

# 高级语言程序设计

## 实验报告

南开大学 工科实验班

姓名 金莫迪

学号 2312578

班级 3309

2024 年 5 月 12 日

# 目录

高级语言程序设计大作业实验报告 .....	3
一. 作业题目 .....	3
二. 开发软件 .....	3
三. 课题要求 .....	3
四. 主要流程 .....	4
<b>第三方库的环境配置：</b> .....	4
Motivation .....	4
图像预处理 .....	5
边缘检测 .....	6
去除背景 .....	7
自动旋转 .....	8
文字检测与擦除 .....	9
QT 图形化界面开发 .....	10
自适应图标大小 .....	10
侧边文件栏 .....	11
用户友好设计 .....	11
UI 设计 .....	11
五. 单元测试 .....	12
六. 收获 .....	13
1、 规范代码习惯 .....	13
2、 版本管理工具 git 的使用 .....	14
3、 openCV 库的使用 .....	14
4、 Tesseract 库的使用 .....	14
5、 环境配置 .....	14
6、 项目开发的细节 .....	15
7、 Qt 工具 .....	15

# 高级语言程序设计大作业实验报告

## 一. 作业题目

**NKU-experimenter: 基于 opencv 和 Tesseract 库的物理实验图像处理器**

利用 opencv 库和 Tesseract 库实现对物理实验图像的裁剪、旋转、擦除、增强。

## 二. 开发软件

Visual Studio 2022

QT Tool 6.7.0

Opencv 4.5

Tesseract

Git

## 三. 课题要求

- 1) 第三方库的环境配置
- 2) 实现对图片的预处理
- 3) 实现对图片的边缘检测
- 4) 实现图片的背景更换
- 5) 实现图片的降噪强处理

- 6) 实现图片的自动旋转
- 7) 实现图片的文字检测
- 8) 实现图片的文字擦除
- 9) 实现图形化界面的设计

## 四. 主要流程

### 第三方库的环境配置：

由于使用了第三方库，因此需要大量地配置环境，其中尤以 Tesseract 的配置最为艰辛，我经历了多次失败，期间尝试使用 cmake 进行配置，也最后失败。国内甚至没有一篇博客或文章介绍了最新版 Tesseract 的配置（各个文章提到的配制方法早在 2021 年就已经停止服务）最后，我通过查阅 Tesseract 的官方手册，使用 vcpkg 成功配置了环境。

### Motivation

在完成物理实验报告时，往往需要加入物理实验原理图，通常有两个办法：

- 1、使用 word 图形插入绘制
- 2、拍照或扫描物理实验书上的实验原理图并粘贴到实验报告文件中。

然而，这两种方法都面临问题。

第一种方法十分麻烦，word 绘图调整绘图元素之间的位置十分不便，而且，有些物理元素不适宜使用 word 绘制。

第二种方法同样面临着问题，一方面是直接拍照容易造成原理图扭曲，并且因为物理教材背景色不是白色，直接粘贴会显得十分突兀。另一方面，即使使用

扫描功能，也无法避免图片扭曲的问题，使用手机自带的文档抓取功能并不能很好地识别原理图（往往会把文字部分包含进来）。并且，使用扫描功能会丢失原理图色彩信息。

因此，我开发了本项目希望实现物理实验图像的识别、切割、处理、导出集一体的一站式操作。

## 图像预处理

通过使用 opencv 库，我们可以将一张图片转为矩阵格式（Mat），之后，我们将图片转为**灰度图**和**HSV 色域图**，这一部分是在我设计的 ImageCV 中的构造函数实现的，构造函数读取图片的绝对路径，之后利用 cvtColor 方法进行转换。

