

# HW3 Watermarking

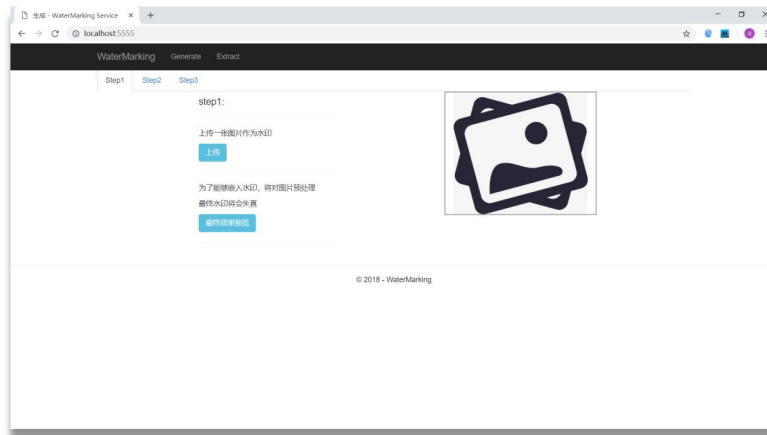
## 一、要求：

- Generation
  - Input
    - $I_c$ : a color image
    - $I_w$ : a watermark image
      - i.e., binary image with watermark information
      - or low resolution color image
      - or you can try QR code image
  - Output
    - $I'_c$ : a watermarked image
- Detection:
  - Input
    - $I_c$ : a watermarked color image
  - Output
    - $I_w$ : a watermark image
- two command lines
- bonus: a simple web based service

