

22336216-陶宇卓-Project5-实验报告

程序功能简要说明

程序运行截图，包括计算功能演示、部分实际运行结果展示、命令行或交互式界面效果等

部分关键代码及其说明

最关键的一步：导入OpenCV库

调用UI对图像进行展示

实现图像数据的压缩存储

彩色图像转变为灰度图像

实现图像尺寸的缩放

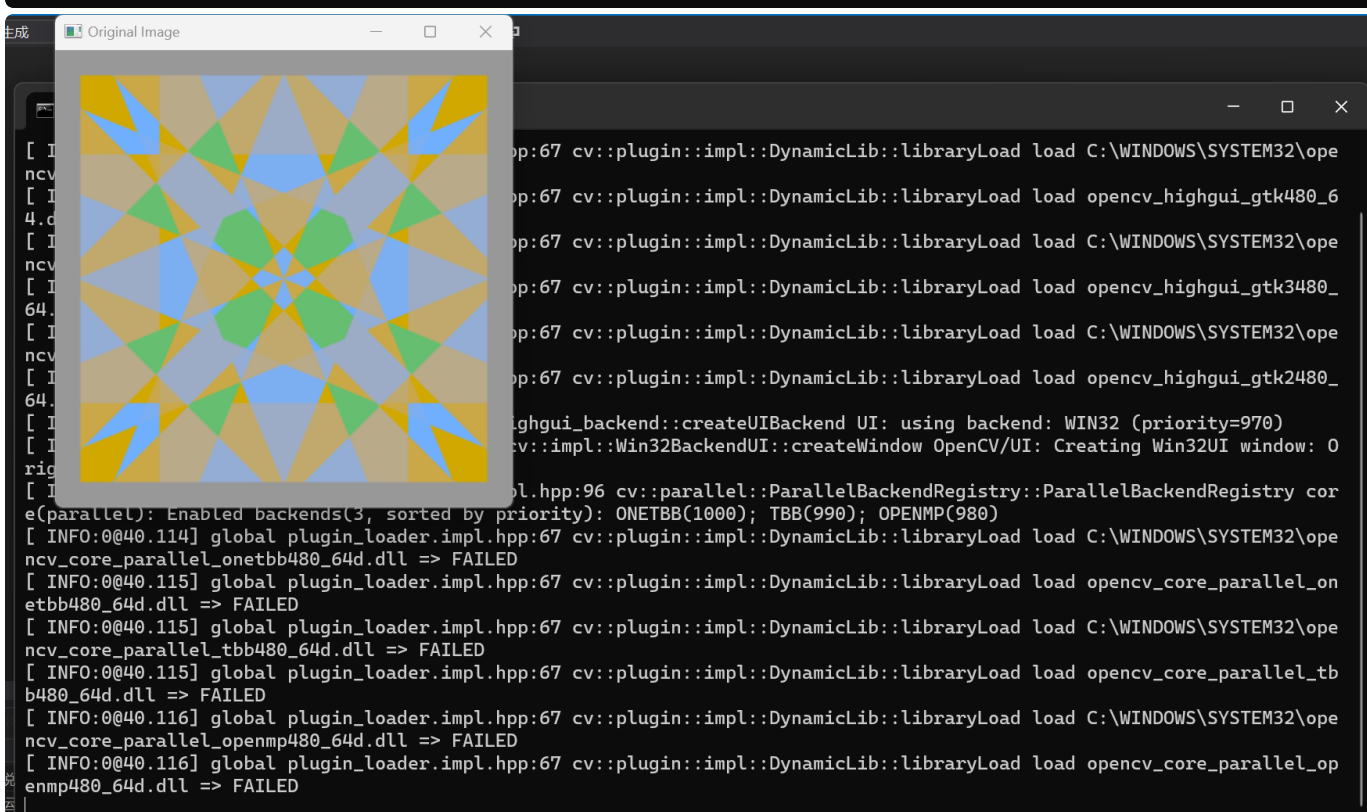
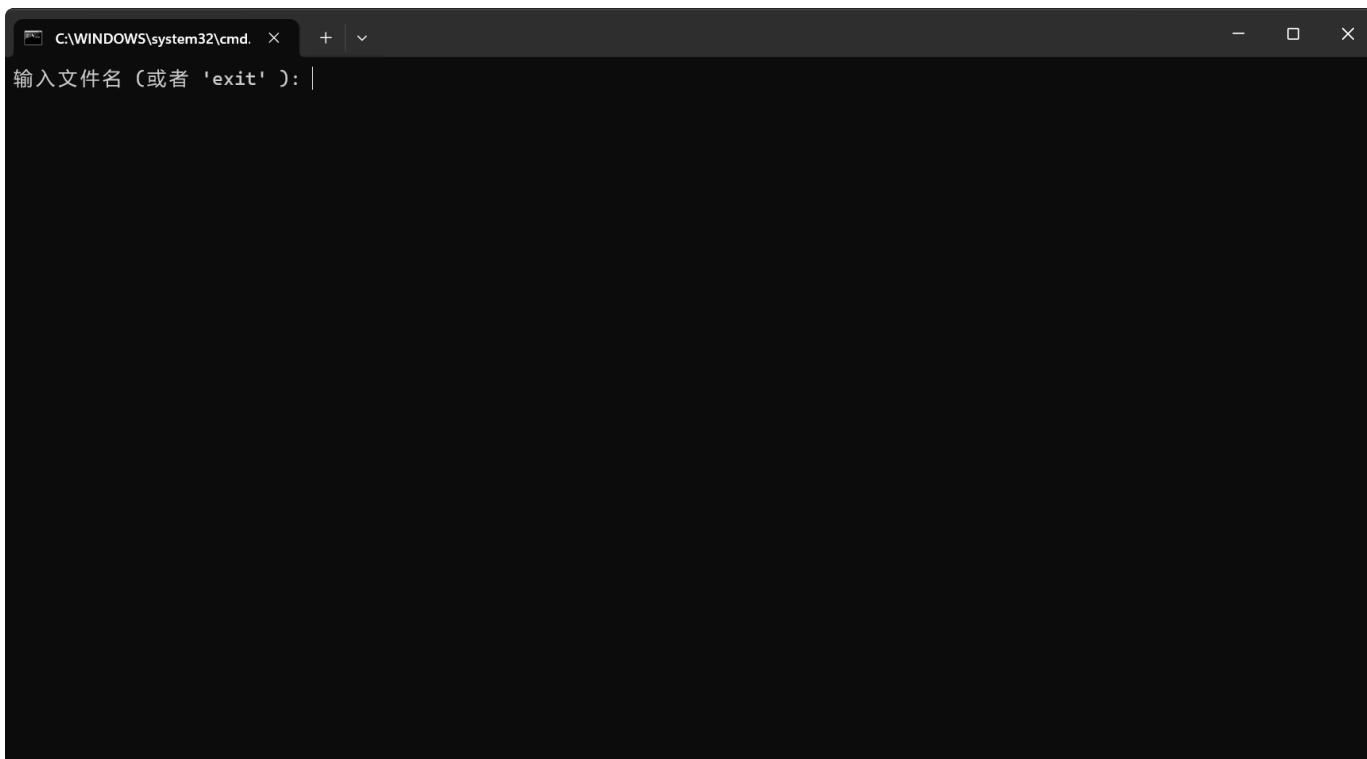
实现图像数据的解压缩

