\*

<https://lps-683.iteye.com/blog/2360806>

* [首页](https://www.csdn.net/)
* [博客](https://blog.csdn.net/)
* [学院](https://edu.csdn.net/)
* [下载](https://download.csdn.net/)
* [图文课](https://gitchat.csdn.net/?utm_source=csdn_toolbar)
* [论坛](https://bbs.csdn.net/)
* [APP](https://www.csdn.net/app/)
* [问答](https://ask.csdn.net/)
* [商城](https://mall.csdn.net/)
* [VIP会员](https://mall.csdn.net/vip_code)
* [活动](https://huiyi.csdn.net/)
* [招聘](http://job.csdn.net/)
* [ITeye](http://www.iteye.com/)
* [GitChat](https://gitbook.cn/?ref=csdn)
* 
* [写博客](https://mp.csdn.net/postedit)
* [[https://csdnimg.cn/public/common/toolbar/images/money.png](https://gitbook.cn/new/gitchat/activity?utm_source=csdnblog1)赚零钱](https://gitbook.cn/new/gitchat/activity?utm_source=csdnblog1)
* [[https://csdnimg.cn/public/common/toolbar/images/message-icon.png](https://i.csdn.net/#/msg/index)消息](https://i.csdn.net/#/msg/index)

[首页](https://www.iteye.com/) [资讯](https://www.iteye.com/news) [精华](https://www.iteye.com/magazines) [论坛](https://www.iteye.com/forums) [问答](https://www.iteye.com/ask) [博客](https://www.iteye.com/blogs) [专栏](https://www.iteye.com/blogs/subjects) [群组](https://www.iteye.com/groups)

[**您还未登录 !**](https://lps-683.iteye.com/login) [登录](https://lps-683.iteye.com/login)

`

[](https://lps-683.iteye.com/)

**lps\_683**

* 浏览: 209629 次
* 性别: Icon_minigender_1
* 来自: 上海
* https://lps-683.iteye.com/images/status/offline.gif

最近访客 [更多访客>>](https://lps-683.iteye.com/blog/user_visits)

[makemyownlife的博客](https://makemyownlife.iteye.com/)

[makemyownlife](https://makemyownlife.iteye.com/)

[xtu502的博客](https://xtu502.iteye.com/)

[xtu502](https://xtu502.iteye.com/)

[之一先生的博客](https://mrzhiyi.iteye.com/)

[之一先生](https://mrzhiyi.iteye.com/)

[soulhez的博客](https://soulhez.iteye.com/)

[soulhez](https://soulhez.iteye.com/)

博主相关

 [博客](https://lps-683.iteye.com/)

 [微博](https://lps-683.iteye.com/weibo)

 [相册](https://lps-683.iteye.com/album)

 [收藏](https://lps-683.iteye.com/link)

 [留言](https://lps-683.iteye.com/blog/guest_book)

 [关于我](https://lps-683.iteye.com/blog/profile)

文章分类

* [全部博客 (59)](https://lps-683.iteye.com/)
* [java (9)](https://lps-683.iteye.com/category/284823)
* [C++ (23)](https://lps-683.iteye.com/category/347812)
* [openCV (24)](https://lps-683.iteye.com/category/347858)
* [C++，算法 (1)](https://lps-683.iteye.com/category/349150)
* [QT (4)](https://lps-683.iteye.com/category/349322)
* [机器学习 (11)](https://lps-683.iteye.com/category/350412)
* [android (2)](https://lps-683.iteye.com/category/350521)
* [心得 (0)](https://lps-683.iteye.com/category/350522)
* [C++，opencvforandroid (0)](https://lps-683.iteye.com/category/352720)
* [opencvforandroid (2)](https://lps-683.iteye.com/category/352721)
* [毕业设计 (9)](https://lps-683.iteye.com/category/354058)
* [心情 (0)](https://lps-683.iteye.com/category/363914)
* [openCV，图像处理 (4)](https://lps-683.iteye.com/category/363979)
* [车道线检测 (2)](https://lps-683.iteye.com/category/364500)
* [openCV，图像处理，曲线识别 (1)](https://lps-683.iteye.com/category/364540)
* [机器学习，python (2)](https://lps-683.iteye.com/category/365176)
* [机器学习，python，数据挖掘 (2)](https://lps-683.iteye.com/category/365178)
* [算法 (3)](https://lps-683.iteye.com/category/365343)
* [MXNet (1)](https://lps-683.iteye.com/category/365595)
* [Tensorflow (0)](https://lps-683.iteye.com/category/365635)
* [flask (0)](https://lps-683.iteye.com/category/367304)
* [JAVAEE (1)](https://lps-683.iteye.com/category/370115)
* [python (1)](https://lps-683.iteye.com/category/375324)
* [趣味问题 (1)](https://lps-683.iteye.com/category/375325)

社区版块

* [我的资讯](https://lps-683.iteye.com/blog/news) ( 0)
* [我的论坛](https://lps-683.iteye.com/blog/post) ( 0)
* [我的问答](https://lps-683.iteye.com/blog/answered_problems) ( 0)

存档分类

* [2017-09](https://lps-683.iteye.com/blog/monthblog/2017-09) ( 1)
* [2017-07](https://lps-683.iteye.com/blog/monthblog/2017-07) ( 1)
* [2017-04](https://lps-683.iteye.com/blog/monthblog/2017-04) ( 1)
* [更多存档...](https://lps-683.iteye.com/blog/monthblog_more)

最新评论

* [liweicong](https://liweicong.iteye.com/)： 这个识别的准确率是多少呢？训练的那个circle和四个参数代表 ...  
  [交通标牌检测与识别](https://lps-683.iteye.com/blog/2360806#bc2402393)
* [13227819390](https://13227819390.iteye.com/)： 请问在博主的percolation代码块里的isFull函数， ...  
  [并查集算法关于渗透模型中虚拟节点的使用问题](https://lps-683.iteye.com/blog/2354888#bc2401138)
* [zzzimooo](https://zzzimooo.iteye.com/)： 请问博主，大概在130行：Bitmap bmp = getRe ...  
  [OpenCVforAndroid应用之银行卡号识别 ------ 实战篇](https://lps-683.iteye.com/blog/2273467#bc2393132)
* [u010074054](https://u010074054.iteye.com/)： 您好，我想请教一下关于银行卡识别的问题，您方便加一下我QQ吗？ ...  
  [OpenCVforAndroid应用之银行卡号识别 ------ 实战篇](https://lps-683.iteye.com/blog/2273467#bc2386735)
* [javams](https://javams.iteye.com/)： 闵大荒是什么？  
  [闵大荒之旅（一）](https://lps-683.iteye.com/blog/2279906#bc2385763)

[交通标牌检测与识别](https://lps-683.iteye.com/blog/2360806)

**博客分类：**

* [C++，算法](https://lps-683.iteye.com/category/349150)
* [openCV](https://lps-683.iteye.com/category/347858)

阅读更多

关于交通标牌检测的博客和论文非常多，例如，本人最近在博客上就看到有一篇简单的交通标志检测与识别介绍文章《[自动驾驶之眼——摄像头是如何认识交通标志的？](http://m.leiphone.com/news/201703/E8UOnNvzhmsGzHTf.html)》，该文很简洁明了地阐述了交通标志检测识别的主要流程。本文将结合上学期的课程设计，来整理一下交通标识牌检测与识别的思路与实现方法。

**一、要求**

首先要明确一下本文到底是要干什么。本文要完成基于视觉的交通标识牌检测与识别，说白了，就两个事：1）在一张图中找到交通标识牌在哪里（检测）；2）认清楚这个标识牌是啥，表达的什么意思（识别）。那么最后得到的结果预览如下：  
  
**二、使用数据**

交通标识牌种类数不胜数，我国的交通标志一共有一百余种，按类别可分为黄底黑边的警告标志、白底红圈的禁令标志、蓝底白字的指示标志，形状上以三角形、圆形和矩形为主。本文主要是为了介绍一下交通标识牌的识别流程和一些主要方法的实现，为了简化工作，本文挑选了以下五类交通标识牌。

Second, the use of data

There are countless types of traffic signs. There are more than one hundred kinds of traffic signs in China. According to the categories, they can be divided into warning signs on the black side of the yellow base, ban marks on the red circle on the white background, and signs on the white background in blue. The shape is triangular. The circle and the rectangle are dominant. This paper is mainly to introduce the identification process of traffic signs and the realization of some main methods. In order to simplify the work, this paper selects the following five types of traffic signs.  
可以看出来，博主用心良苦，选择的交通标识牌具有很清楚的特征：1）颜色上，这五类交通标识牌的外边框都是红色的；2）形状上，标识牌都是标准的圆形。这事实上也表明了，交通标识牌具有着鲜明的特征，故无论是人眼还是机器，都较易识别。(其他种类的交通标志牌也是一样，利用形状和颜色特征来处理)

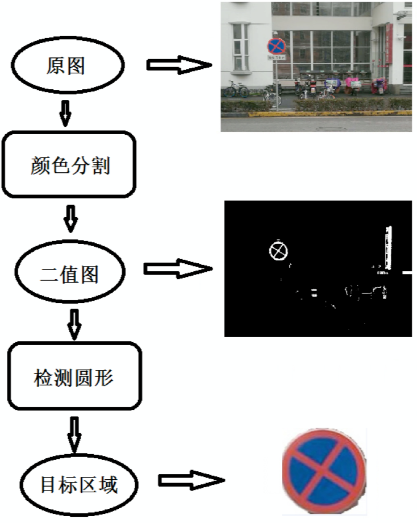
It can be seen that the bloggers have a good intention, and the selected traffic signs have very clear characteristics: 1) in color, the outer borders of the five types of traffic signs are all red; 2) the shape, the signs are standard Round shape. This in fact also shows that the traffic sign has a distinctive feature, so it is easier to identify whether it is the human eye or the machine. (Other types of traffic signs are the same, using shape and color features to deal with)

**三、使用方法 (Third, the method of use)**  
 在我看来，目前处理交通标牌识别的主要有两种方法，1）传统的图像处理+机器学习办法；2）最近很火的深度学习。那么本文采用的是前者，后者后续再进行介绍。

In my opinion, there are currently two main methods for dealing with traffic signage identification: 1) traditional image processing + machine learning methods; 2) recent intense learning. Then the paper is based on the former, and the latter is introduced later.

