大作业说明文档

注：本文档仅作为技术说明，关于软件的操作使用以及功能介绍请参照**“软件说明文档”**

框架、库使用说明

- 软件使用了**MFC**框架
- 软件使用了由老师提供的CImageProcessor.h库来读取图像
- 软件使用了Opencv-4.5.5来进行图像的预览和放缩，**在编译时请确保拥有并成功配置Opencv环境，否则将会无法成功编译**

主要功能技术介绍

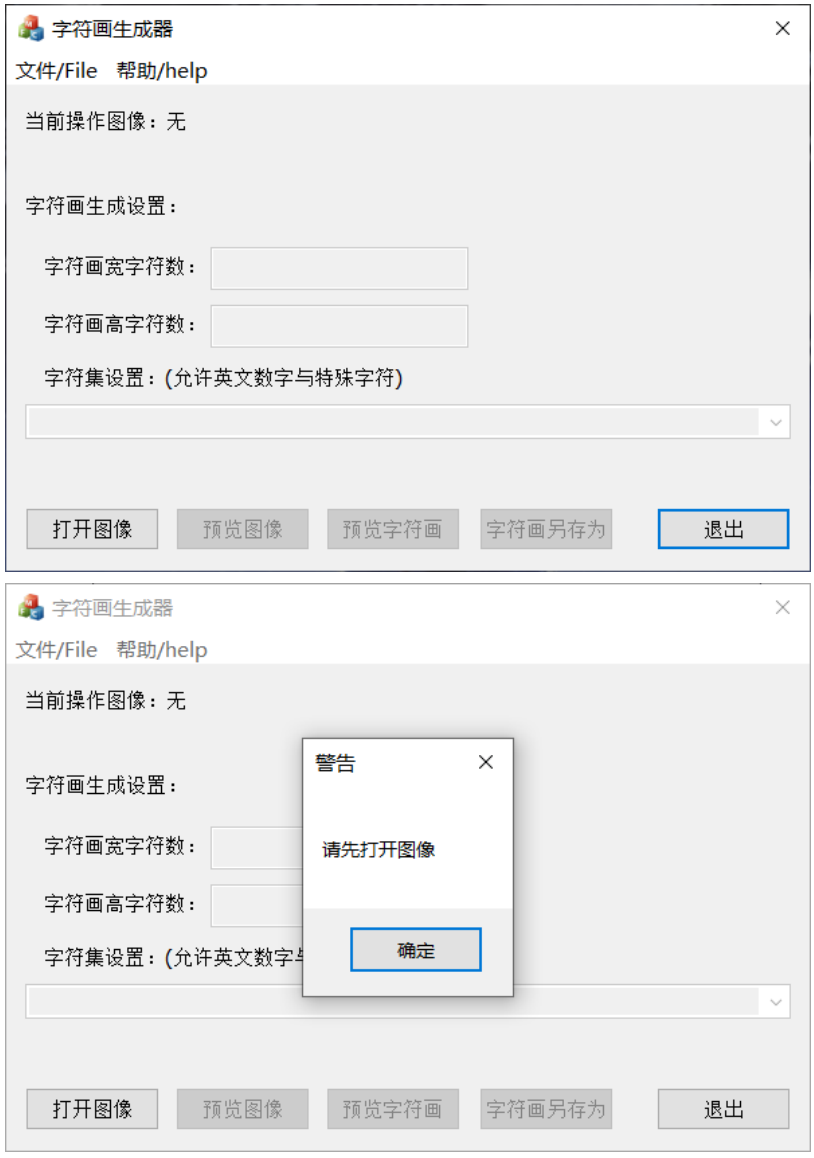
本软件的主要功能为图像转字符画，其基本原理是**根据图像像素点的RGB转换为灰度值，再根据灰度值映射到字符集**，选取某一字符来替代该像素点。代码及详细注释如下：

```
//字符画生成函数
//参数Ccode即为使用的字符集
void CharactorPaintGenerate(CString Ccode) {
    //原图像已经通过Opencv放缩处理后临时储存在temp.bmp中
    CString filename = _T(".\\temp.bmp");
    RGBQUAD* curColorData = NULL;    //图像数据32位
    int bmpW;                   //图像宽
    int bmpH;                   //图像高

    //使用CImageProcessor库获取图像数据，其中cuiColorData中储存了
    //curColorData储存各个像素点的RGB值，同时bmpW和bmpH会赋值为图像的宽和高
    curColorData = CImageProcessor::GetDataFromBMP(filename, bmpW, bmpH);

    int length;//字符集长度
    int gray;//灰度值
    float unit;//灰度映射标准

    CString temp;
    length = Ccode.GetLength();
    unit = (256.0 + 1) / length;
    charToSend = _T("");//传递给预览窗口的CString
    charToSave = _T("");//保存到txt的CString
    wordW = bmpW;//bmpW为放缩后图像的width,对应着字符画的宽字符数
    wordH = bmpH;//bmpH为放缩后图像的height,对应着字符画的高字符数

    //将像素转化为字符并储存
    for (int i = bmpH-1; i >= 0; i--) {
        for (int j = 0; j < bmpW; j++) {

            gray = int(0.2126 * curColorData[i * bmpW+j].rgbRed +
                0.7152 * curColorData[i * bmpW + j].rgbGreen +
                0.0722 * curColorData[i * bmpW + j].rgbBlue);  //计算灰度
            charToSend += Ccode.GetAt((int)(gray/unit));  //不同的灰度对应着不同的字符
            charToSave += Ccode.GetAt((int)(gray / unit));  //不同的灰度对应着不同的字符
```

