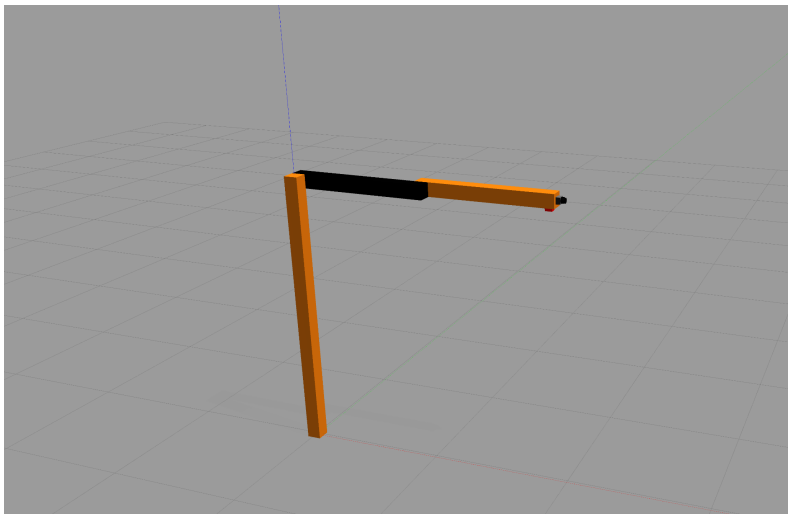
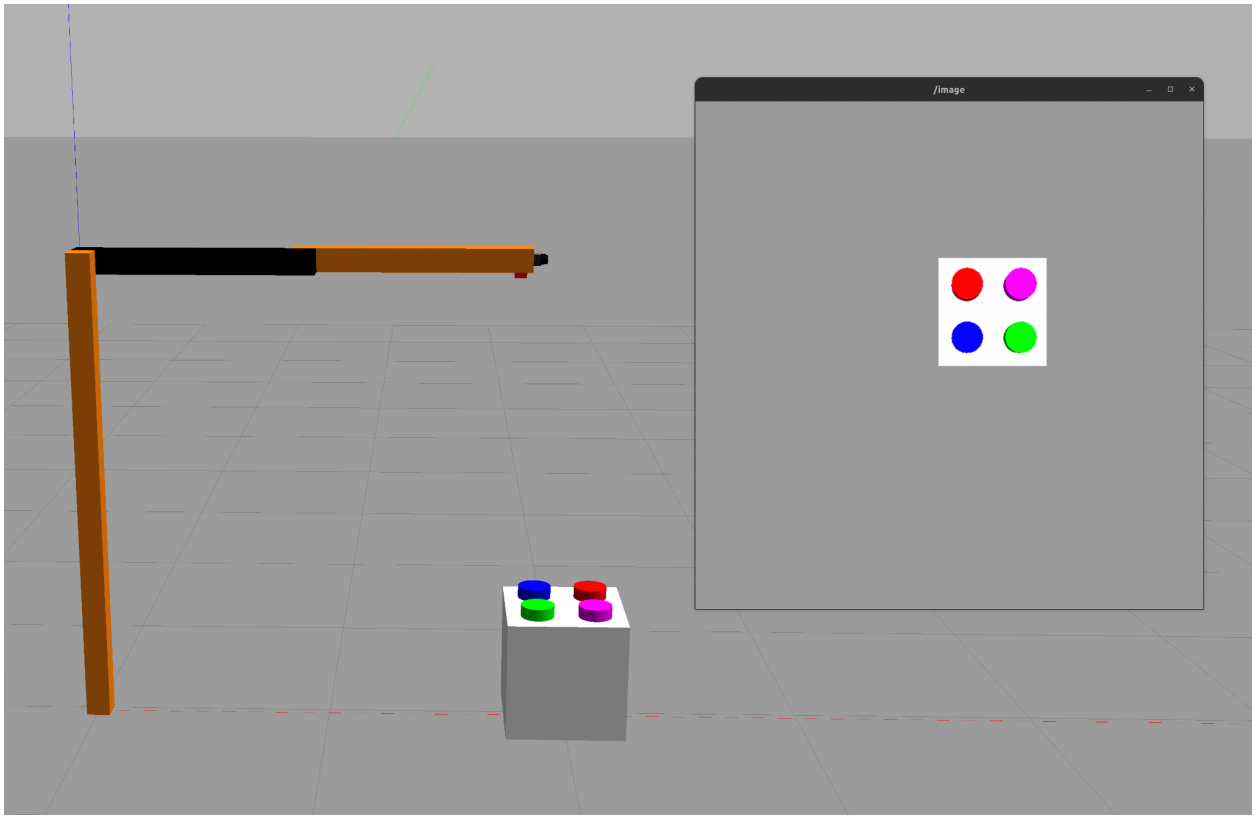
Step 1

