

Halcon轉OpenCV實例--去除紙張中的顏色筆跡/墨跡(附源碼)

OpenCV與AI深度學習 前天

以下文章來源於OpenCV與Halcon視覺，作者Color Space



OpenCV與Halcon視覺

專注OpenCV、Halcon視覺相關學習內容的介紹、分享，喜歡就關注我們吧！



點擊上方“[OpenCV與AI深度學習](#)”，選擇加“[星標](#)”或“[置頂](#)”

重磅乾貨，第一時間送達！

導讀

本文主要介紹一個去除紙張中顏色筆跡/墨蹟的實例，並將Halcon實現轉為OpenCV。

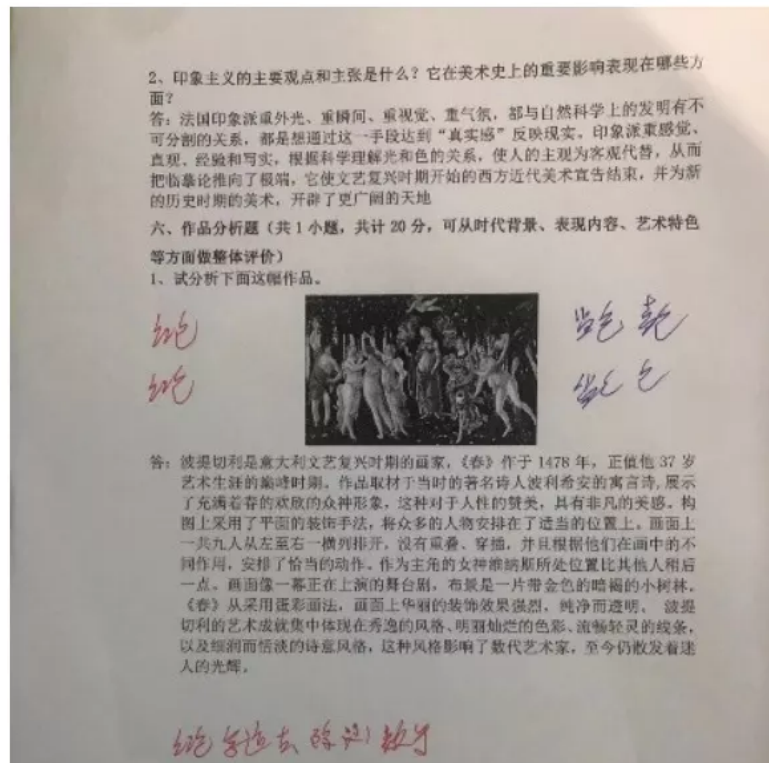
實例來源

實例來源於51Halcon論壇的討論貼：

<https://www.51halcon.com/forum.php?mod=viewthread&tid=4766>

[待解决] 消除红蓝颜色字体 [复制链接]

