前端GUI界面设计与实现

考虑到与用户交互的方便性与简洁性，我们决定在前端设计一个GUI界面。该界面采用新版MATLAB自带的App设计工具——App Designer。

（一）使用App Designer进行前端设计的原因

1. App Designer提供了大量的可视化组件，包括按钮、复选框等标准组件、指示灯等控件和选项卡、网格布局等容器组件，并且只需将可视化组件拖放到设计画布就可以实现布局，编程难度较低。
2. App Designer可以使用集成的MATLAB编辑器来定义App的行为，每个可视化组件都可以通过指定回调函数的方式来指定该组件的行为。

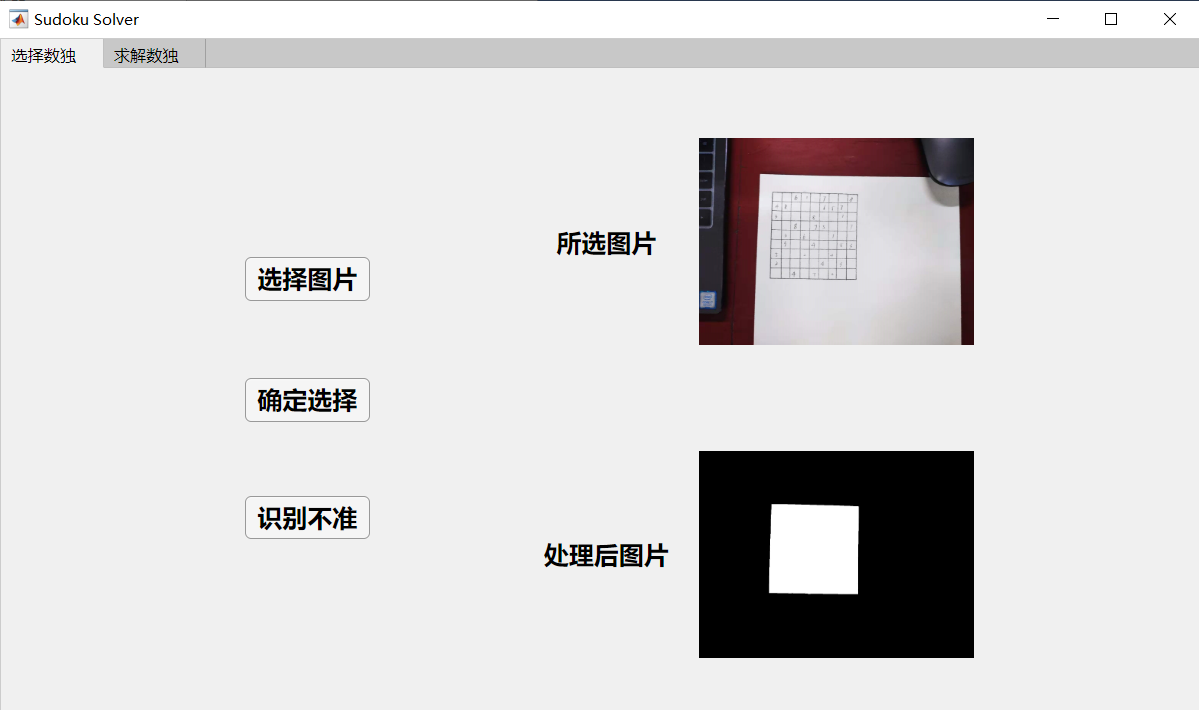
图5-1 GUI界面概览及其使用的部分可视化组件——以“求解数独”页面为例

（二）GUI界面使用方法

