

Subscribe to DeepL Pro to edit this document.  
Visit [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)for more information.

**Opencv based circular marker detection report**

**Department of Automation, Tsinghua University**

**Name: Wang Yuqing**

**Date: June 18, 2021**

**Catalog**

**[1. Test Questions 2](#_Toc19822)**

**[2. Operating Environment 2](#_Toc23407)**

**[3. Testing algorithms 2](#_Toc16529)**

**[4. Experimental results 4](#_Toc20968)**

**[5. Problems encountered and ideas for solving them 9](#_Toc20760)**

**[6. Analysis and Conclusion 10](#_Toc6558)**

# Test Questions

Identify circular markers in uploaded images and mark them with the OpenCV library.

# Operating Environment

Windows 2010 operating system, PyCharm2021

# Testing algorithms

|  |
| --- |
| import numpy as np  import cv2 as cv  import datetime  def filter\_out(src\_frame):  if src\_frame is not None:  hsv = cv.cvtColor(src\_frame, cv.COLOR\_BGR2HSV)  lower\_red = np.array([120, 43, 46])  upper\_red = np.array([180, 255, 255])  # lower\_green= np.array([35, 43, 46])  # upper\_green= np.array([77, 255, 255])  # inRange() method returns a matrix containing only 0,255 (CV\_8U) 0 means not in the interval  mask = cv.inRange(hsv, lower\_red, upper\_red)  return cv.bitwise\_and(src\_frame, src\_frame, mask=mask)  def GammaCorrection(img,c=1.0,g=2.2):  # Gamma correction  out = (np.power(img/255, 1.0/g)/c)\*255  return out.astype(np.uint8)  def readimage(img):  #img = GammaCorrection(img)  hue\_image = cv.cvtColor(img, cv.COLOR\_BGR2HSV)  low\_range = np.array([0, 43, 46])  high\_range = np.array([180, 255, 255])  th = cv.inRange(hue\_image, low\_range, high\_range)  dilated = cv.dilate(  th, cv.getStructuringElement(  cv.MORPH\_ELLIPSE, (2, 2)), iterations=2)  circles = cv.HoughCircles(  # dp: the inverse of the accumulator resolution to the image resolution. the larger the dp get, the smaller the accumulator array.  # minDist: the minimum distance from the center of the circle.  # param1: gradient of edge detection.  # param2: accumulator threshold, the smaller the circle the more.  dilated,  cv.HOUGH\_GRADIENT,  0.8,  70,  param1=50,  param2=16,  minRadius=0,  maxRadius=100)  # print(circles[0][0])  print(len(circles[0]))  if circles is not None:  for i in range(len(circles[0])):  x, y, radius = circles[0][i]  print(x, y, radius)  center = (int(x), int(y))  cv.circle(img, center, int(radius), (0, 255, 255), 2)  return img  if \_\_name\_\_ == "\_\_main\_\_":  starttime = datetime.datetime.now()  img = cv.imread('redpoint1.jpg')  a, b = img.shape[0:2]  img = cv.cvtColor(img, cv.COLOR\_BGR2RGB)  frame = filter\_out(img)  frame = cv.resize(frame, (int(b / 5), int(a / 5)))  cv.imwrite('redpoint01.jpg', frame)  #cv.imshow("capture", frame)  #cv.waitKey(0)  output = cv.imread('redpoint01.jpg')  output = cv.cvtColor(output, cv.COLOR\_BGR2RGB)  output = readimage(output)  cv.imshow('r.png', output)  k = cv.waitKey(0)  cv.imwrite('red1.jpg', output)  endtime = datetime.datetime.now()  print((endtime - starttime)\*100) |

