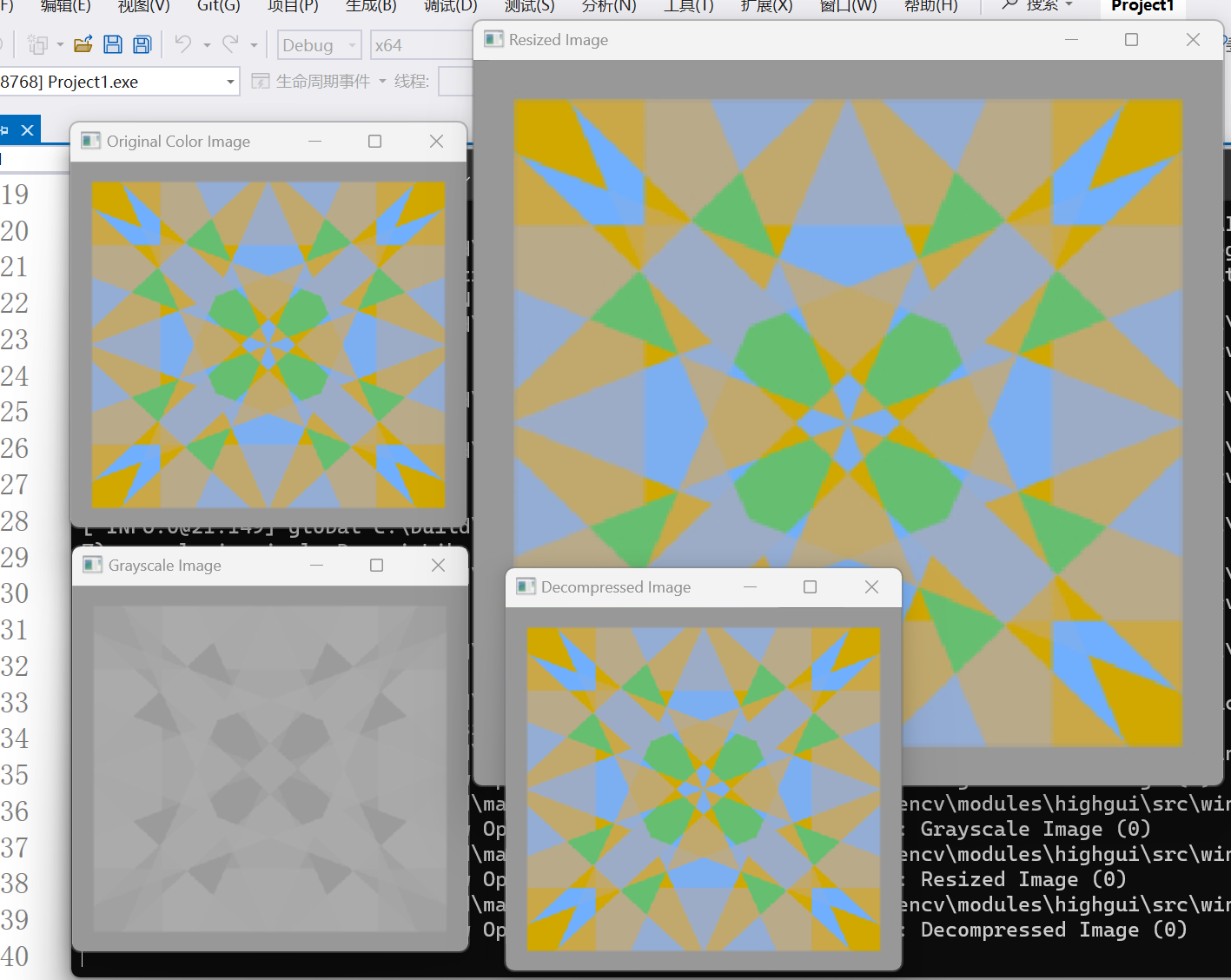
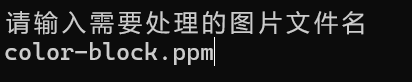
**Project 5 实验报告**

1. **程序简要功能说明**

本程序是一个简单图像处理程序，可以实现对图像数据的读取和写入存储，并可通过第三方库opencv调用UI窗口展示照片；实现图像的压缩存储和解压得到原图像；彩色图像去除色彩得到灰度图像；实现图像尺寸的缩放等功能。

1. **程序运行截图（计算功能演示，部分实际运行结果演示，命令行或交互式界面效果等）**

功能展示(以测试样例为例)：  




说明: 图中Original Color Image窗口显示的为原始图像，Grayscale Image窗口显示的为去彩色后的灰度图像，Resized Image窗口显示的为图片放大后的图像，Decompressed窗口显示的为图片先压缩后解压得到的图像。

1. **部分关键代码及其说明：**
2. 图像数据的读取和存储代码

// 图像数据的读取与写入存储

Mat readImage(const string& filename) {

Mat image = imread(filename);

if (image.empty()) {

cerr << "Failed to read image: " << filename << endl;

exit(1);

}

return image;

}

void saveImage(const Mat& image, const string& filename) {

imwrite(filename, image);

}

1. 图像数据去除色彩代码

// 彩色图像转换为灰度图像

Mat convertToGrayscale(const Mat& colorImage) {

Mat grayscaleImage;

cvtColor(colorImage, grayscaleImage, COLOR\_BGR2GRAY);

return grayscaleImage;

}

1. 图像尺寸的缩放代码

// 图像尺寸的缩放

Mat resizeImage(const Mat& image, int width, int height) {

Mat resizedImage;

resize(image, resizedImage, Size(width, height));

return resizedImage;

}

1. 图像的压缩和解压缩代码

// 图像数据的压缩存储

void compressImage(Mat &image, const string& compressedFileName) {

vector<int> compression\_params;

compression\_params.push\_back(IMWRITE\_JPEG\_QUALITY);

compression\_params.push\_back(50); // Adjust the quality as needed

imwrite(compressedFileName, image, compression\_params);

}

Mat decompressImage(string compressedFileName) {

return imread(compressedFileName, IMREAD\_UNCHANGED);

}

1. **程序运行方式简要说明：**

首先，程序所用的输入的样例图片需要和程序放在同一个目录下，然后，运行程序时需要输入图片的文件名（带后缀）。接着按下回车，程序会自动生成4个新的文件，并且调出4个窗口来展示不同处理后的图像。