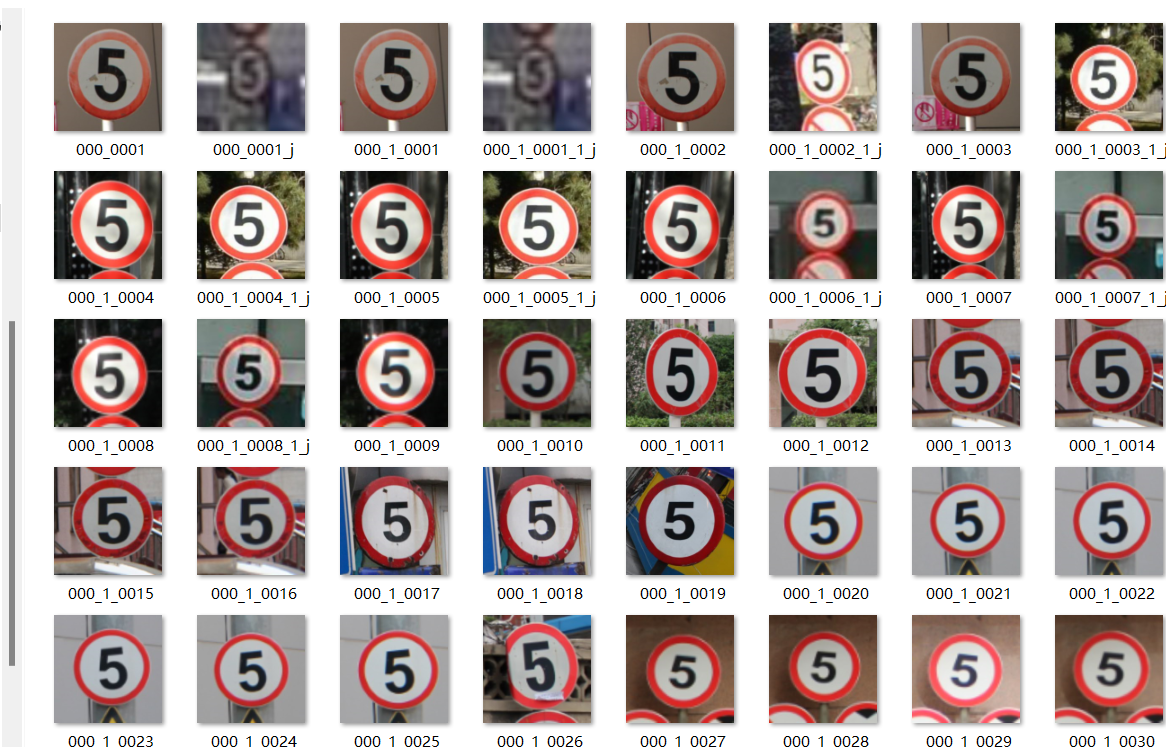
# 基于Python3+OpenCV的小车自动驾驶

**1 实验概括**

本实验基于python3+OpenCV对小车摄像头传入的视频流中的图像进行识别，再让舵机有相应的响应。在该实验中，我们需要在小车行驶的过程中实时识别小车行进轨迹中出现的信号标识牌，包括限速标识牌，转弯标识牌以及停止标识牌。我通过HOG+SVM完成本次实验的设计。

