Image processing and drawing

1、OpenCV image grayscale processing

1.1、Image grayscale

The process of converting a color image into a grayscale image is the grayscale processing of the 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, so there can be more than 16 million pixels (256*256*256=1677256) The range of color changes. The grayscale image is a special color image with the same three components of R, G, and B. The variation range of one pixel is 256. Therefore, in digital image processing, images in various formats are generally converted into grayscale. image to make subsequent images less computationally intensive. The description of a grayscale image, like a color image, still reflects the distribution and characteristics of the overall and local chroma and highlight levels of the entire image.

1.2、Image grayscale processing

Grayscale processing is the process of converting a color image into a grayscale image. The color image is divided into three components: R, G, and B, which display various colors such as red, green, and blue respectively. Grayscale is the process of making the color R, G, and B components equal. Pixels with large grayscale values are brighter (the maximum pixel value is 255, which is white), and vice versa (the lowest pixel is 0, which is black).

The core idea of image grayscale is R = G = B. This value is also called grayscale value.

1)Maximum value 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 is very bright.

2)Average method: The values of R, G, and B after conversion are the average values of R, G, and B before conversion. That is: R=G=B=(R+G+B)/3. This method produces softer grayscale images.

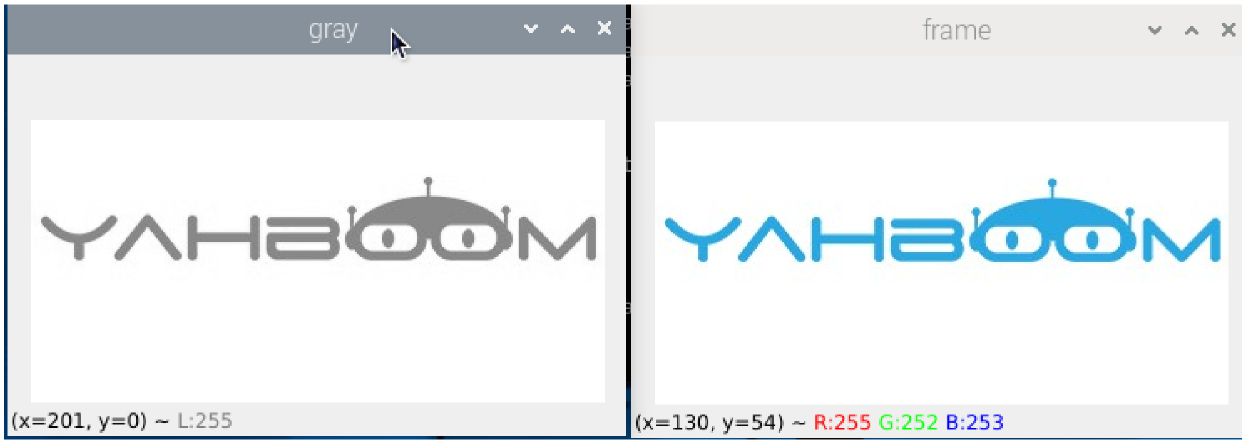
In OpenCV, use cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY) to grayscale the image

1.3、Code and actual effect display

```
cd ~/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()
```

Effect picture after running the program



(x=201, y=0) ~ L:255
(x=130, y=54) ~ R:255 G:252 B:253

2、OpenCV image binarization processing

2.1、The core idea of binarization

Set a threshold, and the value greater than the threshold is 0 (black) or 255 (white), so that the image is called a black and white image. The threshold can be fixed or adaptive. The adaptive threshold is generally a comparison of a point pixel with the mean value of regional pixels or the weighted sum of Gaussian distribution at this point, in which a difference value may or may not be set.

2.2、The threshold function is provided in Python-OpenCV

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

Parameter meaning:

src：original image

threshold：current threshold

maxVal：Maximum threshold, usually 255

thresholdType：Threshold type, generally has the following values enum ThresholdTypes {   THRESH_BINARY   = 0,   #The grayscale value of pixels greater than the threshold is set to maxValue (for example, the maximum 8-bit grayscale value is 255)The gray value of pixels whose gray value is less than the threshold is set to 0.

