

# 基于Python对交通路口的红绿灯进行颜色检测

新机器视觉 昨天

点击下方**卡片**，关注“**新机器视觉**”公众号

视觉/图像重磅干货，第一时间送达



新机器视觉

最前沿的机器视觉与计算机视觉技术

206篇原创内容



公众号

转自 | Python联盟

## 1. 视频读取

首先把视频读取进来，因为我测试的视频是4k的所以我用resize调整了一下视频的分辨大小

```
1 cap = cv2.VideoCapture('video/小路口.mp4')
2 while True:
3     ret, frame = cap.read()
4     if ret == False:
5         break
6     frame = cv2.resize(frame, (1920, 1080))
```

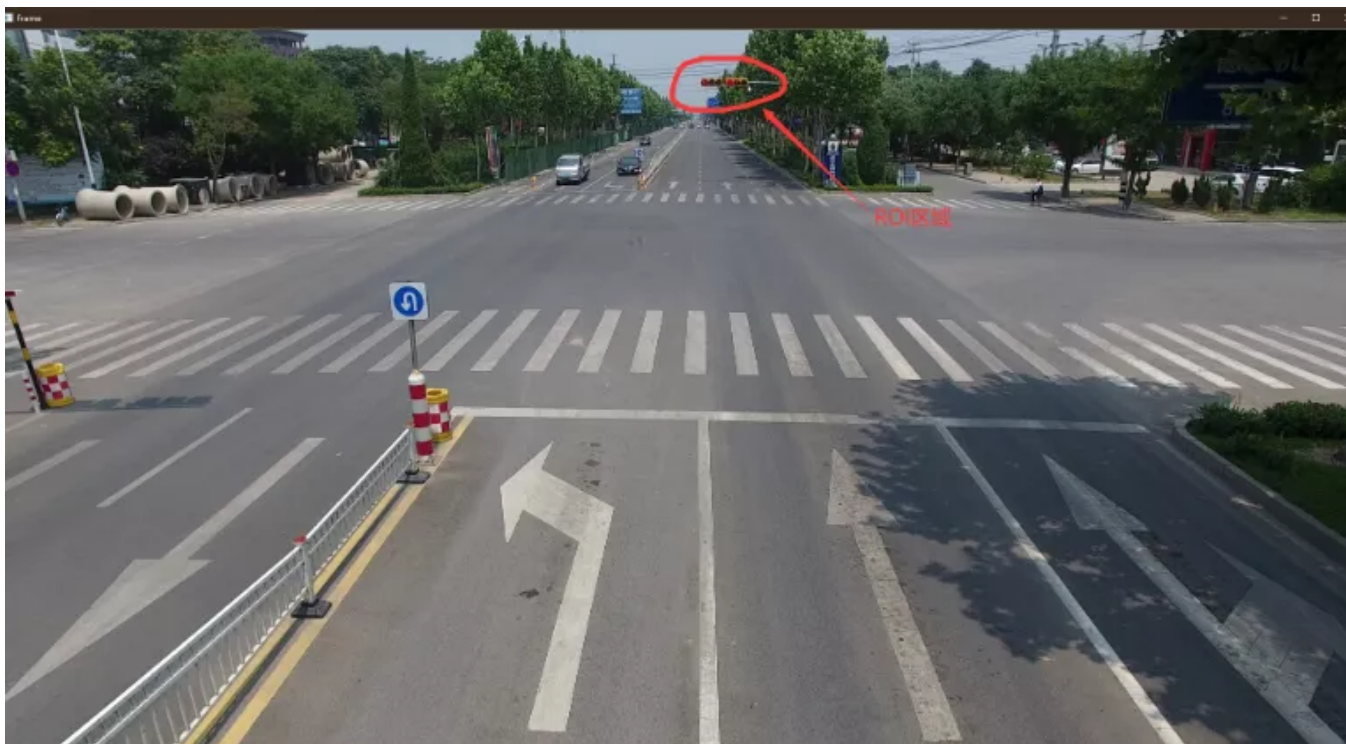
```
7     cv2.imshow('frame', frame)
8     c = cv2.waitKey(10)
9     if c==27:
10         break
```

imshow() (如下图所示)

