**基于PaddleOCR的商场车库车辆牌号统计管理系统的设计与实现**

|  |  |  |  |
| --- | --- | --- | --- |
| 学 院 | 计算机学院 | | |
| 专 业：  姓 名：  指导老师： | 计算机科学与技术 | | |
| 郑毓文 | 学 号：  得 分： | 190021100819 |
| 胡敬朋 |  |

中国·珠海

二○二二 年 六 月

**诚信承诺书**

**本人郑重承诺：**本人承诺呈交的专业综合实训《基于Arduino的智能家居控制系统的设计与实现》是在指导教师的指导下，独立开展研究取得的成果，文中引用他人的观点和材料，均在文后按顺序列出其参考文献，设计使用的数据真实可靠。

本人签名： 

日期： 2022 年 6 月 23 日

**基于PaddleOCR的商场车库车辆牌号统计管理系统的设计与实现**

**摘 要**

伴随着社会发展，人民生活越来越富裕，生活方式也越来越丰富，大家更愿意为了提高生活品质而消费。伴随着人民生活的多样化，商场也成为人们生活中必不可少的一部分。商场的车库管理是一个关键问题，一个好的车库管理系统能够让一个商场获得更多的人流量。

本文基于PaddleOCR的文字识别系统，从车牌入手，设计并实现一个商场车库车辆牌号统计管理系统，该系统连接一个Mysql数据库，每个商场拥有其中一个表的使用权。对识别到的车牌号进行处理，比对数据库的数据，进行初始化写入或更新。内置了一个等级系统，根据某个车牌号进入该商场的总次数进行评级。同一个车牌号来到该商场的次数越多，享受到该商场的优惠就越大。对于一个家庭来说，这或许更能够让他们倾向于使用同一辆车而不是同时使用多辆车进入商场。这在某种程度上也缓解了商场附近的交通压力，同时也让商场在相同人流量的情况下拥有更多的空闲车位，以便带来更多的人流量。

该系统软硬件结合，由于硬件只需要商场自带的两个进出车库摄像头，所以成本很低，整体更侧重于软件部分的实现。PaddleOCR的文字识别精准度非常高，车牌数据识别效果很好。运用文字识别算法和数据库的联动可以很方便的完成对车库的管理，并且可以长时间持续运行。

**关键词：**文字识别；数据库；实时监测；车牌

**Design and implementation of a statistical management system for vehicle license plates in shopping mall garages based on PaddleOCR**

**Abstract**

With social development, people's lives are getting richer and their lifestyles are getting richer, and people are more willing to consume to improve their quality of life. With the diversification of people's lives, shopping malls have also become an indispensable part of people's lives. The garage management of shopping malls is a key issue, and a good garage management system can make a mall get more traffic.

Based on the text recognition system of PaddleOCR, starting from the license plate, this paper designs and implements a statistical management system for the vehicle license plate number of a shopping mall garage. The system is connected to a Mysql database, and each shopping mall has the right to use one of the tables. The recognized license plate number is processed, compared with the data in the database, and initialized writing or updating is performed. There is a built-in rating system, which is rated based on the total number of times a certain license plate number has entered the mall. The more times you come to the mall with the same license plate number, the greater the discount you enjoy in the mall. For a family, this may be more likely to make them more inclined to use the same vehicle rather than multiple vehicles at the same time to enter the mall. This also relieves the traffic pressure near the mall to a certain extent, and also allows the mall to have more free parking spaces under the same traffic flow, so as to bring more traffic.

The system is a combination of software and hardware. Since the hardware only needs two cameras in and out of the garage, the cost is very low, and the overall focus is on the realization of the software part. The text recognition accuracy of PaddleOCR is very high, and the license plate data recognition effect is very good. The use of text recognition algorithm and database linkage can easily complete the management of the garage, and can continue to run for a long time.

**Keywords:**  text recognition database; real-time monitoring; license plate

**目 录**

[1 绪论 1](#_Toc106134642)

[1.1 商场车库管理发展现状 1](#_Toc106134643)

[1.2 设计内容及问题 1](#_Toc106134644)

[2 系统需求分析与结构 3](#_Toc106134645)

[2.1 商场车库车辆牌号统计管理系统的需求分析 3](#_Toc106134646)

[2.1.1 需求概论 3](#_Toc106134647)

[2.1.2 用户需求 3](#_Toc106134648)

[2.1.3 功能需求 3](#_Toc106134649)

[2.2 商场车库车辆牌号统计管理系统的可行性分析 4](#_Toc106134650)

[2.2.1 系统技术可行性分析 4](#_Toc106134651)

[2.3 商场车库车辆牌号统计管理系统的结构 5](#_Toc106134652)

[3 商场车库车辆牌号统计管理系统的设计 6](#_Toc106134653)

[3.1 系统的开发环境 6](#_Toc106134654)

[3.2 系统功能的设计 6](#_Toc106134655)

[3.2.1 PaddleOCR车牌文字识别模块的设计 6](#_Toc106134656)

[3.2.2 数据库的设计 8](#_Toc106134657)

[4 系统测试 9](#_Toc106134658)

[4.1 测试内容 9](#_Toc106134659)

[4.1.1 PaddleOCR车牌文字识别模块测试 9](#_Toc106134660)

[4.1.2 数据库模块测试 10](#_Toc106134661)

[4.2 测试结果 12](#_Toc106134662)

[5 总结与改进 13](#_Toc106134663)

[5.1 商场车库车辆牌号统计管理系统总结 13](#_Toc106134664)

[5.2 商场车库车辆牌号统计管理系统 13](#_Toc106134665)

[参考文献 14](#_Toc106134666)

[谢 辞 15](#_Toc106134667)

[附 录 16](#_Toc106134668)

# 1 绪论

如今，随着经济的迅速发展，商场的数量越来越多，愿意前往商场的人也越来越多。于此同时，随着社会的发展，车辆也越来越多，大多数人都是开车前往商场。随着车辆的普及以及人们生活的改善，商场逐渐成为了一个车辆聚集地。越来越多的商场以及越来越多的车辆，这不仅对商场车库来说是一个负担，同时也会影响到附近的交通状况。一个好的车辆管理系统不仅可以为商场提供更多的利益，对于周边环境以及交通也有着很重要的作用。可以说，车库管理与商场的关系密不可分。如何对商场的车辆进行管理？车牌识别是一个很好的选择。车牌可以唯一的标识一辆车。本设计是一个集OCR文字识别，Mysql数据库，以及前端摄像头为一体的商场车库车辆牌号统计管理系统。