**1>检测：颜色和形状。(1> Detection: color and shape.)**

交通标志牌为了起到其警示作用，在颜色和形状上都有着易区分性，如本文所讨论的五类标志牌，颜色特征为外框均为鲜艳的红色；形状特征为均为圆形。于是，检测的思路如下,最终得到了圆形部分的交通标牌：

(In order to play its warning role, the traffic signs are easy to distinguish between color and shape. For the five types of signs discussed in this article, the color features are bright red in the outer frame; the shape features are all round. Therefore, the idea of detection is as follows, and finally a circular traffic sign is obtained)  
接下来，主要分为颜色分割和形状检测**两部分**进行讨论：(Next, it is divided into two parts: color segmentation and shape detection.)

**基于颜色分割的图像二值化处理：(Image binarization based on color segmentation:)**

最直观、简单的是利用RGB颜色空间来描述图像的色彩情况，但是，RGB色彩空间极易受到光线情况的影响，鲁棒性并不是很好，所以在相关论文中，你会发现，很少有人直接使用RGB色彩空间进行色彩分割。而实际上，本人拿有限的样本和测试集进行测试，RGB色彩分割效果在图像成像质量较理想的时候效果极佳，但是的确容易受到干扰。本文此处选择了HSI色彩空间模型进行色彩分割。先来点理论知识：

The most intuitive and simple is to use RGB color space to describe the color of the image. However, the RGB color space is very susceptible to light conditions, and the robustness is not very good, so in related papers, you will find that very few Someone directly uses the RGB color space for color segmentation. In fact, I used a limited sample and test set to test, RGB color segmentation effect is excellent when the image quality is ideal, but it is easy to be interfered. In this paper, the HSI color space model is selected for color segmentation. First come to the theoretical knowledge:

    色调H（Hue）：与光波的波长有关，它表示人的感官对不同颜色的感受，如红色、绿色、蓝色等，它也可表示一定范围的颜色，如暖色、冷色等。

Hue (Hue): related to the wavelength of the light wave, it represents the human sense of the different colors, such as red, green, blue, etc., it can also represent a range of colors, such as warm, cool and so on.

    饱和度S（Saturation）：表示颜色的纯度，纯光谱色是完全饱和的，加入白光会稀释饱和度。饱和度越大，颜色看起来就会越鲜艳，反之亦然。

Saturation: Indicates the purity of the color. The pure spectral color is fully saturated. Adding white light will dilute the saturation. The greater the saturation, the brighter the color will look and vice versa.

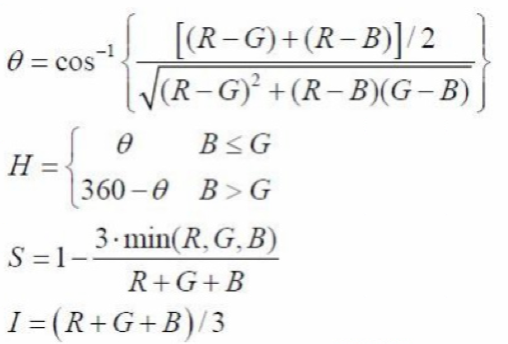
    亮度I（Intensity）：对应成像亮度和图像灰度，是颜色的明亮程度。

从理论上看，HSI色彩空间将饱和度和亮度信息独立了出来，这样一定程度上就降低了光线带来的影响。听上去很有道理，但是，实际上，这也仅仅是一定程度上降低了亮度和色彩的耦合关系，并不是完全地进行了解耦，所以，效果会有提升，但是很难带来质的改变（*这是笔者自己的体验，也许是笔者能力不足，实现得不是很理想*）

那么从RGB色彩空间转换到HSI空间的转换公式如下：

Intensity: Corresponds to the brightness of the image and the gray level of the image, which is the brightness of the color.

In theory, the HSI color space separates the saturation and brightness information, which reduces the impact of light to a certain extent. It sounds very reasonable, but, in fact, this only reduces the coupling relationship between brightness and color to a certain extent. It is not completely decoupled. Therefore, the effect will be improved, but it is difficult to bring quality. Change (this is the author's own experience, perhaps the author's lack of ability, the implementation is not very satisfactory)

Then the conversion formula from RGB color space to HSI space is as follows:  


函数RGB2HSI是将RGB色彩空间转换到HSI色彩空间，其转换的过程参照式(2.2)，最后将饱和度S和强度I均放大100倍，便于操作。 最后得到的H、 S、 I的取值范围分别为[0,360]、 [0,100]、 [0,100]。

 The function RGB2HSI converts the RGB color space into the HSI color space. The conversion process refers to equation (2.2), and finally the saturation S and intensity I are both magnified 100 times for easy operation. The final values of H, S, and I are [0,360], [0,100], [0,100].

**Cpp代码  [收藏代码](javascript:void()) Cpp code**

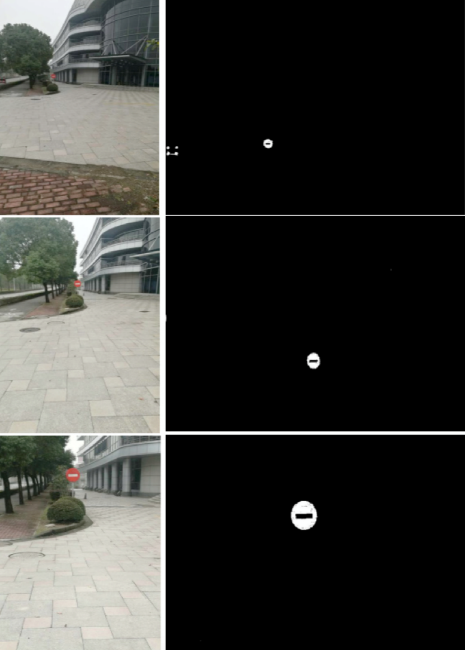
1. **void** RGB2HSV(**double** red, **double** green, **double** blue, **double**& hue, **double**& saturation,
2. **double**& intensity )
3. {
4. **double** r,g,b;
5. **double** h,s,i;
6. **double** sum;
7. **double** minRGB,maxRGB;
8. **double** theta;
9. r = red/255.0;
10. g = green/255.0;
11. b = blue/255.0;
12. minRGB = ((r<g)?(r):(g));
13. minRGB = (minRGB<b)?(minRGB):(b);
14. maxRGB = ((r>g)?(r):(g));
15. maxRGB = (maxRGB>b)?(maxRGB):(b);
16. sum = r+g+b;
17. i = sum/3.0;
18. **if**( i<0.001 || maxRGB-minRGB<0.001 )
19. {
20. h=0.0;
21. s=0.0;
22. }
23. **else**
24. {
25. s = 1.0-3.0\*minRGB/sum;
26. theta = sqrt((r-g)\*(r-g)+(r-b)\*(g-b));
27. theta = acos((r-g+r-b)\*0.5/theta);
28. **if**(b<=g)
29. h = theta;
30. **else**
31. h = 2\*PI - theta;
32. **if**(s<=0.01)
33. h=0;
34. }
35. hue = (**int**)(h\*180/PI);
36. saturation = (**int**)(s\*100);
37. intensity = (**int**)(i\*100);
38. }

在得到HSI空间的基础上，分割出红色像素，事实上这个阈值最好时自己调出来，无论是基于哪个色彩空间，网上的代码或者论文中的数值都是个参考，自己调出来的才靠谱嘛，代码如下：

On the basis of the HSI space, the red pixel is segmented. In fact, when the threshold is the best, it is called up. No matter which color space is used, the code in the online code or the paper is a reference. Well, the code is as follows:

**Cpp代码  [收藏代码](javascript:void())**

1. //得到图像参数
2. **int** width = src.cols; //图像宽度
3. **int** height = src.rows; //图像高度
4. //色彩分割
5. **double** B=0.0,G=0.0,R=0.0,H=0.0,S=0.0,I=0.0;
6. Mat Mat\_rgb = Mat::zeros( src.size(), CV\_8UC1 );
7. **int** x,y,px,py; //循环
8. **for** (y=0; y<height; y++)
9. {
10. **for** ( x=0; x<width; x++)
11. {
12. // 获取 BGR 值
13. B = src.at<Vec3b>(y,x)[0];
14. G = src.at<Vec3b>(y,x)[1];
15. R = src.at<Vec3b>(y,x)[2];
16. RGB2HSV(R,G,B,H,S,I);
17. //红色：337-360
18. **if**((H>=337 && H<=360||H>=0&&H<=10)&&
19. S>=12&&S<=100&&V>20&&V<99)
20. {
21. Mat\_rgb.at<uchar>(y,x) = 255; //分割出红色
22. }
23. }
24. }

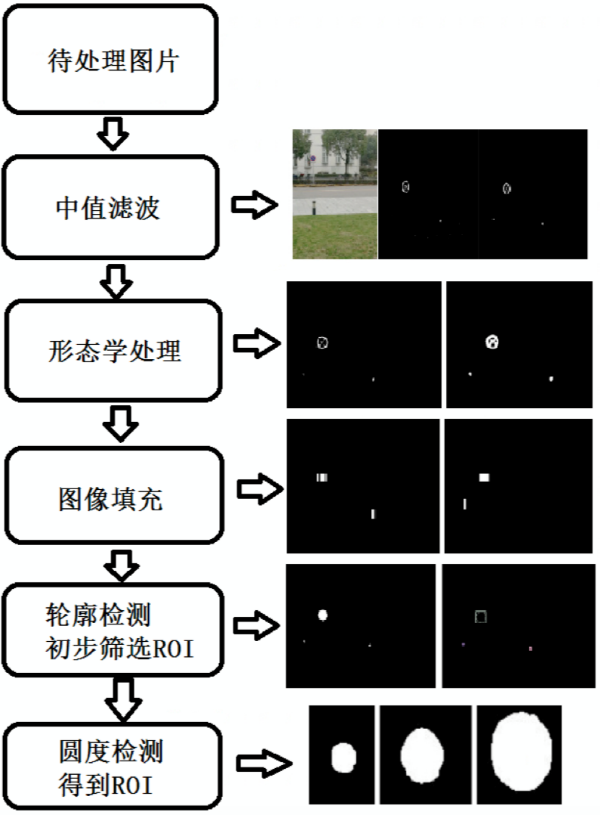
分割效果可见如下组图，由近至远： The segmentation effect can be seen as follows: from near to far:  
⚠️⚠️⚠️注意：有一个很严肃的问题我这里没有提，那就是图像预处理！做图像处理的很重要的一个步骤就是图像预处理，预处理做好了，后面的问题复杂度也就降低了许多。实际上，用颜色分割来二值化图像也可以看作一种预处理。那么颜色分割之前有不有必要做图像预处理呢？是有的。举个例子，我那我的MATE8在学校里拍了一张照，然后使用手机相机自带的功能，**调整其色彩饱和度，亮度等**，得到以下两种图片：Note: There is a very serious problem I have not mentioned here, that is image preprocessing! A very important step in image processing is image preprocessing. Preprocessing is done, and the complexity of the latter problem is much reduced. In fact, binarizing images with color segmentation can also be seen as a pre-processing. So there is no need to do image pre-processing before color segmentation? There is. For example, my MATE8 took a picture at school, and then used the functions that came with the phone camera to adjust its color saturation, brightness, etc., to get the following two pictures:  
          
                                  相机拍的原图                                                  手机调整饱和度、亮度后

这两种图片，显然右边的将更有利于颜色分割！（不信可以试试哦）。本文主要以介绍交通标牌的主要流程为主，预处理的方法包括直方图均衡化、白平衡、亮度调节等等这些就不仔细纠结了，但是，不代表这部分不重要，**图像预处理往往一定程度上决定了最后的效果。**

 The original picture taken by the camera. After adjusting the saturation and brightness of the phone

These two pictures, obviously on the right side will be more conducive to color segmentation! (Do not believe you can try it). This paper mainly introduces the main flow of traffic signs. The preprocessing methods include histogram equalization, white balance, brightness adjustment, etc. These are not carefully entangled, but it does not mean that this part is not important, image preprocessing is often certain. The extent determines the final effect.

**基于形状（圆形）检测的ROI提取 ROI extraction based on shape (circular) detection**  
 在进行颜色分割之后，得到的只是一个粗略的交通标志牌ROI区域， 还会留下一些噪声以及一些和目标区域面积相当或者比目标面积略大的区域，这时候就还需要进行一些图像预处理，为准确检测交通标志牌打下坚实基础。由于交通标志最明显的特征是其颜色和形状，在用颜色分割之后，我们可以通过形状特征来去除其余的干扰。对于本文的研究对象而言，交通标志牌的形状为圆形，可以采用经典的Hough变换进行圆检测，该方法准确性高，但是计算量大，耗时且占用较大内存；也可以采用圆度的方法来提取圆形，该方法原理简单，计算量小，准确率高。综合考虑，本文使用基于圆度的圆检测算法。大概流程如下，后文还会详细介绍：

After the color segmentation, only a rough traffic sign ROI area is obtained, and some noise and some areas corresponding to the target area or slightly larger than the target area are left. At this time, some image preprocessing is required. , to lay a solid foundation for the accurate detection of traffic signs. Since the most obvious feature of traffic signs is their color and shape, after color segmentation, we can remove the remaining interference by shape features. For the research object of this paper, the shape of the traffic sign is circular, and the classic Hough transform can be used for circle detection. The method has high accuracy, but the calculation amount is large, it takes time and takes up large memory; The method of degree extraction to round, the method is simple in principle, small in calculation and high in accuracy. Considering this, this paper uses a circle-based circle detection algorithm. The approximate process is as follows, and will be described in detail later:  


图有点不太清楚，下文中对于关键的部分会再次给出效果图。The diagram is a bit unclear, and the renderings will be given again for the key parts below.

**中值滤波**，这个没啥好说的，图上效果不是很明显，但是实际上可以一定程度上滤掉单个噪点，对得到准确的结果会有一定的帮助；

**Median filtering**, this is nothing to say, the effect on the graph is not very obvious, but in fact it can filter out a single noise to some extent, it will help to get accurate results;

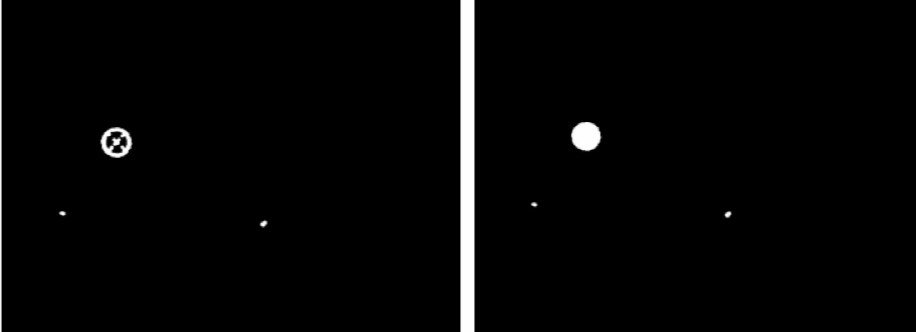
**形态学处理**，最后我们的目的是要得到一个封闭的区域，所以，颜色分割后的结果很可能不会是比较理想封闭的圆形，那么选用的3×3腐蚀模板，7×7膨胀模板，这样检测到的圆形将基本不会产生缺口，保证是一个封闭的形状。  
  
**Morphological processing**, in the end our goal is to get a closed area, so the result of color segmentation is probably not the ideal closed circle, then the 3 × 3 corrosion template, 7 × 7 expansion template, The circle thus detected will have substantially no gaps and is guaranteed to be a closed shape.

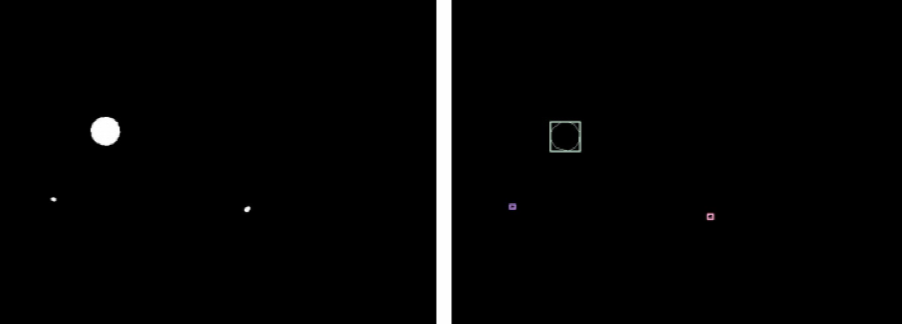
**图像填充**，有了上述步骤得到的封闭圆形，我们接下来就可以填充封闭图形了（这里你可能会问，为啥要这样做。实际上直接进行Hough圆检测可以得到ROI结果，但是本文是换了一个思路，使用圆度来判断圆形，所以算法需要一个实心区域），代码如下：