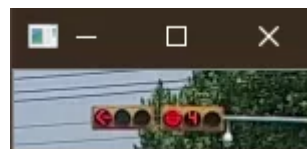


## 2.截取roi区域

截取roi的区域，也就是说，为了避免多余的干扰因素我们要把红绿灯的位置给截取出来（如下图所示）



截取后的roi (如下图所示)



### 3.转换hsv颜色空间

HSV颜色分量范围

(详细参考: <https://www.cnblogs.com/wangyblzu/p/5710715.html>)

一般对颜色空间的图像进行有效处理都是在HSV空间进行的, 然后对于基本色中对应的HSV分量需要给定一个严格的范围, 下面是通过实验计算的模糊范围(准确的范围在网上都没有给出)。

H: 0— 180

S: 0— 255

V: 0— 255

此处把部分红色归为紫色范围(如下图所示):

	黑	灰	白	红		橙	黄	绿	青	蓝	紫
hmin	0	0	0	0	156	11	26	35	78	100	125
hmax	180	180	180	10	180	25	34 ;	77	99	124	155
smin	0	0	0	43		43	43	43	43	43	43
smax	255	43	30	255		255	255	255	255	255	255
vmin	0	46	221	46		46	46	46	46	46	46
vmax	46	220	255	255		255	255	255	255	255	255

上面是已给好特定的颜色值, 如果你的颜色效果不佳, 可以通过python代码来对min和max值的微调, 用opencv中的api来获取你所需理想的颜色, 可以复制以下代码来进行颜色的调整。

1.首先你要截取roi区域的一张图片

2.读取这张图然后调整颜色值

颜色调整代码如下:

