1、OpenCV Basic Course

1.1、Reading Images and Presentations

1.1.1、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 in the image in grayscale mode, which can be represented by 0；

cv2.IMREAD_COLOR：，Read in a color image, which can be represented by 1；

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

1.1.2、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 created open window；

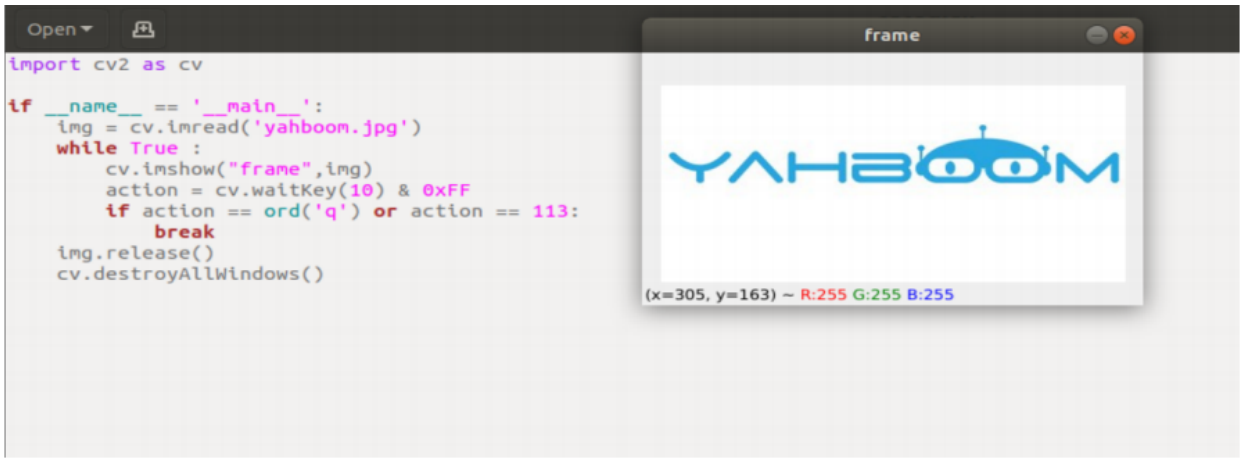
The second parameter represents the image to be displayed.

1.1.3、Code and actual effect display

Run sample program，

```
cd /root/yahboomcar_ros2_ws/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()
```



1.2、OpenCV image writing

1.2.1、Function method：cv2.imwrite('new_img_name', img)

Parameter Meaning：

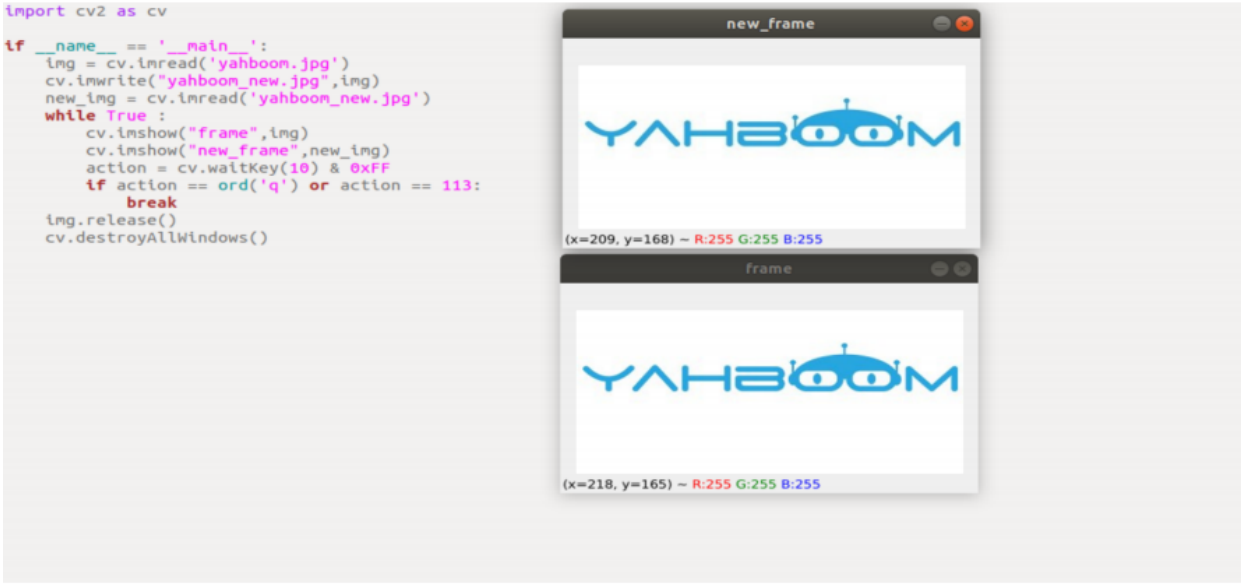
The first parameter is the saved file name, and the second parameter is the saved image.

1.2.2、Code and actual effect display

Run sample program，

```
cd /root/yahboomcar_ros2_ws/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 in it
    new_img = cv.imread('yahboom_new.jpg') #Read newly written images
    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()
```



1.3、OpenCV camera reading and displaying videos

1.3.1、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. If the parameter is the video file path, it means to open the video， such as cap =cv2.VideoCapture("../test.avi")

1.3.2、Display Camera Video

ret,img = frame.read()

Meaning of return value：

ret：ret is a bool value that determines whether the correct frame has been read back

