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 read and display video
        - 1.4.1. camera read
        - 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 the pixel color at any position.
        - 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. As shown in the picture above, we can see the OpenCV logo, which is composed of three small circles of the three primary colors of R, G, and B. In other words, it is an open source API function library for computer vision. This means that

(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 implemented internally;

(3) You can modify the source code of OpenCV and compile it to generate the specific API function you need.

The image processing on ROSMASTER uses some functions of the OpenCV function library, or it can be said that it is indispensable in most image processing design fields. OpenCV has been used in intrusion detection, specific target tracking, target detection, face detection, face recognition, face tracking and other fields many years ago. And these are just the tip of the iceberg of its application. Since we know that OpenCV is so universal, in this chapter we will introduce you to some very basic image processing functions used in our course, which are also some universal functions. Here we first have a general understanding of these knowledge, and then there are two practical projects of 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 that. If you are very interested in the development of Opencv computer vision library and want to learn more, the following 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

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 an image and include 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 indicates the name of the window to be created

The second parameter indicates the image to be displayed

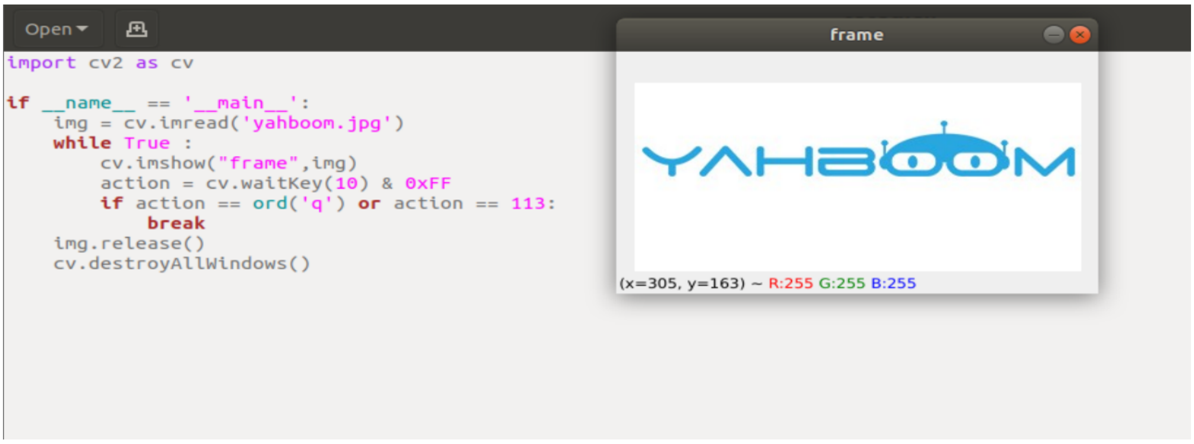
1.2.3, Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-
00001/src/yahboom_navrobo_astra/scripts/opencv/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()
```



