

## 1 方法描述

### 1.1 相关检测

相关检测要求根据目标图像和模板图像的相关性检测出模板图像在目标图像上的匹配位置。它以模板图像的像素作为卷积核，在目标图像上根据卷积公式进行相关运算，并取得分高的区域作为检测区域。

在实验过程中，我分别使用了(1)-(3)三种卷积公式，将模板图像生成的卷积核与原图进行相关运算。在通过相关运算生成得分图后，我采用贪心算法提取检测框，首先提取全图最高得分，记为  $\max\_score$ 。每次选择得分最高得分的位置，如果该位置的得分大于  $\max\_score * \lambda$  ( $\lambda = 0.5, 0.7, 0.9$  或其它数字) 则将以该位置为中心的检测框计入检测结果，并在得分图上将该位置为中心的检测框对应的位置的得分置零，继续迭代，直到最高得分小于  $\max\_score * \lambda$ ，并结束操作。

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] * F[i + u, j + v] \quad (1)$$

$$G[i, j] = \frac{\sum_{u=-k}^k \sum_{v=-k}^k H[u, v] * F[i + u, j + v]}{\sum_{u=-k}^k \sum_{v=-k}^k F[i + u, j + v]} \quad (2)$$

$$G[i, j] = \frac{\sum_{u=-k}^k \sum_{v=-k}^k H[u, v] * F[i + u, j + v]}{\sum_{u=-k}^k \sum_{v=-k}^k F[i + u, j + v]^2} \quad (3)$$

$$G[i, j] = \frac{\sum_{u=-k}^k \sum_{v=-k}^k H[u, v] * F[i + u, j + v]}{\sqrt{\sum_{u=-k}^k \sum_{v=-k}^k F[i + u, j + v]^2} \sqrt{\sum_{u=-k}^k \sum_{v=-k}^k H[i + u, j + v]^2}} \quad (4)$$

### 1.2 椒盐噪声与中值滤波

椒盐噪声是图像中的一种常见噪声，它由一些随机出现的白点或黑点构成。在本次实验中，我采用如下方法生成椒盐噪声。首先生成两个相互独立的在  $[0, 255]$  均匀分布的随机矩阵  $t1, t2$ ，然后根据(4)得到椒盐噪声图像。其中  $f_0(x, y)$  表示输入图像的灰度值。

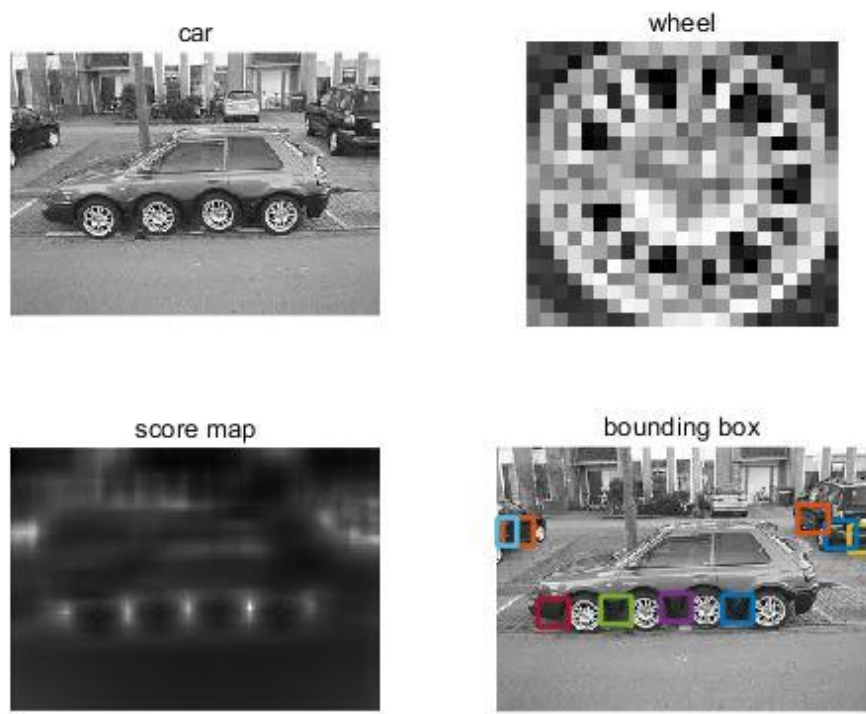
$$f(x, y) = \begin{cases} 255 & \text{if } f_0(x, y) > t_1(x, y) \\ 0 & \text{if } f_0(x, y) < t_2(x, y) \\ f_0(x, y) & \text{otherwise} \end{cases} \quad (5)$$

