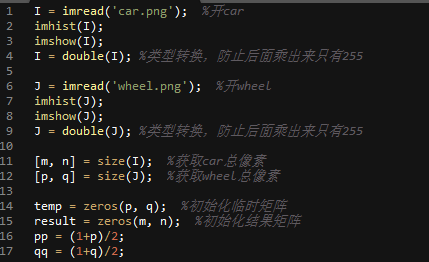
15331346 严晓珊

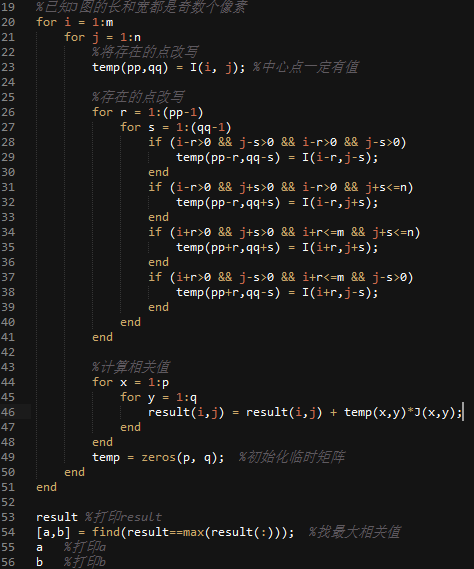
**相关值**

**算法描述：**

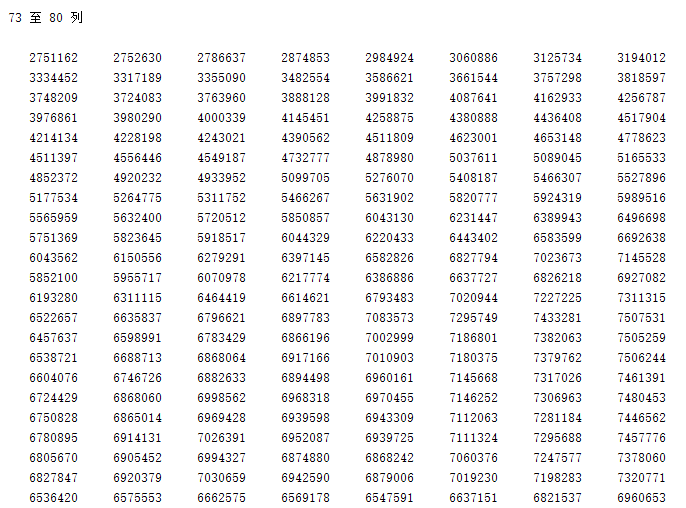
先分别读入car图像和wheel模板。考虑到图像矩阵都是unit8数据类型，也就是说在接下来计算相关值加法乘法的时候极有可能停在255。所以先将两个图像矩阵转换成double类型。先计算wheel矩阵的行数和列数，确定都是奇数之后进行下一步操作。创建一个跟wheel一样大的数组temp，并初始化为0，是用于提取car图像中的像素，以及后来跟wheel矩阵作乘法加法运算的。再创建一个result的数组，和car一样大，用于存放相关值。在car矩阵中遍历每一个像素，以当前像素点为中心点，往上下左右扩展选择像素点，填补temp矩阵。遇到边缘取不到的地方，直接用0来代替。当填充完temp后，再遍历temp，使得对应的位置与wheel矩阵对应的位置相乘相加，计算出相关值并记录在result相应的位置。对于取得最大相关值，可以使用find函数进行寻找，并且找到最大相关值的位置。

**代码：**

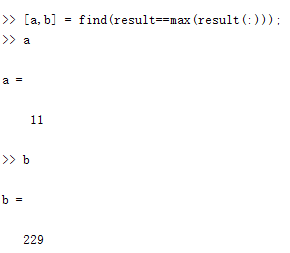




**结果：**（1）car图像的相关值矩阵result节选如下



（2）最大相关值的位置(x,y)

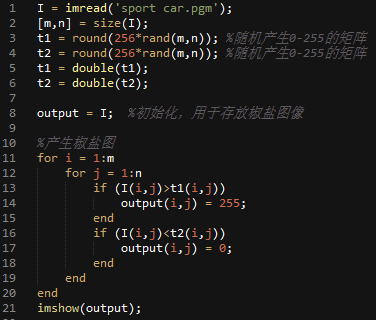


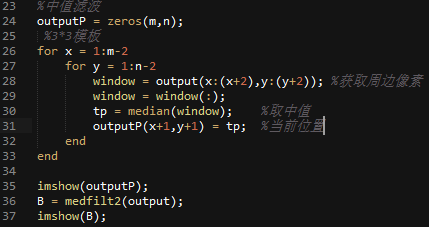
**椒盐和中值滤波**

**算法描述：**

读入sport car图像，确定长和宽。构建两个矩阵t1和t2，它们的大小和sport car图像的矩阵相同。利用随机数生成0-255的矩阵t1和t2。产生椒盐噪声，先构建一个output矩阵作为输出用，再按照指定条件改变特定像素的灰度值。中值滤波，利用3\*3模板，获取像素附近的点，然后在这些点里面取出中值，并记录到outputP矩阵的相应位置上。最后调用medfilt2来对比效果。

**代码：**





效果截图分析：

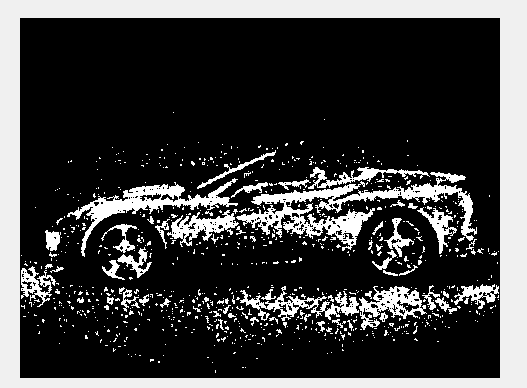
（1）原图：



（2）椒盐噪声图：跟原图比，有些地方变亮了，有些地方变暗了。



（3）中值滤波图：噪声变集中了，亮的地方更亮了，没有特别暗的地方。但是减噪完之后，轮廓变得更明显。



（4）medfilt2图：系统自带函数，在色彩分布上更柔和，不会像自创中值滤波函数的效果那样有些地方过于亮。噪声的分布变集中，变少了。

