*import* cv2  
*import* numpy *as* np  
  
cap = cv2.VideoCapture(0)  
  
*def* nothing(x):  
 *pass*cv2.namedWindow("Tracking")  
cv2.createTrackbar('LH', 'Tracking', 0, 255, nothing) # LH(lower hue) nothing function is just for formality.  
cv2.createTrackbar('LS', 'Tracking', 0, 255, nothing) #LS(lower saturation)  
cv2.createTrackbar('LV', 'Tracking', 0, 255, nothing) #LV(lower value)  
cv2.createTrackbar('UH', 'Tracking', 255, 255, nothing) #UH(upper hue)  
cv2.createTrackbar('US', 'Tracking', 255, 255, nothing) #US(upper saturation)  
cv2.createTrackbar('UV', 'Tracking', 255, 255, nothing) #UV(upper value) 255,255 becaue it is upper value.  
  
*while* cap.isOpened():  
 \_, frame = cap.read()  
 # gray=cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
  
 hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV) # converting image into hsv colours.  
  
 l\_h = cv2.getTrackbarPos('LH', "Tracking")  
 l\_s = cv2.getTrackbarPos('LS', "Tracking")  
 l\_v = cv2.getTrackbarPos('LV', "Tracking")  
  
 u\_h = cv2.getTrackbarPos('UH', "Tracking")  
 u\_s = cv2.getTrackbarPos('US', "Tracking")  
 u\_v = cv2.getTrackbarPos('UV', "Tracking")  
  
 l\_c = np.array([l\_h, l\_s, l\_v]) # l\_c(lower\_colour) which we want to detect.  
 u\_c = np.array([u\_h, u\_s, u\_v])  
  
 mask = cv2.inRange(hsv, l\_c, u\_c) # it is our a certain type window with is help to detect colour using bitwise operator.  
  
 img = cv2.bitwise\_and(frame, frame, mask=mask)  
  
 gray=cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
  
 \_, thresh = cv2.threshold(gray, 80, 255, cv2.THRESH\_BINARY)  
 # dilated = cv2.dilate(thresh, None, iterations=3)  
 # contours, \_ = cv2.findContours(thresh, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  
  
 #cv2.drawContours(frame, contours, -1, (0,255,0), 2)  
  
 # for i in contours:  
 # (x, y, w, h) = cv2.boundingRect(i)  
 # cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)  
  
 cv2.imshow('mask', mask)  
 cv2.imshow('frame', frame)  
  
 *if* cv2.waitKey(1) == ord('q'):  
 *break*cap.release()  
cv2.destroyAllWindows()