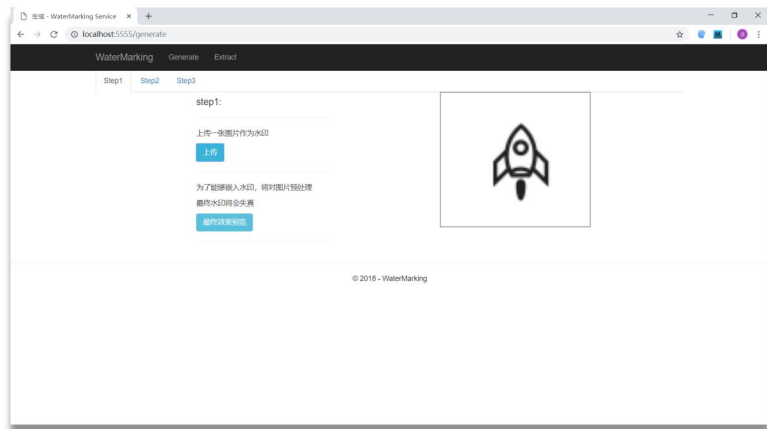
## 二、简单的网页水印服务实现：

- **Generation**
  - **Input**
    - **$I_w$ : a watermark image**

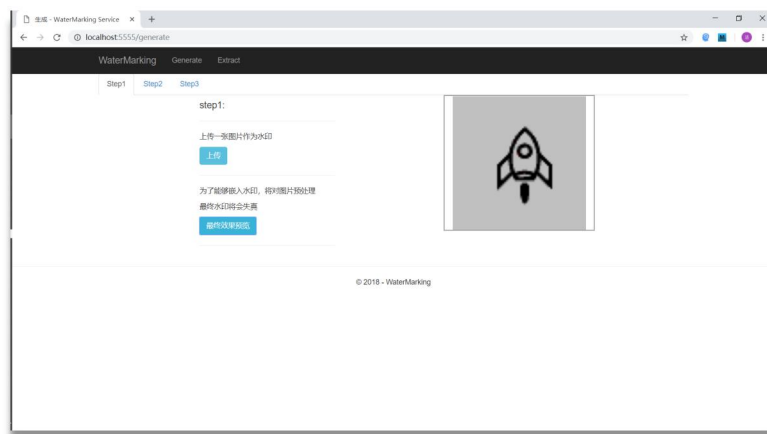
上传：



预览：

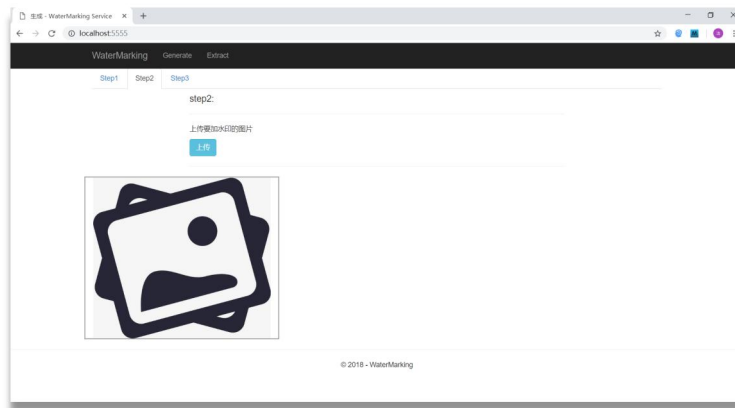


预览水印效果：

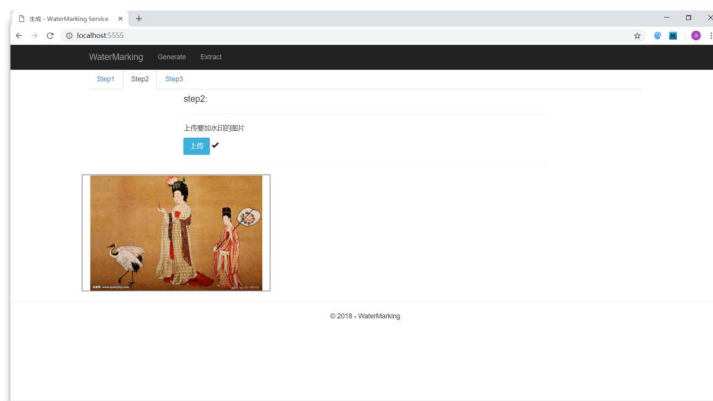


- **lc: a color image**

上传：



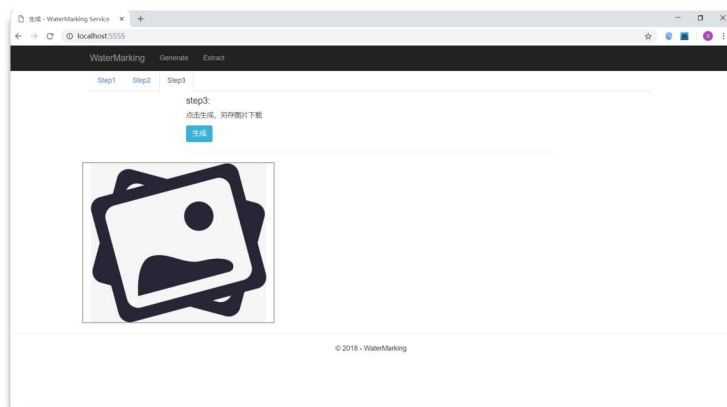
上传成功显示“√”，预览：



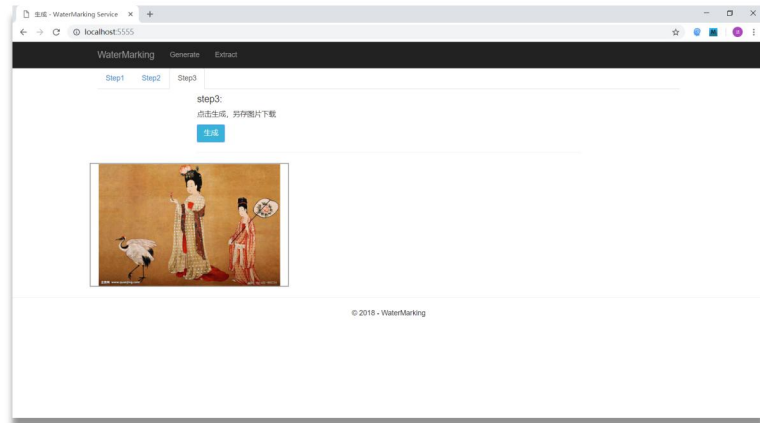
- **Output**

- **I' c: a watermarked image**

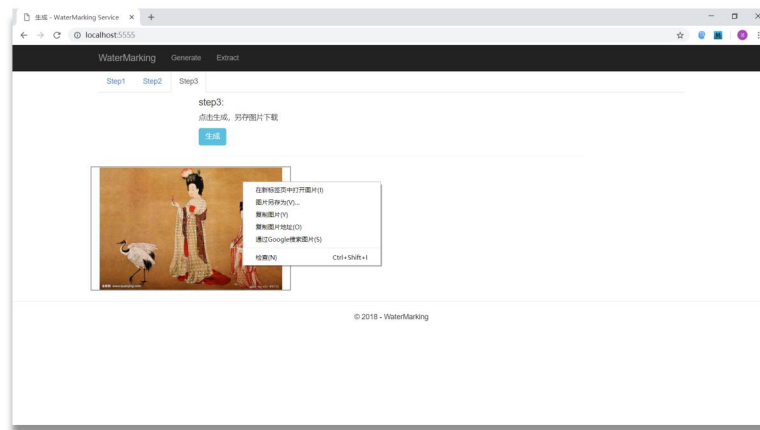
确认生成：



显示：



右键选中图片，另存图片：

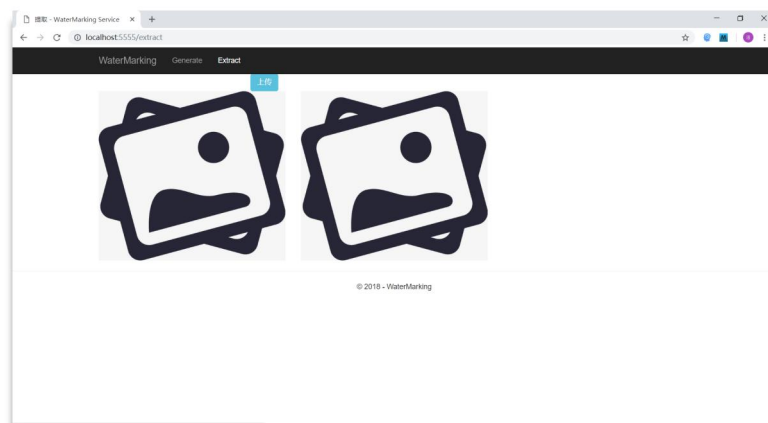


- **Detection:**

- **Input**

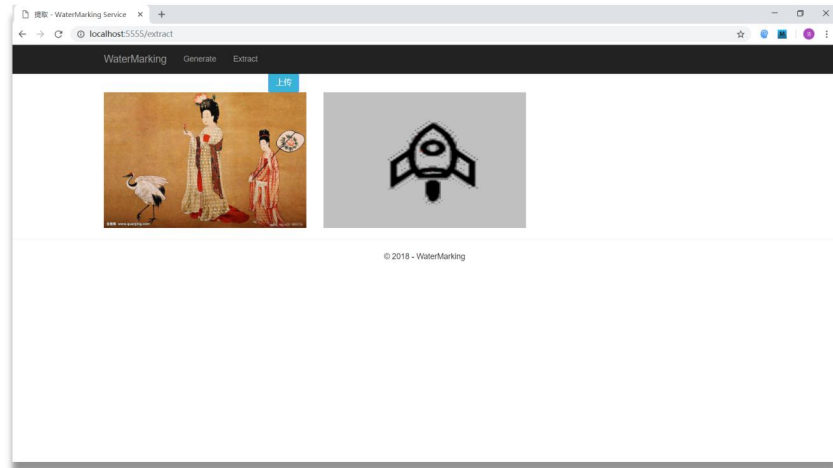
