opencv图像处理常用函数[处理效果在具体实验中均有所体现，此处仅给出函数说明]：

cv2.imread(文件名, [显示控制参数])

作用：读入图像

[显示控制参数]：①cv.IMREAD\_COLOR：加载彩色图像，任何图像的透明度都会被忽视，为默认标志。

②cv.IMREAD\_GRAYSCALE：以灰度模式加载图像

③cv.IMREAD\_UNCHANGED：加载图像，包括alpha通道

示例：cv2.imread(‘sjtu.jpg’, cv.IMREAD\_COLOR)

cv.imshow(窗口名，图像对象)

作用：显示图像

示例：cv2.imshow(‘sjtu’,img)

cv2.imwrite(文件地址，文件名)

作用：保存图像名

示例：cv2.imwrite(‘/home/sjtu.jpg’,img)

cv2.waitKey(等待时间)

作用：等待键盘响应，毫秒级，0表示按下任意键终止

示例：cv2.waitKey(0)

cv2.destroyAllWindows()

作用：关闭所有图像窗口

示例：cv2.destroyAllWindows()

cv2.VideoCapture(参数)

作用：cv2.VideoCapture可以捕获摄像头，用数字来控制不同的设备，例如0,1；如果是视频文件，直接指定好路径即可。

示例：cv2.VideoCapture('test.mp4') 读取视频文件

cv2.VideoCapture(0) 捕获摄像头

cv2.split(图像对象)

作用：颜色通道提取，opencv是b,g,r而非r,g,b的格式

示例：b,g,r=cv2.split(img)

cv2.copyMakeBorder(图像对象,上边界,下边界,左边界,右边界,[边界参数])

作用：边界填充

[边界参数]：

①BORDER\_REPLICATE：复制法，也就是复制最边缘像素。

②BORDER\_REFLECT：反射法，对感兴趣的图像中的像素在两边进行复制例如：fedcba|abcdefgh|hgfedcb

③BORDER\_REFLECT\_101：反射法，也就是以最边缘像素为轴，对称，gfedcb|abcdefgh|gfedcba

④BORDER\_WRAP：外包装法cdefgh|abcdefgh|abcdefg

⑤BORDER\_CONSTANT：常量法，常数值填充

示例：

top\_size,bottom\_size,left\_size,right\_size = (50,50,50,50)

replicate = cv2.copyMakeBorder(img, top\_size, bottom\_size, left\_size, right\_size, borderType=cv2.BORDER\_REPLICATE)

cv2.threshold(src, thresh, maxval, type)

参数含义：

src：输入图，只能输入单通道图像，通常来说为灰度图

dst：输出图

thresh：阈值

maxval：当像素值超过了阈值（或者小于阈值，根据type来决定），所赋予的值

type：二值化操作的类型，包含以下5种类型： cv2.THRESH\_BINARY； cv2.THRESH\_BINARY\_INV； cv2.THRESH\_TRUNC； cv2.THRESH\_TOZERO；cv2.THRESH\_TOZERO\_INV

①cv2.THRESH\_BINARY 超过阈值部分取maxval（最大值），否则取0

②cv2.THRESH\_BINARY\_INV THRESH\_BINARY的反转

③cv2.THRESH\_TRUNC 大于阈值部分设为阈值，否则不变

④cv2.THRESH\_TOZERO 大于阈值部分不改变，否则设为0

⑤cv2.THRESH\_TOZERO\_INV THRESH\_TOZERO的反转

作用：根据阈值过滤图像

示例：ret, thresh = cv2.threshold(img\_gray, 127, 255, cv2.THRESH\_BINARY)

注：处理后得到的图像对象是thresh

cv2.blur(图像对象,卷积核) //均值滤波，进行简单的平均卷积操作

cv2.boxFilter(图像对象,目标图像深度,卷积核, normalize属性) //方框滤波，基本和均值一样，可以选择归一化

cv2.GaussianBlur(图像对象,卷积核, x方向方差) //高斯滤波，高斯模糊的卷积核里的数值是满足高斯分布

cv2.medianBlur(图像对象,卷积核大小) //中值滤波，相当于用中值代替

参数说明：

①卷积核可看作进行处理的最小单元，是一个n×n的像素矩阵

②目标图像深度通常取-1

③normalize属性决定是否归一化处理，默认归一化

④x方向方差通常取1

作用：均为图像平滑处理函数

示例：

blur = cv2.blur(img, (3, 3))

cv2.boxFilter(img,-1,(3,3), normalize=True)

aussian = cv2.GaussianBlur(img, (5, 5), 1)

median = cv2.medianBlur(img, 5)

cv2.erode(图像对象,卷积核,迭代次数)

参数说明：

①卷积核可看作进行处理的最小单元，是一个n×n的像素矩阵

②迭代次数默认值为1，迭代次数越多，腐蚀得越多

作用：对图像进行形态学腐蚀操作，常被用作去除图像毛刺

示例：

kernel = np.ones((3,3),np.uint8) //需先构造一个卷积核

erosion = cv2.erode(img,kernel,iterations = 1)

cv2.erode(图像对象,卷积核,迭代次数)

作用：对图像进行形态学膨胀操作，可用于加粗图像中的文字

示例：

kernel = np.ones((3,3),np.uint8)

dige\_erosion = cv2.erode(img,kernel,iterations = 1)

cv2.Canny(图像对象,minVal,maxVal)

参数说明：



作用：对图像进行边缘检测

示例：v1= cv2.Canny(img,80,150)

cv2.pyrUp(图像对象)

cv2.pyrDown(图像对象)

作用：高斯金字塔，pyrUp可用于放大图像，pyrDown可用于缩小图像（会丢失一部分像素信息），每次操作会导致图像的长和宽放大或缩小2倍

示例：up=cv2.pyrUp(img) down=cv2.pyrDown(img)

cv2.findContours(图像对象,mode,method)

参数说明：

①mode:轮廓检索模式

RETR\_EXTERNAL：只检索最外面的轮廓；

RETR\_LIST：检索所有的轮廓，并将其保存到一条链表当中；

RETR\_CCOMP：检索所有的轮廓，并将他们组织为两层：顶层是各部分的外部边界，第二层是空洞的边界;

RETR\_TREE：检索所有的轮廓，并重构嵌套轮廓的整个层次;

②method:轮廓逼近方法

CHAIN\_APPROX\_NONE：以Freeman链码的方式输出轮廓，所有其他方法输出多边形（顶点的序列）。

CHAIN\_APPROX\_SIMPLE:压缩水平的、垂直的和斜的部分，也就是，函数只保留他们的终点部分。

作用：得到图像轮廓

示例：

cv2.findContours(mask,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE) //mask为二值图像