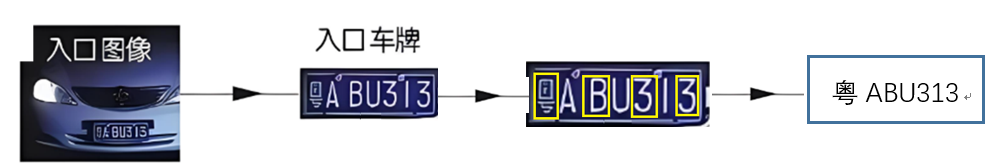
**实验二 图像处理—-车牌识别（提取车牌）**

# 车牌识别原理

车牌识别主要包括三个流程：

1. 提取车牌：将车牌从复杂的背景中提取出来
2. 拆分字符：将车牌拆分成一个个独立的字符
3. 识别字符：识别从车牌上提取的一个个字符

## 提取车牌

提取车牌需要完成一系列的去噪（滤波）、色彩空间转换等操作。

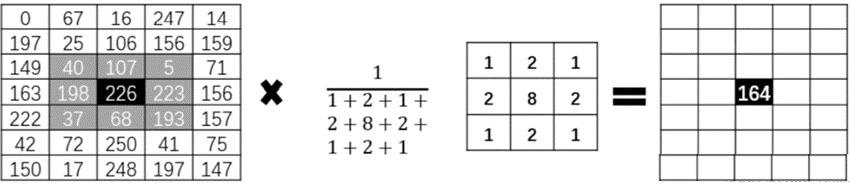
### 滤波与高斯滤波

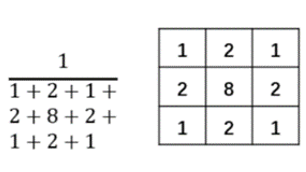
滤波的目的的去除图像内的噪声信息，对图像进行平滑处理。例如：一幅图像中如果一个像素点与周围像素点的像素值差异较大，就将其调整为周围临近像素点像素值的近似值（如均值）。平滑处理通常伴随图像模糊操作，因此平滑处理有时也被称为图像模糊处理(Blurring Images)。图像平滑处理==图像模糊处理==图像滤波==图像质量增强。

取近似值的方法很多，主要有均值滤波、高斯滤波、中值滤波等方法。



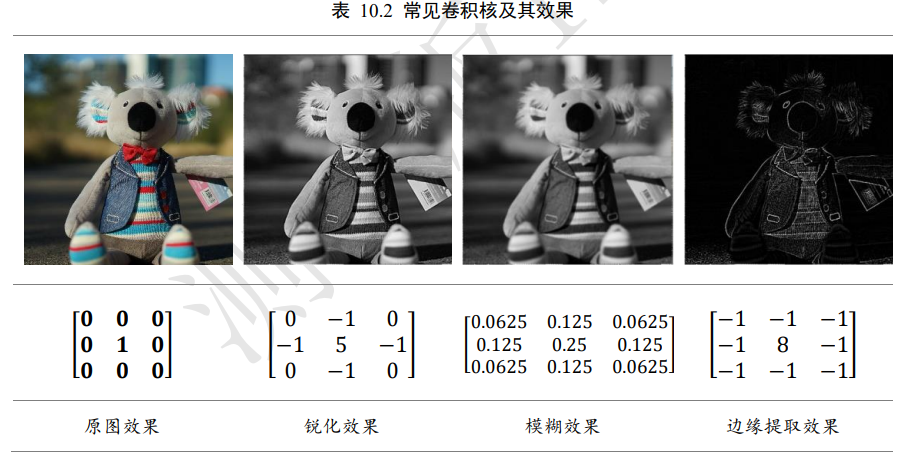
本实验使用高斯滤波。在进行均值滤波时，其邻域内每个像素的权重是相等的。在高斯滤波中，会将中心点的权重值加大，远离中心点的权重值减小，在此基础上计算邻域内各个像素值不同权重的和。例如：下图给出了高斯卷积的示例。



 卷积核为3\*3大小

卷积是一种基本的信号处理方法，在图像处理中，卷积操作是一种非常重要的操作。它可以提取图像的特征，如边缘、纹理等信息；可以实现图像的平滑和增强。均值滤波、高斯滤波等分别使用了不同的卷积核，从而实现不同的滤波效果。

下图展示了不同卷积核提取不同特征的效果，例如，边缘、模糊、锐化、角等特征。



常见卷积核及效果

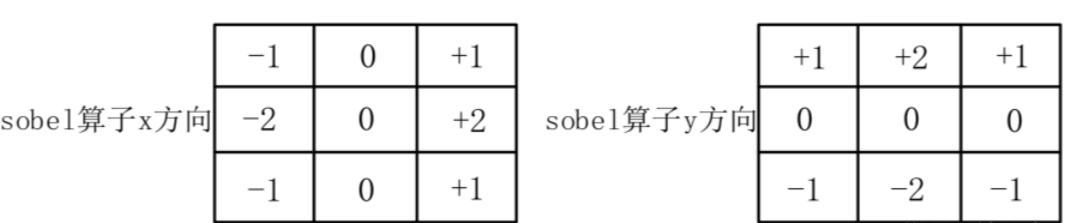
### 色彩空间转换

RGB色彩空间是一种比较常见的色彩空间，除此之外比较常见的还有GRAY（灰度图像）,HSV色彩空间、HLS色彩空间等。每个色彩空间都有自己擅长处理的问题，要针对处理的问题，选用不同的色彩空间。例如HSV色彩空间适合找到图像中的皮肤；灰度空间适合进行图像的特征提取；只考虑形状特征时，可以将色彩空间的图像转化为二值图像。

