****

**图像处理课程设计**

**工程技术学院**

**题 目： 图片水印文字去除**

**学 号： 201808140323**

**姓 名：**  **彭威龙**

**专业班级： 计算机科学与技术183班**

**指导教师： 杨洁 职称： 副教授**

**2021 年 6 月 12 日**

图片文字水印去除

工程技术学院 计算机科学与技术183 彭威龙 指导教师：杨洁

**摘要：**随着计算机的普及和数字图像技术的广泛应用，数字图像处理已经成为计算机视觉领域的一个研究热点。图像修复是图像处理的重要组成部分，是对图像中受损部分进行信息填充的过程，尽可能的还原原始图像。

**关键词：**OpenCV；图像修复；图像处理；

# 目录

[目录 iii](#_Toc4929)

[1 引言 4](#_Toc24936)

[1.1 课题背景及意义 4](#_Toc14034)

[1.2 研究内容 4](#_Toc11045)

[2 系统设计与实现 5](#_Toc5094)

[2.1 设计说明 5](#_Toc24102)

[2.2 系统分析 5](#_Toc12493)

[2.3 图像预处理 5](#_Toc10196)

[2.3.1 灰度化 5](#_Toc11325)

[2.3.2 图像增强 6](#_Toc32565)

[2.3.3 滤波去噪 6](#_Toc20391)

[2.4 获取Mask 7](#_Toc1136)

[2.4.1 通过阈值获取Mask 7](#_Toc11560)

[2.4.2 形态学处理 8](#_Toc29052)

[2.5 图像修复 8](#_Toc28154)

[3 总结与展望 10](#_Toc19776)

[3.1 总结 10](#_Toc23965)

[3.2 展望 10](#_Toc23722)

# 引言

## 课题背景及意义

随着我国社会地进步和人民物质生活条件地不断改善，拍照摄影已成为人们生活中不可或缺的部分，无论是平时旅游还是举办重大活动，人们往往都会拿出手机拍照，记录下这有意义的时刻。而有些照片在拍摄时会有一些噪声腐蚀，也有一些照片图片会被加上水印，这对我们想要获得完整清晰的图片造成一定的影响，而图像修复则主要是修复这些图像，尽可能还原其原有的样子。

## 研究内容

本课题以VS2017为工具，用Opencv 实现文字水印去除，包含三个步骤:

预处理:处理图像，改善图像质量，为接下去的步骤提供便捷，压缩图像大小以减少图像数据量并提高操作效率。接着进行灰度化操作，简化和保留图像信息，提高运算速度；用滤波进行除燥

获取图像掩码Mask:通过阈值处理或手工选定获得图像掩码Mask，该Mask为接下去的图像修复提供修复范围。

图像修复:通过Opencv内置函数inpaint()，选择mask和合适的修复算法最终实现图像的修复，来和原图进行对比。

# 系统设计与实现

## 设计说明

本文采用对多种情况方法的分析和对比，来确定最佳处理效果

## 系统分析

本文利用VS软件研究各个模块的算法，完成编程，处理输入的图像并输出修复后的图像，整体逻辑如图 2‑1所示。

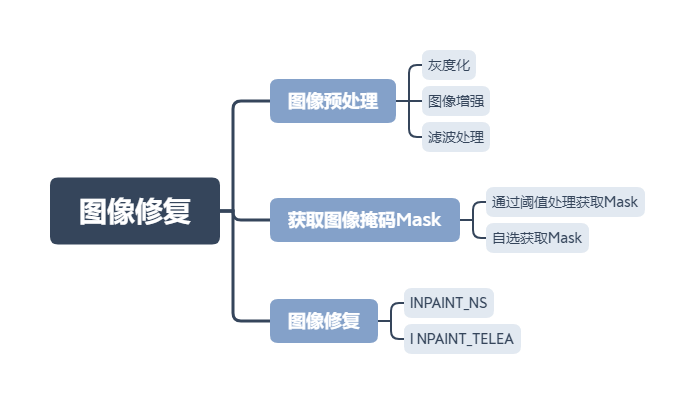


图 2‑1 系统流程图

本文系统分为三个大的步骤：图像预处理，获取图像掩码Mask和图像修复。预处理主要作用是减少噪声，让图像更为清晰，更好地辨别出水印，文字或其他干扰图片的事物，从而容易获取Mask来进行修复；获取Mask可以通过阈值来区分图像要保留的和要去除的文字水印，是inpaint()函数中必要的参数；图像修复选择一个合适的算法。

## 图像预处理

通常输入系统的牌照由于图像不清晰或包含较多的噪声干扰，会对图像的分辨率产生一定的影响。因而有必要提高图像质量，对图像进行一系列增强操作。经过预处理后便可以让图像内的各种元素更为清晰。

