Line segment drawing

When using OpenCV to process images, we sometimes need to draw line segments, rectangles, etc. on the image. OpenCV uses the line(dst, pt1, pt2, color, thickness=None, lineType=None, shift=None) function to draw line segments.

Parameter Description:

dst: output image.

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

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

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 is anti-aliasing

- Start Docker

After entering the Raspberry Pi 5 desktop, open a terminal and run the following command to start the container corresponding to Dofbot:

```
./Docker_Ros.sh
```

Access Jupyter Lab within Docker:

```
IP:9999 // Example: 192.168.1.11:9999
```

Code path:/root/Dofbot/4.opencv/3.draw_picture/04_draw_line.ipynb

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
newImageInfo = (600, 600, 3)
dst = np.zeros(newImageInfo,np.uint8)

# line
# Draw line segments 1 dst 2 begin 3 end 4 color
cv2.line(dst, (100,100), (450,300), (0,0,255))
# 5 line w
cv2.line(dst, (100,200), (400,200), (0,255,255), 10)
# 6 line type
cv2.line(dst, (100,300), (400,300), (0,255,0), 10, cv2.LINE_AA)

cv2.line(dst, (200,150), (50,250), (25,100,255))
cv2.line(dst, (50,250), (400,380), (25,100,255))
cv2.line(dst, (400,380), (200,150), (25,100,255))

# cv2.imshow('dst',dst)
#cv2.waitKey(0)
dst = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)
plt.imshow(dst)
```

```
    plt.show()
```