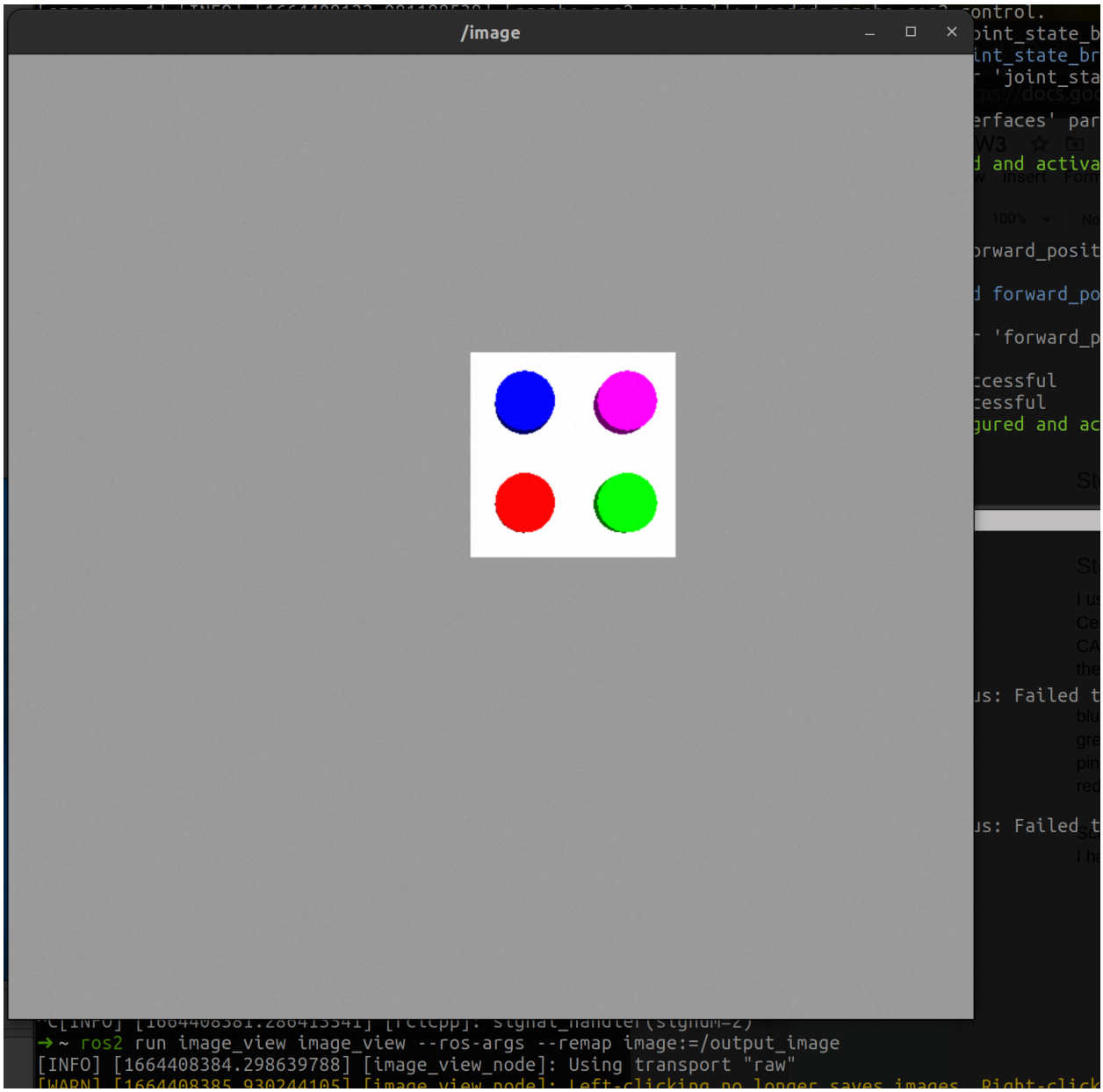


Step 2



Step3

Below is the output from running test.py



Step 4

I used HSV space to segment color.
Centers I got from thresholding
CAUTION : the centers I got are from opencv convention. To access the array we need to invert the x and y

blue:(427.4, 287.9)
green:(511.1, 371.6)

pink:(511.0, 288.0)
red:(427.7, 371.3)

Screenshot of the code is next page
