

## 一、实验内容

### 1、添加水印

输入一张原图，和一张水印图片，生成一张添加了水印的图片

### 2、解水印

输入一张加了水印的图片，提取出图片中的水印

### 3、实现上述功能的网页服务

## 二、具体实现

### （一）添加水印

#### 1、网页实现上传图片功能

Html 文件中使用表单标签，将网页上传的两张图片分别命名为 picture 和 watermark，传到 <http://localhost:5000/uploader> 服务器。

```
<div class="sub-main-w3">
  <form action = "http://localhost:5000/uploader" method = "POST" enctype = "multipart/form-data">
    <div class="form-style-agile">
      <label>
        <i class="fas fa-user"></i>
        Original image
      </label>
      <input type = "file" name = "picture" />
    </div>
    <div class="form-style-agile">
      <label>
        <i class="fas fa-unlock-alt"></i>
        Watermark image
      </label>
      <input type = "file" name = "watermark" />
    </div>
    <input type="submit" value="Submit">
  </form>
</div>
```

#### 2、在 uploader 路由器中，接受图片，并保存到本地

```
@app.route('/uploader', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        #save the photos uploaded to the page
        f1 = request.files['picture']
        f1.save("picture.png")
        f2=request.files['watermark']
        f2.save("watermark.png")
        #produce the watermarked picture
        watermark()
    return render_template('result.html')
```

将图片保存到本地根目录文件夹下，分别命名为 picture.png , watermark.png 。

### 3、对保存下来的图片进行添加水印处理

调用 watermark()函数：

```
def watermark():  
    image=np.array(Image.open('picture.png').resize((900,600)))  
    remain=np.mod(image,4)  
    image=image-remain;  
    mark=np.array(Image.open('watermark.png').resize((900,600)))/85  
    res=mark+image  
    resimg=Image.fromarray(res.astype('uint8'))  
    resimg.save("static/res.png", "png")
```

去原图 image 的最后两个 bit，存在 remain 中，再用 image 减去 remain，得到抹去最后两个 bit 的原图数据。

再将水印图片除以 85，添加到原来的图片中，即可得到加上了水印的图片。

### 4、跳转到结果界面

#### （二）提取水印

#### 1、网页实现上传图片功能

Html 文件中使用表单标签，将网页上传的带水印图片命名为 watermarked，传到 <http://localhost:5000/decoder> 服务器。

```
<div class="sub-main-w3">  
  <form action = "http://localhost:5000/decoder" method = "POST" enctype = "multipart/form-data">  
    <div class="form-style-agile">  
      <label>  
        <i class="fas fa-user"></i>  
        Watermarked image  
      </label>  
      <input type = "file" name = "watermarked" />  
    </div>  
    <input type="submit" value="Submit">  
  </form>  
</div>
```

#### 2、在 uploader 路由器中，接受图片，并保存到本地

```
@app.route('/decoder', methods=['GET', 'POST'])
def decoder():
    if request.method=='POST':
        f=request.files['watermarked']
        f.save("watermarked.png")
        decode()
    return render_template('result.html')
```

将图片保存到本地根目录文件夹下，分别命名为 watermarked.png。

### 3、提取图片中的水印

调用 decode() 函数

```
def decode():
    image=np.array(Image.open('watermarked.png'))
    water=np.mod(image,4)*85
    dec=Image.fromarray(water.astype('uint8'))
    dec.save("static//res.png","png")
```