基于PaddleOCR的商场车库车辆牌号统计管理系统的设计与实现，是一个通过摄像头对车牌进行图像识别，然后通过后端处理信息，最后通过连接的数据库进行数据写入以及更新的系统。本系统分为两个模块，一个是PaddleOCR文字识别模块，一个是数据库写入以及更新模块。PaddleOCR文字识别功能强大，识别速度快，结果精准，非常适合用于商场车库的车牌识别，结合数据库能够达到很好的管理效果。

## 1.1 商场车库管理发展现状

八十年代，我国汽车保有量少，对停车场管理系统的需求有限，主要依靠引进国外停车场设备来满足这小部分的需求。随着改革开放的不断深入，我国经济开始腾飞，国内汽车保有量逐年上升，催生停车场设备的巨大需求。九十年代初，我国出现了一些停车场设备自主研发和生产的企业，由此正式开启了我国停车场管理系统行业。近年来，许多企业结合国内行业现状和特点开发出较高技术水平的停车设备，相关技术得到大幅提升。随着我国停车场管理系统行业技术的不断升级，该行业也进入了由老式管理系统向新型管理系统升级的高峰时期，落后的传统接触读写管理系统正在逐渐被淘汰。到了现在，智能的车库管理系统正是大势所趋。

一个智能的商场车库车辆牌号统计管理系统能够给车库提供更好的管理，给商场带来更多的利益，并且能够相应地减缓交通压力。

**1.2 设计内容及问题**

本系统主要是使用PaddleOCR文字识别系统以及数据库，实现车库进出车辆的车牌号信息的采集与功能的实现，通过商场设置在车库出入口的摄像头采集车牌信息，通过连接的MySQL数据库对相关数据进行写入或更新。后续可加载车牌剪切功能，优先识别车牌，将其截取后再使用PaddleOCR进行文字识别，提高准确度。

本系统达到的目标如下：

（1）在商场车库入口，通过PaddleOCR文字识别，识别出进出车库的车牌号；

（2）设定一个数据库，使用MySQL，将识别到的车牌号统计入数据库；

（3）即将统计入数据库时进行以下操作：

1.若数据库内不存在该车牌号，则新建一行数据用于统计该车牌号的相关数据，同时将该车牌进商场总次数初始化为1，在场状态置1（1表示在场，0表示离开），并记录时间；

2.若数据库内存在该车牌号，则将在场状态取反，且若进商场总次数为偶数（进商场），则将该车牌进商场总次数加1，否则总计数不变，最后记录时间；

**2 系统需求分析与结构**

**2.1 商场车库车辆牌号统计管理系统的需求分析**

**2.1.1 需求概论**

近年来，随着经济的快速发展，各个城市的各种商场越来越多，同时也越来越多的人愿意前往商场。在这样超大量的人流量下，车库管理成为了每个商场必须要解决的最重要问题之一。 商场管理者意识到车库管理系统能够极大的改善车库管理，同时也能为商场带来更大的流量。在这些年的车库管理系统的发展中，各个商家也意识到车库管理系统不仅要功能齐全，运行稳定，操作也要相对简单，成本也不必要太高。于是，商场车库车辆牌号统计管理系统应运而生。在普通车库管理系统的基础上，从车辆识别转为车牌识别，从根本上改变车库管理系统的分享。车牌具有唯一性和持久性，通过识别车牌对车库进行管理，要比通过识别车辆对车库进行管理准确快速得多。同时数据库能够记录来往的车辆，后续能对车牌数据进行分析，得到某个群体经常关顾什么地方，从而进行更好的商业规划。对于顾客来说，进入有商场车库车辆牌号统计管理系统的车库，进入车库时仅需进行一次识别便能记录信息，出车库也仅需进行一次识别，便可以通过系统数据进行对应的收费从而离开商场，十分便利。

**2.1.2 用户需求**

商场车库车辆牌号统计管理系统是一个针对大多数商场车库的需要，高精度，快速识别，具有数据库存储信息等多方面的控制系统。了解和分析大多数商场的运营管理，需要实现系统的精度高，速度快，成本低，配置也不需要太高。系统要实现车牌号码的识别采集，还要实现数据库的功能，车牌信息采集完毕后与数据库内部数据进行比对分析，完成一次出入统计，方便商场了解当前车库状况以及近期、长期状况。

根据分析，系统需满足一下用户需求：

（1）考虑到有时候商场车流量大，需要系统的精度和速度达到一定程度，本系统使用PaddleOCR文字识别进行车牌识别功能，精度高的同时识别速度也快。

（2）商场车库车辆牌号统计管理系统最基础的功能实现需要有文字识别模块，数据库模块，这两大功能模块，后续可发展数据分析与UI模块。

（3）商场车库车辆牌号统计管理系统需要长时间的开启，整个系统的功耗和稳定性要到达足够的水平。

**2.1.3 功能需求**

通过对商场车库需求的分析，确定了本个系统功能由PaddleOCR文字识别模块，数据库模块，两大功能模块组成。以下是各功能模块实现的具体功能：

1. PaddleOCR文字识别模块需求：实时视频，定时锁帧识别，将识别到的结果传递给数据库。
2. 数据库模块功能需求：使用pymysql连接MySql的数据库，PaddleOCR文字识别模块识别到的结果传递给数据库，数据库检索当前车牌号信息，若数据库内无相同的车牌id，则将当前识别到的车牌号作为初次录入处理，初始化所有数据——总次数初始化为1，在场状态置1（1表示在场，0表示离开），并记录时间。若数据库内存在对应的车牌id，则对对应数据进行更新处理，将在场状态置反，若进商场总次数为偶数（进商场），则将该车牌进商场总次数加1，否则总计数不变。根据当前商场总次数更新VIP等级，一共五级，具体阈值可由商家自行更改。最后记录当前时间。

**2.2 商场车库车辆牌号统计管理系统的可行性分析**

可行性分析是通过对商场车库车辆牌号统计管理系统的整体系统和相关信息，从[技术](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF/832247)、经济、工程等方面进行整体的一个分析以及调查，进而提供一个综合性的系统的分析方法，为项目决策做准备。提出有关于商场车库车辆牌号统计管理系统这个项目的投资价值和构建该项目的意见，可行性分析应具有前瞻性、公平性、稳定性、科学性的特点。

