3、Open Source CV图片处理和绘制文字线段

3.1、OpenCV图像灰度化处理

1、图像灰度化

彩色图像转化为灰度图像的过程是图像的灰度化处理。彩色图像中的每个像素的颜色由R，G，B三个分量决定，而每个分量中可取值0-255，这样一个像素点可以有1600多万（256*256*256=1677256）的颜色的变化范围。而灰度图像是R，G，B三个分量相同的一种特殊的彩色图像，其中一个像素点的变化范围为256种，所以在数字图像处理中一般将各种格式的图像转化为灰度图像以使后续的图像的计算量少一些。灰度图像的描述与彩色图像一样仍然反映了整副图像的整体和局部的色度和高亮等级的分布和特征。

2、图像灰度化处理

灰度化处理就是将一幅色彩图像转化为灰度图像的过程。彩色图像分为R，G，B三个分量，分别显示出红绿蓝等各种颜色，灰度化就是使彩色的R，G，B分量相等的过程。灰度值大的像素点比较亮（像素值最大为255，为白色），反之比较暗（像素最下为0，为黑色）。

图像灰度化核心思想是 R = G = B ，这个值也叫灰度值。

1)最大值法：使转化后的R，G，B得值等于转化前3个值中最大的一个，即：R=G=B=max（R，G，B）。这种方法转换的灰度图亮度很高。

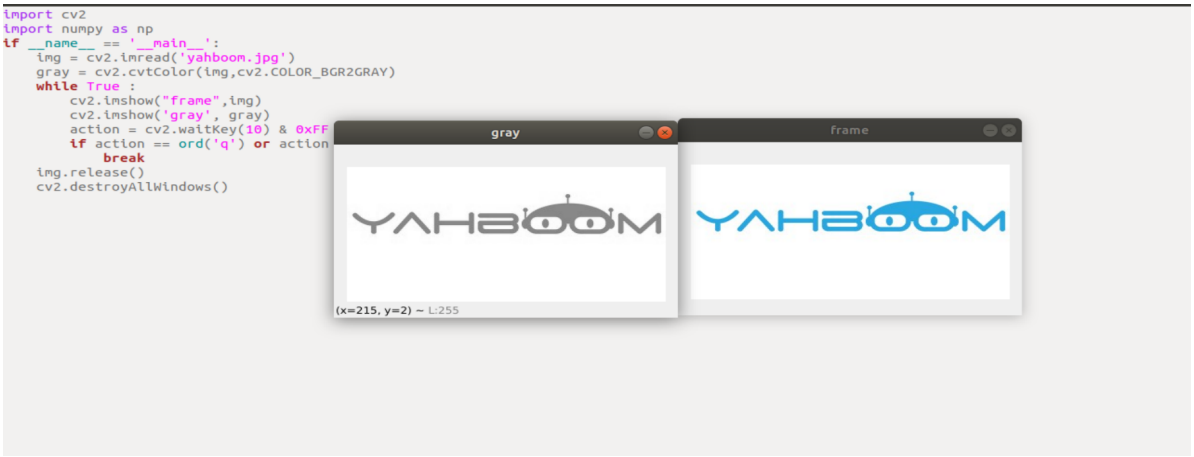
2)平均值法：是转化后R，G，B的值为转化前R,G,B的平均值。即：R=G=B=(R+G+B)/3。这种方法产生的灰度图像比较柔和。