发表于 2021-6-11 21:30:31 | 只看该作者 ▶



OpenCV与Halcon视觉

Halcon實現

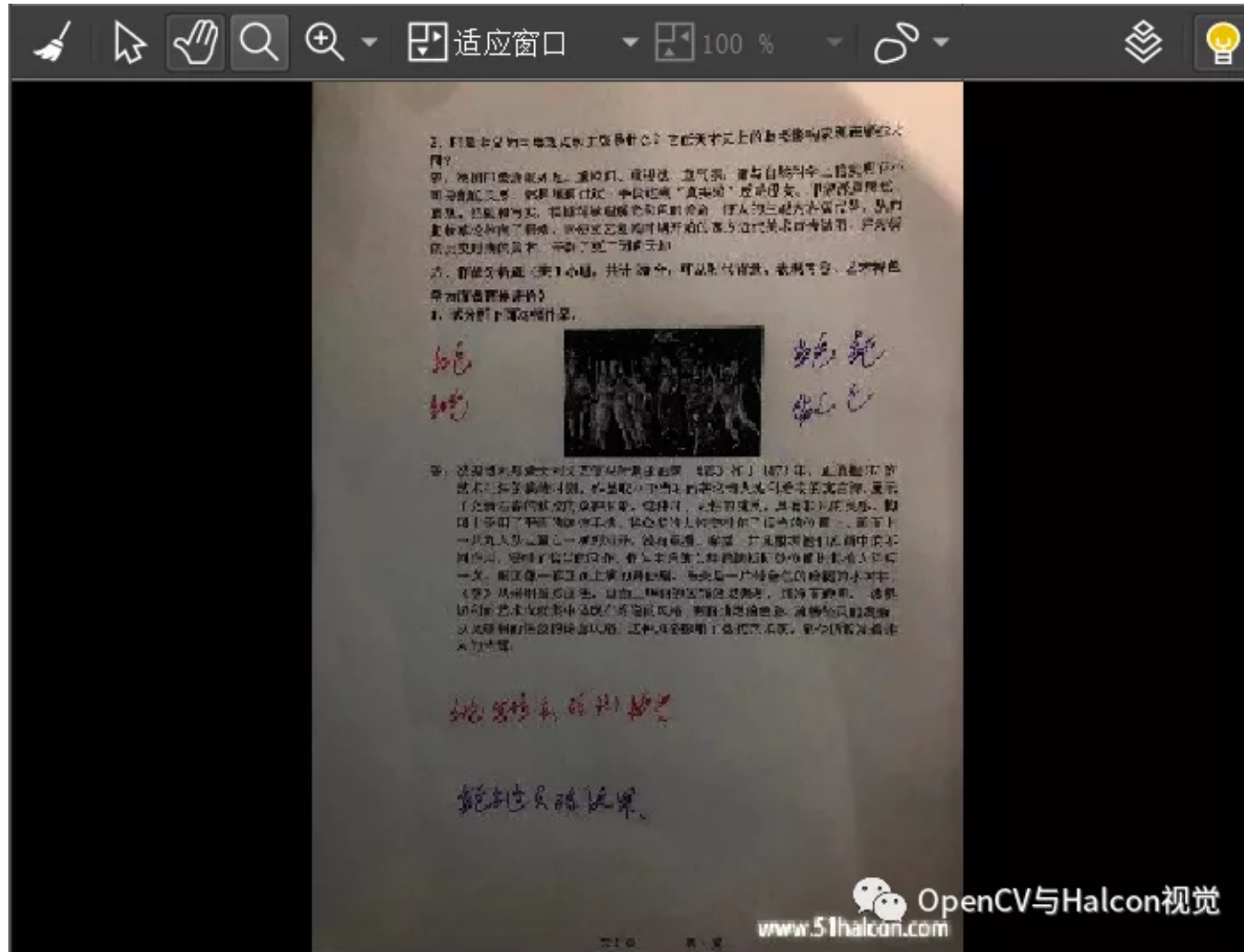
參考回帖內容，將代碼整理如下：

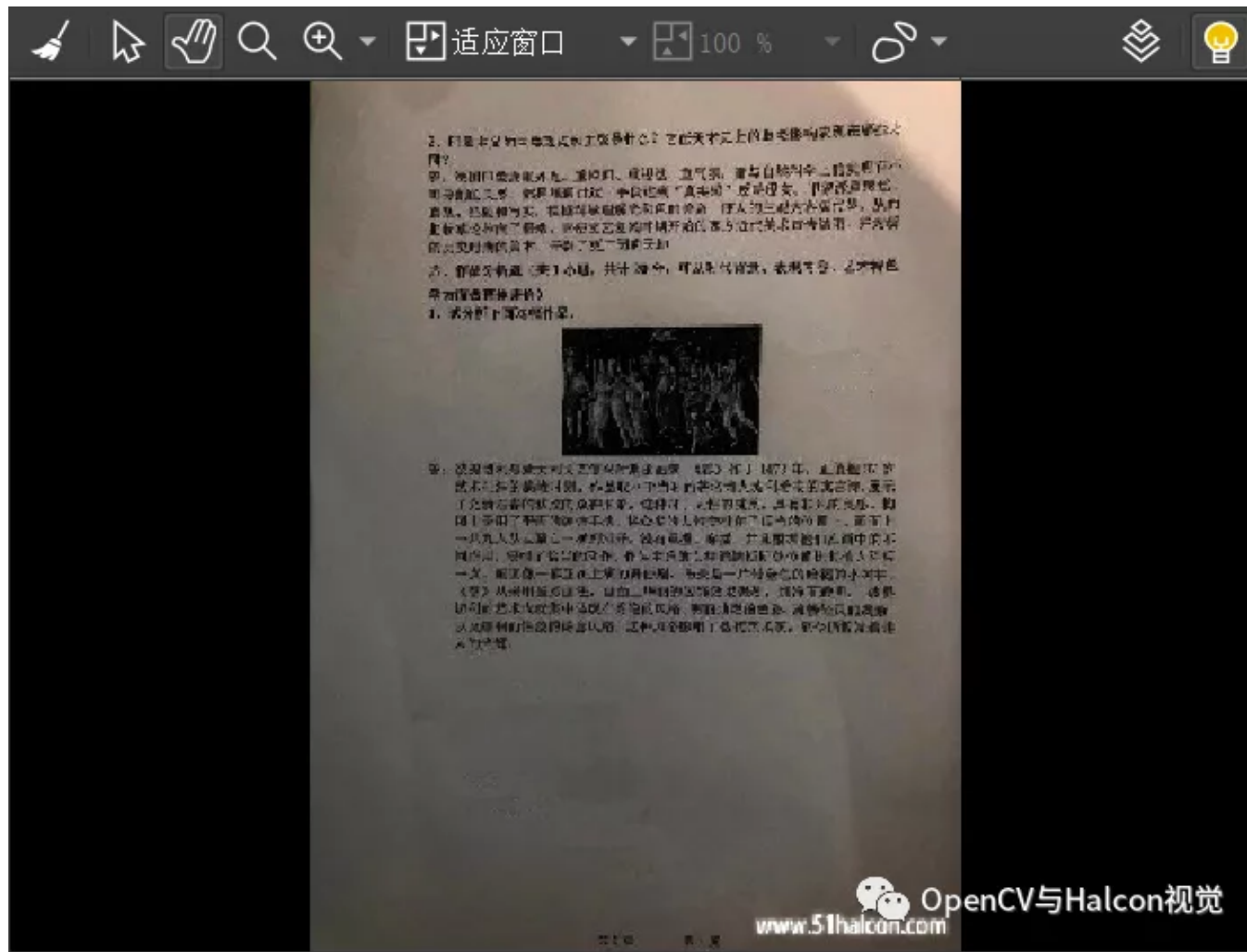
```
1 read_image (Image, './test.jpg')
2 get_image_size (Image, Width, Height)
3 decompose3 (Image, ImageR, ImageG, ImageB)
4 trans_from_rgb (ImageR, ImageG, ImageB, ImageY, ImageI, ImageQ, 'yiq')
```

```

5 threshold (ImageQ, Regions, 132, 255)
6
7 dilation_rectangle1 (Regions, RegionDilation, 25, 15)
8 inpainting_texture (Image, RegionDilation, InpaintedImage, 5, 20, 0, 'none', 0)