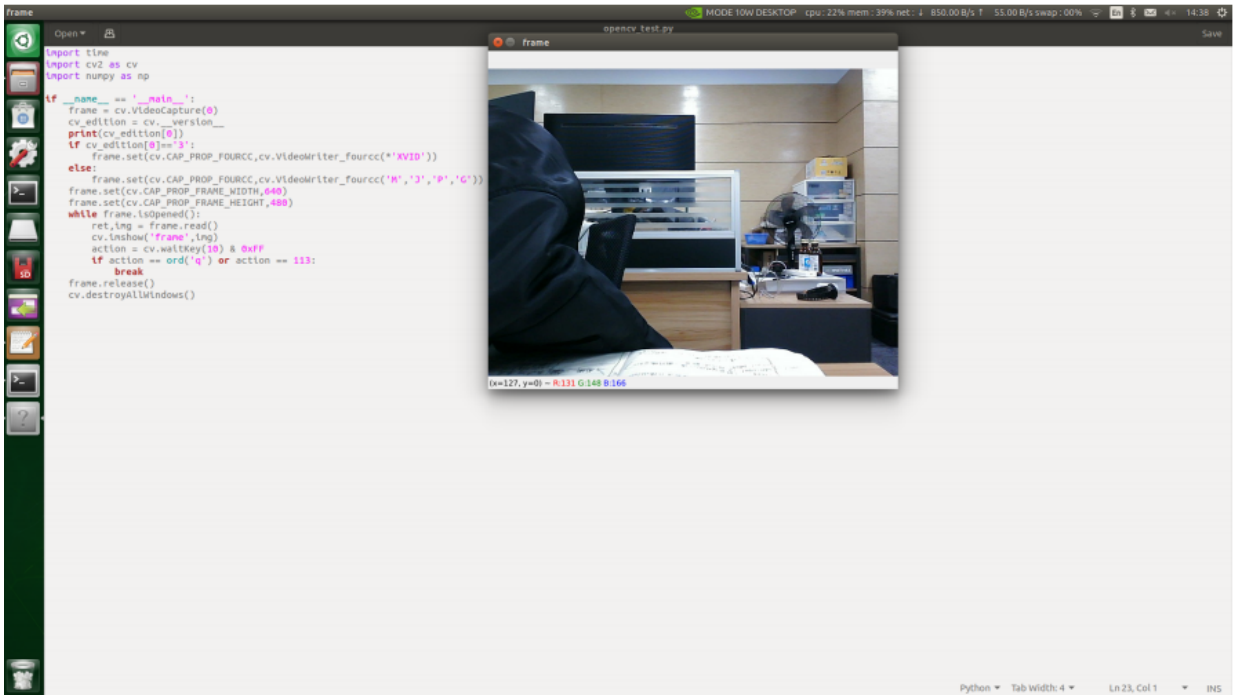
img：Image data for each frame

1.3.3、Code and actual effect display

run a program，

```
cd /root/yahboomcar_ros2_ws/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()
```