在OpenCV中，用cv2.cvtColor(img1，cv2.COLOR_BGR2GRAY)来实现对图像进行灰度化处理

3、代码与实际效果展示

```
运行程序
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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、OpenCV图像二值化处理

1、二值化核心思想

设阈值，大于阈值的为0（黑色）或 255（白色），使图像称为黑白图。阈值可固定，也可以自适应阈值。自适应阈值一般为一点像素与这点为中序的区域像素平均值或者高斯分布加权和的比较，其中可以设置一个差值也可以不设置。

2、Python-OpenCV中提供了阈值（threshold）函数：cv2.threshold（src, threshold, maxValue, thresholdType）

参数含义：

src：原图像

threshold：当前阈值

maxVal：最大阈值，一般为255

thresholdType：阈值类型，一般有下面几个值

enum ThresholdTypes {
    THRESH_BINARY   = 0，#大于阈值的像素点的灰度值设定为maxValue(如8位灰度值最大为255)，灰度值小于阈值的像素点的灰度值设定为0。
    THRESH_BINARY_INV = 1，#大于阈值的像素点的灰度值设定为0，而小于该阈值的设定为maxValue。
    THRESH_TRUNC   = 2，#大于阈值的像素点的灰度值设定为0，而小于该阈值的设定为maxValue。
    THRESH_TOZERO   = 3，#像素点的灰度值小于该阈值的不进行任何改变，而大于该阈值的部分，其灰度值全部变为0。
    THRESH_TOZERO_INV = 4 #像素点的灰度值大于该阈值的不进行任何改变，像素点的灰度值小于该

阈值的，其灰度值全部变为0。
}

返回值：

retval：与参数thresh一致

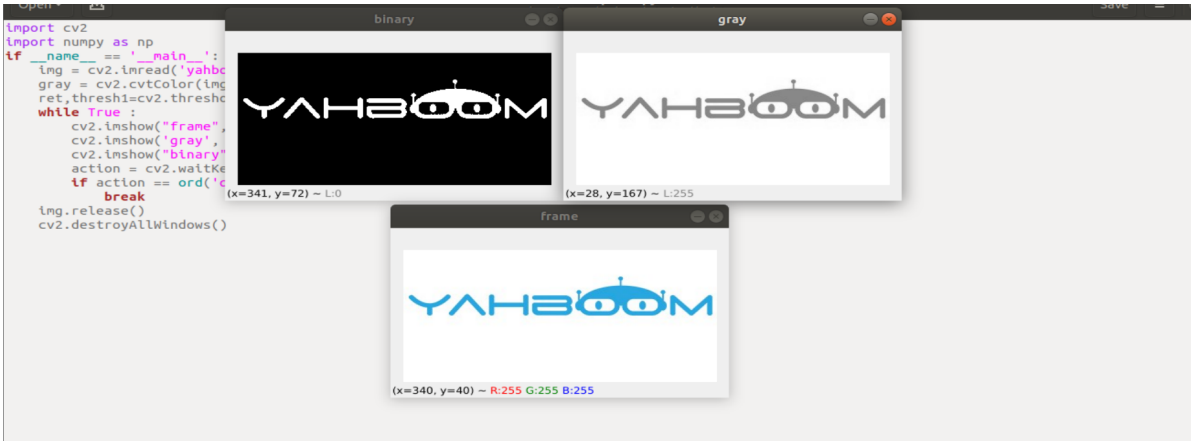
dst： 结果图像

注意：在进行二值化之前，我们需要把彩色图进行灰度化处理，得到灰度图。

3、代码与实际效果展示

```
运行程序
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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图像边缘检测

1、图像边缘检测的目的

在保留原有图像属性的情况下，显著减少图像的数据规模。目前有多种算法可以进行边缘检测，虽然Canny算法年代久远，但可以说它是边缘检测的一种标准算法，而且仍在研究中广泛使用。

2、Canny边缘检测算法

在目前常用的边缘检测方法中，Canny边缘检测算法是具有严格定义的，可以提供良好可靠检测的方法之一。由于它具有满足边缘检测的三个标准和实现过程简单的优势，成为边缘检测最流行的算法之一。