```





如上圖所示，可以較好的去除圖中紅色和藍色的筆跡。

OpenCV實現

分析實現方法與思路：

[1] 將原圖從BGR顏色空間轉為YIQ顏色空間；

[2] 分離Q通道圖像對其做二值化處理；

[3] 矩形核做膨脹操作，將筆跡區域連接

[4] 將膨脹後的圖像作為mask，傳給inpaint函數，通過圖像修復來去除筆跡。

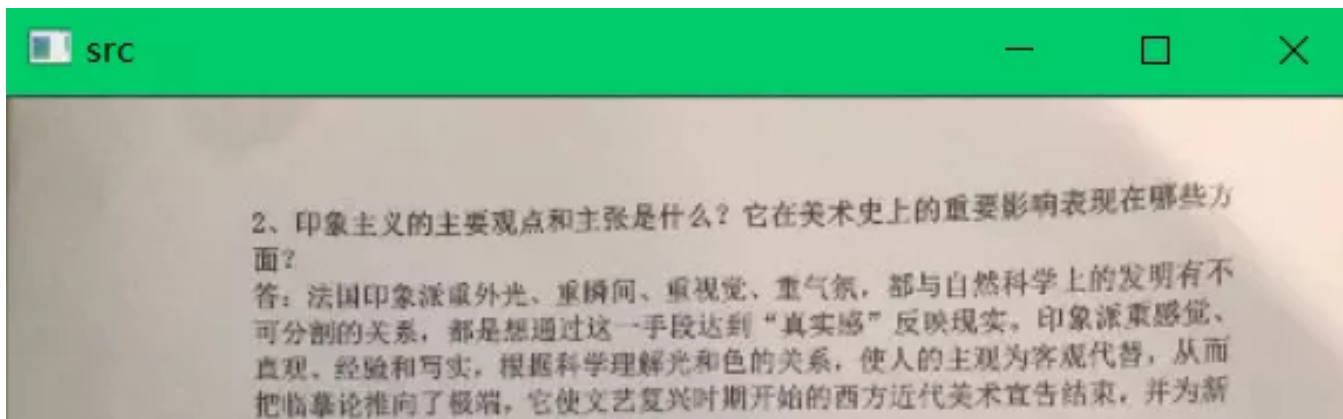
實現代碼(Python-OpenCV)：

```
1  #公众号：OpenCV与Halcon视觉
2  import cv2
3  import numpy as np
4
5  img = cv2.imread('test.jpg')
6  cv2.imshow('src',img)
7
8  img_q = np.zeros((img.shape[0],img.shape[1],1),np.float32)
9
10 for i in range(0,img.shape[0]):
11     for j in range(0,img.shape[1]):
12         B = img[i,j,0]
13         G = img[i,j,1]
14         R = img[i,j,2]
15         img_q[i,j] = 0.214*R - 0.522*G + 0.309*B
16
17 cv2.normalize(img_q, img_q, 0, 255, cv2.NORM_MINMAX, -1)
18
19 imgQ = cv2.convertScaleAbs(img_q)
20 cv2.imshow('imgQ',imgQ)
21
```

```
22 ret, threshImg= cv2.threshold(imgQ, 165, 255, cv2.THRESH_BINARY)
23 cv2.imshow('thresh',threshImg)
24
25 k1=np.ones((20,30), np.uint8)#指定膨脹核大小
26 threshImg = cv2.dilate(threshImg, k1, iterations=1)
27 cv2.imshow('dilation', threshImg)
28
29 result = cv2.inpaint(img,threshImg,3,cv2.INPAINT_NS)
30 cv2.imshow('result', result)
31
32
33 cv2.waitKey(0)
34 cv2.destroyAllWindows()
```