1.4、Openc pixel operation

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

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

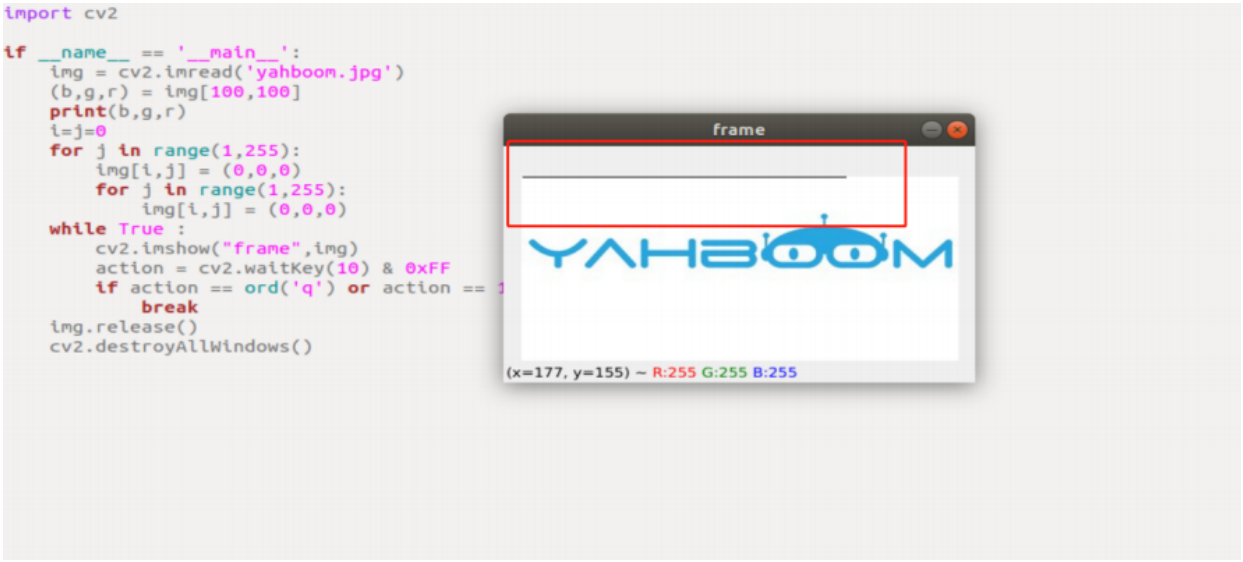
1.4.2、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/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()
```



The red box represents the modified pigment values.

2.1、OpenCV image scaling

2.1.1、cv2.resize(InputArray src,OutputArray dst, Size, fx, fy, interpolation)

Parameter Meaning：

InputArray src：input image

OutputArray ds：Output Picture

Size：Output Image Size

fx,fy：Scale factor along the x-axis and y-axis

interpolation：Insertion method, selectable from INTER_ NEAREST (Nearest-neighbor interpolation)，INTER_ LINEAR (Bilinear interpolation (default))，INTER_ Area (resampling using pixel region relationships)，INTER_ CUBIC (Bicubic interpolation of 4x4 pixel neighborhood)，INTER_ LANCZOS4 (Lanczos interpolation of 8x8 pixel neighborhood).

NOTE：

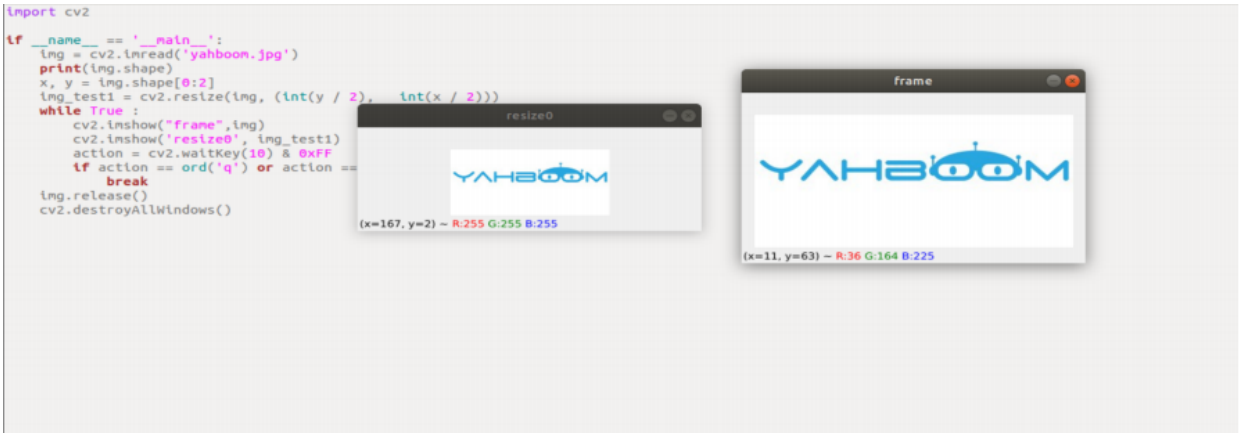
- Output size format is (width, height)
- The default interpolation method is: Bilinear interpolation

2.1.2、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 2_1.py
```

```
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 Clipping

2.2.1、Picture clipping

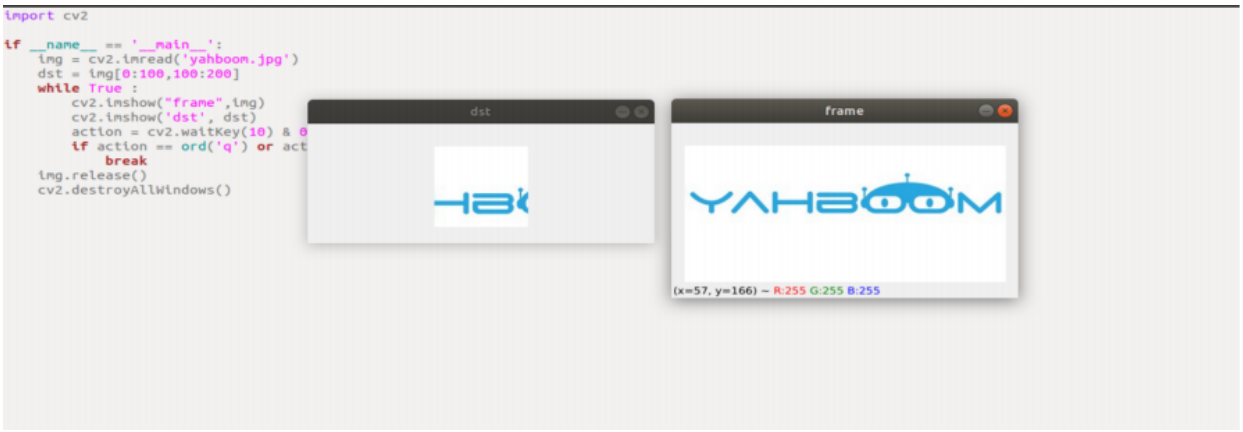
First, read the image, and then obtain the pixel area from the array. Select the shape area X: 300-500 Y: 500-700 in the following code, and note that the image size is 800 * 800, so the selected area should not exceed this resolution.

2.2.2、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 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 Pan

2.3.1、cv2.warpAffine(src, M, dsize[,dst[, flags[, borderMode[, borderValue]]]])

Parameter Meaning：

- src：Input Image
- M：Transformation matrix
- dsize：The size of the output image
- flags：Combination of interpolation methods (int type!)
- borderMode：Boundary pixel mode (int type!)
- borderValue：(Important!) Boundary fill values; By default, it is 0

Among the above parameters, M, as the Affine transformation matrix, generally reflects the relationship between translation and rotation, and is 2 of InputArray type × The transformation matrix of 3.

In daily Affine transformation, only the first three parameters are set, such as cv2.warpAffine (img, M, (rows, cols)).

Basic Affine transformation effect.