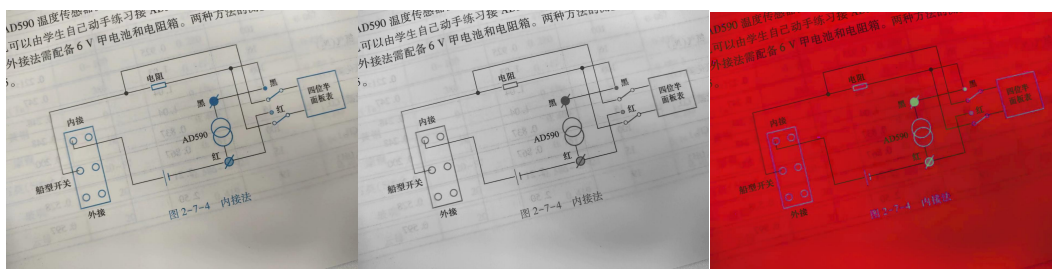


图 1 原图（左）灰度图（中）HSV 图（右）

注解：

**灰度图：**灰度图，Gray Scale Image 或是 Grey Scale Image，又称灰阶图。

把白色与黑色之间按对数关系分为若干等级，称为灰度。灰度分为 256 阶。

**HSV 色域：**HSV(Hue, Saturation, Value)是根据颜色的直观特性由 A. R. Smith 在 1978 年创建的一种颜色空间，也称六角锥体模型(Hexcone Model)。HSV 颜色模型是指 H、S、V 三维颜色空间中的一个可见光子集，它包含某个颜色域的所有颜色。生活中我们最常见的就是 RGB 色彩空间，即红绿蓝三个通道的叠加。

灰度图和 HSV 色域图是图像处理中最常用的两种处理方式，在后续的方法会

被经常调用，因此我将 imgHSV 和 imgGrey 作为成员存储起来。

## 边缘检测

边缘检测是图像处理和计算机视觉中的基本问题，边缘检测的目的是标识数字图像中亮度变化明显的点。在本项目中，使用边缘检测的目的是过滤出“感兴趣”区域，即有用的区域，也就是文字和物理图像。

我使用梯度形态学操作，在图像的亮区域和暗区域之间创建边缘。之后，将图像存入 basicGrad 矩阵中，之后，我使用**大津阈值算法**确定最佳阈值，对 basicGrad 进行自动二值化。这样我们就得到了图像的边缘，即亮度上梯度变化较大的区域，经比较，本方法比 opencv 中最常用的 canny 算法效果更好。

获得图像边缘之后，我们呢还面临一个问题，即我们获取的是图像的边缘，而不是图像的全部。



图 2 边缘检测示意图

比如，上图是我们要处理的图像，红色是我们检测的边缘，如果不对边缘进行处理，则会丢失边缘包围起来的区域内部的信息，因此我们需要对检测的边缘进行手动扩大。

使用 dilate 方法即可对边缘进行手动扩大。

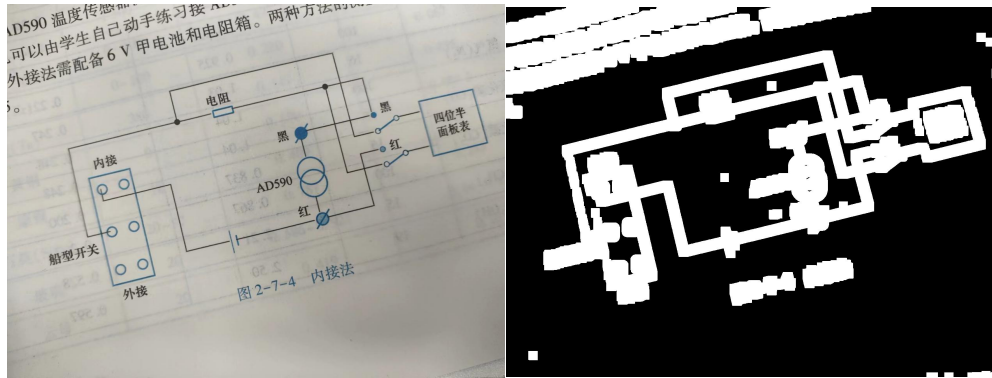


图 3 原图（左）扩大后的边缘（右）

## 注解

**大津阈值算法：**大津阈值算法（Otsu's method），又称为最大类间方差法，是一种用于图像二值化处理的算法，由日本学者大津展之于 1979 年提出。该算法的核心思想是选择一个阈值，使得图像中的前景和背景之间的类间方差最大化，从而实现最佳的分割效果。

## 去除背景

南开大学大学物理实验教材的背景色主要是 #bab8ac（RGB）附近的颜色，因此直接粘贴到 word 的 #ffffff（RGB 即白色）背景上会显得十分突兀，因此我们先将上一步检测到的边缘粘贴到一个三通道的纯白色的背景上（即 (255, 255, 255) ）。

但是，由于我们上一步的手动扩大边缘是径向平均向两侧扩大，所以不可避免地会包含背景颜色，因此我们需要过滤掉背景色。

这里我使用 HSV 创建**掩膜（Mask）**的方式，首先先利用 HSV 颜色过滤器将背景色的范围筛选出来，lowerbound=(0, 0, 146)，upperBound(255, 36, 255)。之后创建颜色掩膜（Mask），将掩膜部分强行赋值为 (255, 255, 255)，就实现了背景颜色的过滤。

