Qt + OpenCV实现彩色图和灰度图的转换

何长鸿 2016141482154

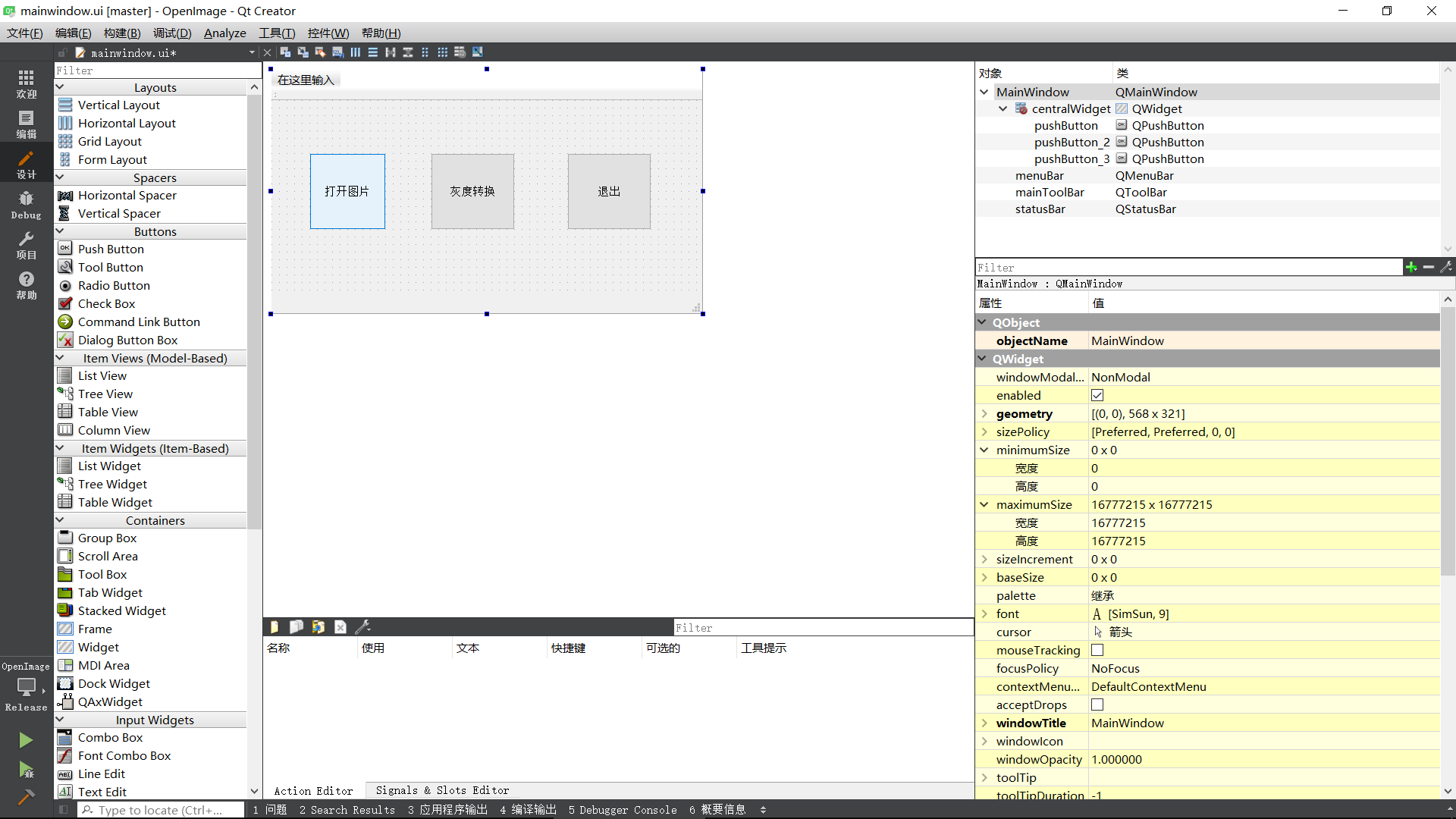
1. 实验目的与要求

**目的：**

* 1. 熟悉Qt可视化开发，理解C++的面向对象思想
  2. 熟悉Qt和Opencv开发环境搭建
  3. 了解Qt消息机制
  4. 初步理解Opencv的用法
  5. 学会使用c++异常处理

