Chapter11: Raspberry Pi color recognition

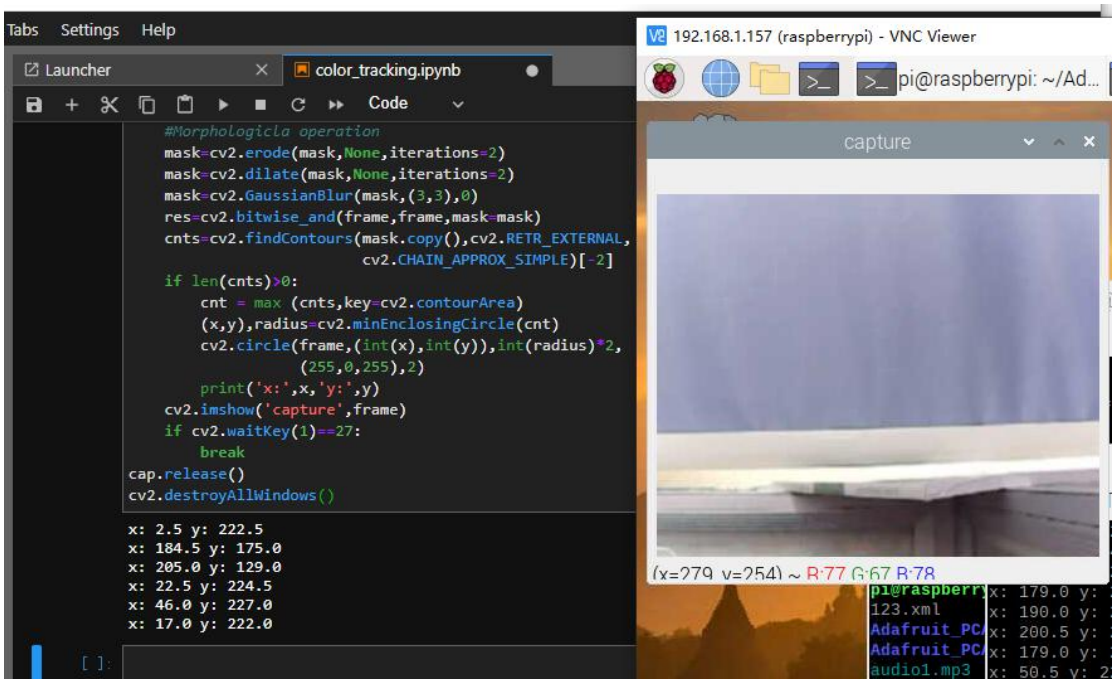
！ Note: When running the program of this course, there must be a desktop for displaying pictures. It is recommended that you use VNC to log in to the system so that the pictures can be displayed.

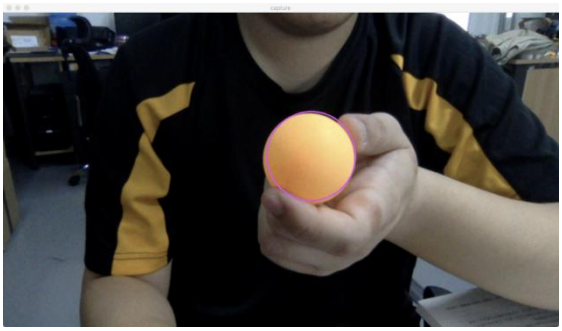
In this lesson , we will test object is a yellow table tennis. We know that the yellow value is ([156,43,46], [180,255,255]),
First, we should convert RGB to hsv by cv2.cvtColor(frame,cv2.COLOR_BGR2HSV),
Then, we can construct a mask according to the threshold value. After morphological treatment by expansion corrosion, perform the bitwise operation of the mask and the original image. When the color is found, draw a circle on the outline of the color to mark it.

The source code of the program is located at:
/home/pi/yahboom/color_tracking/color_tracking
The code as shown in the figure below.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
* @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
* @file         basic_writeAndrRead
* @version      V1.0
* @details
* @par History
* @author       LongfuSun
"""

from __future__ import division
import cv2
import time
import numpy as np

cap=cv2.VideoCapture(0)

#Set the camera ressolution to (640, 480)
#If you feel that the image is stuck, you can reduce it to (320, 240)
cap.set(3,480)
cap.set(4,320)

#Set yellow value
yellow_lower=np.array([26,43,46])
yellow_upper=np.array([34,255,255])

time.sleep(1)

while 1:
    #Ret is whether to find the image, frame is the frame itself
    ret,frame=cap.read()

    frame=cv2.GaussianBlur(frame,(5,5),0)
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv,yellow_lower,yellow_upper)

    #Morphologicla operation
    mask=cv2.erode(mask,None,iterations=2)
    mask=cv2.dilate(mask,None,iterations=2)
    mask=cv2.GaussianBlur(mask,(3,3),0)
    res=cv2.bitwise_and(frame,frame,mask=mask)
    cnts=cv2.findContours(mask.copy(),cv2.RETR_EXTERNAL,
                          cv2.CHAIN_APPROX_SIMPLE)[-2]
    if len(cnts)>0:
        cnt = max (cnts,key=cv2.contourArea)
        (x,y),radius=cv2.minEnclosingCircle(cnt)
        cv2.circle(frame,(int(x),int(y)),int(radius)*2,
                   (255,0,255),2)
        print('x:',x,'y:',y)
    cv2.imshow('capture',frame)
    if cv2.waitKey(1)==27:
        break
cap.release()
cv2.destroyAllWindows()

```

The result is as shown in the figure below. A blue contour with real-time changes is generated in the contour line of yellow table tennis.





Then, we can click [Kernel]-[Restart Kernel and Clear All Outputs] to end this process and clear the output results.