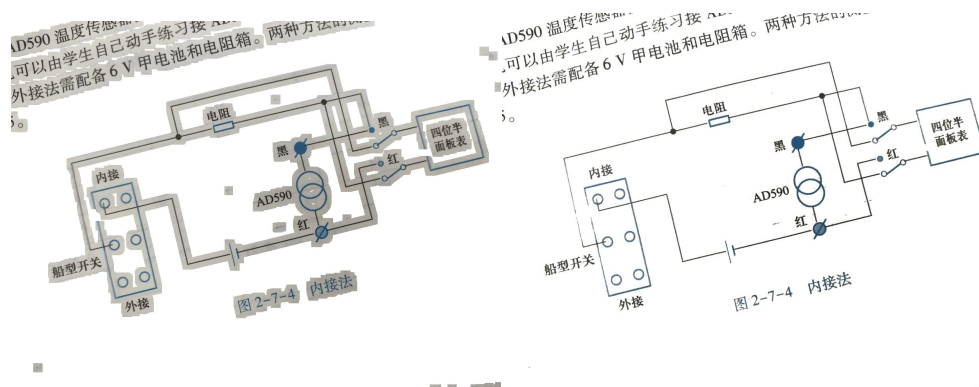


图 4 Mask 处理前（左）Mask 处理后（右）

## 注解

**掩膜（Mask）**：在图像处理和计算机视觉中，掩膜（Mask）是一种非常重要的工具，它用于控制图像处理操作的区域或对图像的特定区域进行操作。掩膜可以看作是一个二维数组，其大小与目标图像相同，但通常只有两种可能的像素值。在本项目中可以理解为我用一个掩膜把图像中所有颜色为背景色的像素点圈出。

## 自动旋转

我们希望能将一张物理图像进行自动旋转，这就需要我们编写一个方法返回旋转的最佳角度。

这里我们使用了关键区域方法，即找到整张图片中最关键的元素，之后让这个元素变为水平，关键元素旋转的角度就是整张图片的旋转角度。

我们使用 **findContours** 从二值化图像中检测轮廓，之后用矩形框出每一个区域。对区域的面积大小进行排序，最大的即为关键区域，我们令关键区域所在矩形的长边变为水平即可确定旋转角度。



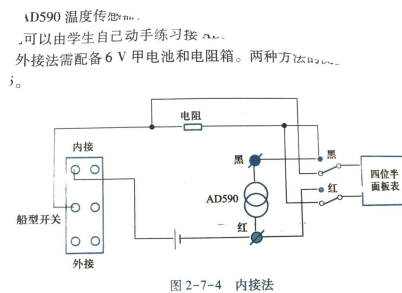
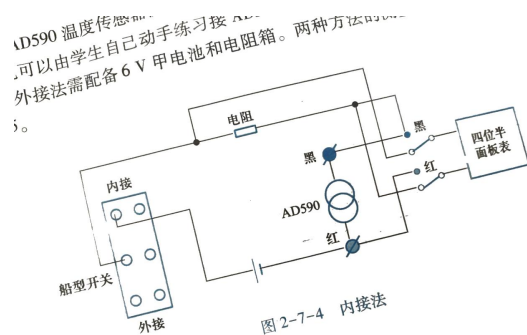


图 5 旋转前（左）旋转后（右）

## 注解

**findContours:** findContours 是 OpenCV 库中的一个函数，用于检测和提取图像中的所有轮廓。这些轮廓可以是任意形状的连通区域的边界，比如几何图形、文字、线条等。

## 文字检测与擦除

进行文字检测，需要使用 Tesseract 库，我们使用 Tesseract 训练好的 **chi\_sim** 库自动检测中文，并以矩形的形式返回，返回后发现，Tesseract 会将整个物理图像检测为一个文字，因此需要按照面积大小对返回的矩形进行排序，面积最大的矩形不做擦除处理。

在进行擦除后发现，Tesseract 库对文字边缘的处理效果不好，经常会忽视边缘，尤其是文字不完整的情况下，因此我们需要对矩形的面积进行手动扩大，扩大后，处理效果变得比较令人满意。