### 灰度化

灰度化就是将RGB彩色三通道数据变成单通道的灰度图，其作用可以减小数据量的处理。调用OpenCV的cvtColor（）函数，设置 COLOR\_BGR2GRAY参数将彩色原图转变成灰度图。灰度化操作后的效果图如图 2‑2所示。

|  |  |
| --- | --- |
| 原图  （a）原图 | 灰度图  （b）灰度图 |

图 2‑2 灰度化前后效果图

### 图像增强

直方图均衡化可以用来提升图像对比度和亮度，其原理是将图像中分布较多像素的灰度值进行加宽，合并像素值较少的灰度值以实现平均的分布，以此增强整体的对比度。调用OpenCV中的equalizeHist（）函数即可实现，效果如图 2‑3所示。

|  |  |
| --- | --- |
| 灰度图  （a）增强前 | 图片增强后  （b）增强后 |

图 2‑3 直方图均衡化效果图

### 滤波去噪

高斯滤波可以有效地消除噪声。其基本原理是使图像中的像素变得光滑，对邻域中位置不同的像素给定不一样的权重。二维分布由式（2-1）可见， *G*（*x*，*y*）为高斯核函数，（*x*，*y*）是点坐标，*σ*是标准差，其值越大，平滑的效果就越明显。

 （2-1）

与其他滤波相比，高斯滤波除了可以使图像平滑，还可以保存整体图像的灰度特征，利于找到水印，因此选用高斯滤波。调用OpenCV提供的GaussianBlur（）函数，选用3\*3大小的高斯核，滤波平滑后的效果如图 2‑4所示。



图 2‑4 高斯滤波效果图

## 获取Mask

Mask是图像修复中的关键部分,清晰准确地Mask能够让水印文字效果更为明显。

### 通过阈值获取Mask

通过调用threshold()函数，获取该图片的Mask。根据不同图片的类型和环境的不同选择不同 的阈值类型。阈值类型有:THRESH\_BINARY,THRESH\_BINARY\_INV,THRESH\_TRUNC,

THRESH\_TOZERO,THRESH\_TOZERO\_INV,THRESH\_OTSU,THRESH\_TRIANGLE.

图2-5是图2-4进行阈值处理后获得的Mask。



图2-5阈值处理后获取的Mask图

OTSU算法是自适应确定阈值的方法。它是根据图像中的灰度值，把图像划分成一个背景和一个前景。假设*k*是阈值，*u*0是前景中的像素数量在全图中的比例，*w*0是平均值，*u*1是背景中的像素数量在全图中的比例，*w*1是均值，则全图的平均值由式（2-2）可见，设立的目标函数由式（2-3）可见。OTSU算法目的就是使*g(t)*获取到全图的最大值。当*g*(*k*)取得最大值时，相应的*k*即为最佳的阈值。

 （2-2）

 （2-3）

调用threshold（）函数处理图像阈值，其中阈值处理参数选择THRESH\_BINARY\_INV和THRESH\_OTSU，第一个参数表示黑白二值反转。本文对原图最终采用的是THRESH\_OTSU。

### 形态学处理

通常，在获取Mask之后，由于阈值不够准确造成的误差，可以适当对Mask进行膨胀处理，增加Mask的面积。

腐蚀和膨胀是对白色部分（高亮部分）而言的，不是黑色部分。膨胀就是图像中的高亮部分进行膨胀，“领域扩张”。

对Mask区域进行膨胀处理后的图如2-6所示。



图2-6膨胀后的图

## 图像修复

通过自己划定选取，减小整张图片无关因素带来的干扰来提升修复的准确度。通过event创建事件，自己确定选区，然后通过选区内的Mask进行图像修复，如图2-7所示。



图2-7效果图

所选区域如图2-8所示，Mask如图2-9所示。

|  |  |
| --- | --- |
| roi  图2-8选区图 | Mask  图2-9Mask图 |

最后通过inpaint()用INPAINT\_NS修复算法修复后与原图的对比图如图2-10所示。

|  |  |
| --- | --- |
| 原图  （a）原图 | 修复后  （b）修复后 |

图 2‑10 修复前后对比图

# 总结与展望

## 总结

本文通过对图像进行预处理，对图像进行优化，使其更为清晰，通过阈值获取合适的Mask,再通过膨胀处理增加其部分面积，Mask的好坏直接影响到inpaint()的修复，在对不同图片时要选择合适的方法，争取取得较为准确地选区。

在刚开始进行对图片进行修复时，也出现过很多情况，比如修复不完整，造成这个的原因就是没有选择合适的阈值类型，需要通过改变阈值类型，选择较好的效果。此外，取图像掩码时，取整张图片往往会对一些无关要素进行修复，破坏了原图，而解决这个问题则是需要自己取部分区域进行修复。

总体来说对于一般的图片是能够进行修复，能做到一定的去水印文字效果。

## 展望

本文研究的主要任务是水印文字去除。由于条件和时间的限制，该研究仍存在许多不足，但大致的构造和基本的功能已经实现，本研究有待改进之处总结如下：1）并不能根据图片的类型自己选择合适的阈值类型，每次换一张图则要自己改一下类型；2）测试的图片较少，对部分较为复杂的图可能并不能很好的去除；3)以后可以自己增加掩码模板来让他自己识别去除，对特定的水印和文字进行去除;4)不同的修复算法还是有所区别，可以尝试改进。