**Image fill,** with the closed circle obtained in the above steps, we can then fill the closed graph (here you may ask, why you should do this. In fact, directly performing Hough circle detection can get ROI results, but this article is Changed the idea, using roundness to judge the circle, so the algorithm needs a solid area), the code is as follows

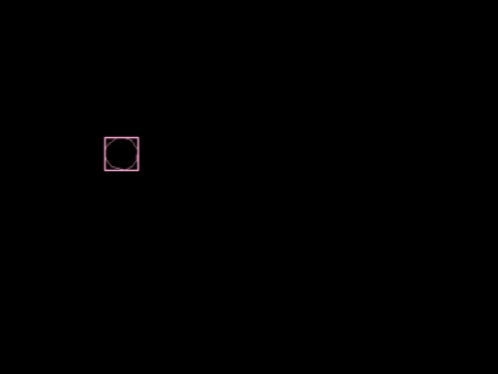
**Cpp代码  [收藏代码](javascript:void())**

1. **void** fillHole(**const** Mat srcBw, Mat &dstBw)
2. {
3. Size m\_Size = srcBw.size();
4. Mat Temp=Mat::zeros(m\_Size.height+2,m\_Size.width+2,srcBw.type());//延展图像
5. srcBw.copyTo(Temp(Range(1, m\_Size.height + 1), Range(1, m\_Size.width + 1)));
6. cv::floodFill(Temp, Point(0, 0), Scalar(255));//填充区域
7. Mat cutImg;//裁剪延展的图像
8. Temp(Range(1, m\_Size.height + 1), Range(1, m\_Size.width + 1)).copyTo(cutImg);
9. dstBw = srcBw | (~cutImg);
10. }

  
 **轮廓检测，初步筛选ROI**，要想使用基于圆度的圆检测算法，则需要从图像中提取初步的ROI来进行筛选。这里使用轮廓检测法来检测图片中的ROI区域。可以看到，一些细小的噪声也被检测进来。

**Contour detection, preliminary screening of ROI**, in order to use the circle-based circle detection algorithm, you need to extract the preliminary ROI from the image for screening. Here, contour detection is used to detect the ROI area in the picture. It can be seen that some small noise is also detected.  


所以，本文先通过对检测区域的宽高比、面积大小进行限制，筛选出有效的检测区域，经过实验，可以确定宽高比限制在0.5-2之间，面积最小值设定为400，可以进一步得到下图的检测效果，可以看到，此时小面积的噪声已经被排除。

Therefore, this paper firstly limits the aspect ratio and area of the detection area, and selects an effective detection area. After the experiment, it can be determined that the aspect ratio is limited to 0.5-2, and the area minimum is set to 400. Further, the detection effect of the following figure is obtained, and it can be seen that the small area noise has been eliminated at this time.  
代码如下：code show as below:

**Cpp代码  [收藏代码](javascript:void())**

1. //找轮廓
2. vector<vector<Point> > contours;
3. vector<Vec4i> hierarchy;
4. findContours( Mat\_rgb, contours, hierarchy, CV\_RETR\_EXTERNAL,
5. CV\_CHAIN\_APPROX\_SIMPLE, Point(0, 0) );
6. /// 多边形逼近轮廓 + 获取矩形和圆形边界框
7. vector<vector<Point> > contours\_poly( contours.size() );
8. vector<Rect> boundRect( contours.size() );
9. vector<Point2f>center( contours.size() );
10. vector<**float**>radius( contours.size() );
11. //得到轮廓矩形框
12. **for**( **int** i = 0; i < contours.size(); i++ )
13. {
14. approxPolyDP( Mat(contours[i]), contours\_poly[i], 3, **true** );
15. boundRect[i] = boundingRect( Mat(contours\_poly[i]) );
16. minEnclosingCircle( contours\_poly[i], center[i], radius[i] );
17. }
18. /// 画多边形轮廓 + 包围的矩形框
19. Mat drawing = Mat::zeros( Mat\_rgb.size(), CV\_8UC3 );
20. **for**( **int** i = 0; i< contours.size(); i++ )
21. {
22. Rect rect = boundRect[i];
23. //首先进行一定的限制，筛选出区域
24. //高宽比限制
25. **float** ratio = (**float**)rect.width / (**float**)rect.height;
26. //轮廓面积
27. **float** Area = (**float**)rect.width \* (**float**)rect.height;
28. **float** dConArea = (**float**)contourArea(contours[i]);
29. **float** dConLen = (**float**)arcLength(contours[i],1);
30. **if**(dConArea <400)//ROI 区域面积限制
31. **continue**;
32. **if**(ratio>2||ratio<0.5)//ROI 区域宽高比限制
33. **continue**;
34. //检测到了！
35. Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
36. //绘制轮廓和检测到的轮廓外接矩形
37. drawContours( drawing, contours\_poly, i, color, 1, 8, vector<Vec4i>(), 0, Point() );
38. rectangle( drawing, boundRect[i].tl(), boundRect[i].br(), color, 2, 8, 0 );
39. rectangle( src, boundRect[i].tl(), boundRect[i].br(), color, 2, 8, 0 );
40. }

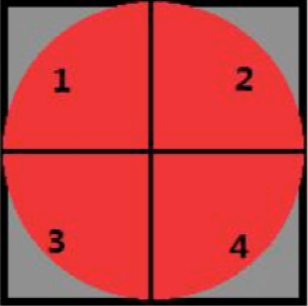
**圆度算法检测**，实际上这是利用了非常简单的数学约束，来对检测到的区域进行圆形验证。圆度定义如下：

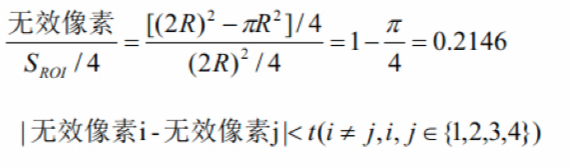
**Roundness algorithm detection**, in fact, utilizes very simple mathematical constraints to perform circular verification of the detected area. The roundness is defined as follows:  
http://dl2.iteye.com/upload/attachment/0123/5912/59f78a3d-0ae4-389d-8dcb-c85be243de6b.png

其中，S为圆的面积，L为圆的周长，C为圆度。圆度值越接近1，则表示该图形与圆形的契合程度越高。经过大量的实验，可以得出圆度大于0.5时，即 4 . 0  C 时，可以筛选出巨大部分的圆形。

Where S is the area of the circle, L is the circumference of the circle, and C is the roundness. The closer the roundness value is to 1, the higher the degree of fit of the graphic to the circle. After a lot of experiments, it can be concluded that when the roundness is greater than 0.5, that is, 4.0  C, a large part of the circle can be screened.

**ROI区域无效像素面积约束**，这是进一步确定筛选后的ROI区域是目标圆形区域。该约束条件是基于ROI区域中圆形的缺失面积而得到的。前文中得到的ROI区域是包含圆形交通标志牌的矩形区域，如下图所示，可以将整个ROI区域分成1、2、3、4四块，其中红色部分为交通标志牌，灰色部分为ROI区域中的无效像素。可以直观地看到，1、2、3、4四块的无效像素满足一定的数学关系，

The ROI area has an invalid pixel area constraint, which is to further determine that the filtered ROI area is the target circular area. This constraint is based on the missing area of the circle in the ROI region. The ROI area obtained in the foregoing is a rectangular area containing a circular traffic sign. As shown in the following figure, the entire ROI area can be divided into four blocks of 1, 2, 3, and 4, wherein the red part is a traffic sign and the gray part is a ROI. Invalid pixels in the area. It can be intuitively seen that the invalid pixels of 1, 2, 3, and 4 blocks satisfy a certain mathematical relationship.  


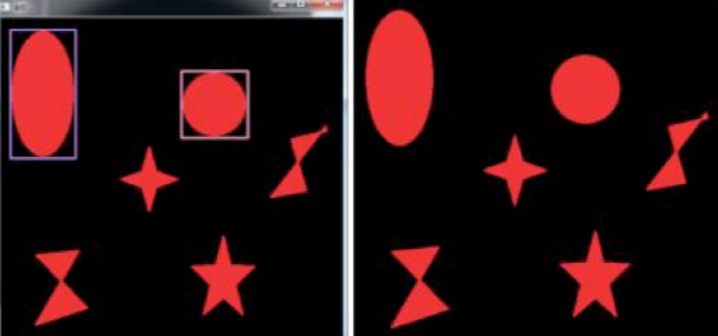
有如下约束：There are the following constraints:  


圆度代码即为一个约束条件，对面个轮廓检测得到的ROI进行验证，无效像素面积约束则代码如下：The roundness code is a constraint, and the ROI obtained by the contour detection is verified. The invalid pixel area constraint is as follows:

**Cpp代码  [收藏代码](javascript:void())**

1. **bool** isCircle(**const** Mat srcBw,Mat& mytemp)//（待改进）
2. {//输入的是一个灰度图像
3. Mat temp = Mat::zeros( srcBw.size(), CV\_8UC1 );;
4. **bool** iscircle = **false**;
5. //获得 srcBw 信息
6. **int** w = srcBw.cols;
7. **int** h = srcBw.rows;
8. **int** count1=0;//各部分的缺失像素计数器
9. **int** count2=0;
10. **int** count3=0;
11. **int** count4=0;
12. //将 srcBw 平均分成四份,进行访问缺失的像素个数、所占比重
13. //先访问左上
14. **for**(**int** i = 0;i < h/2;i ++)
15. {
16. **for**(**int** j = 0;j < w/2;j ++)
17. {
18. **if**(srcBw.at<uchar>(i,j) == 0)
19. {
20. temp.at<uchar>(i,j) = 255;
21. mytemp.at<uchar>(i,j) = 255;
22. count1 ++;
23. }
24. }
25. }
26. //右上
27. **for**(**int** i = 0;i < h/2;i ++)
28. {
29. **for**(**int** j = w/2-1;j < w;j ++)
30. {
31. **if**(srcBw.at<uchar>(i,j) == 0)
32. {
33. temp.at<uchar>(i,j) = 255;
34. mytemp.at<uchar>(i,j) = 255;
35. count2 ++;
36. }
37. }
38. }
39. //左下
40. **for**(**int** i = h/2-1;i < h;i ++)
41. {
42. **for**(**int** j = 0;j < w/2;j ++)
43. {
44. **if**(srcBw.at<uchar>(i,j) == 0)
45. {
46. temp.at<uchar>(i,j) = 255;
47. mytemp.at<uchar>(i,j) = 255;
48. count3 ++;
49. }
50. }
51. }
52. //右下
53. **for**(**int** i = h/2-1;i < h;i ++)
54. {
55. **for**(**int** j = w/2-1;j < w;j ++)
56. {
57. **if**(srcBw.at<uchar>(i,j) == 0)
58. {
59. temp.at<uchar>(i,j) = 255;
60. mytemp.at<uchar>(i,j) = 255;
61. count4 ++;
62. }
63. }
64. }
65. **float** c1 = (**float**)count1/(**float**)(w\*h);//左上
66. **float** c2 = (**float**)count2/(**float**)(w\*h);//右上
67. **float** c3 = (**float**)count3/(**float**)(w\*h);//左下
68. **float** c4 = (**float**)count4/(**float**)(w\*h);//右下
69. cout << "result: " << c1 << "," << c2
70. << "," << c3 << "," << c4 << endl;
71. //限定每个比率的差值范围
72. **if**((c1>0.037&&c1<0.12)&&(c2>0.037&&c2<0.12)&&(c2>0.037&&c2<0.12)&&(c2>0.037
73. &&c2<0.12))
74. {
75. //限制差值,差值比较容错，相邻块之间差值相近，如左上=右上&&左下=右下或左上=左下&&右上=右下
76. **if**((abs(c1-c2)<0.04&&abs(c3-c4)<0.04)||(abs(c1-c3)<0.04&&abs(c2-c4)<0.04))
77. {
78. iscircle = **true**;
79. }
80. }
81. **return** iscircle;
82. }

利用轮廓检测、圆度约束和无效面积约束，可以测试得到如下效果图，

Using the contour detection, the roundness constraint and the invalid area constraint, the following effect diagram can be tested.  
  
 最后，目标区域提取的效果如下：Finally, the effect of the target area extraction is as follows:  
  
 **2>识别：SVM分类。2> Identification: SVM classification.**

有了上文提取ROI的基础，分类过程实际上和我之前写过的[箭头分类](https://lps-683.iteye.com/blog/2343613)如出一辙。

**图像预处理**，首先将无效像素全部去除，只留下圆形ROI有效区域，

With the foundation for extracting ROI above, the classification process is actually the same as the arrow classification I wrote before.

Image preprocessing, first remove all invalid pixels, leaving only the circular ROI effective area,  


然后进行二值化处理，二值化后的图像特征更为清晰

Then binarization is performed, and the image features after binarization are clearer.  
可以选择所有像素作为特征，当然更科学的是Hu不变矩、Zernike不变矩、二者混合矩等特征。关于[Hu、Zernike特征](http://download.csdn.net/detail/leechi1010/9260601)的代码网上比比皆是，这里仅推荐一个作为参考。本文为了简单实现框架，拿全部像素特征进行训练。准备好样本和测试集，并给这五类交通标牌设置标签“stop”，“20t”，“car forbidden”，“5”，“stop2”

SVM代码框架如下。All pixels can be selected as features, of course, the more scientific are Hu invariant moments, Zernike invariant moments, and the mixed moments of the two. Codes about Hu and Zernike features are everywhere, and only one is recommended here. In order to simply implement the framework, this paper uses all pixel features for training. Prepare samples and test sets, and set labels for these five types of traffic signs "stop", "20t", "car forbidden", "5", "stop2"

The SVM code framework is as follows.

SVM训练 SVM training

**Cpp代码  [收藏代码](javascript:void())**

1. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SVM 训练部分\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. //准备开始训练
3. CvSVM classifier;
4. CvSVMParams SVM\_params;
5. SVM\_params.kernel\_type = CvSVM::LINEAR; //使用 RBF 分类非线性问题
6. SVM\_params.svm\_type = CvSVM::C\_SVC;
7. SVM\_params.degree = 0;
8. SVM\_params.gamma = 0.01;
9. SVM\_params.term\_crit = cvTermCriteria(CV\_TERMCRIT\_ITER, 1000, FLT\_EPSILON);
10. SVM\_params.C = 100;
11. SVM\_params.coef0 = 0;
12. SVM\_params.nu = 0;
13. SVM\_params.p = 0.005;
14. classifier.train(train,labels ,Mat(),Mat(),SVM\_params); //SVM 训练，线性核上述参数 C 起作用

SVM保存 SVM save

**Cpp代码  [收藏代码](javascript:void())**

1. classifier.save("model180.txt");

SVM读取 SVM read

**Cpp代码  [收藏代码](javascript:void())**

1. //这里载入分类器，方便直接训练
2. CvSVM classifier;
3. classifier.load("model180.txt");

SVM预测 SVM prediction

**Cpp代码  [收藏代码](javascript:void())**

1. **for** (**int** i = 0;i < testdata.size() ; ++i)
2. {
3. **int** result = (**int**)classifier.predict(testdata[i]);
4. std::cout<<"测试样本"<<i+1<<"的测试结果为："
5. <<result<< " " << labelname[result-1] << "\n";
6. }

 最后得到了文中开始展示的效果。  
 本文完整代码和数据，已托管在Github上

Finally, I got the effect that I started showing in the text.

  The complete code and data for this article is hosted on Github <https://github.com/lps683/TrafficsSignDetection>。这些东西也许在高手看来不值一提，但是，若能给一部分人带来一些哪怕一点点收获，那么花这么多功夫写这篇文章也不算白费。

These things may not be worth mentioning in the hands of the masters, but if you can bring some even some gains to some people, it would not be a waste of time to spend so much effort to write this article.

* [查看图片附件](https://lps-683.iteye.com/blog/2360806) • View image attachments

**[平均薪酬38K! 前端工程师凭什么这么值钱？](http://www.baidu.com/cb.php?c=IgF_pyfqnHmkPj6vPjm0IZ0qnfK9ujYzrHn3rj6z0Aw-5Hnsn16YnjT0TAq15HfLP1fkrHn0T1dWnvw9ujR3nAfLuAmLPHcd0AwY5HDdPHD3rHDLnWm0IgF_5y9YIZK1rBtEuywdQhP1uA38UhNYQvPEIgF1uiqbugw9pysEP1bsPWqdIAdxTvqdThP-5yF_UvIGIAN4ufKzujY4rHb0mhYqnfKsTWYs0ZNGujYkPHTYn1mk0AqGujYknWb3rjDY0APGujYLn1bdn1n0ULI85H00TZbqnW0v0APzm1YknHnkn6" \t "_blank)**

[2019年，这类程序员需求增速达174%？原因竟是...](http://www.baidu.com/cb.php?c=IgF_pyfqnHmkPj6vPjm0IZ0qnfK9ujYzrHn3rj6z0Aw-5Hnsn16YnjT0TAq15HfLP1fkrHn0T1dWnvw9ujR3nAfLuAmLPHcd0AwY5HDdPHD3rHDLnWm0IgF_5y9YIZK1rBtEuywdQhP1uA38UhNYQvPEIgF1uiqbugw9pysEP1bsPWqdIAdxTvqdThP-5yF_UvIGIAN4ufKzujY4rHb0mhYqnfKsTWYs0ZNGujYkPHTYn1mk0AqGujYknWb3rjDY0APGujYLn1bdn1n0ULI85H00TZbqnW0v0APzm1YknHnkn6" \t "_blank)

分享到： [https://lps-683.iteye.com/images/sina.jpg](javascript:;) [https://lps-683.iteye.com/images/tec.jpg](javascript:;)

[JavaEE入门看这个系列真的就够了(一)---- ...](https://lps-683.iteye.com/blog/2360233) | [并查集算法关于渗透模型中虚拟节点的使用问 ...](https://lps-683.iteye.com/blog/2354888)

* 2017-03-09 19:39
* 浏览 10825
* [评论(1)](https://lps-683.iteye.com/blog/2360806#comments)
* 分类:[编程语言](https://www.iteye.com/blogs/category/language)
* [查看更多](https://www.iteye.com/wiki/blog/2360806)

评论

1 楼 [liweicong](https://liweicong.iteye.com/) 2018-01-19

这个识别的准确率是多少呢？训练的那个circle和四个参数代表什么意思呢？谢谢！

发表评论

[https://lps-683.iteye.com/images/login_icon.png 您还没有登录,请您登录后再发表评论](https://lps-683.iteye.com/login)

相关资源推荐

* [交通标牌检测与识别](https://blog.csdn.net/qq_19281769/article/details/84859993)

关于交通标牌检测的博客和论文非常多，例如，本人最近在博客上就看到有一篇简单的交通标志检测与识别介绍文章《自动驾驶之眼——摄像头是如何认识交通标志的？》，该文很简洁明了地阐述了交通标志检测识别的主要流程。本文将结合上学期的课程设计，来整理一下交通标识牌检测与识别的思路与实现方法。   一、要求 首先要明确一下本文到底是要干什么。本文要完成基于视觉的交通标识牌检测与识别，说白了，就两个事：1...

