



闽江学院

本科毕业论文（设计）

题 目	车牌识别管理系统的设计与实现
学 生 姓 名	苏汉斌
学 号	3197103109
学 院	数学与数据科学学院（软件学院）
年 级	2019
专 业	软件工程（闽台合作）
指 导 教 师	曹永忠
职 称	教授
完 成 日 期	2023 年 4 月

闽江学院毕业论文（设计）诚信声明书

本人郑重声明：

兹提交的毕业论文（设计）《基于卷积神经网络的车牌识别系统》，是本人在指导老师 曹永忠 的指导下独立研究、撰写的成果；论文（设计）未剽窃、抄袭他人的学术观点、思想和成果，未篡改研究数据，论文（设计）中所引用的文字、研究成果均已在论文（设计）中以明确的方式标明；在毕业论文（设计）工作过程中，本人恪守学术规范，遵守学校有关规定，依法享有和承担由此论文（设计）产生的权利和责任。

声明人（签名）：

年 月 日

摘 要

本研究提出了一种以卷积神经网络为基础的车牌识别方法。系统接收车牌图像作为输入，通过 YOLOv5 进行车框的定位，随后利用多层卷积神经网络进行特征提取和识别，相比传统的模式识别提高了识别的准确性，并且传统方法对干扰因素的鲁棒性较差，可能导致识别性能下降。为了增强系统的鲁棒性和识别准确性，我们采用数据增广和随机抽样技术来扩展训练集，从而避免了过拟合问题，并为车牌识别系统的进一步智能化提供了有力支持。

关键词： 车牌检测；YOLOv5；深度学习

Abstract

In this study, a convolutional neural network-based license plate recognition method is proposed. The system receives the license plate image as input, locates the frame by YOLOv5, and subsequently uses a multilayer convolutional neural network for feature extraction and recognition, which improves the recognition accuracy compared with traditional pattern recognition, and the traditional method is less robust to interference factors, which may lead to degradation of recognition performance. In order to enhance the robustness and recognition accuracy of the system, we use data augmentation and random sampling techniques to extend the training set, thus avoiding the overfitting problem and providing strong support for further intelligence of the license plate recognition system.

Keywords: License Plate Detection; YOLOv5; Deep Learning

目 录

1 引言	- 2 -
1.1 研究背景	- 2 -
1.2 研究意义	- 2 -
2 系统模型梳理	- 2 -
2.1 Python 语言	- 3 -
2.2 MySQL 数据库	- 3 -
2.3 YOLOv5 的基本网络结构	- 4 -
2.4 LPRnet 的基本网络结构	- 6 -
2.5 Qt 框架	- 7 -
2.6 PyQt 框架	- 8 -
2.7 PyTorch 框架	- 8 -
3 系统分析与设计	- 10 -
3.1 数据库的概念结构设计	- 12 -
3.2 数据库的逻辑结构设计	- 13 -
4 系统的实现	- 14 -
4.1 登陆的实现	- 14 -
4.2 注册的实现	- 15 -
4.3 功能的实现	- 16 -
5 实验结果的讨论与分析	- 21 -
5.1 讨论	- 14 -
5.2 分析	- 15 -
参考文献	- 23 -
致 谢	- 25 -

1 引言

1.1 研究背景

随着城市化发展加速，交通运输需求急剧上升，导致交通拥堵问题愈发严重。与此同时，交通事故时有发生，安全隐患已成为社会关注的焦点。传统的交通监管方式已不足以应对不断增长的交通管理需求。作为新一代智能交通监测系统，车牌识别系统能够实现自动化、高效的车辆辨识与数据处理，从而提升交通管理水平，减少交通事故和拥堵现象。此外，智慧城市建设近年来已成为国家重点发展战略。车牌识别系统作为智慧城市建设的核心要素，通过智能车辆管理、停车场管理和交通监管等多方面功能，为城市智能化和精细化管理提供了坚实的支持。

1.2 研究目的

车牌识别系统是一种结合图像处理和模式识别技术的应用系统，在智能交通领域占据了举足轻重的地位。在万物互联的时代背景下，该系统在车辆信息管理和车辆位置规划方面发挥着至关重要的作用。车牌识别算法主要分为两个阶段，即车牌边框定位和车牌字符识别^[1]。

车牌定位算法的目标是利用图像^[2]处理技术和模式识别方法，在图像中自动识别车辆的车牌区域。作为车牌识别系统的第一步，车牌定位至关重要，因为其精度和速度直接决定了后续字符识别和信息提取任务的准确性和效率。车牌定位的常用方法包括基于颜色特征、形状特征和边缘检测的方法^[3]。

近年来，随着深度学习技术的飞速发展，基于深度学习的车牌定位算法已经成为广泛应用的主流技术。这些算法利用深度学习模型对车牌图像进行高效、准确的定位，取代了传统的手工特征提取和分类方法，大大提升了车牌识别的精度和效率，通过神经网络模型自动学习车牌的特征，从而实现高速度的车牌定位。深度学习算法具有较强的泛化能力，可以在各种光照、天气条件和车牌形态下保持较高的定位准确性。因此，深度学习方法在车牌识别系统中得到了广泛应用，为智能交通和智慧城市建设提供了有力支持。

2 系统模型梳理

2.1 Python 语言

Python 是一种高级面向对象编程语言，它有着极高的效率和易用性著称。它可以很好地作为 C++ 和 Java 等语言的扩展语言，实现简化的接口使用。Python 作为一种解释型的语言，它将代码在计算机内由解释器翻译成字节码的形式，然后再由字节码的形式转换为机器语言，从而运行程序，这一点有点类似 Java 的执行过程。Python 面向对象的思想也使其能够编写出高复用性的代码，加快了项目开发进度。作为一种脚本语言，Python 具有胶水语言的外号，能够与 C++ 和 Java 等其他语言实现十分容易的交互，互相编写新的模块。Python 能够流行起来的另一大因素就是它具有十分庞大的标准库和第三方库，能够以十分简短的代码实现一个较为复杂的功能，极大提升开发效率和工程验证，这也是本文使用 Python 作为开发语言的原因。

2.2 MySQL 数据库

"数据库"这一概念最早出现在 20 世纪 60 年代，至今已经过了半个多世纪的演变。从最初的简单数据库发展到现在广泛使用的关系型数据库，其发展速度之快令人瞩目。在信息技术不断完善的当今时代，程序员们创造了各种各样的系统，而这些系统基本上都离不开数据库的支持。