程序运行方式简要说明

程序功能简要说明

- (1) 图像数据的读取与写入存储，并可通过第三方库调用UI窗口展示图片。
- (2) 实现图像数据的压缩存储。首先读取图像，并将图像信息以三元组结构存储。其次，对三元组数据进行压缩存储，以实现数据压缩的功能。最后，将压缩后的数据进行读取，解码，得到原始的图像数据，并进行保存。
- (3) 彩色图像转变为灰度图像。将彩色的“color-block.ppm”图像转换为灰度图像，并进行保存和展示。
- (4) 实现图像尺寸的缩放。例如将“lena-128-gray.ppm”图像放大为256x256大小；或将“lena-512-gray.ppm”图像缩小为256x256大小。


程序运行截图，包括计算功能演示、部分实际运行结果展示、命令行或交互式界面效果等

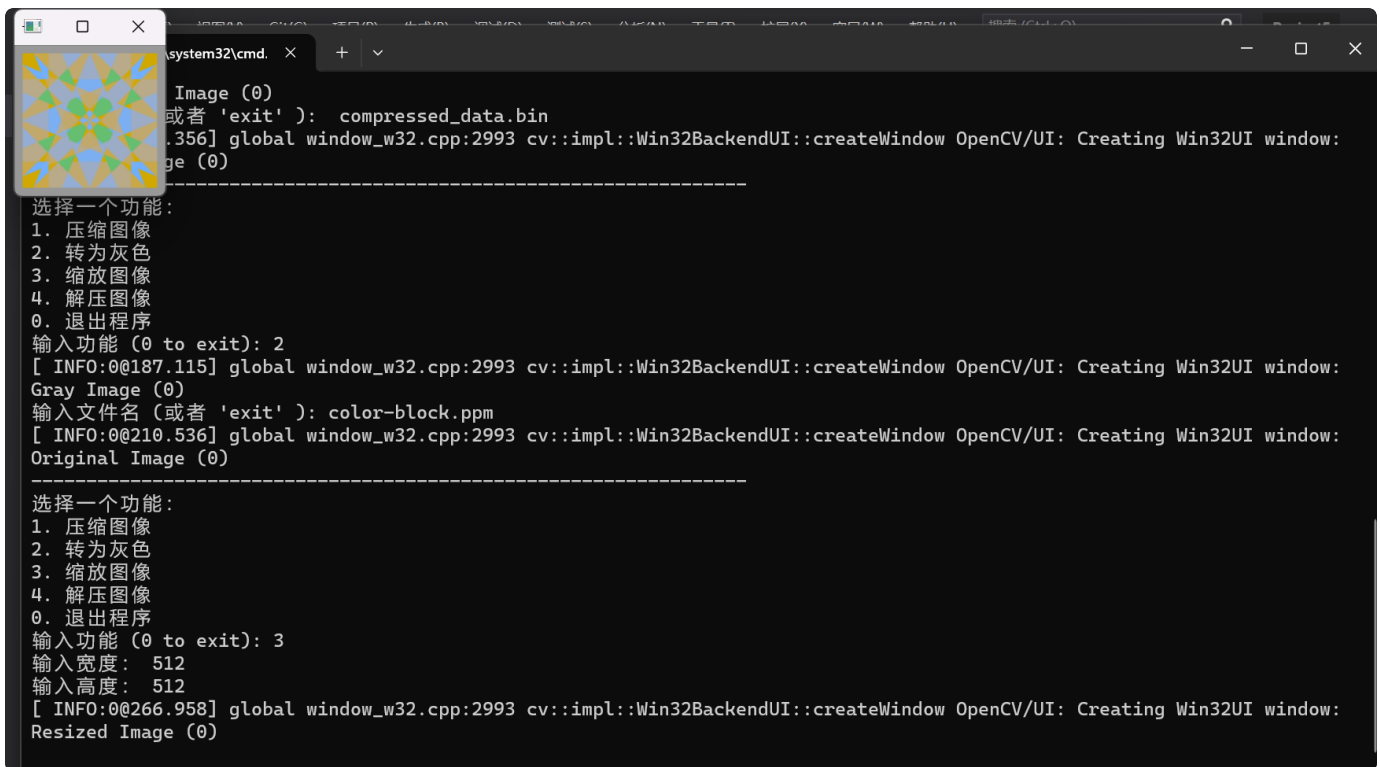


```
C:\WINDOWS\system32\cmd. x + v
64.dll => FAILED
[ INFO:0@40.046] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load C:\WINDOWS\SYSTEM32\ope
ncv_highgui_gtk2480_64.dll => FAILED
[ INFO:0@40.047] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load opencv_highgui_gtk2480_
64.dll => FAILED
[ INFO:0@40.049] global backend.cpp:90 cv::highgui_backend::createUIBackend UI: using backend: WIN32 (priority=970)
[ INFO:0@40.049] global window_w32.cpp:2993 cv::impl::Win32BackendUI::createWindow OpenCV/UI: Creating Win32UI window: 0
riginal Image (0)
[ INFO:0@40.113] global registry_parallel.impl.hpp:96 cv::parallel::ParallelBackendRegistry::ParallelBackendRegistry cor
e(parallel): Enabled backends(3, sorted by priority): ONETBB(1000); TBB(990); OPENMP(980)
[ INFO:0@40.114] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load C:\WINDOWS\SYSTEM32\ope
ncv_core_parallel_onetbb480_64d.dll => FAILED
[ INFO:0@40.115] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load opencv_core_parallel_on
etbb480_64d.dll => FAILED
[ INFO:0@40.115] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load C:\WINDOWS\SYSTEM32\ope
ncv_core_parallel_tbb480_64d.dll => FAILED
[ INFO:0@40.115] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load opencv_core_parallel_tb
b480_64d.dll => FAILED
[ INFO:0@40.116] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load C:\WINDOWS\SYSTEM32\ope
ncv_core_parallel_openmp480_64d.dll => FAILED
[ INFO:0@40.116] global plugin_loader.impl.hpp:67 cv::plugin::impl::DynamicLib::libraryLoad load opencv_core_parallel_op
enmp480_64d.dll => FAILED
-----
选择一个功能:
1. 压缩图像
2. 转为灰色
3. 缩放图像
4. 解压图像
0. 退出程序
输入功能 (0 to exit): |
```