# Experimental results

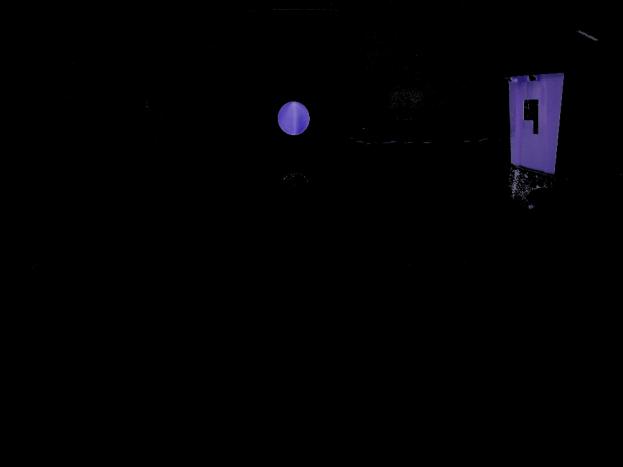
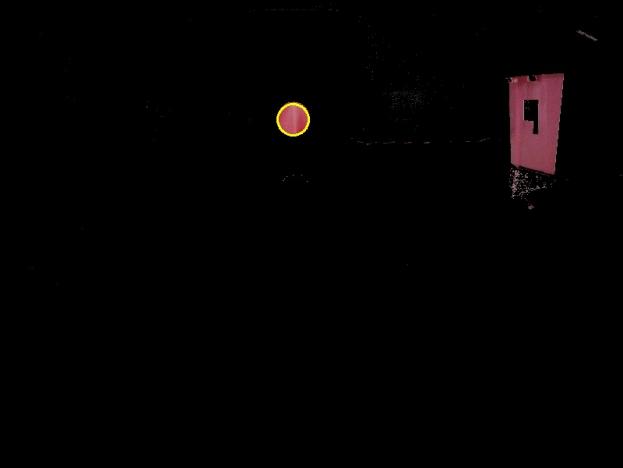
 

Figure 1 Original image of red circular marker 1 Figure 2 Original image of red circular marker 1 after extraction of red



-Red circular marker 1 Hoff circle detection parameters

dp = 0.8,

minDisk = 70,

param1=50,

param2=16,

minRadius=0,

maxRadius=100

Figure 3 Red circular marker detection



Fig. 4 Original image of red circular marker 2 Fig. 5 Red circular marker 2 after extraction of red



-Red circular marker 2 Hoff circle detection parameters

dp = 0.8,

minDisk = 70,

param1=50,

param2=16,

minRadius=0,

maxRadius=100

Figure 6 Red circular marker 2 detection



Figure 7 Original red circular marker 3 Figure 8 Original red circular marker 3 after extraction of red

-Red circular marker 3 Hoff circle detection coefficient

dp = 0.8,

minDisk = 70,

param1=50,

param2=16,

minRadius=0,

maxRadius=100

Figure 9 Red circular marker 3 detection

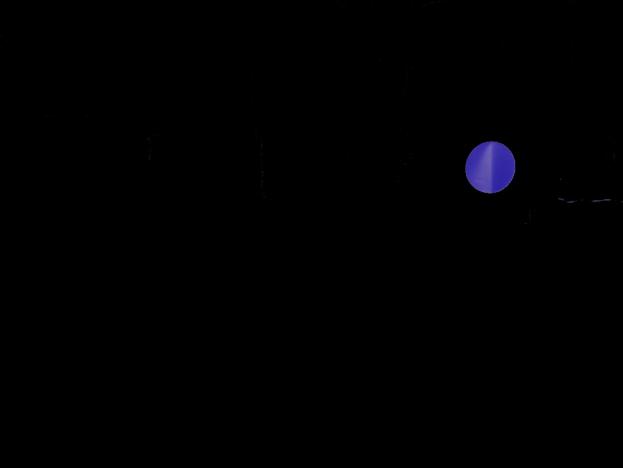
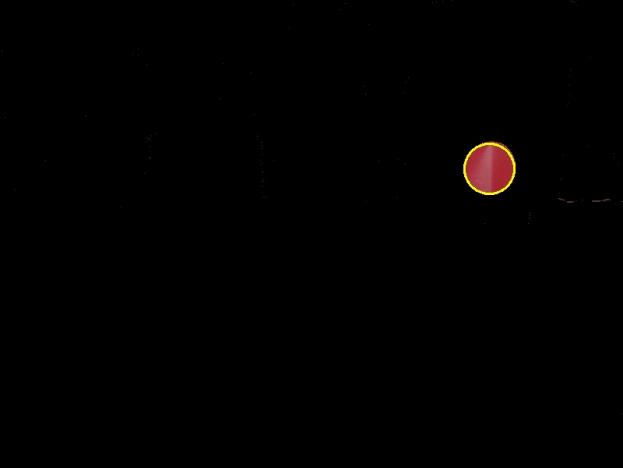


