

```

import cv2
import numpy as np

#single camera, first camera index
cap = cv2.VideoCapture(0)

#width for our frame with id 3
cap.set(3,640)

#height for our frame with id 4
cap.set(4,480)

#brightness for id 10
cap.set(10,150)


#list of colors of maximum and minimum hue and saturation values
mycolors=[[5,107,0,19,255,255],
          [133,56,0,159,156,255],
          [57,76,0,100,255,255]]


#define all colors bgr values
mycolorvalue=[[51,153,255],
              [255,0,255],
              [0,255,0]]


#create a list of points for x y and index of color
mypoints=[] ##[x, y, colorId]


#input img, mycolors, mycolorvalue
#function to find a color
def findcolor(img,mycolors,mycolorvalue):
    imgHSV=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)

    #counter to count how many times, if count is 0 frst color nd so on
    count=0

    #define new point everytime

```

```

newpoint=[]
#for loop to detect all 3 colors
for color in mycolors:
    # lower limit of mycolors
    lower=np.array(color[0:3])
    #upper limit of mycolors
    upper=np.array(color[3:6])
    mask=cv2.inRange(imghsv,lower,upper)
    #send mask to draw contours
    #get center from contours ie center of bounding box
    x,y= getContours(mask)
    #draw circle tip for imgresult with center point x y and
    #radius 10 and color it detects
    cv2.circle(imgresult,(x,y),10,mycolorvalue[count],cv2.FILLED)
    #if x and y are zero dont draw
    if x!=0 and y!=0:
        newpoint.append([x,y,count])
    count+=1
    #to test for 3 different colors, 3 different windows
    #cv2.imshow(str(color[0]),mask)
return newpoint

```

```

#to get object and define contours nd bounding box
def getContours(img):
    contours,hierarchy=cv2.findContours(img,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
    #declare as 0 if nothing is returned
    x,y,w,h=0,0,0,0
    for cnt in contours:
        area=cv2.contourArea(cnt)
        #cv2.drawContours(imgresult,cnt,-1,(255,0,0),3)
        peri=cv2.arcLength(cnt,True)

```

```

    approx = cv2.approxPolyDP(cnt, 0.02*peri,True)
    x, y, w, h =cv2.boundingRect(approx)
    #to draw from tip(center) of pen
    return x+w//2, y

```

```

def drawOnCanvas(mypoints,mycolorvalue):

```

```

    for point in mypoints:
        #point0 is x point1 is y and colorid is 2
        cv2.circle(imgresult, (point[0],point[1]), 10, mycolorvalue[point[2]], cv2.FILLED)

```

```

while True:

```

```

    #using while we r getting image
    success, img = cap.read()
    #create a new image for final information
    imgresult=img.copy()
    newpoint=findcolor(img,mycolors,mycolorvalue)
    #check if new point is there or not
    if len(newpoint)!=0:
        #iterate through new points bcz it is list
        for newp in newpoint:
            mypoints.append(newp)
    #draw if len is not 0
    if len(mypoints)!=0:
        drawOnCanvas(mypoints,mycolorvalue)
    #displaying image
    cv2.imshow("video", imgresult)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

```