1.3.4 Drawing line segment

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

Parameter Description

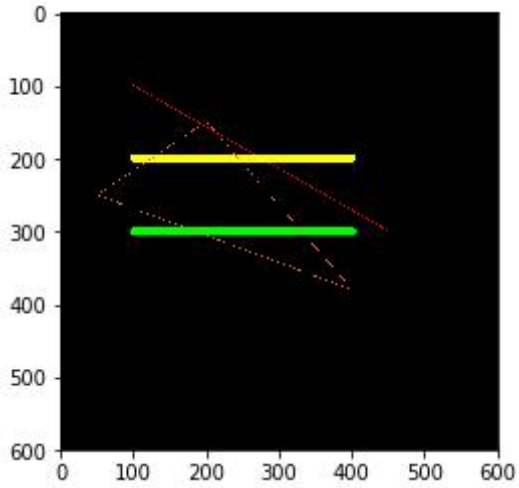
- dst: output image.
- pt1, pt2: required parameters, pt1 is the starting point, pt2 is the ending point.
- color: required parameters, be used to set color of line segment.
- Thickness: optional parameters, be used to set weight of line segment.

lineType: optional parameters, be used to set the type of line segment, 8 (8 adjacent connection line-default), 4 (4 adjacent connection line) and cv2.LINE_AA are anti-aliased

Path: /home/jetson/Dofbot\4.opencv\3.IP_Draw_text_line_segments\ 04_Drawing line segment.ipynb

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

newImageInfo = (600, 600, 3)
dst = np.zeros(newImageInfo,np.uint8)

# line
# Drawing line segment 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()
```

After running the following program, a picture will be displayed in the jupyterLab control interface, as shown below.


0
100
200
300
400
500
600
0
100
200
300
400
500
600