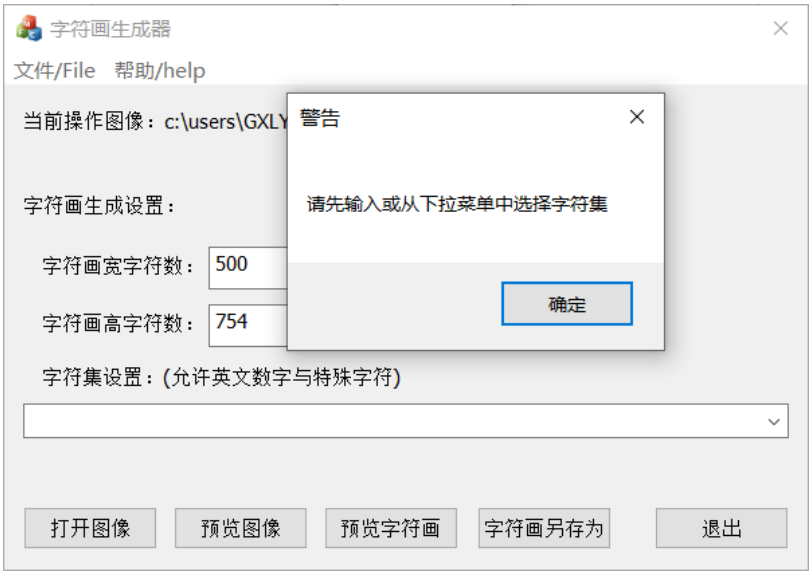
```
35          }
36          charToSend += _T("\r\n");//在预览窗口显示时的换行为CRLF
37          charToSave += _T("\n");//储存在txt中的换行为LF
38      }
39      //temp.bmp将会在生成temp.bmp的函数中删除
40  }
```