2.3.2、How to obtain the transformation matrix M? Here is an example to illustrate.

Convert the original image src to the target image dst through the conversion matrix M:

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

Move the original image src 200 pixels to the right and 100 pixels down, and the corresponding relationship is:

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

Complete the above expression, namely:

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

Based on the above expression, it can be determined that the values of each element in the corresponding transformation matrix M are:

M11=1

M12=0

M13=200

M21=0

M22=1

M23=100

Substitute the above values into the transformation matrix M to obtain:

M = [ ]

2.3.3、Code and actual effect display

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 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 mirroring

2.4.1、The principle of image mirroring

There are two types of image mirroring transformations: horizontal mirroring and vertical mirroring. Horizontal mirroring takes the vertical centerline of the image as the axis, swapping the pixels of the image, that is, swapping the left and right halves of the image. Vertical mirroring is based on the horizontal centerline of the image, swapping the upper and lower halves of the image.

Transformation principle:

Set the width of the image to width and the length to height. (x, y) represents the transformed coordinates, and (x0, y0) represents the coordinates of the original image,

- Horizontal Mirror Transform

  Forward mapping：x=width-x0-1,y=y0

  Backward mapping：x0=width-x-1,y0=y

- Vertical Mirror Transform

  Upward mapping：x=x0,y=height-y0-1

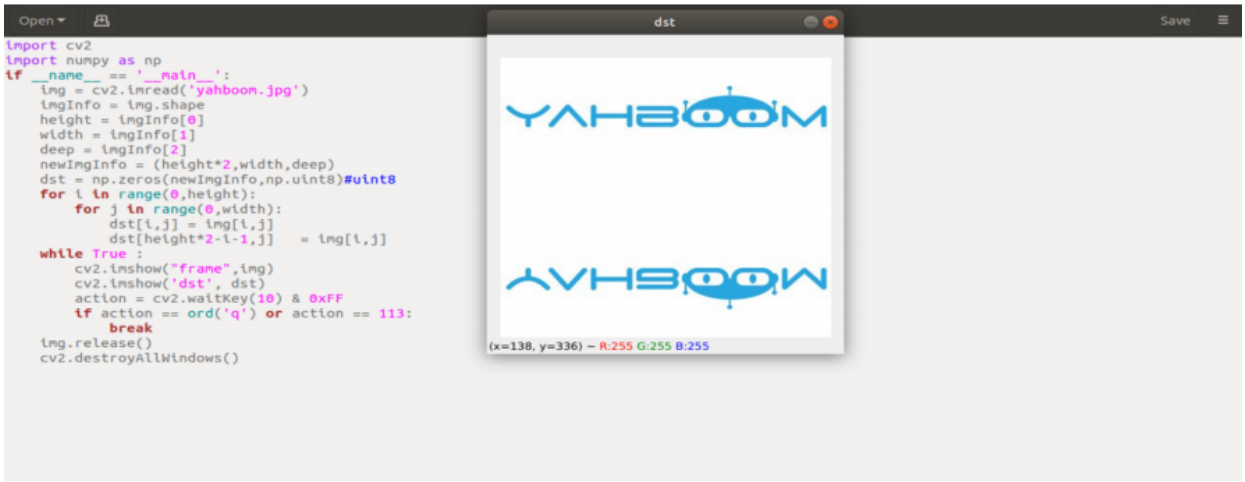
  Downward mapping：x0=x，y0=height-y-1

Summary: During the horizontal mirror transformation, the entire image was traversed and each pixel was processed based on the mapping relationship. In fact, horizontal mirror transformation is the process of moving the columns of image coordinates to the right and the columns on the right to the left, which can be transformed in columns. The same applies to vertical mirror transformations, which can be transformed in behavioral units.

2.4.2、Taking vertical transformation as an example, let's see how Python implements it

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 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()
```