**要求：**

1. 使用Qt编写一程序，点击按钮从电脑目录选择jpg图片，显示在界面上。
2. 再设置一按钮或者菜单，点击后将图片黑白化或者灰度化，保存到另一个目录里，并显示出来。
3. 再次点击按钮，重新加载彩色图像，实现两种图像的转换。
4. 在读取文件和保存过程中，要加入异常处理（try...catch）来确保错误捕捉到自己的处理程序范围内。
5. 工具与准备工作
   1. 实验工具
      1. Qt\_5.8\_mingw\_WIN32
      2. Opencv\_3.4
      3. Win10\_x64
   2. 环境搭建
      1. Qt官网下载安装以上所述版本
      2. Opencv官网下载3.4版本源码。
      3. 由于Qt使用Mingw32位编译器，OpenCV需要自己编译。使用Qt打开OpenCV源码目录下的MakeLists.txt文件，构建项目配置为install，选择正确的编译输出目录开始编译
      4. 编译完成后新建项目，配置OpenCV环境依赖包。
6. 分析
   1. UI界面设计，使用Qt creator的可视化界面编辑工具。需要添加打开文件按钮、色彩转换按钮、退出按钮



* 1. 需要给按钮设置对应的消息槽函数，当点击按钮时调用对应的函数。
  2. 类声明如下：

MainWindow类

|  |
| --- |
| #ifndef MAINWINDOW\_H  #define MAINWINDOW\_H  #include <QMainWindow>  #include <QFileDialog>  #include <QDebug>  #include <opencv2/imgproc/imgproc.hpp>  #include <opencv2/highgui/highgui.hpp>  #include <opencv2/core/core.hpp>  using namespace cv;  namespace Ui {  class MainWindow;  }  class MainWindow : public QMainWindow  {  Q\_OBJECT  public:  explicit MainWindow(QWidget \*parent = 0);  ~*MainWindow*();  private slots:  void on\_pushButton\_clicked();  void on\_pushButton\_2\_clicked();  void on\_pushButton\_3\_clicked();  private:  Ui::MainWindow \*ui;  QImage img;  Mat src;  bool colorful = true;  QString img\_name;  };  #endif // MAINWINDOW\_H |

Main函数

|  |
| --- |
| #include "mainwindow.h"  #include <QApplication>  int main(int argc, char \*argv[])  {  QApplication a(argc, argv);  MainWindow w;  w.show();  return a.exec();  } |

1. 实现步骤
   1. 编辑ui界面，添加3个push\_button,并设置好布局。
   2. 分别给三个push\_button创建对象的相应函数。
   3. 打开文件使用文件选择窗口，选择文件后返回文件路径。

|  |
| --- |
| img\_name = QFileDialog::getOpenFileName( this, tr("Open Image"), ".",tr("Image Files(\*.png \*.jpg \*.jpeg \*.bmp)"));  QTextCodec \*code = QTextCodec::codecForName("gb18030");  string name = code->fromUnicode(img\_name).data();  if(name.length()<1) return;  qDebug() << "打开图片：" << img\_name;  try{  src=imread(img\_name.toUtf8().data());  namedWindow( "src", WINDOW\_NORMAL);  if(!src.empty()){  imshow("src",src);  }  }catch(Exception e){  cout << e.err << endl;  } |

* 1. 色彩转换：设置bool成员变量colorful，为true表示当前为彩色状态，反之为灰度图。使用OpenCV的cvtColor(img, img, COLOR\_RGB2GRAY)函数，参数COLOR\_RGB2GRAY表示将OpenCV的Mat对象由RBG图转换为gray灰度图。

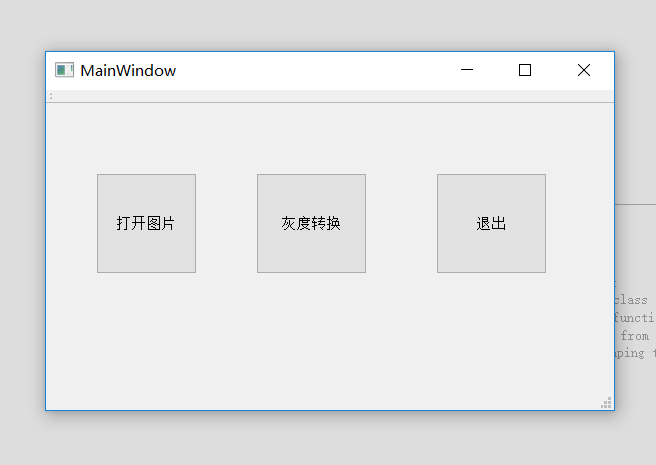
|  |
| --- |
| try{  if(colorful)  {  cvtColor(src, src, COLOR\_RGB2GRAY);  imshow("src",src);  }  else  {  src=imread(img\_name.toUtf8().data());//  imshow("src",src);  }  colorful = !colorful;  }catch(Exception e){  cout << e.err << endl; |