本次实验要求采用  $3 \times 3$  的窗口，实现中值滤波。中值滤波的原理比较简单，对于每个像素点，在感受野的灰度值取中位数作为滤波后的结果即可。

## 2 实验结果

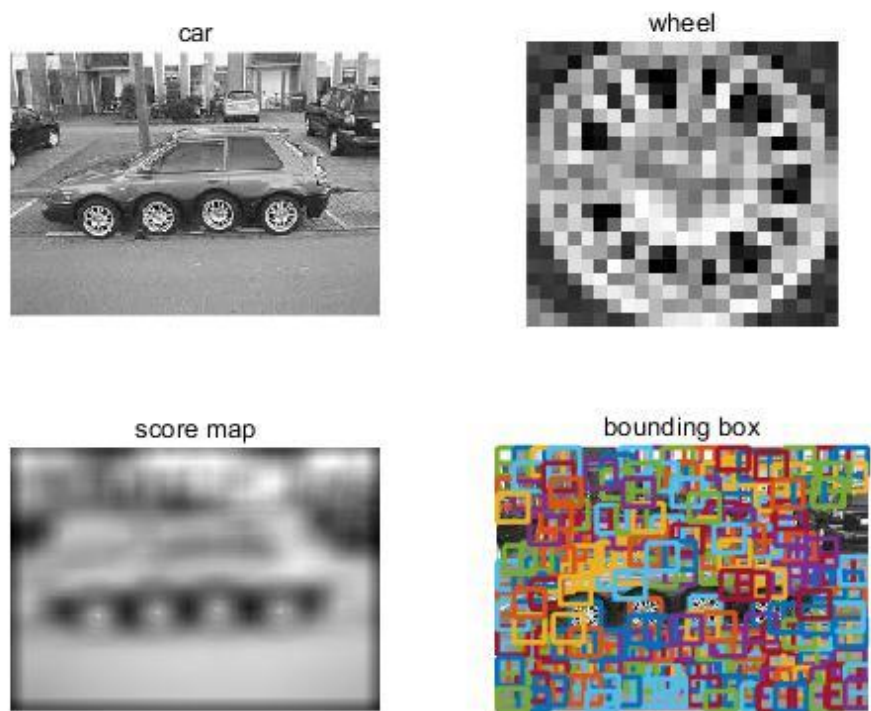
### 2.1 相关检测

我首先使用公式(3)，取  $\lambda=0.5$  进行相关性检测，结果如图 1 所示。可以看到检测框在轮子的附近。造成这个结果的主要原因是公式(3)进行了比较强的归一化，结果导致原图中灰度值较低的区域获得了较高的得分。由图可见，高得分区域在原图中都是比较暗的。我认为这样的一个公式并不能很好地检测出车轮。



(图 1)

之后我又使用了不加归一化的公式(1)，取 $\lambda=0.5$  进行相关性检测。可以看到高分区域集中在原图中较亮的区域，也就是灰度值较高的区域。检测结果的查全率很高，但查准率很低。如果我们调高 $\lambda$ 到 0.7，则会发现查全率下降，四个轮子都无法被检测到。再提高到 0.8，则一个轮子都不会被检测到。