**2.2.1 系统技术可行性分析**

商场车库车辆牌号统计管理系统主要通过软件部分来实现基本功能，软件部分主要是通过python实现。

首先是车牌文字识别模块。本系统的车牌文字识别模块使用的是PaddleOCR。OCR全称为Optical Character Recognition，即光学字符识别技术，主要由DB文本检测、检测框矫正和CRNN文本识别三部分组成。PaddleOCR 中集成了很多 OCR 算法，文本检测算法有 DB、EAST、SAST 等等，文本识别算法有 CRNN、RARE、StarNet、Rosetta、SRN 等算法。本系统使用的是ch\_ppocr\_server\_v2.0\_det 推理模型，这是一个PaddleOCR文字识别通用模型，支持中英文、多语种文本检测，比超轻量模型更大，但效果更好。

同时系统还搭配了一个数据库模块。本系统的数据库模块使用的是Mysql数据库，通过python的pymysql库实现基于python的文字识别模块与MySql数据库的连接。具有高效性、准确性、可发展性与可持续性，能够实现高精度高速度的数据处理。

**2.3 商场车库车辆牌号统计管理系统的结构**

本系统使用的是两个串联模块，即PaddleOCR和数据库模块进行功能实现，首先通过商场车库出入口的摄像头或许进出车辆的车牌信息，通过PaddleOCR模块进行信息采集，若无信息，即无车辆进出，则摄像头不停止地循环获取当前所拍摄的内容。若检测到数据，即有车牌即将进出车库，此时将车牌信息传进Mysql数据库，进行初步数据分析，判断是否新的车牌号码。若是新的车牌号，即检测到数据库内无此车牌id，则将该车牌id录入数据库并初始化其所属所有内容。若数据库内已有该车牌号，则对该车牌id所属所有数据进行更新。在进行数据库操作后，回归最初的实时检测操作。

系统的基础设计方案如下：

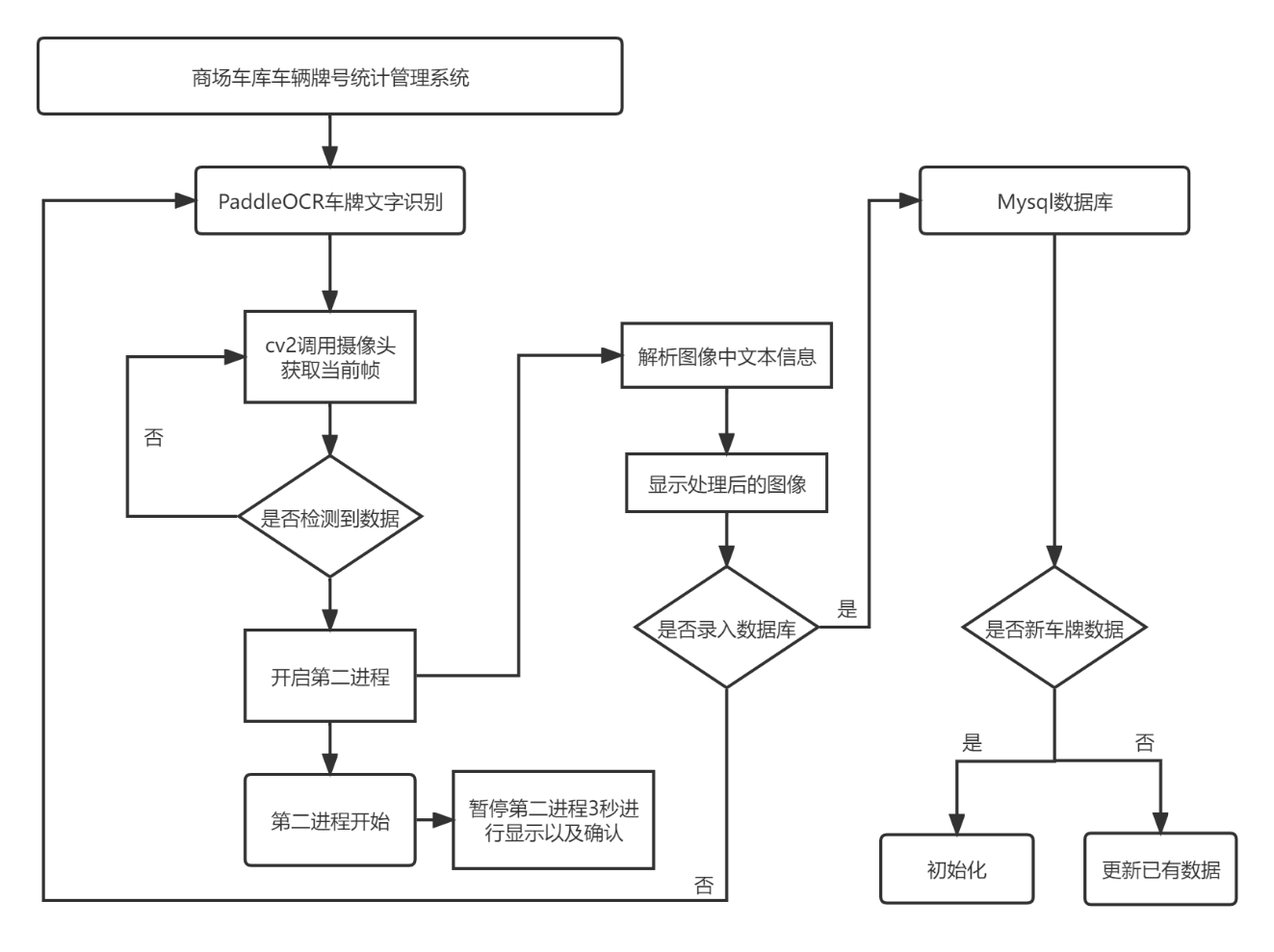


图2.1 系统设计结构图

**3 商场车库车辆牌号统计管理系统的设计**

**3.1 系统的开发环境**

本系统的开发环境是window10，代码环境是在python3.8上，搭载了opencv 4.4版本，以及部分其他第三方库的使用，详见图3.1。数据库方面使用了Mysql 8.0版本。

