络的训练，基于 CRNN+CTC 的车牌识别模块识别出车辆的车牌号码，由云端数据库记录下该车辆入库记录，从云端数据库读出相应数据与管理者前端和用户前端显示。在搭建整个应用性系统的同时，要与其他车牌检测与识别方法进行比较，证明该系统对于复杂环境下的车牌检测与识别有较好的准确性与鲁棒性。还要注意该系统部署的实用性细节，要有较小的参数计算量以及较快的运行效率。

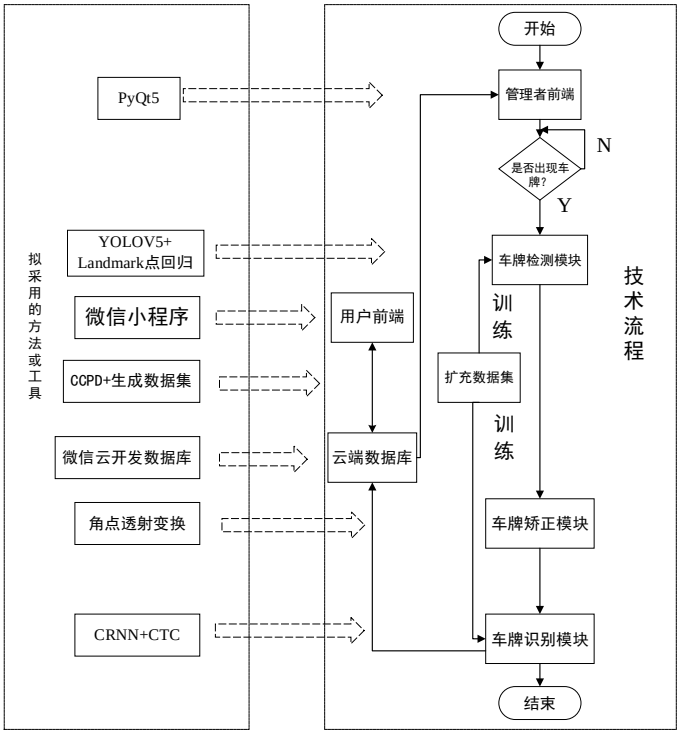


图 1-1 技术路线

Fig. 1-1 Technology route

本文研究内容详述如下：

(1) 了解国内外对车牌这种复杂小对象检测研究的最新进展，根据研究过程中对算法的具体要求，完成深度学习对复杂环境下车牌检测算法的选择。

本项研究要求掌握数字图像处理以及深度学习相关的理论，还需要具备良好的论文阅读能力，学习到国内外车牌检测与识别技术的最新研究成果。本研究拟采用 YOLOV5 深度学习网络作为车牌检测方法，同时在 CCPD 数据集上与其他深度学习检测方法进行对比。

(2) 采用基于四个点的透视变换，实现倾斜车牌的矫正。

修改 YOLOV5 模型，加入 landmark 关键点回归以获得车牌的四个角点。根据四个角点进行透视变换，以矫正车牌图像用于识别。

(3) 采用深度学习方法对车牌进行识别，分析不同环境下的识别效果。

识别方法拟采用 CRNN+CTC 网络进行识别，最后在模糊、不同光照变化、不同远近距离、具有一定倾斜程度、不同天气等不同条件下对车牌进行识别，分析识别效果。

(4) 研究车牌数据集扩充方法，并分析其使用效果。

由于 CCPD 数据集只有蓝牌和绿牌数据集，因此研究生成数据集是否可代替真实数据集用于网络训练，提升网络的检测能力与识别能力。收集中国车牌的背板图像以及对应的字符样式，根据 GA 36-2018 机动车号码牌标准来进行图像的拼接，再加上光照变化和高斯噪声来模拟现实的车牌视觉表现。

(5) 将实现好的算法进行封装，设计一套智能停车系统。

智能停车系统包含基于 PyQt5 设计的管理者前端、基于微信小程序设计的用户前端、基于微信云开发数据库设计的后端。要求管理者前端能看到车牌检测与识别情况以及目前的停车情况，用户前端能够登录、注册、查询车辆停车状况，数据库后端能够存储用户注册后收录的个人信息、车辆信息、车辆入库信息。

1.4 论文框架结构

基于深度学习的复杂环境下车牌检测与识别方法研究论文框架结构如图 1-2 所示：

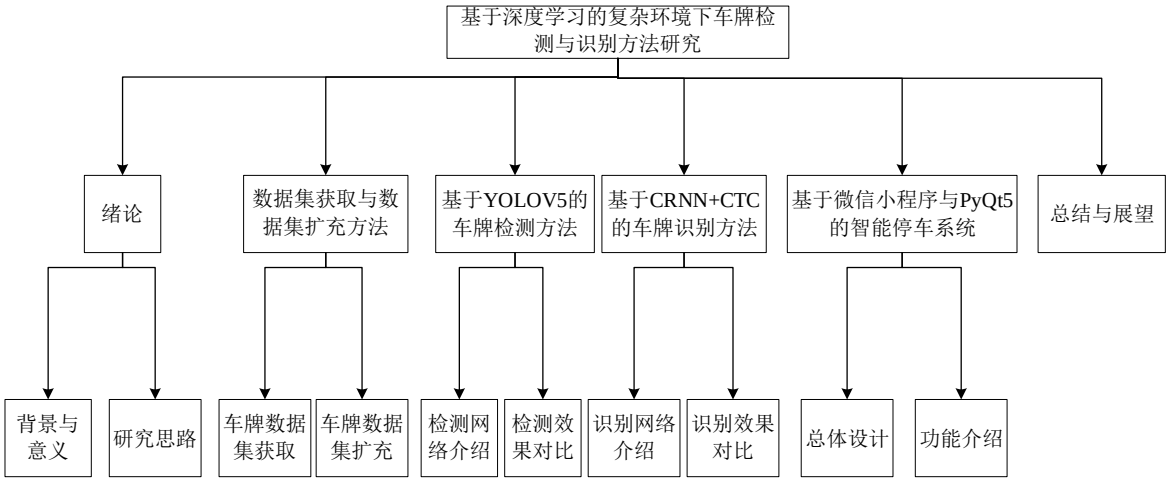


图 1-2 论文框架结构

Fig. 1-2 Paper structure

本课题论文分为七部分，分别为：绪论、数据集获取与数据集扩充方法、基于 YOLOV5 的车牌检测方法、基于 CRNN+CTC 的车牌识别方法、基于微信小程序的用户前端设计、基于 PyQt5 的管理者前端设计、总结与展望。其中第二部分介绍了本文主要使用的数据集 CCPD，并介绍了生成数据集的拼接方法、还原现实处理手段以及基于掩膜的粘贴方法。接着介绍了本研究使用的车牌检测网络，用 CCPD 数据集与生成数据集都进行了训练验证效果并与其他学者的实验结果进行对比，然后介绍了本研究使用的车牌识别网络，用 CCPD 数据集与生成数据集也分别进行了网络训练，将验证效果与其他学者的实验结果进行对比，验证本研究的方法对于检测与识别复杂环境下的车牌有着良好的性能，并且生成数据集代替真实数据集进行训练能取得一定的检测与识别效果。最后将研究成果进行封装成了基于微信小程序的用户前端与基于 PyQt5 的管理者前端，并对其设计方法与使用效果进行了展示。

第二章 车牌数据集获取与数据集扩充方法

车牌数据集的获取是车牌检测与识别的首要工作，不仅可以用于网络训练矫正模型参数权重，使得网络获得对车牌的检测和识别的能力，还可以用于测试研究结果的准确性与有效性。目前常用的数据扩充方法有旋转、裁剪、镜像、模糊、调节饱和度与色度等方法，但对于车牌检测与识别项目来说，这些方法并不完全适用，因此本研究项目将提出一项最适用与车牌检测与识别项目的数据集扩充方法。本章主要介绍车牌数据集的获取、车牌数据集的扩充方法，为以后章节提供基础且可靠的使用素材。

2.1 车牌数据集的获取

目前市面上存在的开源中国车牌数据集较少，大多数学者研究中国车牌的检测与识别都是靠自己收集与标注车牌数据集，形成一个数量、种类较少且通常不具有代表性的私有数据集库。在车牌识别方面，由于不同区域都有不同的车牌省份号码，要手工收集到省份种类繁多的车牌信息是一件既费时又费力的事情。本研究项目是针对于复杂环境下的车牌检测与识别研究，使用的数据集需要有复杂环境(角度倾斜、不良天气条件、光照不均、模糊)下包含车牌的车辆图像，且车牌的省份信息应尽可能地多，这样的数据集给予车牌检测与识别更大的挑战以获得更好的性能。

CCPD 数据集是中国科学技术大学学者收集的中国城市停车数据集，从安徽省合肥市的路边停车场收集而来。以图 2-1 为例，画出其包围盒位置和四个角点位置阐述 CCPD 数据集图像的标注规则， CCPD 每一幅图像都具有关于车牌的完备注释：(1)车辆编号；(2)水平倾斜程度和垂直倾斜程度；(3)车牌区域包围盒位置；(4)车牌四个角点位置；(5)车牌的字符信息；(6)亮度、模糊度等其他备注信息。其中车牌号的省份与字符信息分别按照表 2-1 与表 2-2 的命名规则对图像名称进行了标注。

图 2-1 CCPD 数据集标注规则

Fig.2-1 CCPD dataset annotation rules

表 2-1 车牌省份命名规则

Table 2-1 Naming rules of provinces on license plate

省份	编号	省份	编号	省份	编号	省份	编号
皖	0	吉	8	豫	16	云	24
沪	1	黑	9	鄂	17	藏	25
津	2	苏	10	湘	18	陕	26
渝	3	浙	11	粤	19	甘	27
冀	4	京	12	桂	20	青	28
晋	5	闵	13	琼	21	宁	29
蒙	6	赣	14	川	22	新	30
辽	7	鲁	15	贵	23		

表 2-2 车牌字符命名规则

Table 2-2 Naming rules of characters on license plate

字符	编号	字符	编号	字符	编号	字符	编号
A	0	K	9	U	18	3	27
B	1	L	10	V	19	4	28
C	2	M	11	W	20	5	29
D	3	N	12	X	21	6	30
E	4	P	13	Y	22	7	31
F	5	Q	14	Z	23	8	32
G	6	R	15	0	24	9	33
H	7	S	16	1	25		
J	8	T	17	2	26		

CCPD 数据集的种类分布如图 2-2 所示，其中每一类的图像如图 2-3 所示。一共包含超过二十九万幅分辨率统一为 720×1160 的车辆图像。其中 CCPD-Base 包含二十万幅拍摄正常的车牌，CCPD-FN 包含两万幅车牌距离摄像头相当远或相当近的图像，CCPD-DB 包含两万幅车牌光线较暗或较亮的图像，CCPD-Green 包含一万幅带有车牌的新能源汽车图像，CCPD-Rotate 包含一万幅车牌水平倾斜角度 20-25°、垂直倾斜角度-10-10°的图像，CCPD-Tilt 包含一万幅车牌水平倾斜 15-45°、垂直倾斜 15-45°的图像，CCPD-Weather 包含一万幅处于雨天、雪天以及雾天的包含车牌的车辆图像，CCPD-Challenge 包含一万幅具有多种复杂条件下最具有挑战性的包含车牌的车辆图像，CCPD-Blur 包含五千幅由于相机抖动导致模糊的包含车牌车辆图像，CCPD-NP 包含 5000 幅没有车牌的车辆图像。

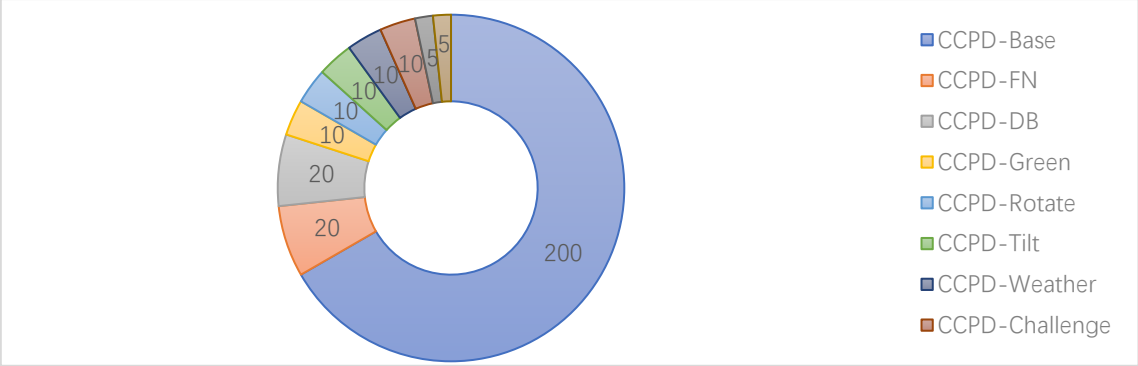


图 2-2 CCPD 数据集种类分布

Fig.2-2 Distribution of CCPD dataset

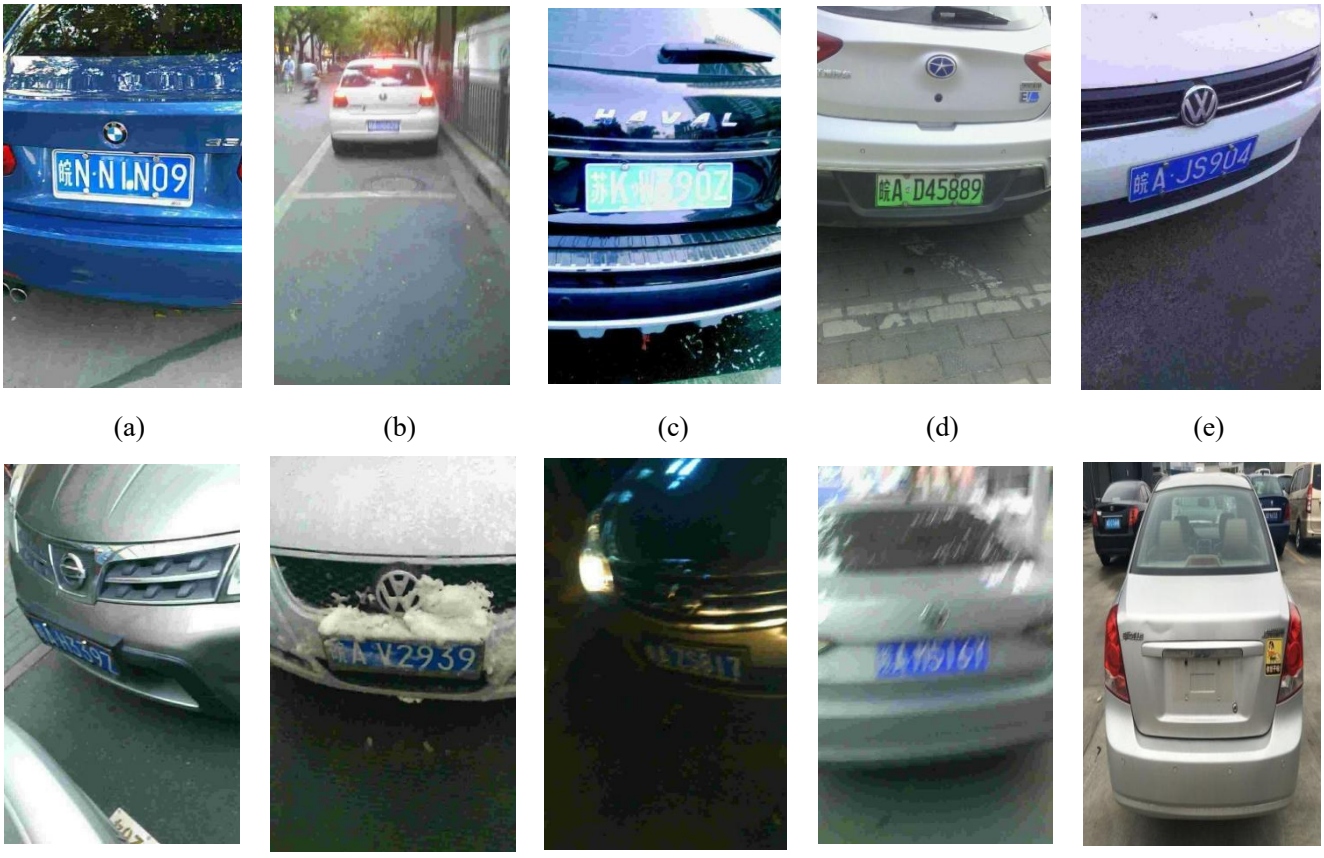


图 2-3 CCPD 数据集车辆图像

Fig.2-3 Images in CCPD dataset

在车牌检测方面，CCPD 数据集有海量的蓝色车牌与绿色车牌的车辆图像，足以形成完备的数据集库来训练车牌检测网络。但是美中不足的是缺少了黄色车牌的车辆图像。此外，通过检索 CCPD 蓝色车牌数据集与绿色车牌数据集中车牌省份信息，各个省份的图像数量如表 2-3、2-4 所示，发现 CCPD 数据集蓝色车牌与绿色车牌上均存在省份不均的现象，因为是在安徽省进行收集的，因此“皖”车牌占了数据集中绝大多数，对于车

牌的识别方面存在不利影响。

表 2-3 蓝色车牌省份数量

Table 2-3 Number of provinces on blue license plate

省份	数量	省份	数量	省份	数量	省份	数量
皖	279794	吉	26	豫	1361	云	142
沪	1542	黑	134	鄂	1124	藏	1
津	407	苏	7328	湘	721	陕	593
渝	381	浙	3418	粤	1291	甘	95
冀	639	京	1076	桂	38	青	14
晋	421	闵	845	琼	13	宁	8
蒙	49	赣	780	川	425	新	35
辽	312	鲁	1107	贵	53		

表 2-4 绿色车牌省份数量

Table 2-4 Number of provinces on green license plate

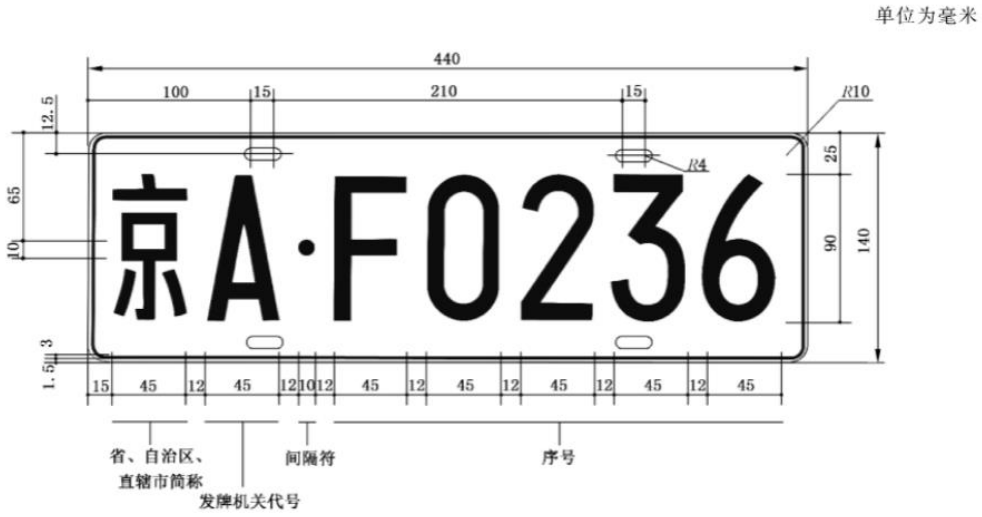
省份	数量	省份	数量	省份	数量	省份	数量
皖	11373	吉	0	豫	1	云	0
沪	151	黑	0	鄂	2	藏	0
津	2	苏	139	湘	3	陕	0
渝	0	浙	25	粤	22	甘	0
冀	0	京	2	桂	0	青	0
晋	0	闵	0	琼	0	宁	1
蒙	0	赣	0	川	0	新	0
辽	0	鲁	5	贵	5		

2.2 扩充数据集方法

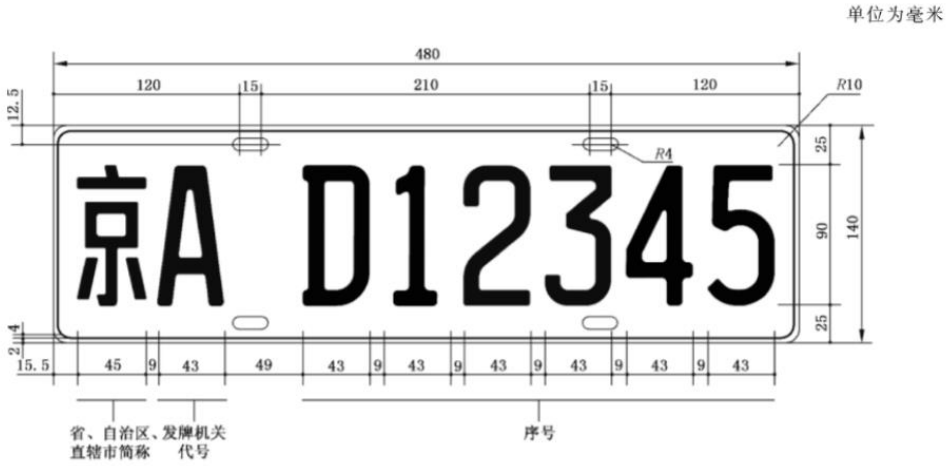
针对 2.1 节提到的 CCPD 数据集存在着省份不均与缺少黄色车牌的情况，本节将介绍一项针对中国车牌数据集扩充方法。

2.2.1 车牌标准与图像素材

为了生成一定的车牌替代图像，本研究项目根据最新的 GA 36-2018 机动车号码牌标准，按照如图 2-4 所示的样式，收集了如图 2-5 所示的车牌底板与车牌字符图像，为后续两者按照 GA 36-2018 机动车号码牌标准进行拼接打下基础。



(a)

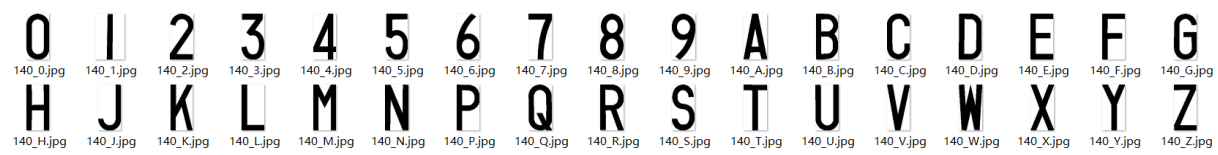


(b)

图 2-4 GA 36-2018 机动车号码牌标准中蓝、黄、绿车牌标准

Fig.2-4 GA 36-2018 standard specification for three kinds of license plate

(a) (b) (c)



(d)

(e)

图 2-5 车牌拼接图像素材

Fig.2-5 Material for image mosaicking

2.2.2 车牌图像素材拼接方法

根据 GA 36-2018 机动车号码牌标准可以得知，蓝色和黄色车牌的大小为 440mm×140mm，新能源绿色车牌的大小为 480mm×140mm，因此可以将蓝色、黄色车牌底板图像分辨率调为 440px×140px，新能源绿色车牌调为 480px×140px。此外，蓝色车牌的款式为蓝底白字，与其不同的是，黄色车牌与新能源绿色车牌的款式分别为黄底黑字与绿底黑字。新能源车牌的第二位字符仅可为“D”、“F”，分别代表纯电动新能源汽车与非纯电动新能源汽车。根据上述实施细节标准将字符之间的距离(mm)转变为像素距离(pixels)，使用 python 语言编写程序读取底板图像、省份字体、字符可以随机车牌号或者指定车牌号批量拼接出如图 2-6 所示的蓝牌、黄牌、新能源绿牌的基础生成数据集。

(a) (b) (c)

图 2-6 基础生成数据集

Fig.2-6 Basic generating yellow license plate

2.2.3 生成数据集图像还原现实处理

图 2-6 的基础生成数据集图像质量过高，省份、字符信息过于清晰，距离还原真实环境下的车牌图像还有些差距，因此需采用一定的数字图像处理方法进行处理，尽可能还原现实车牌表现。

（1）字符腐蚀与膨胀

在不同的光照条件与视角变换下或者受到噪声的影响，车牌字符的大小在肉眼的视觉表现可能有所缩小或者变大，因此在生成基础生成数据集前，对车牌的字符进行一定的处理还原视觉表现是有必要的。

数学形态学是图像处理中常用的基本处理方法，它的基本思想是把图像像素看作一个集合，然后设计一种“探针”也可称为结构元素，对图像像素集合内部进行检索找到一些能放下其位置的像素点集合，对适合放入结构元素的图像像素集合内部的位置进行标记，这样的步骤形成了多种形态学运算方法。其中腐蚀和膨胀运算具有缩小与放大图

像像素集合的作用，可以很好地对车牌字符进行缩小与放大(安静 2016)。

如式 2-1 所示，图像像素集合 A 被结构元素“探针”B 进行探测，找到图像像素集合 A 中能够容下“探针”B 的区域即α。这一过程能够消除图像像素集合的边界点，使得图像像素集合向内部收缩。

$$A\Theta B = \{\alpha: B + \alpha \subset A\} \tag{2-1}$$

如式 2-2 所示，该式也被称为明夫斯基和形式。图像像素集合 A 被结构元素“探针”B 进行探测，探针与图像像素集合进行叠加。其作用是对二值化物体边界点扩充，将与物体接触的所有背景点合并到该物体中，使边界向外扩张。

$$A \oplus B = \cup \{A + b | b \in B\} \tag{2-2}$$

在对车牌的每个字符图像按照 50%的概率，随机对每个字符进行膨胀或腐蚀操作，最终结果如图 2-7 所示，车牌字符都随机经过了腐蚀或者膨胀处理。

(a) (b) (c)