逐步效果演示

原圖：src



的历史时期的美术，开辟了更广阔的天地

六、作品分析题（共1小题，共计20分，可从时代背景、表现内容、艺术特色等方面做整体评价）

1、试分析下面这幅作品。

绝色
绝色



绝色
绝色

答：波提切利是意大利文艺复兴时期的画家，《春》作于1478年，正值他37岁艺术生涯的巅峰时期。作品取材于当时的著名诗人彼利希安的寓言诗，展示了充满着春的欢欣的众神形象，这种对于人性的赞美，具有非凡的关感。构图上采用了平面的装饰手法，将众多的人物安排在了适当的位置上，画面上一共九人从左至右一横列排开，没有重叠、穿插，并且根据他们在画中的不同作用，安排了恰当的动作。作为主角的女神维纳斯所处位置比其他人稍后一点。画面像一幕正在上演的舞台剧，布景是一片带金色的暗褐的小树林。《春》从采用蛋彩画法，画面上华丽的装饰效果强烈，纯净而透明。波提切利的艺术成就集中体现在秀逸的风格、明丽灿烂的色彩、流畅轻灵的线条，以及细腻而恬淡的诗意风格，这种风格影响了数代艺术家，至今仍散发着迷人的光辉。

绝色佳人陈润敏

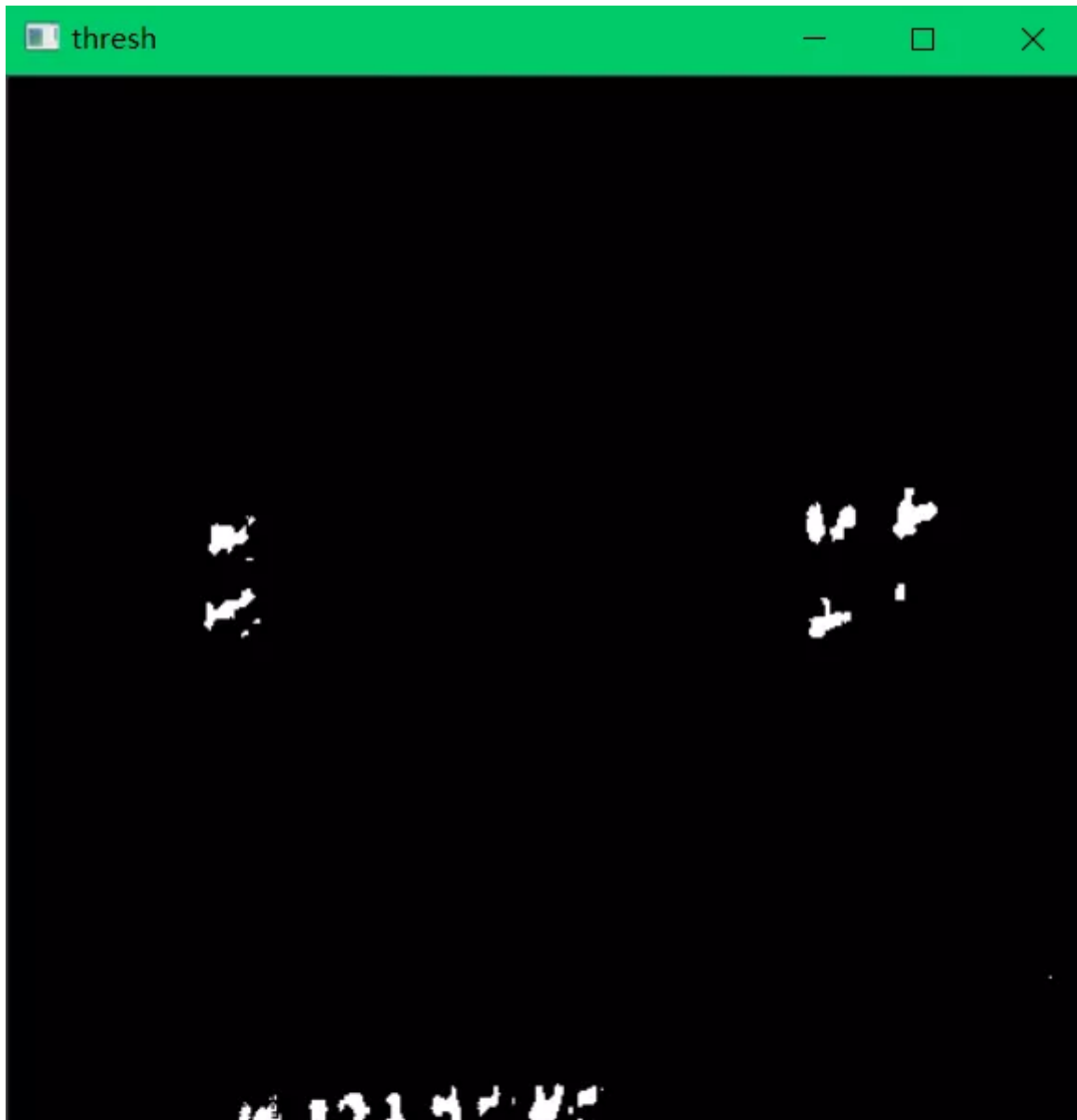
绝色佳人陈润敏