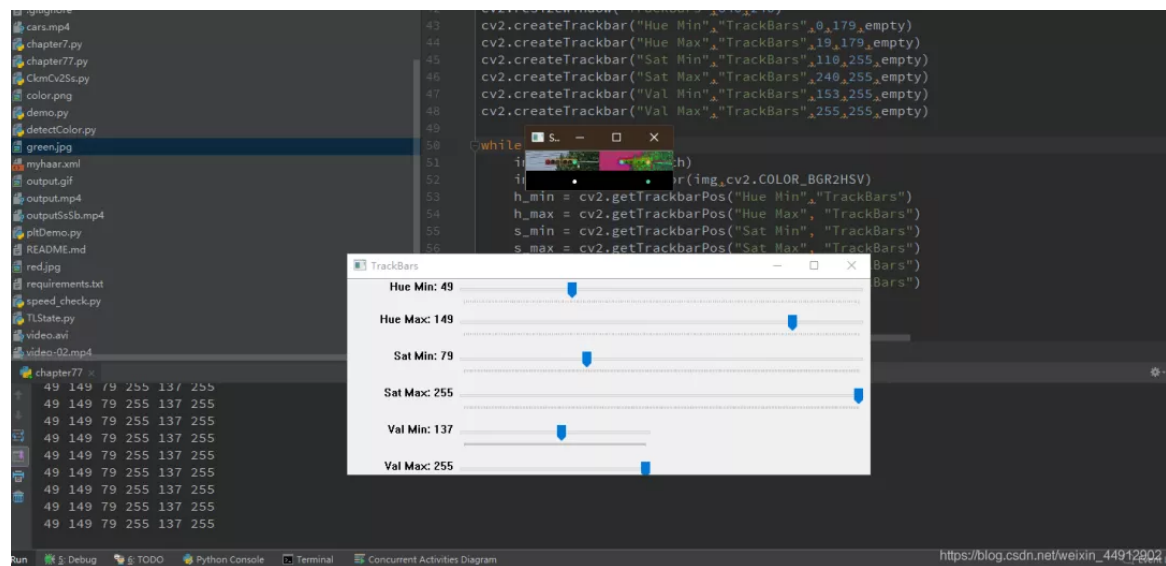
(详细参考: <https://www.bilibili.com/video/BV16K411W7x9>)

```
1 import cv2
2 import numpy as np
3
4 def empty(a):
5     pass
6
7 def stackImages(scale,imgArray):
8     rows = len(imgArray)
9     cols = len(imgArray[0])
10    rowsAvailable = isinstance(imgArray[0], list)
11    width = imgArray[0][0].shape[1]
12    height = imgArray[0][0].shape[0]
13    if rowsAvailable:
14        for x in range ( 0, rows):
15            for y in range(0, cols):
16                if imgArray[x][y].shape[:2] == imgArray[0][0].shape [:2]:
17                    imgArray[x][y] = cv2.resize(imgArray[x][y], (0, 0), None, scale, scale)
18                else:
19                    imgArray[x][y] = cv2.resize(imgArray[x][y], (imgArray[0][0].shape[1], imgArray[0]
20 [0].shape[0]), None, scale, scale)
21                if len(imgArray[x][y].shape) == 2: imgArray[x][y]= cv2.cvtColor( imgArray[x][y],
22 cv2.COLOR_GRAY2BGR)
23            imageBlank = np.zeros((height, width, 3), np.uint8)
24            hor = [imageBlank]*rows
25            hor_con = [imageBlank]*rows
26            for x in range(0, rows):
27                hor[x] = np.hstack(imgArray[x])
```

```
28     ver = np.vstack(hor)
29     else:
30         for x in range(0, rows):
31             if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
32                 imgArray[x] = cv2.resize(imgArray[x], (0, 0), None, scale, scale)
33             else:
34                 imgArray[x] = cv2.resize(imgArray[x], (imgArray[0].shape[1], imgArray[0].shape[0]), None, scale,
35 scale)
36             if len(imgArray[x].shape) == 2: imgArray[x] = cv2.cvtColor(imgArray[x], cv2.COLOR_GRAY2BGR)
37         hor= np.hstack(imgArray)
38         ver = hor
39     return ver
40
41 #读取的图片路径
42 path = './green.jpg'
43 cv2.namedWindow("TrackBars")
44 cv2.resizeWindow("TrackBars", 640, 240)
45 cv2.createTrackbar("Hue Min", "TrackBars", 0, 179, empty)
46 cv2.createTrackbar("Hue Max", "TrackBars", 19, 179, empty)
47 cv2.createTrackbar("Sat Min", "TrackBars", 110, 255, empty)
48 cv2.createTrackbar("Sat Max", "TrackBars", 240, 255, empty)
49 cv2.createTrackbar("Val Min", "TrackBars", 153, 255, empty)
50 cv2.createTrackbar("Val Max", "TrackBars", 255, 255, empty)
51
52 while True:
53     img = cv2.imread(path)
54     imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

```
55     h_min = cv2.getTrackbarPos("Hue Min", "TrackBars")
56     h_max = cv2.getTrackbarPos("Hue Max", "TrackBars")
57     s_min = cv2.getTrackbarPos("Sat Min", "TrackBars")
58     s_max = cv2.getTrackbarPos("Sat Max", "TrackBars")
59     v_min = cv2.getTrackbarPos("Val Min", "TrackBars")
60     v_max = cv2.getTrackbarPos("Val Max", "TrackBars")
61     print(h_min, h_max, s_min, s_max, v_min, v_max)
62     lower = np.array([h_min, s_min, v_min])
63     upper = np.array([h_max, s_max, v_max])
64     mask = cv2.inRange(imgHSV, lower, upper)
65     imgResult = cv2.bitwise_and(img, img, mask=mask)
66
67
    imgStack = stackImages(0.6, ([img, imgHSV], [mask, imgResult]))
    cv2.imshow("Stacked Images", imgStack)
    cv2.waitKey(1)
```

运行代码后调整的结果（如下图所示），很明显可以看到绿色已经被获取到。



## 4.二值图像颜色判定

因为图像是二值的图像，所以如果图像出现白点，也就是255，那么就取他的max最大值255，视频帧的不断变化然后遍历每个颜色值

```

1 red_color = np.max(red_blur)
2 green_color = np.max(green_blur)
3 if red_color == 255:
4     print('red')
5 elif green_color == 255:
6     print('green')

```



## 5.颜色结果画在图像上

用矩形框来框选出红绿灯区域

