

# 1. Getting started with Open Source CV

---

## 1. Getting started with Open Source CV

### 1.1. Introduction to OpenCV

### 1.2. OpenCV image reading and display

#### 1.2.1. Image reading:

#### 1.2.2. Image display

#### 1.2.3. Code and actual effect display

### 1.3. OpenCV image writing

#### 1.3.1. Function method: `cv2.imwrite('new_img_name', img)`

#### 1.3.2. Code and actual effect display

### 1.4. OpenCV camera to read and display video

#### 1.4.1. Camera reading

#### 1.4.2. Display camera video

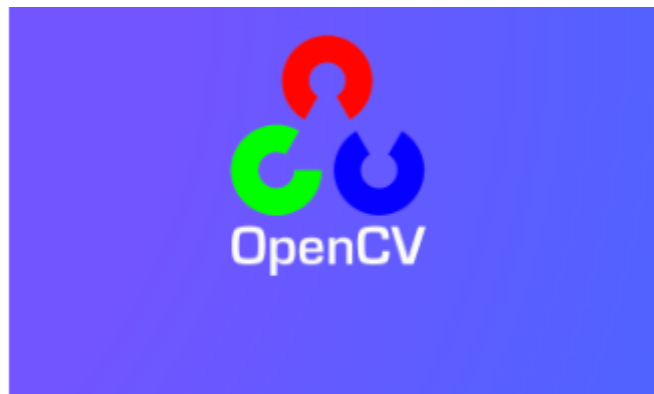
#### 1.4.3. Code and actual effect display

### 1.5. OpenCV pixel operation

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

#### 1.5.2. Code and actual effect display

## 1.1. Introduction to OpenCV



What is OpenCV? Its full name is Open source Computer Vision Library, an open source computer vision library. As shown in the figure above, what we see is the logo of OpenCV, and we can see three small circles with three distinct primary colors of R, G, and B. Composition, that is, it is a set of open source API function library 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 to compile and generate the specific API functions you need.

The image processing on ROSMASTER uses some functions of the OpenCV function library, or it can be said that its existence is inseparable in most image processing design fields. As early as many years ago, it has been used in intrusion detection and specific target tracking. , target detection, face detection, face recognition, face tracking and other fields, OpenCV can be said to show its talents, and these are only the tip of the iceberg of its application. Now that we realize that OpenCV is so general, in this chapter, we will introduce you to some very basic image

processing functions that we use in our courses, and also some general functions. Here we first have a general understanding of these knowledge. After a while, there are two practical projects in the back: color recognition and tracking, face recognition and tracking to teach you how to get started, but the powerful application functions provided by OpenCV are far more than that. If you are interested in the development of OpenCV computer vision library, you want to go deeper. If you understand, here are a few 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>

## 1.2. OpenCV image reading and display

### 1.2.1. Image reading:

`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 image, which can be represented by 1; the default value

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

### 1.2.2. Image display

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

Parameter meaning:

The first parameter is the name of the window to be created

The second parameter indicates the image to be displayed

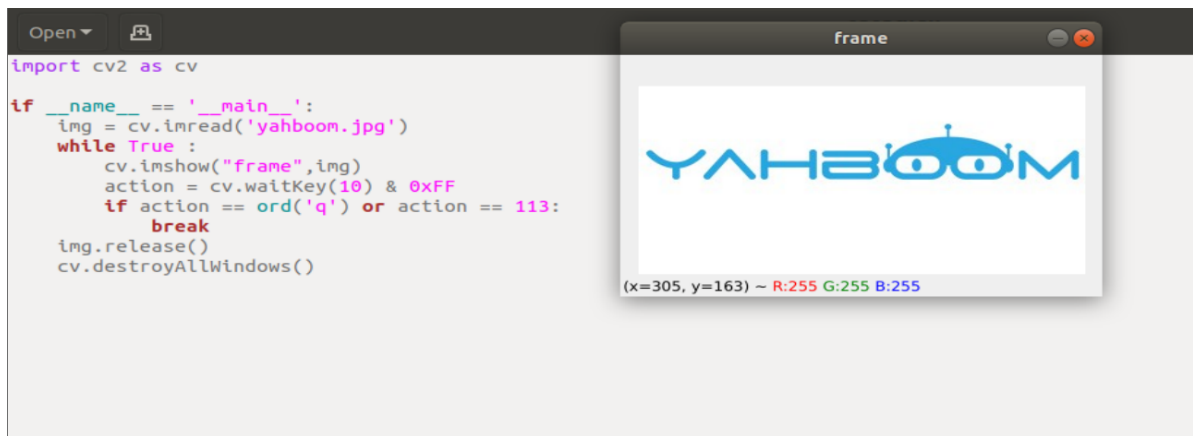
### 1.2.3. Code and actual effect display

run the program

```
cd ~/transbot_ws/src/transbot_visual/opencv
python 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()
```