* 1. 退出按钮

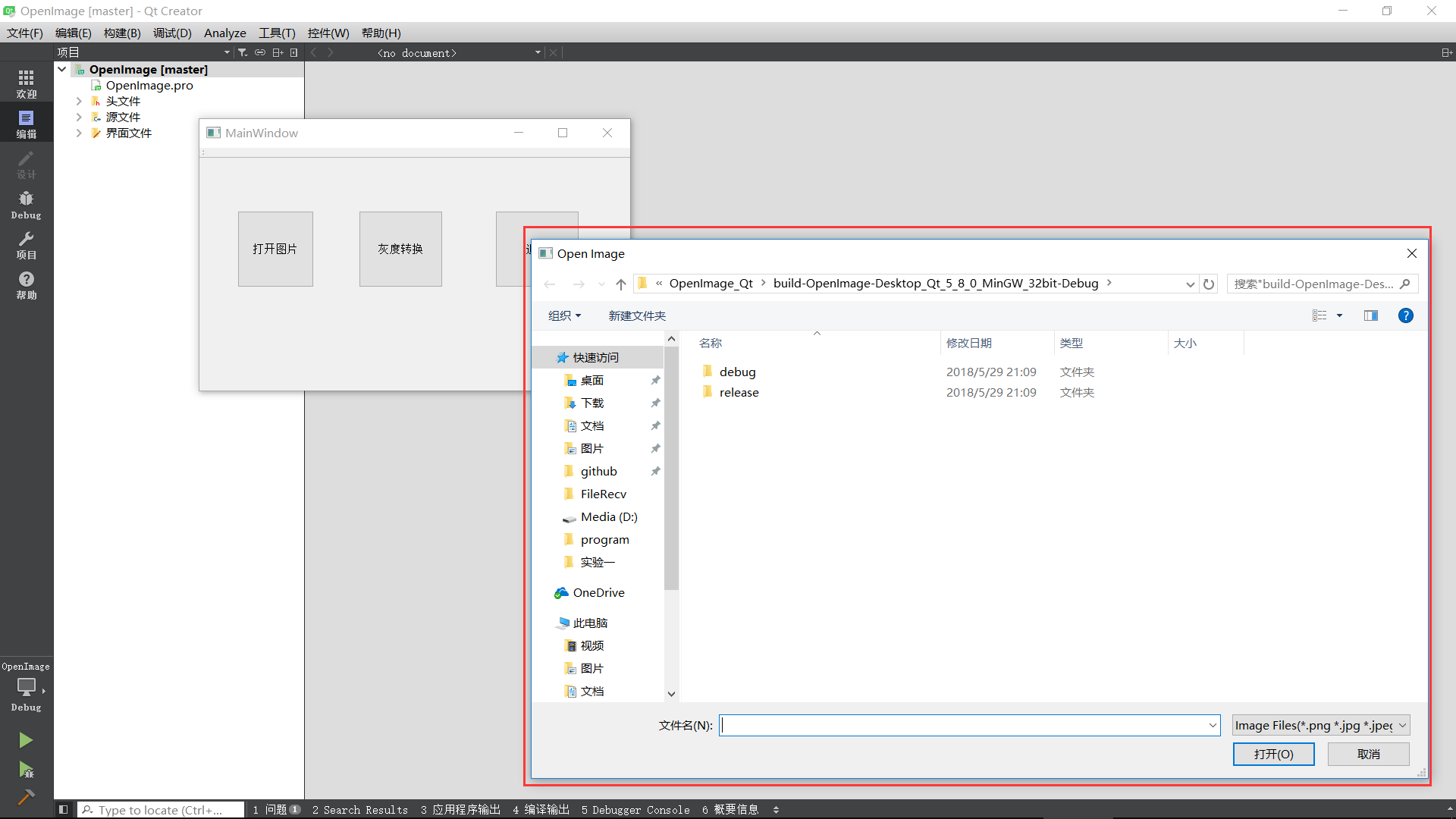
|  |
| --- |
| 直接调用exit(0); |

* 1. 打包发布，将程序 所需动态库与exe程序拷贝到同一文件夹。

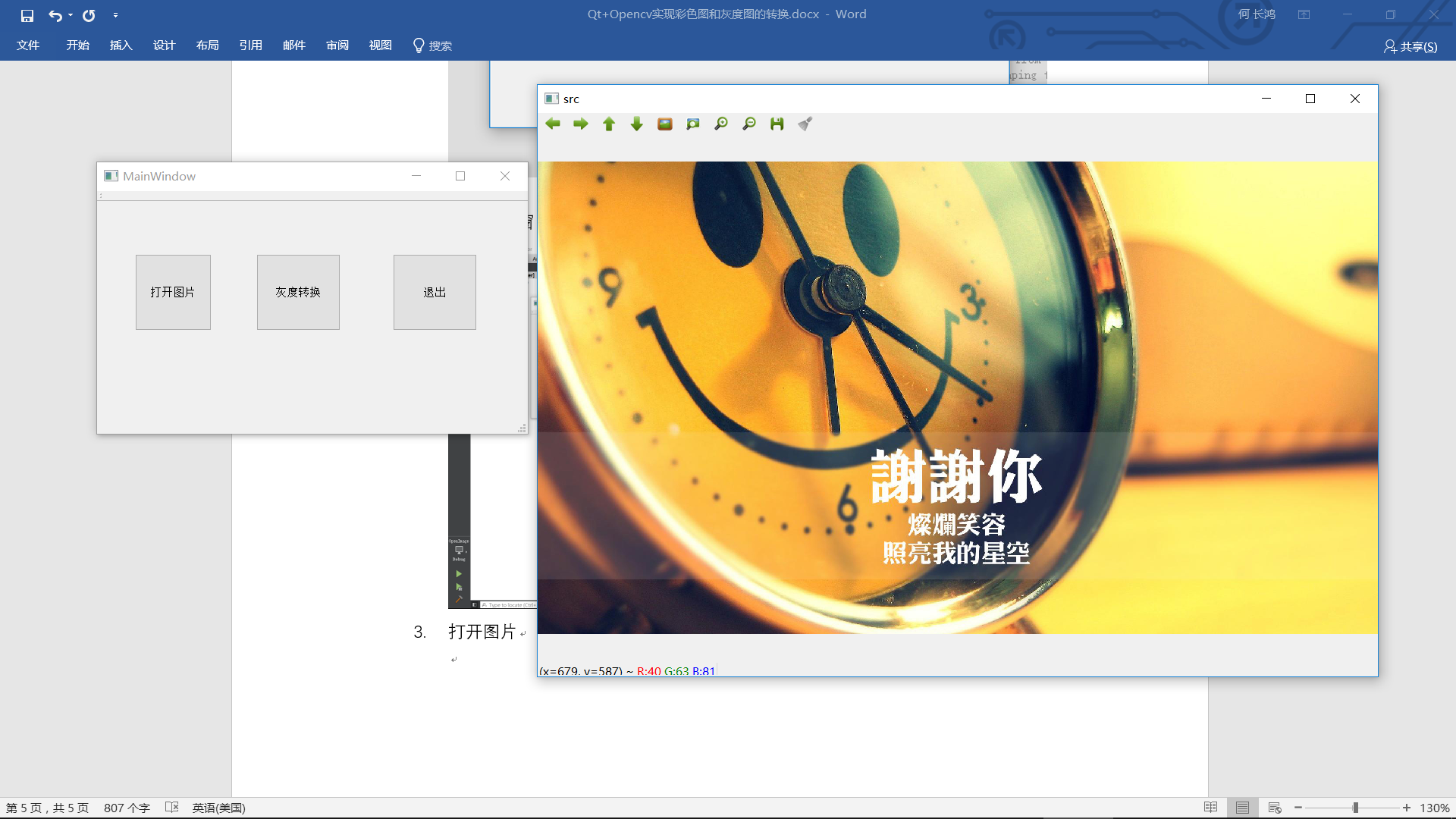
1. 测试与结论