- **Ic: a watermarked color image**

上传：



- **Output**

- **lw: a watermark image**



- **运行方法:**

1. 配置: requirement.txt
2. CMD: python runserver.py
3. 按照指令打开网页。

## 三、技术细节

### 1. 框架:

Python Flask + HTML + CSS + JavaScript

### 2. 传输:

Ajax + FileReader +base64

#### 细节:

前端使用 `filereader`，读取方式为: `readAsDataURL`。此种方法将以 `base64` 的方式读取图片，并在字符串头部放置图片格式信息。将此字符串封装在一个 `json` 对象中，通过 `POST` 传递到后端。

后端接收并解析 base64。

经过后端处理的图片，加密成 base64 格式，并且加上 DataURL 头部，返回给前端。

#### 难点：

展开 DataURL，分析 base64 所在的位置，实现提取和封装。

### 3. 图片预览：

EventListener

#### 细节：

图片展示区域初始 src 为默认图片，当监听到上传组件有所改变，则检测上传文件的格式，做出相应反馈，符合要求，则将展示区域的 src 改为上传的文件。

生成水印图和提取水印也同理，只不过是在请求响应时，将 src 改为返回的图片资源。

#### 难点：

熟悉 JavaScript 的语法。

### 4. 水印：

原理与课上所讲的 stenography 相同，但是细节上还有：水印的尺寸调整。

## 四、发现与思考

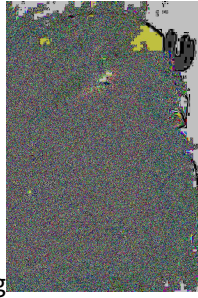
在最初实现水印算法时，遇到了以下问题：

输入：



实际提取的水印：

l'w.jpg

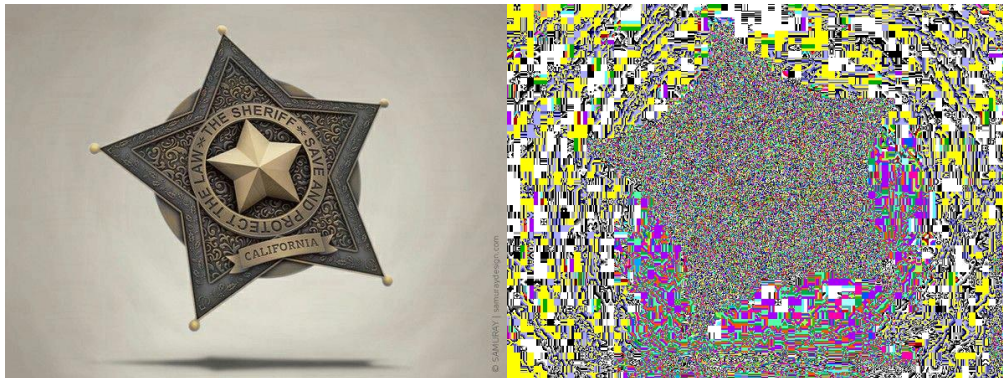


可以看到水印被噪声覆盖，有一部分可见，而这一部分是原图中几乎全黑的一部分。

不断分析检查后发现，这是因为图片格式均采用 jpeg 导致的。除了 jpeg2000，jpeg 格式都是有损压缩的。这就使得我们保存在图像低位的水印信息很容易受到干扰。改换成 png、bmp 格式则能正确提取信息。

为了解决这个问题，考虑减小水印尺寸，在图像中平铺，即使被扰动了，总能有地方显现出水印。

但还是遇到了问题，在一张对比柔和的图片中，插入 8\*8 张水印，提取结果都不能令人满意。此时，插入张数已经到达极限，可以看到提取的图片中有很多黄色斑点，这些就是压缩后的水印，但同时，我们完全失去了水印的信息（水印变成几个像素点了）。



综上，steganography 的方法所添加的水印确实能够安全地传递信息，但是在抵抗图片压缩上效果极差，在数字版权管理上的用途不是很大。