I have included line numbers in the screenshot so that code can be referred from test1.py

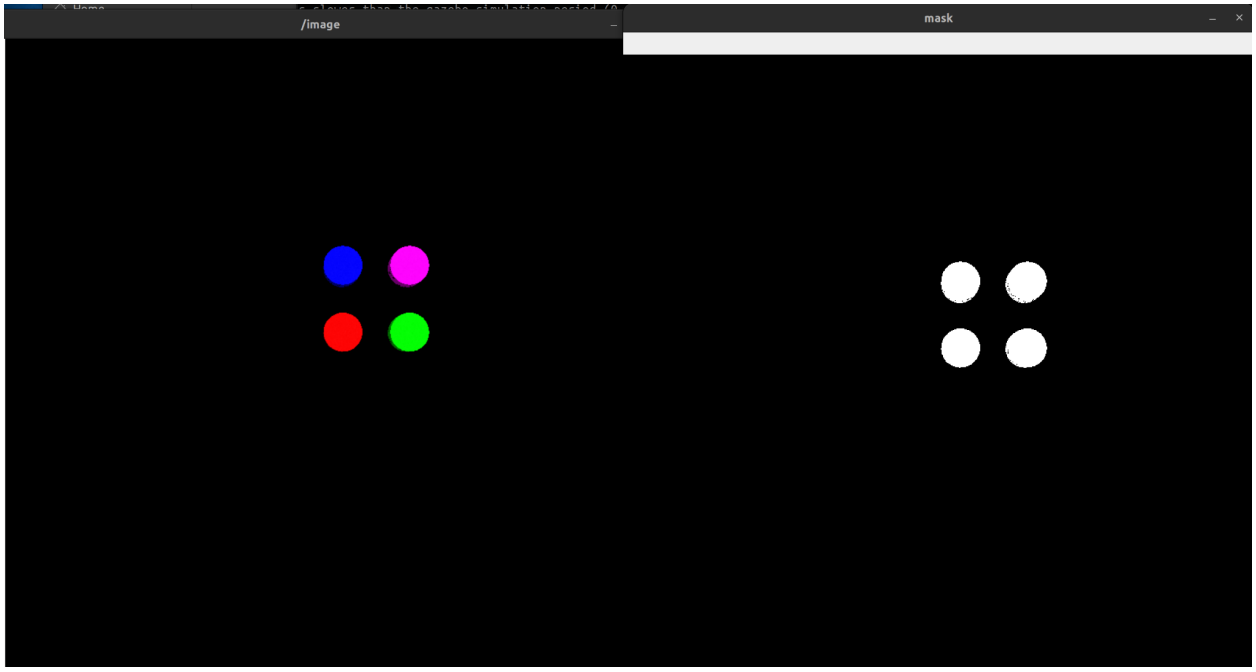
```python
import numpy as np

def get_center_of_mask( mask):
    x_i = 0;
    y_i = 0;
    count = 0;
    coords = np.argwhere(mask)
    count = coords.shape[0]
    coords = np.sum(coords,axis=0)
    if count > 0:
        coords = coords/count
        return coords

    return (0,0)
```