   1. 运行程序主界面



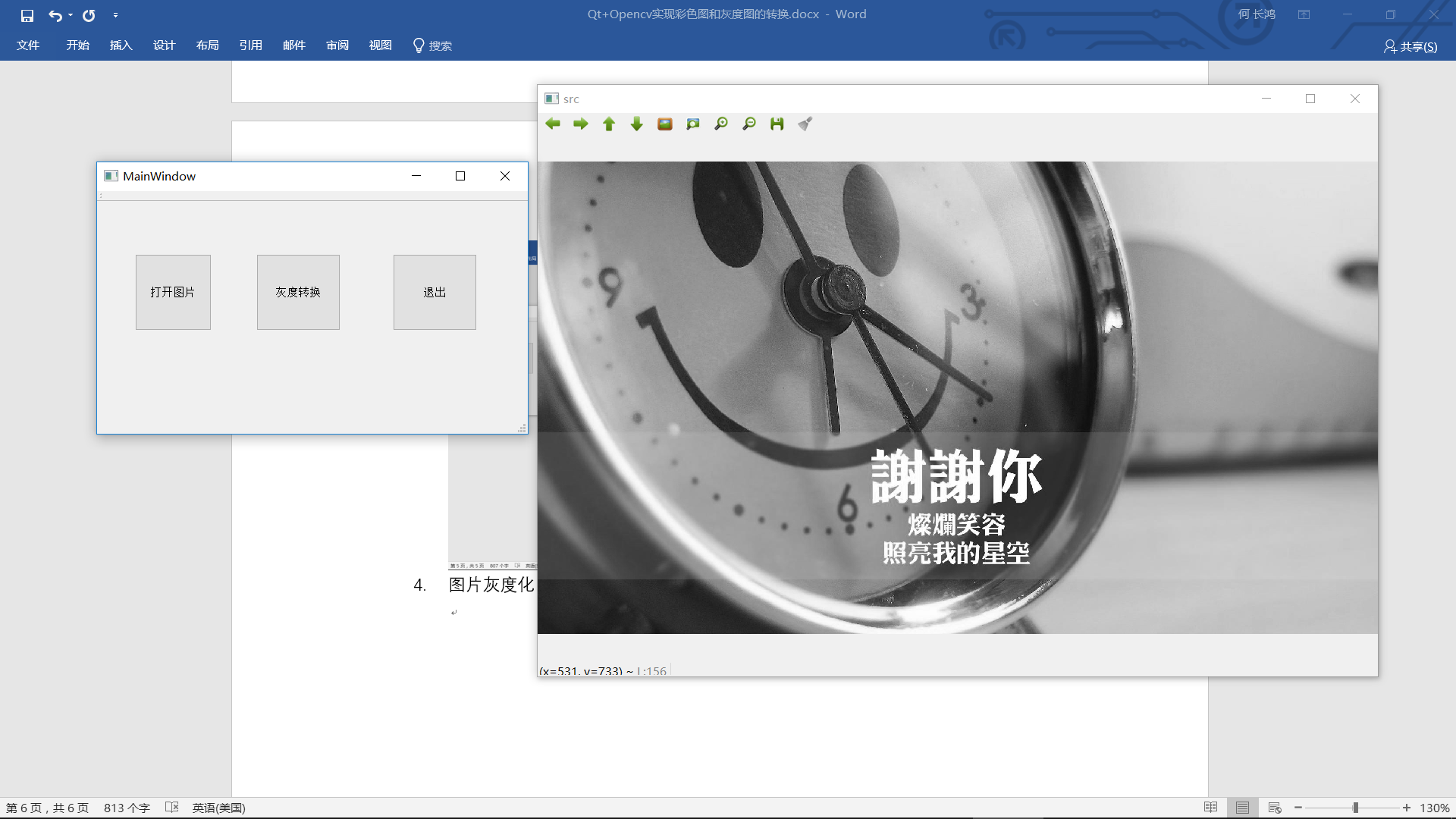
* 1. 选择图片窗口



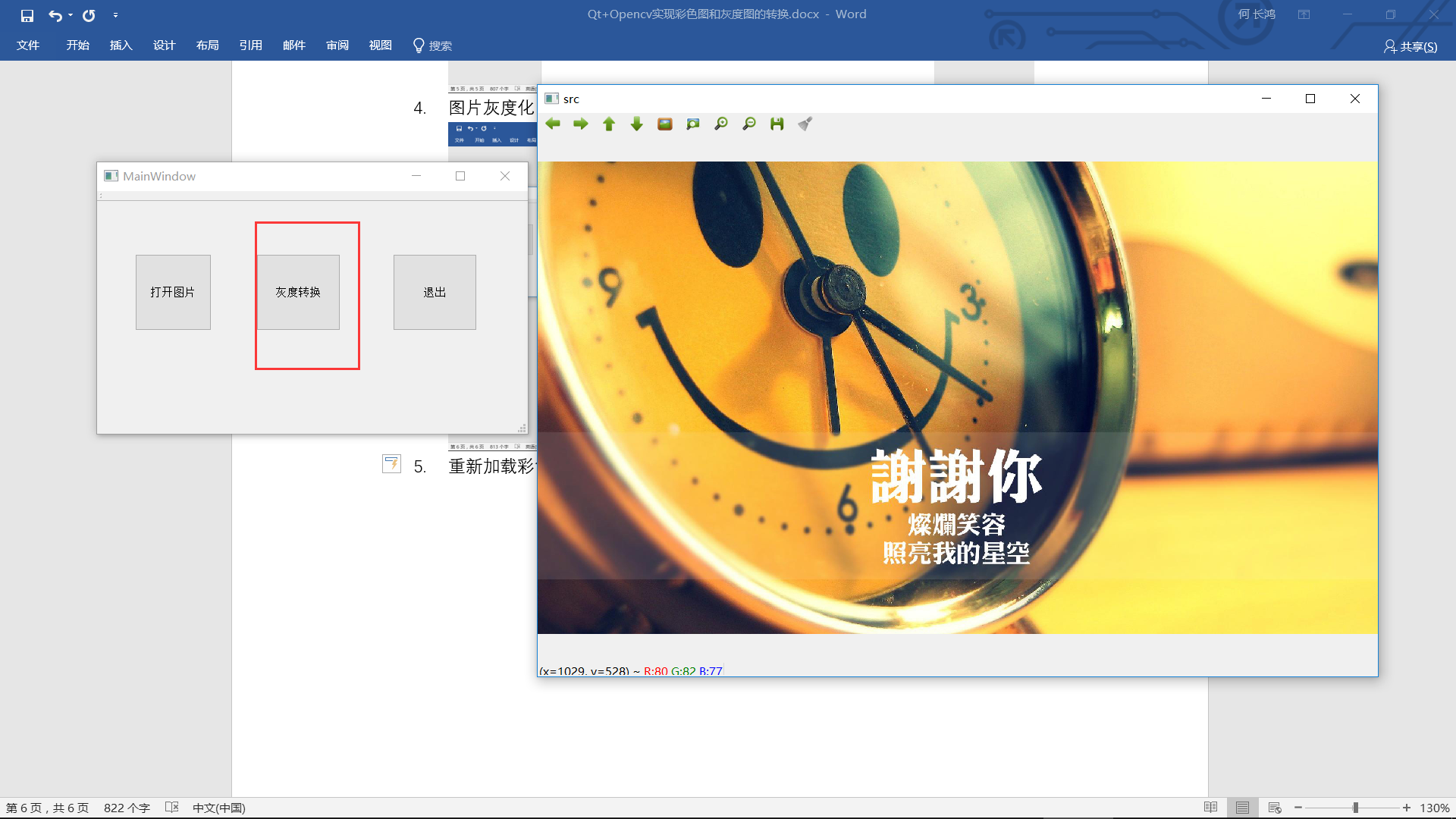
* 1. 打开图片



* 1. 图片灰度化



* 1. 重新加载彩色图片



1. 实验总结
   1. 创新点
      1. 将c++课堂学习的内容应用到QT桌面开发，其中多次用到类继承、多态等面向对象思想
      2. 使用开源图像处理库opencv，完成源代码编译动态库等工作，并学会使用基本的opencv函数显示图片和灰度化图片。
      3. 将图形界面与按钮信号关联，实现操作可视化界面
   2. 感悟

学会C语言、C++的基础语法，对于我们专业来说只是很小的一步，要完成我们所梦想的炫酷的程序，还有很多工具需要了解，比如这次试验的程序，虽然只是实现了很单一的功能，却需要我花大量的时间搭建qt和opencv开发环境，并且学习他们的使用方法。但是，在学习了基础的编程语言之后，的确给了我很多信心去了解并探究更多的工具和技能。