3.1、Grayscale processing of opencv images

3.1.1、Image Grayscale

The process of transforming color image into Grayscale is the graying of image. The color of each pixel in a color image is determined by three components: R, G, and B, and each component can take a value of 0-255. This allows a pixel to have a color range of over 16 million (256 * 256 * 256=1677256). Grayscale is a special color image with the same R, G, and B components, and one pixel has a change range of 256. Therefore, in digital image processing, images of various formats are generally converted into Grayscale to reduce the calculation of subsequent images. The description of Grayscale, like color image, still reflects the distribution and characteristics of the overall and local chroma and highlight levels of the whole image.

3.1.2、Image grayscale processing

Grayscale processing is the process of transforming a color image into a Grayscale. A color image is divided into three components: R, G, and B, each displaying various colors such as red, green, and blue. Grayscale is the process of equalizing the R, G, and B components of a color. Pixels with large grayscale values are brighter (with a maximum pixel value of 255 indicating white), while those with lower grayscale values are darker (with a minimum pixel value of 0 indicating black). The core idea of grayscale image is that R=G=B, which is also known as the grayscale value.

1)Maximum method: Make the converted values of R, G, and B equal to the largest of the three values before conversion, that is, R=G=B=max (R, G, B). The grayscale image converted by this method has high brightness.

2)Average method: The values of R, G, and B after conversion are the average values of R, G, and B before conversion. Namely: R=G=B=(R+G+B)/3. The Grayscale produced by this method is relatively soft.

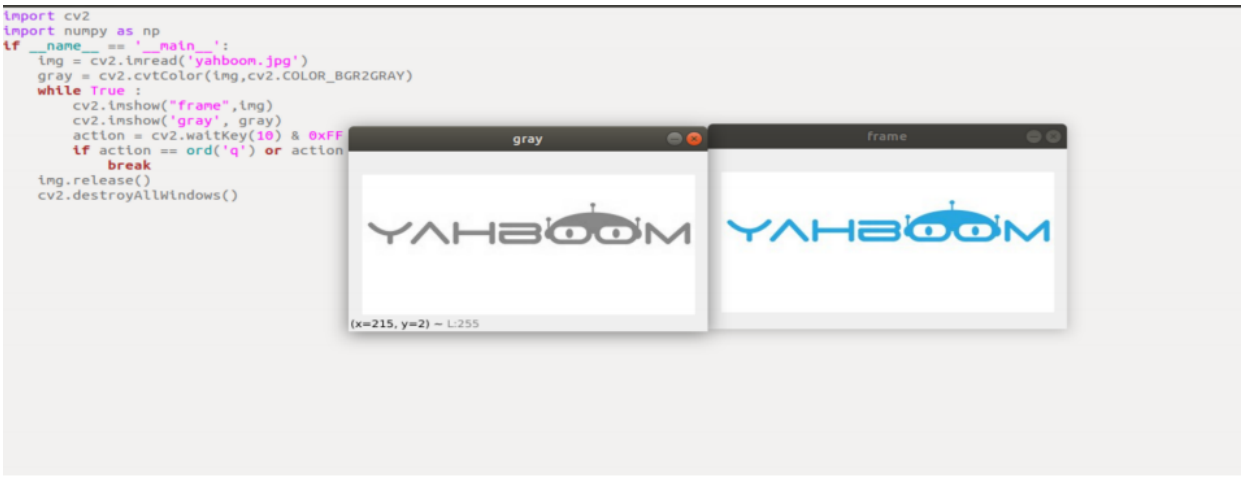
In OpenCV, use cv2. cvtColor (img1, cv2. COLOR_BGR2GRAY) to perform grayscale processing on images