test1.py

```python
    # PLACE YOUR CODE HERE. PROCESS THE CURRENT FRAME AND PUBLISH IT
IONS
    #cv2.imshow("output_image", current_frame)
    #cv2.waitKey(1)
    # read once and save
    #cv2.imwrite("output_image.png",current_frame)
    #exit()
    blue_low = (100,100,100)
    blue_high = (120,255,255)

    green_low = (50,100,100)
    green_high = (100,255,255)

    red_low = (0,100,100)
    red_high = (50,255,255)

    pink_low = (145,100,100)
    pink_high = (150,255,255)


    hsv = cv2.cvtColor(current_frame,cv2.COLOR_BGR2HSV)
    blue_mask = cv2.inRange(hsv,blue_low,blue_high)
    red_mask = cv2.inRange(hsv,red_low,red_high)
    green_mask = cv2.inRange(hsv,green_low,green_high)
    pink_mask = cv2.inRange(hsv,pink_low,pink_high)

    final_mask = blue_mask + red_mask + green_mask + pink_mask
    cv2.imshow("output_image", final_mask)

    x_blue,y_blue =  get_center_of_mask(blue_mask)
    x_green,y_green =  get_center_of_mask(green_mask)
    x_pink,y_pink =  get_center_of_mask(pink_mask)
    x_red,y_red =  get_center_of_mask(red_mask)
    print(f"blue:{x_blue,y_blue}")
    print(f"green:{x_green,y_green}")
    print(f"pink:{x_pink,y_pink}")
    print(f"red:{x_red,y_red}")
```