图 2-7 经腐蚀与膨胀后的生成数据集

Fig.2-7 Generating dataset with corroding and dilating

（2）加入光照变化与噪声

通常车辆所处的环境光照大小都是不同的，很大程度上影响到了车牌区域的亮度，因此施加一些光照变化可以很好地还原现实场景中的光照情况。HSV 颜色空间很好地还原了人类视觉系统对现实中色彩的感知，用色调、饱和度和亮度来表示视觉中的颜色。因此如式(2-3)-(2-6)所示，将车牌图像设为g，将g转到 HSV 颜色空间后，对g的色调、饱和度、亮度，三个颜色通道通过随机数α进行一定的随机化处理。但是要保证在色调分量上要维持一定的取值范围，不能让车牌颜色发生很大的变化。

$$HSV(g[:,:,0]) = HSV(g[:,:,0]) * (0.8 + \alpha * 0.2) \tag{2-3}$$

$$HSV(g[:,:,1]) = HSV(g[:,:,1]) * (0.3 + \alpha * 0.7) \tag{2-4}$$

$$HSV(g[:,:,2]) = HSV(g[:,:,2]) * (0.2 + \alpha * 0.8) \tag{2-5}$$

$$\alpha = \{x | 0 < x < 1\} \tag{2-6}$$

最终的处理结果如图 2-8 所示，车牌色彩不再像之前这么鲜亮，反而多了一定程度的褪色、暗沉的特点。

(a) (b) (c)

图 2-8 经光照变化处理后的生成数据集

Fig.2-8 Generating dataset with illumination change

经过 HSV 颜色空间处理后，亮度和色彩的变化大而不自然，且图像质量还是较高，没有较多的噪声。通常车牌图像在一幅车辆图像中只占一小片区域，在摄像机像素的影响下，受到噪声的干扰较大，因此还需对生成数据集加入一定的噪声。如式(2-7)-(2-10)所示，在 RGB 颜色空间中对图像进行噪声计算。

$$d_{f\beta} = 255 - \max\left(g[:,:,\beta]\right) \quad \beta = \{0,1,2\} \tag{2-7}$$

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{(-\frac{x^2}{2})} \tag{2-8}$$

$$noise = \frac{f(x) - \min\left(f(x)\right)}{\max\left(f(x)\right) - \min\left(f(x)\right)} \tag{2-9}$$

$$d_s = noise * d_{f\beta} \tag{2-10}$$

$$g[:,:,\beta] = d_s + noise \qquad \beta = \{0,1,2\} \tag{2-11}$$

将车牌图像设为g，将g在 RGB 空间三个分量中的最大值进行各自进行反转处理得到 $d_{f\beta}$，定义$f(x)$属于正态分布，$noise$是$f(x)$进行计算后的噪声变量，d_s是经过修正后的噪声变量。最终计算得到的噪声车牌图像如图 2-9 所示，生成车牌图片多了不少噪点，且暗沉和褪色的程度得到了一定缓和，图像质量得到了降低。

(a) (b) (c)

图 2-9 经噪声处理后的生成数据集

Fig.2-9 Generating dataset with noise

通常由于在被拍摄时还在运动过程中，极易在拍摄时出现运动模糊的情况，因此为了还原这一情况，本研究适当在车牌图像上加入运动模糊。物体在按照一定的速度、一定的方向与相机做着相对运动，这是运动模糊的出现的直接原因，在按下快门时的曝光时间具有一个间隔，因此会出现拍摄物体在某个方向上叠加的情况，运动模糊可由式(2-11)所示。

$$g_0(x,y) = \int_0^T g[x - x_0(t), y - y_0(t)]dt \tag{2-12}$$

式中，$g_0(x,y)$为模糊后的图像，假设图像$g(x,y)$在平面内运动，令$x_0(t)$和$y_0(t)$二者为在 x 和 y 方向上进行运动的分量，T 表示在此方向上相机瞬间曝光时间间隔。经运动模糊处理后的车牌图像如图 2-10 所示，生成车牌图像被随机赋予了一定方向和长度的

模糊。

(a) (b) (c)

图 2-10 经运动模糊处理后的生成数据集

Fig.2-10 Generating dataset with motion blurring

2.2.4 车牌图像替换

为了在车牌检测训练过程中，有更多的生成数据集可以加入数据集中进行补充，就要给生成数据集更多的背景信息，因此本节的思路是使用透视变换将生成数据集进行校正，再使用透视变换掩膜的方法将生成数据集图像“贴”在原车牌的位置。

2.2.4.1 透视变换

透视变换本质是基于目标点、像点、透视中心点三点共线的条件，通过透视旋转定理使承影面(透视面)绕迹线(透视轴)旋转某一角度，破坏原有的光学投影柱所呈现的视觉效果，仍能保持承影面上投影的几何图形原有信息不变的变换。换言之，就是将一个初始平面通过一个投影矩阵投影到指定平面上。透视变换可表示为式(2-12)所示,X、Y、Z 表示的是目标点，x、y、z 表示的是原点，因为是原图像是在二维平面上进行变换，因此 z=1。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2-13}$$

其中，透视变换矩阵可以表示如式(2-13)的 A 所示。

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{2-14}$$

如式(2-14)、(2-15)所示，又可做进一步的变换，$\left(X',Y',Z'\right)$表示的是变换后图像上的一个点的情况。其中a_{33}默认为 1。

$$\begin{cases} X' = \frac{X}{Z} \\ Y' = \frac{X}{Z} \\ Z' = \frac{Z}{Z} \end{cases} \tag{2-15}$$

$$\begin{cases} X' = \frac{a_{11}x+a_{12}y+a_{13}}{a_{31}x+a_{32}y+a_{33}} \\ Y' = \frac{a_{21}x+a_{22}y+a_{23}}{a_{31}x+a_{32}y+a_{33}} \\ Z' = 1 \end{cases} \tag{2-16}$$

为了解透视变换矩阵中的八个未知数，四个点可以得到八个方程用于求解，假设源点四个坐标为(x_0,y_0)、(x_1,y_1)、(x_2,y_2)、(x_3,y_3)，目标点四个坐标分别为(x'_0,y'_0)、(x'_1,y'_1)、x'_2,y'_2、(x'_3,y'_3)，最终可以如式(2-15)所示求出最后四个点解的形式。

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0x'_0 & -y_0x'_0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0y'_0 & -y_0y'_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{bmatrix} = \begin{bmatrix} x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{bmatrix} \tag{2-17}$$

在透视变换操作时，以车牌图像的四个角点为源点，然后 CCPD 数据集中随机抽取一张带有车牌的车辆图像，以图像中的车牌四个角点为目标点作透视变换，可得到透视变换矫正后的图像。

2.2.4.2 掩膜粘贴方法

为了让生成数据集可用于车牌的检测，因此本研究将生成数据集相片按照掩膜粘贴的方法粘到其他车辆相片上，以获得更多的背景信息，粘贴前的原图如图 2-11 与图 2-12 所示。原始车辆图像随机选取于 CCPD 的数据集中，车牌图像则随机选取于生成数据集中。

图 2-11 原始车辆图像

Fig.2-11 Original vehicle image

图 2-12 生成数据集中的素材

Fig.2-12 Materials of generating dataset

根据车辆图像中四个角点与车牌图像四个角点进行透视变换得到透视矩阵后，使用一张纯黑二值图作为掩膜，使用车牌图像透视变换后的矩阵一同进行透视变换，结果如

图 2-13 所示，三色车牌根据原始车辆上的车牌位置进行了矫正而掩膜也根据透视变换得到了一幅标记着车牌位置和无关背景的掩膜图像。

图 2-13 透视变换后的车牌图像与掩膜

Fig.2-13 Images and mask of license plate after perspective transformation

然后计算粘贴起始点的坐标，通过式子(2-18)与(2-19)，根据四个角点中 x 与 y 坐标中的最小值可以计算出起始点的坐标。以图 2-14 为例，根据起始点坐标与掩膜宽高度，对车辆图像中这个范围内的像素进行迭代访问，在掩膜中像素值为 0 的位置，将透视变换后车牌图像对应位置的像素三通道值赋给原始车辆图像中该位置的像素三通道值。处理结果如图 2-15 所示，生成数据集很好地覆盖了原来车牌的位置。

$$x_{start} = \min(x_0, x_1, x_2, x_3) \tag{2-18}$$

$$y_{start} = \min(y_0, y_1, y_2, y_3) \tag{2-19}$$

图 2-14 掩膜粘贴方法

Fig.2-14 Method of mask pasting

(a) (b) (c)

图 2-15 掩膜粘贴后的车辆图像

Fig.2-15 Vehicle images after mask pasting

2.3 本章小结

(1) 介绍了本研究所用到的数据集；

(2) 描述了生成数据集的拼接原理；

(3) 阐述了生成数据集如何进行还原现实处理以及基于透视变换的掩膜替换方法。

第三章 基于 YOLOV5 的车牌检测方法

在数据集扩充完毕后，可选择基于候选框的目标检测算法来实现对车牌的检测。常见的算法有 R-CNN 与 Fast-RCNN，2020 年新提出的 YOLOV5 算法因其高检测速度、高准确率的特点比 R-CNN 与 Fast-RCNN 更胜一筹。车牌检测是车牌号识别前的首要操作。车牌检测是一个深度学习网络学习和推理的过程，将车辆图像作为学习内容送入网络中，网络对其经卷积等操作后提取的特征进行学习以形成参数权重，然后使用训练得到的权重进行推理，即可将图像内车牌区域提取出来。本章节将采用 YOLOV5+Landmark 的网络结构，在推理出车牌区域的同时也将车牌的四个角点回归出来，为后续的车牌矫正奠定基础。

3.1 检测网络

YOLOV5 中提供了四种 YOLOV5 的模型可供使用，其在 COCO 数据集上进行检测的效果如图 3-1 所示，更精准的数据如表 3-1 所示。

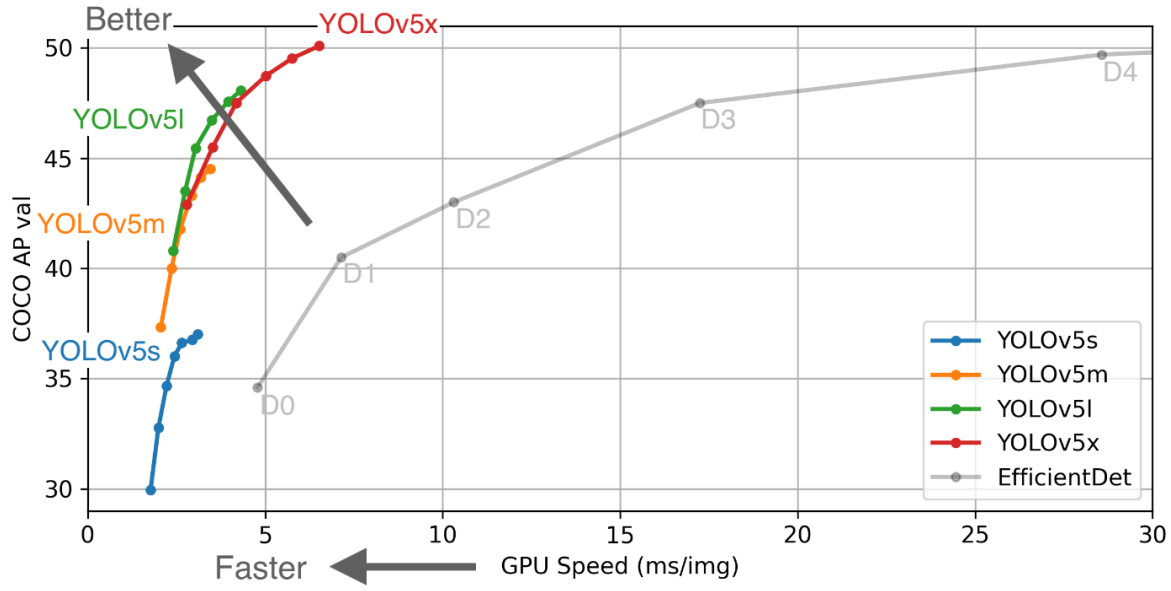


图 3-1 四种 YOLOV5 网络性能对比

Fig.3-1 Performance comparison for four kinds of YOLOV5

表 3-1 四种 YOLOV5 网络性能参数

Table 3-1 Performance parameters for four kinds of YOOOV5

模型	图像大小	*AP*(%)	推理速度(ms)	*FPS*	参数量(M)	*GLOPS*
YOLOV5s	640*640	36.8	2.2	455	7.3	17.0
YOLOV5m	640*640	44.5	2.9	345	21.4	51.3
YOLOV5l	640*640	48.1	3.8	264	47.0	115.4
YOLOV5x	640*640	50.1	6.0	167	87.7	218.8

其中，数据表 3-1 中的 *AP*(Average Precision)指的是平均精确度，*FPS*(Frames Per Second)指的是模型处理图像时每秒可达到的帧数，*GLOPS*(Giga Floating-point Operations Per Second)指的是每秒 10 亿次的浮点运算数。可见随着参数量的提升，*AP* 和 *GLOPS* 也随之提升，但是推理的速度反而跟着下降。本研究计划设计一个经济且易于移植的模型，参数量越小越好、推理速度越快越好，因此采用 YOLOV5s 作为车牌检测的检测网络。

本研究使用的 YOLOV5 的网络结构如图 3-2 所示，YOLOV5 的网络结构能大致分为输入端、Backbone 结构、Neck 结构、Prediction 结构。

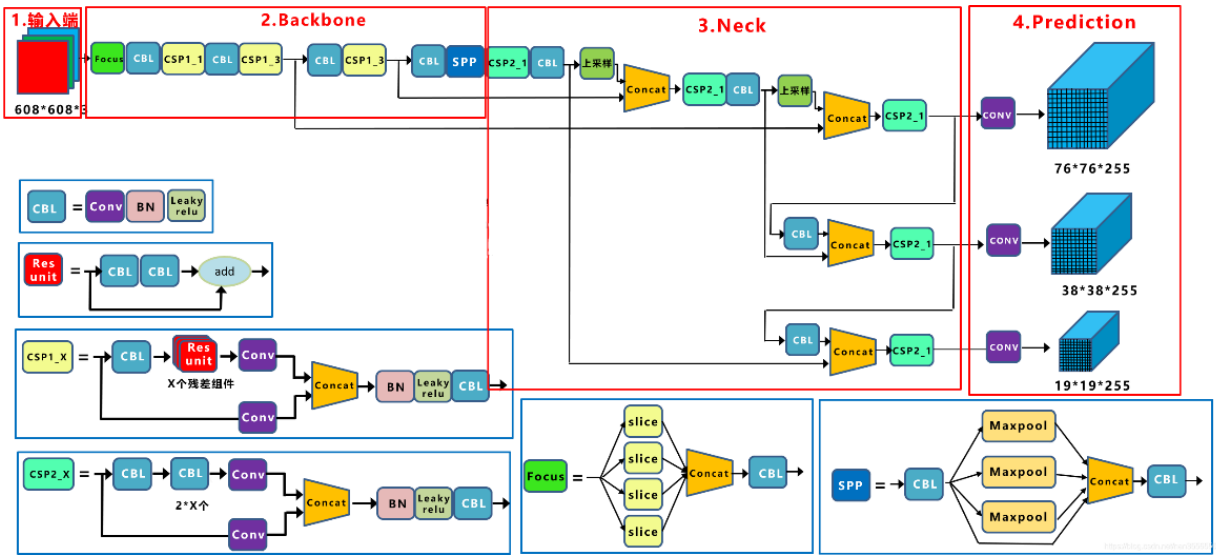


图 3-2 YOLOV5 网络结构

Fig.3-2 Structure of YOLOV5 network

3.1.1 输入端

输入端处，YOLOV5 使用了三种预处理方法增强图像的学习特征，即 Mosaic 数据增强、自适应锚框计算、自适应图像缩放技术，这三种技术经证实，均对网络的训练有益。

（1）Mosaic 数据增强

Mosaic 数据增强是一种提升对小目标检测的数据集增强方法，其效果如图 3-3 所示。

图 3-3 Mosaic 数据增强结果

Fig.3-3 Result of Mosaic data augmentation

Mosaic 数据增强就是将一个 batch 中的图像进行随机抽取，然后进行随机缩放、随机裁剪、随机排布于一张画布中，然后将这样一张一张画布送入网络中进行训练。该方法通过对图像的缩放，使得检测的目标变得较原图更小或更大，有利于学习更多目标细节。通过剪裁与随机排布让不同的几张图像背景信息连续排布，使网络在学习过程中能“看”见更多的背景信息。对背景和检测目标“一大一小”的处理大大增强了对小目标的检测效果。

（2）自适应锚框计算

在传统的目标检测方法中，给定一个固定尺寸的窗口，即可将一张图像分为若干个窗口，再设定步长，把对应计算到的窗口内像素内容输入至网络进行预测和分类，但是这样计算存在着三种缺点：

(1) 由于窗口尺寸比较固定，大目标时通常无法完全包含，因此不适合大目标的检测；

(2) 需要计算的窗口内容很多，计算量大；

(3) 预测时单个窗口只能和单个目标一一对应，网络接收窗口中的像素内容，计算其预测概率，根据概率为窗口内的内容分配类别。

在 YOLOV5 中，锚框初始的宽高只需要预设，然后在网络训练中，网络会根据初始设定的锚框与标签中框定目标的真实框进行比对，使用 K-means 对锚框进行自动更新。对锚框进行 K-means 的步骤为：(1)随机选取 K 个锚框作为初始锚框，中心点作为锚点；(2)使用如式子(3-1)所示的交并比(Intersection over Union, IoU)指标，A 代表初始锚框，B 代表框定目标的真实框，将每个初始锚框分配给距离最近(即 IoU 最大)的锚点；(3)计算每个锚点中认定为一类的的锚框宽和高的平均值，更新锚点；(4)重复 2-3 直到锚点位置固定或者到了设置的迭代终止条件。这样针对不同的数据集中目标的大小、长宽比等信

息，都能自适应生成对应的锚框。

$$IoU = \frac{A \cap B}{A \cup B} \tag{3-1}$$

（3）自适应图像缩放

因为可能输入的图像大小不同，因此在输入网络前需要进行缩放到统一标准后再送入网络，YOLOV5 中是将图像缩放为宽高统一的图像后再送去训练。在缩放过程中，为了缩放到每张图片宽高都统一尺寸通常需要填充一定的黑边，但如果黑边填充过多，反而会造成信息过于冗余，降低网络对于图像的推理速度，因此如图 3-4 所示，YOLOV5 在填充黑边时填充最少的黑边，如此简单的改进据官方介绍可以减少 34%的推理时间。

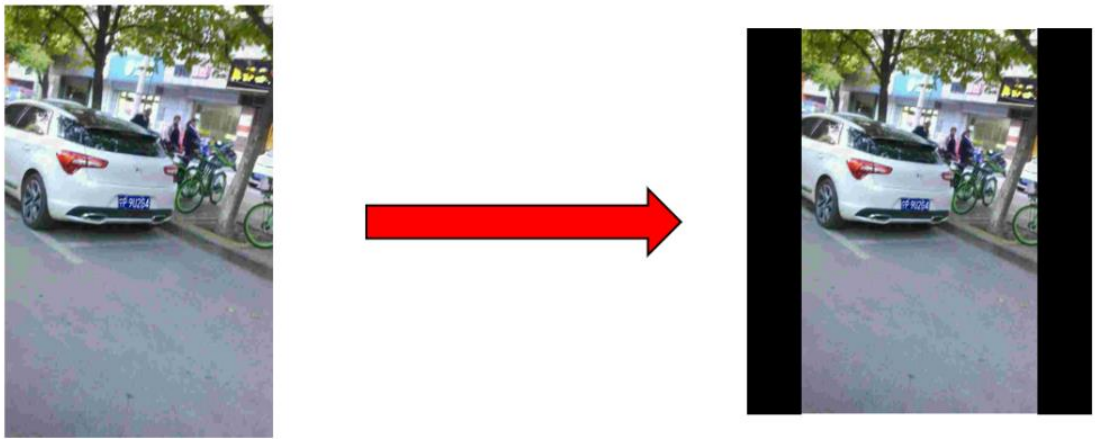


图 3-4 自适应图像缩放

Fig.3-4 Adaptive image scaling

3.1.2 Backbone 结构

在主干网络 Backbone 的作用是提取网络特征，在 YOLOV5 的 Backbone 中主要是 CSP 结构与其他 YOLO 版本中没有的 Focus 结构。

（1）CSP 结构

CSPNet 全称是 Cross Stage Paritial Network，是一种轻量级的网络结构，具有计算量小、推理速度快的特点。梯度信息重复是导致计算冗余、计算量大的直接原因，因此在设计层面上 CSPNet 卷积层提取特征映射时分为两个支路，在后续层次结构将支路进行合并，尽可能保证准确率的同时减小了计算量。综上，在 YOLOV5 中使用该网络结构是因为其主要的两个优点：

(1) 增强网络提取特征的能力，在尽可能保证准确率的同时减小了计算量；

(2) 降低计算量与内存成本。

（2）Focus 结构

Focus 结构的作用是对图像进行图 3-5 所示的切片操作，具体操作时一张相片每隔

一个像素取一个值，这样一张相片可以分成四张相片且没有任何的信息缺失，但是输入通道则扩大了四倍。这样做的好处是在卷积运算时经过 Focus 处理的特征图在计算过程中计算量是原来的四倍，特征提取地更加充分。

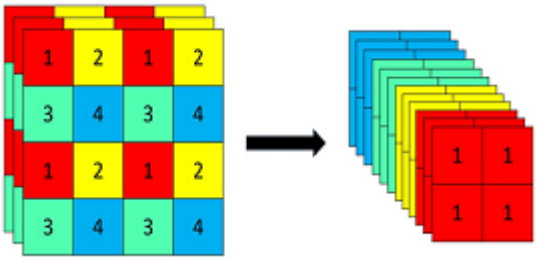


图 3-5 Focus 结构的切片操作

Fig.3-5 Slices operation in Focus structure

3.1.3 Neck 结构

Neck 结构通常位于主干网络和预测之间，目的是为了更好地利用主干网络中提取后的特征。如图 3-6 所示采用的是特征金字塔(Feature Pyramid Networks, FPN)+路径聚合网络(Path Aggregation Network, PAN)结构加强特征的融合能力。

FPN 是使用多层特殊的 CNN 模型提取图像中各维度特征的方法，FPN 主要是通过融合高底层特征提升目标检测的效果，尤其可以提高小尺寸目标的检测效果。通过该方法可以提取出表达原图特征能力更强的特征图，可以在后续的计算机视觉任务目标检测、目标分割、关键点提取等来使用。自下至上的通路：CNN 卷积提取特征从下到上一步一步压缩特征的过程，层次靠下的卷积层反映粗略的图像信息特征像边缘等；层次靠上的卷积层可以反映较深层次的图像特征信息，例如：物体边缘甚至是目标类别等；高层卷积层输出的 feature map 小于低层次，但却能表示更大维度、更深层次的图像信息，因此我们在处理每一层信息时会参考上一层的深层次信息做为其输入做本层的卷积以及生成一个通路将上层信息做一个上采样，然后将本层信息经一个卷积层后与上采样结果进行相加即得出本层的输出。

FPN 层后是一个自底而上的特征金字塔，与前面的特征金字塔不一样的是，该特征金字塔包含两个 PAN 结构。前一层 FPN 层自顶向下传达强语义特征，而后一层包含 PAN 的特征金字塔则自底向上传达强定位特征，两者结合，从不同的卷积层与其对应的检测层进行参数聚合。

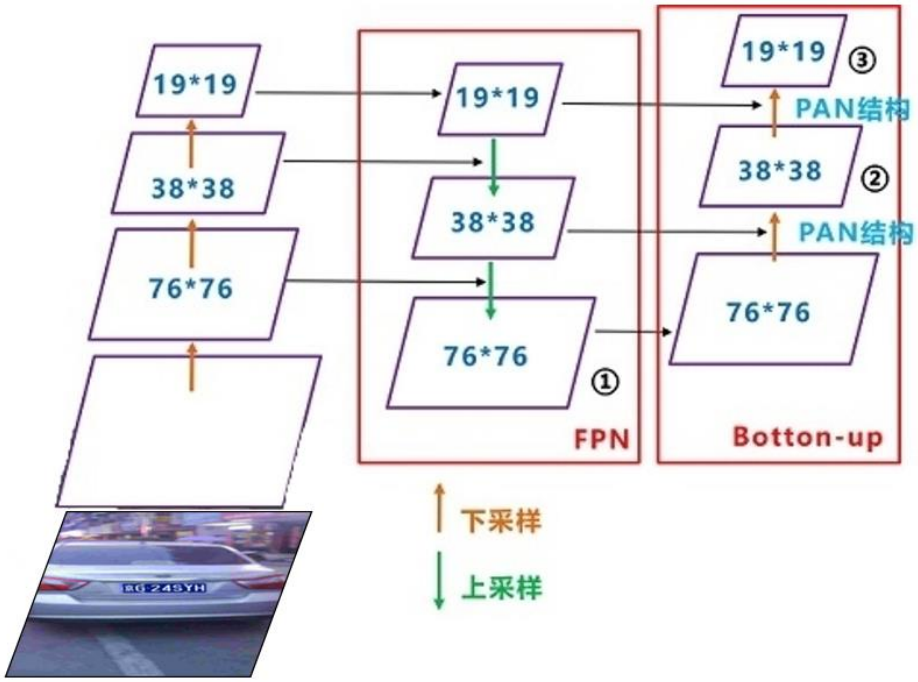


图 3-6 FPN+PAN 结构

Fig.3-6 FPN+PAN structure

3.1.4 输出端

（1）Bounding box 损失函数

四维向量(x,y,w,h)可以表示常见的预测框结构，x、y 表示窗口的中心点坐标，w、h 表示预测框的宽高。对于图 3-7，红色的框 P 作为网络初步输出的预测框，绿色的框 G 即为包含目标的真实框 Ground Truth，网络学习特征的目的是寻找一种关系，构造损失函数，通过降低损失函数，使得输入原始的窗口 P 经过映射得到一个跟真实窗口 G 更接近的回归窗口$\hat{G}$。

