**工程实践报告**

**题目：基于CNN的车牌识别系统**

**项目提出者： Gavin**

**撰 写 人： ltr**

目录

[1项目背景 3](#_Toc13870536)

[2需求分析 3](#_Toc13870537)

[2.1可行性分析 3](#_Toc13870538)

[2.1.1技术可行性 4](#_Toc13870539)

[2.1.2经济可行性 4](#_Toc13870540)

[2.2功能性需求分析 4](#_Toc13870541)

[2.2.1系统用例图 5](#_Toc13870542)

[2.3非功能性需求 5](#_Toc13870543)

[2.4业务流程分析 5](#_Toc13870544)

[2.4.1技术路线 6](#_Toc13870545)

[2.5支持平台 6](#_Toc13870546)

[3概要设计 6](#_Toc13870547)

[3.1总体设计 7](#_Toc13870548)

[3.1.1需求规定 7](#_Toc13870549)

[3.1.2概念和处理流程 7](#_Toc13870550)

[3.1.3功能模块与程序 7](#_Toc13870551)

[3.2接口设计 8](#_Toc13870552)

[3.2.1用户接口 8](#_Toc13870553)

[3.2.2外部接口 8](#_Toc13870554)

[3.2.3内部接口 8](#_Toc13870555)

[3.3运行设计 8](#_Toc13870556)

[3.3.1运行控制 8](#_Toc13870557)

[3.3.2运行时间 8](#_Toc13870558)

[3.3.3运行模块组合 9](#_Toc13870559)

[4测试与分析 9](#_Toc13870560)

[4.1测试环境 9](#_Toc13870561)

[4.2测试方法 9](#_Toc13870562)

[4.3准确率 10](#_Toc13870563)

[4.4时延 11](#_Toc13870564)

# 1项目背景

随着我国现代化和城市化水平的不断进步，人们的生活水平日益提高。近年来，我国机动车的保有量得到快速增长。这些车辆给我们的日常生活提供相当的便利，同时，也给我们的交通管理，城市管理带来了一些问题，例如车辆的乱停乱放，违犯交通标识和信号灯规则等交通违章现象日益突出。

另一方面，由于计算机处理速度的提升和图像处理技术，模式识别技术的飞速发展，车牌号码的自动识别成为可能。近年来，车牌号码自动识别技术在智能交通监控和智能停车场管理系统中得到广泛应用，使得传统的车牌登记，监控和管理发生了重大变化。车牌号码自动识别技术是为了提高车辆监控和管理效率而产生的一种图像处理或模式识别技术。

通常，车牌号码识别包括车牌图像预处理，车牌号码定位，车牌字符分割，字符识别等四个部分。

实际应用中，传统车牌字符识别方法存在诸多不足与困难。比如，我国车牌号码本身就具有复杂的特征：

(1)车牌字符中不仅仅有汉字，还有英文字母和阿拉伯数字。由于汉字的复杂性，其识别难度远远大于字母和数字；

(2)具有多种颜色分类，主要分为四种：黑字黄底，白字蓝底，黑字白底和白字黑底；

(3)车牌格式较多，我国的车牌格式包括：军车车牌，武警车牌，民用车牌，外交车牌，特种车牌，公安警察车牌，消防车牌等；

(4)车牌的悬挂方式不唯一，由于不同汽车公司出产的汽车型号和外形各不相同，导致了车牌的悬挂位置不唯一。

以上种种原因，导致利用传统的图像处理或模式识别技术，很难同时克服全部困难。由于深度神经网络具有很强的学习能力, 本文将在图像处理领域应用最广泛的深度卷积神经网络运用到车牌字符识别技术中，根据车牌字符识别的特点，构造合适的深度卷积神经网络结构，通过大量样本训练，让深度卷积神经网络 (CNN)学习车牌字符的图像特征，进而自动识别出其中的车牌字符。实验结果表明，深度卷积神经网络可克服实际应用中诸多困难，从而获得较高的车牌字符的识别率。

# 2需求分析

进行充分的需求分析是系统生命周期的关键一步，对系统的后续开发有着重要影响。系统或者系统部件要满足合同、标准、规范或者其他正式规定文档所需具有的条件或权能。在这一小节中将从功能性需求和非功能性需求两个方面对系统进行分析。