THRESH_BINARY_INV = 1,   #The gray value of pixels greater than the threshold is set to 0, and the gray value of pixels less than the threshold is set to maxValue.

THRESH_TRUNC   = 2,   #The gray value of pixels greater than the threshold is set to 0, and the gray value of pixels less than the threshold is set to maxValue.

THRESH_TOZERO   = 3, #The gray value of the pixels less than the threshold will not be changed, and the gray values of the parts greater than the threshold will all become 0.

THRESH_TOZERO_INV = 4   #If the gray value of the pixel is greater than the threshold, no change will be made. If the gray value of the pixel is less than the threshold, all the gray values will be changed to 0. }

return value：

retval：Consistent with parameter thresh

dst： Result image

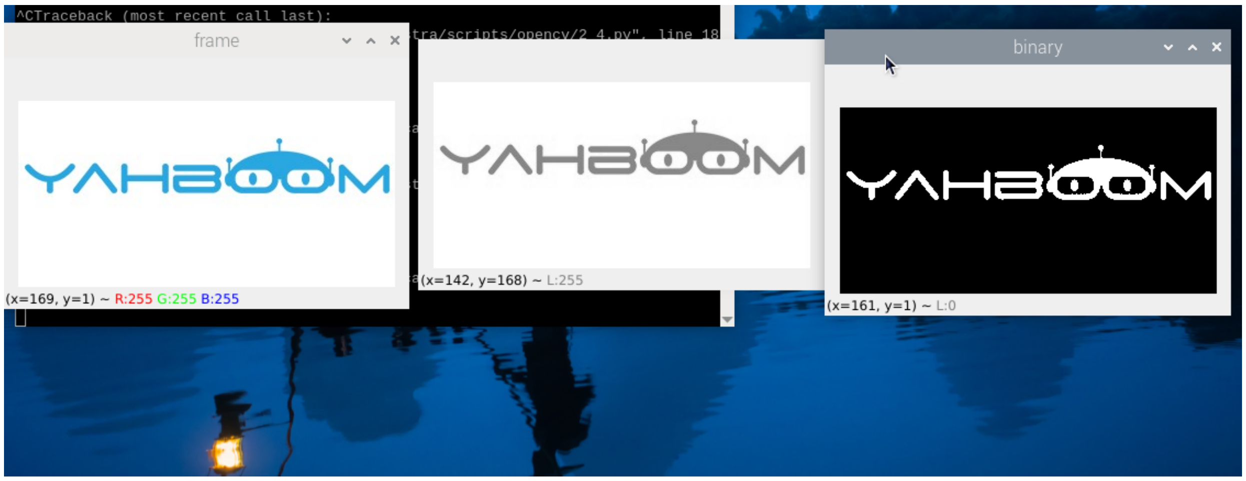
Note: Before binarization, we need to grayscale the color image to obtain a grayscale image.

2.3、Code and actual effect display

```
cd ~/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()
```

Effect picture after running the program


```
^CTraceback (most recent call last):
tra/scripts/opencv/2_4.py", line 18
```

3、OpenCV image edge detection

3.1、The purpose of image edge detection

Significantly reduce the data size of the image while retaining the original image attributes. There are currently many algorithms for edge detection. Although the Canny algorithm is old, it can be said that it is a standard algorithm for edge detection and is still widely used in research.

3.2. Canny edge detection algorithm

Among currently commonly used edge detection methods, the Canny edge detection algorithm is one of the methods that is strictly defined and can provide good and reliable detection. Because it has the advantages of meeting the three criteria of edge detection and being simple to implement, 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:

（1）. Use a Gaussian filter to smooth the image and filter out noise

（2）. Calculate the gradient intensity and direction of each pixel in the image

（3）.Apply Non-Maximum Suppression to eliminate spurious responses from edge detection

（4）. Apply Double-Threshold detection to determine real and potential edges

（5）. Edge detection is finally accomplished by suppressing isolated weak edges.