随着大数据时代的到来，人们接触到的信息量不断增加。为了满足个性化需求，推荐系统应运而生。在这些推荐系统中，数据库扮演着举足轻重的角色。无论是用户的基本信息、兴趣爱好，还是系统需要分析和处理的数据，都存储在数据库之中。MySQL 数据库是其中的佼佼者，这是一款具有高安全性、高效率且可跨平台的数据库系统。它能够与 PHP、Java 等主流编程语言无缝对接，具有极强的兼容性。这使得它在众多数据库系统中脱颖而出，成为了许多程序员和企业的首选。

在过去的半个多世纪中，数据库技术不断发展，涌现出许多种类的数据库，如：分布式数据库、面向对象数据库、NoSQL 数据库等。这些数据库各有特点，适用于不同的应用场景。如今，数据库已经成为现代计算机系统中不可或缺的组成部分，它们在各行各业的信息处理和存储中发挥着重要作用。

随着云计算、物联网等新兴技术的发展，数据库技术仍在不断创新和进步。未来，我们有理由相信，数据库将会在更加广泛的领域、更加复杂的场景中发挥出更大的价值。而作为程序员和企业，紧跟数据库技术的发展趋势，掌握先进的数据库技能，将有助于我们在信息时代中立于不败之地。本文推荐系统的数据存储在 MySQL 数据库中，由于其具有安全性高及存储容量大的特点，因此成为支撑本系统数据的不二选择[4]。

2.3 YOLOv5 的基本网络结构

YOLO 系列是一种非常流行的目标检测算法，其中包括了 YOLOv1、YOLOv2、YOLOv3、YOLOv4、YOLOv5 等版本。这类算法的目的是检测图像或者视频当中的物体，并输出它们的类别、位置或者是大小等信息。而对比其它检测算法，YOLO 有以下特点：1. YOLO 采用的是单次前向计算，因此它只需要遍历图像一次，所以导致速度非常快。2. 全卷积网络结构的它可以对整个图像进行密集检测，通过多层特征融合的处理方法，可以提高检测的精度。3. 对于一些只能检测单个类别的算法，它的应用场景显得更加的广泛。而 YOLOv5 相较于早期版本，它的检测速度更快，并且由于支持了 CPU、TensorRT 和 ONNX，使其更易于训练和部署。YOLOv5 按照网络结构的复杂度可分为 YOLOv5s、YOLOv5m、YOLOv5x 等版本，大小也是依次增加，本篇文章考虑到了模型的运行速度以及对设备的负载压力，以其中复杂度最小的 YOLOv5s 作为车牌定位的模型来进行训练。

YOLOv5s 网络结构如图 2-1 所示，主要由 Backbone、Neck、Head 等部分组成^[5]。Backbone 是特征提取网络，由 Focus、csp 等结构组成。Focus 即为切片操作，通过扩大通道数量，再进行卷积实现下采样，在不提高模型计算量的前提下，保留更多图像信息。YOLOv5s 在骨干网络中采用了 CSP 的结构，在这当中 CSP1 负责提取信息，而 CSP2 则是负责特征的融合。Neck 结构由 FPN 特征金字塔和 PAN 路径聚合结构组成^[6]。FPN 自上而下传播语义信息，而 PAN 从下至上传递图像的位置信息。Neck 结构把这两种结构进行了整合，实现了主干网络中不同尺度信息的融合。Head 即目标检测部分，主要用来检测不同尺寸图像，不同大小的 Head 头部负责检测大、中、小不同尺寸的物体。可由于 YOLOv5s 的设计主要是在中、大尺寸物体的检测，小尺寸物体由于分辨率低、图像信息不丰

富，因此识别率较低，YOLOv5 在这方面有着一定的改进空间^[7]。

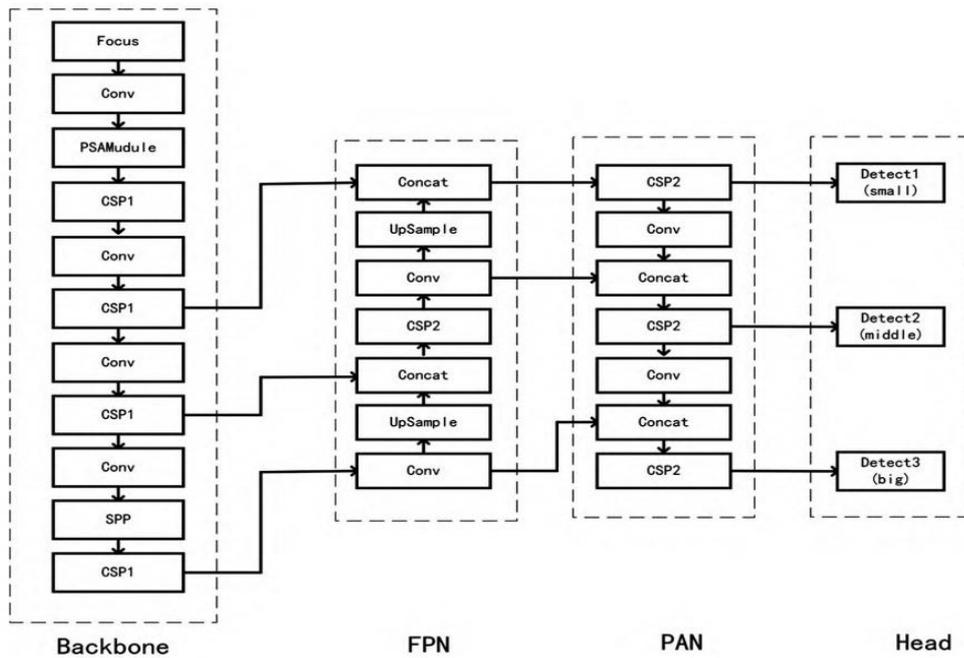


图 2-1 YOLOv5 结构图

如图 2-2 所示，Focus 模块在 v5 中的主要作用是在图像进入 backbone 之前，对图像执行切片操作。它从图像当中每间隔一个像素就会取一个值，这样的话我们就可以得到四张互补的图像。并且它们不会有信息的丢失^[8]。

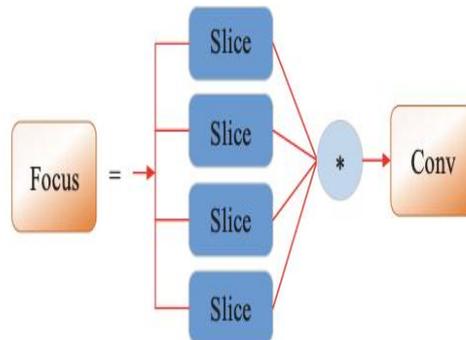


图 2-2 Focus 结构图

Conv 模块是一个综合卷积模块，它在许多关键模块中作为基本组成部分出现。如图 2-3 所示，这个模块包含了卷积层、批量归一化 (BN) 层和激活函数(SiLU)。