* [交通标识的定位以及识别](https://download.csdn.net/download/weishuang123123/9804461)

c++实现交通标识的定位以及识别，采用颜色分割以及连通域实现交通标识的初步定位，在采用hog特征以及svm对交通标示进行识别，使用vs2013和opencv实现，效果很好

* [交通标志牌的检测与识别](https://blog.csdn.net/u011966172/article/details/11524713)

Driver Assistance System已经成为学术研究的一大领域，作为智能交通的一部分，有大量研究人员投身其中，每年都会有大量的成果发表在IEEE trans on intelligent transportation system等顶级期刊，会议上。有些成熟技术甚至已走出学术界，应用于工业界，如福特公司的自动刹车系统，Google 的无人驾驶汽车，而 Mobileye 甚至已将其固

* [交通标志识别](https://blog.csdn.net/sinat_14916279/article/details/72452195)

交通标志识别，采用GTSRB数据集和MCDNN多列卷积神经网络进行识别实验。

* [车拍条件下交通标志实时识别](https://blog.csdn.net/hgzhgzhgz/article/details/57423810)

自己项目做的实时交通标志识别。主要基于RGB颜色空间进行分割，图像识别运用了BP神经网络，特征是提取zernike矩，因为zernike特征具有旋转不变形。 交通标志有着显著的颜色和形状特征，主要功能以指示、提示和警示为主。根据交通标志的以上特点，我们将图像中的目标进行颜色和形状的分割，在处理后加以匹配和识别。 项目总体分为三个部分，分别是目标提取、特征提取、识别。

Related Resources Recommended

• Traffic signage detection and identification

There are many blogs and papers on traffic signage detection. For example, I recently saw a simple article on traffic sign detection and identification on the blog, "The Eye of Autopilot" - How does the camera know the traffic sign? "This article is a very clear and concise description of the main process of traffic sign detection and identification. This article will combine the curriculum design of the last semester to sort out the ideas and implementation methods of traffic signage detection and recognition. First, the requirements First of all, we must clearly understand what this article is doing. This article is to complete the vision-based traffic signage detection and identification, to put it bluntly, two things: 1...

**• Location and identification of traffic signs**

C++ realizes the location and identification of traffic signs, adopts color segmentation and connected domains to realize the initial positioning of traffic signs, and uses hog features and svm to identify traffic signs, using vs2013 and opencv to achieve good results.

**• Detection and identification of traffic signs**

The Driver Assistance System has become a major area of ​​academic research. As part of intelligent transportation, a large number of researchers have devoted themselves to it. Every year, a large number of results are published in top journals such as the IEEE trans on intelligent transportation system. Some mature technologies have even gone out of academia, applied to industry, such as Ford's automatic braking system, Google's driverless car, and Mobileye has even fixed it

**• Traffic sign recognition**

Traffic sign recognition, using GTSRB data sets and MCDNN multi-column convolution neural network for identification experiments.

• **Real-time identification** of traffic signs under car shooting conditions

Real-time traffic sign recognition by own projects. It is mainly based on RGB color space for segmentation. Image recognition uses BP neural network. The feature is to extract the zernike moment because the zernike feature has rotation without deformation. Traffic signs have significant color and shape characteristics, and the main functions are indications, prompts, and warnings. According to the above characteristics of traffic signs, we divide the objects in the image into colors and shapes, and match and identify them after processing. The project is divided into three parts, namely target extraction, feature extraction and recognition.

* [opencv 2 + python 交通指示牌检测](https://blog.csdn.net/qq_41943402/article/details/81121896)
* • opencv 2 + python traffic sign detection
* [TensorFlow实战（四）——交通标志牌（BelgiumTS交通数据集）的识别（一）](https://blog.csdn.net/weixin_41695564/article/details/80270153)

• TensorFlow combat (4) – Identification of traffic signs (BelgiumTS traffic dataset) (1)

前言：    本文将学习基于TensorFlow，利用比利时交通标志数据集——BelgiumTS交通信号数据集（包含62中交通信号），进行训练简单的卷积神经网络完成识别。一、交通标志数据集      1.1 数据集下载      交通标志识别是一个很好的入门练手项目，在网上可以找到很多的相关资料。首先呢，去找到一个优秀的训练数据集是至关重要的。在这里，我选择了比利时交通标志数据集，那么如何下载数据...

Foreword: This article will learn to use TensorFlow based on the Belgian traffic sign data set - BelgiumTS traffic signal data set (including 62 traffic signals), to perform a simple training convolutional neural network to complete the identification. First, traffic signs data set 1.1 Data set download Traffic sign recognition is a good entry-level training project, you can find a lot of relevant information online. First of all, it is crucial to find a good training data set. Here, I chose the Belgian traffic sign dataset, then how to download the data...

* [交通标志牌识别样本数据及相应算法](https://blog.csdn.net/alec1987/article/details/18040011)
* [基于机器视觉的交通标志识别系统](https://download.csdn.net/download/weixin_38902664/10717139)

1、有一个 图形化界面，左面有一个显示窗口，可以 显示视频或 者图片，并且有两个按钮，分别为导入视频、导入图片；右侧为 交通标志检测结果显示模块，可以对 左侧视频或图片中出现的交 通标志进行实时显示，对于视频的显示，一帧显示检测结果后， 下一帧检测结果可以覆盖上一帧检测结果。 2、要能够对视频进行 实时处理，首先检测环境中存在的交通标 志，然后对于交通标志进行识别； 3、对于交通标志的 检测与识别两部分， 都要结合机器学习，能 够实现优化，可以让我 使用正样本及负样本对于程序进行训练， 对于新增的样本，我也要可以导入到程序中进行训练，不局限于 一部分样本； 4、一帧图像中，有 多个交通标志时，也要做到全部显示出来， 不局限于只显示一个交通标志；

* [交通标识牌检测及识别c++代码实例及运行结果 （可自行在网上下载图片测试）](https://blog.csdn.net/IT_job/article/details/79028868)

运行环境：vs2013+opencv2.4.9+win10 数据来源于GTSRB 效果不是很理想（预处理方法、检测用的rgb2hsv、圆度检测，、参数，总之改变程序中很多东西可以尝试提高准确率），但检测及识别的道路是打通了 c++代码 #include #include #include #define PI 3.1415926 using namespace std; using

* [交通标志识别程序及数据](https://download.csdn.net/download/applefl/6966463)

利用神经网络算法做的交通标志识别程序，界面友好，附有操作说明，可以作为学习之用。 开发环境VS2010+OpenCV2.4.6（注意环境的配置，程序本身应该没什么问题，环境没配好不要怪我^\_^）

* [使用TensorFlow识别交通标志](https://blog.csdn.net/fu_shuwu/article/details/77073857)

这篇博客是翻译Waleed Abdulla写的使用TensorFlow识别交通标志，作者已经授权翻译，这是原文。 我看到了速度限制标志，但只是没有看见你 这是使用深度学习模型去识别交通标志的第一部分。本系列的目的是学习如何使用深度模型去构建一个系统，如何你也有兴趣可以和我一起学习。在网上，你能找到很多的讲解神经网络数学理论的资源，因此我将专注于应用实践方面的分享。接

* [交通标识检测c++代码实例及运行结果](https://blog.csdn.net/IT_job/article/details/78988301)

环境vs2013+opencv2.4.9 交通标识分为检测和识别两部分，检测只检测红色标识，其他标识同理。本片博客介绍交通标识检测部分，识别部分后续介绍。 c++代码 #include #include #define PI 3.1415926 using namespace std; using namespace cv; void RGB2HSV(double red, do

* [交通标识牌识别c++代码实例及运行结果](https://blog.csdn.net/IT_job/article/details/79016426)
* [交通标志的检测与识别](https://blog.csdn.net/xiaojiajia007/article/details/53183102)

本科毕业设计的时候曾经做过交通标志的检测与识别，这个作为很好的图像处理与识别领域的入门project还是非常不错的，接下来如果有时间的话我将慢慢整理一下，本次首先给出我的总体目录。在我的工作里，既有理论的推导，也有实际的测试程序可以运行。不过打算在博客里基本上只进行文字性说明，如果真的可以帮到大家，那再上传我的源程序吧。大家一起学习！ 第1章  绪 论 1.1 交通标志识别算法的研究背