Canny边缘检测算法可以分为以下5个步骤：

（1）. 使用高斯滤波器，以平滑图像，滤除噪声

（2）. 计算图像中每个像素点的梯度强度和方向

（3）. 应用非极大值（Non-Maximum Suppression）抑制，以消除边缘检测带来的杂散响应

（4）. 应用双阈值（Double-Threshold）检测来确定真实的和潜在的边缘

（5）. 通过抑制孤立的弱边缘最终完成边缘检测

3、在OpenCV中我们如何实现呢？很简单，分为三个步骤走

（1）. 图像灰度化：gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

（2）. 高斯过滤（减噪处理）图像：GaussianBlur（src，ksize，sigmaX [，dst [，sigmaY [，borderType]]]）-> dst

参数含义：

src：输入的图像，一般是灰度后的图片

ksize：高斯内核大小

sigmaX： X方向上的高斯核标准偏差

sigmaY： Y方向上的高斯核标准差

dst：处理后的图像

（3）. Canny方法处理得到图像：edges=cv2.Canny(image, threshold1, threshold2[, apertureSize[, L2gradient]])

参数含义：

edges：计算得到的边缘图像

image： 计算得到的边缘图像，一般是高斯处理后得到的图像

threshold1： 处理过程中的第一个阈值

threshold2： 处理过程中的第二个阈值

apertureSize： Sobel 算子的孔径大小

L2gradient： 计算图像梯度幅度（gradient magnitude）的标识默认值为 False。如果为 True，则使用更精确的 L2 范数进行计算（即两个方向的导数的平方和再开方），否则使用 L1 范数（直接将两个方向导数的绝对值相加）。