图 2-3 Conv 结构图

Bottleneck 利用了两个 Conv 模块将通道数进行了先减小再扩大对齐，以此进行特征值的信息提取，并使用 ShortCut 控制是否进行残差连接^[9]，如图 2-4。

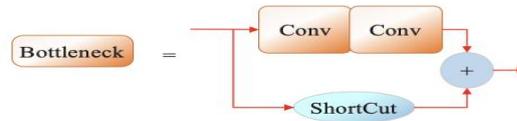


图 2-4 Bottleneck 结构图

SPP 模块是空间金字塔池化模块，其结构图如图 2-5。先把输入的特征图经过一个卷积操作来进行通道数的减半，然后分别做三种不同卷积核的最大池化下采样，再将三种池化结果与输入特征图按照通道来进行拼接，合并后的通道数变为原来的两倍，这样做便能实现以较小的代价提高感受野^[9]，从而有效提升模块的性能。

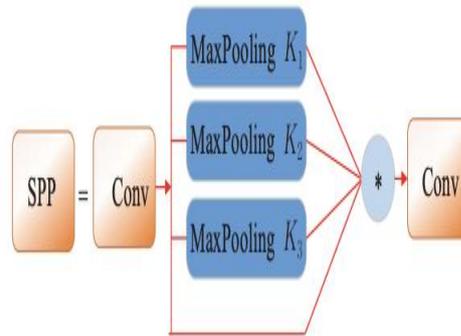


图 2-5 SPP 模块

2.4 LPRnet 的基本网络结构

车牌识别是城市管理、监控、车牌识别和停车场当中一项非常繁琐并且重要的任务。在车牌识别过程中遇到许多问题，因为图片、照明条件、天气条件、图像变形和车牌倾斜角度大等等。一个强大的字符识别系统应该能够在不损失精度的情况下处理不同环境的转换。LPRnet 没有对字符进行预分割，可以说是一个端到端的车牌识别算法^[10]。

卷积神经网络在图像分类、目标检测等计算机任务中有着显著的有效性和优势。可以使用卷积神经网络所组成的 LPRnet 从端到端对其进行训练。LPRnet 的结构中并没有用 RNN 的实时识别系统，所以可以创建 LPRnet 算法并将其用于嵌入式系统，在检测相对复杂的中文车牌时轻量级别的 LPRnet 仍然有不错的性能。LPRnet 主干网络如图 2-7 所示，其接收最原始的 RGB 图像作为输入，并计算大

量函数的空间分布。宽卷积（1*13 卷积核心）使用本地字符的上下文结构替换了基于 LSTM 的 RNN 神经网络^[11]，从而摆脱了对 RNN 的依赖。子网络的输出可以被视为一个带有概率的序列，代表对应字符的可能性，其长度仅等于输入图像的宽度。

Layer Type	Parameters
Input	94x24 pixels RGB image
Convolution	#64 3x3 stride 1
MaxPooling	#64 3x3 stride 1
Small basic block	#128 3x3 stride 1
MaxPooling	#64 3x3 stride (2, 1)
Small basic block	#256 3x3 stride 1
Small basic block	#256 3x3 stride 1
MaxPooling	#64 3x3 stride (2, 1)
Dropout	0.5 ratio
Convolution	#256 4x1 stride 1
Dropout	0.5 ratio
Convolution	# class_number 1x13 stride 1

图 2-7 Lprnet 网络结构

由于解码器输出与目标序列的长度不对应，因此引入了 CTC 丢失函数，而无需分段的端到端训练。CTC 损失函数是一种广泛用于解决输入和输出序列不一致问题的方法。原始 RGB 图像是带有源网络的 RGB 图像，用作 CNN 的输入并提取图像特征。LPRNet 在输入端采用 STN(Spatial Transformer Network)网络对输入图像进行处理，STN 处理是可选的过程，STN 的出现使我们完成如何变换图像来获得优秀的车牌识别输入图像^[12]。

2.6 Qt 框架

Qt 是一款由 Haavard Nord 和 Eirik Chambe-Eng 于 1991 年共同开发的高效且通用的 C++ 应用程序开发框架，广泛应用于图形用户界面 (GUI) 程序开发^[14]。得益于跨平台兼容性、面向对象设计、丰富的 API 和详尽的开发文档等优点，Qt 成为了当今最强大的 GUI 开发库之一。经过 30 多年的发展，Qt 建立了庞大的生态系统，

类似于 MFC，内部整合了丰富的 API，为 GUI 开发提供便利。虽然 Qt 是纯 C++ 开发，要求掌握 C++ 语言（相对于 Python 学习成本较高），但得益于 Python 语言的灵活性，Qt 在 Python 中也提供了相应的接口库，即本文所使用的 PyQt。

2.7 PyQt 框架

PyQt^[15] 是 Digia 公司开发的一套将 Qt 接口应用到 Python 上的模块，以库的形式在 Python 中被调用，包含了 Qt 常见的类、函数和接口等，与 Qt 均能在各种主流操作系统上运行，具有跨平台和开发简易的特点。

Qt^[16] 集成了可视化界面设计工具 Qt Designer，其遵循 MVC 架构，将视图与逻辑分离，提高了开发效率。Qt Designer 的使用高度便捷，支持直接拖拽控件到预定位置，并能实时查看布局效果。其生成的 .ui 文件最后可以通过 PyUIC 工具转换成 Python 类，从而在 Python 的 GUI 编程中灵活调用，Qt 基础模块构成如图 2-10^[17]。

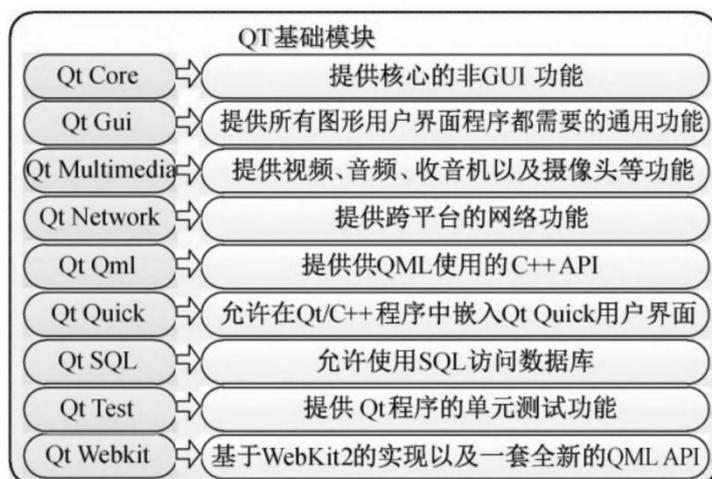


图 2-10 Qt 基础模块构成

2.8 PyTorch 框架

技术应用的普及与技术的高效实现密不可分，深度学习框架通过将深度学习算法模块化封装，能够实现训练、测试、调优模型的快速搭建，为技术应用的预测与落地的决策提供有力支持^[18]。现如今，伴随着人工智能的全面发展与社会上众多学习框架的诞生，使得该生态愈发完整。主流深度学习框架的概述如表 2-1 所示。PyTorch 是 Facebook 在 2017 年发布的一个深度学习框架，它源于 Torch