## 1.3. OpenCV image writing

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

Parameter meaning:

The first parameter is the name of the file to save

The second parameter is the saved image

### 1.3.2. Code and actual effect display

run the program

```
cd ~/transbot_ws/src/transbot_visual/opencv
python 1_2.py
```

```
import cv2 as cv

if __name__ == '__main__':
    img = cv.imread('yahboom.jpg')
    cv.imwrite("yahboom_new.jpg",img)
    new_img = cv.imread('yahboom_new.jpg')
    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()
```

```
import cv2 as cv
if __name__ == '__main__':
    img = cv.imread('yahboom.jpg')
    cv.imwrite("yahboom_new.jpg",img)
    new_img = cv.imread('yahboom_new.jpg')
    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()
```



## 1.4. OpenCV camera to read and display video

### 1.4.1. Camera reading

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

Parameter meaning:

The parameter in VideoCapture() is 0, which means to open the built-in camera of the notebook, and the parameter is the video file path to open the video, such as cap = cv2.VideoCapture("../test.avi")

### 1.4.2. Display camera video

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

Return value meaning:

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

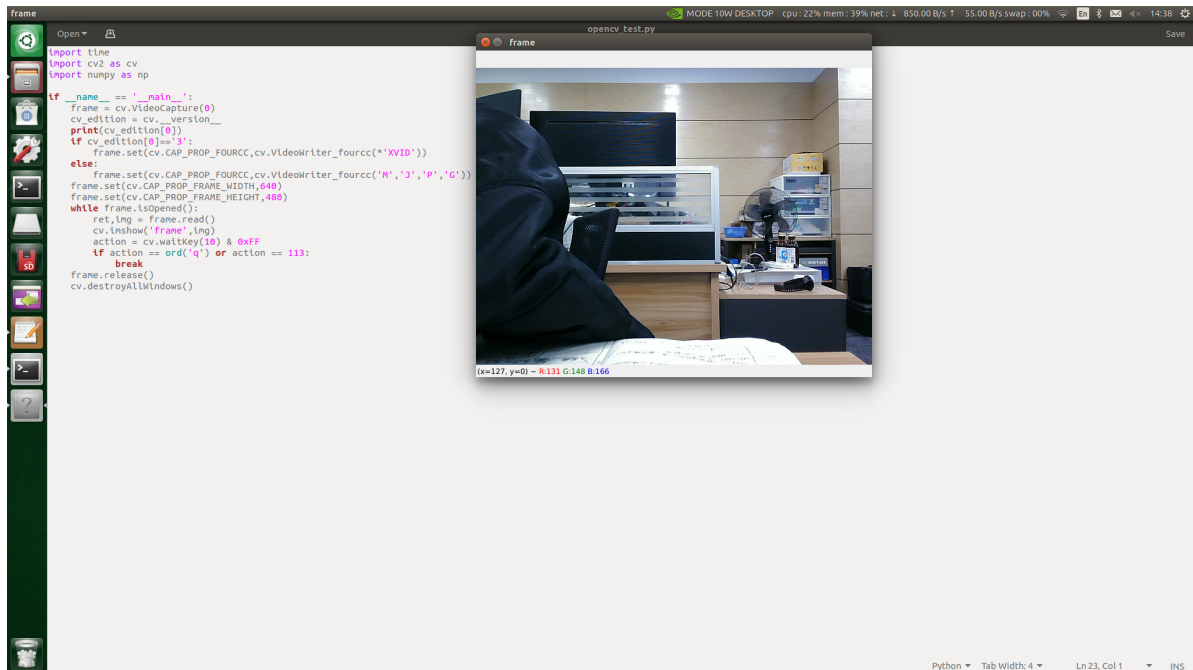
img: image data for each frame

### 1.4.3. Code and actual effect display

run the program

```
cd ~/transbot_ws/src/transbot_visual/opencv
python 1_3.py
```

```
import cv2 as cv
if __name__ == '__main__':
    frame = cv.VideoCapture(0)
    while frame.isopen():
        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()
```



## 1.5. OpenCV pixel operation

### 1.5.1. 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 be black.

### 1.5.2. Code and actual effect display

run the program

```
cd ~/transbot_ws/src/transbot_visual/opencv
python 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()

```

```

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()

```



The red box part is the modified color value.