Q通道轉8位顯示：imgQ



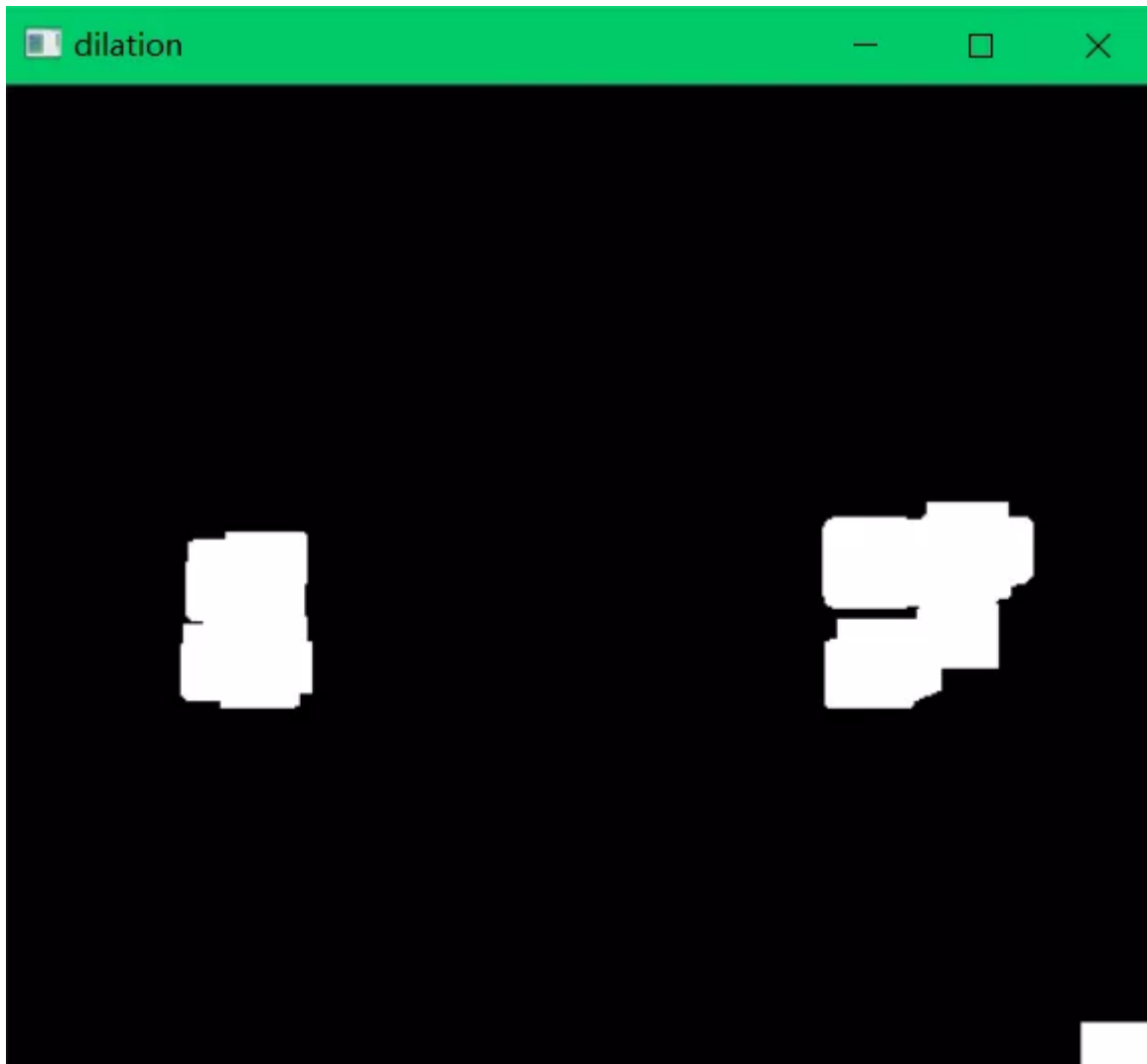


二值化效果：thresImg



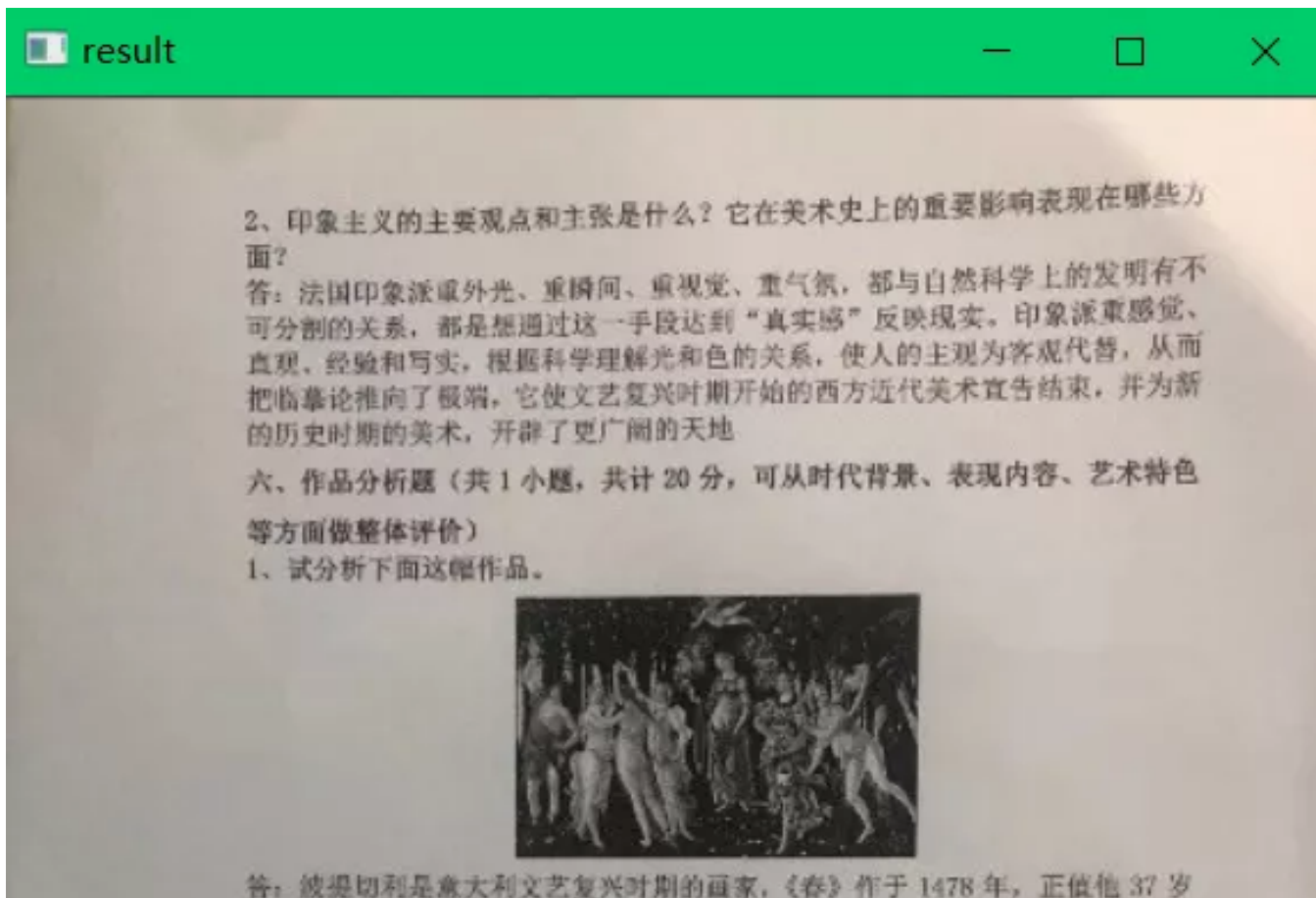


膨脹效果：dilation






inpaint修復效果：result



艺术生涯的巅峰时期。作品取材于当时的著名诗人波利希安的寓言诗，展示了充满着春的欢欣的众神形象，这种对于人性的赞美，具有非凡的美感。构图上采用了平面的装饰手法，将众多的人物安排在了适当的位置上。画面上共九人从左至右一横列排开，没有重叠、穿插，并且根据他们在画中的不同作用，安排了恰当的动作。作为主角的女神维纳斯所处位置比其他人稍后一点。画面像一幕正在上演的舞台剧，布景是一片带金色的暗褐的小树林。《春》从采用蛋彩画法，画面上华丽的装饰效果强烈，纯净而透明。波提切利的艺术成就集中体现在秀逸的风格、明丽灿烂的色彩、流畅轻灵的线条，以及细腻而恬淡的诗意风格，这种风格影响了数代艺术家，至今仍散发着迷人的光辉。

 OpenCV与Halcon视觉


結尾語