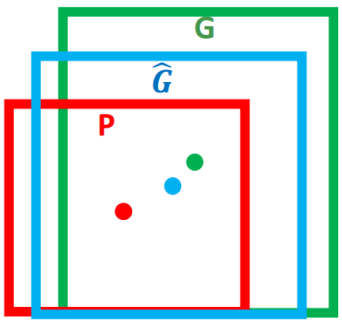


图 3-7 预测框回归

Fig.3-7 Prediction box regression

YOLOV5 中使用 GIOU_Loss 做 Bounding box 的损失函数。其大致过程如图 3-8 所示，设黄色的真实框为A_1，绿色的预测框为A_2，最大外接矩形为红色框 C，GIOU 的计算公式如式(3-2)与(3-3)所示。通过减小该损失函数来达到预测框接近真实框的目的。

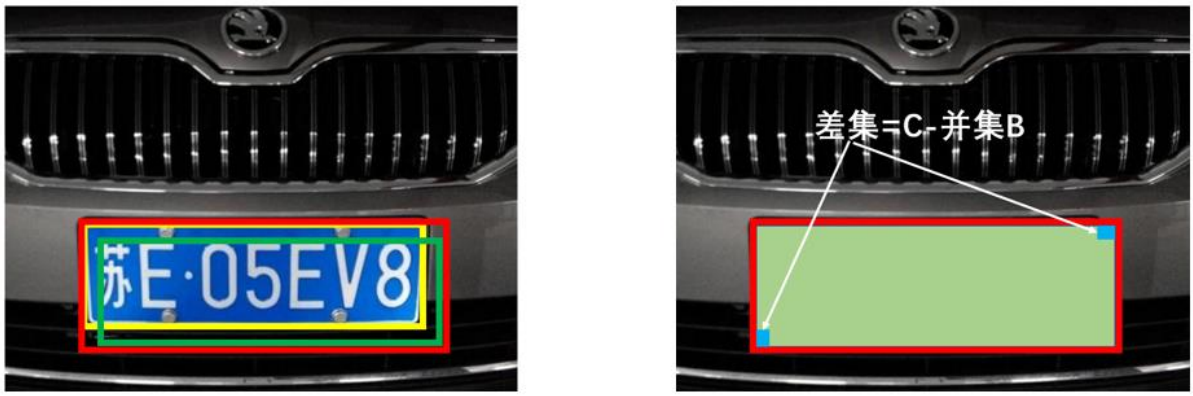


图 3-8 GIOU Loss 计算过程

Fig.3-8 Calculation process of GIOU Loss

$$B = A_1 \cup A_2 \tag{3-2}$$

$$GIOU_{Loss} = 1 - (IOU - \frac{|C-B|}{|C|}) \tag{3-3}$$

（2）非极大值抑制

非极大值抑制(Non Maximum Suppression, NMS)，用于目标检测时可能会出现是多窗口检测的情景，对于多个预测框都进行评分，目标是保留一个预测效果最优的预测框其余的全部舍弃。其过程可分为以下几步：

(1) 若有 A~F 多个矩形框，根据预测概率对其进行排序 A<B<C<D<E<F；

(2) 假设最大概率矩形框是 F，那就将除 F 以外的矩形框与 F 进行重叠度 IOU 计算，判断结果是否大于某个设定的阈值；

(3) 如果 B、D 与 F 的计算的 IOU 超过设定筛选的阈值，那么就舍弃 B、D，并确定 F 是最优且可保留下来无需再做计算的预测框；

(4) 从剩下的矩形框再选择概率最大的矩形框，然后其余的矩形框与概率最大的矩形框计算 IOU，结果大于一定的阈值则舍弃，保留下最优的预测框；

(5) 重复上述过程，得到的矩形框就是最终最优的结果。

（3）置信度与分类损失函数

置信度是网络判断网格中存在目标物体的可能程度，YOLOV5 通过优化置信度函数来提升网络识别网格中是否存在目标物体的能力。置信度损失函数L_{obj}可由式子(3-4)所示。

$$\sum_{i=0}^{K*K}\sum_{j=0}^{M} I_{ij}^{obj}[\hat{C}_i \log(C_i) + (1-\hat{C}_i)\log(1-C_i)] - \lambda_{noobj}\sum_{i=0}^{K*K}\sum_{j=0}^{M} I_{ij}^{noobj}[\hat{C}_i \log(C_i) + (1-\hat{C}_i)\log(1-C_i)] \tag{3-4}$$

网格共有 $K\times K$ 个，每个网格产生 M 个候选框 anchor，每个 anchor 经过网络会得到相应的 bounding box，最终形成 $K\times K\times M$ 个 bounding box，如果 box 内没有物体，则只计算该 box 的置信 loss。$\hat{C}_i$表示网络推理出的物体种类，C_i表示该物体真实的种类，λ_{noobj}

表示没有物体的置信度损失权重，若存在物体I_{ij}^{obj}的值为 1，I_{ij}^{noobj}的值为 0，反之I_{ij}^{noobj}为 1，I_{ij}^{obj}为 0。

分类损失来减小预测类别与真实类别之间的差距，YOLOV5 使用交叉熵的方法来表示，分类损失L_{cls}可由式子(3-5)表示。

$$\lambda_{class}\sum_{j=0}^{k*k}I_{ij}^{obj}\sum_{c\in class}[\hat{p}_i(c)\log(p_i(c))+(1-\hat{p}_i(c))\log(1-p_i(c))] \tag{3-5}$$

λ_{class}为分类损失的权重，c 是所有种类的集合，$\hat{p}_i$(c)是网络预测该物体为种类 c 的概率，$p_i(c)$是标签中物体为种类 c 的概率，是则为 1 不是则为 0。

（4）landmark 回归损失函数

为了让 YOLOV5 具备回归车牌角点的能力，使用 landmark 损失对关键点位置进行校正，landmark 损失L_{land}可被表示如式子(3-6)所示。

$$L_i^{landmark}=\left\|\hat{y}_i^{\,landmark}-y_i^{landmark}\right\|_2^2 \tag{3-6}$$

$\hat{y}_i^{\,landmark}$表示从网络中获得关键点的坐标，也就是包围盒锚点的坐标，$y_i^{landmark}$是标签中手工标注的真实关键点位置，减小L_{land}就是在减小他们坐标之间欧几里得距离的过程。

3.2 检测结果

为了验证网络的性能与目前学术界上的研究结果作对比，本研究验证结果时，分成了 CCPD 蓝牌数据集检测、生成数据集检测以及真实数据集和生成数据集混合检测。检测过程使用的设备是神舟 Z7-KP7S1，CPU 为 i7，显卡是显存为 6G 的 GTX1060。

3.2.1 CCPD 数据集检测

将 CCPD 数据集每一部分随机以 8:1:1 的比例分成训练集、验证集和测试集，初始学习率 0.01 采用 Warmup 策略，随着学习过程逐步提升至 0.1，batchsize 为 64，epochs 为 20，每张图像大小被调整为 640×640px。为了消除误检的干扰，置信度为 0.7 以上的候选区域才最终可以被认为是车牌区域。Z. Xu 等(2018)在 CCPD 数据集上使用了其他目标检测网络如：Cascade classifier、SSD300、YOLO9000、Faster-RCNN、TE2E、RPnet 方法进行对比(S. Wang et al. 2007; W. Liu et al. 2016; Redmon et al. 2016; S. Ren et al. 2015; H. Li et al. 2017)，得到了当时最先进的结果，因此本研究的方法与其论文中方法在 CCPD 数据集对比结果可由表 3-2 所示。

表 3-2 实验对比结果

Table 3-2 Result of experimental comparison

方法	FPS	AP(%)	Base(%)	DB(%)	FN(%)	Rotate(%)	Tilt(%)	Weather(%)	Challenge(%)	Blur(%)	Green(%)
Cascade classifier	32	47.2	55.4	49.2	52.7	0.4	0.6	51.5	27.5	—	—
SSD300	40	94.4	99.1	89.2	84.7	95.6	94.9	83.4	93.1	—	—
YOLO9000	42	93.1	98.8	89.6	77.3	93.3	91.8	84.2	88.6	—	—
Faster-RCNN	15	92.9	98.1	92.1	83.7	91.8	89.4	81.8	83.9	—	—
TE2E	3	94.2	98.5	91.7	83.8	95.1	94.5	83.6	93.1	—	—
RPnet	61	94.5	99.3	89.5	85.3	94.7	93.2	84.1	92.8	—	—
YOLOV5S	100	99.8	100	99.7	100	100	100	100	99.8	99.9	98.6

由表中数据可以看出，YOLOV5S 处理每幅图像的速度为 0.01s，对应的 FPS 为 100f/s。平均精度 AP 可以达到 99.8%，在 CCPD 各部分数据集中均能达到接近 100%的效果，因此在与其他学者的结果对比上，均能达到最佳的结果。

(a) (b) (c) (d) (e)

(f) (g) (h) (i)

图 3-9 检测成功的案例

Fig.3-9 Cases of successful detection

检测失败的图像如图 3-10 所示，由于车牌区域受到强光照或光照十分弱的情况下，光线在车牌上有明显的亮度过度，模型对此类的车牌检测还是具有一定的难度，会将其

误认为是背景区域因此无法检测出来。

(a)

(b)

图 3-10 失败案例

Fig.3-10 Failure cases

3.2.2 生成数据集检测

将生成数据集中的蓝黄绿数据集打乱，以 8:1:1 的比例分成训练集、验证集和测试集，初始学习率 0.01 采用 Warmup 策略，随着学习过程逐步提升至 0.1，batchsize 为 64，epochs 为 60，每幅图像大小被调整为 640×640 像素。为了消除误检情况，置信度为 0.7 以上的候选区域才被认为是车牌区域。

为了保证检测结果的严谨性，本研究使用生成数据集训练得到的权重在真实车牌上进行测试。在此先做了一个预实验，在上一小节 CCPD 蓝牌数据集测试使用的 CCPD-Base 测试集中进行检测，发现检测的成功率为 0.2%，在 CCPD-green 的测试集中进行检测，检测成功率仅为 78%，据此可以判断是出现了过拟合的现象，因为要学习一种生成数据集到真实数据集中的映射，是一个跨域的问题，若在生成数据集中过度学习就会导致在真实数据集的识别效果不佳，因此要尽可能降低训练的 epochs，找到两者之间最佳的平衡点。对于黄牌的检测效果，本研究通过网上收集各式实际黄牌数据集约 350 幅作为测试集。在此本研究也做了一个消融实验，观察经过还原现实处理数据集以及未经还原现实处理的数据集在检测真实数据集的效果上有何区别。

表 3-3 实验对比结果

Table 3-3 Result of experimental comparison

方法	FPS	AP (%)	Base (%)	DB (%)	FN (%)	Rotate (%)	Tilt (%)	Weather (%)	Challenge (%)	Blur (%)	Green (%)	Yellow (%)
Cascade classifier	32	47.2	55.4	49.2	52.7	0.4	0.6	51.5	27.5	—	—	—
SSD300	40	94.4	99.1	89.2	84.7	95.6	94.9	83.4	93.1	—	—	—
YOLO9000	42	93.1	98.8	89.6	77.3	93.3	91.8	84.2	88.6	—	—	—
Faster-RCNN	15	92.9	98.1	92.1	83.7	91.8	89.4	81.8	83.9	—	—	—
TE2E	3	94.2	98.5	91.7	83.8	95.1	94.5	83.6	93.1	—	—	—
RPnet	61	94.5	99.3	89.5	85.3	94.7	93.2	84.1	92.8	—	—	—
Ours(数据集 1)	100	40.8	41.3	7.3	28.6	21.4	15.3	20.4	37.8	4.3	91.8	94.6
Ours(数据集 2)	100	3.6	6.1	0.3	1.7	2.8	1.3	1.3	0.8	0.1	5.3	3.2

数据集 1 指的是经过还原现实处理的数据集，数据集 2 指的是未经过还原现实处理的基础数据集。从实验结果中可以看出使用经过还原现实处理的生成数据集训练的权重来检测 CCPD 真实蓝牌车牌数据集效果不佳，但是在图像质量较高且清晰的 Green 数据集上与 Yellow 数据集上有着不错的检测精度，因此在检测新能源车牌和黄色车牌时可以使用生成数据集来进行网络的训练。但是未经还原现实处理的生成数据集 2 训练的权重识别效果非常不好，因为其只能识别较为清晰的图片，但是测试集中的图片质量普遍较低。

3.3 本章小结

(1) 介绍了检测网络 YOLOV5+Landmark 网络结构的原理；

(2) 测试了 CCPD 数据集训练的权重在测试集上的检测效果；

(3) 测试了生成数据集训练的权重在 CCPD 数据集上检测效果，证明使用生成数据集做检测时可以在一定程度上替代真实数据集来训练。

第四章 基于 CRNN+CTC 的车牌识别方法

车牌中中文、英文、阿拉伯数字混合的字符可看做一个序列，与识别物体对象种类的深度学习方法不同的是，对于一连串字符序列的识别需要深度学习网络能够推理出一系列的标签，而不是预先规定的多个类别，并且对于字符标签的实际字符组合也没有预先规定组合类别。车牌字符识别可以认为是上述说明的序列识别问题，车牌中序列还有一个特征就是它的长度并不固定，例如蓝牌中是七位字符而绿牌是八位字符。RNN 是深度学习网络中的重要分支，其有着不需要固定序列中每个元素的位置的优点，为了强化 RNN 的检索能力，本研究使用 CRNN+CTC 网络来进行车牌字符的识别。

4.1 网络结构

CRNN 的总体网络结构可如图 4-1 所示，网络中可以分为三个部分：卷积层、递归层、转录层。其中，卷积层负责提取车牌图像中的特征，递归层对每一帧进行预测，转录层将每帧最后翻译成序列标签。该框架的优点是：

(1) CRNN 中有较大的感受野和融合了自注意机制，很好地解决了像车牌长序列转换到定长向量，这一过程可能会出现的感受野中信息损失的情况，有效地提高了对字符的识别能力。

(2) CTC Loss 避免了文本对齐，是在传统文本识别 Loss 上进行更新的计算方法，不管字符样式宽窄、大小的相同字符也可以输出相同的序列，因此在近年的文本和语音识别应用中逐渐成为一种常用的方法。

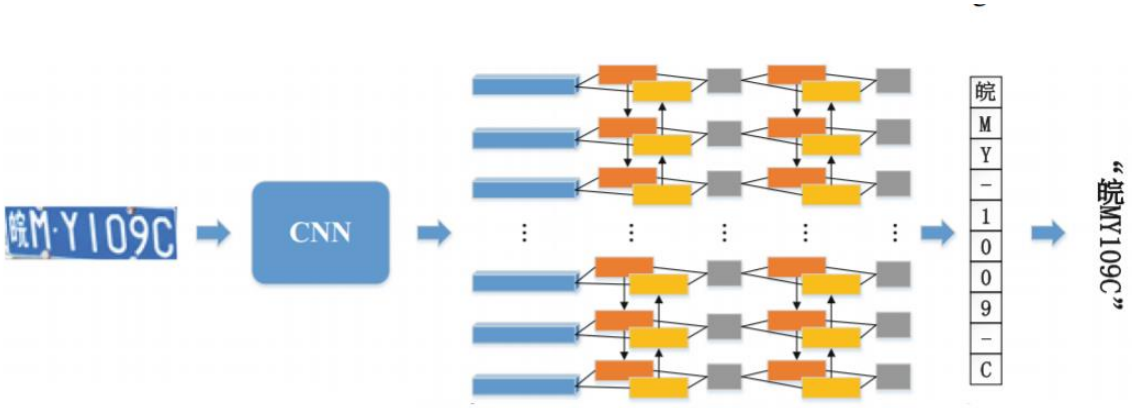


图 4-1 车牌识别网络

Fig.4-1 License plate recognition network

4.1.1 卷积层

卷积层部分是通过卷积神经网络层、最大池化层和激活函数层三者结合来构建的，输入网络之前，车牌图像要被缩放到统一的宽高。卷积后得到的特征图上对应着车牌图片自左向右排序的特征向量，特征长度是人为进行设置的，每一列的宽度固定为一定长

度的像素。由于卷积层、最大池化层和激活函数层都是在局部区域上操作的，且这种操作具备着平移不变性。总而言之，特征图的每一列对应于图 4-2 中原始图像的矩形感受野区域，图中车牌图像线条划分的部分排列的顺序与自左往右与在特征图上排列的特征向量顺序是相同的。

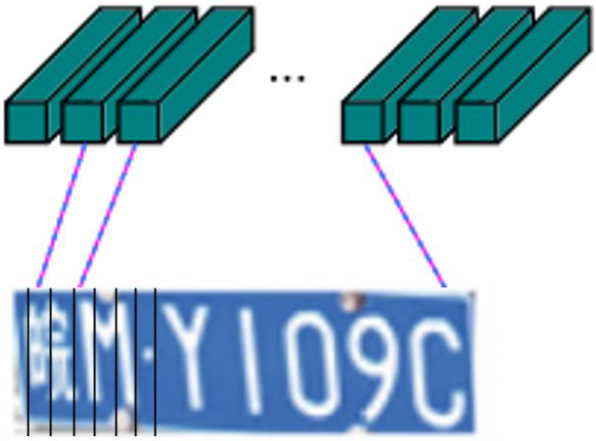


图 4-2 特征序列在原始图像上对应关系

Fig.4-2 Relationship of feature on origin image

4.1.2 递归层

图像经过卷积层处理后就到了递归层，其能起到的效果就是对特征序列$x = x1, \dots, x_T$的每一帧x_t，推理出标签分布y_t。递归层与卷积层联合使用有三个特点：

(1) 有很强的捕捉序列特征上下文关系的能力，使用上下文信息进行预测比单独使用一个符号进行预测更加地稳定，一些不确定的字符也可通过上下文关系得到其细微的区别；

(2) 可以将误差的微分进行反向传播到卷积层，使得可以在一个网络中联合训练卷积层与递归层；

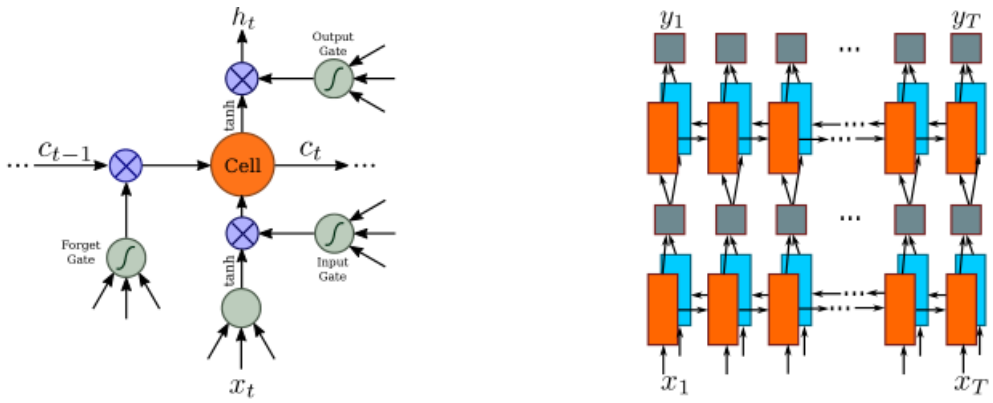
(3) 可以从头到尾对任意的序列进行遍历。

传统 RNN 中有一个隐藏层，其可以在接收一个帧x_t后，联合过去状态h_{t-1}和当前输入x_t，更新目前的状态h_t，即由式子(4-1)所示，然后基于h_t对序列分布y_t进行一个预测。但是传统 RNN 隐藏层存在着梯度消失的问题，且会增大训练的负担，限制上下文存储的范围。因此学者们基于上述的情况设计了长短期记忆(Long-Short Term Memory, LSTM)帮助解决这类问题，如图 4-3(a)所示，LSTM 由三个乘法门和一个存储单元组成，构成了输入、输出、遗忘门，LSTM 可以在减轻长期训练负担的情况下捕捉到序列上下文关系。

$$h_t = g(x_t, h_{t-1}) \tag{4-1}$$

使用 LSTM 基本单元进行堆叠，接收来自前后的信息，形成如图 4-3(b)双向的 LSTM 有更好的上下文识别能力(A. Graves et al. 2013)，这就为序列到映射搭起了桥

梁。



(a) (b)

图 4-3 递归层结构中的 LSTM 单元和双向 LSTM

Fig.4-3 LSTM unit and Bi-directional LSTM in structure of recursive layer

4.1.3 转录层

转录层的工作就是将每帧预测的结果转化为序列标签，从数学的角度上看，就是在每帧预测的前提下，找到概率最高的序列标签。

本研究使用 CTC 使用的条件概率，条件概率可以简述为一个长度为 n 的序列在经过递归层后如式(4-2)输出的后验概率矩阵，其中每一列可以表示为如式(4-3)所示。

$$y = y_1, \dots\dots, y_n \tag{4-2}$$

$$y^t = {y_1}^t, \dots\dots, {y_n}^t \tag{4-3}$$

由于是概率的形式，如式(4-4)求和后其值为 1。

$$\sum_{k=0}^{T} {y_n}^t = 1 \tag{4-4}$$

对每一列都做$argmax()$处理后，即可转化为输出字符的类别。这时经过 LSTM 的序列可以表示如式(4-5)所示。

$$N_w: (R^m)^T \to (R^n)^T \tag{4-5}$$

原来的字符长度为 m，经过递归层后变成了 n，这是因为考虑到一些位置没有字符，因此要定义一些空字符 blank 插入到这些地方。定义L是包含车牌所有字符的集合，如式(4-6)所示，加入一些空白字符到集合中。

$$L' = L \cup \{blank\} \tag{4-6}$$

将车牌字符集合插入空白字符变换到原本车牌字符集合的变换定义为如式(4-7)所示的B变换，其中L'是上述加入空白字符后长度为 n 的集合，经过B变换后变成了原始的序列L，显然L的长度要小于 n。

$$B: {L'}^n \to L^{\leq n} \tag{4-7}$$

例如：如果 n=12 时，B变换可能有以下式子(4-8)至(4-10)几种可能，将不被空白字符隔开的相同字符进行合并，即可得到正确的序列。显然对y进行B变换不是一对一的映

射，对于π_1~π_3这三种情况都有可能得到正确的序列。

$$B(\pi_1) = B(\text{—皖 MMY} - 10 - 9\text{C}) = \text{皖 MY109C} \tag{4-8}$$

$$B(\pi_2) = B(-\text{皖} - \text{MM} - \text{Y10} - 9\text{C}) = \text{皖 MY109C} \tag{4-9}$$

$$B(\pi_3) = B(\text{皖} - \text{MM} - \text{Y} - 109 - \text{C}) = \text{皖 MY109C} \tag{4-10}$$

因此，CTC 的本质是对于 LSTM 给定的输入x的情况下，输出L的概率要最大，如式(4-11)所示的概率经过前向和后向推导训练递推求最大值，其中$\pi \in B^{-1}(L)$表示经过B变换后L可能经过的所有可能π。

$$p(L|x) = \sum_{\pi \in B^{-1}(L)} P(\pi|x) \tag{4-11}$$

4.2 识别结果

在检测网络检测到车牌区域，并回归出车牌四个角点位置后，将该检测结果反馈到识别模块。在做车牌字符识别前，还需要对倾斜角度下的车牌进行透视变换校正。

4.2.1 倾斜矫正

本小节的透视变换过程与 2.2.4 小节中原理大致相同，唯一不同的地方是，本小节的操作是将车牌四个角点往一张 256×96 的画布四角进行透视变换操作，结果如图 4-4 所示，拍摄角度倾斜的车牌都被很好地矫正了过来，并且车牌字符也被正确地识别。

(a)

(b)

(c)