的升级并成为了一款全新的产品。作为一个与 PyTorch 无缝集成的原生包，PyTorch 采用了命令式编码风格。得益于动态计算图、简洁的后端和高度可扩展性等优点，PyTorch 已广泛应用于众多领域。

表 2-1 部分主流深度学习框架概况

序号	框架名称	开发/支持	支持语言	开源/发布时间
1	TensorFlow(Keras)	Google	Python/C++/Java/Go/R/Swift/C#/JavaScript	2015-11(2018-09)
2	PyTorch(Caffe2)	Facebook(贾扬清)	Python/C++	2017-01(2018-04)
3	飞桨 PaddlePaddle	百度	Python/C++	2016-09
4	CNTK	Microsoft	C++/C#/Python	2016-01(2019-08)
5	MXNet	Apache/Amazon	Python/R/Julia/C++/Scala/MATLAB/JavaScript	2016-11
6	DeepLearning4J	SkyMind	Java/Scala	2014-06
7	Chainer	Preferred Networks	Python	2015-06(2019-12)

PyTorch 在已经在聊天机器人、机器翻译、文本搜索、文本到语音转换和图像与视频分类等项目中取得了巨大成功^[19]。PyTorch 和 TensorFlow 是目前大众最常使用的深度学习框架之一。虽然它们的目标都是提供深度学习算法的实现和加速，但两者之间存在很多差异。PyTorch 集成了一些非常有用的工具，如 TensorBoard，可以轻松地对模型进行可视化和调试。与此相反，TensorFlow 的可视化和调试工具可能需要更多的配置和学习。在一些情况下，PyTorch 的训练速度可能比 TensorFlow 更快。这得益于 PyTorch 的动态前向和反向传播，它们可以在运行时进行优化。此外，与 TensorFlow 相比，PyTorch 的 GPU 内存管理更加灵活，能够更好地防止内存泄漏和减少 GPU 内存浪费。

为了深入了解 TensorFlow 和 PyTorch 两个框架的异同，我们通过整理官方网站信息、论坛反馈和实际操作等多种资源，得出了表 2-2 中的对比结果。

表 2-2 TensorFlow 与 PyTorch 对比情况

序号	参量	TensorFlow	PyTorch
1	安装难易程度	良好	优秀
2	上手难易程度	良好	优秀
3	代码理解程度	良好	优秀
4	API 丰富程度	优秀	良好

5	模型丰富程度	优秀	良好
6	社区支持程度	优秀	良好
7	语言支持程度	优秀	良好
8	可视化程度	优秀	良好

3 系统分析与设计

3.1 YOLOv5 和 LPRnet 的融合

YOLOv5 与 LPRnet 的结合实现了高效的车牌识别。在训练阶段，首先整合数据，使得 LPRnet 无需过多的训练模型，对于单一车型仅需进行一次训练。假如训练模型质量高，后续将无需花费额外时间训练。接下来，将训练后的车牌检测网络 YOLOv5 与字符识别网络 LPRnet 相连接，将输入图像送入检测网络以提取检测框，然后将输出的候选框作为 LPRnet 网络的输入。

为了较容易确定车牌四个角点的位置，可以通过简单的车牌二值化图像进行处理。接着，利用透视变换矫正车牌，再将矫正后的车牌输入 LPRnet 中。字符识别网络 LPRnet 将首先对检测到的候选框利用主干网络提取特征信息，接着通过 CTC 和集束搜索输出最终的车牌号。

在实际应用场景中，车牌的倾斜角度会不断变化，且摄像头捕捉到的绝大部分车牌图像都存在畸变。本文所提出的车牌识别系统能够有效地缩短检测时间，从而实现更优的识别效果。车牌识别的整体流程如图 2-8 所示。

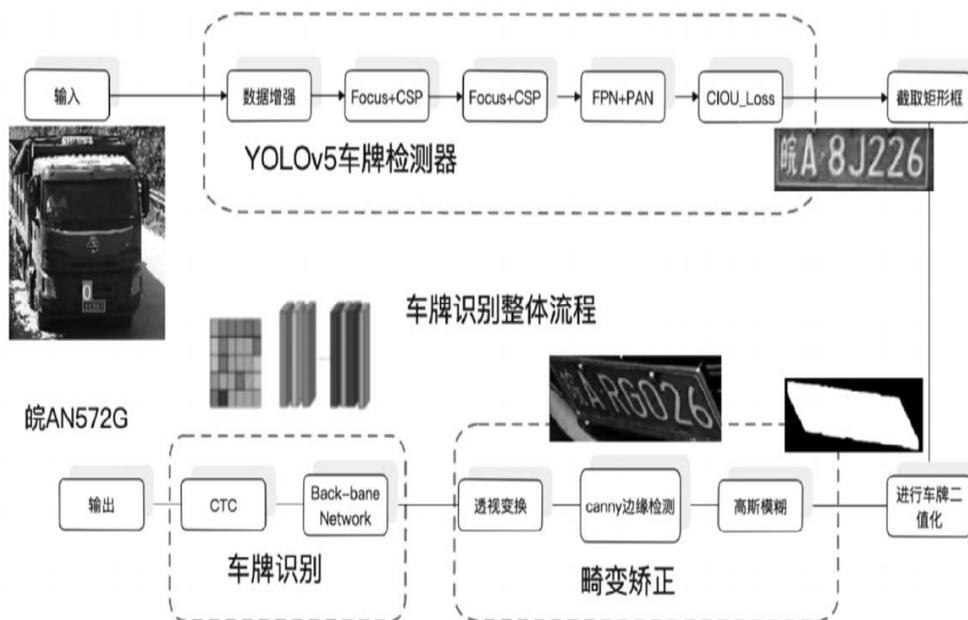


图 2-8 车牌识别整体流程

通常把 LSTM 与 CTC (Connectionist Temporal Classification) 相结合、Seq2Seq 与 CTC 相结合, 或使用 CNN 来实现字符识别而无需分割字符。通过 LPRnet 主干网络进行特征提取, 再利用卷积核获得车牌序列, 由于 LPRnet 输出和目标序列长度有所差异, 因此使用 CTC 损失。通过 Softmax-Loss 让每一个字符元素与每一列输出相对应, 也就是要在训练集标出图片中每个字符的位置, 这一过程会浪费大量时间。而 CTC 恰恰能够处理网络标签与输出不对齐的情况, 通过使用如下公式来计算损失情况:

$$\text{权重指数: } G = e^{XW^T} \quad [13] \quad (2-1)$$

$$\text{损失函数: } Loss = \frac{1}{o} [P \log(Q^T)]^D \times E \quad [13]$$

另外通过启发式函数让每个节点的检测能力进行各自评估。当图的解空间非常大的时候经常使用该算法, 同时为降低占用存储空间与时间资源, 当深度扩展时, 将质量差的结点去除, 在降低空间消耗的同时也能够增强时间效率在这篇文章中, 我们可以看到车牌识别使用相当小的卷积神经网络来实现。