* [交通图标识别](https://blog.csdn.net/forest_world/article/details/77918377)
* [python数据分析与挖掘学习笔记（7）-交通路标自动识别实战与神经网络算法](https://blog.csdn.net/Ying_Xu/article/details/54951354)

这一节主要涉及神经网络算法，由此展开交通路标自动识别的应用。 交通路标的自动识别其实就是一个分类问题。对于分类问题，我们有很多的方法来实现，比如KNN，贝叶斯等。关键点在于图片转文本。本节采用人工神经网络算法来进行识别。 人工神经网络（Artificial Neural Network）是简称神经网络(NN)，是基于生物学中神经网络的基本原理，在理解和抽象了人脑结构和外界刺激响应机制后，以网络

* [德国交通标志检测识别数据集](https://blog.csdn.net/qq_34793133/article/details/80401496)
* [基于opencv的交通标志识别](https://download.csdn.net/download/u013842516/9659031)

基于opencv的交通标志识别，主要运用轮廓识别和模板匹配。适用于简单自然条件下的识别

* [python3下tensorflow练习（七）之交通标志识别（两层神经网络）](https://blog.csdn.net/orange_littlegirl/article/details/80302360)

根据udacity的自动驾驶课程，搭建两层神经网络对43种交通标志进行分类识别，用到的数据库在该网站可以下载到：数据集 我已经将数据集下好放在百度云分享给各位小伙伴：百度云链接 本文用到的思想和上一篇博文类似，我们先回顾一下上篇博文MNIST多层神经网络识别 好了直接开始撸代码吧： 1.加载需要的模块 import numpy as np import tensorflow as tf...

* [基于GTSRB数据集CNN交通标志识别](https://download.csdn.net/download/u013842516/9881904)

基于GTSRB数据集CNN，卷积神经网络交通标志识别

* [利用pytorch实现交通标志识别](https://blog.csdn.net/ZOUZHEN_ID/article/details/83958772)

利用pytorch实现交通标志识别代码实现主要问题——数据读取 本文是在https://www.jianshu.com/p/d8feaddc7bdf文章的基础上用Pytorch实现的 话不多说，直接上代码，具体的可以看代码中的解释 代码实现 import os import torch import torchvision as tv import torchvision.transforms a...

* [TensorFlow实战（五）——交通标志牌（GTSRB数据集）的格式转换（一）](https://blog.csdn.net/weixin_41695564/article/details/80292556)

前言：    本文将学习基于TensorFlow，利用德国交通标志数据集——GTSRB交通信号数据集（包含62中交通信号），进行训练简单的卷积神经网络完成识别。一、交通标志数据集      1.1 数据集下载      交通标志识别是一个很好的入门练手项目，在网上可以找到很多的相关资料。首先呢，去找到一个优秀的训练数据集是至关重要的。在这里，我选择了德国交通标志数据集，那么如何下载数据集呢？   ...

* [交通信号灯的检测与识别](https://blog.csdn.net/step_forward_ML/article/details/79891535)

交通信号灯的检测与识别是无人驾驶与辅助驾驶必不可少的一部分，其识别精度直接关乎智能驾驶的安全。一般而言，在实际的道路场景中采集的交通信号灯图像具有复杂的背景，且感兴趣的信号灯区域只占很少的一部分，如下图所示。针对这些难点，国内外的众多研究者提出了相应的解决方案。总的来说，大多基于传统的图像处理方法；但目前也有用强学习能力的卷积神经网络去进行识别，但这类方法往往需要大量的训练样本避免过拟合的风险。

* [自动驾驶汽车 实现实时交通信号灯检测和分类](https://blog.csdn.net/gikod/article/details/78883181)

# 深度学习 自动驾驶汽车 实现实时交通信号灯检测和分类 优化的R-CNN（https://arxiv.org/abs/1506.01497）     Google使用提取检测到的交通信号灯的方法，然后在该方法上运行第二个分类器。提供了灵活性;然而，根据实现的不同，它可能会增加管道复杂度和计算成本。更重要的是，它似乎依赖于对预期交通信号灯位置的先验信息。一般来说，将分类作为第二步添加第二个网络进

* [利用tensorflow进行德国交通标志识别》](https://blog.csdn.net/chaojipikaqiu/article/details/82149686)

一，前言 利用暑假学习了一段时间的DL，德国交通标志马上实践一下，下面主要提到我用的模型以及数据处理，如果有哪些地方大佬们觉得哪些地方做得不好或者可以改进得，非常欢迎直接在评论里提出来，初学需要一起努力。 二，数据   上面是网站中给出来的数据我已经下载下来并且保存到百度网盘中了，看情况下载（网盘限速），网盘链接：链接：https://pan.baidu.com/s/1yJ202Etr...

* [卷积神经网络CNN（7）—— 限速交通标志分类](https://blog.csdn.net/Fate_fjh/article/details/74784634)

1.前言限速交通标志识别在ADAS或者自动驾驶领域中相对基础的范畴，因此限速交通标志识别的要求都是又快又准，同时使用相对简单的方法实现，本文将整个限速交通标志检测与分类的过程以及实现方法。不过实际场景中，会出现限速交通标志容易被阻挡，或是离当前车道很远无法检测出来等问题有待解决，而且都具有一定挑战性。2.限速交通标志分类流程1.交通标志位置与大小检测：使用物体检测的CNN完成，本人使用的是yolo。

* [matlab的交通灯信号识别](https://download.csdn.net/download/xieexiaotuzi/6531257)

基于matlab的信号灯识别 嗯 有详细代码 可实现

* [无人驾驶项目——交通标志识别](https://blog.csdn.net/zhangqian_shai/article/details/80031817)

在无人驾驶项目中，实现交通标志识别是一项重要工作。本文以德国交通标志数据集为训练对象，采用深度神经网络LeNet架构处理图像，实现交通标志识别。具体处理过程包括包括：数据导入探索和可视化数据集数据预处理构建、训练和测试模型架构采用该模型对新图片进行预测分析新图片的softmax概率1.数据导入下载数据 &quot;traffic-signs-data.zip&quot;读取文件&quot; train.p&quot; and &quot;test...

* [阿波罗——交通信号灯感知](https://blog.csdn.net/baobei0112/article/details/81220465)

交通信号灯感知 本文档详细的介绍了Apollo2.0中交通信号感知模块的工作原理。 简介 交通信号灯感知模块通过使用摄像头提供精确全面的路面交通信号灯状态。 通常情况下，交通信号灯有3种状态： 红 黄 绿 然而当信号灯不能正常工作时，它可能是黑色的或者闪烁着红灯或黄灯。有时候在摄像头的视野内找不到信号灯，从而导致无法正确检测信号灯状态。 为了覆盖全部的情况，交通信号灯感知模块提供了...

上滑加载更多

[交通标牌检测与识别](https://blog.csdn.net/qq_19281769/article/details/84859993" \t "_blank)

[2017-03-09 279](https://blog.csdn.net/qq_19281769/article/details/84859993" \t "_blank)

[关于交通标牌检测的博客和论文非常多，例如，本人最近在博客上就看到有一篇简单的交通标志检测与识别介绍文章《自动驾驶之眼——摄像头是如何认识交通标志的？》，该文很简洁明了地阐述了交通标志检测识别的主要流程。本文将结合上学期的课程设计，来整理一下交通标识牌检测与识别的思路与实现方法。   一、要求 首先要明确一下本文到底是要干什么。本文要完成基于视觉的交通标识牌检测与识别，说白了，就两个事：1...](https://blog.csdn.net/qq_19281769/article/details/84859993" \t "_blank)

[交通标识的定位以及识别](https://download.csdn.net/download/weishuang123123/9804461" \t "_blank)

[2017-04-05 18](https://download.csdn.net/download/weishuang123123/9804461" \t "_blank)

[c++实现交通标识的定位以及识别，采用颜色分割以及连通域实现交通标识的初步定位，在采用hog特征以及svm对交通标示进行识别，使用vs2013和opencv实现，效果很好](https://download.csdn.net/download/weishuang123123/9804461" \t "_blank)

[交通标志牌的检测与识别](https://blog.csdn.net/u011966172/article/details/11524713" \t "_blank)

[2013-09-10 9758](https://blog.csdn.net/u011966172/article/details/11524713" \t "_blank)

[Driver Assistance System已经成为学术研究的一大领域，作为智能交通的一部分，有大量研究人员投身其中，每年都会有大量的成果发表在IEEE trans on intelligent transportation system等顶级期刊，会议上。有些成熟技术甚至已走出学术界，应用于工业界，如福特公司的自动刹车系统，Google 的无人驾驶汽车，而 Mobileye 甚至已将其固](https://blog.csdn.net/u011966172/article/details/11524713" \t "_blank)

[车拍条件下交通标志实时识别](https://blog.csdn.net/hgzhgzhgz/article/details/57423810" \t "_blank)

[2017-02-26 5850](https://blog.csdn.net/hgzhgzhgz/article/details/57423810" \t "_blank)