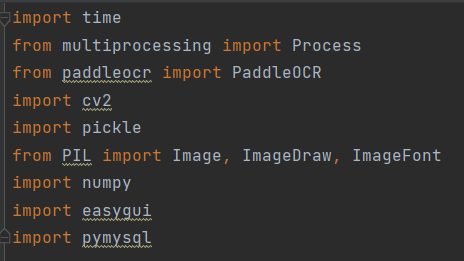


图3.1 使用的库

**3.2 系统功能的设计**

本系统为了确保准确性，需要一名操作人员对输入数据进行点击确认，防止由于锁帧时车辆位置有偏移导致识别位置发生错误，其余功能均为智能自动进行。本系统的基本功能实现由两个模块组成，分别是PaddleOCR车牌文字识别模块以及数据库模块。PaddleOCR车牌文字识别模块实现关于车牌的文字识别并传递给数据库模块对应的数据，数据库模块接收数据并对数据进行入库操作。

**3.2.1 PaddleOCR车牌文字识别模块的设计**

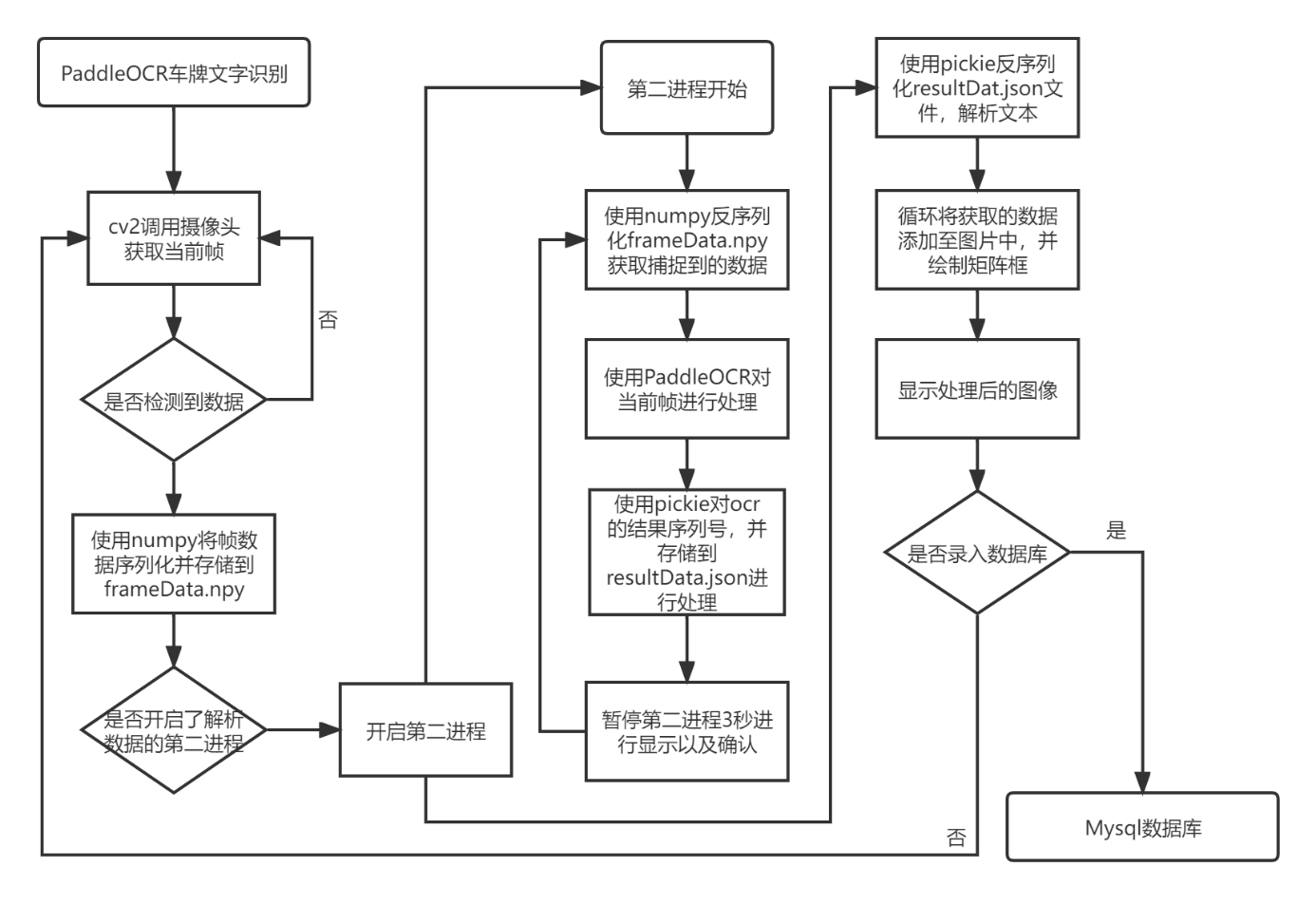
本功能能够实现根据摄像头获取的实时图像，通过PaddleOCR文字识别模块识别当前进出车辆的车牌号码。功能图如图3.1所示。

图3.1 PaddleOCR功能图

PaddleOCR分为文本检测、文本识别和方向分类器三部分，其中文本检测有三个模型，分别是MobileNetV3、ResNet18\_vd和ResNet50。对于文本检测，有97k个训练图像和500个验证图像。在训练图像中，68K幅图像是真实的场景图像，这些图像来自一些公共数据集和百度图像搜索。对于方向分类，有600k训练图像和310K验证图像。在训练图像中，100K图像是真实场景图像，来自公共数据集（LSVT、RCTW-17、MTWI 2018）。对于文本识别，有17.9M的训练图像和18.7K的验证图像。在训练图像中，有190万张是真实场景图像，这些图像来自一些公共数据集和百度图像搜索。使用的公共数据集包括LSVT、RCTW-17、MTWI 2018和CCPD 2019。

具体模块流程设计如下：首先使用cv2等工具包获取摄像头实时拍摄到的图像，使用numpy将帧数据序列化并存储到frameData.npy，然后将数据送入检测模型预测，开始第二进程。在第二进程中，使用numpy反序列化frameData.npy获取捕捉到的数据，然后使用PaddleOCR对当前帧进行处理，再使用pickie对ocr的结果序列号，并存储到resultData.json进行处理，最后暂停第二进程3秒进行显示以及确认，第二进程结束。同时，在第二进程完成一次后，主进程中使用pickie反序列化resultDat.json文件，解析文本，循环将获取的数据添加至图片中，并绘制矩阵框，最后显示处理后的图像并确认是否将识别到的车牌id录入数据库。若取消录入则返回摄像头循环，若确认录入则进入数据库模块。