为了应对具有挑战性的车牌畸变数据, 我们引入了 LPRnet 模型。实验结果显示, LPRnet 系统能够在多种硬件架构上, 如 CPU 和 GPU, 实现实时推理。此外, 本文提出的车牌字符识别系统可以在专业的嵌入式低功耗设备上获得实时性能。

为了进一步降低计算复杂度, 我们使用了现代量化技术对网络模型进行压缩。这种方法有助于使车牌识别系统在更多类型的设备上实现实时性能, 同时保持准确率和识别速度。总之, LPRnet 模型在车牌识别领域具有很大的潜力, 未来可能会在更多应用场景中发挥关键作用。作为未来的研究方向可以将基于 cnn 的检测部分合并到算法中, 扩展 LPRnet 工作, 将检测和识别任务作为一个单一的网络进行评估, 以提高检测的质量^[13], 车牌识别最终效果如图 3-1 所示。



图 3-1 车牌识别效果图

3.2 数据库的概念结构设计

数据库的概念结构数据是指数据模型的抽象表示，它独立于特定数据库管理系统和存储结构。该结构体现了实体及其之间关系的逻辑组织，为数据的高级视图提供了基础。概念结构数据使用一致的语义和数据约束来描述现实世界中的信息，有助于确保数据完整性和一致性。通过这种方式，它弥补了现实世界与数据库之间的鸿沟，使得数据更容易被理解和使用。

User		
PK	account_id	VARCHAR(255)
	account_password	VARCHAR(255)
FK	user_position	VARCHAR(255)
	Tel_number	VARCHAR(255)

图 3-2 用户信息 E-R 图

Power_Position		
PK	id	VARCHAR(255)
	position	VARCHAR(255)
	power	VARCHAR(255)

图 3-3 用户权限 E-R 图

license_plate		
PK	car_number	VARCHAR(255)
	account_id	VARCHAR(255)
	province	VARCHAR(255)
FK	option	VARCHAR(255)

图 3-4 车牌信息 E-R 图

Function_choose		
PK	id	VARCHAR(255)
	function	VARCHAR(255)

图 3-5 功能选择 E-R 图

Park		
PK	id	VARCHAR(255)
	province	VARCHAR(255)
	pay	VARCHAR(255)
	Time_in	TIME
	Time_out	TIME

图 3-6 停车场 E-R 图

3.3 数据库的表结构设计

表 3-1 User (用户表)

序号	字段	说明	数据类型	长度	自增	主键	副键	允许空	默认值
1	Account_id	用户账号	Varchar	30	×	√	×	×	NULL
2	password	用户密码	Varchar	30	×	×	×	×	NULL
3	User_position	用户等级	Varchar	30	×	×	√	×	NULL
4	Tel_number	电话号码	Varchar	50	×	×	×	×	NULL

表 3-2 Power_Position (用户权限表)

序号	字段	说明	数据类型	长度	自增	主键	副键	允许空	默认值
----	----	----	------	----	----	----	----	-----	-----

1	Id	权限编号	Varchar	30	×	√	×	×	NULL
2	Position	等级名称	Varchar	30	×	×	×	×	NULL
3	Power	权限范围	Varchar	30	×	×	×	×	NULL

表 3-3 Function_choose (功能选择表)

序号	字段	说明	数据类型	长度	自增	主键	副键	允许空	默认值
1	Id	权限编号	Varchar	30	×	√	×	×	NULL
2	Function	功能选项	Varchar	20	×	×	×	×	NULL

表 3-4 Lincense_Plate (车牌信息表)

序号	字段	说明	数据类型	长度	自增	主键	副键	允许空	默认值
1	Account_id	用户账号	Varchar	30	×	√	×	×	NULL
2	Car_number	车牌号码	Varchar	30	×	×	×	×	NULL
3	Province	车牌省份	Varchar	20	×	×	×	×	NULL
4	Option	操作编号	Varchar	30	×	×	√	×	NULL

表 3-5 Park (停车场信息表)

序号	字段	说明	数据类型	长度	自增	主键	副键	允许空	默认值
1	Id	车牌号码	Varchar	30	×	√	×	×	NULL
2	Province	车牌省份	Varchar	20	×	×	×	×	NULL
3	Time_in	进入停车场时间	Time		×	×	×	×	NULL
4	Time_out	离开停车场时间	Time		×	×	×	×	NULL
5	Pay	支付费用	Int	30	×	×	×	×	NULL

4 系统的实现

4.1 登陆的实现

本系统的界面由 PyQt5 设计制作，登录页面展示效果请参见图 4-1。

登录流程图请参见图 4-2。

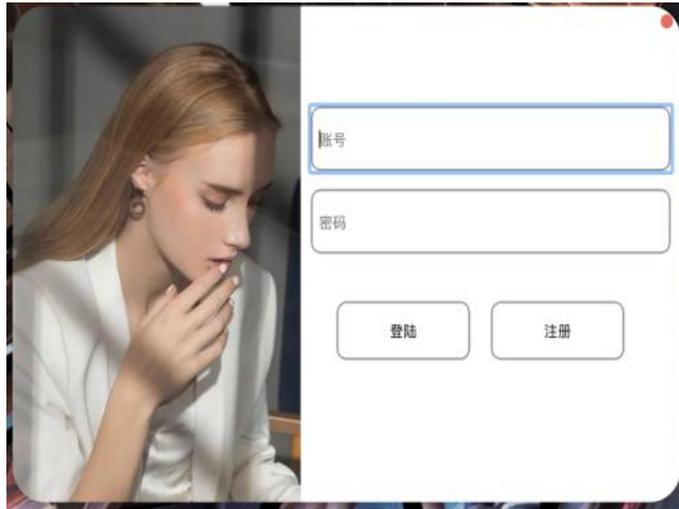


图 4-1 登陆页面效果图

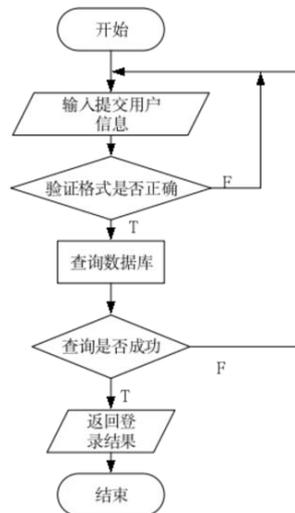


图 4-2 登陆流程图

当我们输入相应的 account 和 password 以后，单击登录的按键就会触发账号密码验证功能。这部分运用 MySQL 来对比我们输入的账号与密码与数据库当中所存储的信息是否一致。如果查询结果为空，则会显示“账号或密码不存在”的提示；若查询成功，则会自动跳转至功能选择页面。

