

# 关于利用水印模式重复性攻击满屏水印的一些思考

我们所研究的谷歌算法的优点：

- 一旦获得了全局的 $W$ 和 $\alpha$ ，就能对新的一张加有同样水印的图快速进行去水印过程
- 对于大量的、具有相同水印模式的图片集，可以批量去水印

目前，我们考虑的都是单一水印处于图像中的某一位置的情况。

现代，有些人喜欢用满屏水印来保护图像，如下：



如果使用ps、美图秀秀，需要多次选择水印的区域，操作繁琐。

如何应用谷歌的算法来攻击满屏水印？

## • 思路一

- 单张图上的满屏水印构成了我们算法中要求的水印的重复性【加水印方式一致，即 $W$ 与 $\alpha$ 相同】
- 如果我们将图片分割成一块块小区域，每一块都作为独立的图像看，就约等于构建了一个具有相同模式水印的图片集

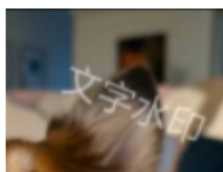


我裂开了

- 对这些小块图片运行我们的算法，得到 $W$ 和阿尔法
- 去水印

- 思路二

- 单张图上背景变化模式可能差不多，可能叠加不能相互抵消？（梯度 $W_m = \text{median}(\text{梯度})$ 估计可能不准？）
- 仍然考虑多张图片，每张图片上取一小块/2小块/.../ $k$ 小块区域作为cropped images，对所有的这些图像运行我们的算法



J3



J8

例如这张图上只随机取这俩，然后再用上其他图

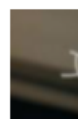
- 去水印

- 思考点

- 如果只利用单张图像就能获得较好的效果，意味着我们利用了满屏水印的重复性，能够对单张加了满屏水印的图像进行攻击
- 如果单张图像作为输入（思路一）不能获得较好的效果，每张图中取多少块小图作为输入合适（ $k$ 值）？
- 如何使用得到的 $W$ 和 $\alpha$ 去水印？
  - 方式一：分割图像，每一小块独立去水印，最后拼接回原图
  - 方式二：将 $W$ 和 $\alpha$ 扩展为满屏水印大小（原图大小），直接对大图去水印
  - ...
- 对于在图片边缘的，水印不完整的区域该怎么办？



栗子1



栗子2

- 对于加满屏水印的启发

- 重复的水印不要是相同的角度，可以对水印进行旋转，这意味着每一小块的 $W$ 不同（淦，好丑）



- 对重复的水印分别采用随机的透明度，这意味着每一小块的 $\alpha$ 不同（好像好看点）

