2. Open Source CV geometric transformation

2.1. OpenCV image scaling

2.1.1. In OpenCV, the function for image scaling is: cv2.resize(InputArray src, OutputArray dst, Size, fx, fy, interpolation)

Parameter meaning:

InputArray src: input image

OutputArray ds: output image

Size: output image size

fx,fy: scaling factors along the x-axis and y-axis

interpolation: interpolation method, you can choose INTER_NEAREST (nearest neighbor interpolation), INTER_LINEAR (bilinear interpolation (default setting)), INTER_AREA (resample using pixel area relationship), INTER_CUBIC (bicubic interpolation of 4x4 pixel neighborhood), INTER_LANCZOS4 (Lanczos interpolation of 8x8 pixel neighborhood)

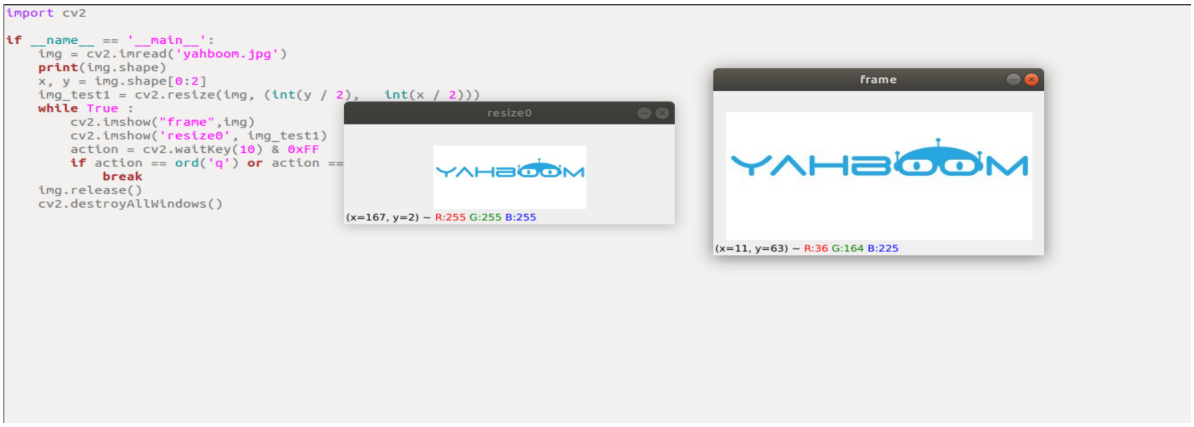
Note:

1. The output size format is (width, height)
2. The default interpolation method is: bilinear interpolation

2.1.2, code and actual effect display

Run the program

```
import cv2
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    print(img.shape)
    x, y = img.shape[0:2]
    img_test1 = cv2.resize(img, (int(y / 2),   int(x / 2)))
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('resize0', img_test1)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



2.2, OpenCV image cropping

2.2.1, Image cropping

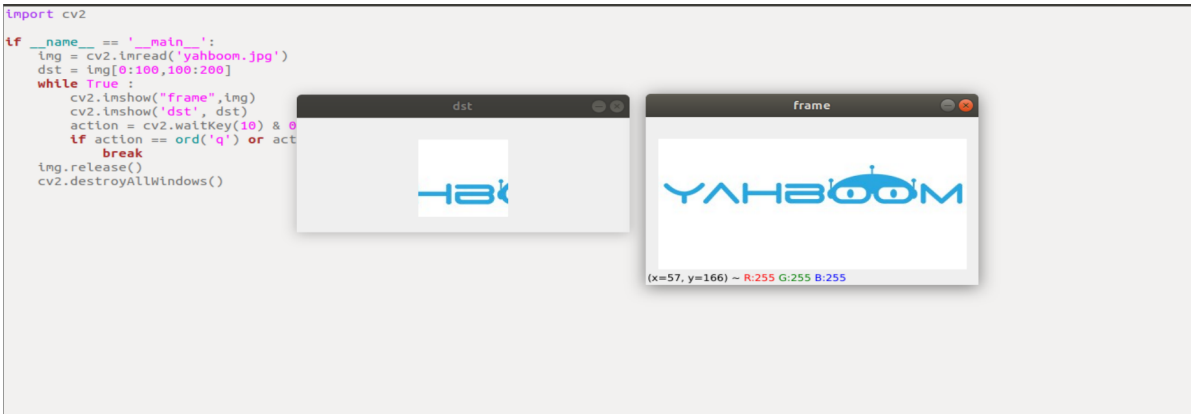
First read the image, and then get the pixel area in the array. In the following code, select the shape area X: 300-500 Y: 500-700. Note that the image size is 800*800, so the selected area should not exceed this resolution.

2.2.2, Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-
00001/src/yahboom_navrobo_astra/scripts/opencv/2_2.py
```

```
import cv2
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    dst = img[0:100,100:200]
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('dst', dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



2.3、OpenCV image translation

2.3.1、In OpenCV, image translation is achieved through affine transformation. The method used is cv2.warpAffine(src, M, dsize[,dst[, flags[, borderMode[, borderValue]]]])

Parameter meaning:

src - input image.

M - transformation matrix.

dsize - size of output image.

flags - combination of interpolation methods (int type!)

borderMode - border pixel mode (int type!)

borderValue - (important!) border fill value; by default, it is 0.

In the above parameters: M is an affine transformation matrix, which generally reflects the relationship of translation or rotation and is a 2×3 transformation matrix of InputArray type. In daily affine transformation, only the first three parameters are set, such as cv2.warpAffine(img,M,(rows,cols)), which can achieve basic affine transformation effects.

2.3.2. How to get the transformation matrix M? The following example illustrates that:

The original image src is converted to the target image dst through the conversion matrix M:

dst(x, y) = src(M11x + M12y+M13, M21x+M22y+M23)

The original image src is moved 200 pixels to the right and 100 pixels downward, and the corresponding relationship is:

dst(x, y) = src(x+200, y+100)

Complete the above expression, that is:

dst(x, y) = src(1·x + 0·y + 200, 0·x + 1·y + 100)

According to the above expression, the values of each element in the corresponding transformation matrix M can be determined as follows:

M11=1

M12=0

M13=200

M21=0

M22=1

M23=100

Substitute the above values into the transformation matrix M and get:

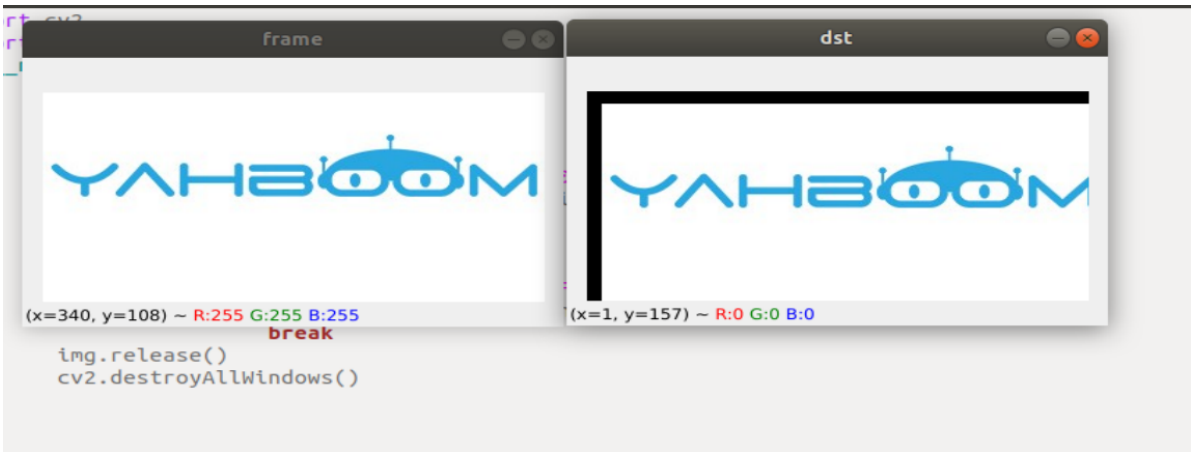
M = [ ]

2.3.3, code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-
00001/src/yahboom_navrobo_astra/scripts/opencv/2_3.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    matShift = np.float32([[1,0,10],[0,1,10]])# 2*3
    dst = cv2.warpAffine(img, matShift, (width,height))
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('dst', dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



2.4, OpenCV image mirror

2.4.1. Principle of image mirroring

Image mirroring transformation can be divided into two types: horizontal mirroring and vertical mirroring. Horizontal mirroring uses the vertical center line of the image as the axis to swap the pixels of the image, that is, swap the left and right halves of the image. Vertical mirroring uses the horizontal center line of the image as the axis to swap the upper and lower halves of the image.

Transformation principle:

Let the width of the image be width and the length be height. (x, y) are the coordinates after transformation, and (x0, y0) are the coordinates of the original image

Horizontal mirroring transformation

Forward mapping: x=width-x0-1, y=y0

Backward mapping: x0=width-x-1, y0=y

Vertical mirroring transformation

Upward mapping: x=x0, y=height-y0-1

Downward mapping: x0=x, y0=height-y-1

Summary:

During horizontal mirroring transformation, the entire image is traversed, and then each pixel is processed according to the mapping relationship. In fact, horizontal mirror transformation is to change the image coordinate column to the right and the right column to the left. The transformation can be done in columns. The same is true for vertical mirror transformation, which can be done in rows.

2.4.2, take vertical transformation as an example, let's see how Python is written

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-
00001/src/yahboom_navrobo_astra/scripts/opencv/2_4.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    deep = imgInfo[2]
    newImgInfo = (height*2,width,deep)
    dst = np.zeros(newImgInfo,np.uint8)#uint8
    for i in range(0,height):
        for j in range(0,width):
            dst[i,j] = img[i,j]
            dst[height*2-i-1,j]   = img[i,j]
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('dst', dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

Open

Save

```python
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    deep = imgInfo[2]
    newImgInfo = (height*2,width,deep)
    dst = np.zeros(newImgInfo,np.uint8)#uint8
    for i in range(0,height):
        for j in range(0,width):
            dst[i,j] = img[i,j]
            dst[height*2-i-1,j]   = img[i,j]
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('dst', dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```