**2 数据获取**

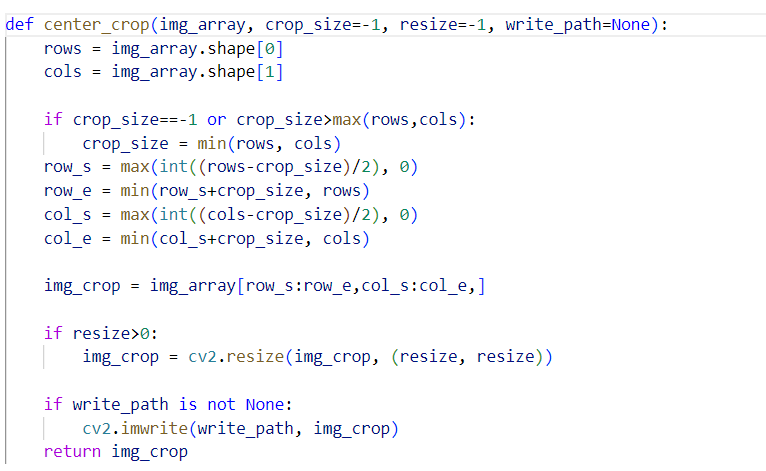
我通过python爬虫在网上下载了6000张路标标识符，总共有58种。

****

**3 图像预处理**

由于得到的数据集图像大小不一，所以需要首先从中心区域裁剪并调整正方形图像的大小，然后将处理后的图像保存到训练集和测试集里面。

**3.1 从中心区域裁剪并调整正方形图像的大小**



**3.2 根据测试集和训练集自动生成带有尺寸等信息的xml文件**



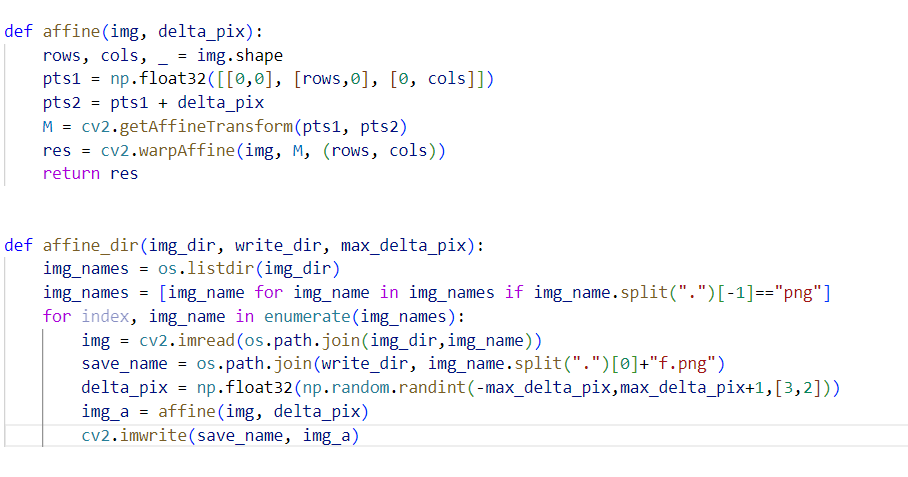
**3.3 对测试集和训练集里的图片进行标注**

**3.4 裁剪图片并分类**

对上述图片利用添加的矩形进行裁剪且重命名来分类。解析XML文档，根据<name>标签进行分类，如果是直行、右转、左转、停止，那么就把它从原图中裁剪下来并重命名,如果没有<object>那么就认为是负样本，其中在处理负样本的时候，我进行了颜色识别，把一张负样本图片根据颜色（红色、蓝色）裁剪成几张负样本，这样做的好处是：在进行交通标志的识别时，也是使用的颜色识别来选取到交通标志，从负样本中分割出来的相近颜色样本有利于负样本的训练，提高模型精度。

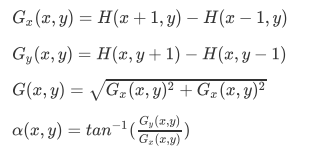


为了提高模型的精确度，还对目标图片进行仿射变换来扩充训练集。



**4 HOG特征提取**

首先，我将连续的视频转换为离散的图像，完成之后，我们采用gamma矫正法对输入图像进行颜色空间的标准化，调节图像对比度，抑制噪声的干扰。最终，我们总共提取R和B两个维度的灰度图。之后，我们将图像分割成大小为8\*8的cell，计算cell内每个像素的梯度，包含了方向和大小。梯度的计算以差值的形式进行：

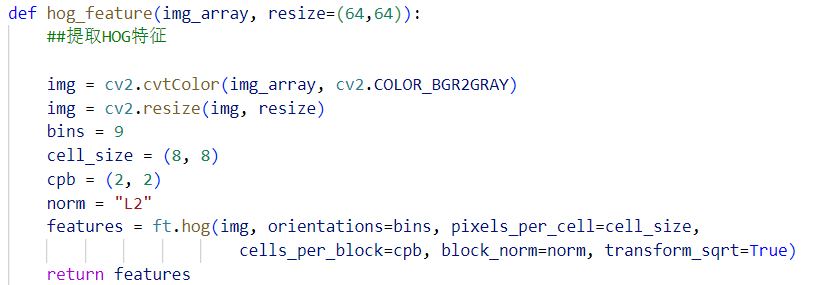


统计cell内各像素点的梯度形成直方图，作为该cell的描述子。考虑到在车辆移动的过程中，标识牌 的大小是在不断变化的，因此我选择对不同大小的rect区域都进行检测。