输入功能 (0 to exit): 1
压缩后的图像已经被保存到 compressed_data.bin.
输入文件名 (或者 'exit'):

```
C:\WINDOWS\system32\cmd. x + v
0. 退出程序
输入功能
压缩后的图
输入文件
[ INFO:0@
Original
-----
选择一个功
1. 压缩图
2. 转为灰
3. 缩放图
4. 解压图
0. 退出程
输入功能
[ INFO:0@
Decompres
输入文件
[ INFO:0@
Original
-----
选择一个功
1. 压缩图
2. 转为灰
3. 缩放图
4. 解压图
0. 退出程序
输入功能 (0 to exit): 2
[ INFO:0@187.115] global window_w32.cpp:2993 cv::impl::Win32BackendUI::createWindow OpenCV/UI: Creating Win32UI window:
Gray Image (0)
```





部分关键代码及其说明

最关键的一步：导入OpenCV库

在vs2019上导入OpenCV就花了我半天时间，然后我也只能实现在x64debug环境下生成程序。

```
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <opencv2/opencv.hpp>
4  #include <iostream>
5
6  using namespace cv;
7  using namespace std;
```

调用UI对图像进行展示

```
displayImage C++ |
1 void displayImage(const Mat& image, const string& windowName = "Image") {
2     namedWindow(windowName, WINDOW_AUTOSIZE);
3     imshow(windowName, image);
4     waitKey(0);
5     destroyAllWindows();
6 }
```

调用OpenCV库中函数imshow对图像进行展示，namedWindow函数的第二个参数被设置为WINDOW_AUTOSIZE，这样窗口就不会自动调整大小，之后按下任何按键窗口都会关闭。

实现图像数据的压缩存储

调用OpenCV库中的imencode函数对图像压缩并返回一个无符号字符（uchar）数组。

```
compressImage C++ |
1 vector<uchar> compressImage(const Mat& image) {
2     vector<uchar> compressed_data;
3     imencode(".ppm", image, compressed_data);
4     return compressed_data;
5 }
```

老师好，请问project5里面对于图像的压缩处理可以用cv库里的imencode函数吗🤔



可以的👉

好滴 谢谢老师🤔



彩色图像转变为灰度图像

调用OpenCV库中的cvtColor函数对图像进行灰度转换，其实这里也可以把BGR三个通道的色彩值分离之后进行加权处理，但是后来我为省事采用了调库(doge)

▼ convertToGray

C++

```
1 ▼ Mat convertToGray(const Mat& colorImage) {  
2     Mat grayImage;  
3     cvtColor(colorImage, grayImage, COLOR_BGR2GRAY);  
4     return grayImage;  
5 }
```

实现图像尺寸的缩放

调用OpenCV库里的resize函数对图像进行缩放处理。

▼ resizeImage

C++

```
1 ▼ Mat resizeImage(const Mat& originalImage, int width, int height) {  
2     Mat resizedImage;  
3     resize(originalImage, resizedImage, Size(width, height));  
4     return resizedImage;  
5 }
```