4.2 注册的实现

当我们使用注册页面的时候，需要输入 User_id、Password 以及 Password Again。如果 Password 与 Password Again 不匹配，那么这个时候系统就会提示“输入的两次密码不相同”。若所有的信息都没有出现错误时，用户只需点击注册按钮便可完

成注册过程。注册完成后，用户可以点击“Login In”按钮页面将自动跳转回初始的登录界面。

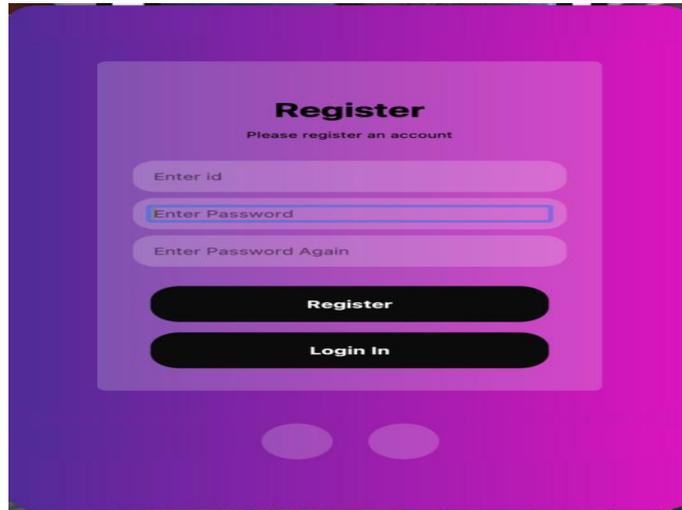


图 4-3 账号注册页面

4.3 功能的实现

在功能页面中，我们采用了堆栈布局，这样用户可以通过点击左侧的功能选项来切换不同的功能模块。相应的功能页面将显示在右侧的 **stack** 区域中。这种设计方式使得各个功能模块之间的切换变得更加方便快捷，用户可以轻松找到并使用所需的功能。如图 4-4 所示，功能页面的初始效果展示了各个功能模块的概览。通过使用堆栈布局，我们实现了一个整洁、易用的用户界面，有助于提高用户体验和操作效率。



图 4-4 功能页面初始化

系统首先展示的页面是车牌搜索页面，其中包含一个搜索按钮。为了提高界面美观，我们对搜索按钮的图标进行了定制。当用户点击搜索按钮时，系统将触发车牌搜索功能。该功能通过在 MySQL 数据库中的 license_plate 表进行模糊搜索，以找到用户输入的车牌信息。搜索到的车牌图片将通过 QScrollArea 控件展示，以方便用户查看。由于程序限制，搜索结果仅显示前 100 张图片。如需更精确地搜索，用户需要输入完整的车牌号。最终效果图可参见图 4-5。这样的设计使得车牌搜索功能既直观又易用，用户可以快速找到相关车辆信息，提高效率。



图 4-5 车牌搜索效果图

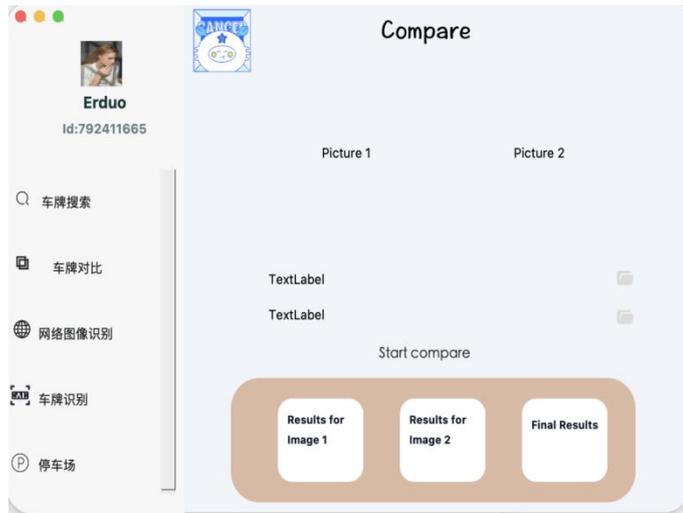


图 4-6 车牌对比界面

Picture1 和 Picture2 的控件类型设置为 label，用于展示待比较的两张图片。点击文件夹按钮可以访问本地文件，选定文件的路径将显示在 TextLabel 中。打开的第一张图片将显示在 Picture1 中，而第二张图片将显示在 Picture2 中。

当用户点击"Start Compare"按钮时，系统将开始对两张图片进行比较。第一张车牌图片的识别结果将显示在 result 框中，第二张图片的识别结果将显示在 result2 框中。如果两张图片的识别结果相同，"Final Result"将显示"相同"；如果识别结果不一致，"Final Result"将显示"不一致"。这样的设计使得用户可以方便地进行车牌图片比较，快速确认车牌信息是否相同。整个过程简单明了，有助于提高工作效率和用户体验。



图 4-7 车牌对比效果图

在网络图像识别功能模块中，实现方法是通过获取图片的 URL 地址，下载网络图片并将其保存为本地文件“./tmp.jpg”。接着，通过调用 YOLO 框架来实现图像识别。如图 4-8 所示，当我们将图片 URL 输入到预设的 lineEdit 区域中并点击“Load”按钮后，网络图像将在指定的 label 框中显示。识别完成后，最终结果也会呈现在“Final Result”部分。

这种设计允许用户直接使用网络图片进行识别，无需先下载到本地，提高了操作效率。借助 YOLO 框架的强大识别能力，网络图像识别功能可以准确地分析和提取图像中的信息。

