实验二：图像文件读写

【**实验目的**】

1.       掌握通用图像格式（JPEG、PNG等）文件的读写方法；

2.       掌握自定义图像格式文件的编解码方法；

3.       掌握使用文件对话框打开指定图像文件的方法；

4.       培养图像处理程序的调试与优化能力。

【**知识点**】

图像文件存储结构，图像编解码技术

【**背景知识**】

在数字图像处理领域，图像文件的读写与显示是基础且关键的环节。图像文件不仅承载着视觉信息，还包含了图像的格式、编码方式以及相关的元数据。因此，对图像文件的正确读取、解析和显示，是后续图像处理的前提和基础。

图像文件格式是图像数据的存储规范，常见的格式包括JPEG、PNG、BMP等。每种格式都有其独特的编码方式和应用场景。图像文件的读写涉及对文件格式的解析和数据的提取，读取图像是指从文件中提取图像数据，并将其转换为内存中的像素矩阵，保存图像是指将处理后的图像数据按照指定格式编码并写入文件。图像显示是将图像数据通过屏幕或其他显示设备呈现出来，显示过程需要将像素矩阵转换为可视化的图像。不同的显示需求可能需要不同的处理方式，对于常见的图像格式（如JPEG、PNG等），可以直接使用图像处理库进行加载和显示；如果图像采用了特殊的编码或格式，需要先解析其编码方式，再将其转换为通用的像素矩阵进行显示。

【**实验内容**】

1.       在实验1的基础上，添加“打开文件”的按键控件，打开并显示指定路径下的通用格式图像，例如JPEG等格式的文件；

2.       显示打开文件中的图片，并使用Label控件显示图像的基本参数，包括图像矩阵大小与颜色通道信息等；

3.       自定义函数read\_raw(file\_name)与write\_raw(file\_name, array)。read\_raw(file\_name)函数实现功能：使用Python自带的struct库读取二进制图像（RAW）文件，由参数file\_name指定，将其保存为二维数组并返回；write\_raw(file\_name, array)函数实现功能：将二维数组array保存为RAW格式图像文件；

4.       在主函数中使用内容3中的read\_raw(file\_name)函数读取指定目录下的所有RAW格式图像文件；

5.       将读入的图像重新保存为重新命名的RAW格式文件，与原始RAW格式图像比较，确认两者一致；

【**温馨提示**】

1.       图像读取时需注意图像文件路径是否正确；

2.       读取多张图像时，可先将文件路径保存到列表中，再循环遍历执行相应操作；

3.       使用struct读写二进制文件见相关课件。

【**实验要求**】

1.       编写实现上述实验内容的Python程序，显示程序运行结果图。

2.       结合图像读写与编解码的基本原理，对程序运行结果进行分析和讨论。

3.       在实验过程中，若发现问题，要进行探究，并给出解决问题的方案。

【**拓展实验**】

在实验内容5的基础上，显示指定目录下的多张RAW格式图像。自定义事件img\_forward(event)与img\_backward(event)，来显示当前图像的上一张与下一张图像，并在主函数中将鼠标左键单击事件绑定img\_forward响应，鼠标右键单击事件绑定img\_backward响应。