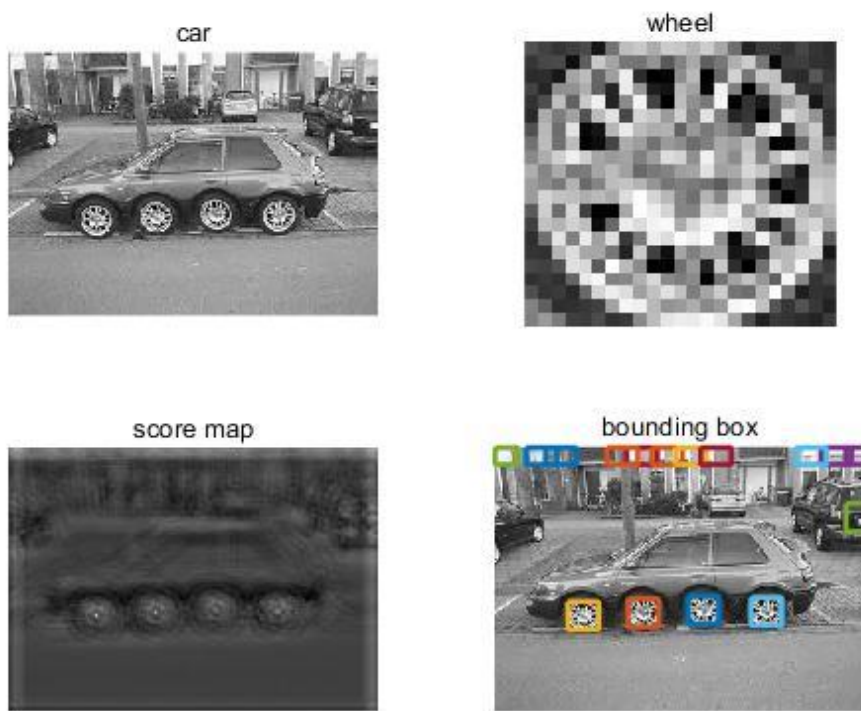


(图 2)

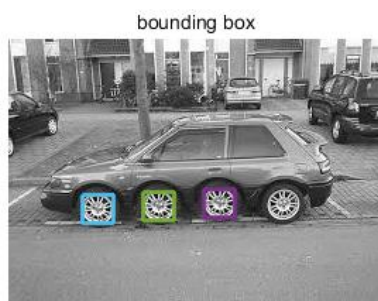


(图 3, 左边 $\lambda=0.7$ , 右边 $\lambda=0.8$ )

采用公式(2), 如图 4 可以看到在 $\lambda$ 取 0.5 的时候, 四个轮子都能被检测到。不过边缘的一些较亮的区域也被检测到了。如果调高 $\lambda$ 到 0.7, 如图 5 所示, 查准率会降低, 即有一个轮子没被检测到。不过查准率却提高了不少, 没有再出现错误的框。