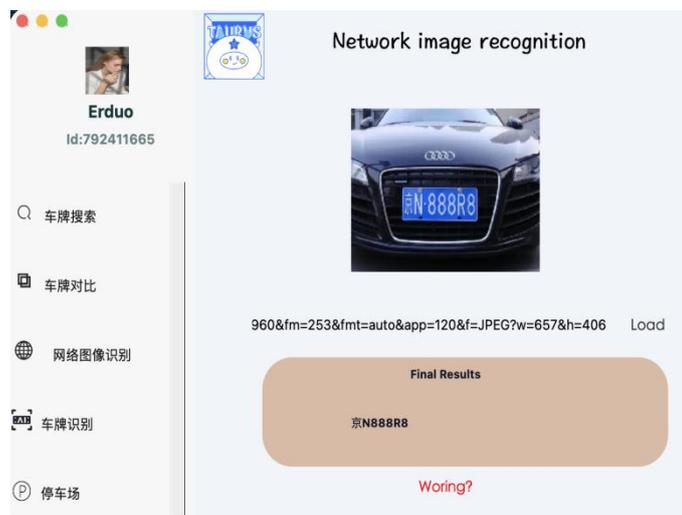


图 4-8 网络图像识别效果图

紧接下来的功能是车牌识别模块。通过点击文件夹按钮，用户可以轻松地打开本地文件，选择需要识别的车辆照片。选定的车辆图片将会展示在界面的 label 区域中，方便用户进行预览。在识别过程完成后，识别结果将在“Final Result”部分显示，以使用户快速查看识别出的车牌号。如图 4-9 所示，车牌识别功能的界面设计简洁明了，使用户能够轻松地完成车牌识别任务，提高了工作效率。同时，该功能可以与其他模块相互配合，如停车场管理，进一步提升整体系统的实用性和便捷性。

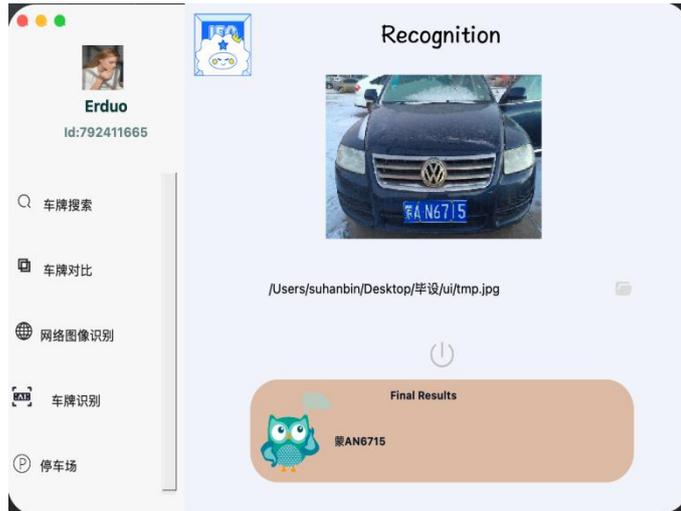


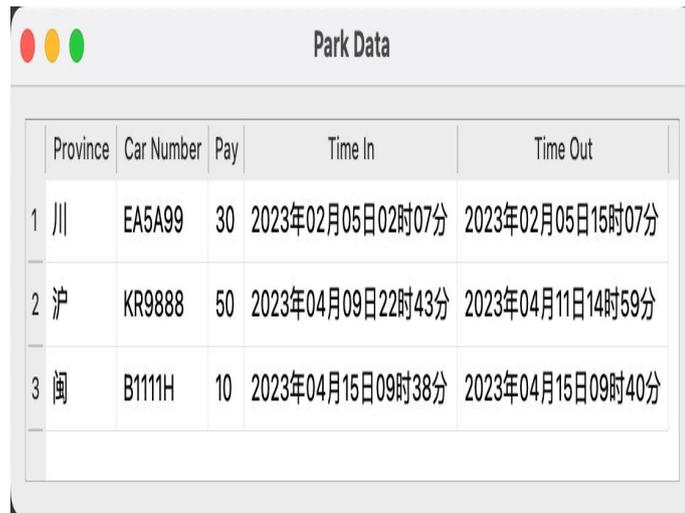
图 4-9 车牌识别效果图

最终功能是停车场管理，在右上角展示了停车场剩余车位数量。实现这一功能的方法是通过查询 MySQL 数据库中 Park 表的数据条数并进行计算。本文设定的初始车位数为 70，适合小型停车场的需求。界面中间部分显示车辆图片以及车辆成功进入停车场的相关信息。用户可以通过点击"Entering the parking lot"旁的"Join"按钮实现车辆进入停车场的操作，其他功能按钮操作类似。最终效果图见图 4-10。同时，当车辆进入停车场时，系统会以车辆入场的当前时间为准，将车辆图片保存至指定位置。通过这样的设计，停车场管理功能实现了方便快捷的车位监控与车辆进出管理。



图 4-10 停车场界面

点击"All vehicle information"按钮后，系统会连接到数据库中的 Park 表，以获取停车场内所有车辆的详细信息。通过使用 QTableWidgetItem 控件，信息将以表格形式展示在界面上，使用户能够清晰地查看车辆数据。如图 4-11 所展示的效果图，表格中包含了各项信息，如车牌号、入场时间等，方便管理员对停车场内车辆情况进行实时监控。用户可以更便捷地找到特定车辆的信息，从而提高停车场管理效率。



	Province	Car Number	Pay	Time In	Time Out
1	川	EA5A99	30	2023年02月05日02时07分	2023年02月05日15时07分
2	沪	KR9888	50	2023年04月09日22时43分	2023年04月11日14时59分
3	闽	B1111H	10	2023年04月15日09时38分	2023年04月15日09时40分

图 4-11 停车场信息效果图

5 实验结果的讨论与分析

5.1 讨论

本文的数据集采用了网络上公开的 CCPD 数据集，部分停车场数据集和自采车牌图片，共收集到 10628 张车牌图像。在实际的识别过程中，车牌边框在图像中分布的位置、所占的整个图像的比例，影响着最后的识别结果。如图 5-1(a)为目标框的中心点在图中的分布概率，图 5-1(b)为样本边框与图片的长、宽比。由目标的空间、大小分布看，目标框由顶部至中间部分，依次增大，底部数据量较少，且小目标车牌的数据量较大^[7]