**3.2.2 数据库的设计**

数据库部分使用Mysql数据库进行设计实现。通过获取PaddleOCR车牌文字识别模块传递的数据，比对数据库内容，进行相关数据库操作。数据库检索当前车牌号信息，若数据库内无相同的车牌id，则将当前识别到的车牌号作为初次录入处理，初始化所有数据——总次数初始化为1，在场状态置1（1表示在场，0表示离开），并记录时间。若数据库内存在对应的车牌id，则对对应数据进行更新处理，将在场状态置反，若进商场总次数为偶数，则将该车牌进商场总次数加1，否则总计数不变。根据当前商场总次数更新VIP等级，一共五级，具体阈值可由商家自行更改。最后记录当前时间。功能图如图3.2所示。

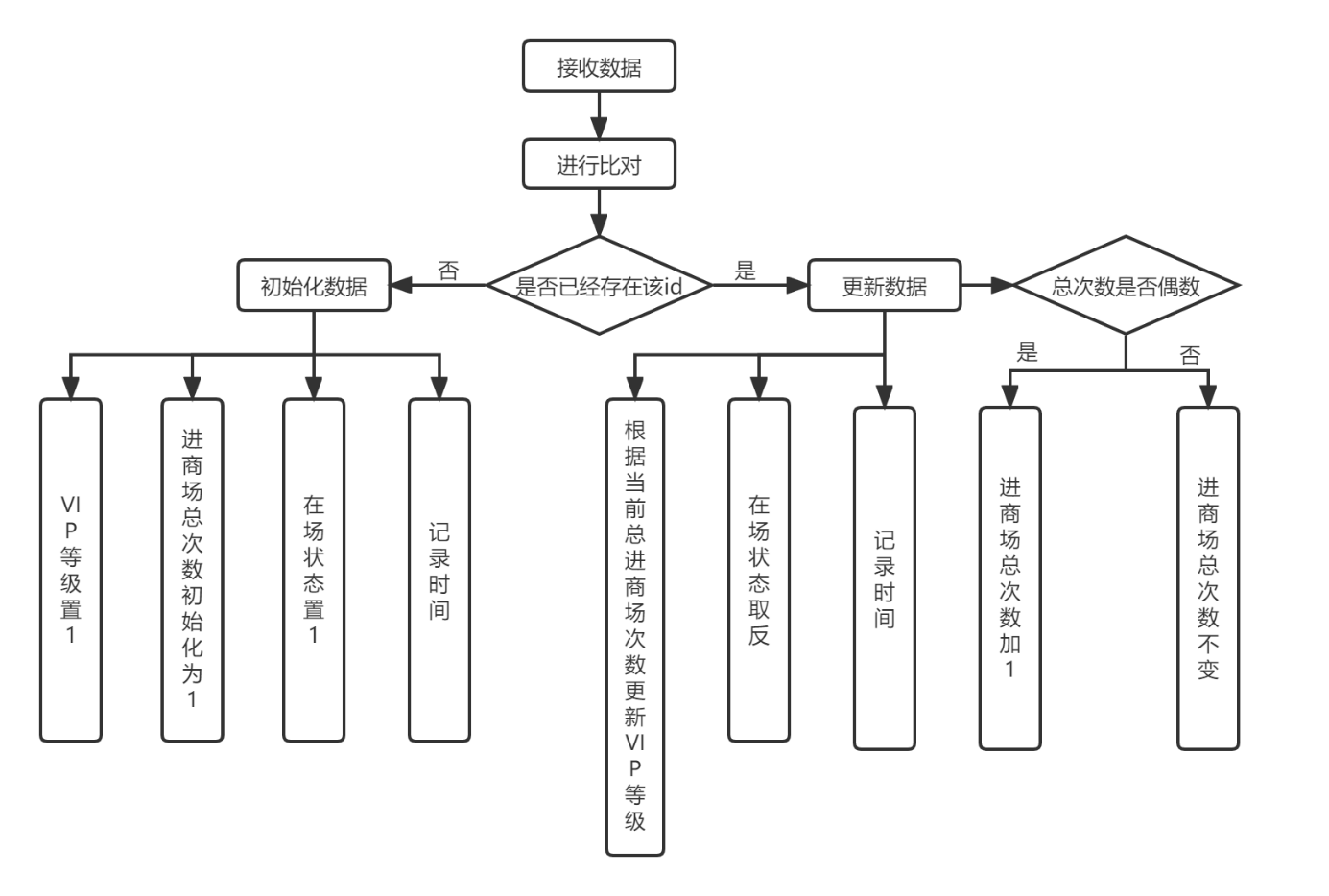


图3.2 数据库功能设计

**4 系统测试**

**4.1 测试内容**

**4.1.1 PaddleOCR车牌文字识别模块测试**

打开video.py文件并运行代码，可打开实时摄像头，开启检测。设定是每三秒进行一次检测，若检测到当前图像内有数据，立即锁帧并开启弹窗，让用户确认车牌号是否正确，有确认车牌和取消录入两个选项，防止有错误数据录入数据库。