3.3、How do we implement this in OpenCV? It's very simple, divided into three steps

（1）. Image grayscale： gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

（2）. Gaussian filtered (noise reduction) image： GaussianBlur（src，ksize，sigmaX [，dst [，sigmaY [，borderType]]]）-> dst

Parameter meaning：

src：The input image is usually a grayscale image.

ksize：Gaussian kernel size

sigmaX ：Gaussian kernel standard deviation in X direction sigmaY： Gaussian kernel standard deviation in Y direction

dst：processed image

（3）. Canny method to process the image：edges=cv2.Canny(image, threshold1, threshold2[, apertureSize[, L2gradient]])

Parameter meaning：

edges：Calculated edge image

image ：the calculated edge image

threshold1 ：the first threshold during processing

threshold2 ：the second threshold during processing

apertureSize ：the aperture size of the Sobel operator

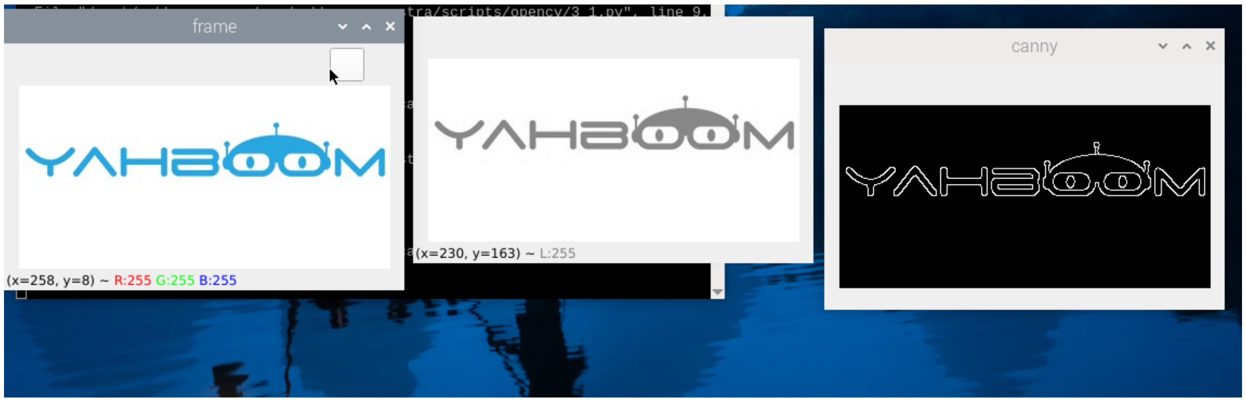
L2gradient ：The flag for calculating the gradient magnitude of the image. The default value is False. If True, the more precise L2 norm is used for the calculation (ie, the sum of the squares of the derivatives in the two directions is re-squared), otherwise the L1 norm is used (the absolute value of the derivatives in the two directions is directly added).

3.4、Code and actual effect display

```
cd ~/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()
```

Effect picture after running the program



4、OpenCV line segment drawing

4.1、When using OpenCV to process images, we sometimes need to draw line segments, rectangles, etc. on the image. used in OpenCV

The cv2.line(dst, pt1, pt2, color, thickness=None, lineType=None, shift=None) function draws line segments.

Parameter meaning:

dst：output image.

pt1，pt2：Required parameters. The coordinate points of the line segment, representing the starting point and the ending point, respectively

color：required parameter. Used to set the color of the line segment

thickness：optional parameter. Used to set the width of the line segment

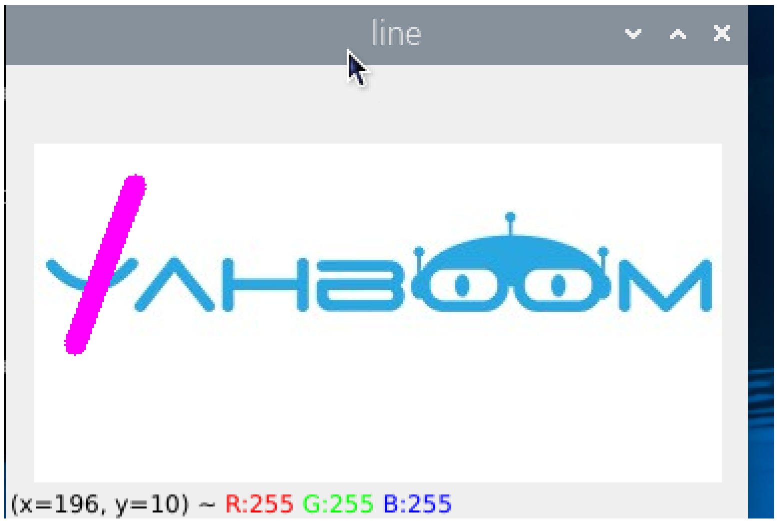
lineType：optional parameter. Used to set the type of line segment, optional 8 (8 adjacent connecting lines - default), 4 (4 adjacent connecting lines) and cv2.LINE_AA for antialiasing .