```
1 cv2.rectangle(frame,(1020,50),(1060,90),(0,0,255),2) #按坐标画出矩形框
2 cv2.putText(frame, "red", (1020, 40), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255),2)#显示red文本信息
```

## 6.完整代码

```
1 import cv2
2 import numpy as np
3
4 cap = cv2.VideoCapture('video/小路口.mp4')
5 while True:
6     ret,frame = cap.read()
7     if ret == False:
8         break
9     frame = cv2.resize(frame,(1920,1080))
10    #截取roi区域
11    roiColor = frame[50:90,950:1100]
12    #转换hsv颜色空间
13    hsv = cv2.cvtColor(roiColor,cv2.COLOR_BGR2HSV)
14
15    #red
```

```

16 lower_hsv_red = np.array([157,177,122])
17 upper_hsv_red = np.array([179,255,255])
18 mask_red = cv2.inRange(hsv,lowerb=lower_hsv_red,upperb=upper_hsv_red)
19 #中值滤波
20 red_blur = cv2.medianBlur(mask_red, 7)
21 #green
22 lower_hsv_green = np.array([49,79,137])
23 upper_hsv_green = np.array([90,255,255])
24 mask_green = cv2.inRange(hsv,lowerb=lower_hsv_green,upperb=upper_hsv_green)
25 #中值滤波
26 green_blur = cv2.medianBlur(mask_green, 7)
27
28 #因为图像是二值的图像，所以如果图像出现白点，也就是255，那么就取他的max最大值255
29 red_color = np.max(red_blur)
30 green_color = np.max(green_blur)
31 #在red_color中判断二值图像如果数值等于255，那么就判定为red
32 if red_color == 255:
33     print('red')
34     #。。。这是我经常会混淆的坐标。。。就列举出来记一下。。。
35     #      y  y+h x  x+w
36     #frame[50:90,950:1100]
37
38     #      x  y      x+w  y+h
39     cv2.rectangle(frame,(1020,50),(1060,90),(0,0,255),2) #按坐标画出矩形框
40     cv2.putText(frame, "red", (1020, 40), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255),2)#显示red文本信息
41 #在green_color中判断二值图像如果数值等于255，那么就判定为green
42 elif green_color == 255:

```

```
43     print('green')
44     cv2.rectangle(frame, (1020, 50), (1060, 90), (0, 255, 0), 2)
45     cv2.putText(frame, "green", (1020, 40), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
46
47     cv2.imshow('frame', frame)
48     red_blur = cv2.resize(red_blur, (300, 200))
49     green_blur = cv2.resize(green_blur, (300, 200))
50     cv2.imshow('red_window', red_blur)
51     cv2.imshow('green_window', green_blur)
52
53     c = cv2.waitKey(10)
54     if c == 27:
55         break
```

检测红灯的效果（如下图所示）



检测绿灯的效果（如下图所示）



最后!!!

第一次接触opencv! 所以请各位视觉领域的大佬们勿喷我这个小菜鸡! (/狗头)

代码量非常少, 无泛化能力, 很low的一种做法。。。不过对于小白的我来说学习hsv颜色空间还是很有帮助滴! 干就完了! 奥利给!

—版权声明—

仅用于学术分享, 版权属于原作者。

若有侵权, 请联系微信号:yiyang-sy 删除或修改!

—THE END—

## 走进新机器视觉 · 拥抱机器视觉新时代

新机器视觉 —— 机器视觉领域服务平台  
媒体论坛/智库咨询/投资孵化/技术服务

商务合作：

投稿咨询：

产品采购：



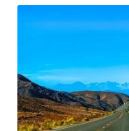
长按扫描右侧二维码关注“新机器视觉”公众号



喜欢此内容的人还喜欢

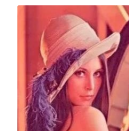
### 一文读懂javascript深拷贝与浅拷贝

前端技术优选



### 整理 Python 中的图像处理利器（共10个）

新机器视觉



### 深度学习优化算法的总结与梳理（从 SGD 到 AdamW 原理和代码解读）

新机器视觉

