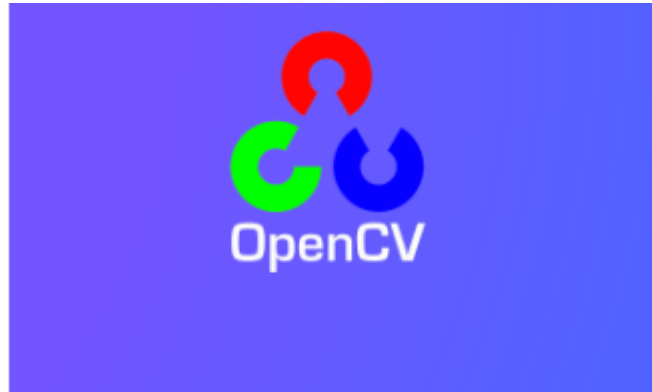


# Getting started with Opencv CV

---

## 1.Introduction to Open Source CV

---



What is OpenCV? Its full name is Open source Computer Vision Library, open source computer vision library. As shown in the picture above, what we see is the OpenCV logo. We can see that it consists of three small rings in the three distinct primary colors of R, G, and B. In other words, it is a set of open source API function libraries about computer vision. This also means,

- (1) Whether it is scientific research or commercial application, it can be used for development;
- (2) The source code of all API functions is public, and you can see the program steps of its internal implementation;
- (3) You can modify the source code of OpenCV and compile and generate the specific API functions you need.

Image processing on ROSMASTER uses certain functions of the OpenCV function library, or it can be said that it is inseparable from its existence in most image processing design fields. As early as many years ago, OpenCV has been showing its talents in the fields of intrusion detection, specific target tracking, target detection, face detection, face recognition, face tracking, etc., and these are just the tip of the iceberg of its applications. Since we realize that OpenCV is so versatile, in this chapter we will introduce you to some very basic image processing functions that we use in our courses, as well as some universal functions. Here we first have a general understanding of this knowledge, and then there are two practical projects on color recognition and tracking, and face recognition and tracking to teach you how to get started. However, the powerful application functions provided by OpenCV are far more than this. If you are interested in OpenCV computer vision library development and want to learn more about it, here are several websites for your reference and study:

OpenCV Official homepage :<https://www.opencv.org>

OpenCV Chinese forum:<http://www.opencv.org.cn>

OpenCV CSDN forum :<https://bbs.csdn.net/forums/OpenCV>

## 2.Image reading and display

---

## 2.1. Reading of images:

`img = cv2.imread('yahboom.jpg', 0)` The first parameter is the path of the image, and the second parameter is how to read the image.

`cv2.IMREAD_UNCHANGED`: Keep the original format unchanged, `-1`;

`cv2.IMREAD_GRAYSCALE`: Read the image in grayscale mode, which can be represented by `0`;

`cv2.IMREAD_COLOR`: , read a color picture, which can be represented by `1`; default value

`cv2.IMREAD_UNCHANGED`: Read in an image and include its alpha channel, which can be represented by `2`.

## 2.2. Image display

`cv.imshow('frame', frame)`: Open a window named `frame` and display `frame` data (image/video data)

Parameter meaning:

The first parameter represents the name of the window that is created and opened.