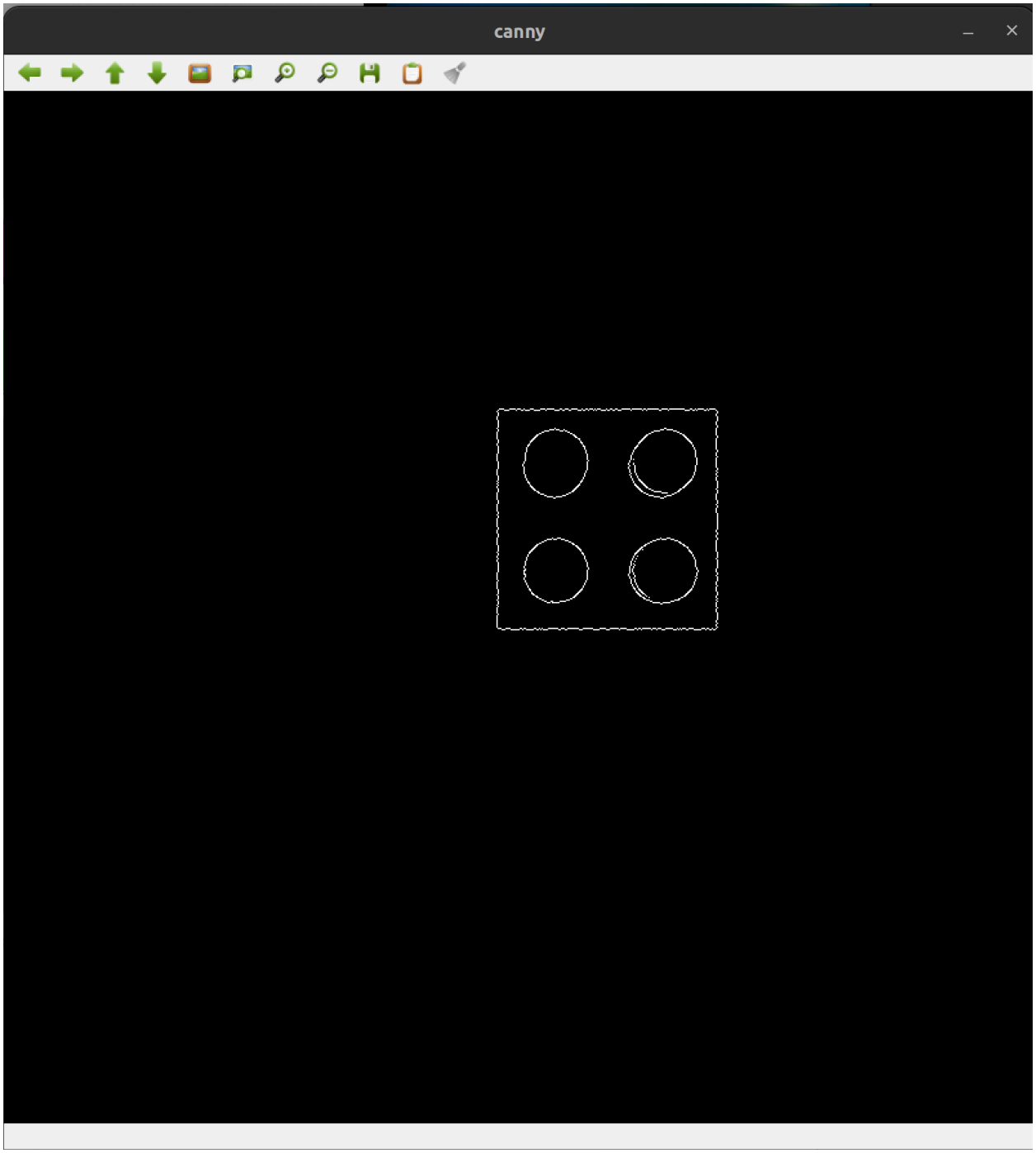
test1.py

Below is the segmented color and mask



Step 5

Below is the output canny edge detection

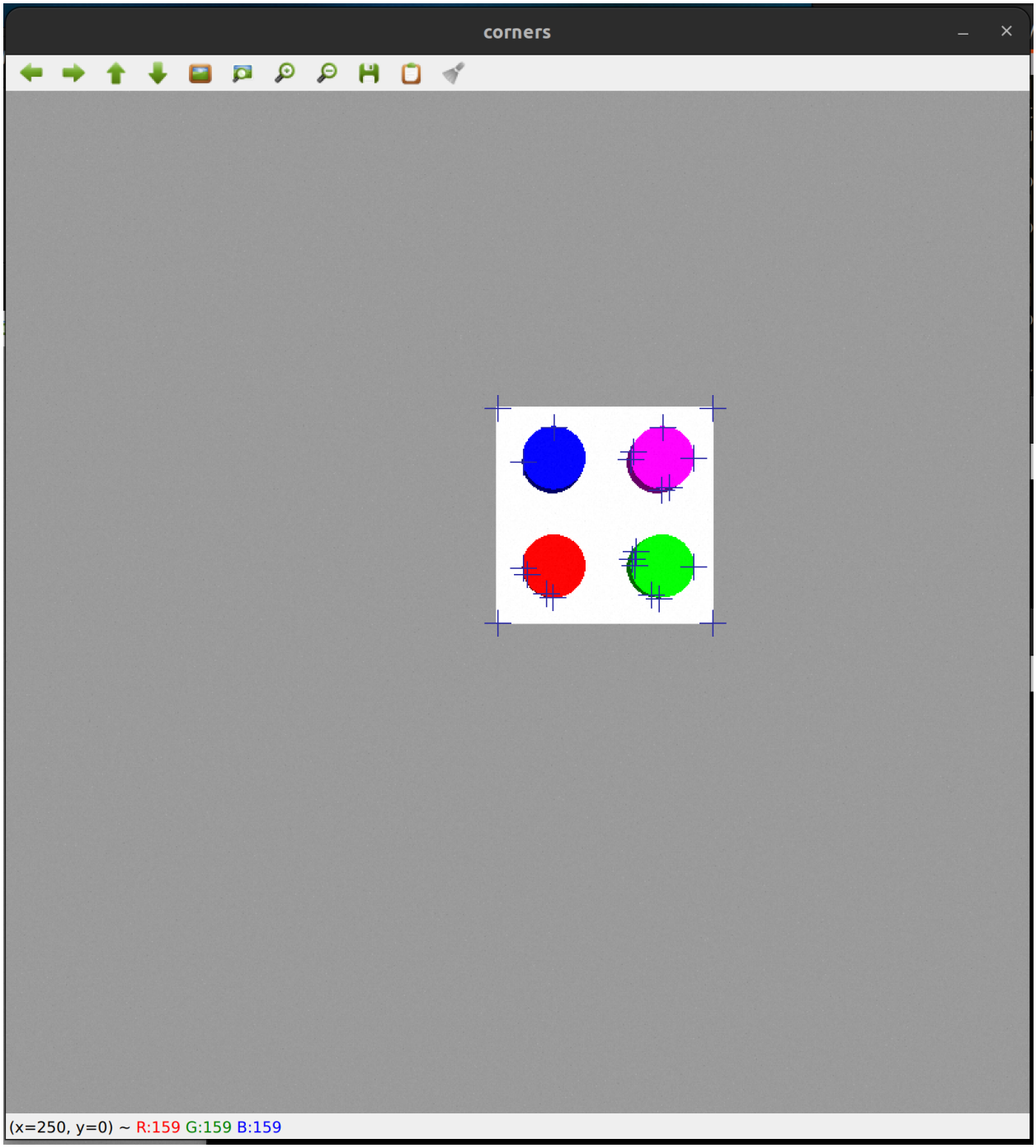


Code:

```
111
112     # get gray scale image
113     gray = cv2.cvtColor(current_fram, cv2.COLOR_BGR2GRAY)
114
115     ## Canny Edge detection
116     canny = cv2.Canny(gray,200,300)
117     cv2.imshow("canny", canny)
118     cv2.waitKey(1)
119 nny edge detection
120
```

Step 6

Below is the output from corner harris.
Dark Blue cross marks are the local maxima corners obtained after thresholding corner harris scores