分别对训练集和测试集进行特征提取得到相应的txt文件。

**4.1 提取HOG特征**

****

**4.2提取目录中所有图像HOG特征**

****

**5 模型训练**

利用得到的HOG特征进行训练模型得到svm\_model.pkl。

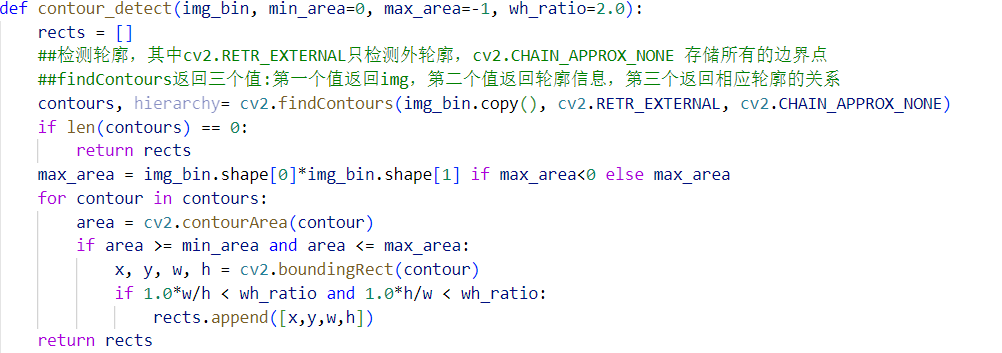
****

**6 电脑端实践测试**

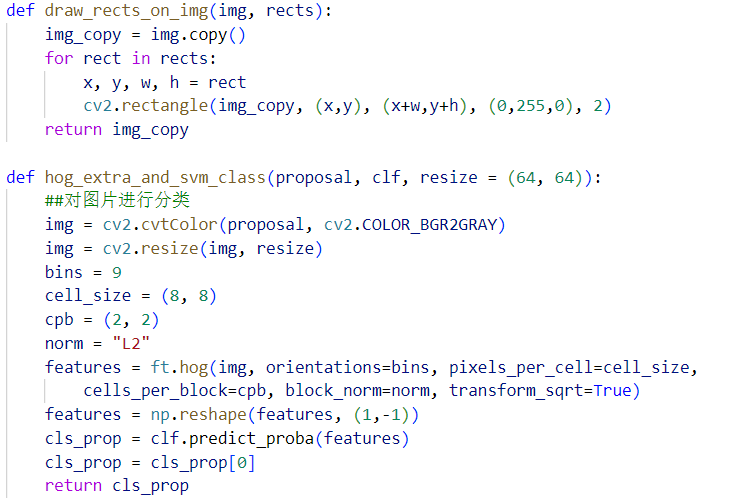
**6.1通过颜色阈值分割选出蓝色和红色对应的区域得到二值化图像。**

****

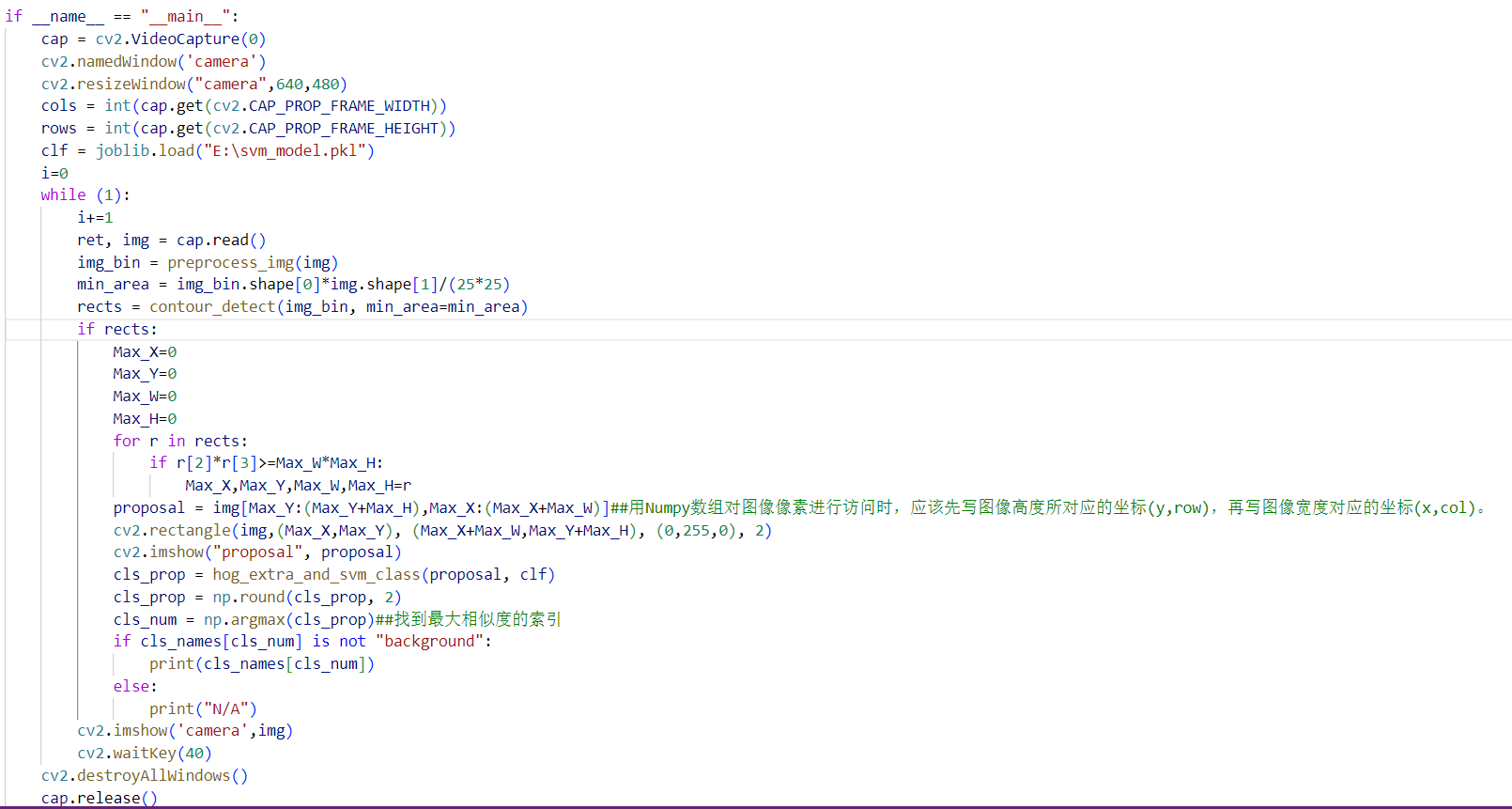
**6.2提取轮廓,返回轮廓矩形框**

****

**6.3 返回带有矩阵框的img**

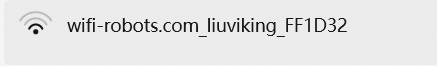
****

**6.4视频识别**

****

**7 小车指令**

**首先用wifi连接小车**

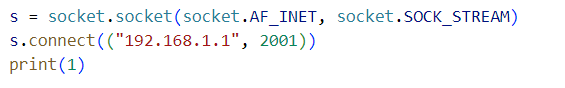


**7.1 小车指令**

数据位即可以来设置速度



**7.2 运行下图代码来获取视频流**



到此可以实现最基础的识别和运动，精度也不错但实际过程中我发现了很多问题。

**8 主要问题总结**

**1.视频流可能会出错传出null导致程序崩溃、**

**2.可能一次识别同一个路标好几次**

**3.有好几个小车的wifi名字一样，如果在较近的距离进行调试，电脑会自动连接wifi信号最强的那台小车。**

**4.小车的左右舵机存在个体差异，比如设定一样的速度实际上会有速度差从而导致不能直行。**

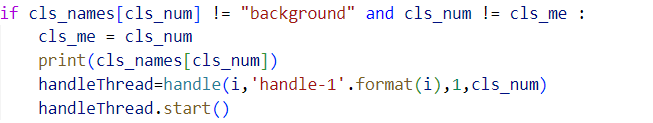