图 4-4 车牌矫正过程

Fig.4-4 License plate correction process

4.2.2 CCPD 数据集测试

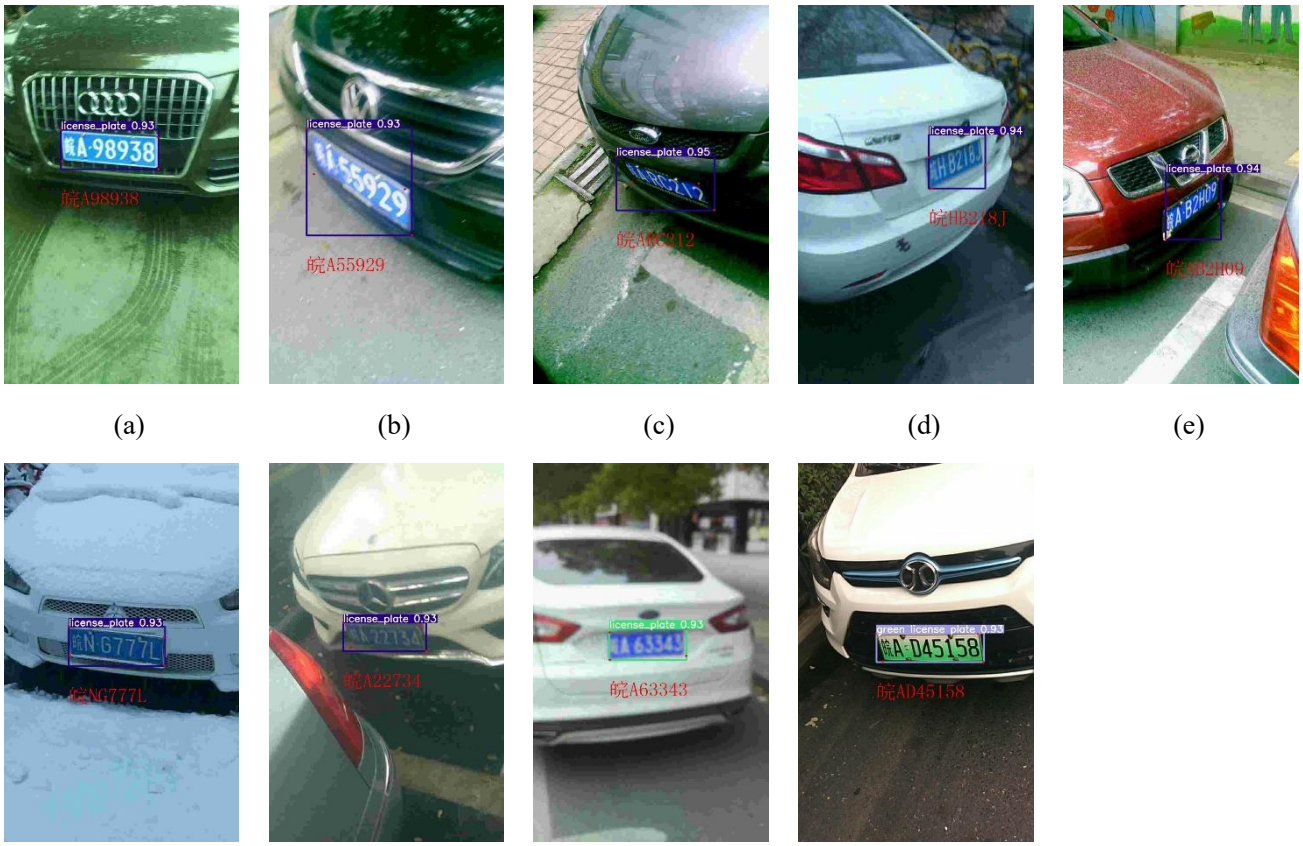
将 CCPD 中数据集使用检测模块进行检测并矫正角度后，每一部分以 8:1:1 的比例随机分成训练集、验证集和测试集，初始学习率 0.01 采用 Warmup 策略，随着学习过程逐步提升至 0.1，batchsize 为 256，epochs 为 100，每张图像大小被调整为 256×96px。与其他学者的方法在 CCPD 数据集对比结果可由表 4-1 所示，其中 HC 指的是 Holistic-CNN 网络结构。

表 4-1 实验对比结果

Table 4-1 Result of experimental comparison

方法	FPS	AP (%)	Base (%)	DB (%)	FN (%)	Rotate (%)	Tilt (%)	Weather (%)	Challenge (%)	Blur (%)	Green (%)
Cascade classifier+HC	29	58.9	69.7	67.2	69.7	0.1	3.1	52.3	30.9	—	—
SSD300+HC	35	95.2	98.3	96.6	95.9	88.4	91.5	87.3	83.8	—	—
YOLO9000+HC	36	93.7	98.1	96.0	88.2	84.5	88.5	87.0	80.5	—	—
Faster-RCNN+HC	13	92.8	97.2	94.4	90.9	82.9	87.3	85.5	76.3	—	—
TE2E	3	94.4	97.8	94.8	94.5	87.9	92.1	86.8	81.2	—	—
RPnet	61	95.5	98.5	96.9	94.3	90.8	92.5	87.9	85.1	—	—
YOLOV5S+CRNN	67	98.8	99.9	97.1	97.9	98.0	98.9	97.1	96.5	98.1	97.8

从实验结果中可以看出，CRNN+CTC 的识别方法不仅识别速度快，且在性能生均优于其他学者的方法，并且在 Rotate 上，由于做了倾斜矫正，识别效果明显优于其他方法。在 CCPD 数据集上各个例子的识别结果如图 4-5 所示。



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

图 4-5 识别成功的案例

Fig.4-5 Cases of successful recognition

4.2.3 生成数据集测试

对于生成数据集，我们采用与 CCPD 数据集测试中同样的训练方法进行训练，检测权重使用 CCPD 数据集训练的权重保证检测尽量少地出错。在此也做了一个消融实验，观察同样 epochs 训练出来的经过还原现实处理数据集以及未经还原现实处理的数据集在识别真实数据集的效果上有何区别。在 CCPD 数据集上各项实验结果如表 4-2 所示。

表 4-2 实验对比结果

Table 4-2 Result of experimental comparison

方法	FPS	AP (%)	Base (%)	DB (%)	FN (%)	Rotate (%)	Tilt (%)	Weather (%)	Challenge (%)	Blur (%)	Green (%)	Yellow (%)
Cascade classifier+HC	29	58.9	69.7	67.2	69.7	0.1	3.1	52.3	30.9	—	—	—
SSD300+HC	35	95.2	98.3	96.6	95.9	88.4	91.5	87.3	83.8	—	—	—
YOLO9000+HC	36	93.7	98.1	96.0	88.2	84.5	88.5	87.0	80.5	—	—	—
Faster-RCNN+HC	13	92.8	97.2	94.4	90.9	82.9	87.3	85.5	76.3	—	—	—
TE2E	3	94.4	97.8	94.8	94.5	87.9	92.1	86.8	81.2	—	—	—

表 4-2 实验对比结果（续）

Table 4-2 Result of experimental comparison

RPnet	61	95.5	98.5	96.9	94.3	90.8	92.5	87.9	85.1	—	—	—
Ours(数据集 1)	67	37.3	60.8	1.7	17.5	23.4	13.4	35.1	54.6	2.3	23.6	37.4
Ours(数据集 2)	67	0.1	0.1	0.0	0	0.0	0	0	0	0	0.3	0.3

数据集 1 指的是经过还原现实处理的数据集，数据集 2 指的是未经过还原现实处理的基础数据集。由表中数据可以看出，使用经过还原现实处理的数据集进行训练的权重，在识别难度较小的 Base 数据集和多种复杂情况结合的 Challenge 数据集上有一定的识别能力，但是 DB 和 Blur 数据集中车牌面积在整幅图片占比过小，提取的车牌图像过于小且模糊，因此识别效果不佳，总体上看识别精度只有 37.3%，还是达不到应用的要求。但是相比于生成数据集 2 训练的权重，本研究的还原现实处理方法在提升真实车牌识别能力上还是有不错的效果，这是因为数据集 2 训练的权重基本识别不出视觉效果复杂的真实车牌，但经过还原现实处理后即可获得一定的识别能力。

4.2.4 生成数据集辅助多省份识别

在 2.1 节中介绍 CCPD 数据集各省份车牌数量时就提到存在省份不均的问题，皖省车牌图像居多，而其他省份的车牌图像较少，特别是在“藏”和“宁”省份上识别效果非常不佳。因此本小节将尝试在 CCPD 数据集中的一些省份数量稀少的数据集上添加一些生成数据集，观察是否能够起到缓解省份不均问题的效果。

为了验证效果，本研究将 CCPD 数据集中每个省份的图片保留三张作为测试集，其余的都用作训练。然后填充生成数据集将每个省份的数量都保持在 2000 幅以上。原本在没扩充省份数量前，在保留的省份测试集上识别精度为 67.5%，扩充数据集后，识别精度可达到 75.2%，因此可以说明生成数据集能帮助缓解省份不均问题，缓解效果的一些实例如图 4-6 所示，在数量稀少省份上有了明显改善。

(a)

(b)

(c)

(d)

(e) (f)

(g) (h)

图 4-6 省份识别成功案例

Fig.4-6 Successful cases of province recognition

4.3 本章小结

(1) 介绍了 CRNN+CTC 网络结构的原理；

(2) 测试了 CCPD 数据集训练的权重在测试集上的识别效果；

(3) 测试了生成数据集训练的权重在 CCPD 数据集上识别的效果；

(4) 验证了生成数据集具有缓和训练集中省份不均问题的作用。

第五章 基于微信小程序与 PyQt5 的智能停车系统

智能停车系统的前端分为管理者端以及用户端，本章节为用户前端设计。对于用户层面，需要设计一套能够登陆、注册、查询停车情况的手机程序。对于管理者层面，需要设计一套能够观察车辆入库时的车牌信息是否有误、观察目前的入库情况的 PC 前端界面。

5.1 基于微信小程序的用户前端设计

5.1.1 总体设计

微信小程序有着完善的编程环境，稳定且易于开发，编写程序完毕后即可在微信中上线使用。使用微信小程序自带函数库可设计美观的登录界面和注册界面，获取并保存用户信息。在车牌算法获取到车辆的入库行为后，即可在停车信息处观察到自车辆的停车记录。

数据库需要与管理者前端和消费者前端建立起联系，常用的数据库有 MYSQL、SQL Server、微信云开发数据库，微信云开发数据库可以直接在微信小程序开发时建立，并且通过 Python 的 Requests 库编写 API 语句可以访问到微信云开发数据库中的数据，因此数据库开发过程中微信云开发数据库为首选方法。本研究将用云开发数据库来存储用户信息、用户密码、用户车牌号码、车辆入库行为等信息。

因此本研究采用微信小程序设计前端，微信云开发数据库设计后端的方法来完成用户前端的整体设计，本微信小程序结构如图 5-1 所示。本章节着重介绍登录、注册、个人信息界面、车辆登记界面以及入库查询界面的设计实现。

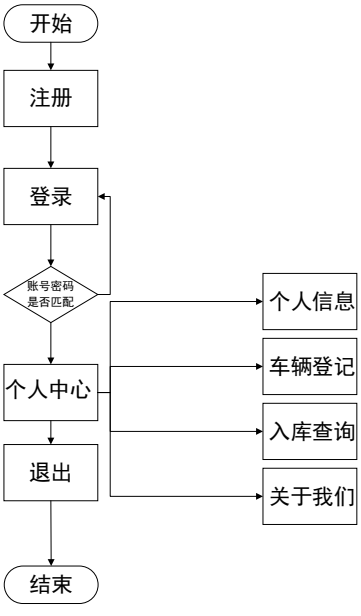
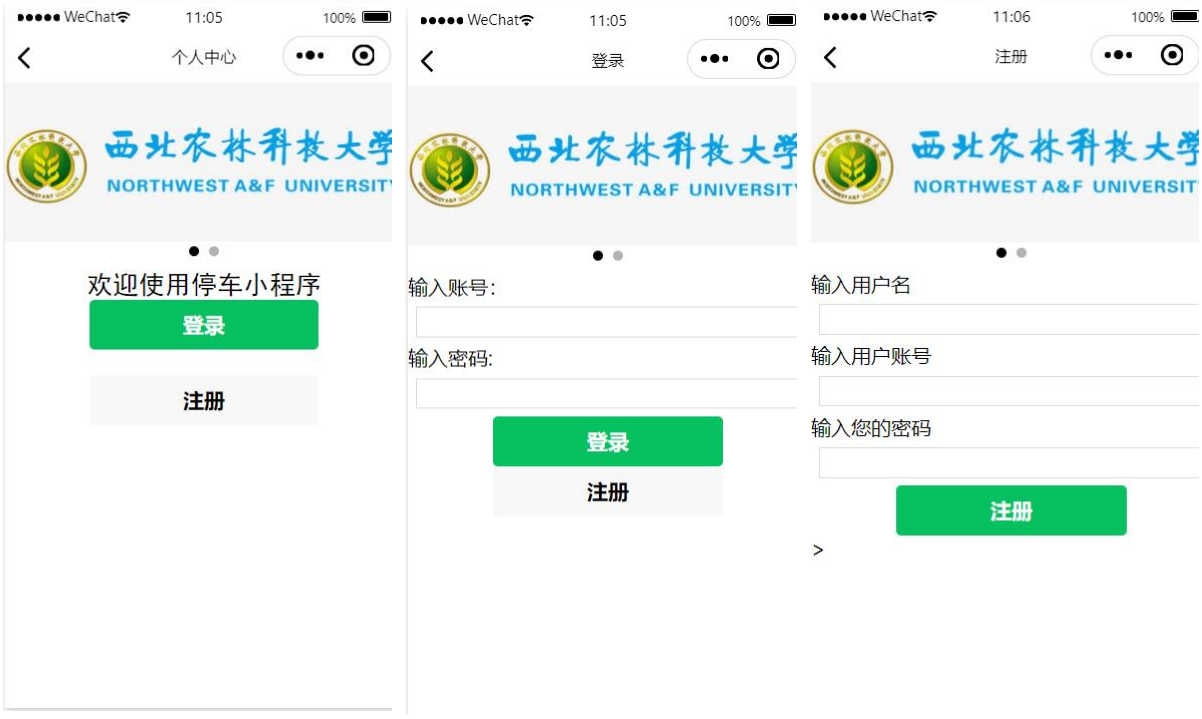


图 5-1 用户前端设计流程

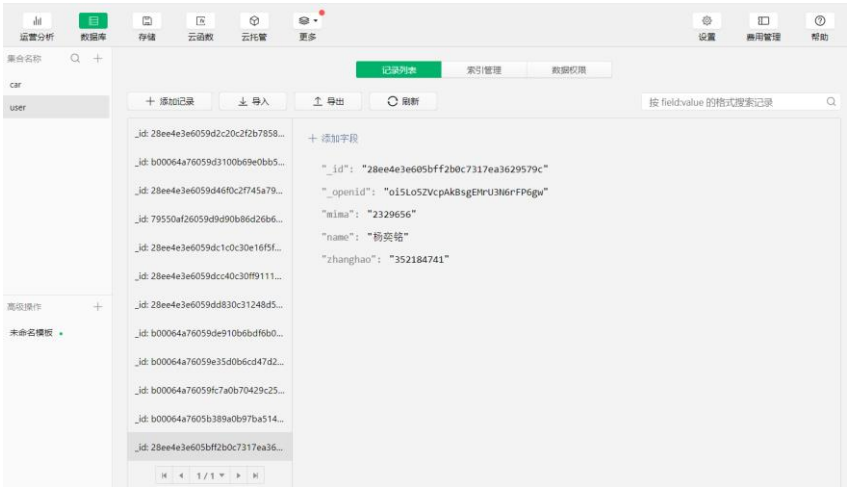
Fig.5-1 Design process of client

5.1.2 登录与注册

如图 5-2 所示，进入小程序后第一个界面是用户登录与注册界面。注册操作时为了让账号和密码更加规范，因此设了一个限制条件，账号与密码都是大于四位数小于十位数的字符串，且账号未被注册过才允予注册。输入用户名、账号、密码后选择注册按钮，微信小程序会自动到云开发数据库 user 表中进行检索，如果账号没被注册过，则添加用户名、账号、密码进入 user 表中。



(a) (b) (c)



(d)

图 5-2 用户登录与注册

Fig.5-2 User login and registration

在登陆界面进行登录时，微信小程序会进入到云开发数据库 user 表中进行检索，找到用户名与密码与输入的用户名和密码进行匹配，如果匹配成功则允许用户进入到个人

中心界面，并且将用户的登录信息记录到如图 5-3 所示的本地的缓存中。这么做的目的是减少对数据库访问的次数增加运行效率，如果还需要用到用户信息时直接从本地缓存中进行读取即可。

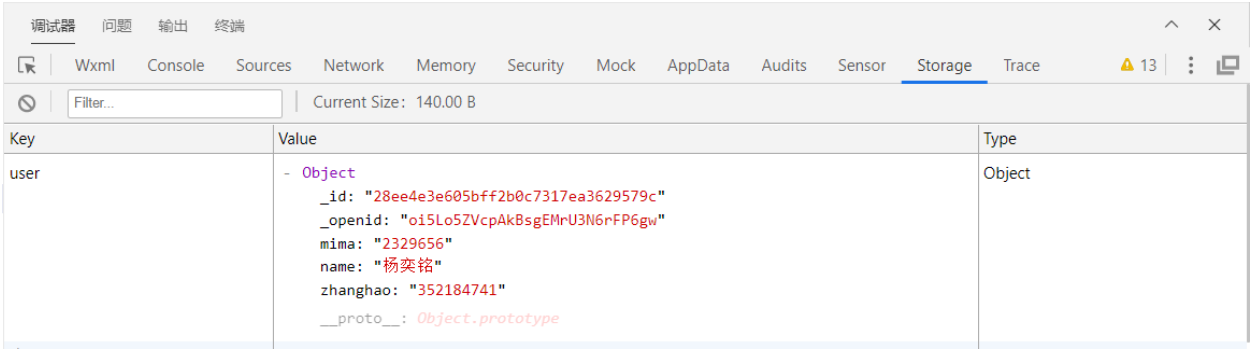


图 5-3 本地缓存

Fig.5-3 Local cache

成功登陆后则进入到了如图 5-4 所示的个人中心页面，包含个人信息、车辆登记、入库查询、关于我们、退出，这五个可操作的按钮。

图 5-4 个人中心界面

Fig.5-4 Personal center screen

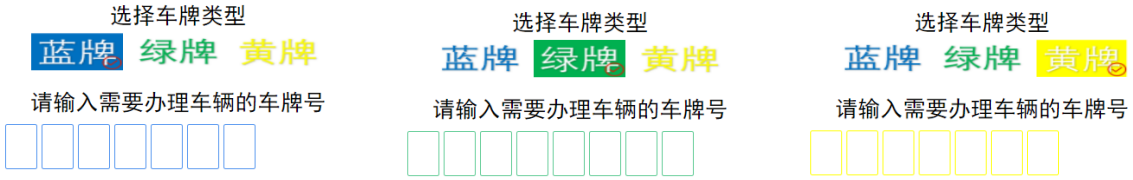
5.1.3 车辆登记

为了能够正常使用停车小程序，进入到个人中心后第一步要做的是对自己的车辆进行登记，车辆登记界面如图 5-5 所示。

图 5-5 车辆登记界面

Fig.5-5 Vehicle registration screen

为了便于更好地确定用户的车牌类型，在这里设置了三种车牌类型的登记方式，分为蓝牌、绿牌和黄牌，并且对于三样车牌的输入格子数根据蓝牌、黄牌 7 位车牌号，绿牌 8 位车牌号进行如图 5-6 所示的自动调整。

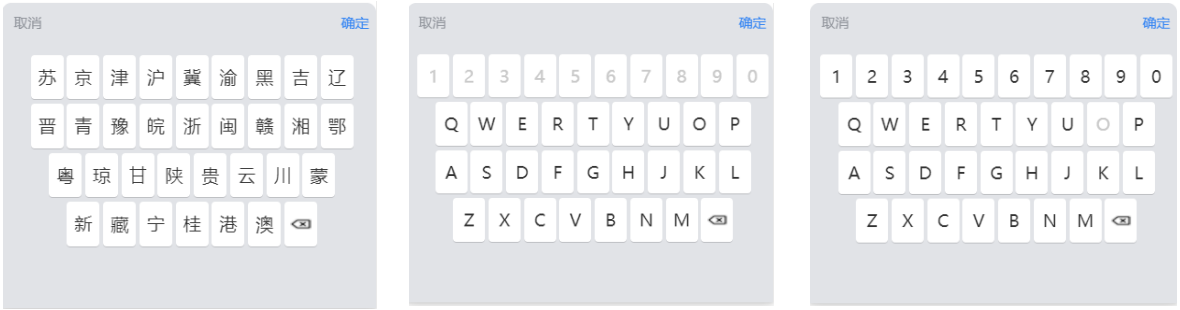


(a) (b) (c)

图 5-6 车牌类型选择

Fig.5-6 Vehicle type selection

在输入车牌字符的过程中，可以设计一个如图 5-7 车牌字符输入的输入法代替用户自带的输入法，车牌字符输入法中可以只包含车牌字符中的样式，免去了输入字符时拼音或者手写的繁琐操作。输入对应的位，车牌输入法会切换到不同的样式。由于车牌第一位是省份，因此第一位时是省份输入法。车牌第二位是除了字符“I”以外的英文字符，因此第二位时是英文字符输入法。第三位以及随后几位是除了字符“I”与“O”外的数字与英文字符的组合，因此第三位即随后几位是数字与英文字符组合输入法。



(a) (b) (c)

图 5-7 车辆登记输入法

Fig.5-7 Input method of vehicle registration

选定车牌类型，输入完车牌信息后，点击确定。如图 5-8 所示，微信小程序会进入到云端服务器中对用户数据进行更新，然后还会对用户本地缓存信息进行更新，更新结果如图 5-9 所示。添加了字段“license_plate”和“tabIndex”分别存储车牌号与车牌类型信息，“tabIndex”中 0、1、2 分别代表了蓝色车牌、绿色车牌、黄色车牌。


```
"_id": "28ee4e3e605bff2b0c7317ea3629579c"
"_openid": "oi5Lo5ZVcpAkBsgEMrU3N6rFP6gw"
"license_plate": "陕A88888"
"mima": "2329656" (string)
"name": "杨奕铭"
"tabIndex": "0"
"zhanghao": "352184741"
```


图 5-8 云端数据库个人信息更新

Fig.5-8 Personal information update in cloud database

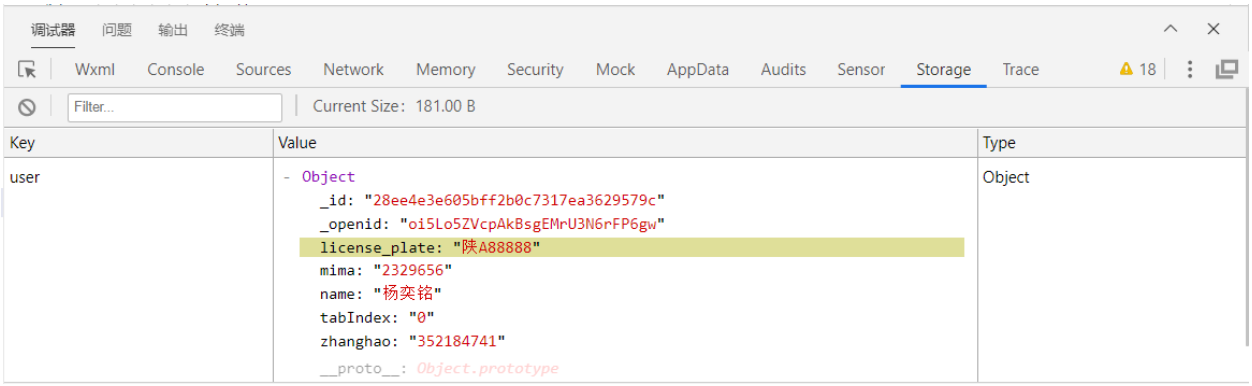


图 5-9 本地缓存更新

Fig.5-9 Local cache update

5.1.4 个人信息查询

用户登记完车辆信息后，返回个人中心界面可以点击个人信息界面查询目前登记的个人信息是否有误。如图 5-10 所示，小程序自动在云开发数据库的 user 表中找到当前

登录用户 id 所对应的信息记录，然后将记录中不同属性词段进行检索。将“zhanghao”对应的用户账号，“name”对应的用户名，“tabIndex”对应的三种车牌类型，“license_plate”对应的车牌号检索出来，并显示在界面的对应位置上供用户查询。

图 5-10 个人信息界面

Fig.5-10 Personal information screen

5.1.5 入库信息查询

在管理者前端处检测与识别到车辆进入时，就会将停车地点、停车时间、车牌号码进行记录，再查询登记该车牌号的用户“user_id”，查询到之后再一同记录在云开发数据库‘car’表内。

当用户在个人中心页处按下入库查询按钮时，则进入到停车入库查询界面。微信小程序会根据当前用户的“user_id”在云开发数据库的‘car’表内进行检索，查找出该“user_id”登记车辆的停车记录，例如图 5-11 所示，查询到有两个入库的信息。


```
"_id": "79550af2605ab6fd0b9e60a00a3fc722"
"cfCarPark": "西北农林科技大学"
"inTime": "2021.3.24"
"numberPlate": "陕A88888"
"outTime": "2021.3.25"
"payTime": true
"user_id": "b00064a76059de910b6bdf6b0ce444b9"
```

(a)

```
"_id": "b00064a7605abd960b82a8f6281165c2"
"cfCarPark": "西北农林科技大学"
"inTime": "2021.3.25"
"numberPlate": "陕A88888"
"outTime": "2021.3.26"
"payTime": true
"user_id": "b00064a76059de910b6bdf6b0ce444b9"
```

(b)

图 5-11 云开发数据库中 car 表记录的入库信息

Fig.5-11 Information recorded by CAR table in cloud database

根据入库信息次数的多少，如图 5-12 所示，在当前小程序界面中生成对应个数的表栏，在表栏内显示用户登记下车辆的入库信息，包括停车地点、车牌号、入场时间、出场时间信息，除此之外还可以显示该次停车是否支付了停车费。由于没有企业微信号，因此支付功能暂时无法实现。