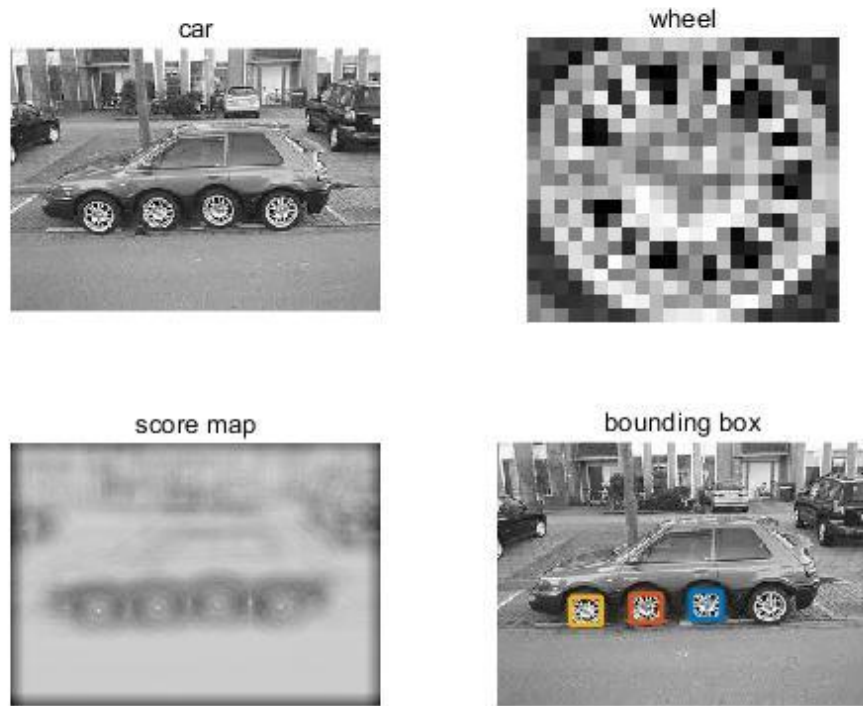


(图 4)



(图 5,  $\lambda=0.7$ )

最后采用公式(4)，如图 5 可以看到在 $\lambda$ 取 0.85 的时候可以检测出三个轮子，但将 $\lambda$ 继续调低也无法检测出第四个轮子。

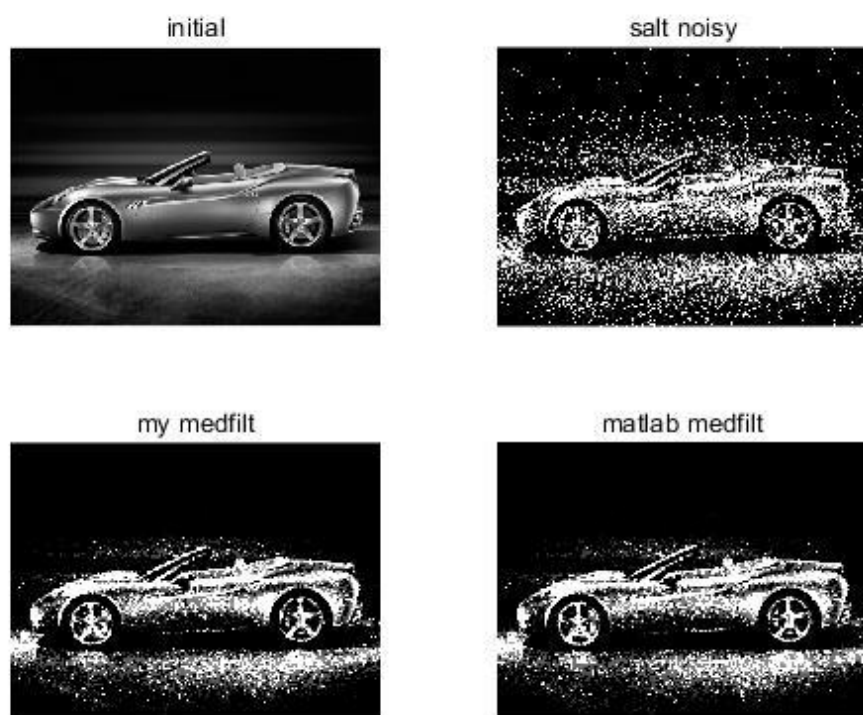


(图 6)

## 2.2 椒盐噪声与中值滤波

如图 6 为中值滤波与椒盐噪声的实验结果。首先我根据 1.2 节的方式生成椒盐噪声进行检测。可以看到我的实验结果与调用 `medfilt` 的实验结果基本相近。

我发现这样生成的椒盐噪声太密集，又尝试调用 `imnoise` 库生成椒盐噪声，如图 7 所示中值滤波效果良好。



(图 7)



(图 8)