3.1.3、Code and actual effect display

Run the program,

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_1.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('gray', gray)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
        img.release()
        cv2.destroyAllWindows()
```



3.2、Binary processing of opencv images

3.2.1、The core idea of binarization

Set the threshold to 0 (black) or 255 (white) for images larger than the threshold, making the image a black and white image. The threshold can be fixed or adaptive. The adaptive threshold is generally a comparison between the average value of a pixel and the weighted sum of Gaussian distribution pixels in a region where this point is in the middle order. A difference can be set or not.

3.2.2、cv2.threshold（src, threshold, maxValue,thresholdType)

Parameter Meaning：

- src：original image
- threshold：Current threshold
- maxVal：Maximum threshold, usually 255
- thresholdType：The threshold type generally has the following values:,
    - THRESH_BINARY = 0，#The grayscale value of pixels larger than the threshold is set to maxValue (such as an 8-bit grayscale value with a maximum of 255), and the grayscale value of pixels smaller than the threshold is set to 0.

    - THRESH_BINARY_INV = 1，#The grayscale value of pixels larger than the threshold is set to 0, while those smaller than the threshold are set to maxValue.
    - THRESH_TRUNC = 2，#The grayscale value of pixels larger than the threshold is set to 0, while those smaller than the threshold are set to maxValue.
    - THRESH_TOZERO = 3，#If the grayscale value of a pixel is less than the threshold, no change will be made, while if it is greater than the threshold, all grayscale values will become 0.
    - THRESH_TOZERO_INV = 4 #If the grayscale value of a pixel is greater than this threshold, no change will be made. If the grayscale value of a pixel is less than this threshold, all grayscale values will become 0.
- return value：
    - retval：Consistent with parameter threshold 3
    - dst： Final image

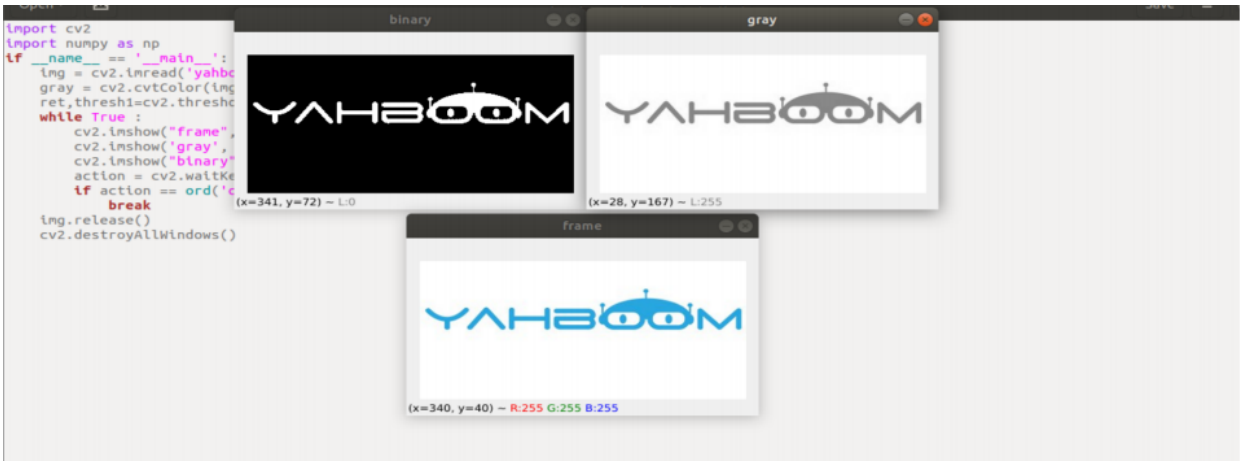
Note: Before binarization, we need to grayscale the color image to obtain a grayscale image.

3.2.3、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_2.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    ret,thresh1=cv2.threshold(gray,180,255,cv2.THRESH_BINARY_INV)
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('gray', gray)
        cv2.imshow("binary",thresh1)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



3.3、OpenCV image edge detection

3.3.1、The purpose of image edge detection

Significantly reduce the data size of the image while retaining its original image attributes. At present, there are multiple algorithms that can perform edge detection. Although Canny algorithm has a long history, it can be said that it is a standard algorithm for edge detection and is still widely used in research.

3.3.2、Canny edge detection algorithm

Among the currently commonly used edge detection methods, the Canny edge detection algorithm is one of the methods with strict definitions that can provide good and reliable detection. Due to its advantages of meeting the three standards of edge detection and simple implementation process, it has become one of the most popular algorithms for edge detection.

The Canny edge detection algorithm can be divided into the following 5 steps:

- Use Gaussian filter to smooth the image and filter out noise
- Calculate the gradient intensity and direction of each pixel in the image
- Apply Non Maximum Suppression to eliminate spurious responses caused by edge detection
- Applying Double Threshold Detection to Determine Real and Potential Edges
- Edge detection is ultimately achieved by suppressing isolated weak edges

3.3.3、OpenCV Implementation Steps

- Image Grayscale：gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
- Gaussian filtering (denoising) image：GaussianBlur（src，ksize，sigmaX [，dst [，sigmaY [，borderType]]]）-> dst
    - src：The input image is usually a grayscale image
    - ksize：Gaussian kernel size
    - sigmaX：Gaussian kernel standard deviation in the X direction
    - sigmaY：Gaussian kernel standard deviation in the Y direction
    - dst：Processed image

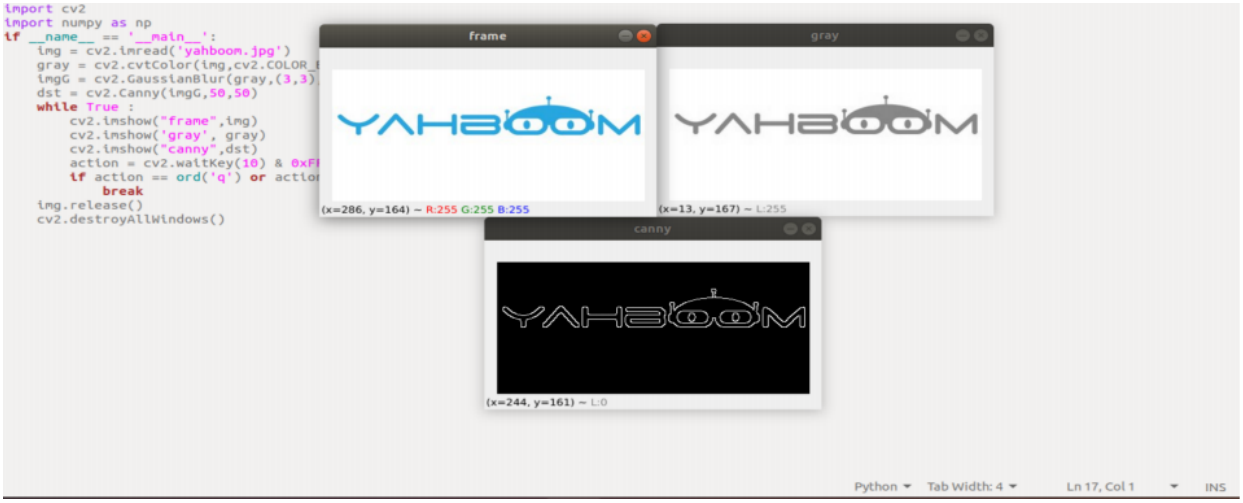
- Canny method for processing images：edges=cv2.Canny(image, threshold1, threshold2[, apertureSize[,L2gradient]])
    - edges：Calculated edge image
    - image ： The calculated edge image is generally the image obtained after Gaussian processing
    - threshold1 ： The first threshold during processing
    - threshold2 ： Second threshold during processing
    - apertureSize ： Aperture size of Sobel operator
    - L2gradient ： The default value for calculating the gradient amplitude of an image is False. If it is true, the more accurate L2 norm will be used for calculation (that is, the square sum of the derivatives in two directions will be derived again), otherwise the L1 norm will be used (the absolute values of the two Directional derivative will be added directly).

3.3.4、Code and actual effect display

run a program，

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_3.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    imgG = cv2.GaussianBlur(gray,(3,3),0)
    dst = cv2.Canny(imgG,50,50)
    while True :
        cv2.imshow("frame",img)
        cv2.imshow('gray', gray)
        cv2.imshow("canny",dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



3.4、OpenCV line segment drawing

3.4.1、cv2.line (dst, pt1, pt2, color, thickness=None, lineType=None, shift=None)

Parameter Meaning:

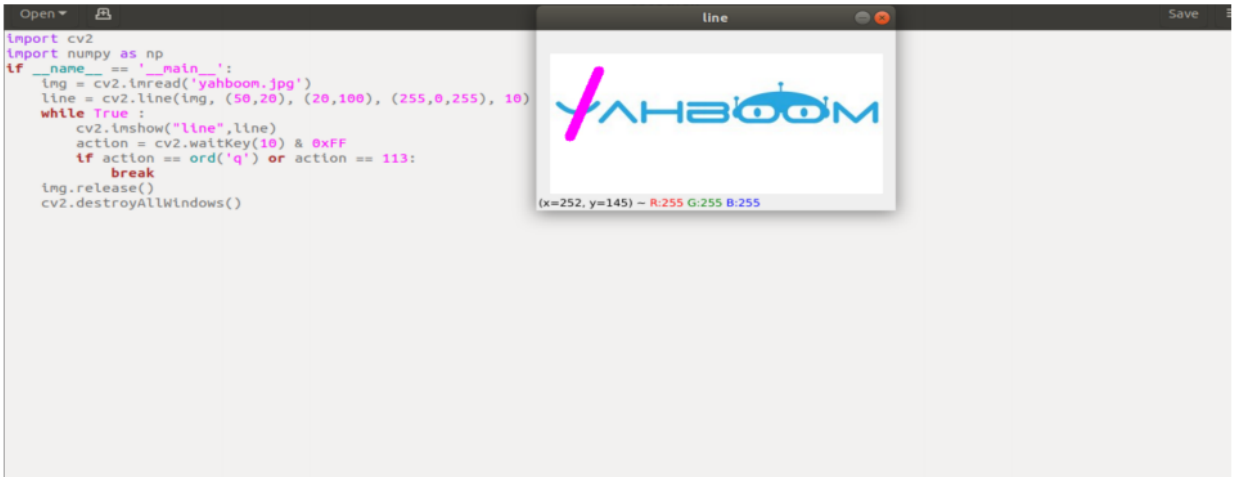
- dst: Output image
- pt1, pt2: Required parameter. The coordinate points of a line segment represent the starting and ending points, respectively
- color: Required parameter. Used to set the color of line segments
- thickness: Optional parameters. Used to set the width of line segments
- lineType: Optional parameters. Used to set the type of line segment, selectable as 8 (8 adjacent connecting lines - default), 4 (4 adjacent connecting lines), and cv2. LINE_ AA is anti aliasing

3.4.2、Running program code and displaying actual effects

run a program,

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_4.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    line = cv2.line(img, (50,20), (20,100), (255,0,255), 10)
    while True :
        cv2.imshow("line",line)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
        img.release()
        cv2.destroyAllWindows()
```