图 5-12 入库信息查询界面

Fig.5-12 Incoming information query screen

5.2 基于 PyQt5 的管理者前端设计

5.2.1 总体设计

车牌算法的主体功能实现大多在使用最为广泛的 Windows 系统中运行，因此管理者前端要通过能在 Windows 系统中使用且移植性强的方法来设计，常用的 Windows 前端设计方法有 Java、Qt、PyQt5。其中 PyQt5 因其编写语言为 Python，允许使用 Python 语言调用 Qt 库中的 API。这样做的最大好处就是在保留了 Qt 高运行效率的同时，大大提高了开发效率，Python 多样化的第三方库为开发提供了大量的便利，车牌检测、识别程序也通过 Python 语言进行编写，便于直接与管理者前端建立起联系，在前端中直接植入算法，因此 PyQt5 是设计管理者前端的首选办法。

管理者界面的总体设计如图 5-13 所示，打开界面后会因需要加载界面资源和导入车牌检测与车牌识别模型而延迟数秒，然后输入车牌图片后即可获得识别出车牌区域与车牌号的结果图像，车牌号码也会以字符的形式得到。将车牌号码输入到微信云开发数据库中存储用户信息的‘user’表内，与用户的车牌号码进行匹配，如果匹配到了一致的车牌号码，则将入库信息登记到微信云开发数据库中存储入库信息的‘car’表内，同时

车辆被允许放行进入到车库中。如果用户没有提前在微信小程序内完成用户注册和车牌登记，则获取到的用户车牌号码进入到数据库中进行匹配时就无法找到该用户的车牌信息，因此不将改用户的入库行为登记入‘car’表内，停车系统入口处放行失败。

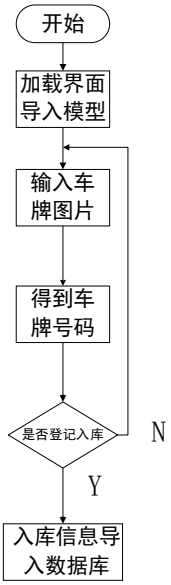
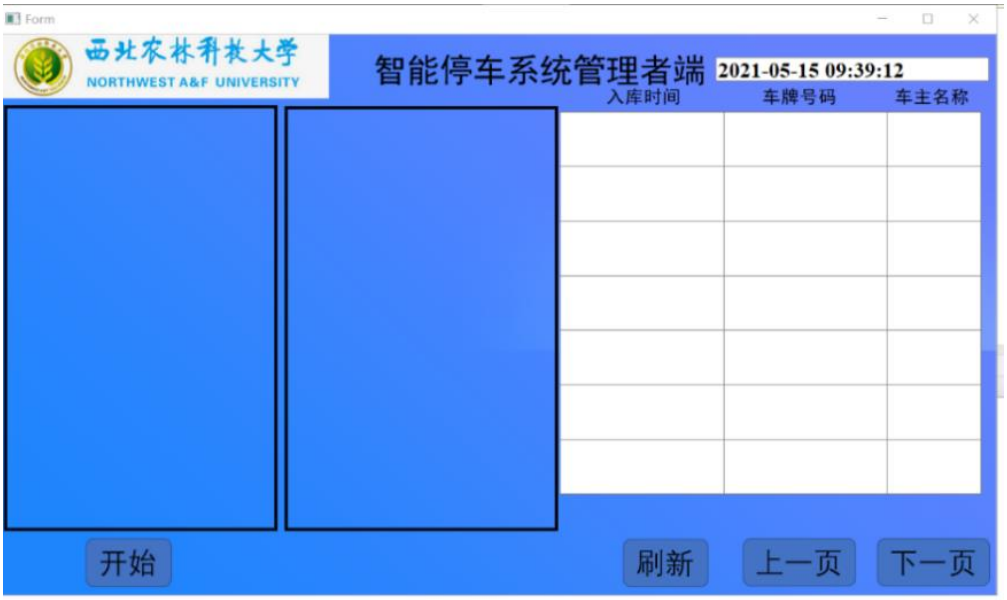


图 5-13 管理者端设计流程

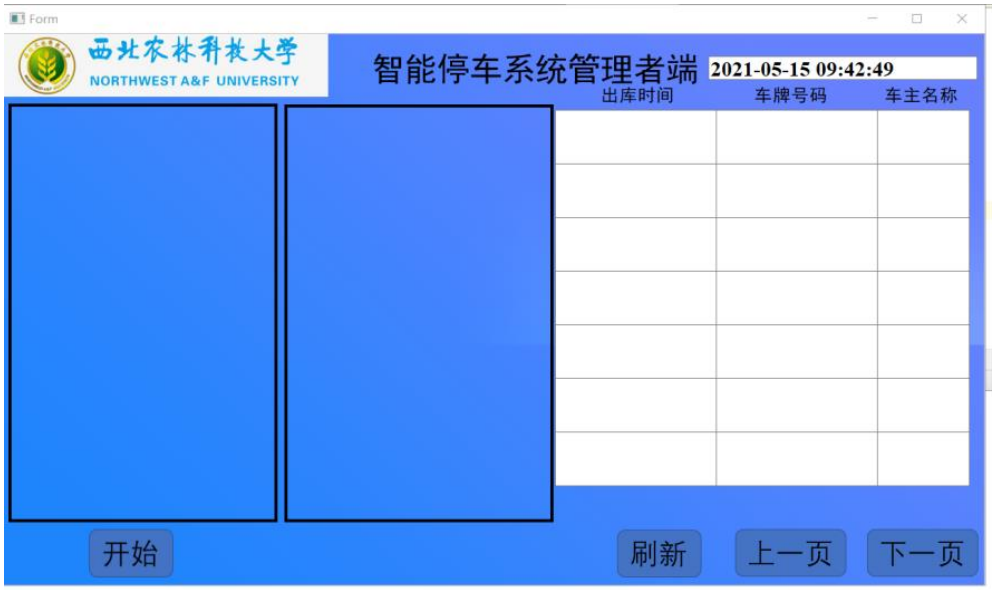
Fig.5-13 Design process of manager

5.2.2 界面设计

界面设计如图 5-14 所示，考虑到一些大型的停车场，入口和出口处在不同通道，因此会在入口与出口处分别设立两个管理者岗，故而本界面也相应地分为了入库界面与出库界面。在界面左侧的两个方框用于显示入库的原始图像与识别到车牌号后的图像，‘开始’按钮用于导入图像，右侧的表格内显示入库时间、车牌号码、车主名称，点击‘刷新’按钮后，界面自动使用 Python 的 requests 库，按照编好的 url 语句向微信云开发数据库请求查询‘car’表内的入库信息，查询成功后直接显示在表格内。‘上一页’与‘下一页’按钮用于翻看由于表格大小有限无法完全显示在表格中的入库记录。入库管理者界面和出库管理者界面唯一不同的是，前者记录并显示入库时间，后者记录并显示出库时间，为了保障用户的隐私性，因此没有直接把入库和出库时间都显示在同一界面上，只有有权利进入云开发数据库查询数据的管理者才能看到完整的入库与出库信息。



(a)



(b)

图 5-14 管理者端设计

Fig.5-14 Design of manager screen

5.2.3 使用演示

首先我们在微信小程序中注册一个账号，用户名称为“吴亦凡”，登记的车牌号位渝 A55555，注册并登记好车牌后的个人信息页面如图 5-15 所示。

图 5-15 个人信息

Fig.5-15 Personal information

如图 5-16 所示，在以入库管理者端为例，当管理者端检测到渝 A55555 车牌时，则会到数据库‘user’表中寻找车牌号码为渝 A55555 的账号信息，找到以后点击“更新”，界面将自动把入库时间、车牌号码、用户名称显示在界面的表格中供查看。

图 5-16 入库信息查看

Fig.5-16 View incoming information

同时，如图 5-17 所示，在用户界面的微信小程序内，在入库查询界面可以看到对应的入库信息。因此这里可以看出，管理者前端和用户前端成功通过微信云开发数据库建立起联系，共享其中的数据，给管理者和用户都带来便利。

图 5-17 入库信息查看

Fig.5-17 View incoming information

5.3 本章小结

(1) 介绍了小程序的总体方案设计；

(2) 介绍了小程序车辆登记功能实现；

(3) 介绍了小程序个人信息查询功能实现；

(4) 介绍了小程序入库信息查询功能实现；

(5) 介绍了管理者端的总体设计与程序逻辑；

(6) 介绍了管理者端的界面设计情况与各部分的作用；

(7) 对管理者端的使用方式进行了阐述。

第六章 结论与展望

6.1 结论

本研究基于深度学习技术，分别利用 YOLOV5+Landmark 和 CRNN+CTC 两种网络结构，实现了复杂环境下车牌的检测与识别功能，针对车牌数据集收集困难的问题，提出了生成车牌数据集的方法，并能取得一定的检测与识别效果，最终围绕着上述研究成果打造了一套包含微信小程序前端、PyQt5 管理者前端、云开发数据库的智能停车系统。本研究对于多种复杂环境条件下提高我国车牌检测识别率，从而促进城市的智慧交通管理具有重要的研究意义，主要结论如下：

(1) 为了解决车牌图像获取困难、车牌训练集中不同省份数据不均的问题，提出了一种生成车牌数据集的方法并进行了验证，结果表明该方法生成的数据集可在车牌检测、车牌识别方面起到较好的辅助作用；

(2) 为了准确定位并提取车牌区域，采用深度学习网络 YOLOV5 对车牌区域进行检测，并且加入 Landmark 回归的方法以定位车牌的四个角点位置，为后续倾斜车牌矫正奠定了基础。网络选择方法在综合考虑平均精度、推理速度、参数量、权重大小等因素后，优选使用 YOLOV5S 为主要网络结构，经 CCPD 数据集训练后的测试结果与学者们的实验结果对比表明，本研究的车牌检测网络在 CCPD 数据集上的平均准确率为 99.8%，每帧处理速度为 10ms，表明本研究使用的车牌检测网络效果具有检测速度快、检测精度高的特点，可在各种复杂环境下实现车牌的精确检测；

(3) 在识别车牌算法方面，考虑到车牌识别是一个特殊的字符识别实例，选择使用 CRNN+CTC 网络结构作为车牌识别网络，同时为了提升算法对倾斜车牌的识别效果，使用透视变换实现了倾斜车牌的矫正。使用 CCPD 数据集做训练后的测试结果与学者们的实验结果对比表明，本研究的车牌识别网络在 CCPD 数据集上的平均准确率为 98.8%，每帧处理速度为 15ms，说明本研究的车牌识别网络在保证识别速度的同时，具备良好的识别效果，特别是对于倾斜车牌的识别效果有了进一步的提高；

(4) 使用微信小程序与 PyQt5 分别设计了用户前端与管理者前端，后端使用了微信云端数据库进行搭建，共同实现了注册与登录、车牌信息登记、个人信息查询、停车信息查询等功能。其功能完善，界面美观的优点，可以很好地满足用户与管理者两方的需求。

6.2 创新点

总结本文的上述内容，本研究的创新点总结如下：

(1) 提出一种扩充车牌数据集的方法，且该方法生成的数据在车牌检测与识别方面能起到辅助作用；

(2) 在 YOLOV5S 的网络基础上进行改进，加入了关键点回归以提取车牌角点大致区域，为后续车牌倾斜矫正奠定基础；

(3) 使用微信小程序与 PyQt5 搭建用户前端与管理者前端，并将车牌检测与识别算法嵌入至管理者前端中，让手机程序与电脑程序通过云开发数据库构建起联系，打造了一套智能停车系统。

6.3 展望

本研究基于深度学习方法，实现了对复杂环境下的车牌检测与识别，对于本研究目前取得的一些进展中，尚有部分可以提升与改进的地方：

(1) 本研究在 CCPD 各部分复杂车牌环境测试集上均取得了不错的效果并且也有使用车牌检测与识别权重进行过现实环境中的实测，测试环境光照充足时能得好的效果，但是光照过暗或者光照不均则检测识别效果不佳；

(2) 使用经还原现实处理的生成数据集进行训练的网络在检测与识别简单的车牌实例上能取得了较好的效果，但是在检测与识别复杂环境下的车牌效果不佳，因此在还原现实处理方法方面仍有进步的空间；

(3) 绿色车牌与黄色车牌部分，因为没有比较完备的真实绿色与黄色车牌数据集，因此尚且无法很好地完成在绿色与黄色车牌的识别；

(4) 本研究的网络结构都是使用 GPU 进行推理计算，尚不清楚网络的移植难度和成本适不适合应用于城市的车辆管理上。

参考文献

李媛, 石嘉诚, 陈军辉, 等. 2021. 2010～2017 年四川省机动车污染物排放趋势分析.环境科学, 42(02):643-652.

盖盈盈. 2017 北京市汽车保有量增长的原因及对交通拥堵影响研究[硕士学位论文].北京：北京交通大学.

陈璇璇. 2008. 人口、资源与环境视角下的汽车保有量影响因素研究[硕士学位论文].北京:首都经济贸易大学.

安静. 2016.基于数学形态学的图像增强算法及其应用[硕士学位论文].兰州:西北师范大学.

A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In ICASSP, 2013.

A. Graves, S. Fernández, F. Gomez, et al. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks// Proceedings of the 23rd international conference on Machine learning, New York.

Al-Ghaili A M, Mashohor S, Ismail A, et al. 2008. A new vertical edge detection algorithm and its application// Computer Engineering & Systems International Conference on. IEEE.

Cynthia, Lum, Julie, et al. 2011. License plate reader (LPR) police patrols in crime hot spots: an experimental evaluation in two adjacent jurisdictions. Journal of Experimental Criminology, 7(4):321-345.

C. Ding, D. Wang, C. Liu, et al. 2017. Exploring the influence of built environment on travel mode choice considering the mediating effects of car ownership and travel distance. Transportation Research Part A Policy & Practice , 100:65-80.

C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, et al. 2008. License plate recognition from still images and video sequences: a survey. IEEE Transactions on Intelligent Transportation Systems, 9(3):377-391.

C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, et al. 2006. A license plate-recognition algorithm for intelligent transportation system applications. IEEE Transactions on Intelligent Transportation Systems, 7(3):377-392.

C. Busch, R. Domer, C. Freytag, et al. 1998. Feature based recognition of traffic video streams for online route tracing// IEEE Vehicular Technology Conference. IEEE.

C. Hsieh, Y. Juan, K. Hung. 2005. Multiple license plate detection for complex background// Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on. IEEE Computer Society.

D. Zheng, Y. Zhao, J. Wang. 2005. An efficient method of license plate location. Pattern Recognition Letters,

26(15):2431-2438.

E. Lee, P. Kim, H. Kim. 1994. Automatic recognition of a car license plate using color image processing// Image Processing, Icip-94, IEEE International Conference. IEEE.

F. Kahraman, B. Kurt, M. Gkmen. 2003. License plate character segmentation based on the Gabor transform and vector quantization// International Symposium on Computer and Information Sciences. Springer, Berlin, Heidelberg.

F. Alegria, P. S. ao. 2006. Vehicle plate recognition for wireless traffic control and law enforcement system// IEEE International Conference on Industrial Technology. IEEE.

G. Liu, Z. Ma, Z. Du, et al. 2011.The calculation method of road travel time based on license plate recognition technology. Proc. Adv. Inform , 201(2011):385–389.

H. Bai, C. Liu. A hybrid license plate extraction method based on edge statistics and morphology// IEEE, 2004.

H. Lee, S. Chen, S. Wang. 2004. Extraction and recognition of license plates of motorcycles and vehicles on highways// International Conference on Pattern Recognition. IEEE.

H. Wu, H. Chen, R. Wu, et al. 2006. License plate extraction in low resolution video// International Conference on Pattern Recognition. IEEE.

H. Caner, H. S. Gecim, A. Z. Alkar. 2008. Efficient embedded neural-network-based license plate recognition system[J]. IEEE Trans Vehicular Technology, 57(5): 2675–2683.

H. Zhang, W. Jia, X. He, et al. 2006. Learning-Based license plate detection using global and local features// International Conference on Pattern Recognition. IEEE.

H. Li, P. Wang, M. Ou, et al. 2018. Reading car license plates using deep neural networks. Computer Vision And Image Understanding, 72:14-23.

H. Li, P. Wang, C. Shen. 2017. Towards end-to-end car license plates detection and recognition with deep neural networks. IEEE Transactions on Intelligent Transportation Systems, 30(3): 1126-1136.

J . Matas, K. Zimmermann. 2005. Unconstrained licence plate and text localization and recognition// IEEE Intelligent Transportation Systems. IEEE.

J. Zhuang, S. Hou, Z. Wang, et al. 2018. Towards human-level license plate recognition// Proceedings of the European Conference on Computer Vision (ECCV), Munich.

J. Wang, H. Huang, X. Qian, et al. 2018. Sequence recognition of Chinese license plates. Neurocomputing, 317(23):149-158.

K. Deb, H. U. Chae, K. H. Jo. 2009. Vehicle license plate detection method based on sliding concentric windows and histogram. Journal of Computers, 4(8):771-777.

K. Miyamoto, K. Nagano, M. Tamagawa, et al. 1991.Vehicle license-plate recognition by image analysis//

Proceedings of the International Electronic Control Instruments. IEEE .

L. Xie, T. Ahmad, L. Jin, et al. 2018. A new CNN-based method for multi-directional car license plate detection. IEEE Transactions on Intelligent Transportation Systems, 19(2):507-517.

M. Sarfraz, Ahmed M J, 2003. Ghazi S A. Saudi Arabian license plate recognition system[C]// International Conference on Geometric Modeling & Graphics. IEEE.

M.I. Chacon， A. Zimmerman. 2003.License plate location based on a dynamic PCNN scheme// Proceedings of the International Joint Conference on. IEEE.

M. Dong, D. He, C. Luo, et al. 2017. A CNN-based approach for automatic license plate recognition in the wild// British Machine Vision Conference 2017.

N. Bellas, S. M. Chai, M. Dwyer, et al. 2006. FPGA implementation of a license plate recognition SoC using automatically generated streaming accelerators// Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. IEEE.

P. K. Chaney, T. M. Devinney, R. S. Winer. 1993. The impact of new product introductions on the market value of firms. Journal of Product Innovation Management, 12(1):82-83.

R. Zunino, S. Rovetta. 2000. Vector quantization for license-plate location and image coding. IEEE Transactions on Industrial Electronics, 47(1):159-167.

Redmon, Farhadi, A. 2016. Yolo9000: better, faster, stronger. arXiv preprint 1612

S. Kranthi, K. Pranathi, A. Srisaila. 2011. Automatic number plate recognition. Image Analysis for Transport Applications, 2(3): 408–422.

S. Wang, H. Lee. 2003. Detection and recognition of license plate characters with different appearances// Intelligent Transportation Systems. IEEE.

S. Wang, H Lee. 2007. A cascade framework for a real-time statistical plate recognition system. IEEE Transactions on Information Forensics and Security 2(2):267–282

S. Chang, L. Chen, Y. Chung, S. Chen. 2004. Automatic license plate recognition// Wseas International Conference on Mathematical Methods & Computational Techniques in Electrical Engineering. World Scientific and Engineering Academy and Society (WSEAS).

S. Ren, K. He, R. Girshick. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. 91–99

V. Kamat, S. Ganesan. 1995. An efficient implementation of the Hough transform for detecting vehicle license plates using DSP'S. Proc. Real-Time Tech, 7(11):58-59.

W. Wang, J. Tu . 2020. Research on license plate recognition algorithms based on deep learning in complex environment. IEEE Access, 8: 91661- 91675.

W. Chu. 2011. How the Chinese government promoted a global automobile industry. Industrial & Corporate

Change, 7(5):5-21.

W. T. Ho, H. W. Lim, Y. H. Tay. 2009. Two-stage license plate detection using Gentle Adaboost and SIFT-SVM// First Asian Conference on Intelligent Information and Database Systems, ACIIDS 2009, Dong hoi, Quang binh, Vietnam, April 1-3, IEEE Computer Society.

W. Liu, Anguelov, Erhan et al. 2016. Single shot multibox detector. In: European conference on computer vision, Springer, 21–37.

Y. Chiou, L. Lan，C. Tseng et al. 2012. Optimal locations of license plate recognition to enhance the origin-destination matrix estimation. Asian Transport Studies, 8(11):1-14.

Y . Lecun, Y. Bengio, G. Hinton. 2015. Deep learning. Nature, 521(7553):436-444.

Y. Liu, H. Huang, J. Cao, et al. 2017. Convolutional neural networks-based intelligent recognition of Chinese license plates. Soft Computing, 22:2403-2419.

Z. Xu, W. Yang, A. Meng, et al. 2018. Towards end-to-end license plate detection and recognition: a large dataset and baseline// European Conference on Computer Vision. Springer, Cham.

致 谢

四年光阴，悄然流逝，又到了一年夏天，到了要毕业的季节。在西农我度过了忙碌且充实的本科时光，是我人生最为难忘的人生旅程。从一个远走他乡、害怕社交的青年，到独当一面、敢于担当的班长，到不忘初心、牢记使命的党员，再到独立思考、刻苦钻研的准研究生。每当遇见难以克服的困难之时，总有贴心陪伴的父母、良师益友的老师、情同手足的挚友出手相助，没有他们的鼓励以及为我指引正确的方向，我将无法在困难面前坚守信念，坚定大学四年要达到的目标，将无法战胜一个又一个挑战，成为大家心中的骄傲。

首先，我想感谢母校西北农林科技大学，在高考失意之时来到西农，从一开始的抗拒，到认识她光荣历史的敬佩，再到她给予我一个机会上升到更高平台学习的感激。在西农这么广阔的平台中，我才有机会在本科期间磨砺自身，向光明的未来前行。

其次，我衷心感谢我的指导老师宋怀波老师。从接手电信 1701 班开始，宋老师在我们身上倾注了无数心血。我们成为师生短短三年内，每当我抑郁失意，不知该如何确定未来努力的方向时，宋老师总能从我的实际情况出发，殚精竭虑地为我着想。他伟岸宽广的胸怀、待人接物的作风、严谨负责的态度使我终身受益。想起过往老师为我付出的种种，感激之情萦然缭绕，无言以表唯有铭记于心！

此外，本科期间里我接受到秦立峰老师、龙燕老师、郭交老师、赵继政老师、余克强老师、王东老师、张佐经老师、胡瑾老师、张海辉老师等诸多老师的循循教导，感谢老师们赋予的知识，让我跨进了电子信息专业领域的大门，未来我将继续在电子信息专业领域刻苦钻研，用更高更远的建树回报各位老师们的辛勤教导。

感谢完成本论文引用到的文献、专著的作者，没有各位作者在专业领域中筚路蓝缕、以启山林，就没有我们后人乘凉，循着前辈的足迹扩宽道路。

特别感谢我的亲人、爱人给予我在精神上和物质上的支持，你们分担了我的忧愁，激励了我的成长，未来我会成长为值得你们信赖、值得你们依靠的人！

最后也对百忙之中抽出宝贵时间来审阅论文的专家教授们表示诚挚的感谢！

逆风的方向，更适合飞翔，感谢相遇，不知不觉，不止感谢。

杨奕铭

2021 年 5 月 30 日

作者简介

杨奕铭，2021 年毕业于西北农林科技大学机械与电子工程学院电子信息工程专业，获得学士学位。

学术研究情况：

[1] **杨奕铭**,邸馨瑶,宋怀波.融合超像素分割算法与肢体特征的奶牛躯干精确分割方法[J]. 江苏农业科学,2020,48(15):253-260.

[2] **杨奕铭**,陈凤,蔡婷,宋怀波.基于视频分析的振荡果实轨迹跟踪方法[J].江苏农业科学, 2020,48(14):261-267.