防止用户误操作

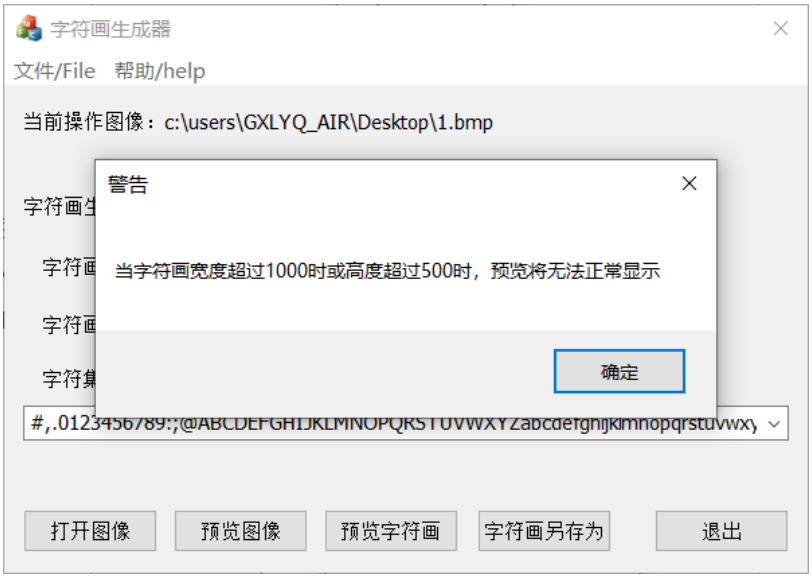
- 在用户未打开图像的情况下，字符画设置和某些功能按钮均被禁用，使用菜单栏内的“另存为”将会弹出提示并终止操作