3.5、OpenCV Draw Rectangle

3.5.1、cv2.rectangle (img, pt1, pt2, color, thickness=None, lineType=None, shift=None)

Parameter Meaning:

- img: Canvas or carrier images
- Required parameter. Used to set the color of the rectangle
- color: Required parameter. Used to set the color of the rectangle
- thickness: Optional parameters. Used to set the width of the rectangular edge. When the value is negative, it indicates filling the rectangle
- lineType: Optional parameters. Used to set the type of line segment, optional 8 (8 adjacent connecting lines - default), 4 (4 adjacent connecting lines) cv2. LINE_ AA is anti aliasing

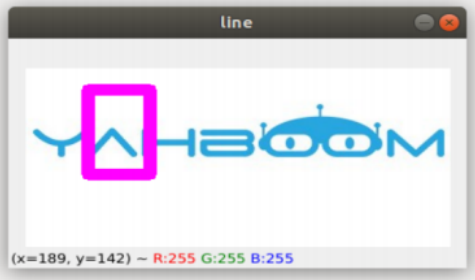
3.5.2、Code and Effect Display

run a program,

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_5.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    rect = cv2.rectangle(img, (50,20), (100,100), (255,0,255), 10)
    while True :
        cv2.imshow("line",rect)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

```python
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    rect = cv2.rectangle(img, (50,20), (100,100), (255,0,255), 10)
    while True :
        cv2.imshow("line",rect)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



3.6、Draw a circle with opencv

3.6.1、cv2.circle(img, center, radius, color[,thickness[,lineType]])

Parameter Meaning：

- img：Drawing or carrier image cloth
- center：It is the center coordinate, format： (50,50)
- radius：radius
- thickness： The thickness of the lines. Default is 1. If -1, it is filled with solid
- lineType：Line type. The default is 8, connection type. The following table explains

parameter	Parameter Description
cv2.FILLED	Fill
cv2.LINE_4	4 Connection Types
cv2.LINE_8	8 Connection Types
cv2.LINE_AA	Anti aliasing, this parameter will make the lines smoother

3.6.2、Code and actual effect display

run a program，

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_6.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    circle = cv2.circle(img, (80,80), 50, (255,0,255), 10)
    while True :
        cv2.imshow("circle",circle)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

3.7、Drawing Ellipses with OpenCV

3.7.1、cv2.ellipse(img, center, axes, angle, StartAngle, endAngle, color[,thickness[,lineType])

Parameter Meaning：

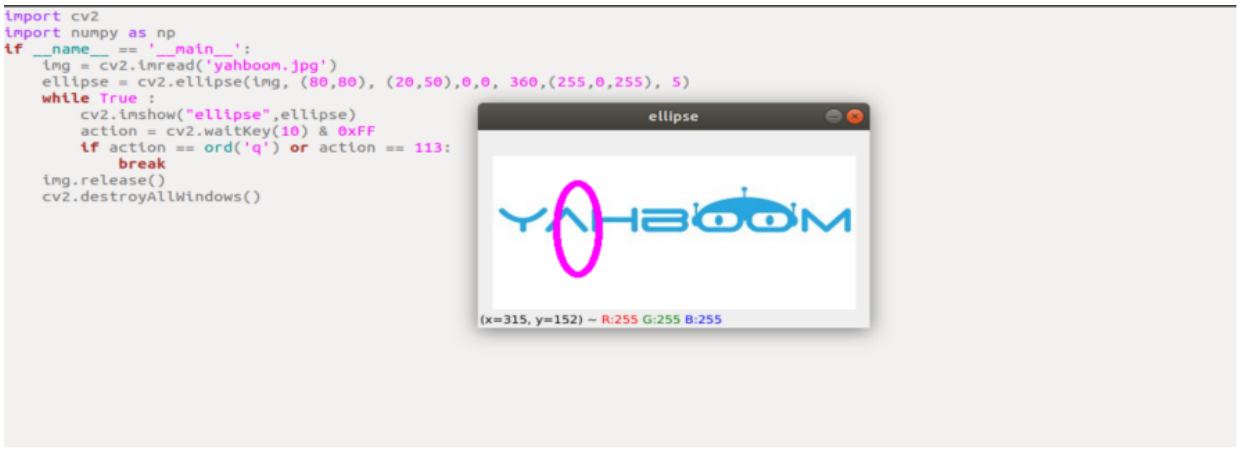
- center：The center point of an ellipse，（x，y）
- axes：Refers to short radius and long radius，（x，y）
- StartAngle：The angle of the starting angle of the arc
- endAngle：The angle of the ending angle of the arc
- Img, color, thickness, and lineType can refer to the description of the circle

3.7.2、Code and actual effect display

run a program，

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_7.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    ellipse = cv2.ellipse(img, (80,80), (20,50),0,0, 360,(255,0,255), 5)
    while True :
        cv2.imshow("ellipse",ellipse)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



3.8、Opencv drawing polygons

3.8.1、cv2.polylines(img,[pts],isClosed, color[,thickness[,lineType]])

Parameter Meaning：

- pts：Polygon Vertex
- isClosed：Closed or not.（True/False）
- Other parameters refer to the drawing parameters of the circle

3.8.2、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_8.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    points = np.array([[120,50], [40,140], [120,70], [110,110], [50,50]],np.int32)
    polylines = cv2.polylines(img, [points],True,(255,0,255), 5)
    while True :
        cv2.imshow("polylines",polylines)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

3.9、OpenCV drawing text

3.9.1、cv2.putText(img, str, origin, font, size,color,thickness)

Parameter Meaning：

- img：input image
- str：Drawn Text
- origin：The upper left corner coordinate (integer) can be understood as where the text starts
- font：typeface
- size：font size
- color：font color
- thickness：font-weight

3.9.2、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 3_9.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    cv2.putText(img,'This is Yahboom!',(50,50),cv2.FONT_HERSHEY_SIMPLEX,1,
(0,200,0),2)
    while True :
        cv2.imshow("img",img)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

4.1、OpenCV repair image

Inpainting is a kind of algorithm in computer vision, whose goal is to fill the area in the image or video. This area is identified using binary masks, and filling is usually done based on the boundary information of the area that needs to be filled. The most common application of Inpainting is to restore old scanned photos. It is also used to remove small unnecessary objects from images.

4.1.1、dst = cv2.inpaint(src, inpaintMask, inpaintRadius, flags)

Parameter Meaning：

- src：Source image, which is the image that needs to be repaired
- inpaintMask：Binary mask, indicating the pixels to be repaired.
- dst：Final image
- inpaintRadius：Represents the radius of the repair
- flags : Repair algorithm, mainly including INPAINT_NS (Navier-Stokes based method) or INPAINT_TELEA (Fastmarching based method)