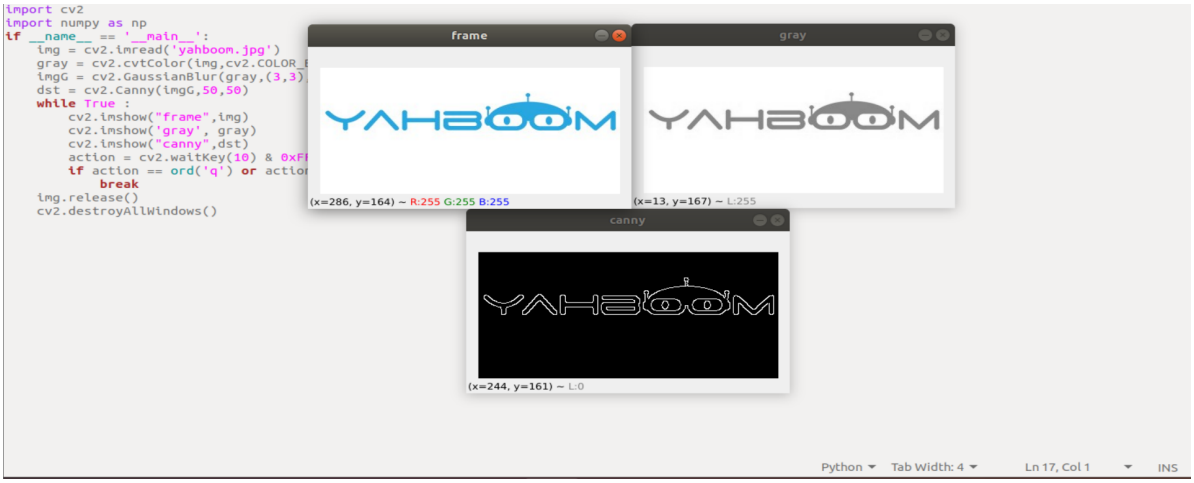
4、代码和实际效果展示

运行程序

```
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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线段绘制

1、在使用OpenCV处理图像时，我们有时候会需要在图像上画线段、矩形等。OpenCV中使用

cv2.line（dst，pt1，pt2，color，thickness=None，lineType=None，shift=None）函数进行线段的绘制。

参数含义：

dst：输出图像。

pt1，pt2：必选参数。线段的坐标点，分别表示起始点和终止点

color：必选参数。用于设置线段的颜色

thickness：可选参数。用于设置线段的宽度

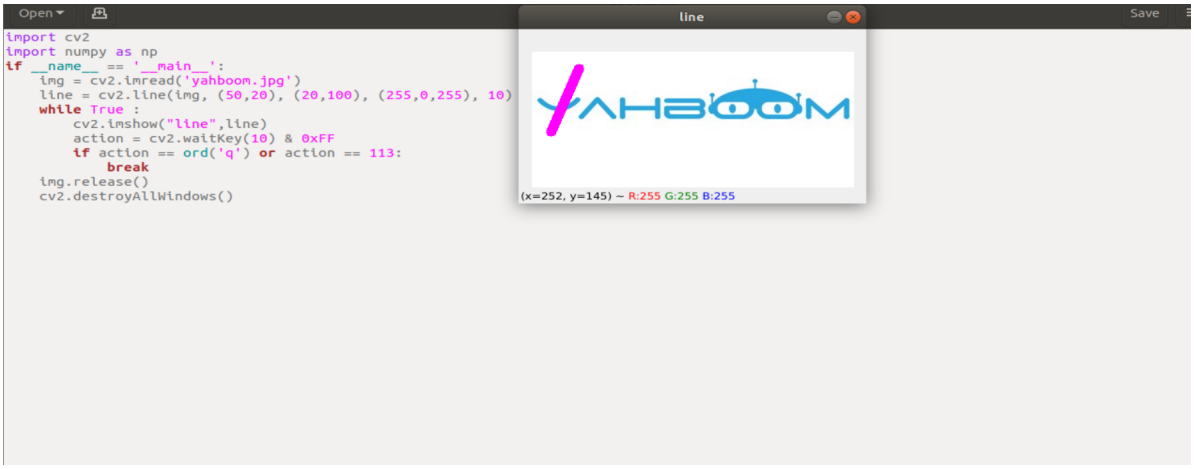
lineType：可选参数。用于设置线段的类型，可选8（8邻接连接线-默认）、4（4邻接连接线）和cv2.LINE_AA 为抗锯齿

2、代码与实际效果展示

运行程序

```
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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绘制矩形

1、在OpenCV中，绘制矩形用到的是

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

参数含义：

img：画布或者载体图像

pt1，pt2：必选参数。矩形的顶点，分别表示顶点与对角顶点，即矩形的左上角与右下角（这两个顶点可以确定一个唯一的矩形），可以理解成是对角线。

color：必选参数。用于设置矩形的颜色

thickness：可选参数。用于设置矩形边的宽度，当值为负数时，表示对矩形进行填充

lineType：可选参数。用于设置线段的类型，可选8（8邻接连接线-默认）、4（4邻接连接线）和cv2.LINE_AA 为抗锯齿

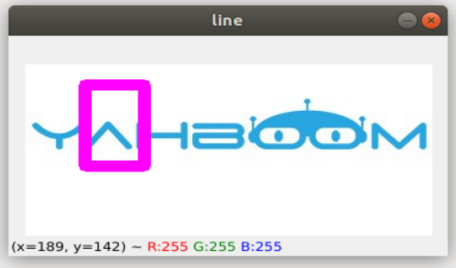
2、代码与效果展示

运行程序

```
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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、OpenCV绘制圆形

1、在OpenCV中，绘制圆形用到的是

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

参数含义：

img:画或者载体图像布

center：为圆心坐标，格式： （50,50）

radius：半径

thickness： 线条粗细。默认为1.如果-1则为填充实心

lineType：线条类型。默认是8，连接类型。如下表说明

参数	说明
cv2.FILLED	填充
cv2.LINE_4	4连接类型
cv2.LINE_8	8连接类型
cv2.LINE_AA	抗锯齿，该参数会让线条更平滑

2、代码和实际效果展示

运行程序

```
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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、OpenCV绘制椭圆