## 2.1可行性分析

可行性分析是系统实施前的必备工作之一，在进行工作之前必须先了解用户的需求。一个好的系统需要考虑多方面的因素，比如技术，经济，维护，营运等。这些因素对之后的系统开发有重要的影响，所以必须充分对这些因素进行分析才能保证系统高效的开发。

### 2.1.1技术可行性

卷积神经网络是近年发展起来，并引起广泛重视的一种高效识别方法。CNN主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于CNN的特征检测层通过训练数据进行学习，所以在使用CNN时，避免了显示的特征抽取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际的生物神经网络，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。目前CNN网络使用已经比较成熟，也有大量的用例可供参考。这就为本次工程实践打下了技术基础。

### 2.1.2经济可行性

作为实施前不可少的环节，经济可行性评估是在项目实施前对项目在设计、开发和维护阶段的成本与效益进行评估，通过评估确定方案的可行性和最佳的实施方案，为后续投资决策提供科学的依据。

由于本系统采用Python等开源语言和工具实施，并不需要昂贵计算机设备。因此经济方面投入也比较小。

## 2.2功能性需求分析

本小结通过用例图来展示用例和系统各功能之间的关系。本系统面向用户主要有三个基本功能：识别图片，打开摄像头，关闭摄像头。

### 2.2.1系统用例图



打开图片：通过读取系统中图片来进行识别车牌

打开摄像头：通过设备上链接的USB摄像头读取捕获车牌

## 2.3非功能性需求

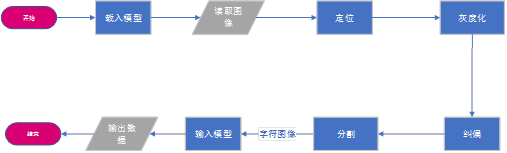
（1）可靠性。可靠性是系统安全的基本要求之一。是指在规定的条件下和时间内，系统能准确完成任务而不产生错误。影响系统可靠性的因素是多种多样的，主要有开发环境，开发方法，软件测试等因素。本系统在开发时考虑了各种情况的出现并进行了必要的测试以确保本系统具有较高的可靠性。

（2）安全性。所有的系统都应该把安全性考虑进去，尤其是一些重要的系统部门。安全性对一个系统开发来说是必须要考虑的问题，一个不安全的系统可能会引起许多问题。本系统设置了用户验证和数据加密等功能以确保用户信息的安全。

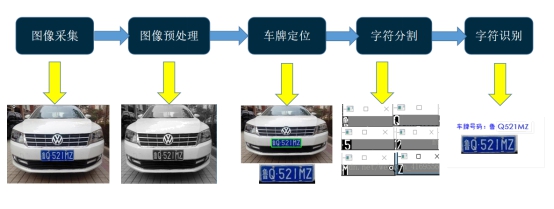
## 2.4业务流程分析

系统主要由两部分组成：模型训练与平台开发。

业务流程图描绘的是一个系统内部的各个单位或者部门人员之间的业务关系、作业的先后顺序的图表。业务流程图的绘制是根据业务的具体处理步骤进行的。通过业务流程图可以看到用户功能实现的虚拟步骤。



### 2.4.1技术路线



## 2.5支持平台

软件：Linux操作系统，WINDOWS系统

硬件：笔记本电脑，树莓派开发板，摄像头

# 3概要设计

概要设计就是利用比较抽象的语言对整个需求进行概括，确定系统的物理配置需求、整个系统的处理流程和系统的数据结构、接口设计、人机界面等，实现对系统的初步设计设计，使得预期读者对整个目标系统有清晰的认识。