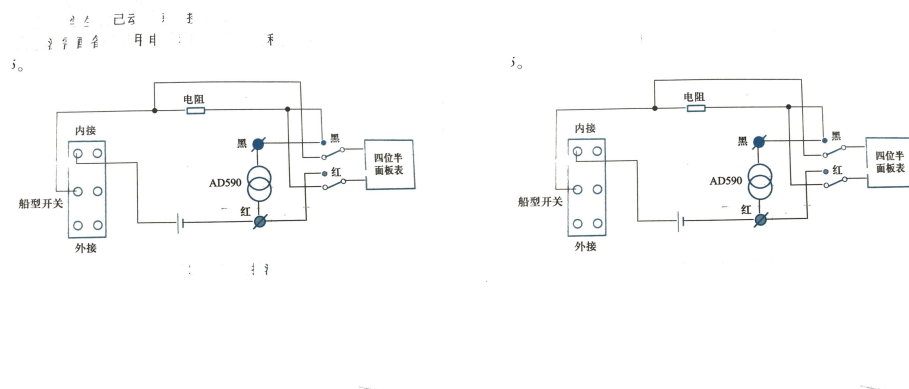


图 6 扩大擦出前（左）扩大擦除后（右）

## 注解

**chi\_sim**: 指 Tesseract 提供的训练好的简体中文数据集

## QT 图形化界面开发

与算法设计相比，图形化界面的开发比较简单，因此这里仅选取几个比较有趣的点展开。

### 自适应图标大小

本项目中所有的按钮按键都使用了导入素材图片的方法，因此需要解决在窗口拖拽（改变大小）时，图标的大小不跟着改变的问题，

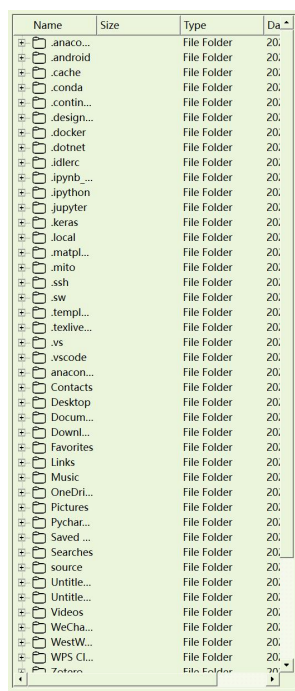
我将组件的属性全部改为 expanding，并重写的 QT 定义的 resize 监测，在监测到组件大小改变的时候，自动重新按照当前组件大小调整素材比例。



图 7 缩小时页面状态（左）放大后页面状态（右）

## 侧边文件栏

我使用了 QTreeWidget 来显示文件层级关系，并引入了双击的鼠标检测，实现了双击侧边栏中的图片自动上传的功能。



Name	Size	Type	Da
.anaco...		File Folder	20.
.android		File Folder	20.
.cache		File Folder	20.
.conda		File Folder	20.
.contin...		File Folder	20.
.design...		File Folder	20.
.docker		File Folder	20.
.dotnet		File Folder	20.
.idlerc		File Folder	20.
.ipynb...		File Folder	20.
.ipython		File Folder	20.
.jupyter		File Folder	20.
.keras		File Folder	20.
.local		File Folder	20.
.matpl...		File Folder	20.
.mito		File Folder	20.
.ssh		File Folder	20.
.sw		File Folder	20.
.templ...		File Folder	20.
.texlive...		File Folder	20.
.vs		File Folder	20.
.vscode		File Folder	20.
.anacon...		File Folder	20.
Contacts		File Folder	20.
Desktop		File Folder	20.
Docum...		File Folder	20.
Downl...		File Folder	20.
Favories		File Folder	20.
Links		File Folder	20.
Music		File Folder	20.
OneDri...		File Folder	20.
Pictures		File Folder	20.
Pychar...		File Folder	20.
Saved ...		File Folder	20.
Searches		File Folder	20.
source		File Folder	20.
Untitle...		File Folder	20.
Untitle...		File Folder	20.
Videos		File Folder	20.
WeCha...		File Folder	20.
WestW...		File Folder	20.
WPS Cl...		File Folder	20.
Python		File Folder	20.