```
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 saved file name

The second parameter is the saved image

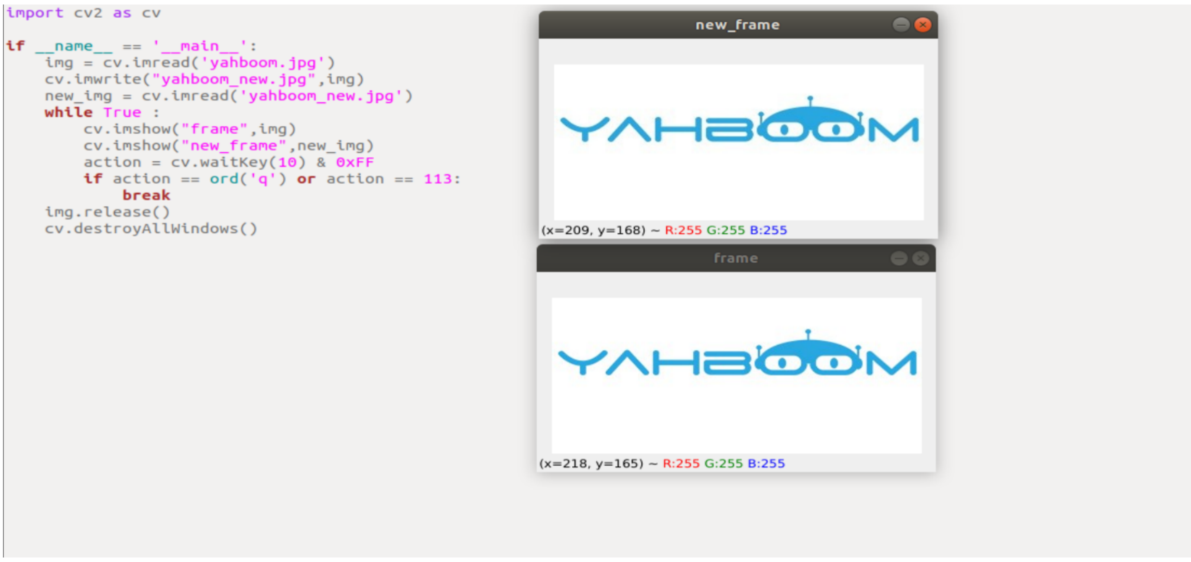
1.3.2. Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom_navrobo_astra/scripts/opencv/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 the newly written image
    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()
```

new_frame
YAHBOOM
(x=209, y=168) ~ R:255 G:255 B:255
frame
YAHBOOM
(x=218, y=165) ~ R:255 G:255 B:255

1.4, OpenCV camera read and display video

1.4.1. camera read

capture=cv.VideoCapture(0)

Parameter meaning:

The parameter in VideoCapture() is 0, which means to open the built-in camera of the notebook. 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, which determines whether the correct frame is read back

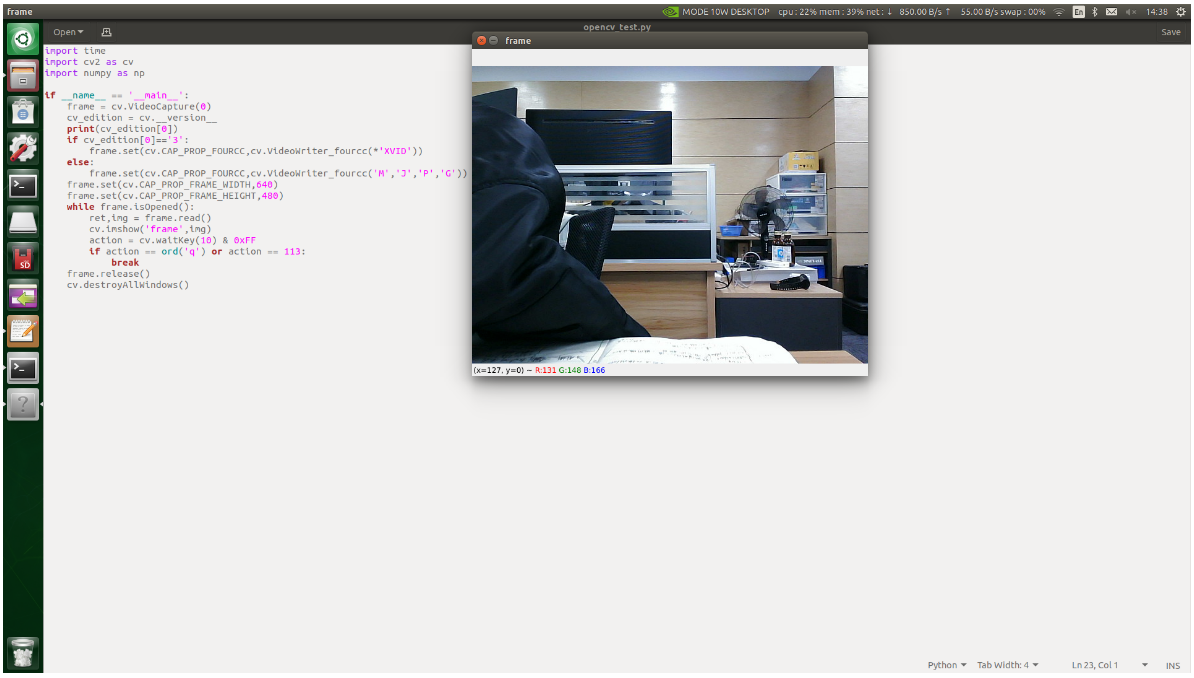
img: image data of each frame

1.4.3.Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom_navrobo_astra/scripts/opencv/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()
```



```
import time
import cv2 as cv
import numpy as np

if __name__ == '__main__':
    frame = cv.VideoCapture(0)
    cv_edition = cv.__version__
    print(cv_edition[0])
    if cv_edition[0]=='3':
        frame.set(cv.CAP_PROP_FOURCC,cv.VideoWriter_fourcc(*'XVID'))
    else:
        frame.set(cv.CAP_PROP_FOURCC,cv.VideoWriter_fourcc('M','J','P','G'))
    frame.set(cv.CAP_PROP_FRAME_WIDTH,640)
    frame.set(cv.CAP_PROP_FRAME_HEIGHT,480)
    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()
```