## 3.1总体设计

### **3.1.1需求规定**

输入：带车牌的汽车照片

输出：车牌识别结果。

### 3.1.2概念和处理流程

本系统的总体流程可以概括为：

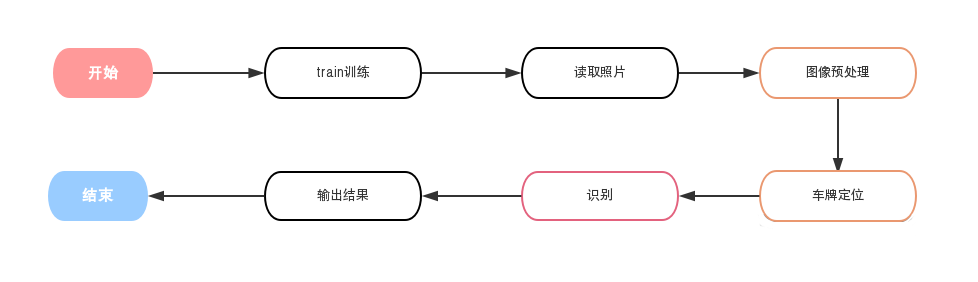
1. 提取charsChinese中的汉字样本和chars2中的数字样本构成训练样本

2. 使用CNN算法对样本进行训练得到模型

3. 爬取网站车牌图片测试样本

4. 通过训练好的模型对车牌照片进行识别

流程图如下：



### 3.1.3功能模块与程序

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块文件及函数名 | 功能 |
| 图片预模块 | test\_cv2.py、test\_transpose.py  predic.py | 图像颜色空间转换、保存标签、灰度变换、滤波、车牌定位、倾斜校正、分割 |
| 数据集制造模块 | carplate\_dataset\_make.py | 按比例抽取图片分为训练集和测试集 |
| CNN模型训练 | carplate\_CNN\_train&save.py  carplate\_CNN\_load&predict.py | 利用样本图片，训练、生成和保存模型 |
| 车牌识别模块 | predic.py | 在图像处理后进行车牌号的识别 |

## 3.2接口设计

### 3.2.1用户接口

用户选择图片的来源（摄像头或本地图片），系统处理后将显示结果。

### 3.2.2外部接口

连接USB摄像头，通过摄像头驱动获取图像，并通过本系统识别车牌。

### 3.2.3内部接口

|  |  |
| --- | --- |
| 图像预处理模块 | test\_cv2.py  test\_transpose.py |
| CNN模型相关模块 | carplate\_CNN\_train&save.py  carplate\_CNN\_load&predict.py |
| 数据集制造模块 | carplate\_dataset\_make.py |
| 界面模块 | surface.py |
| 摄像头拍照模块 | face\_detection.py |
| 车牌定位与识别模块 | predict.py |

## 3.3运行设计

### 3.3.1运行控制

1. 选择图片来源（本地、摄像头）

2. 系统将自动给出识别结果

### 3.3.2运行时间

|  |  |
| --- | --- |
| 图像预处理模块 | 约1ms |
| 数据集制造模块 | 约5s |
| 模型训练模块 | 数小时 |
| 识别模块 | < 20ms |
| 界面模块 | < 1ms |

### 3.3.3运行模块组合

Main主函数

图片获取模块

数据集制造模块

模型训练模块

识别模块

预处理模块

界面模块

# 4测试与分析

## 4.1测试环境

|  |  |
| --- | --- |
| 操作系统 | Win 8/8.1/10 |
| 解释器 | Python 3.5.3 |
| IDE | JetBrains PyCharm Community Edition 2018.3.3 x64 |
| 外部包 | Keras 2.2.4, opencv-python 3.4.4.19 |
| 补充动态库 | libhdf5-dev, libatlas-base-dev, libqt4-test, libqtgui4 |

## 4.2测试方法

本次测试采用了黑盒测试的方法，输入用例均为完整的包含车牌的图片，但图片中车牌区域不保证清晰可见。测试图片均来自互联网，保存在源码文件的test/文件夹中。根据系统的输出结果，来统计系统的性能指标。

## 4.3准确率

识别率是识别系统最关键的性能指标之一。根据系统的输出结果，每个车牌含有7个字符，输出字符与实际车牌字符相同时，则认为该字符识别正确。

其中

## 4.4时延

与准确率的测试方法类似，我们在统计识别成功率的同时，也记录了识别所耗费的时间。

识别成功时的平均时延 = 0.233 s

识别失败时的平均时延 = 0.002 s