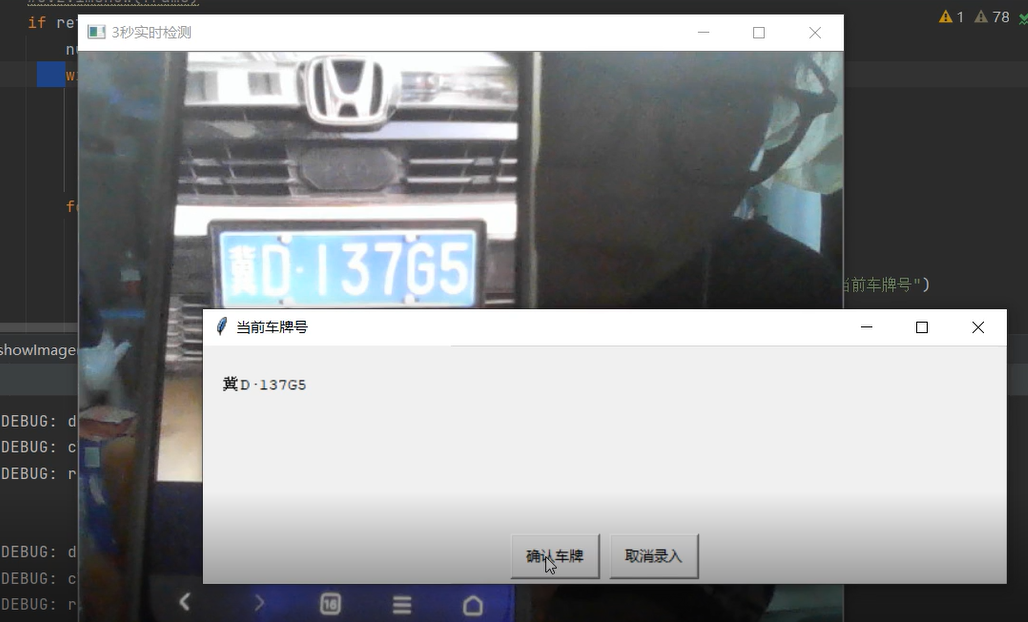


图4.1 PaddleOCR车牌文字识别模块测试1

几乎是即时给出结果，高速度高精度。

在弹窗存在期间，视频不会继续检测。此时，若点击取消录入，则不会进行任何数据库操作，继续进行实时监控捕捉车牌号码，如图4.2。

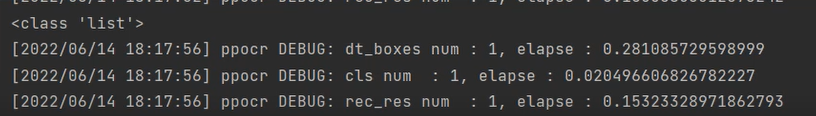


图4.2 取消录入检测到的数据

如果点击确认车牌，则会根据数据库的内容继续判断，自动给出决策。本系统设定有两种决策，一个是该车牌id在数据库中是第一次出现，另一个是该车牌id在数据库中曾经出现过。图4.3是该车牌id在数据库中曾经出现过的结果。

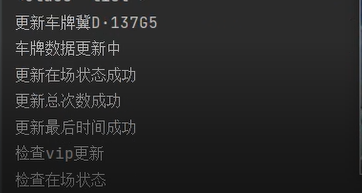


图4.3 在数据库中曾经出现过

若检测到一个在当前商场车库从未出现过的车牌id，则如图4.4，图4.5所示，此时数据库会更新数据。

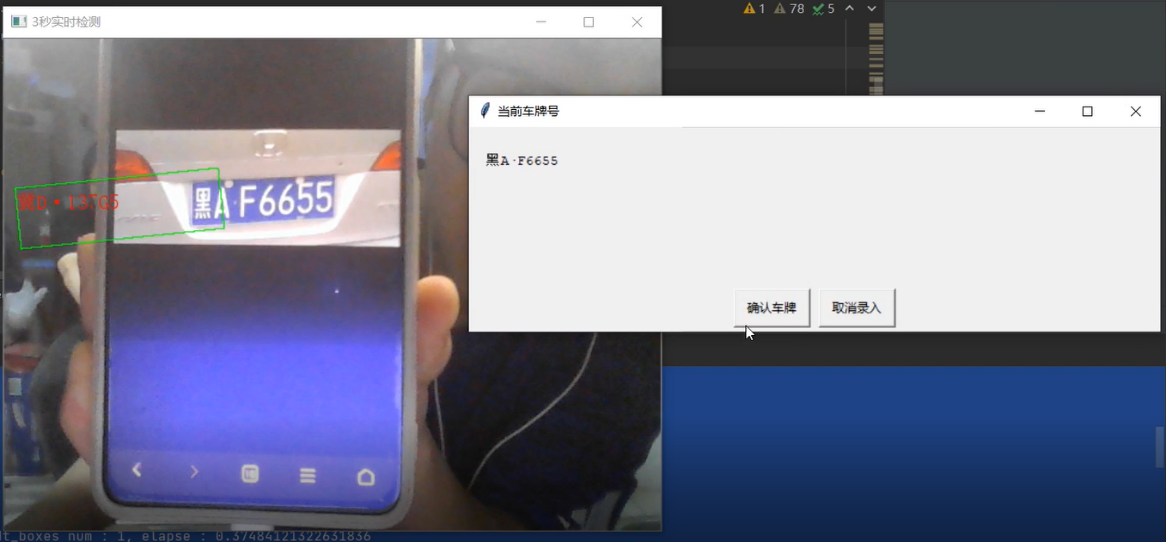


图4.4 PaddleOCR车牌文字识别模块测试2

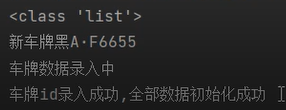


图4.5 录入新车牌

**4.1.2 数据库模块测试**

在PaddleOCR车牌文字识别模块识别到结果后，将会进入数据库模块进行数据库操作。数据库录入基本逻辑如表4.1。

表4.1 数据库操作

|  |  |  |
| --- | --- | --- |
| 初步检测情况 | 操作步骤 | 预期结果 |
| 车牌id在数据库中第一次出现 | 初始化数据 | （1）总次数初始化为1。  （2）在场状态置1。  （3）记录时间。 |
| 车牌id在数据库中曾经出现 | 更新数据 | （1）将在场状态置反。  （2）若进商场总次数为偶数，则将该车牌进商场总次数加1，否则总计数不变。  （3）根据当前商场总次数更新VIP等级.  （4）最后记录当前时间。 |

在PaddleOCR车牌文字识别模块测试中，图4.1的录入结果如图4.6所示，可见在数据库中，由于该车牌曾经出现过，因此根据数据库设计的逻辑进行数据库操作结果如下。



图4.6 已有车牌更新数据

图4.3的录入结果如图4.7所示，可见在数据库中，由于该车牌是第一次出现过，因此根据数据库设计的逻辑进行数据库操作结果如下。



图4.7 新车牌初始化数据

**4.2 测试结果**

本次测试测试了系统的所有功能，目前系统通过了本次测试的所有测试，表明系统目前可以稳定正常的运行。所有测试的最后结果见图4.8。

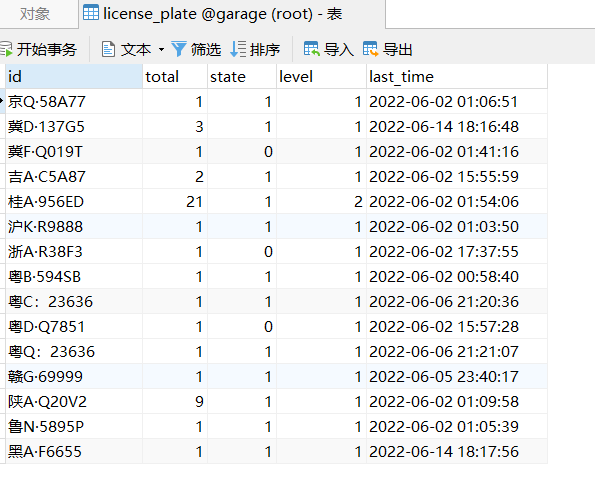


图4.8 最终测试结果

**5 总结与改进**

**5.1 商场车库车辆牌号统计管理系统总结**

本设计通过PaddleOCR车牌文字识别模块与数据库模块两大模块的结合，实现了商场车库出入口识别出入车辆的车牌，并载入数据库的基础功能。本设计使用PaddleOCR继续文字识别，速度快精度高，并且硬件少，这保证了系统的准确性，降低了整体功耗，削减了开发整个系统的成本。在数据库模块方面，使用了Mysql数据，速度快的同时错误率低，大多数基础功能实现都是通过操作数据库来完成，本系统的数据库设计也是较为完善，基本没有出现什么问题。在设计中也碰到些麻烦，在UI的设计上由于摄像头的实时检测代码与一般的UI设计有所冲突，导致最后UI设计仅有一个弹窗。不过总体的基础功能实现没有影响，很好地完成了系统的整体功能。