(x=127, y=0) ~ R:131 G:148 B:166

frame
MODE 10W DESKTOP cpu : 22% mem : 39% net : ↓ 850.00 B/s ↑ 55.00 B/s swap : 00%
En
14:38
Open
opencv_test.py
Save
frame
Python
Tab Width: 4
Ln 23, Col 1
INS

1.5, OpenCV pixel operation

1.5.1.pixel operation, we can change the pixel color at any position.

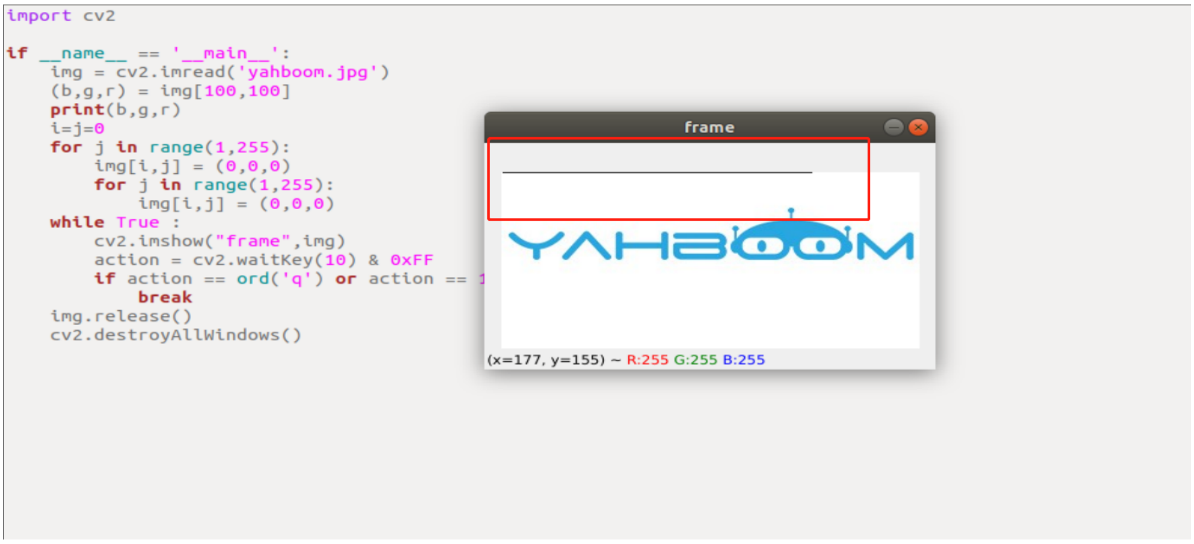
First, we need to read the image, then modify the value of bgr and assign an area to black.

1.5.2.Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-
00001/src/yahboom_navrobo_astra/scripts/opencv1_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 == 1
            break
    img.release()
    cv2.destroyAllWindows()
```

(x=177, y=155) ~ R:255 G:255 B:255

The red frame is the modified color value.