将带水印的图片 water 对 4 取余数，然后\*85，即可得到水印图片。

### 4、跳转到结果界面

## 三、实验结果

### （一）添加水印

#### 1、上传图片

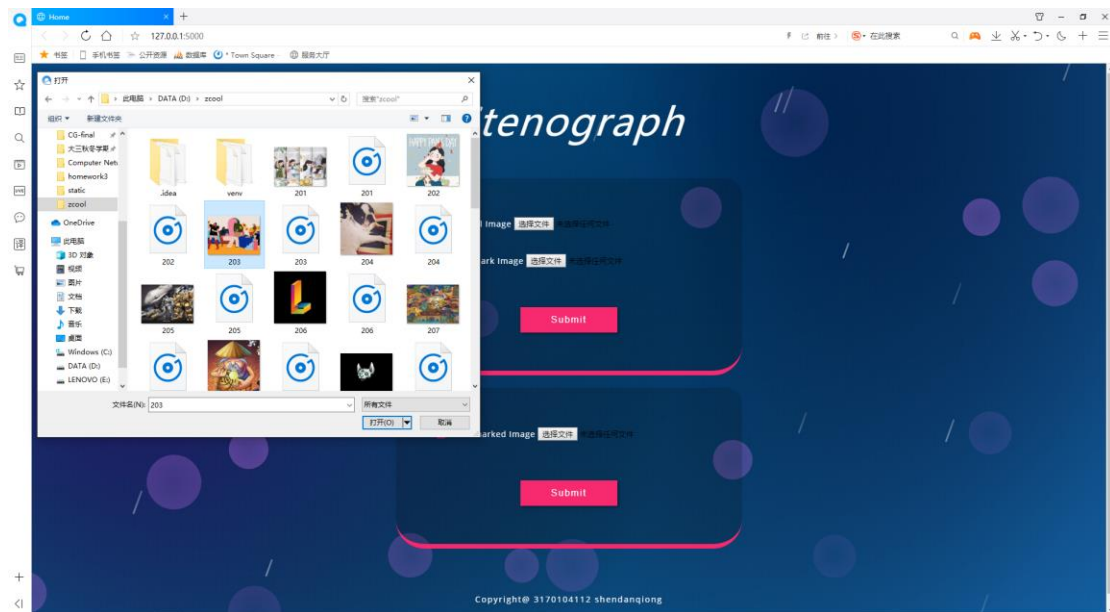
点击选择文件按钮，分别上传：

原图：



水印图：





2、 点击 submit 按钮，跳转到结果界面

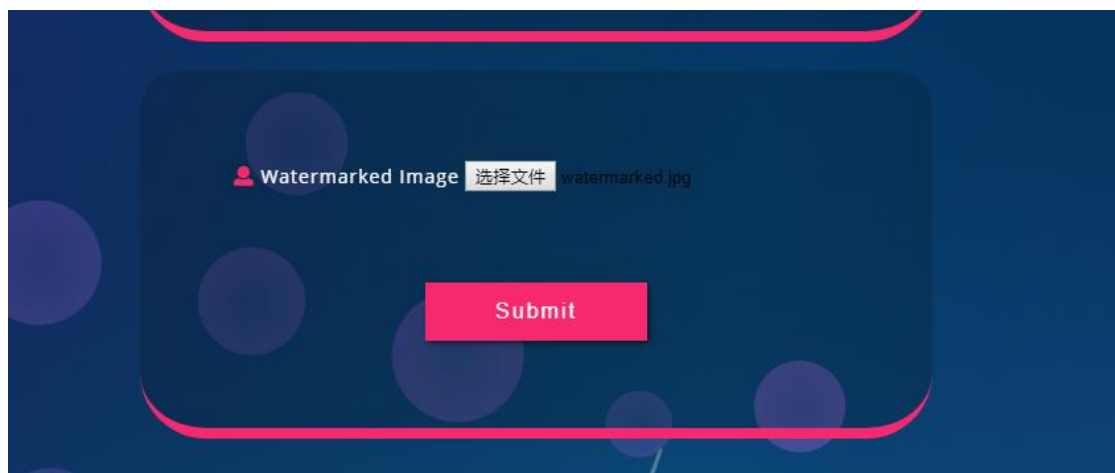


3、 点击下载按钮，将图片保存到本地



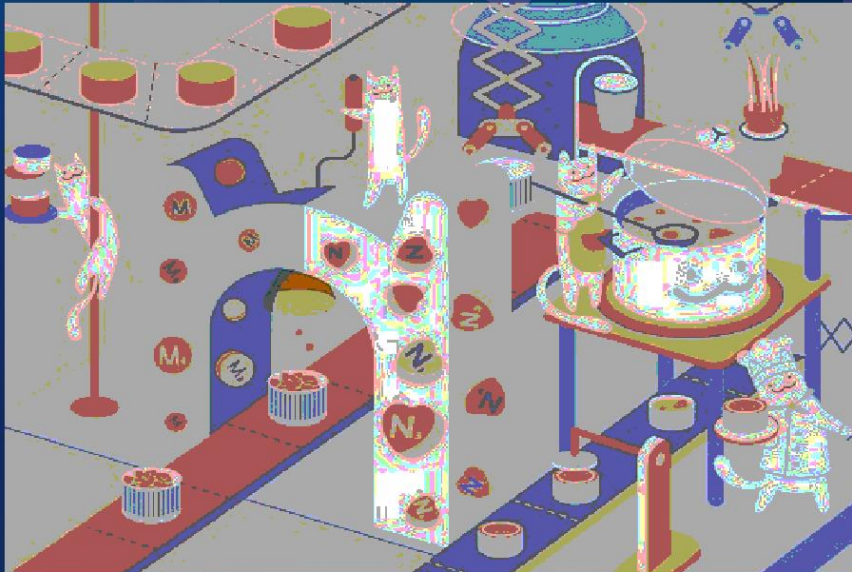
## （二） 提取水印

### 1、上传保存的加水印图片



### 2、点击 submit 呈现结果界面

# Stenograph



download

Copyright@ 3170104112 shendanqiong

## 四、结果分析

1、选择颜色简单的图片作为水印，提取效果较好  
水印原图：



提取后的水印：



2、颜色较复杂的图片容易失真