[1] 提取笔迹的方法有很多种，本文使用的是YIQ颜色空间的Q通道图像(直接对单个通道阈值处理较简单)，大家也可以使用其他颜色空间提取红色和蓝色即可(可能需要使用inRange函数在三个通道处理)；

[2] RGB颜色空间与YIQ颜色空间的转换：

① 网上流行的转换公式：

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

 OpenCV与Halcon视觉

② Halcon算子帮助文档中的转换公式：

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.599 & -0.276 & -0.324 \\ 0.214 & -0.522 & 0.309 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

 OpenCV与Halcon视觉

二者略有差异，本文使用的是Halcon帮助文档中的转换方法，直接计算得到Q通道图像。

完整代码(Python C++实现)与素材将发布在知识星球中，有问题可以留言或在星球中提问。

相关文章连接：

[Halcon缺陷检测实例转OpenCV实现\(一\)](#)

[Halcon缺陷检测实例转OpenCV实现\(二\)](#)

[Halcon缺陷检测实例转OpenCV实现\(三\)](#)

[Halcon缺陷检测实例转OpenCV实现\(四\)](#)

[Halcon缺陷检测实例转OpenCV实现\(五\) 混合颜色药片缺陷检测](#)

[Halcon缺陷检测实例转OpenCV实现\(六\) 瓶口缺陷检测](#)

[Halcon转OpenCV实例--复杂背景下缺陷检测\(附源码\)](#)

更多视觉图像处理相关内容，请长按关注：[OpenCV与AI深度学习](#)。



OpenCV与AI深度学习

致力于OpenCV、Halcon、人工智能、深度学习相关内容的学习、交流、分享！

106篇原创内容



公众号

觉得有用，麻烦给个赞和在看 

喜欢此内容的人还喜欢

SpringBoot 如何统一后端返回格式？老鸟们都是这样玩的！

JAVA日知錄



一門藝術、說說幾個我偶爾會用到的調試方法| Linux 内核

老吳的嵌入式之旅



C++11 thread join 和detach 有什麼區別？

字節流動