1、在OpenCV中，绘制椭圆用到的是

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

参数含义：

center：椭圆的中心点，（x，y）

axes：指的是短半径和长半径，（x，y）

StartAngle：圆弧起始角的角度

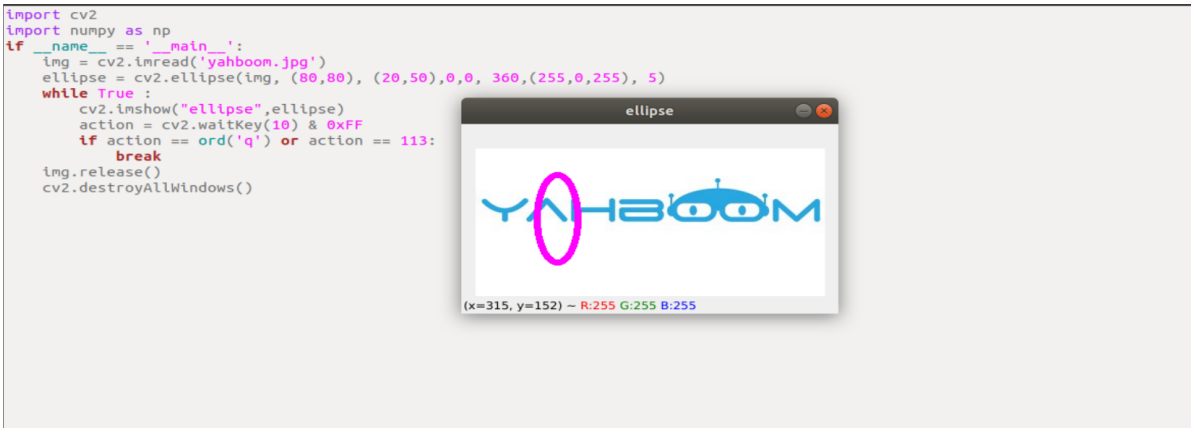
endAngle：圆弧终结角的角度

img、color、thickness、lineType可以参考圆的说明

2、代码和实际效果展示

```
运行程序
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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绘制多边形

1、在OpenCV中，绘制多边形用到的是

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

参数含义：

pts：多边形的顶点

isClosed：是否闭合。（True/False）

其他参数参照圆的绘制参数

2、代码与实际效果展示

```
运行程序
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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绘制文字

1、在OpenCV中，绘制文字用到的是

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

参数含义：

img：输入图像

str：绘制的文字

origin：左上角坐标(整数)，可以理解成文字是从哪里开始的

font：字体

size：字体大小

color：字体颜色

thickness：字体粗细

其中，字体可选

FONT_HERSHEY_SIMPLEX Python: cv.FONT_HERSHEY_SIMPLEX	正常大小sans-serif字体
FONT_HERSHEY_PLAIN Python: cv.FONT_HERSHEY_PLAIN	小尺寸sans-serif字体
FONT_HERSHEY_DUPLEX Python: cv.FONT_HERSHEY_DUPLEX	正常大小的sans-serif字体（比FONT_HERSHEY_SIMPLEX更复杂）
FONT_HERSHEY_COMPLEX Python: cv.FONT_HERSHEY_COMPLEX	正常大小的衬线字体
FONT_HERSHEY_TRIPLEX Python: cv.FONT_HERSHEY_TRIPLEX	正常大小的serif字体（比FONT_HERSHEY_COMPLEX更复杂）
FONT_HERSHEY_COMPLEX_SMALL Python: cv.FONT_HERSHEY_COMPLEX_SMALL	较小版本的FONT_HERSHEY_COMPLEX
FONT_HERSHEY_SCRIPT_SIMPLEX Python: cv.FONT_HERSHEY_SCRIPT_SIMPLEX	手写风格的字体
FONT_HERSHEY_SCRIPT_COMPLEX Python: cv.FONT_HERSHEY_SCRIPT_COMPLEX	更复杂的FONT_HERSHEY_SCRIPT_SIMPLEX变体
FONT_ITALIC Python: cv.FONT_ITALIC	标志为斜体字体

2、代码与实际效果展示

运行程序

```
python ~/yahboomcar_ws/src/yahboomcar_astra/scripts/opencv/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()
```