[3] 实用新型专利《一种带有半“S”型书页固定器的伸缩式阅读支架》，专利号：ZL201921370839.9

[4] 实用新型专利《一种牛饲料移动搅拌装置》，专利号：ZL202020101464.2

竞赛获奖情况：

[1] 第十届全国大学生数学竞赛二等奖；

[2] 第十一届全国大学生数学竞赛二等奖；

[3] 首钢京唐杯第十二届全国大学生节能减排社会实践与科技竞赛三等奖；

[4] 东方红杯第四届全国大学生智能农业装备创新大赛二等奖；

[5] 2019 年“互联网+”竞赛校级金奖。

学号：**2017012872**

西北农林科技大学
NORTHWEST A&F UNIVERSITY

2021 届本科生毕业设计

外文文献翻译

基于深度神经网络的端到端车牌检测与识别

学院（系）：	机械与电子工程学院
专　　业：	电子信息工程
年级班级：	2017 级 1 班
学生姓名：	杨奕铭
指导教师：	宋怀波
协助指导教师：	无
完成日期：	2021 年 5 月

基于深度神经网络的端到端车牌检测与识别

摘 要：在这项工作中，我们解决了自然场景图像中的车牌检测和识别问题。我们提出了一种统一的深度神经网络，它可以在单次前向遍历中同时定位车牌和识别字母。整个网络可以进行端到端的培训。与已有的将车牌检测和识别作为两个独立的任务并分步解决的方法不同，我们的方法通过一个网络联合解决这两个任务。不仅避免了中间误差积累，而且加快了处理速度。为了进行性能评估，测试了三个数据集，包括在不同条件下从不同场景捕获的图像。大量实验表明了该方法的有效性和高效性。

1 总体介绍

汽车车牌的自动检测与识别在智能交通系统中占有重要地位。它在安全、交通控制等领域有着广泛的应用前景，近年来引起了人们的广泛关注。

然而，现有的大多数算法只能在受控条件下或在复杂的图像捕获系统下才能很好地工作。在不可控的环境下准确读取车牌仍然是一项具有挑战性的任务。难点在于高度复杂的背景，如店面、橱窗、护栏或砖块中的一般文字，以及随机拍摄条件，如照明、变形、遮挡或模糊。

以往的车牌检测和识别工作通常将车牌检测和识别视为两个独立的任务，并分别采用不同的方法进行求解。然而，车牌检测和识别的任务是高度相关的。通过检测方法得到准确的包围盒可以提高识别准确率，而识别结果可以用来消除误报，反之亦然。因此，在本文中，我们提出了一个统一的框架，在同一层次上联合处理这两个任务。设计了一种深度神经网络，它以一幅图像为输入，同时输出车牌位置和车牌标签，具有较高的效率和准确性。证明了低层特征既可以用于检测，也可以用于识别。整个网络可以端到端地训练，而不使用任何启发式规则。网络架构如图 1 所示。据我们所知，这是第一个将车牌检测和识别集成到一个网络中并同时解决它们的工作。本工作的主要贡献如下：

一个单一的统一的深度神经网络，它可以从图像中检测车牌，并同时识别所有的标签。整个框架不涉及启发式过程，如使用图版颜色或字符空间，并避免了字符分组或分隔等中间过程。它可以进行端到端的训练，只需要训练所需的图像、车牌位置和标签。该系统在车牌检测和字母识别上都达到了很高的准确率。

其次，检测和识别都共享卷积特征，与单独使用模型相比，参数更少。此外，通过检测损失和识别损失的联合优化，提取的特征将具有更丰富的信息。实验表明，联合训练模型可以提高检测和识别性能。

通过将车牌识别直接集成到检测后，而不是通过单独的模型来处理它们，所产生的系统更高效。使用我们的框架，我们不需要从输入图像中裁剪检测到的车牌，然后通过单独的网络识别它们。整个框架在 Titan X GPU 上每幅图像耗时 0.3−0.4 秒。

论文的其余部分安排如下。第二节对相关工作进行了简要论述。第三节提出了集成

方法，并对各个部分进行了详细介绍。第四节进行了实验验证，第五节给出了结论。

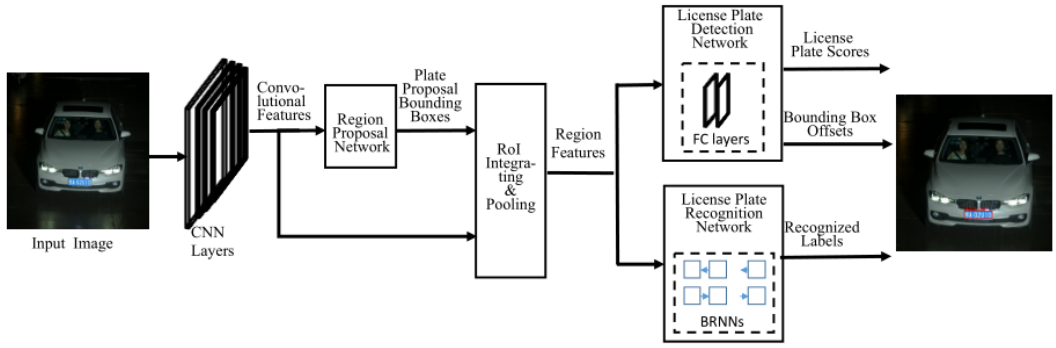


图 1 我们模型的总体结构。该算法由多个卷积层组成，包括用于车牌的区域提取网络、用于车牌区域集成和汇集的网络、用于车牌检测和包围盒回归的多层感知器以及用于车牌识别的 RNN。在给定输入 RGB 图像的情况下，通过单次前向评估，网络同时输出作为车牌的预测边界框分数、具有比例不变平移的边界框偏移量和相对于方案的对数空间高度/宽度移位，以及识别的车牌标签。提取的区域特征既可用于检测，又可用于识别，既减少了计算量，又减小了模型规模。

2 相关工作

由于车牌检测和识别一般是分开处理的，所以我们分别对各个方面的工作做了简要的介绍。

1)车牌检测：车牌检测是将图像中的车牌以包围盒的形式定位。现有的方法大致可以分为四类：基于边缘的、基于颜色的、基于纹理的和基于字符的。由于车牌通常是具有特定纵横比的矩形，并且它们呈现出比图像中其他地方更高的边缘密度，所以边缘信息被广泛地用于车牌检测。提出了一种基于边缘的车牌检测方法。将期望最大化(EM)算法应用于边缘聚类，提取边缘密集、形状与车牌相似的区域作为候选车牌。提出了一种新的线密度滤波方法，用于连接边缘密度较高的区域，并从二值边缘图像中去除每行和每列中的稀疏区域。基于边缘的方法计算速度快，但由于对不需要的边缘过于敏感，不能应用于复杂图像。

基于颜色的方法是基于车牌的颜色通常与车身的颜色不同这一观察结果而提出的。通过对目标颜色像素的分析，提出了一种车牌检测方法。通过剥离搜索，利用彩色几何模板对伊朗车牌进行定位。Chang 等人，提出了一种基于不同前景和背景颜色的 RGB 图像台湾车牌检测方法。他们开发了一种对黑白、红白和绿白边缘敏感的彩色边缘检测器。基于颜色的方法可用于检测倾斜或变形的车牌。然而，它们对自然场景图像中的各种光照条件非常敏感，无法区分图像中与车牌颜色和大小相似的其他对象。

基于纹理的方法试图根据车牌区域的非常规像素强度分布来检测车牌。余等人。首先利用小波变换提取图像的水平和垂直细节。然后利用经验模态分解(EMD)分析对投影数据进行处理，定位出指示车牌位置的期望波峰。Giannoukos 等人，提出了一种基于车牌纹理局部不规则性的滑动同心窗(SCW)车牌识别算法。为了提高检测速度，提出了操

作员上下文扫描(OCS)技术。与边缘或颜色相比，基于纹理的方法使用更具区分性的特征，但会导致更高的计算复杂度。

考虑到车牌是由一串字符组成的，最近出现了很多基于字符特征的工作，因为它包含了更具体的信息。Zhou 等人。将车牌检测问题表述为视觉匹配问题。为每个字符生成包含方向、特征尺度和相对位置等几何线索的主视觉单词(PVW)，并用于车牌提取。Li 等人，在第一阶段应用最大稳定极值区域(MSER)对图像中的候选字符进行提取。然后构造条件随机场(CRF)来表示车牌字符之间的关系。最后通过 CRF 上的置信度传播推理对车牌进行定位。基于字符的方法更可靠，召回率更高。然而，图像背景中的其他文本对性能有很大影响。

2)车牌识别：以往的车牌识别工作通常是先对车牌中的字符进行分割，然后利用光学字符识别(OCR)技术对分割出的字符进行识别。例如，使用极值区域(ER)从粗略检测的车牌中分割字符并细化车牌定位。使用受限 Boltzmann 机器来识别字符。字符分割采用 MSER。使用线性判别分析(LDA)分类器对字符识别的局部二值模式(LBP)特征进行提取和分类。

然而，字符分割本身是一项非常具有挑战性的任务，容易受到图像中不均匀的光照、阴影和噪声的影响。它对车牌识别有立竿见影的影响。即使我们有一个很强的识别器，如果分割不正确，车牌也不能被正确识别。随着深度神经网络的发展，人们提出了无需分割直接识别整个车牌的方法。分割和光学字符识别联合使用隐马尔可夫模型(HMM)，其中最可能的标签序列由维特比算法确定。车牌识别被认为是一个序列标注问题。利用卷积神经网络(CNNs)滑动窗口的方法从车牌包围盒中提取特征向量序列。采用具有连接性时态分类(CTC)的递归神经网络(RNN)对序列数据进行无字符分离的标注。

3 模型介绍

与上面提到的方法不同，我们的方法使用单个深层网络来解决检测和识别问题。如图 1 所示，我们的模型由多个用于提取车牌区分特征的卷积层、一个专门为车牌量身定做的区域建议网络、一个感兴趣区域(ROI)汇聚层、用于车牌检测和包围盒回归的多层感知器以及用于车牌识别的带 CTC 的 RNN 组成。采用这种结构，只需一个网络，对输入图像进行一次前向评估，就可以同时实现车牌检测和识别。此外，整个网络进行了端到端的训练，定位损失和识别损失都得到了联合优化，表现出更好的性能。在以下小节中，我们将对每个组件进行详细说明。

3.1 模型结构

1)低层特征提取：采用 VGG-16 网络进行 CNN 低层特征提取。VGG-16 由 13 层 3×3 卷积、紧随其后的整流线性单元(RELU)非线性、5 层 2×2 最大合并和完全连接层组成。这里，我们保留了所有的卷积层，而放弃了完全连通的层，因为我们需要每个位置的局部特征来进行车牌检测。考虑到车牌与整个图像大小相比较小，我们使用 2 层而不是 5

层，以防车牌的特征信息在拼接后消失。因此，生成的特征地图是原始输入图像的四分之一大小。分辨率较高的特征图有利于小目标的检测。它们被用作检测和识别的基础。

2)车牌提取方案：Ren 等，设计了一种区域建议网络(RPN)进行目标检测，能够在图像中生成候选目标。RPN 是一种完全卷积网络，它以底层卷积特征为输入，输出一组潜在的包围盒。它可以进行端到端的培训，这样就可以生成高质量的建议区域。在这项工作中，我们略微修改了 RPN，使其适用于汽车车牌的提取。

根据我们数据集中车牌的比例和纵横比，我们设计了 6 个比例(高度分别为 5、8、11、14、17、20)，其纵横比(宽/高=5)使得输入特征图的每个位置都有 k = 6 个锚点。此外，受 Inestation-RPN 的启发，我们使用两个 256 维的矩形卷积滤波器(W_1= 5, H_1= 3 和 W_2= 3, H_2= 1)来代替通常使用的一个大小为 3×3 的滤波器。这两个卷积滤波器同时应用于每个滑动位置。提取的局部特征沿通道轴级联，形成 512 维特征向量，然后馈送到两个单独的全卷积层中，用于车牌/非车牌分类和包围盒回归。一方面，这些矩形滤镜更适合长宽比较大的对象(车牌)。另一方面，拼接后的特征既保留了局部信息，又保留了上下文信息，有利于车牌分类。

对于特征地图上每个滑动位置上的 k 个锚点，车牌分类层输出 $2k$ 个分数，其指示锚作为车牌或不是车牌的概率。边界框回归层输出 $4k$ 值，这些值是锚框到附近地面实值的偏移。给定中心在(x_a, y_a)，宽度 w 和高度 ha 的锚点，回归层输出 4 个标量(t_x, t_y, t_w, t_h)，它们是尺度不变的平移和对数空间的高度/宽度移位。回归后的边界框由下式决定：

$$x = x_a + t_x w_a, y = y_a + t_y h_a,$$
$$w = w_a \exp(t_w), h = h_a \exp(t_h),$$

其中 x，y 是回归后边界框的中心坐标，w，h 是其宽度和高度。

对于大小为 M×N 的卷积特征映射，总共有 M ×N ×k 个锚点。这些锚是冗余的，彼此高度重叠。此外，消极的锚点比积极的锚点要多得多，如果我们把所有这些锚点都用上，就会导致训练中的偏差。我们从一张图片中随机抽取 256 个锚点作为小批量样本，其中正负锚点的比例为 1：1，其中 IOU 得分大于 0.7 的锚点与任意地真边界框相交的锚点被选为正锚点，而 IOU 小于 0.3 的锚点被选为负锚点，其中 IOU 得分大于 0.7 的锚点被选为正锚点，IOU 小于 0.3 的主播被选为负锚点。IoU 分数最高的锚点也被视为正面，从而确保每个 Ground Truth 盒子至少有一个正面主播。如果没有足够的正向锚，我们就用负向锚填充。

包围盒分类采用二元 Logistic 损失，盒回归采用平滑 L1 损失。用于训练 RPN 的多任务损失函数为

$$L_{RPN} = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} L_{cls}(p_i, p_i^{\star}) + \frac{1}{N_{reg}} \sum_{i=1}^{N_{reg}} L_{reg}(\mathbf{t}_i, \mathbf{t}_i^{\star}), \quad (1)$$

其中，N_{cls}是小批次的大小，N_{reg}是该批次中的正锚点数量。边界框回归仅适用于正向锚定，因为没有与负向锚定匹配的地面事实边界框。p_i是锚点 i 成为车牌的预测概率，p_i^*是对应的地面真实标签(1 表示正锚点，0 表示负锚点)。t_i是锚点 i 的预测坐标偏移量$(t_{i,x}t_{i,y}t_{i,w}, t_{i,h})$，$t_i^*$是锚点 i 相对于地面真实的相关偏移量。RPN 采用反向传播和随机梯度下降(SGD)进行端到端训练。

在测试时，RPN 的正向评估将产生具有客观性分数和包围盒偏移量的 $M \times N \times k$ 锚。我们采用非最大抑制法(NMS)，根据预测得分选出 100 个置信度较高的提案进行后续处理

3)方案处理和汇集：如前所述，从 $M \times N \times k$ 个锚中抽取 256 个锚来训练 RPN。经过包围盒回归后，这 256 个样本稍后将用于车牌检测和识别。

我们将包围盒样本表示为 p = $(x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)})$，其中$(x^{(1)}, y^{(1)})$是包围盒的左上角坐标，$(x^{(2)}, y^{(2)})$是包围盒的右下角坐标。对于所有的积极建议$p_{i,j}$= $(x_{i,j}^{(1)}, y_{i,j}^{(1)}, x_{i,j}^{(2)}, y_{i,j}^{(2)})$, $i = 1, \ldots, n$ 与同一地面真值车牌g_j相关联的更大的边界框b_j = $(x_j^{(1)}, y_j^{(1)}, x_j^{(2)}, y_j^{(2)})$,其包含所有提案$p_{i,j}$, 即，

$$x_j^{(1)} = \min_{i=1,\ldots,n}(x_{i,j}^{(1)}), \qquad y_j^{(1)} = \min_{i=1,\ldots,n}(y_{i,j}^{(1)}),$$
$$x_j^{(2)} = \max_{i=1,\ldots,n}(x_{i,j}^{(2)}), \qquad y_j^{(2)} = \max_{i=1,\ldots,n}(y_{i,j}^{(2)}).$$

构造的包围盒$b_j, j = 1, \ldots, m$ 将作为阳性样本用于以后的车牌检测和识别。为避免正负样本分布不平衡带来的偏差，我们从 256 个样本中随机抽取 3m 个负样本，组成 4m 个样本的小批量。

考虑到样本大小不同，为了与车牌检测网络和识别网络对接，本文采用 ROI 混合的方法提取固定大小的特征表示。每个感兴趣区域被投影到图像卷积特征映射中，并产生大小为$H' \times W'$的特征映射。然后，将不同大小的特征图$H' \times W'$划分为 $X \times Y$ 网格，其中边界像素通过四舍五入对齐。每个网格中的要素最多集中在一起。由于后续的车牌识别任务，这里我们选择 X=4 和 Y=20，而不是中使用的 7×7。具体地说，由于我们需要识别车牌中的每个字符，如果我们横向保留更多的特征会更好。然而，从这一层到下一个完全连通的层的模型尺寸 p 与 X 和 Y 密切相关，即 $p \propto XY$。较大的特征地图大小会导致更多的参数，增加计算负担。考虑到车牌的纵横比，我们采用较大的宽度 Y=20 和较小的高度 X=4。实验结果表明，这些特征足以用于分类和识别。

4)车牌检测网络：车牌检测网络的目标是判断提出的感兴趣区域是否为汽车车牌，

并对车牌包围盒的坐标进行细化。

该方法采用两层完全连通的 2048 个神经元，dropout 率为 0.5，提取区分特征进行车牌检测。来自每个 ROI 的特征被展平成矢量，并通过两个完全连接的层。然后，编码后的特征被同时馈送到两个单独的线性变换层中，分别用于车牌分类和包围盒回归。车牌分类层有 2 个输出，表示每个 ROI 作为车牌/非车牌的 Softmax 概率。车牌回归层为每个建议生成边界框坐标偏移量，就像在区域建议网络中一样。

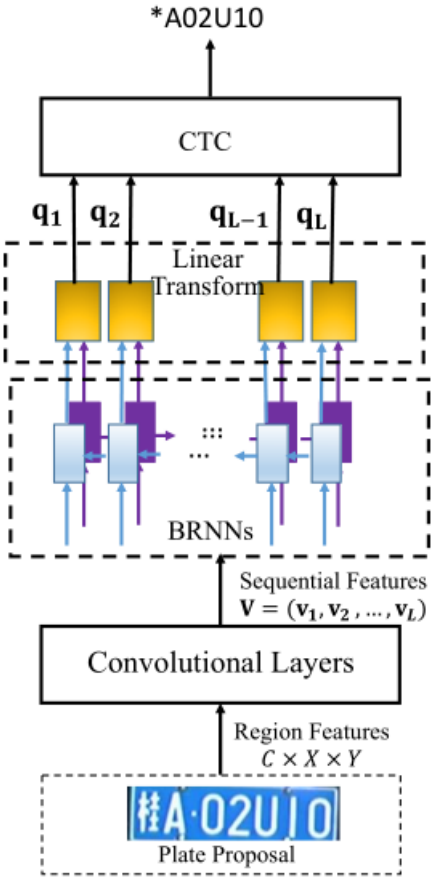


图 2 车牌识别网络。将汇集的区域特征作为特征序列，用 BRNN 进行编码，捕捉两侧的上下文信息。CTC 用于无字符分隔的车牌字符解码。

5)车牌识别网络：车牌识别网络的目标是根据提取的区域特征识别感兴趣区域中的每一个字符。为了避免字符分割这一具有挑战性的任务，我们将车牌识别看作一个序列标注问题。具有 CTC 丢失的双向 RNN(BRNN)被用来标记顺序特征，如图 2 所示。

ROI 合并后的区域特征表示为 $Q \in R^{C\times X\times Y}$，其中 C 为通道大小。首先，我们用 ReLU 增加了两个额外的卷积层。它们都使用 512 过滤器。核大小分别为 3 和 2，在第一卷积层中使用填充 1。在它们之间采用k_W=1 和k_H=2 的矩形合并窗口，这将有利于识别窄形状字符，如 1 和 I。这些操作将区域特征 Q 改造成一个大小为 $D \times L$ 的序列，其中 $D = 512$，$L = 19$。我们将得到的特征记为 $V = (v_1, v_2, \ldots, v_L)$，其中$v_i \in R^D$。

然后在顺序特征之上应用 BRNN。如图 2 所示，使用了包含 512 个单元的两个独立的 RNN 层。前向处理特征序列，隐藏状态通过$h_{\mathrm{t}}^{(\mathrm{f})}$ = g(v_1,$h_{\mathrm{t-1}}^{(\mathrm{f})}$)更新。另一种是反向处理，通过$h_{\mathrm{t}}^{(\mathrm{b})}$= g($v_{\mathrm{t}}$,$h_{\mathrm{t+1}}^{(\mathrm{b})}$)更新隐藏状态。这两个隐藏状态被串联在一起，并馈送到具有 37 个输出的线性变换。跟随 Softmax 层将 37 个输出转换成概率，概率对应于 26 个大写字母、10 个数字和一个特殊的非字符类的分布。

我们记录每个时间步的概率。因此，在 BRNNS 编码之后，特征序列 V 被变换为概率估计序列 q = ($q_1, q_2, \ldots, q_{\mathrm{L}}$)与 V.BRNN 具有相同的长度，可以从两个方向捕捉到丰富的上下文信息，这将使字符识别更加准确。为了克服传统 RNN 训练过程中梯度消失或爆炸的缺点，本文采用了长短时记忆(LSTM)算法。它定义了一种称为存储单元的新单元结构，以及三个可选择性地长期存储信息的乘法门(即输入门、遗忘门和输出门)。

然后在此采用 CTC 层进行序列译码，即通过 BRNN 的输出序列 q，找到概率最大的近似最优路径π^*，

$$\pi^* \approx \mathcal{B}\left(\arg\max_{\pi} P(\pi|\mathbf{q})\right). \tag{2}$$

这里，路径 π 是基于 BRNN 的输出激活的标签序列，并且 $P(\pi|\mathrm{q})=\prod_{t=1}^{L} P(\pi_t|q)$。运算符 B 被定义为从路径中移除重复标签和非字符标签的操作。例如，$B(a-a-b-)=B(-aa--a-bb)=(aab)$，最优标签序列$\pi^*$正好是识别出的车牌标签。

3.2 损失函数与训练

如前所述，在训练时间内，整个网络将图像、车牌包围盒和相关标签作为输入。在获取样本和区域特征后，结合损失项进行车牌检测和识别，并对检测和识别网络进行联合训练。因此，多任务损失函数被定义为

$$\begin{aligned} L_{DRN} = & \frac{1}{N}\sum_{i=1}^{N} L_{cls}(p_i, p_i^{\star}) + \frac{1}{N_+}\sum_{i=1}^{N_+} L_{reg}(\mathbf{t}_i, \mathbf{t}_i^{\star}) \\ & + \frac{1}{N_+}\sum_{i=1}^{N_+} L_{rec}(\mathbf{q}^{(i)}, \mathbf{s}^{(i)}) \end{aligned} \tag{3}$$

其中 N 是用于检测网络的小批次的大小，N_+是该批次中阳性样本的数量。L_{cls}和L_{reg}的定义与 RPN 中使用的定义相同。P_i, P_i^*, t_i, t_i^*也使用与 RPN 中使用的定义相同的定义。$s^{(\mathrm{i})}$是样本 i 的地面真值车牌标号，$q^{(\mathrm{i})}$是 BRNN 对应的输出序列。

观察到 BRNNs 的输出长度$q^{(\mathrm{i})}$与目标标签$s^{(\mathrm{i})}$的长度不一致。在 CTC 丢失之后，车牌识别的目标函数被定义为网络输出正确标签的负对数概率，即，

$$L_{rec}(\mathbf{q}^{(i)}, \mathbf{s}^{(i)}) = -\log P(\mathbf{s}^{(i)}|\mathbf{q}^{(i)}) \tag{4}$$

$$P(\mathbf{s}^{(i)}|\mathbf{q}^{(i)}) = \sum_{\pi:\mathcal{B}(\pi)=\mathbf{s}^{(i)}} P(\pi|\mathbf{q}^{(i)}) \tag{5}$$

它是 B 可以映射到 $s^{(i)}$的所有 π 的概率之和。

我们使用近似的联合训练过程来训练整个网络，忽略关于所提议的盒子坐标的导数。幸运的是，这对性能没有太大影响。我们使用 SGD 对整个网络进行训练。用于提取低层特征的 CNN 是从预先训练的 VGG-16 模型中初始化的。我们不会为了效率而微调前四个卷积层。CNN 的其余层仅在最初的 50K 迭代中进行了微调。其他权值根据高斯分布进行初始化。为了进行优化，我们使用 ADAM，对于预先训练的 VGG-16 模型中的参数，初始学习率为10^{-5}，对于其他参数，初始学习率为10^{-4}。后一种学习速率每 10K 迭代减半，直到10^{-5}。该网络经过了 200K 次迭代的训练。每次迭代使用从训练数据集中随机采样的单个图像。对于每个训练图像，我们将其调整为 700 像素的较短一侧，而较长的一侧不超过 1500 像素。

4 实验

在这一部分中，我们通过实验来验证所提出的方法的有效性。我们的网络是用 Torch7 实现的。实验是在 12 GB 内存的 NVIDIA Titan X GPU 上进行的。

4.1 数据集

本文使用了三个数据集来评估我们提出的方法的有效性。

第一个数据集由来自中国的车牌组成，标记为“CarFlag-Large”。我们收集了 45 万幅图像用于训练，7378 幅图像用于测试。这些图像是由固定监视摄像机在不同天气和光照条件下(如晴天、雨天或夜间)从正面视角拍摄的，分辨率为 1600×2048。这些车牌几乎是水平的。只有图像中最近的车牌才会在地面实况文件中标注。

第二个数据集是面向应用的车牌(AOLP)数据库。它总共有 2049 张带有中国台湾车牌的图像。该数据库分为三个不同难度和拍摄条件的子集：门禁(AC)、交通执法(LE)和道路巡逻(RP)。由于我们没有任何其他中国台湾车牌的图像，为了训练网络，我们使用来自不同子数据集的图像分别进行训练和测试。例如，我们使用来自 LE 和 RP 子集的图像来训练网络，并在 AC 子集上评估其性能。考虑到训练图像数量较少，采用旋转和仿射变换进行数据增强。

第三个数据集由 Yuan 等人发布，命名为“PKUData”。它有 3977 张从各种场景捕捉到的带有中国车牌的图像。它被分为 5 组(即 G1-G5)，对应于不同的配置。但是，只有地面实况文件中给出的车牌边界框。因此，我们只对该数据集的检测性能进行评估。

图 3 利用我们的联合训练模型进行开放式宽幅车牌检测和识别的实例结果。第一行图像来自 CarFlag-Large，第二行图像来自 AOLP，第三行图像来自 PKUData。实验结果表明，该模型能够在白天、黑夜、晴雨天等多种拍摄条件下进行车牌检测和识别。