**5.2 商场车库车辆牌号统计管理系统**

本设计还存在着很多不足，由于时间和能力上的限制，一些一开始想要实现的功能并没有完成，所以系统还有很多可以做得更好的地方：

1. 在PaddleOCR车牌文字识别之前可以加入yolo的图像检测，检测车牌位置并剪切，之后再对剪切的部分进行文字识别准确率会高很多，不会识别到无用信息。
2. 本设计前端UI仅有一个弹窗，外观美观性不够，UI还可以继续优化。
3. 设计中数据库模块还能继续发展更多的功能。

# 参考文献

1. 刘树春 贺盼 马建奇 王佳军. 深度实践OCR:基于深度学习的文字识别[J]. 机械工业出版社, 2020-04.
2. 明日科技. Python OpenCV从入门到精通[J]. 清华大学出版社.
3. 丁晓青 王言伟等. 文字识别：原理、方法和实践[J]. 清华大学出版社，2017-04-01.
4. 赵永江. 楼宇的门禁、监控及车库管理系统/楼宇电气系统安装运行维护丛书[J]. 中国电力出版社，2005-07-01.
5. 明日科技. SQL Server从入门到精通[J]. 清华大学出版社，第四版.
6. (美)施瓦茨、扎伊采夫、特卡琴科. 高性能MySQL(第3版) [J]. 电子工业出版社，2021-04-01.
7. 秦昳 罗晓霞 刘颖. 数据库原理与应用[J]. 清华大学出版社，2021-09.
8. 数据库原理及应用教程[J]. 人民邮电出版社，2022-03.
9. 周浦城. 深度卷积神经网络原理与实践[J]. 电子工业出版社，2020-10.
10. 丛晓峰 彭程威 章军. PyTorch神经网络实战[J]. 机械工业出版社，2022-06

# 

# 谢 辞

时间不等人，时光飞逝。伴随着专业综合实训论文的完成，本次专业综合实训课程也到达了尾声。

在此，我要首先感谢胡敬朋老师在本次实训课对我们的教导，本设计是由胡敬朋老师的悉心教导下完成的。从实训选题，到开题报告再到完成论文，这整个过程中都是胡敬朋老师在耐心的指导我完成。碰到不懂的技术问题，胡敬朋老师也仔细的给我讲解。我从胡敬朋老师感受到了对技术的认真和对教育事业的热情，从中我也学到了很多，胡老师的严格要求也让我努力成为更好的自己。

完成整个实训课程对我大学的学习有着很大的帮助，本次综合实训课主要内容综合了以往在大学中学习的大部分课程，对这些课程所学习的内容加以学习，理解以及应用，最后完成了这一整个实训作品，不仅扎实了以往所学的知识，更是学习掌握了如何去实践。

不知不觉本次综合实训已经结束，我会永远记得在这个课程上所学习的内容以及应用。

**附 录**

附录1 代码整体框架

import time

from multiprocessing import Process

from paddleocr import PaddleOCR

import cv2

import pickle

from PIL import Image, ImageDraw, ImageFont

import numpy

import easygui

import pymysql

# 模型路径下必须含有model和params文件

ocr = PaddleOCR(use\_angle\_cls=True,

use\_gpu=False) # det\_model\_dir='{your\_det\_model\_dir}', rec\_model\_dir='{your\_rec\_model\_dir}', rec\_char\_dict\_path='{your\_rec\_char\_dict\_path}', cls\_model\_dir='{your\_cls\_model\_dir}', use\_angle\_cls=True

img\_path = 'PaddleOCR/doc/result/result.jpg'

global result

global frame

def getImformation(name):

# global frame,result

while (True):

try:

frame = numpy.load('frameData.npy')

result = ocr.ocr(frame, cls=True)

pickle.dump(result, open('resultData.json', 'wb'))

print(type(result))

time.sleep(3)

except:

# i = 1

time.sleep(3)

class System():

def showImage(self):

cap = cv2.VideoCapture(0)

p = Process(target=getImformation, args=("Process1",))

p.start()

self.conn = pymysql.connect(host='localhost', user='root', password='123456', charset='utf8mb4', database='garage')

print('连接数据库成功')

while (True): #开始循环检测

ret, frame = cap.read()

#cv2.imshow(frame)

if ret:

numpy.save('frameData', frame)

with open('resultData.json', 'rb') as f:

try:

result = pickle.load(f)

#print(result)

except:

result = []

for line in result:

print( )

print(line)

number = line[-1][0]

fin = easygui.buttonbox(msg = number,choices = ('确认车牌','取消录入'),title = "当前车牌号")

self.sql(fin, number)

# cv2.rectangle(frame, (int(line[0][0][0]-2), int(line[0][0][1]-2)), (int(line[0][2][0]+2), int(line[0][2][1]+2)),

# (0, 255, 0), 3)

frame = cv2ImgAddText(frame, line[1][0], int(line[0][0][0] + 2), int(line[0][0][1] + 2), (255, 0, 0),

20)

draw\_rectangle(frame, line)

cv2.imshow("3秒实时检测", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

# 关闭连接

self.conn.close()

print('数据库断开连接')

break

time.sleep(3)

def sql(self,fin,number):

if fin == "确认车牌":

judgment = "select exists (SELECT \* from license\_plate where id = '%s');" % (number) # 查看数据库是否存在该车牌

cursor = self.conn.cursor()

cursor.execute(judgment) # 执行SQL语句

flag = cursor.fetchall()[0][0]