### 边缘检测

边缘检测通过标识数字图像中亮度变化明显的点来提取图像中的轮廓和边界信息。常见的边缘检测算法有：Sobel、Canny、Prewitt等。

本实验使用Sobel算法。我们知道边缘点其实就是图像中灰度跳变剧烈的点，Sobel算法通过计算图像像素灰度值的变化（梯度）来识别边缘。使用两个卷积核(Sobel核)对图像进行卷积运算，分别计算图像在水平和垂直方向上的梯度值（分别用水平和垂直方向卷积核(下图)与图像进行卷积操作）。然后，根据梯度的幅值来判断像素是否属于边缘。

****

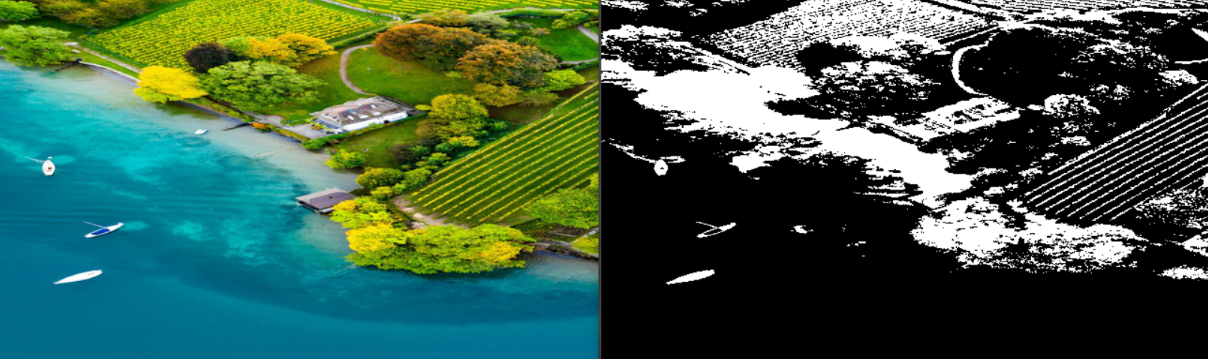
下图展示了水平方向和垂直方向上的梯度图像。



然后将得到的两幅梯度图像进行相加，就可以得到完整的轮廓图像。

### 二值化处理

图像的二值化通常用于分割图像中的目标对象或区域，将它们与背景分开，实现将256位的灰度图、或RGB的彩色图像转换成2位的黑白图像的过程。在这个过程中，256位的灰度图，共有256级，变成黑白图像后，只有2级（通常是0（黑色）或255（白色））。这里需要定义一个值，以此值为界限，大于这个值就为白，小于这个值为黑，这个定义的值，就是阈值。



Sobel边缘检测后，我们需要定义一个阈值，将其处理为二值图像。cv2.threshold是[OpenCV](https://so.csdn.net/so/search?q=OpenCV&spm=1001.2101.3001.7020)中用于进行图像二值化的函数，它的作用是将输入图像的像素值转换为两个可能的值之一，0（黑色）或255（白色）。

### 闭运算与开运算

开运算（Opening）和闭运算（Closing）是常用的图像处理操作，通常用于去除图像中

的噪声、连接物体、分离物体等操作。它们分别由两个基本操作组成：腐蚀（Erosion）和膨胀（Dilation）。

* 腐蚀：实现去噪、元素分割等功能。

* 膨胀：与腐蚀作用相反，对图像的边界进行扩张。如果两个对象距离较近，那么经过膨胀操作后，两个对象可能会连通在一起，因此膨胀有利于填补图像分割后图像内的空白处。

将腐蚀操作和膨胀操作进行组合，就可以实现开运算、闭运算等不同形态学处理。

* 闭运算：先膨胀后腐蚀的操作，有助于关闭物体内部的小孔或去除物体上的小黑点，将不同的前景对象进行连接。使用cv2.morphologyEx配合参数cv2.MORPH\_CLOSE完成。



* 开运算：先腐蚀后膨胀的操作，主要作用是去除图像中的小白噪声，以及分离接触在一起的物体。使用cv2.morphologyEx配合参数cv2.MORPH\_OPEN完成



# 实验内容及要求

1. **实验内容：**

按照下面给出的编程步骤，利用opencv完成车牌的提取，提示：可将车牌提取程序封装程一个函数。

1. 自己准备一张带有车牌的原始图像，读取原始图像
2. 对原始图像进行滤波处理，使用高斯滤波完成，imshow查看滤波结果
3. 将经过步骤(2)处理过的图像进行色彩空间转换（BGR🡪GRAY）,imshow查看灰度图像效果。
4. 提取图像边缘，主要是车牌和其中字符的边缘，使用Sobel算子完成。
5. 对（4）的图像进行二值化处理，使用threshold完成。
6. 先闭运算，再开运算，使得将分散的各个字符连在一起，构成一个整体，再使用开运算去除图像的噪声。查看处理结果
7. 试用中值滤波算法再次去除图像内的噪声
8. 利用findContours查找图像内的所有轮廓。imshow查看
9. *For循环逐个遍历轮廓，筛选出属于车牌的轮廓*，可以将宽高比大于3（ 标准车牌尺寸,宽度45cm,高度15cm）作为车牌轮廓的特征。
10. 显示提取的车牌。
11. **上传要求**
12. 运行效果图
13. 源程序文件