4.2 评价标准

为了评估“端到端”的性能，同时考虑到检测和识别结果，我们遵循了自然场景中一般文本检测的“端到端”评估协议，因为它们都有相似的应用场景。将 IoU 定义为:

$$\mathrm{IoU} = \frac{\mathrm{area}(R_{\mathrm{det}} \cap R_{\mathrm{gt}})}{\mathrm{area}(R_{\mathrm{det}} \cup R_{\mathrm{gt}})} \tag{6}$$

其中R_{det}和R_{gt}分别是检测到的包围盒和地面真值的区域。

如果包围盒与地面真实包围盒的 IoU 超过 50%(IOU>0.5)，并且车牌标签匹配，则认为包围盒是正确的。应该注意，由于 CarFlag-Large 中训练图像都来自一个省并且使用相同的汉字，所以我们将车牌中的所有汉字表示为‘*’。训练后的网络不能用于区分其他汉字。

至于只检测的性能，我们遵循公平竞争的标准，即如果车牌完全被包围盒包围，并且 IOU>0.5，则认为检测是正确的。

4.3 CarFlag-Large 的性能评价

在这一部分，我们想要展示我们的端到端联合培训框架与常用的两阶段方法相比的优越性。如图 4 所示，常用的两阶段方法通过两个独立的模型实现车牌检测和识别。首先进行车牌检测。检测到的对象被裁剪掉，然后由另一个不同的模型识别。相反，我们提出的网络同时输出检测和识别结果，只需一次前向传递，不需要图像裁剪。卷积特征被检测和识别共享，省去了特征的重新计算。为了简单起见，我们将我们的联合训练网络表示为“我们的(联合训练)”，将两阶段的方法表示为“我们的(两阶段)”。仅用于车牌检测的模型被表示为“OURS(仅限检测)”。

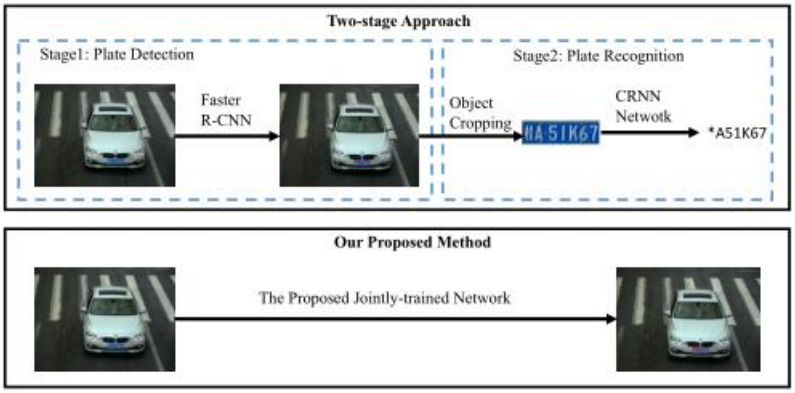


图 4 两阶段方法与我们建议的方法。在两步法中，先用更快的 R-CNN 进行车牌检测，然后将检测到的车牌从图像中裁剪出来，然后用另一个独立的模型(本文中的 CRNN)进行识别。在识别阶段需要重新计算特征。相反，我们提出的网络以图像为输入，在 OneShot 中生成车牌边界框和车牌标签。它避免了图像裁剪等中间过程，并共享卷积特征提取的计算量。

为了公平竞争，我们仅使用用于车牌检测的 450K 训练图像来训练更快的 R-CNN 模型。我们修改了本文中使用的锚的比例和形状，使其适合车牌。使用相同的初始参数和学习率，网络还被训练了 200K 次。在车牌识别方面，我们采用了 CRNN 框架，该框架在一般文本识别方面具有最先进的性能。它是一个端到端的裁剪单词识别框架，包括 CNN 层、RNN 层和 CTC 层，从下到上进行转录。我们从 450000 幅训练图像中裁剪出真实车牌，并将它们的大小调整到 160×32 像素。然后利用这些训练数据对 CRNN 模型进行微调。

为了提高性能，我们在测试阶段对我们提出的网络和只检测的更快的 R-CNN 网络将输入图像重新缩放到多个大小。输入图像的大小分别调整为 600、1200 像素的较短一侧，而较长的一侧小于 1500 像素。在我们的框架下，检测和识别结果一起出来，而在两阶段方法中，我们从输入图像中裁剪检测到的边界框，将它们调整到 160×32 个像素，然后输入到训练好的 CRNN 模型中进行识别。只有分类分数大于 0.95 的包围盒才会通过 NMS 保留和合并。考虑到每幅图像只有一个标记为地面真实的车牌，我们最终选择识别出 7 个字符和/或检测得分最高的车牌进行评价。测试结果如表 1 所示。我们的联合训练网络在 7378 幅测试图像上给出了 96.13%的端到端性能。它比两阶段方法的结果高出 2%左右，这表明在统一的网络中端到端训练对于检测和识别都是有利的。学习到的功能信息更丰富，这两个子任务可以相互帮助。

表 1 在 CARFLAG-Large 数据集上的实验结果。我们用两阶段基线方法比较了我们联合训练网络的性能和运行速度。联合训练网络不仅在检测和“端到端”性能上实现了更高的精度，而且在时间内也达到了更高的精度。

Method	End-to-end Performance (%)	Detection-only Performance (%)	Speed(per image single scale) (ms)
Ours(Jointly-trained)	96.13	98.15	300
Ours(Two-stage)	94.09	97.00	450

在计算速度上，统一框架在单个小输入尺度上进行前向评估时，每幅图像大约需要 300ms，而两阶段方法需要大约 450ms 才能获得检测和识别结果，因为它需要执行图像裁剪和 CNN 特征重新计算。

我们还比较了仅检测性能。我们的联合训练网络的检测准确率为 98.15%，比只检测的更快的 R-CNN 网络的检测准确率提高了 1%。这一结果表明，在训练时间内使用多任务损失可以提高车牌检测的效率。

图 3 的第一行给出了使用我们的联合训练网络的一些实验结果，这表明我们的模型可以处理不同光照条件下的图像。

4.3 AOLP 的性能评价

在本节中，我们将在 AOLP 数据集上将我们的方法与其他最先进的方法的“端到端”性能进行比较。请注意，由于该数据集中的训练图像数量较少，因此网络仅用 15K 次迭代进行训练。此外，由于 AOLP 中车牌的大小几乎相同，车牌与图像大小的比例也是相似的。对于这个数据集，我们在测试阶段只使用了一个短边为 700 像素的单一图像比例。

检测和识别结果显示在图 3 的第二行。与表 2 中其他方法的比较结果表明，我们的方法在具有端到端评估的 AC 和 LE 子集上表现得更好。在所有三个子集上的车牌检测性能也是最好的，平均比 Li 等人使用的基于滑动窗口的方法高 2%，比 Hsu 等人使用的基于边缘的方法高 4%。就计算速度而言，OUT 网络获得检测和识别结果大约需要 400ms，而 Li 等人的方法需要 2−3s，Hsu 等人的方法平均需要 260ms。

需要注意的是，在表 2 中，RP 子集上的“端到端”性能比中差。这可能是因为 RP 中的车牌有很大的旋转和投影方向。对检测到的车牌进行裁剪，并利用 Hough 变换进行方向校正。相反，我们的方法没有显式地处理旋转车牌。将空间转换网络集成到我们的端到端框架中可能是一种解决方案，这是未来的工作。

4.4 PKUData 的性能评价

由于 PKUData 中的地面真实数据文件只提供了车牌边界框，因此我们只对该数据集的检测性能进行了评估。将该方法的检测精度和计算效率与其他方法进行了比较。我们使用由 CarFlag-Large 数据集训练的相同模型，因为它们都是具有中国车牌的数据集。

图 3 第三行上的图像显示了具有检测和识别结果的示例。我们的方法和其他三种方法的检测结果如表 III 所示。我们的联合训练模型在所有 5 个子集上都表现出了绝对的优势，特别是在 G4 上，我们达到了 100%的检测率。实验结果证明了该方法在不同场景和不同条件下的鲁棒性。定性地，我们的联合训练网络达到了 99.80%的平均检测率，比以前最好的性能方法提高了 2%。

此外，如表 3 所示，我们的联合训练网络的检测性能略好于只检测网络，这与在 CarFlag-Large 数据集上的结果是一致的，并再次证明了使用标签信息训练可以提高检测性能。

在计算速度上，Yuan 等人的方法比我们的方法要快，因为他们使用简单的线性支持向量机，而我们使用的是深度 CNN 和 RNN。

表 2 在 AOLP 数据集上的实验结果。AC(访问控制)是最简单的数据集，当车辆通过低速或停车的固定通道时，可以捕捉到图像。LE(执法)数据集由路边摄像头在车辆违反交通法规时拍摄到的图像组成。RP(道路巡逻)是指将摄像机放在巡逻车辆上，以任意视点和距离拍摄图像的情况。在性能和运行速度上，我们将我们提出的方法与其他最先进的方法进行了比较。我们的联合训练网络对车牌处于接近水平位置的图像显示出更好的性能。

Method	End-to-end Performance(%)			Detection-only Performance(%)			Speed (per image single scale) (ms)
	AC	LE	RP	AC	LE	RP	
Hsu et al.	-	-	-	96	95	94	260
Li et al.	94.85	94.19	88.38	98.38	97.62	95.58	1000-2000
Ours(Jointly-trained)	95.29	96.57	83.63	99.56	99.34	98.85	400

表 3 在 PKUDATA 上的实验结果。将本文提出的方法与其他先进方法的检测性能和运行速度进行了比较。G1-G5 对应不同的图像捕获条件。我们的联合训练网络达到了 99.80%的平均检测率，比以前最好的性能方法提高了 2%。此外，联合训练的网络综合了检测和识别损失，比仅用检测信息训练的网络具有更好的性能。

Method	Detection Performance(%)						Speed (per image single scale) (ms)
	G1	G2	G3	G4	G5	Average	
Zhou et al.	95.43	97.85	94.21	81.23	82.37	90.22	475
Li et al.	98.89	98.42	95.83	81.17	83.31	91.52	672
Yuan et al.	98.76	98.42	97.72	96.23	97.32	97.69	42
Ours(Detection-only)	99.88	99.71	99.87	99.65	98.81	99.58	300
Ours(Jointly-trained)	99.88	99.86	99.87	100	99.38	99.80	300

5 结论

在本文中，我们提出了一个联合训练的网络，用于同时进行车牌检测和识别。有了

这个网络，汽车车牌的检测和识别可以一次过一次完成，准确率和效率都很高。通过与检测和识别网络共享卷积特征，大大减小了模型的规模。整个网络可以端到端地训练，而不需要像图像裁剪或字符分离这样的中间处理。对三个不同方法的数据集进行综合评价和比较，验证了该方法的优越性。未来，我们将把我们的网络扩展到多方向的汽车车牌。此外，通过时间分析发现，NMS 占用了整个处理时间的一半左右。因此，我们将对前向通道进行优化，以加快处理速度。

arXiv:1709.08828v1 [cs.CV] 26 Sep 2017

Towards End-to-End Car License Plates Detection and Recognition with Deep Neural Networks

Hui Li, Peng Wang†, and Chunhua Shen

***Abstract*—In this work, we tackle the problem of car license plate detection and recognition in natural scene images. We propose a unified deep neural network which can localize license plates and recognize the letters simultaneously in a single forward pass. The whole network can be trained end-to-end. In contrast to existing approaches which take license plate detection and recognition as two separate tasks and settle them step by step, our method jointly solves these two tasks by a single network. It not only avoids intermediate error accumulation, but also accelerates the processing speed. For performance evaluation, three datasets including images captured from various scenes under different conditions are tested. Extensive experiments show the effectiveness and efficiency of our proposed approach.**

***Index Terms*—Car plate detection and recognition, Convolutional neural networks, Recurrent neural networks**

I. Introduction

AUTOMATIC car license plate detection and recognition plays an important role in intelligent transportation systems. It has a variety of potential applications ranging from security to traffic control, and attracts considerable research attentions during recent years.

However, most of the existing algorithms only work well either under controlled conditions or with sophisticated image capture systems. It is still a challenging task to read license plates accurately in an uncontrolled environment. The difficulty lies in the highly complicated backgrounds, like the general text in shop boards, windows, guardrail or bricks, and random photographing conditions, such as illumination, distortion, occlusion or blurring.

Previous works on license plate detection and recognition usually consider plate detection and recognition as two separate tasks, and solve them respectively by different methods. However, the tasks of plate detection and recognition are highly correlated. Accurate bounding boxes obtained via detection method can improve recognition accuracy, while the recognition result can be used to eliminate false positives vice versa. Thus in this paper, we propose a unified framework to jointly tackle these two tasks at the same level. A deep neural network is designed, which takes an image as input and outputs the locations of license plates as well as plate labels simultaneously, with both high efficiency and accuracy. We prove that the low level features can be used for both detection and recognition. The whole network can be trained end-to-end, without using any heuristic rule. An overview of the network architecture is shown in Figure 1. To our knowledge, this is the first work that integrates both license plate detection and recognition into a single network and solves them at the same time. The main contributions of this work are as follows:

- A single unified deep neural network which can detect license plates from an image and recognize the labels all at once. The whole framework involves no heuristic processes, such as the use of plate colors or character space, and avoids intermediate procedures like character grouping or separation. It can be trained end-to-end, with only the image, plate positions and labels needed for training. The resulting system achieves high accuracy on both plate detection and letter recognition.
- Secondly, the convolutional features are shared by both detection and recognition, which leads to fewer parameters compared to using separated models. Moreover, with the joint optimization of both detection and recognition losses, the extracted features would have richer information. Experiments show that both detection and recognition performance can be boosted via using the jointly trained model.
- By integrating plate recognition directly into the detection pipeline, instead of addressing them by separate models, the resulting system is more efficient. With our framework, we do not need to crop the detected license plates from the input image and then recognize them by a separate network. The whole framework takes $0.3 - 0.4$ second per image on a Titan X GPU.

The rest of the paper is organized as follows. Section 2 gives a brief discussion on related work. Section 3 presents the integrated method, and introduces each part in detail. Experimental verifications are followed in Section 4, and conclusions are drawn in Section 5.

II. Related work

As license plate detection and recognition are generally addressed separately, we give a brief introduction to previous work on each aspect respectively.

1) License Plate Detection: License plate detection aims to localize the license plates in the image in the form of bounding boxes. Existing methods can be roughly classified into four categories [1], [2], [3]: edge-based, color-based, texture-based, and character-based.

Since license plates are normally in a rectangular shape with a specific aspect ratio, and they present higher edge density than elsewhere in the image, edge information is used widely

H. Li, and C. Shen are with the School of Computer Science, The University of Adelaide, SA 5005, Australia. P. Wang is with the School of Computer Science, Northwestern Polytechnical University, China. †P. Wang is the corresponding author (E-mail: peng.wang@nwpu.edu.cn). This work was done when P. Wang was with The University of Adelaide.

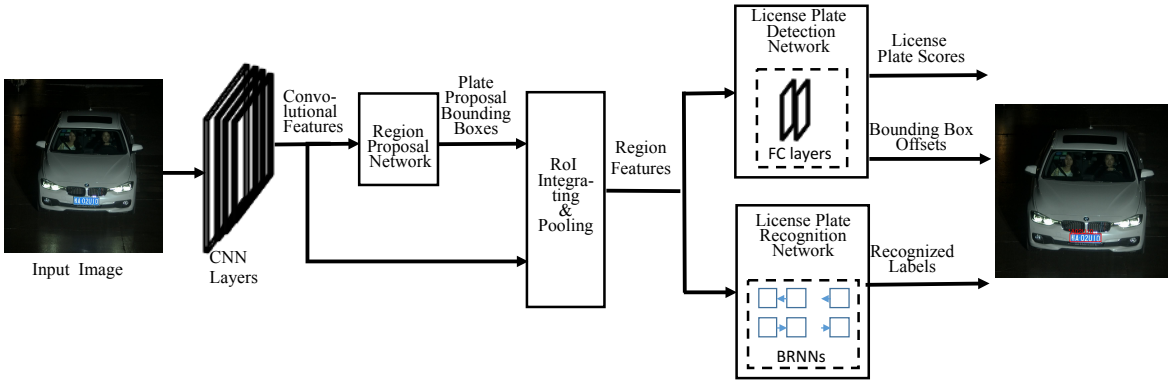


Fig. 1. The overall structure of our model. It consists of several convolutional layers, a region proposal network for license plate proposals generation, proposal integrating and pooling layer, multi-layer perceptrons for plate detection and bounding box regression, and RNNs for plate recognition. Given an input RGB image, with a single forward evaluation, the network outputs scores of predicted bounding boxes being license plates, bounding box offsets with a scale-invariant translation and log-space height/width shift relative to a proposal, as well as the recognized license plate labels at the same time. The extracted region features are used by both detection and recognition, which not only shares computation, but also reduces model size.

to detect license plates. In [4] an edge-based method was developed for plate detection. Expectation Maximization (EM) was applied for edge clustering which extracts the regions with dense sets of edges and with shapes similar to plates as the candidate license plates. In [5], a novel line density filter approach was proposed to connect regions with high edge density and remove sparse regions in each row and column from a binary edge image. Edge-based methods are fast in computation, but they cannot be applied to complex images as they are too sensitive to unwanted edges.

Color-based approaches are based on the observation that color of the license plate is usually different from that of the car body. In [6], a plate detection method was developed by analyzing the target color pixels. A color-geometric template was utilized to localize Iranian license plates via strip search. Chang *et al*. [7] proposed a method to detect Taiwan license plates in RGB images based on the different foreground and background colors. They developed a color edge detector which is sensitive to black-white, red-white and green-white edges. Color-based methods can be used to detect inclined or deformed license plates. However, they are very sensitive to various illumination conditions in natural scene images, and they cannot distinguish other objects in the image with similar color and size as the license plates.

Texture-based approaches attempted to detect license plates according to the unconventional pixel intensity distribution in plate regions. Yu *et al*. [8] used a wavelet transform at first to get the horizontal and vertical details of an image. Empirical Mode Decomposition (EMD) analysis was then employed to deal with the projection data and locate the desired wave crest which indicates the position of a license plate. Giannoukos *et al*. [9] developed a Sliding Concentric Window (SCW) algorithm to identify license plates based on the local irregularity property of plate texture in an image. Operator Context Scanning (OCS) was proposed to accelerate detection speed. Texture-based methods use more discriminative characteristics than edge or color, but result in a higher computational complexity.

Considering the fact that license plates consist of a string of characters, much work appeared recently based on the character-based feature as it includes more specific information. Zhou *et al*. [2] formulated license plate detection as a visual matching problem. Principal Visual Word (PVW) was generated for each character which contains geometric clues such as orientation, characteristic scale and relative position, and used for plate extraction. Li *et al*. [10] applied Maximally Stable Extremal Region (MSER) at the first stage to extract candidate characters in images. Conditional Random Field (CRF) was then constructed to represent the relationship among license plate characters. License plates were finally localized through the belief propagation inference on CRF. Character-based methods are more reliable and can lead to a high recall. However, the performance is affected largely by other text in the image background.

2) License plate recognition: Previous work on license plate recognition typically segments characters in the license plate firstly, and then recognizes each segmented character using Optical Character Recognition (OCR) techniques. For example, In [11], Extremal Regions (ER) were employed to segment characters from coarsely detected license plates and to refine plate location. Restricted Boltzmann machines were applied to recognize the characters. In [4], MSER was adopted for character segmentation. Local Binary Pattern (LBP) features were extracted and classified using a Linear Discriminant Analysis (LDA) classifier for character recognition.

However, character segmentation by itself is a really challenging task that is prone to be influenced by uneven lighting, shadow and noise in the image. It has an immediate impact on plate recognition. The plate cannot be recognized correctly if the segmentation is improper, even if we have a strong recognizer. With the development of deep neural networks, approaches were proposed to recognize the whole license plate directly with segmentation free. In [12], segmentation and optical character recognition were jointly performed using Hidden Markov Models (HMMs) where the most likely label sequence was determined by Viterbi algorithm. In [13], plate recognition was regarded as a sequence labeling problem. Convolutional Neural Networks (CNNs) was employed in

a sliding window manner to extract a sequence of feature vectors from license plate bounding box. Recurrent Neural Networks (RNNs) with Connectionist Temporal Classification (CTC) [14] were adopted to label the sequential data without character separation.

III. Model

Different from the above-mentioned methods, our approach addresses both detection and recognition using a single deep network. As illustrated in Figure 1, our model consists of a number of convolutional layers to extract discriminate features for license plates, a region proposal network tailored specifically for car license plates, a Region of Interest (RoI) pooling layer, multi-layer perceptrons for plate detection and bounding box regression, and RNNs with CTC for plate recognition. With this architecture, the plate detection and recognition can be achieved simultaneously, with one network and a single forward evaluation of the input image. Moreover, the whole network is trained end-to-end, with both localization loss and recognition loss being jointly optimized, and shows improved performance. In the following subsections, we give a detailed description about each component.

A. Model Architecture

1) Low-level Feature Extraction: The VGG-16 network [15] is adopted here to extract low level CNN features. VGG-16 consists of 13 layers of 3×3 convolutions followed by Rectified Linear Unit (ReLU) non-linearity, 5 layers of 2×2 max-pooling, and fully connected layers. Here we keep all the convolutional layers and abandon the fully connected layers as we require local features at each position for plate detection. Given that the license plates are small compared with the whole image size, we use 2 pooling layers instead of 5, in case the feature information of license plates is vanished after pooling. So the resulting feature maps are one fourth size of the original input image. The higher-resolution feature maps will benefit the detection of small objects [16]. They are used as a base for both detection and recognition.

2) Plate Proposal Generation: Ren *et al*. [17] designed a Region Proposal Network (RPN) for object detection, which can generate candidate objects in images. RPN is a fully convolutional network which takes the low-level convolutional features as input, and outputs a set of potential bounding boxes. It can be trained end-to-end so that high quality proposals can be generated. In this work, we modify RPN slightly to make it suitable for car license plate proposal.

According to the scales and aspect ratios of license plates in our datasets, we designed 6 scales (the heights are respectively 5, 8, 11, 14, 17, 20) with an aspect ratio (width/height $= 5$), which results in $k = 6$ anchors at each position of the input feature maps. In addition, inspired by inception-RPN [18], we use two 256-d rectangle convolutional filters ($W_1 = 5, H_1 = 3$ and $W_2 = 3, H_2 = 1$) instead of the regularly used one filter size 3×3. The two convolutional filters are applied simultaneously across each sliding position. The extracted local features are concatenated along the channel axis and form a 512-d feature vector, which is then fed into two separate fully convolutional layers for plate/non-plate classification and box regression. On one hand, these rectangle filters are more suitable for objects with larger aspect ratios (license plates). On the other hand, the concatenated features keep both local and contextual information, which will benefit the plate classification.

For k anchors at each sliding position on the feature map, the plate classification layer outputs $2k$ scores which indicate the probabilities of the anchors as license plates or not. The bounding box regression layer outputs $4k$ values which are the offsets of anchor boxes to a nearby ground-truth. Given an anchor with the center at (x_a, y_a), width w_a and height h_a, the regression layer outputs 4 scalars (t_x, t_y, t_w, t_h) which are the scale-invariant translation and log-space height/width shift. The bounding box after regression is given by

$$x = x_a + t_x w_a, y = y_a + t_y h_a,$$
$$w = w_a \exp(t_w), h = h_a \exp(t_h),$$

where x, y are the center coordinates of the bounding box after regression, and w, h are its width and height.

For a convolutional feature map with size $M \times N$, there will be $M \times N \times k$ anchors in total. Those anchors are redundant and highly overlapped with each other. Moreover, there are much more negative anchors than positive ones, which will lead to bias during training if we use all those anchors. We randomly sample 256 anchors from one image as a mini-batch, where the ratio between positive and negative anchors is up to 1:1. The anchors that have Intersection over Union (IoU) scores larger than 0.7 with any ground-truth bounding box are selected as positives, while anchors with IoU lower than 0.3 as negatives. The anchors with the highest IoU scores are also regarded as positives, so as to make sure that every ground-truth box has at least one positive anchor. If there are not enough positive anchors, we pad with negative ones.

The binary logistic loss is used here for box classification, and smooth L_1 loss [17] is employed for box regression. The multi-task loss function used for training RPN is

$$L_{RPN} = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} L_{cls}(p_i, p_i^\star) + \frac{1}{N_{reg}} \sum_{i=1}^{N_{reg}} L_{reg}(\mathbf{t}_i, \mathbf{t}_i^\star), \quad (1)$$

where N_{cls} is the size of a mini-batch and N_{reg} is the number of positive anchors in this batch. Bounding box regression is only for positive anchors, as there is no ground-truth bounding box matched with negative ones. p_i is the predicted probability of anchor i being a license plate and $p_i^\star$ is the corresponding ground-truth label (1 for positive anchor, 0 for negative anchor). $\mathbf{t}_i$ is the predicted coordinate offsets $(\mathbf{t}_{i,x}, \mathbf{t}_{i,y}, \mathbf{t}_{i,w}, \mathbf{t}_{i,h})$ for anchor i, and $\mathbf{t}_i^\star$ is the associated offsets for anchor i relative to the ground-truth. RPN is trained end-to-end with back-propagation and Stochastic Gradient Descent (SGD).

At test time, the forward evaluation of RPN will result in $M \times N \times k$ anchors with objectiveness scores as well as bounding box offsets. We employ Non-Maximum Suppression (NMS) to select 100 proposals with higher confidences based on the predicted scores for the following processing.

3) Proposal Processing and Pooling: As we state before, 256 anchors are sampled from the $M \times N \times k$ anchors to train RPN. After bounding box regression, the 256 samples will later be used for plate detection and recognition.

