# 引言

* 1. 编写目的

本报告是用以描述SmartEeys所要实现的功能，旨在使读者了解到整个作品的系统功能分析与创意设计。该文档计划说明我们的SmartEyes的框架，介绍作品各个方面的具体功能说明和是实现目的。

* 1. 应用背景

随着经济社会的迅猛发展，人们的生活水平的提高，机动车辆的数量也越来越多。为了提高车辆的管理效率，缓解各个停车场压力，我们必须找到一种解决方案。而作为汽车“身份证”的汽车车牌，是在公众场合能够唯一确定汽车身份的凭证。我们可以以此为依据，设计一种车牌识别系统监控各个停车位的情况。为此，我们的决定设计一套自动识别车位上的车辆，确定其“身份”，并进行计费工作

# 特色与创新

2.1 机器学习

# 功能规划

3.0 背景知识介绍

3.0.1 opencv系列函数

Cut.py

|  |  |  |  |
| --- | --- | --- | --- |
| 函数 | 函数名称 | 参数 | 返回值 |
| cv2.imread('plates\_to\_test3/timg1.jpg', -1) | 图像读入函数 | 'plates\_to\_test3/timg1.jpg'是要读取的图片的文件名  参数flag是-1则读出来是原始图像，为0是读出来是灰色图像 | 无 |
| cv2.threshold(cv2.cvtColor(img.copy(), cv2.COLOR\_BGR2GRAY), 125, 255, cv2.THRESH\_BINARY) | 二值化函数 |  |  |
| sorted(contours, key=sort\_func) | 排序函数 | contours为待排数据 ；key 是带一个参数的函数, 用来为每个元素提取比较值. 默认为 None, 即直接比较每个元素. | 返回contours数组 |
| cv2.boundingRect(c) | 矩形边框 | c为需要用矩形包围的车牌图片 | 无 |
| cv2.imwrite(dir + str(i) + 'ch.png', dst) | 保存图片函数 | 第一个参数是保存图片的文件名，第二个参数是读取的图片 | 无 |
| cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 1) | 画矩形函数 | 五个参数分别是图片名称，两个对角点的坐标，颜色数组，线条的宽度 | 无 |

Detect.py

|  |  |  |  |
| --- | --- | --- | --- |
| 函数 | 函数名称 | 参数 | 返回值 |
| void GaussianBlur(InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY=0, int borderType=BORDER\_DEFAULT ) | 高斯滤波函数 | Src和dst分别是输入图像和输出图像ksize为高斯滤波器末班大小，sigmax和sigmay分别为高斯滤波在横线和竖向的滤波系数。Bordertype为边缘点插值类 | Dst |
| cv2.findContours(img,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE)  #该函数接受的是二值图，不是灰度图 | 轮廓检测 | 第一个参数是需要想寻找轮廓的图像；第二个参数表示轮廓的检索模式，本函数中用到的是建立一个等级树结构的轮廓；第三个参数是轮廓的近似方法。 | 返回图像的轮廓 |
| cv2.approxPolyDP(cnt, epsilon, **True**) | 轮廓近似 | 第一个参数是输入的点集；第二个是指定的精度；第三个参数若为true则说明近近似曲线是闭合的 | 返回图像的近似轮廓 |

Generate\_plates.py

|  |  |  |  |
| --- | --- | --- | --- |
| 函数 | 函数名称 | 函数参数 | 返回值 |
| cv2.getPerspectiveTransform(pts1, pts2) | 透视变换矩阵 | 两个参数分别是变换前点后的位置以及变换后点的位置 | 返回透视变换矩阵 |
| cv2.warpPerspective(img, m, size) | 透视变换 | 参数分别表示源图像，变换矩阵，输出图像大小 | 返回透视变换结果 |
| cv2.resize(env, (img.shape[1], img.shape[0])) | 重新调整图像env | 参数分别表示源图像，目标图像 |  |

3.0.2 mxnet系列函数

3.1 车牌切割

3.1.1 车牌切割介绍

这项功能通过对树莓派上连接的摄像头捕捉到的图片进行识别，树莓派会切割出我们所需要的车牌号部分，下面是本部分操作的流程图：

3.1.2 车牌切割示例

下面是通过切割一张图片之后的示例图，图中的绿色框内的就是从图片上切割下来的车牌号 ：

开始

高斯平滑滤波

边缘检测

腐蚀膨胀

查找车牌

3.2 图片二值化

3.2.1图片二值化介绍

3.2.2 图片二值化示例

3.3 字符切割

3.3.1 字符切割介绍

3.3.2 字符切割示例

3.4 数据传输（WiFi）

3.4.1 数据传输介绍

3.4.2 数据传输示例

# 第四章 硬件组成

4.1 树莓派

树莓派是一个微型的计算机，运行相应的程序，对硬件实现操控.

4.2 摄像头

摄像头负责拍照工作

4.3 树莓派电源

负责树莓派的供电工作

4.4 内存卡

为树莓派提供存储空间.

# 第五章 开发环境

5.1 操作系统

操作系统为linux系统，由于树莓派的操作系统为linux，于是我们选择在linux系统下进行图片的切割，字符分割，最后对图片进行归一化处理，得出我们要的车牌号信息。

5.2 OpenCV

OpenCV是一个跨平台的视觉库，可以实现图像处理和计算机视觉方面的很多通用算法。

# 第六章 数据传输应用

6.1 WiFi实现数据传输

6.2 数据库存储车辆信息

# 其他

6.1 成本估算

|  |  |  |  |
| --- | --- | --- | --- |
| 物品名称 | 价格（单位/元） | 数量（个/车位） | 总计（元） |
| 树莓派（Raspberry Pi 3） |  | 1 |  |
| 800万像素摄像头 |  | 1 |  |
| 5v充电头，安卓手机数据线 |  | 1 |  |
| 32GB内存卡 |  | 1 |  |

6.2 SmartEyes的可行性

6.3我们的优点和不足

6.3.1 优点

TODOs

1. opencv函数名称 描述 参数列表 返回值
2. mxnet --------