Figure 10 Original image of red circular marker 4 Figure 11 Red circular marker 4 after extraction of red



-Red circular marker 4 Hoff circle detection coefficient

dp = 0.8,

mindisk = 70,

param1=50,

param2=16,

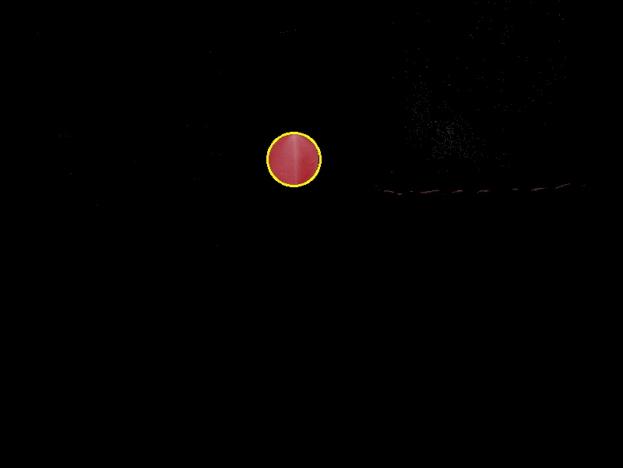
minRadius=0,

maxRadius=100

Figure 12 Red circular marker 4 detection



Figure 13 Original image of red circular marker 5 Figure 14 Red circular marker 5 after extraction of red



-Red circular marker 5 Hoff circle detection coefficient dp = 0.8,

minDisk = 70,

param1 = 50,

param2 = 16,

minRadius = 0,

maxRadius = 100

Figure 15 Red circular marker 5 detection



Figure 16 Original image of green circular marker 1 Figure 17 Green circular marker 1 after green extraction



-Green circular marker 1 Hoff circle detection coefficient

dp = 0.8,

minDisk = 70,

param1 = 50,

param2 = 16,

minRadius = 0,

maxRadiu s= 100)

Figure 18 Green circular marker 1 detection

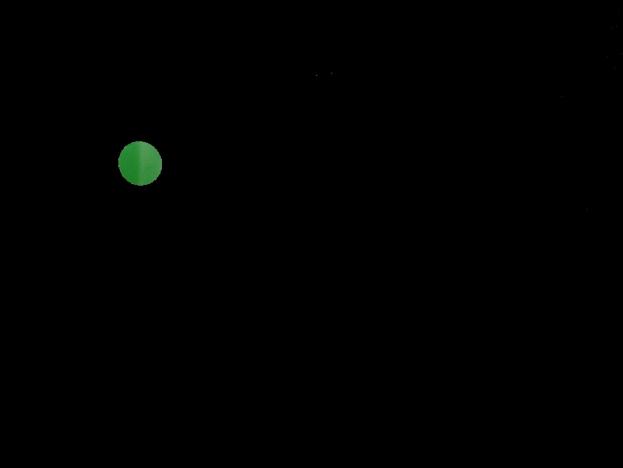
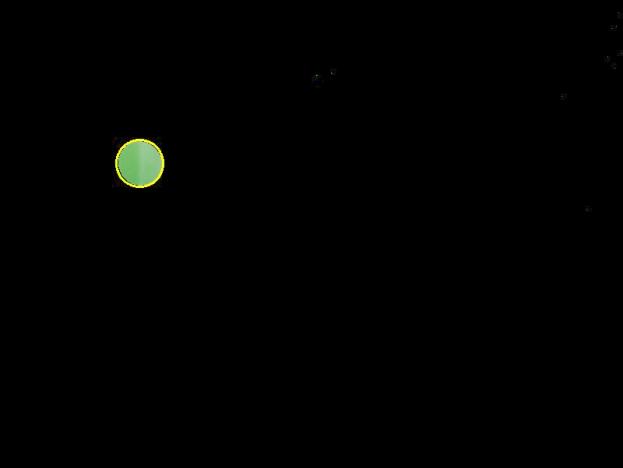