[自己项目做的实时交通标志识别。主要基于RGB颜色空间进行分割，图像识别运用了BP神经网络，特征是提取zernike矩，因为zernike特征具有旋转不变形。 交通标志有着显著的颜色和形状特征，主要功能以指示、提示和警示为主。根据交通标志的以上特点，我们将图像中的目标进行颜色和形状的分割，在处理后加以匹配和识别。 项目总体分为三个部分，分别是目标提取、特征提取、识别。](https://blog.csdn.net/hgzhgzhgz/article/details/57423810" \t "_blank)

[opencv 2 + python 交通指示牌检测](https://blog.csdn.net/qq_41943402/article/details/81121896" \t "_blank)

[2018-07-25 693](https://blog.csdn.net/qq_41943402/article/details/81121896" \t "_blank)

[import cv2 import numpy as np np.set\_printoptions(suppress=True) img=cv2.imread('/home/gg/Pictures/jt.jpg') print('img:',type(img),img.shape,img.dtype) cv2.imshow('img',img) hsv=cv2.cvtColor(img,cv...](https://blog.csdn.net/qq_41943402/article/details/81121896" \t "_blank)

[交通标志识别](https://blog.csdn.net/sinat_14916279/article/details/72452195" \t "_blank)

[2017-05-17 13378](https://blog.csdn.net/sinat_14916279/article/details/72452195" \t "_blank)

[交通标志识别，采用GTSRB数据集和MCDNN多列卷积神经网络进行识别实验。](https://blog.csdn.net/sinat_14916279/article/details/72452195" \t "_blank)

[TensorFlow实战（四）——交通标志牌（BelgiumTS交通数据集）的识别（一）](https://blog.csdn.net/weixin_41695564/article/details/80270153" \t "_blank)

[2018-05-12 2315](https://blog.csdn.net/weixin_41695564/article/details/80270153" \t "_blank)

[前言：    本文将学习基于TensorFlow，利用比利时交通标志数据集——BelgiumTS交通信号数据集（包含62中交通信号），进行训练简单的卷积神经网络完成识别。一、交通标志数据集      1.1 数据集下载      交通标志识别是一个很好的入门练手项目，在网上可以找到很多的相关资料。首先呢，去找到一个优秀的训练数据集是至关重要的。在这里，我选择了比利时交通标志数据集，那么如何下载数据...](https://blog.csdn.net/weixin_41695564/article/details/80270153" \t "_blank)

[基于机器视觉的交通标志识别系统](https://download.csdn.net/download/weixin_38902664/10717139" \t "_blank)

[2018-10-12 19](https://download.csdn.net/download/weixin_38902664/10717139" \t "_blank)

[1、有一个 图形化界面，左面有一个显示窗口，可以 显示视频或 者图片，并且有两个按钮，分别为导入视频、导入图片；右侧为 交通标志检测结果显示模块，可以对 左侧视频或图片中出现的交 通标志进行实时显示，对于视频的显示，一帧显示检测结果后， 下一帧检测结果可以覆盖上一帧检测结果。 2、要能够对视频进行 实时处理，首先检测环境中存在的交通标 志，然后对于交通标志进行识别； 3、对于交通标志的 检测与识别两部分， 都要结合机器学习，能 够实现优化，可以让我 使用正样本及负样本对于程序进行训练， 对于新增的样本，我也要可以导入到程序中进行训练，不局限于 一部分样本； 4、一帧图像中，有 多个交通标志时，也要做到全部显示出来， 不局限于只显示一个交通标志；](https://download.csdn.net/download/weixin_38902664/10717139" \t "_blank)

[交通标志牌识别样本数据及相应算法](https://blog.csdn.net/alec1987/article/details/18040011" \t "_blank)

[2014-01-09 3755](https://blog.csdn.net/alec1987/article/details/18040011" \t "_blank)

[http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset HOME GTSRB NEWS ABOUT DATASET RESULTS SCHEDULE SUBMISSIONS CONTACT GTSDB LOGIN / SIGN UP     Dataset Overview St](https://blog.csdn.net/alec1987/article/details/18040011" \t "_blank)

[交通标识牌检测及识别c++代码实例及运行结果 （可自行在网上下载图片测试）](https://blog.csdn.net/IT_job/article/details/79028868" \t "_blank)

[2018-01-10 1672](https://blog.csdn.net/IT_job/article/details/79028868" \t "_blank)

[运行环境：vs2013+opencv2.4.9+win10 数据来源于GTSRB 效果不是很理想（预处理方法、检测用的rgb2hsv、圆度检测，、参数，总之改变程序中很多东西可以尝试提高准确率），但检测及识别的道路是打通了 c++代码 #include #include #include #define PI 3.1415926 using namespace std; using](https://blog.csdn.net/IT_job/article/details/79028868" \t "_blank)

[交通标志识别程序及数据](https://download.csdn.net/download/applefl/6966463" \t "_blank)

[2014-02-26 218](https://download.csdn.net/download/applefl/6966463" \t "_blank)

[利用神经网络算法做的交通标志识别程序，界面友好，附有操作说明，可以作为学习之用。 开发环境VS2010+OpenCV2.4.6（注意环境的配置，程序本身应该没什么问题，环境没配好不要怪我^\_^）](https://download.csdn.net/download/applefl/6966463" \t "_blank)

[使用TensorFlow识别交通标志](https://blog.csdn.net/fu_shuwu/article/details/77073857" \t "_blank)

[2017-08-10 7938](https://blog.csdn.net/fu_shuwu/article/details/77073857" \t "_blank)

[这篇博客是翻译Waleed Abdulla写的使用TensorFlow识别交通标志，作者已经授权翻译，这是原文。 我看到了速度限制标志，但只是没有看见你 这是使用深度学习模型去识别交通标志的第一部分。本系列的目的是学习如何使用深度模型去构建一个系统，如何你也有兴趣可以和我一起学习。在网上，你能找到很多的讲解神经网络数学理论的资源，因此我将专注于应用实践方面的分享。接](https://blog.csdn.net/fu_shuwu/article/details/77073857" \t "_blank)

[交通标识检测c++代码实例及运行结果](https://blog.csdn.net/IT_job/article/details/78988301" \t "_blank)

[2018-01-06 1032](https://blog.csdn.net/IT_job/article/details/78988301" \t "_blank)

[环境vs2013+opencv2.4.9 交通标识分为检测和识别两部分，检测只检测红色标识，其他标识同理。本片博客介绍交通标识检测部分，识别部分后续介绍。 c++代码 #include #include #define PI 3.1415926 using namespace std; using namespace cv; void RGB2HSV(double red, do](https://blog.csdn.net/IT_job/article/details/78988301" \t "_blank)

[交通标识牌识别c++代码实例及运行结果](https://blog.csdn.net/IT_job/article/details/79016426" \t "_blank)

[2018-01-09 741](https://blog.csdn.net/IT_job/article/details/79016426" \t "_blank)

[运行环境：vs2013+opencv2.4.9+win10 数据来源于GTSRB。 模型训练的C++代码 #include #include #include #include #include #include using namespace cv; using namespace std; int main() { int imgWidht = 48;//重](https://blog.csdn.net/IT_job/article/details/79016426" \t "_blank)

[交通标志的检测与识别](https://blog.csdn.net/xiaojiajia007/article/details/53183102" \t "_blank)

[2016-11-16 10132](https://blog.csdn.net/xiaojiajia007/article/details/53183102" \t "_blank)

[本科毕业设计的时候曾经做过交通标志的检测与识别，这个作为很好的图像处理与识别领域的入门project还是非常不错的，接下来如果有时间的话我将慢慢整理一下，本次首先给出我的总体目录。在我的工作里，既有理论的推导，也有实际的测试程序可以运行。不过打算在博客里基本上只进行文字性说明，如果真的可以帮到大家，那再上传我的源程序吧。大家一起学习！ 第1章  绪 论 1.1 交通标志识别算法的研究背](https://blog.csdn.net/xiaojiajia007/article/details/53183102" \t "_blank)

[](https://bbs.csdn.net/forums/Service?utm_source=csdn_footer)

微信客服

[](http://wpa.b.qq.com/cgi/wpa.php?ln=1&key=XzgwMDE4MDEwNl80ODc3MzVfODAwMTgwMTA2XzJf)

QQ客服

[kefu@csdn.net](mailto:webmaster@csdn.net)[QQ客服](http://wpa.b.qq.com/cgi/wpa.php?ln=1&key=XzgwMDE4MDEwNl80ODc3MzVfODAwMTgwMTA2XzJf)

[客服论坛](http://bbs.csdn.net/forums/Service)400-660-0108

工作时间 8:30-22:00

[**关于我们**](https://www.csdn.net/company/index.html#about)[**招聘**](https://www.csdn.net/company/index.html#recruit)[**广告服务**](https://www.csdn.net/company/index.html#contact) [**网站地图**](https://www.csdn.net/gather/A)

[百度提供站内搜索](https://zn.baidu.com/cse/home/index) [京ICP证19004658号](http://www.miibeian.gov.cn/)

©1999-2019 北京创新乐知网络技术有限公司

经营性网站备案信息[网络110报警服务](http://www.cyberpolice.cn/)

[北京互联网违法和不良信息举报中心](http://www.bjjubao.org/)

[中国互联网举报中心](http://www.12377.cn/)

 Global site tag (gtag.js) - Google Analytics