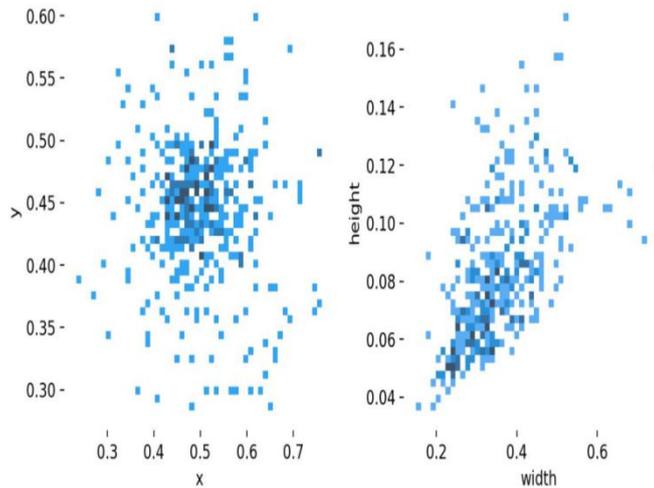


图 5-1 (左) 目标框分布概率 (右) 目标框与图片的长度比

由于 CCPD 数据集已经打好了标签，并将位置等信息都存放在命名当中。例如以下命名格式：

0299826388889-91_268-189&438_503&535-499&535_189&531_192&438_503&447-0_0_3_24_30_25_29_25-117-48.jpg

1. 0299826388889 表示的是区域
2. 91_268 对应的是车牌的两个倾斜角的度数，分别就是 91 度，竖直倾斜角度是 268 度
3. 189&438_503&535 所对应的就是边界框左上角和右下角的坐标：左上(189,438),右下角(503,535)
4. 499&535_189&531_192&438_503&447 对应的是车牌的四个端点的坐标位置右下角(499,535),左下角(189,531),左上角(192,438),右上角(503,447)
5. 0_0_3_24_30_25_29_25 所表示的就是车牌号码，第一个 0 表示的就是省份的信息，对应省份字典当中的“皖”，第二个 0 表示的就是该车牌所在地区的市级代码，所对应的就是 alphabets 的“A” 后 5 位为字母和文字，查看车牌号 ads 字典，如 3 为 D，24 为 0，30 为 6，25 为 1，最终车牌号码为皖 AD06151。

因此我们还需要对数据集进行一些预处理操作，将其转换为可以进行 YOLO 训练的格式，数据格式转换完毕以后可以运用 `check_label` 对结果进行查看。结果图如图 5-2



图 5-2 标签转换结果图

5.2 分析

本文章从当前实际需求着手，分析了传统识别车牌内容的缺陷，并在 PyTorch 框架的基础上设计构建了一个简单的车牌识别系统。对于该系统，先阐述了系统整体的运行流程和整体设计模块，较为仔细地分析了系统的设计思路；再通过输入包含目标车牌的待检测图像，测试系统能否正确识别，并将模型的识别效果指标进行量化输出。实验结果表明，设计的车牌识别系统可达到较高的车牌识别率，各模块功能完善，运行良好，可满足实际应用场景的需求^[20]。

参考文献

- [1]李春娟, 赵艳花. 一种基于形态学和几何特征的车牌定位方法 [J]. 微计算机信息, 2009, 25 (31): 175-7.
- [2]唐愉顺, 张生果, 牛潞, et al. 基于色彩的车牌识别研究 [J]. 现代计算机, 2020, (32): 63-6+71.
- [3]朱威威. 恶劣天气下基于集合划分的车牌识别方法 [D]; 信阳师范学院, 2019.
- [4]焦宇, 李民, 王欢, et al. 基于MySQL性能调优的推荐系统优化设计 [J]. 软件导刊, 2022, 21(09): 108-12.
- [5]李煜. 基于深度学习的三通管表面缺陷检测算法研究 [D]; 浙江大学, 2022.
- [6]王淑青, 张鹏飞, 要若天, et al. 基于改进YOLOv5的太阳能电池片表面缺陷检测 [J]. 仪表技术与传感器, 2022, (05): 111-6.
- [7]蔡先治, 王栋, 鲁旭葆, et al. 基于改进的YOLOv5的端到端车牌识别算法 [J]. 计算机时代,

2022, (12): 28-33.

- [8]刘钰发. 基于深度学习的驾驶员疲劳监测系统设计 [D]; 大连海洋大学, 2022.
- [9]邱天衡, 王玲, 王鹏, et al. 基于改进YOLOv5的目标检测算法研究 [J]. 计算机工程与应用, 2022, 58(13): 63-73.
- [10]周乐. 基于YOLO v4 以及CRNN+CTC算法的中文车牌识别系统 [D]; 南京邮电大学, 2022.
- [11]ZHERZDEV S, GRUZDEV A. Lprnet: License plate recognition via deep neural networks [J]. arXiv preprint arXiv:180610447, 2018.
- [12]李世伟. 基于YOLOv5算法的目标检测与车牌识别系统 [J]. 电子技术与软件工程, 2022, (01): 138-41.
- [13]张琦, 崔琳, 王超. 基于YOLOv5和LPRnet的车辆精细化识别研究 [J]. 长江信息通信, 2022, 35(03): 40-3.
- [14]刘中正. 基于ARM的六轴机械臂控制系统的开发 [D]; 华侨大学, 2021.
- [15]陶文玲, 侯冬青. PyQt5与Qt设计师在GUI开发中的应用 [J]. 湖南邮电职业技术学院学报, 2020, 19(01): 19-21.
- [16]王正奕. 基于PyQt框架的三维目标检测工具设计与实现 [D]; 电子科技大学, 2022.
- [17]蔡俊, 赵超, 沈晓波, et al. 基于SIFT算法和PyQt的停车位标志识别系统 [J]. 湖北大学学报(自然科学版), 2022, 44(01): 1-7.
- [18]黄玉萍, 梁炜萱, 肖祖环. 基于TensorFlow和PyTorch的深度学习框架对比分析 [J]. 现代信息科技, 2020, 4(04): 80-2+7.
- [19]张康林, 叶春明, 李钊慧, et al. 基于PyTorch的LSTM模型对股价的分析与预测 [J]. 计算机技术与发展, 2021, 31(01): 161-7.
- [20]许婷婷. 基于Python的行人重识别系统设计与实现 [J]. 信息与电脑(理论版), 2022, 34(23): 176-8.

致 谢

在我的论文完成之际,我想要向那些给予我帮助和支持的人们致以最真挚的感谢。首先,我要感谢我的导师曹永忠教授。他在整个研究过程中给予了我很多宝贵的指导和建议,不断鼓励和支持我。他的教诲和指导让我不断提升自己的学术水平,使我能够在论文中提出有价值的贡献。我还要感谢我的家人和朋友们。他们一直在我身边给予我鼓励和支持,帮助我克服了研究过程中的挫折和困难。没有他们的支持和理解,我无法完成这篇论文。

最后,我要感谢所有为我提供帮助和支持的人们。他们的指导、鼓励和支持对我的研究工作产生了重要的影响。尽管无法一一列举,但是我想对他们的支持和帮助表示衷心的感谢。

感谢每一个帮助过我的人,我将一直珍视你们的支持和帮助。