We denote the bounding box samples as $p = (x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)})$, where $(x^{(1)}, y^{(1)})$ is the top-left coordinate of the bounding box, and $(x^{(2)}, y^{(2)})$ is the bottom-right coordinate of the bounding box. For all the positive proposals $p_{i,j} = (x_{i,j}^{(1)}, y_{i,j}^{(1)}, x_{i,j}^{(2)}, y_{i,j}^{(2)})$, $i = 1, \ldots, n$ that are associated with the same ground truth plate g_j, a bigger bounding box $b_j = (x_j^{(1)}, y_j^{(1)}, x_j^{(2)}, y_j^{(2)})$ is constructed that encompasses all proposals $p_{i,j}$, *i.e.*,

$$x_j^{(1)} = \min_{i=1,\ldots,n}(x_{i,j}^{(1)}), \qquad y_j^{(1)} = \min_{i=1,\ldots,n}(y_{i,j}^{(1)}),$$
$$x_j^{(2)} = \max_{i=1,\ldots,n}(x_{i,j}^{(2)}), \qquad y_j^{(2)} = \max_{i=1,\ldots,n}(y_{i,j}^{(2)}).$$

The constructed bounding boxes b_j, $j = 1, \ldots, m$ will then be used as positive samples for later plate detection and recognition. To avoid the bias caused by the unbalanced distribution between positive and negative samples, we randomly choose $3m$ negative ones from the 256 samples and form a mini-batch with $4m$ samples.

Considering that the sizes of the samples are different from each other, in order to interface with the plate detection network as well as the recognition network, RoI pooling [19] is adopted here to extract fixed-size feature representation. Each RoI is projected into the image convolutional feature maps, and results in feature maps of size $H' \times W'$. The varying sized feature maps $H' \times W'$ are then divided into $X \times Y$ grids, where boundary pixels are aligned by rounding. Features are max-pooled within each grid. Here we choose $X = 4$ and $Y = 20$ instead of 7×7 that is used in [19], because of the subsequent plate recognition task. To be specific, since we need to recognize each character in the license plate, it would be better if we keep more feature horizontally. However, the model size p from this layer to the next fully connected layer is closely related to X and Y, *i.e.*, $p \propto XY$. A larger feature map size will result in more parameters and increase the computation burden. Considering the aspect ratio of license plates, we use a longer width $Y = 20$ and a shorter height $X = 4$. Experimental results demonstrate that these features are sufficient for classification and recognition.

4) Plate Detection Network: Plate detection network aims to judge whether the proposed RoIs are car license plate or not, and refine the coordinates of plate bounding boxes.

Two fully connected layers with 2048 neurons and a dropout rate of 0.5 are employed here to extract discriminate features for license plate detection. The features from each RoI are flattened into a vector and passed through the two fully connected layers. The encoded features are then fed concurrently into two separate linear transformation layers respectively for plate classification and bounding box regression. The plate classification layer has 2 outputs, which indicate the softmax probability of each RoI as plate/non-plate. The plate regression layer produces the bounding box coordinate offsets for each proposal, as in region proposal network.

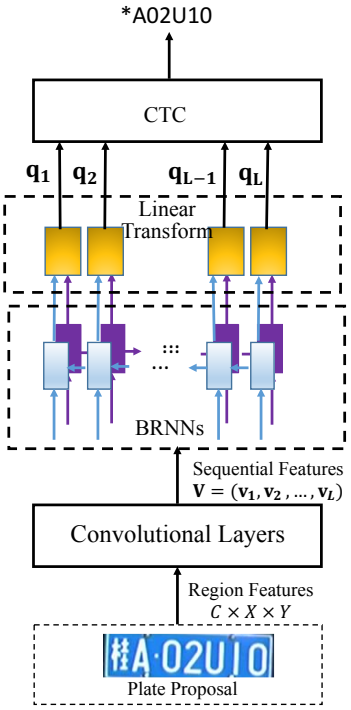


Fig. 2. Plate Recognition Network. The pooled region features are regarded as a feature sequence, and encoded by BRNNs, which capture the context information in both sides. CTC are used for plate decoding without character separation.

5) Plate Recognition Network: Plate recognition network aims to recognize each character in RoIs based on the extracted region features. To avoid the challenging task of character segmentation, we regard the plate recognition as a sequence labeling problem. Bidirectional RNNs (BRNNs) with CTC loss [14] are employed to label the sequential features, which is illustrated in Figure 2.

The region features after RoI pooling are denoted as $\mathbf{Q} \in \mathbb{R}^{C \times X \times Y}$, where C is the channel size. First of all, we add two additional convolutional layers with ReLUs. Both of them use 512 filters. The kernel sizes are 3 and 2 respectively, with a padding of 1 used in the first convolutional layer. A rectangular pooling window with $k_W = 1$ and $k_H = 2$ is adopted between them, which would be beneficial for recognizing characters with narrow shapes, such as 1 and I, referring to [20]. These operations will reform the region features $\mathbf{Q}$ to a sequence with the size as $D \times L$, where $D = 512$ and $L = 19$. We denote the resulting features as $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L)$, where $\mathbf{v}_i \in \mathbb{R}^D$.

Then BRNNs are applied on top of the sequential features. As presented in Figure 2, Two separated RNN layers with 512 units are used. One processes the feature sequence forward, with the hidden state updated via $\mathbf{h}_t^{(f)} = \mathrm{g}(\mathbf{v}_t, \mathbf{h}_{t-1}^{(f)})$. The other one processes it backward with the hidden state updated via $\mathbf{h}_t^{(b)} = \mathrm{g}(\mathbf{v}_t, \mathbf{h}_{t+1}^{(b)})$. The two hidden states are concatenated together and fed to a linear transformation with 37 outputs. Softmax layer is followed to transform the 37 outputs into probabilities, which correspond to the distributions over 26 capital letters, 10 digits, and a special non-character class.

We record the probabilities at each time step. Hence, after BRNNs encoding, the feature sequence $\mathbf{V}$ is transformed into a sequence of probability estimation $\mathbf{q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_L)$ with the same length as $\mathbf{V}$. BRNNs capture abundant contextual information from both directions, which will make the character recognition more accurate. To overcome the shortcoming of gradient vanishing or exploding during traditional RNN training, Long-Short Term Memory (LSTM) [21] is employed here. It defines a new cell structure called memory cell, and three multiplicative gates (*i.e.*, input gate, forget gate and output gate), which can selectively store information for a long time.

Then CTC layer [14] is adopted here for sequence decoding, which is to find an approximately optimal path π^* with maximum probability through the BRNNs' output sequence $\mathbf{q}$, *i.e.*,

$$\pi^* \approx \mathcal{B}\left(\arg\max_{\pi} P(\pi|\mathbf{q})\right). \tag{2}$$

Here a path π is a label sequence based on the output activation of BRNNs, and $P(\pi|\mathbf{q}) = \prod_{t=1}^{L} P(\pi_t|\mathbf{q})$. The operator $\mathcal{B}$ is defined as the operation of removing the repeated labels and the non-character label from the path. For example, $\mathcal{B}(a - a - b -) = \mathcal{B}(-aa - -a - bb) = (aab)$. Details of CTC can refer to [14]. The optimal label sequence π^* is exactly the recognized plate label.

B. Loss Functions and Training

As we demonstrate previously, the whole network takes as inputs an image, the plate bounding boxes and the associated labels during training time. After we obtain the samples as well as the region features, we combine the loss terms for plate detection and recognition, and jointly train the detection and recognition networks. Hence, the multi-task loss function is defined as

$$\begin{aligned} L_{DRN} = &\frac{1}{N}\sum_{i=1}^{N} L_{cls}(p_i, p_i^\star) + \frac{1}{N_+}\sum_{i=1}^{N_+} L_{reg}(\mathbf{t}_i, \mathbf{t}_i^\star) \\ &+ \frac{1}{N_+}\sum_{i=1}^{N_+} L_{rec}(\mathbf{q}^{(i)}, \mathbf{s}^{(i)}) \end{aligned} \tag{3}$$

where N is the size of a mini-batch used in detection network and N_+ is the number of positive samples in this batch. The definitions of L_{cls} and L_{reg} are the same as that used in RPN. $p_i, p_i^\star, \mathbf{t}_i, \mathbf{t}_i^\star$ also use the same definition as that used in RPN. $\mathbf{s}^{(i)}$ is the ground truth plate label for sample i and $\mathbf{q}^{(i)}$ is the corresponding output sequence by BRNNs.

It is observed that the length of BRNNs' outputs $\mathbf{q}^{(i)}$ is not consistent with the length of target label $\mathbf{s}^{(i)}$. Following CTC loss in [14], the objective function for plate recognition is defined as the negative log probability of the network outputting correct label, *i.e.*,

$$L_{rec}(\mathbf{q}^{(i)}, \mathbf{s}^{(i)}) = -\log P(\mathbf{s}^{(i)}|\mathbf{q}^{(i)}) \tag{4}$$

where

$$P(\mathbf{s}^{(i)}|\mathbf{q}^{(i)}) = \sum_{\pi:\mathcal{B}(\pi)=\mathbf{s}^{(i)}} P(\pi|\mathbf{q}^{(i)}) \tag{5}$$

which is the sum of probabilities of all π that can be mapped to $\mathbf{s}^{(i)}$ by $\mathcal{B}$.

We use the approximate joint training process [17] to train the whole network, ignoring the derivatives with respect to the proposed boxes' coordinates. Fortunately, this does not have a great influence on the performance [17]. We train the whole network using SGD. CNNs for extracting low-level features are initialized from the pre-trained VGG-16 model. We do not fine-tune the first four convolutional layers for efficiency. The rest of CNN layers are fine-tuned only in the first $50K$ iterations. The other weights are initialized according to Gaussian distribution. For optimization, we use ADAM [22], with an initial learning rate of 10^{-5} for parameters in the pre-trained VGG-16 model, and 10^{-4} for other parameters. The latter learning rate is halved every $10K$ iterations until 10^{-5}. The network is trained for $200K$ iterations. Each iteration uses a single image sampled randomly from training dataset. For each training image, we resize it to the shorter side of 700 pixels, while the longer side no more than 1500 pixels.

IV. Experiments

In this section, we conduct experiments to verify the effectiveness of the proposed methods. Our network is implemented using Torch 7. The experiments are performed on NVIDIA Titan X GPU with 12GB memory.

A. Datasets

Three datasets are used here to evaluate the effectiveness of our proposed method.

The first dataset is composed of car license plates from China, denoted as "CarFlag-Large". We collected $450K$ images for training, and 7378 images for test. The images are captured from frontal viewpoint by fixed surveillance cameras under different weather and illumination conditions, *e.g.*, in sunny days, in rainy days, or at night time, with a resolution of 1600×2048. The plates are nearly horizontal. Only the nearest license plate in the image is labeled in the ground truth file.

The second dataset is the Application-Oriented License Plate (AOLP) database [4]. It has 2049 images in total with Taiwan license plates. This database is categorized into three subsets with different level of difficulty and photographing condition, as refer to [4]: Access Control (AC), Traffic Law Enforcement (LE), and Road Patrol (RP). Since we do not have any other images with Taiwan license plates, to train the network, we use images from different sub-datasets for training and test separately. For example, we use images from LE and RP subsets to train the network, and evaluate the performance on AC subset. Considering the small number of training images, data augmentation is implemented by rotation and affine transformation.

The third dataset is issued by Yuan *et al.* [5], and denoted as "PKUData". It has 3977 images with Chinese license plates captured from various scenes. It is categorized into 5 groups (*i.e.*, G1-G5) corresponding to different configurations, as introduced in [5]. However, there are only the plate bounding boxes given in the ground truth file. Hence, we merely evaluate the detection performance on this dataset.

Fig. 3. Example results for open wide car license plate detection and recognition by our jointly trained model. Images in the first line are from CarFlag-Large, the second line are from AOLP and the third line are from PKUData. The results demonstrate that our model can detect and recognize car license plates under various photographing conditions, such as day and night, sunny and rainy days, *etc*.

B. Evaluation Criterion

To evaluate the "End-to-end" performance with both detection and recognition results considered, we follow the "End-to-end" evaluation protocol for general text spotting in natural scene [23] as they have similar application scenario. Define IoU as

$$\text{IoU} = \frac{\text{area}(R_{\text{det}} \cap R_{\text{gt}})}{\text{area}(R_{\text{det}} \cup R_{\text{gt}})} \tag{6}$$

where R_{det} and R_{gt} are regions of the detected bounding box and ground-truth respectively.

The bounding box is considered to be correct if its IoU with a ground truth bounding box is more than 50% (IoU > 0.5), and the plate labels match. It should be note that we denote all Chinese character in license plates as '*', since the training images in CarFlag-Large are all from one province and use the same Chinese character. The trained network can not be used to distinguish other Chinese characters.

As to the detection-only performance, we follow the criterion used in [5] for fair competition, *i.e.*, a detection is considered to be correct if the license plate is totally encompassed by the bounding box, and IoU > 0.5.

C. Performance Evaluation on CarFlag-Large

In this section, we would like to demonstrate the superiority of our end-to-end jointly trained framework compared with commonly used two-stage approaches. As illustrated in Figure 4, a commonly used two-stage approach implements plate detection and recognition by two separated models. Plate detection is carried out firstly. The detected objects are cropped out and then recognized by another different model. In contrast, our proposed network outputs both detection and recognition results at the same time, with a single forward pass and requiring no image cropping. The convolutional features are shared by both detection and recognition, which omits feature re-computation. For simplicity, we denote our jointly trained network as "Ours (Jointly-trained)", and the two stage approach as "Ours (Two-stage)". The model used only for plate detection is denoted as "Ours (Detection-only)".

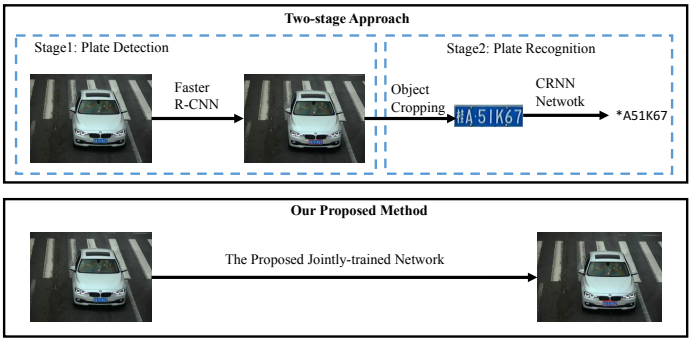


Fig. 4. Two-stage approach VS. our proposed method. In the two-stage approach, after license plate detection by Faster R-CNN, we crop the detected license plates from the image, and then recognize them by another separate model (CRNN in this paper). The features need to be re-computed during recognition phase. In contrast our proposed network takes an image as input, and produces license plate bounding boxes and plate labels in one-shot. It avoids some intermediate processes like image cropping, and share computation for convolutional feature extraction.

For fair competition, we train a Faster R-CNN [17] model using the $450K$ training images for plate detection only. We

modify the scales and shapes of anchors as the ones we used in this paper so that they fit the license plates. The network is also trained with $200K$ iterations, using the same initial parameters and learning rate. As to the plate recognition, we employ CRNN framework [20], which produces the state-of-the-art performance on general text recognition. It is an end-to-end framework for cropped word recognition, including CNN layers, RNN layers and CTC for transcription, from bottom to top. We crop the groud-truth license plates from the 450000 training images, and resize them to 160×32 pixels. Then we fine-tune the CRNN model with these training data.

In order to boost the performance, we rescale the input image into multiple sizes during test phase for both our proposed network and the detection-only Faster R-CNN network. The input images are resized to the shorter side of 600, 1200 pixels respectively, while the longer side less than 1500 pixels. With our framework, both detection and recognition results come out together, while with the two-stage approach, we crop the detected bounding boxes from input images, resize them to 160×32 pixels, and then feed into the trained CRNN model for recognition. Only bounding boxes with classification score larger than 0.95 are kept and merged via NMS. Considering that there is only one plate labeled as ground truth per image, we finally choose the one that has 7 characters recognized and/or with the highest detection score for evaluation. The test results are presented in Table I. Our jointly trained network gives the "End-to-end" performance with F-measure of 96.13% on 7378 test images. It is around 2% higher than the results by the two-stage approach, which demonstrates the advantage of end-to-end training for both detection and recognition in an unified network. The learned features are more informative, and the two subtasks can help with each other.

TABLE I
EXPERIMENTAL RESULTS ON CARFLAG-LARGE DATASET. WE COMPARE BOTH PERFORMANCE AND RUNNING SPEED OF OUR JOINTLY TRAINED NETWORK WITH A TWO-STAGE BASELINE METHOD. THE JOINTLY TRAINED NETWORK ACHIEVES NOT ONLY HIGHER ACCURACIES ON BOTH DETECTION AND "END-TO-END" PERFORMANCE, BUT ALSO IN A SHORTER TIME.

Method	End-to-end Performance (%)	Detection-only Performance (%)	Speed (per image single scale) (ms)
Ours(Jointly-trained)	**96.13**	**98.15**	**300**
Ours(Two-stage)	94.09	97.00	450

In terms of the computational speed, the unified framework takes about $300ms$ per image for a forward evaluation on the single small input scale, while the two-stage approach needs around $450ms$ to get both detection and recognition results, as it needs to implement image cropping and CNN feature re-calculation.

We also compare the detection-only performance. Our jointly trained network produces a detection accuracy of 98.15%, which is 1% higher than the result given by detection-only Faster R-CNN network. This result illustrates that car license plate detection can be improved with the multi-task loss used during training time.

Some experimental results using our jointly trained network are presented in the first row of Figure 3, which show that our model can deal with images under different illumination conditions.

D. Performance Evaluation on AOLP

In this section, we compare the "End-to-end" performance of our method with other state-of-the-art methods on the AOLP dataset. Note that the network is only trained with $15K$ iterations because of the small number of training images in this dataset. Moreover, since the sizes of license plates in AOLP are almost the same, and the ratios between license plates and images sizes are also similar. For this dataset, we only use a single image scale with shorter side as 700 pixels in test phase.

The detection and recognition results are presented on the second row in Figure 3. Comparison results with other methods in Table II show that our approach performs better on AC and LE subsets with "End-to-end" evaluation. It also gives the best performance for plate detection on all three subsets, with averagely 2% higher than the sliding window based method used in Li *et al.* [13], and 4% higher than the edge based method used in Hsu *et al.* [4]. As to the computational speed, out network takes about $400ms$ to get both detection and recognition results, while Li *et al.*'s method [13] costs $2-3s$, and Hsu *et al.*'s approach [4] needs averagely $260ms$.

It should be noted that in Table II, "End-to-end" performance on RP subset is worse than that in [13]. That may be because the license plates in RP have a large degree of rotation and projective orientation. In [13], the detected license plates are cropped out and Hough transform is employed to correct the orientation. In contrast, our method does not explicitly handle the rotated plates. Integrating spatial transform network into our end-to-end framework may be a solution, referring to [24], which is a future work.

E. Performance Evaluation on PKUData

Because the ground truth file in PKUData only provides the plate bounding boxes, we simply evaluate the detection performance on this dataset. Both the detection accuracy and computational efficiency are compared with other methods [10], [2], [5]. We use the same model trained by the CarFlag-Large dataset, as they are both datasets with Chinese license plates.

Images on the third line of Figure 3 show examples with both detection and recognition results. The detection-only results by our approach and other three methods are presented in Table III. Our jointly trained model demonstrates absolute advantage on all 5 subsets, especially on G4, where we achieve 100% detection rate. This result proves the robustness of our approach in face of various scenes and diverse conditions. Qualitatively, our jointly trained network achieves a average detection ratio of 99.80%, which is 2% higher than the previous best performance method.

In addition, the detection performance by our jointly trained network is slightly better than that by the detection-only

TABLE II

EXPERIMENTAL RESULTS ON AOLP DATASET. AC (ACCESS CONTROL) IS THE EASIEST DATASET WHERE IMAGES ARE CAPTURED WHEN VEHICLES PASS A FIXED PASSAGE WITH A LOWER SPEED OR FULL STOP. LE (LAW ENFORCEMENT) DATASET CONSISTS OF IMAGES CAPTURED BY ROADSIDE CAMERA WHEN A VEHICLE VIOLATES TRAFFIC LAWS. RP (ROAD PATROL) REFERS TO THE CASES THAT THE CAMERA IS HELD ON A PATROLLING VEHICLE, AND THE IMAGES ARE TAKEN WITH ARBITRARY VIEWPOINTS AND DISTANCES. WE COMPARE OUR PROPOSED METHOD WITH OTHER STATE-OF-THE-ART METHODS ON BOTH PERFORMANCE AND RUNNING SPEED. OUR JOINTLY-TRAINED NETWORK SHOWS IMPROVED PERFORMANCE FOR IMAGES WITH LICENSE PLATES IN NEARLY HORIZONTAL POSITION.

Method	End-to-end Performance (%)			Detection-only Performance (%)			Speed (per image single scale) (ms)
	AC	LE	RP	AC	LE	RP	
Hsu *et al.* [4]	–	–	–	96	95	94	**260**
Li *et al.* [13]	94.85	94.19	**88.38**	98.38	97.62	95.58	$1000-2000$
Ours(Jointly-trained)	**95.29**	**96.57**	83.63	**99.56**	**99.34**	**98.85**	400

TABLE III

EXPERIMENTAL RESULTS ON PKUDATA. DETECTION PERFORMANCE AND RUNNING SPEED ARE COMPARED BETWEEN OUR PROPOSED METHOD AND OTHER STATE-OF-THE-ART METHODS. G1 - G5 CORRESPOND TO DIFFERENT IMAGE CAPTURING CONDITIONS. OUR JOINTLY TRAINED NETWORK ACHIEVES A AVERAGE DETECTION RATIO OF 99.80%, WHICH IS 2% HIGHER THAN THE PREVIOUS BEST PERFORMANCE METHOD. IN ADDITION, THE JOINTLY TRAINED NETWORK, WHICH INTEGRATES BOTH DETECTION AND RECOGNITION LOSSES, PERFORMS BETTER THAN THAT TRAINED ONLY WITH THE DETECTION INFORMATION.

Method	Detection Performance (%)						Speed (per image single scale) (ms)
	G1	G2	G3	G4	G5	Average	
Zhou *et al.* [2]	95.43	97.85	94.21	81.23	82.37	90.22	475
Li *et al.* [10]	98.89	98.42	95.83	81.17	83.31	91.52	672
Yuan *et al.* [5]	98.76	98.42	97.72	96.23	97.32	97.69	42
Ours(Detection-only)	**99.88**	99.71	**99.87**	99.65	98.81	99.58	**300**
Ours(Jointly-trained)	**99.88**	**99.86**	**99.87**	**100**	**99.38**	**99.80**	**300**

network as seen from Table III. This is consistent with the outcome on CarFlag-Large dataset, and proves again that the detection performance can be boosted when training with the label information.

In terms of computational speed, Yuan *et al.*'s method [5] is relatively faster than ours', since they use simple linear SVMs, while we use deep CNNs and RNNs.

V. CONCLUSION

In this paper we have presented a jointly trained network for simultaneous car license plate detection and recognition. With this network, car license plates can be detected and recognized all at once in a single forward pass, with both high accuracy and efficiency. By sharing convolutional features with both detection and recognition network, the model size decreases largely. The whole network can be trained approximately end-to-end, without intermediate processing like image cropping or character separation. Comprehensive evaluation and comparison on three datasets with different approaches validate the advantage of our method. In the future, we will extend our network to multi-oriented car license plates. In addition, with the time analysis, it is found that NMS takes about half of the whole processing time. Hence, we will optimize NMS to accelerate the processing speed.

REFERENCES

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (alpr): A state-of-the-art review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, 2013.

[2] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, 2012.

[3] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377–392, 2006.

[4] G. Hsu, J. Chen, and Y. Chung, "Application-oriented license plate recognition," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 552–561, 2013.

[5] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A robust and efficient approach to license plate detection." *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1102–1114, 2017.

[6] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, pp. 1690–1705, 2014.

[7] S. Chang, L. Chen, Y. Chung, and S. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, p. 4253, 2004.

[8] S. Yu, B. Li, Q. Zhang, C. Liu, and M. Meng, "A novel license plate location method based on wavelet transform and emd analysis," *Pattern Recogn.*, vol. 48, no. 1, p. 114125, 2015.

[9] I. Giannoukos, C.-N. Anagnostopoulos, V. Loumos, and E. Kayafas, "Operator context scanning to support high segmentation rates for real time license plate recognition," *Pattern Recogn.*, vol. 43, no. 11, p. 38663878, 2010.

[10] B. Li, B. Tian, Y. Li, and D. Wen, "Component-based license plate detection using conditional random field model," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1690–1699, 2013.

[11] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, pp. 1096–1107, 2016.

[12] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 1 – 13, 2017.

[13] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and lstms," 2016. [Online]. Available: https://arxiv.org/abs/1601.05610

[14] A. Graves, M. Liwicki, and S. Fernandez, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, 2009.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[16] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. [Online]. Available: https://arxiv.org/abs/1612.08242

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.

[18] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "Deeptext:a unified framework for text proposal generation and text detection in natural images," *CoRR*, vol. abs/1605.07314, 2016.

[19] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.

[20] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *CoRR*, vol. abs/1507.05717, 2015.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[23] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 robust reading competition," in *Proc. Int. Conf. Doc. Anal. Recog.*, 2015, pp. 1156–1160.

[24] B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.