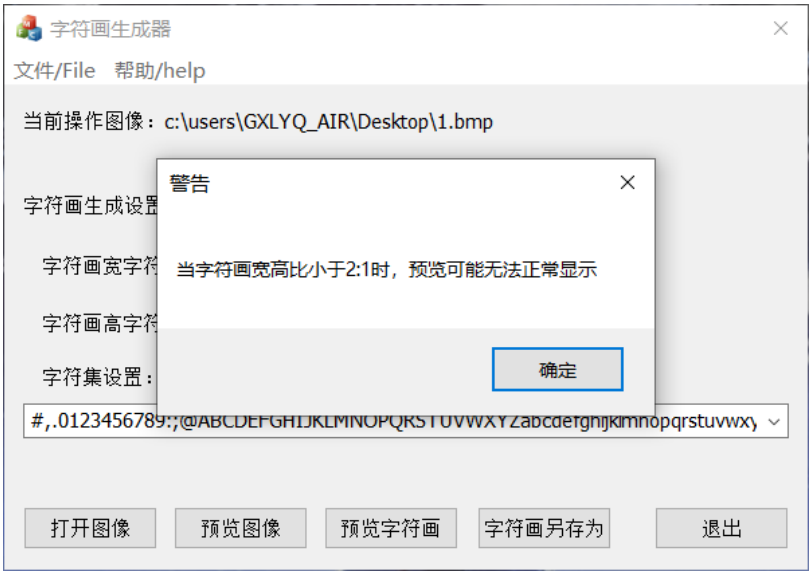


- 在未指定字符集的情况下进行字符画相关操作均会弹出提示并终止操作



- 在预览字符画可能无法正确显示时弹出提示





关于字符画预览的说明

字符数的限制

预览将在新窗口中进行，其中有一个EditBox。由于窗口大小的限制，因此**预览只能显示最多横向1000个字符以及纵向500个字符**

字体的选择

由于是将像素点替换为字符，因此字符画的展示对于字符呈现有着极高的要求：

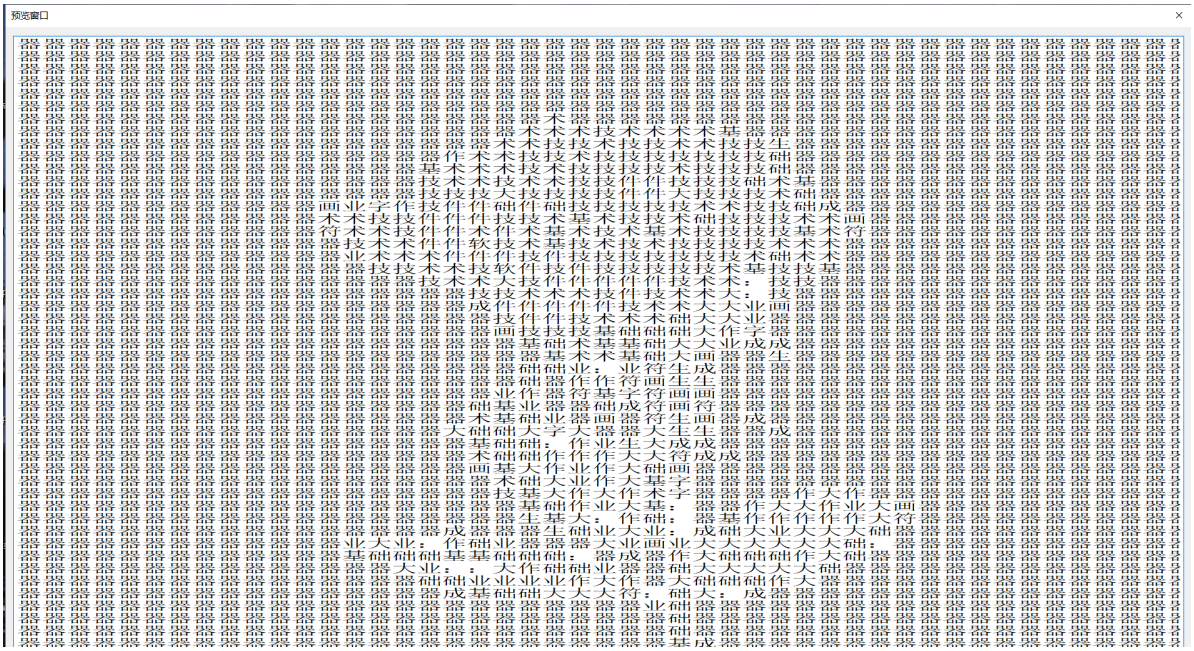
- 字体需要为等宽字体
- 字符的展示中，行高要与字高一致（这点在vscode，记事本等一系列文本查看软件中均较难实现，推荐使用命令行窗口）

为了实现该要求，开发过程中采用了自定义字体，具体如下：

```
    LOGFONT lf; //定义等宽登高字体结构
    lf.lfWeight = 1; //字体磅数

    lf.lfHeight = (1000 / wordW < 1)? 1: 1000 / wordW; //字体高度
    lf.lfWidth = (1000 / wordW < 1) ? 1 : 1000 / wordW; //字体宽度
    //以下代码理论上是上方两行代码的优化形式，因其可以通过同时计算预览字符画的宽和高来调整字体大小
    //但是实际测试中发现，即使调试的时候，字体的宽和高都是相等的，最后两者显示出来的效果却是不同的
    //对此我也很奇怪，但既然这代码就是针对用户违规操作做的优化，最后就放弃优化了（破防了）。
    /*if (1000 / wordW < 1 || 500 / wordH < 1) {
        lf.lfHeight = lf.lfWeight = 1;
    }
    else if (1000 / wordW <= 500 / wordH) {
        lf.lfHeight = lf.lfWeight = 1000 / wordW;
    }
    else if (1000 / wordW > 500 / wordH) {
        lf.lfHeight =  lf.lfWeight = 500 / wordH;
    }*/
    lf.lfUnderline = FALSE; //无下划线
    lf.lfStrikeOut = FALSE; //无删除线
    lf.lfItalic = FALSE; //非斜体
    lf.lfEscapement = 0;
    lf.lfCharSet = DEFAULT_CHARSET; //使用缺省字符集

    CFont myFont; //定义字体对象
    myFont.CreateFontIndirect(&lf); //创建逻辑字体

    GetDlgItem(IDC_EDIT1)->SetFont(&myFont);
    SetDlgItemText(IDC_EDIT1, _T(""));
    SetDlgItemText(IDC_EDIT1,charToSend);
```

字符集为中文时的情况

当字符集为中文时，字符画预览时会出现单个字的宽度是高度的两倍的情况（具体情况如图），原因应该是因为中文字为全角字符。而且当字符集为中文时，字符画的保存将会变得异常。故限制了字符集中不允许出现中文。


预览窗口