# print(flag)

self.conn.commit() # 提交到数据库

cursor.close() # 关闭游标

if (flag == 0): # 此车牌第一次进入该车库

print('新车牌' + number)

print("车牌数据录入中")

# easygui.msgbox("车牌数据录入中", title="车辆数据录入结果")

sql = "INSERT INTO license\_plate (id,state,total,level,last\_time) " \

"VALUES('%s',1,1,1,NOW());" % (number) # 录入id

try:

cursor = self.conn.cursor()

cursor.execute(sql) # 执行SQL语句

print("车牌id录入成功,全部数据初始化成功")

# easygui.msgbox("车牌id录入成功,全部数据初始化成功!", ok\_button="继续检测", title = "车辆数据录入结果")

self.conn.commit() # 提交到数据库

cursor.close() # 关闭游标

except:

# 如果发生错误则回滚

self.conn.rollback()

print("车牌数据录入失败，请手动更改数据库或重新录入")

easygui.msgbox("车牌数据录入失败，请手动更改数据库或重新录入", ok\_button="继续检测", title="车辆数据录入结果")

# 关闭游标

cursor.close()

return

return

else:

print('更新车牌' + number)

print("车牌数据更新中")

sql6 = "update license\_plate set total = total+1 where id = '%s' and state = 0;" % (

number) # 更新总次数(进商场加1，出商场不计入)

sql7 = "update license\_plate set state = ABS(state-1) where id = '%s';" % (number) # 更新在场状态

sql8 = "update license\_plate set last\_time = NOW() where id = '%s';" % (number) # 更新最后时间

vip1 = "SELECT total from license\_plate where id = '%s';" % (number) # 检查VIP更新

state1 = "SELECT state from license\_plate where id = '%s';" % (number) # 检查在场状态

vip2 = "update license\_plate set level = 2 where id = '%s';" % (number) # VIP2

vip3 = "update license\_plate set level = 3 where id = '%s';" % (number) # VIP3

vip4 = "update license\_plate set level = 4 where id = '%s';" % (number) # VIP4

vip5 = "update license\_plate set level = 5 where id = '%s';" % (number) # VIP5

try:

cursor = self.conn.cursor()

cursor.execute(sql6) # 执行SQL语句

print("更新在场状态成功")

cursor.execute(sql7) # 执行SQL语句

print("更新总次数成功")

cursor.execute(sql8) # 执行SQL语句

print("更新最后时间成功")

cursor.execute(vip1) # 执行SQL语句

print("检查vip更新")

vipflag = cursor.fetchall()[0][0]

cursor.execute(state1) # 执行SQL语句

print("检查在场状态")

stateflag = cursor.fetchall()[0][0]

if (vipflag == 20 and stateflag == 1):

cursor.execute(vip2) # 执行SQL语句

print(number + "车主升级为VIP2！")

elif (vipflag == 30 and stateflag == 1):

cursor.execute(vip3) # 执行SQL语句

print(number + "车主升级为VIP3！")

elif (vipflag == 40 and stateflag == 1):

cursor.execute(vip4) # 执行SQL语句

print(number + "车主升级为VIP4！")

elif (vipflag == 50 and stateflag == 1):

cursor.execute(vip5) # 执行SQL语句

print(number + "车主升级为VIP5！")

# print(vipflag)

self.conn.commit() # 提交到数据库

cursor.close() # 关闭游标

except:

# 如果发生错误则回滚

self.conn.rollback()

print("车牌数据更新失败，请手动更新或取消")

# 关闭游标

cursor.close()

return

return

def draw\_rectangle(frame, pos):

cv2.line(frame, (int(pos[0][0][0]), int(pos[0][0][1])), (int(pos[0][1][0]), int(pos[0][1][1])), (0, 255, 0), 1, 4)

cv2.line(frame, (int(pos[0][1][0]), int(pos[0][1][1])), (int(pos[0][2][0]), int(pos[0][2][1])), (0, 255, 0), 1, 4)

cv2.line(frame, (int(pos[0][2][0]), int(pos[0][2][1])), (int(pos[0][3][0]), int(pos[0][3][1])), (0, 255, 0), 1, 4)

cv2.line(frame, (int(pos[0][3][0]), int(pos[0][3][1])), (int(pos[0][0][0]), int(pos[0][0][1])), (0, 255, 0), 1, 4)

def cv2ImgAddText(img, text, left, top, textColor, textSize):

if (isinstance(img, numpy.ndarray)): # 判断是否OpenCV图片类型

img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))

draw = ImageDraw.Draw(img)

# 字体的格式

fontStyle = ImageFont.truetype(

"font/simsun.ttc", textSize, encoding="utf-8")

# 绘制文本

draw.text((left, top), text, textColor, font=fontStyle)

# 转换回OpenCV格式

return cv2.cvtColor(numpy.asarray(img), cv2.COLOR\_RGB2BGR)

def main():

System().showImage()

if \_\_name\_\_ == "\_\_main\_\_":

main()

其中，使用PaddleOCR进行检测的是System类中的showImage函数，操作数据库的是System类中的sql函数。

附录2 数据库创建表

import pymysql

# 创建连接

conn = pymysql.connect(host='localhost',user='root',password='123456',charset='utf8mb4',database='garage')

# 创建游标

cursor = conn.cursor()

# 创建表

sql\_2 = '''CREATE TABLE `license\_plate` (

`id` varchar(255) NOT NULL, #车牌号

`total` INT DEFAULT NULL, #总次数

`state` INT DEFAULT NULL, #在车库的状态

`level` INT DEFAULT NULL, #贵宾等级

`last\_time` datetime DEFAULT NULL, #最后离开时间

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

'''

cursor.execute(sql\_2)

附录3 数据库测试

import pymysql

# 查询所有字段

def list\_col(tabls\_name):

db = pymysql.connect(host='localhost',user='root',password='123456',charset='utf8mb4',database='garage')

cursor = db.cursor()

cursor.execute("select \* from %s" % tabls\_name)

col\_name\_list = [tuple[0] for tuple in cursor.description]

db.close()

return col\_name\_list

# 列出所有的表

def list\_table():

db = pymysql.connect(host='localhost',user='root',password='123456',charset='utf8mb4',database='garage')

cursor = db.cursor()

cursor.execute("show tables")

table\_list = [tuple[0] for tuple in cursor.fetchall()]

db.close()

return table\_list

tables = list\_table() # 获取所有表，返回的是一个可迭代对象

print(tables)

for table in tables:

col\_names = list\_col(table)

print(col\_names) # 输出所有字段名