图 8 TreeView

## 用户友好设计

我设计了鼠标悬浮在按钮上自动提示的功能，增强了用户使用体验感

我设计了详尽的错误提示分支，在代码中使用 try 结构和 if 判断，一方面引导用户按照正确方法使用程序，另一方面减少了程序直接崩溃的可能。

我设计了帮助文档，可以引导用户使用程序。

## UI 设计

我使用了即时设计来设计 UI 界面，先行预览组件位置和效果，按照给出的像素和坐标直接在 QT 上录入，极大提高了设计效率。我同时设计了欢迎界面，

增加了用户体验。



图 9 欢迎界面

## 五. 单元测试

我使用了从南开大学大学物理实验教科书上拍取的多张图片进行了效果检验, 都收获了不错的处理效果。



图 10 欢迎界面

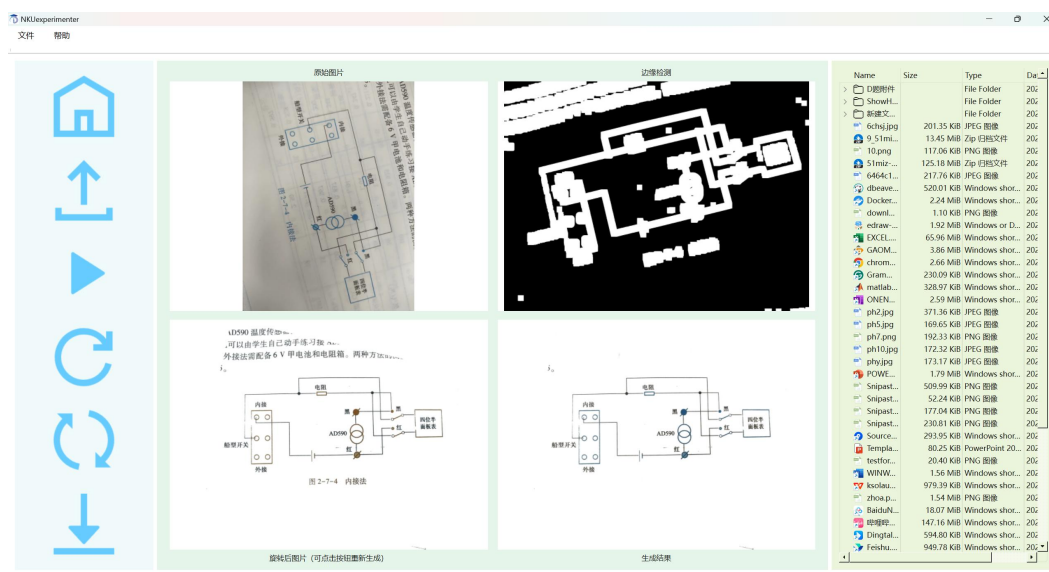


图 11 程序处理界面

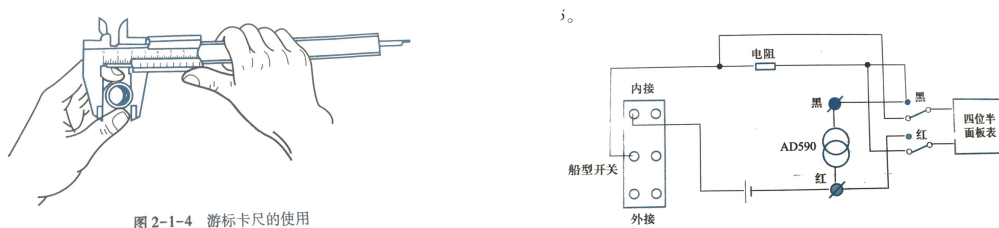


图 12 一些处理后的图片

## 六. 收获

### 1、规范代码习惯

本次开发过程中，变量命名全部使用小驼峰法（注：小驼峰命名法（Camel Case）是一种在编程中常用的命名规则，它的特点是除了第一个单词之外，后续单词的首字母大写。）变量名符合习惯，如 return 某个成员使用 get...，判断使用 is...

## 2、版本管理工具 git 的使用

本项目开发过程中，全程使用 git 管理代码，通过 checkout 和 new branch 的方式对代码进行跟踪管理，并在开发过程中通过 checkout 回溯进度的方式避免了一次项目崩溃。



图 13 Git 图形化管理工具 Sources Tree

## 3、openCV 库的使用

通过本项目，我学习了 openCV 的使用，并且熟悉了 opencv 中的一些方法和接口。

## 4、Tesseract 库的使用

通过本项目，我学习了 Tesseract 库的使用，并且熟悉了 Tesseract 库中的一些方法和接口。

## 5、环境配置

在本项目中，由于使用了第三方库，因此需要配置第三方环境，其中 opencv

需要配置头文件和库文件，因为在 github 上可以直接下载，较为方便，而 Tesseract 需要对文件进行编译处理，通过对官方英文使用文档的查询，我使用了 vcpkg 包管理工具实现了 Tesseract 的环境配置，在探索的过程中，我还学习了 cmake 的使用。

## 6、项目开发的细节

在项目的开发过程中，应避免使用多个 using namespace，有可能会致命名空间冲突，在开发过程中，我就遇到了 std 和 cv 中 sort 的冲突现象。

## 7、Qt 工具

在开发的过程中，我学习了 QT 图形化界面工具，包括：UI 界面的设计、按钮的交互功能、槽和信号函数、鼠标和键盘的检测、QLabel 下的图片和文字显示，QAction 的连接和热键、TreeView 的文件显示、Opencv 和 Qt 的联合使用、QLabel 的大小自适应，QMessage 的报错与提醒....