The second parameter represents the image to be displayed

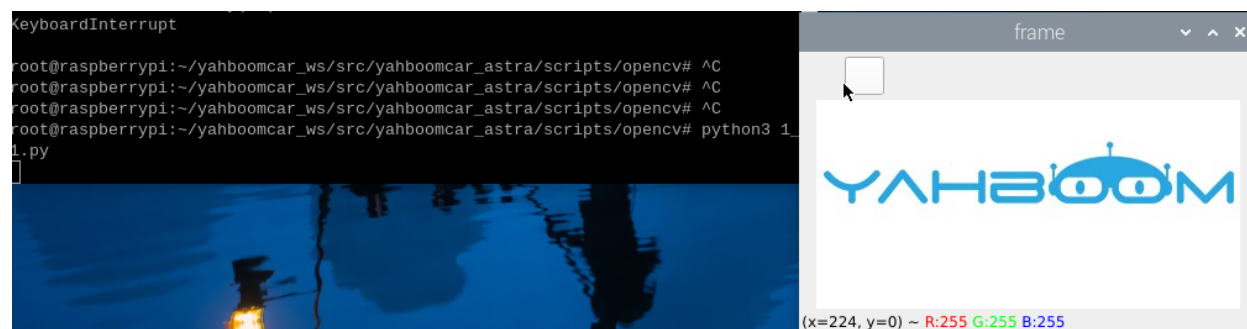
## 2.3. Code and actual effect display

run code

```
cd ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/  
python3 1_1.py
```

```
import cv2 as cv  
.  
if __name__ == '__main__':  
    img = cv.imread('yahboom.jpg')  
    while True :  
        cv.imshow("frame",img)  
        action = cv.waitKey(10) & 0xFF  
        if action == ord('q') or action == 113:  
            break  
    img.release()  
    cv.destroyAllWindows()
```

Rendering after running the program:



## 3、OpenCV image writing

### 3.1、Function method: cv2.imwrite('new\_img\_name', img)

Parameter meaning:

The first parameter is the saved file name

The second parameter is the saved image

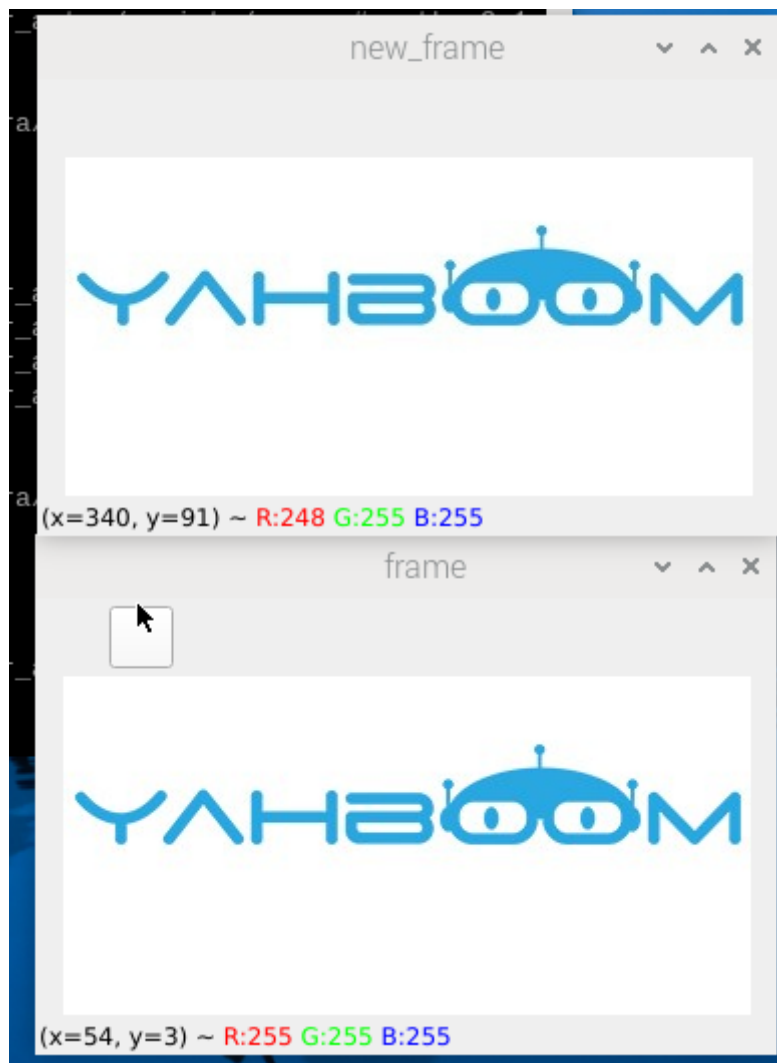
### 3.2、Code and actual effect display

Run program

```
cd ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/  
python3 1_2.py
```

```
import cv2 as cv  
.  
if __name__ == '__main__':  
    img = cv.imread('yahboom.jpg')  
    cv.imwrite("yahboom_new.jpg",img)    #Create a new file yahboom_new.jpg and write  
    yahboom.jpg into it  
    new_img = cv.imread('yahboom_new.jpg') #Read newly written pictures  
    while True :  
        cv.imshow("frame",img)  
        cv.imshow("new_frame",new_img)  
        action = cv.waitKey(10) & 0xFF  
        if action == ord('q') or action == 113:  
            break  
    img.release()  
    cv.destroyAllWindows()
```

Effect picture after running the program



## 4、 OpenCV camera reading and displaying video

### 4.1、 Camera reading

```
capture=cv.VideoCapture(0)
```

Parameter meaning:

The parameter in VideoCapture() is 0, which means opening the laptop's built-in camera. If the parameter is the video file path, the video will be opened, such as `cap = cv2.VideoCapture("../test.avi")`

### 4.2、 Show camera video

```
ret,img = frame.read()
```

Return value meaning:

ret: ret is a bool value to determine whether the correct frame is read back.