**9 问题解决**

**1.针对问题1：我使用了多线程的方式用thread来改善。**

核心代码：



**2.针对问题2:我使用内置的cls\_me即记忆来判断，如果这次识别的和上次不一样则算一次成功识别，也就是说同一个路标不能识别两次。**

****

**3.针对问题3：我只能选wifi最好的小车或者离得很远，这点很搞人心态，周五下午在这个没意义的问题上浪费了好多小时。**

**4.针对问题4：通过调试了解每个舵机具体情况，可以在robot\_control那个脚本中运行直行来调制，给左右舵机速度差。**

**10 总结与心得**

**首先很感激楼老师在实验过程中提供的各种帮助，耽误了您很多时间。其次通过这次实验我对模型训练，以及图像识别，再到小车的控制，通信协议的认识有了不一样的心得，收获到了很多。**

**我课下花了很多时间来做训练以及代码的改进，光是上周在东4调试就从早上九点干到晚上10点回宿舍。起初我的模型精度太低我甚至想放弃这个题目去做窃听，和楼老师讨论完后觉得自己不能这么容易放弃，就回去爬了大量数据，光是样本图片我就用了4个G的图像进行训练，借别人高性能电脑训练14个小时训练出了一个精度非常高的模型，也成功验收了，然而我没注意要体现变速，我的模型没有用不同路标，所以只能重新训练。第二次训练我没用那么多图像，从网上随便找了几张，大概花了3分钟就训练好了。令我吃惊的是这个模型精度也还是不错的，只是不能过滤一些干扰项图标，容易把红色的东西识别成路标，但是在纯净且较白的背景下还是可以达到实验目的的。**

**总而言之，我从零到有自己独立完成了整个项目还是很有成就感。**