4.2、Code and actual effect display

```
cd ~/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()
```

Effect picture after running the program.



5、OpenCV draw rectangle

5.1、In OpenCV, the functions used to draw rectangles are as follows

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

参数含义：

img： canvas or carrier image

pt1，pt2： required parameters. The vertices of the rectangle represent the vertex and the diagonal vertex respectively, that is, the upper left corner and the lower right corner of the rectangle (these two vertices can determine a unique rectangle), which can be understood as a diagonal line.

color：required parameter. Used to set the color of the rectangle

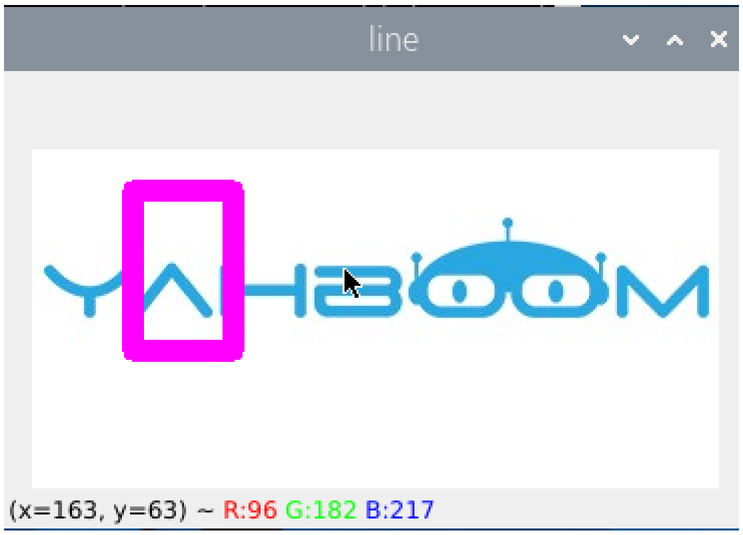
thickness：optional parameter. Used to set the width of the side of the rectangle. When the value is negative, it means to fill the rectangle

lineType：optional parameter. Used to set the type of line segment, optional 8 (8 adjacent connecting lines - default), 4 (4 adjacent connecting lines) and cv2.LINE_AA for antialiasing

5.2、 Code and effect display

```
cd ~/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()
```



6、. OpenCV draws a circle

6.1、In OpenCV, the functions used to draw circles are as follows

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

Parameter meaning:

img: painting or vector image cloth

center: coordinates of the center of the circle, format: (50,50)

radius: radius

thickness: Line thickness. Default is 1. If -1 then fill solid

lineType: Line type. The default is 8, the connection type. Described in the table below

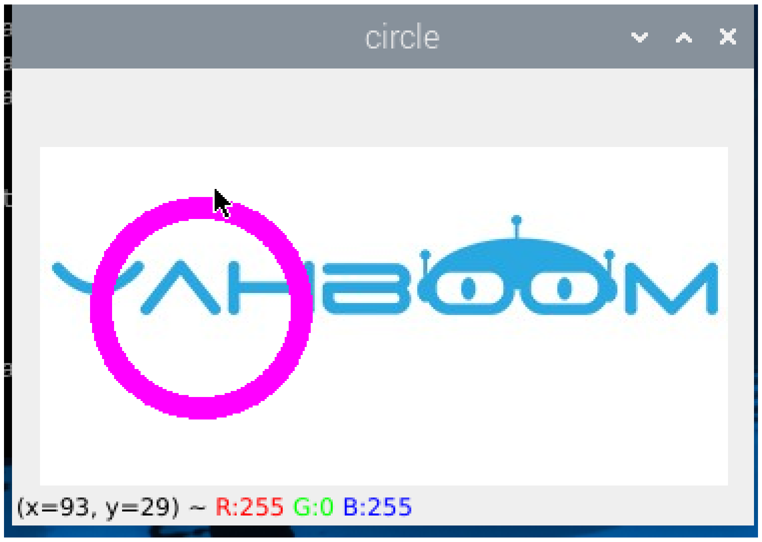
parameter	illustrate
cv2.FILLED	filling
cv2.LINE_4	4 Connection Types
cv2.LINE_8	8 connection types