img: Image data for each frame

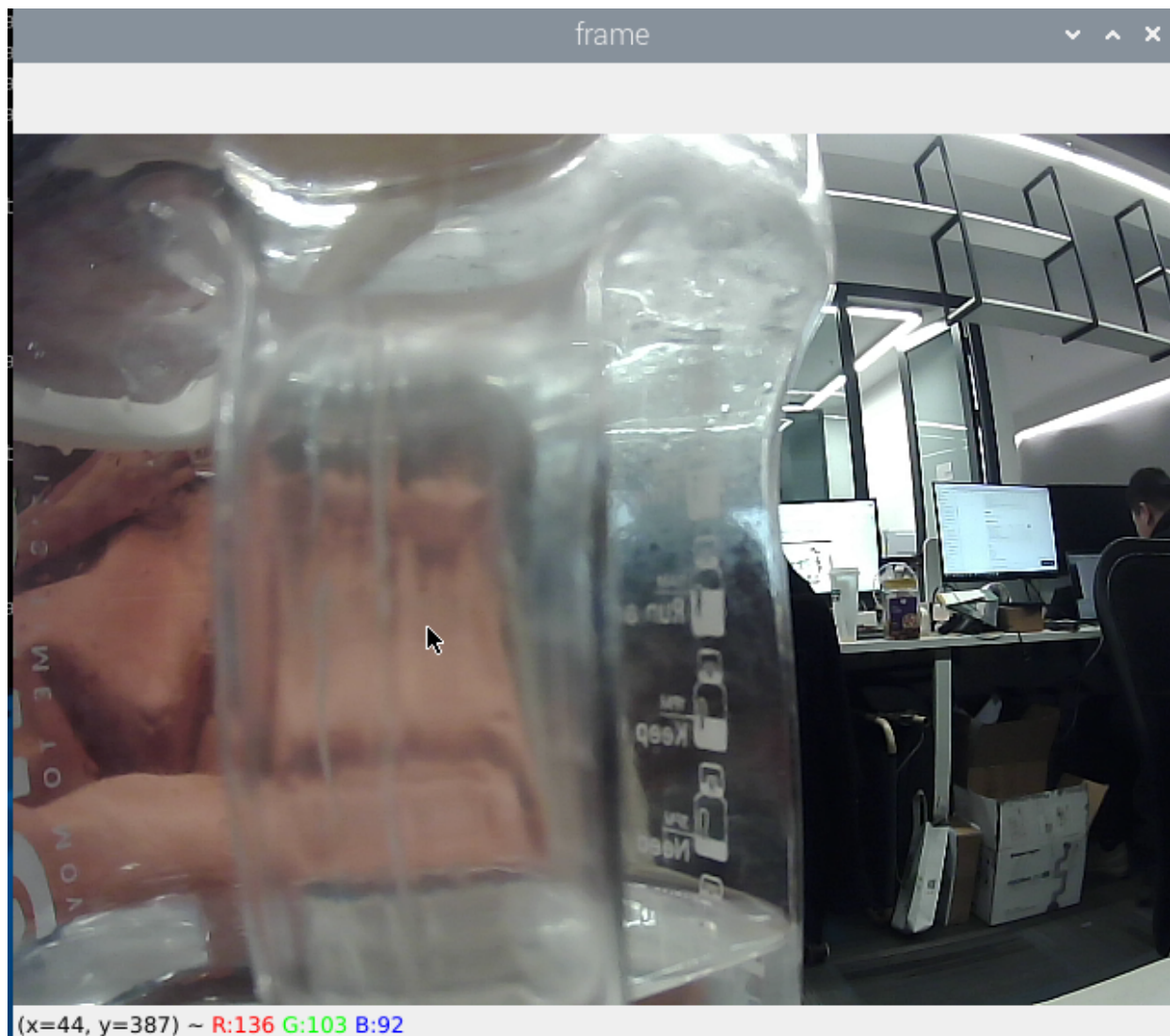
### 4.3、Code and actual effect display

Run program

```
cd ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/  
python3 1_3.py
```

```
import cv2 as cv  
if __name__ == '__main__':  
    frame = cv.VideoCapture(0)  
    while frame.isOpened():  
        ret,img = frame.read()  
        cv.imshow('frame',img)  
        action = cv.waitKey(10) & 0xFF  
        if action == ord('q') or action == 113:  
            break  
    frame.release()  
    cv.destroyAllWindows()
```

After the program is run, the camera image is displayed.



## 5、 OpenCV pixel operations

### 5.1、 Pixel operation, we can change any position to a new pixel color.

First, we need to read the image, then modify the value of bgr and assign an area to black.

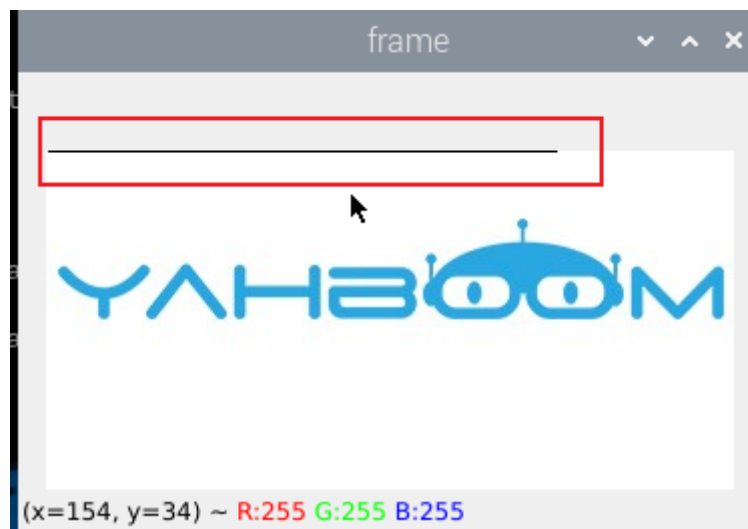
### 5.2、 Code and actual effect display

Run program in terminal

```
cd ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/  
python3 1_4.py
```

```
import cv2  
if __name__ == '__main__':  
    img = cv2.imread('yahboom.jpg')  
    (b,g,r) = img[100,100]  
    print(b,g,r)  
    i=j=0  
    for j in range(1,255):  
        img[i,j] = (0,0,0)  
        for j in range(1,255):  
            img[i,j] = (0,0,0)  
    while True :  
        cv2.imshow("frame",img)  
        action = cv2.waitKey(10) & 0xFF  
        if action == ord('q') or action == 113:  
            break  
    img.release()  
    cv2.destroyAllWindows()
```

Effect picture after running the program.



The red box part is the modified pigment value.