Figure 19 Original image of green circular marker 2 Figure 20 Green circular marker 2 after green extraction



-Green circular marker 2 Hoff circle detection coefficient

dp = 0.8,

minDisk = 70,

param1 = 50,

param2 = 16,

minRadius =0,

maxRadius =100)

Figure 21 Green circular marker 2 detection

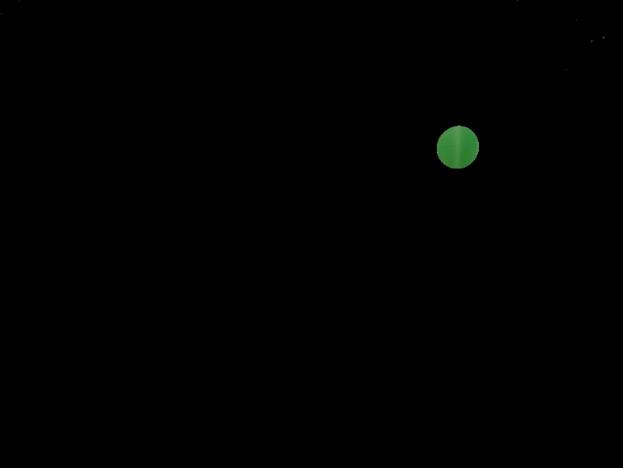


Figure 22 Original image of green circular marker 3 Figure 23 Green circular marker 3 after green extraction



-Green circular marker 3 Hoff circle detection coefficient

dp = 0.8,

minDisk = 70,

param1 = 50,

param2 = 16,

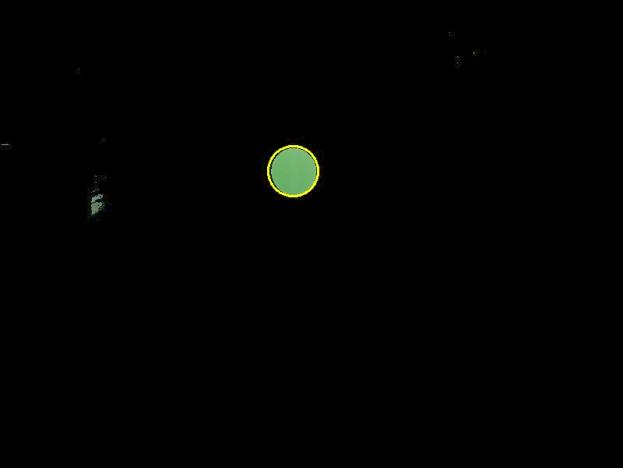
minRadius =0,

maxRadius =100)

Figure 2 4 Green circular marker 3 detection



Figure 25 Original image of green circular marker 4 Figure 26 Green circular marker 4 after green extraction



-Green circular marker 4 Hoff circle detection coefficient

dp = 0.8,

minDisk = 70,

param1 = 50,

param2 = 16,

minRadius =0,

maxRadius =100)

Figure 27 Green circular marker 4 detection

# Problems encountered and ideas for solving them

Problem: Low accuracy when identifying circular markers directly from the original image.

Solution: The image is segmented according to the range of the marker color in the HSV color space. Pixels belonging to the marker color are kept (become white) and the rest of the pixels are set to black to reduce the interference of other colors on the marker recognition.

# Analysis and Conclusion

The algorithm can accurately identify and label nine images of circular markers with different colors, angles and sizes. And the Hoff circle coefficients are the same for each image, only different color markers need different color removal coefficients. It is clear that this algorithm has a wide range of adaptability for circular markers. The algorithm processes 100 images in about 40 seconds.