实现图像数据的解压缩

调用OpenCV库里的imdecode函数对图像进行解压缩。

▼ decompressImage

C++

```
1 ▼ Mat decompressImage(const vector<uchar>& compressedData) {  
2     Mat decompressedImage = imdecode(compressedData, IMREAD_COLOR);  
3     return decompressedImage;  
4 }  
5
```

程序运行方式简要说明

首先，程序进入循环，提示用户输入文件名，检测是否合法并调用UI展示图像。之后提示用户选择功能。

```

1  while (true) {
2      // 用户输入图像文件名
3      cout << "输入文件名 (或者 'exit') : ";
4      string fileName;
5      cin >> fileName;
6
7      if (fileName == "exit") {
8          cout << "退出程序." << endl;
9          break;
10     }
11
12     // 读取图像
13     Mat originalImage = imread(fileName);
14
15     // 检查图像是否成功读取
16     if (originalImage.empty()) {
17         cout << "错误: 读取文件失败." << endl;
18         continue; // 回到循环开始, 重新输入图像文件名
19     }
20
21     // 显示原始图像
22     displayImage(originalImage, fileName);
23     // 提示用户选择功能
24     cout << "-----" << endl;
25     cout << "选择一个功能: " << endl;
26     cout << "1. 压缩图像" << endl;
27     cout << "2. 转为灰色" << endl;
28     cout << "3. 缩放图像" << endl;
29     cout << "4. 解压图像" << endl;
30     cout << "0. 退出程序" << endl;
31
32     // 用户选择功能
33     int choice;
34     cout << "输入功能 (0 to exit): ";
35     cin >> choice;
36

```

功能1: 图像压缩

创建一个类型为uchar的vector, 创建并打开一个叫compressed_data.bin的二进制文件, 对compressImage函数返回的值进行写入。


```
▼ C++ |
1 case 1: {
2     // (2) 实现图像数据的压缩存储
3     vector<uchar> compressedData = compressImage(originalImage);
4
5     FILE* file;
6     if (fopen_s(&file, "compressed_data.bin", "wb") != 0) {
7         cout << "错误: 读取文件失败." << endl;
8         return -1;
9     }
10
11     fwrite(compressedData.data(), sizeof(uchar), compressedData.size(), file);
12     fclose(file);
13
14     cout << "压缩后的图像已经被保存到 compressed_data.bin." << endl;
15     break;
16 }
17
```

功能2: 彩色图像转变为灰度图像

```
▼ C++ |
1 case 2: {
2     // 彩色图像转变为灰度图像
3     Mat grayImage = convertToGray(originalImage);
4
5     // 将灰度图像保存和展示
6     imwrite("gray_image.png", grayImage);
7     displayImage(grayImage, "Gray Image");
8     break;
9 }
10
```

功能3: 实现图像尺寸的缩放

```
1  case 3: {  
2      // 实现图像尺寸的缩放  
3      int newWidth, newHeight;  
4      cout << "输入宽度: ";  
5      cin >> newWidth;  
6      cout << "输入高度: ";  
7      cin >> newHeight;  
8  
9      Mat resizedImage = resizeImage(originalImage, newWidth, newHeight);  
10  
11     // 将图像保存和展示  
12     imwrite("resized_image.png", resizedImage);  
13     displayImage(resizedImage, "Resized Image");  
14     break;  
15 }
```

功能4: 解压缩

```
1  case 4: {
2      // (5) 实现图像数据的解压缩
3      FILE* file;
4      if (fopen_s(&file, "compressed_data.bin", "rb") != 0) {
5          // 处理文件打开失败的情况
6          cout << "错误: 打开压缩文件失败." << endl;
7          return -1;
8      }
9
10     fseek(file, 0, SEEK_END);
11     long fileSize = ftell(file);
12     rewind(file);
13
14     vector<uchar> loadedCompressedData(fileSize);
15     fread(loadedCompressedData.data(), sizeof(uchar), fileSize, fi
le);
16
17     fclose(file);
18
19     Mat decompressedImage = decompressImage(loadedCompressedData);
20
21     // 将解压缩后的图像保存和展示
22     imwrite("decompressed_image.png", decompressedImage);
23     displayImage(decompressedImage, "Decompressed Image");
24     break;
25 }
```