parameter	illustrate
cv2.LINE_AA	Antialiasing, this parameter will make the lines smoother

6.2、 Code and actual effect display

```
cd ~/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()
```



(x=93, y=29) ~ R:255 G:0 B:255

7、OpenCV draw ellipse

7.1、In OpenCV, the functions used to draw ellipses are as follows

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

参数含义：

center：center: the center point of the ellipse，（x，y)

axes：refers to the short radius and long radius，（x，y)

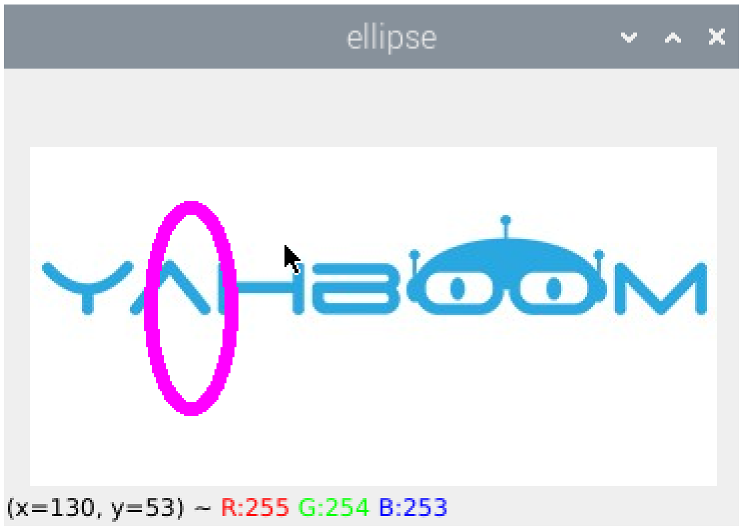
StartAngle：The angle of the starting angle of the arc

endAngle：the angle of the end angle of the arc img、color、thickness and lineType can refer to the description of the circle

7.2、Code and actual effect display

```
cd ~/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()
```



(x=130, y=53) ~ R:255 G:254 B:253

8、OpenCV draws polygons

8.1、In OpenCV, the functions used to draw polygons are as follows

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

Parameter meaning:

pts：vertices of the polygon

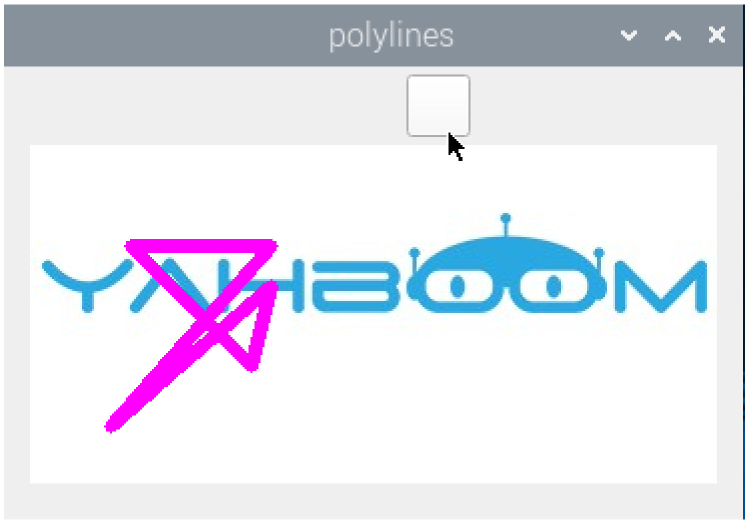
isClosed：Whether it is closed。 (True/False)