Below is the code for the same

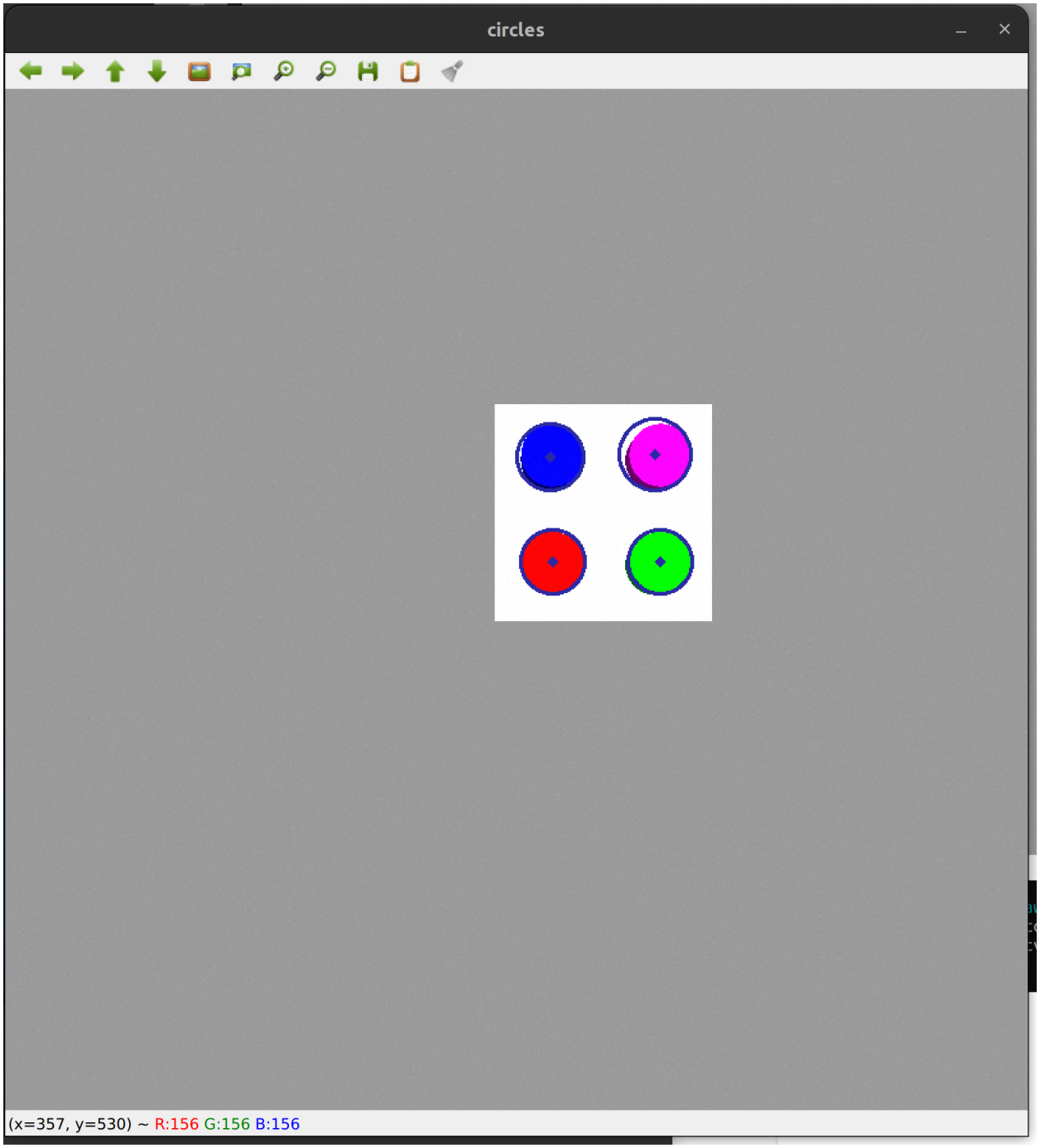
```
120
121     ## corners
122     # apply harris corners
123     img_corners = cv2.cornerHarris(gray,4,7,0.04)
124
125     # find local maximas in the corner score image
126     coords = peak_local_max(img_corners,min_distance=3,threshold_abs=0.01)
127
128     # draw markers in the image
129     for coord in coords:
130         cv2.drawMarker(current_frame, [coord[1],coord[0]], color=[165,42,42])
131
132
```

Step 7

Centers from hough circles algo. As before CAUTION that below numbers are in opencv coordinate convention
blue center:(426, 288)
green center:(512, 370)
pink center:(508, 286)
red center:(428, 370)

They are very close to the centers calculated from Step 4

below is the output from hough circles algorithm
As before, Dark Blue cross marks and Dark Blue circles indicate the output



Below is the code for the same

```
131
132
133     # hough circles
134     circles = cv2.HoughCircles(gray,
135                                cv2.HOUGH_GRADIENT,1,20,
136                                param1=100,
137                                param2=30,
138                                minRadius=0,
139                                maxRadius=0)
140
141     circles = np.uint16(np.around(circles))
142     img_circles = current_frame.copy()
143     for i in circles[0,:]:
144         cv2.circle(img_circles,(i[0],i[1]),i[2],(165,42,42),2)
145         cv2.circle(img_circles,(i[0],i[1]),2,(165,42,42),3)
146     #blue center:(426, 288)
147     #red center:(428, 370)
148     #pink center:(508, 286)
149     #green center:(512, 370)
150
151
152     cv2.imshow("circles",img_circles)
153
```