The repair based on Navier Stokes should be slower and tend to produce more ambiguous results than the fast marching method. In practice, we have not found this situation. INPAINT_ NS produced better results in our testing and slightly faster than INPAINT_ TELEA.

4.1.2、Code and actual effect display

（1）、Firstly, based on the intact image, we will add damage to it, which can be understood as modifying the pixel values of specific parts,

run a program，

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 4_1_1.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    for i in range(50,100):
        img[i,50] = (0,0,0)
        img[i,50+1] = (0,0,0)
        img[i,50-1] = (0,0,0)
    for i in range(100,150):
        img[150,i] = (0,0,0)
        img[150,i+1] = (0,0,0)
        img[150-1,i] = (0,0,0)
    cv2.imwrite("damaged.jpg",img)
    dam_img = cv2.imread('damaged.jpg')
    while True :
        cv2.imshow("dam_img",dam_img)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

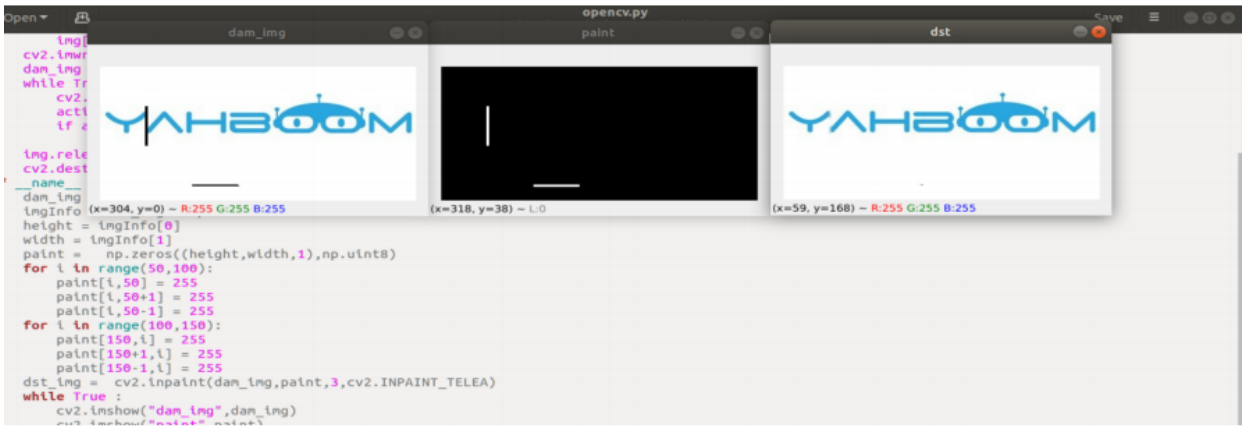
After running, an image will be generated, which is considered as a damaged image of the original image,

（2）、To fix the photo you just created, first read it, then create a mask, and finally use the function to fix it

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 4_1_2.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    dam_img = cv2.imread('damaged.jpg')
    imgInfo = dam_img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    paint = np.zeros((height,width,1),np.uint8)
    for i in range(50,100):
        paint[i,50] = 255
        paint[i,50+1] = 255
        paint[i,50-1] = 255
    for i in range(100,150):
        paint[150,i] = 255
        paint[150+1,i] = 255
        paint[150-1,i] = 255
    dst_img = cv2.inpaint(dam_img,paint,3,cv2.INPAINT_TELEA)
    while True :
        cv2.imshow("dam_img",dam_img)
        cv2.imshow("paint",paint)
        cv2.imshow("dst",dst_img)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



As shown in the figure, the left side shows the pre repair image, the middle is the mask image, and the right side is the original image after repair.

4.2、OpenCV image brightness enhancement

Implementation process: Synchronize and amplify the three channel values of each pixel, while keeping the channel values between 0 and 255. In fact, it means traversing each pixel, adding or subtracting numerical values, and then determining whether the three channel rgb is in the 0 to 255 range. If it is greater than or less than, the value is 255 or 0.

4.2.1、Code and actual effect display

run a program，

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 4_2.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    dst = np.zeros((height,width,3),np.uint8)
    for i in range(0,height):
        for j in range(0,width):
            (b,g,r) = img[i,j]
            bb = int(b) + 100
            gg = int(g) + 100
            rr = int(r) + 100
            if bb > 255:
                bb = 255
            if gg > 255:
                gg = 255
            if rr > 255:
                rr = 255
            dst[i,j] = (bb,gg,rr)
    while True :
        cv2.imshow("dst",dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

The image on the left is the original image, while the image on the back shows the photo with increased brightness.

4.3、OpenCV Image Peeling and Whitening

OpenCV implements the function of peeling and whitening images, and the principle of implementation is basically the same as the principle of "1.20 OpenCV image brightness enhancement". However, here we do not need to process the r value, we only need to follow this formula, p=p (x) * 1.4+y, where p (x) represents the b or g channel, y represents the value that needs to be increased or decreased. Similarly, after adding a value, we need to make a judgment on the value.

4.3.1、Code and actual effect display

run a program

```
cd /root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/
python3 4_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]
    dst = np.zeros((height,width,3),np.uint8)
    for i in range(0,height):
        for j in range(0,width):
            (b,g,r) = img[i,j]
            bb = int(b*1.4) + 5
            gg = int(g*1.4) + 5
            if bb > 255:
                bb = 255
            if gg > 255:
                gg = 255
            dst[i,j] = (bb,gg,r)
    while True :
        cv2.imshow("origin",img)
        cv2.imshow("dst",dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

```python
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    dst =   np.zeros((height,width,3),np
    for i in range(0,height):
        for j in range(0,width):
            (b,g,r) = img[i,j]
            bb = int(b*1.2) + 5
            gg = int(g*1.2) + 5
            if bb > 255:
                bb = 255
            if gg > 255:
                gg = 255
            dst[i,j] = (bb,gg,r)
    while True :
        cv2.imshow("origin",img)
        cv2.imshow("dst",dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```