Other parameters refer to the drawing parameters of the circle

8.2、 Code and actual effect display

```
cd ~/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()
```



9、OpenCV to draw text

9.1、In OpenCV, the functions used to draw text are as follows

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

Parameter meaning:

img： input image

str：the drawn text

origin： the upper left corner coordinate (integer), which can be understood as where the text starts

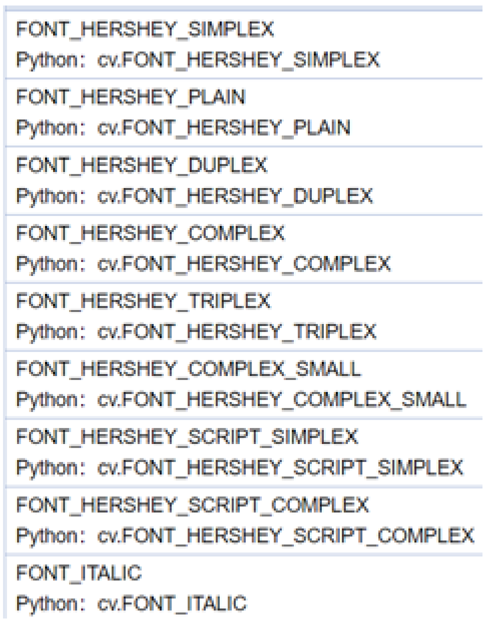
font：font

size：font size

color： font color

thickness： font thickness

Among them, the font is optional

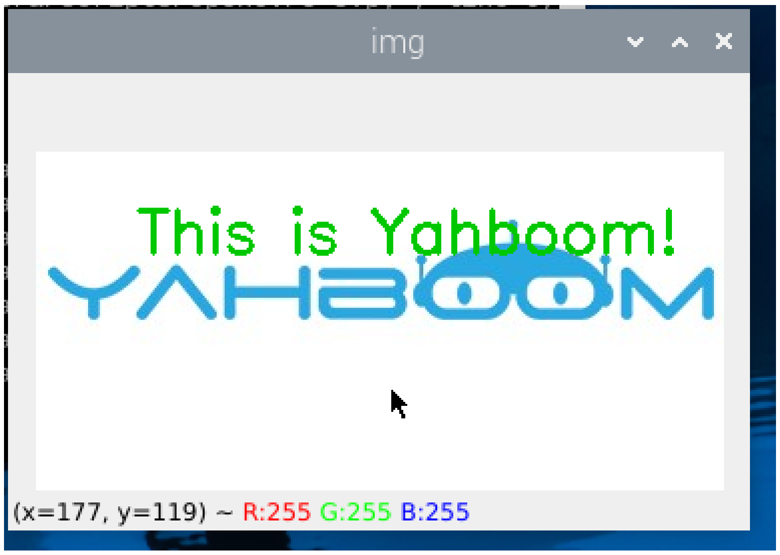


FONT_HERSHEY_SIMPLEX	Python：cv.FONT_HERSHEY_SIMPLEX
FONT_HERSHEY_PLAIN	Python：cv.FONT_HERSHEY_PLAIN
FONT_HERSHEY_DUPLEX	Python：cv.FONT_HERSHEY_DUPLEX
FONT_HERSHEY_COMPLEX	Python：cv.FONT_HERSHEY_COMPLEX
FONT_HERSHEY_TRIPLEX	Python：cv.FONT_HERSHEY_TRIPLEX
FONT_HERSHEY_COMPLEX_SMALL	Python：cv.FONT_HERSHEY_COMPLEX_SMALL
FONT_HERSHEY_SCRIPT_SIMPLEX	Python：cv.FONT_HERSHEY_SCRIPT_SIMPLEX
FONT_HERSHEY_SCRIPT_COMPLEX	Python：cv.FONT_HERSHEY_SCRIPT_COMPLEX
FONT_ITALIC	Python：cv.FONT_ITALIC

9.2、Code and actual effect display

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

```python
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()
```



```
(x=177, y=119) ~ R:255 G:255 B:255
```