1. 运行sudoku.mlapp，进入“选择数独”页面。点击“选择图片”，选择一张包含数独的图片，要求格式为.jpg或.png；

图5-2 初启完成的GUI界面

1. 选择完毕后“所选图片”被自动置为用户选择的图片，“处理后图片”为图像处理部分自动识别的数独轮廓蒙版；

图5-3 用户点击“选择图片”并选择一张图片文件后的界面显示

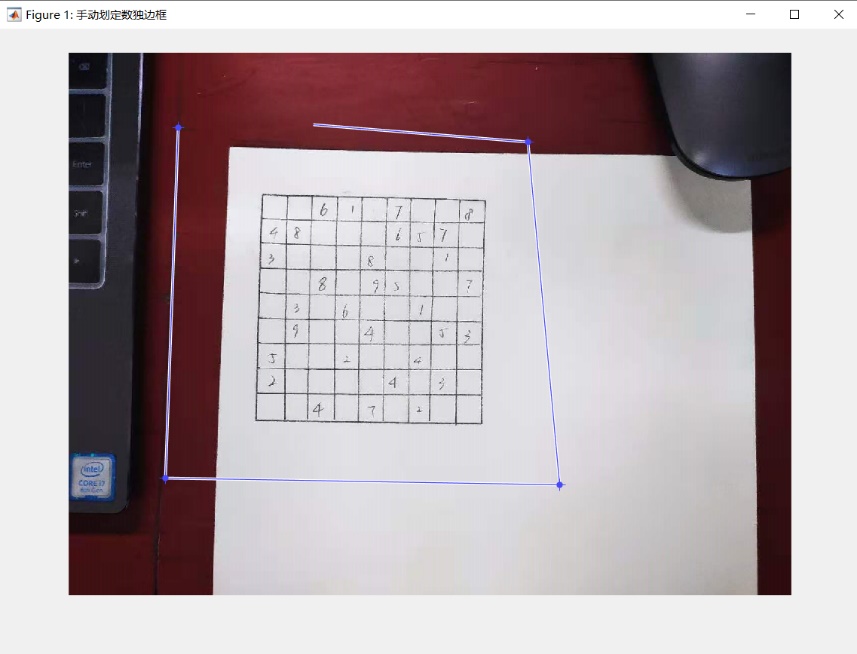
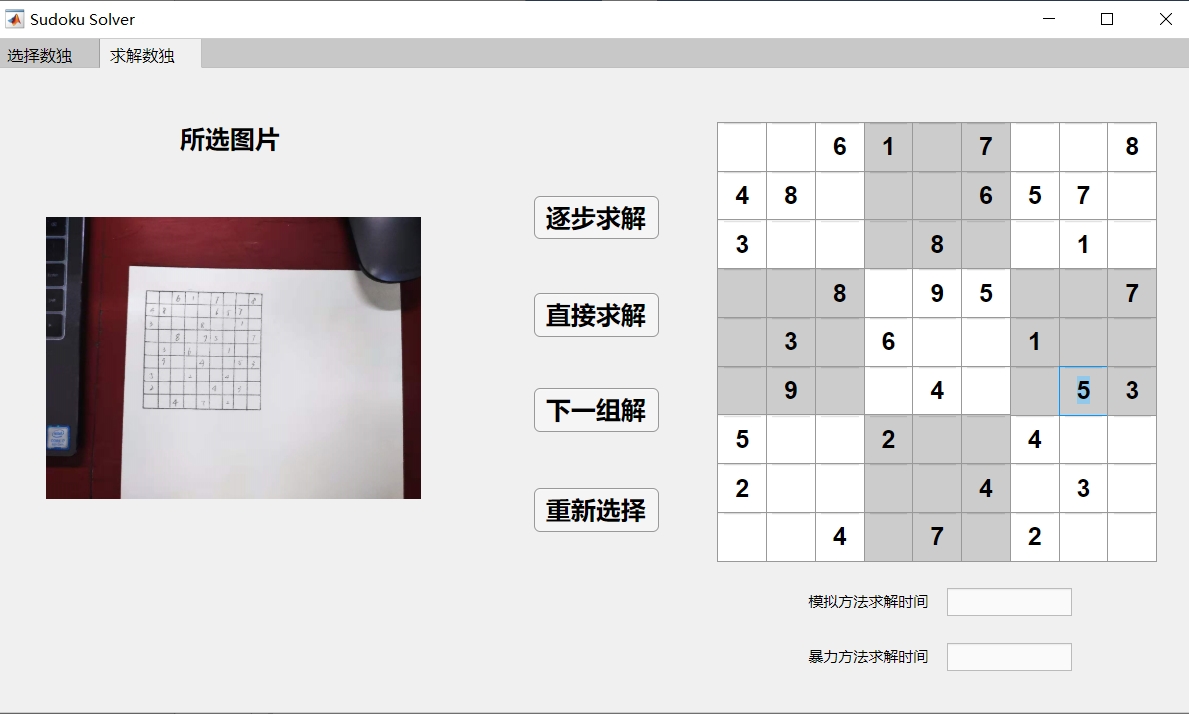
1. 若用户认为自动识别的数独轮廓蒙版不准确，可点击“识别不准”手动标注所选图片中的数独轮廓；

图5-4 用户点击“识别不准”并重新标注数独轮廓

1. 点击“确定选择”，界面自动跳转到“求解数独”页面。此页面中“所选图片”为用户选择的图片，右侧9×9的方格已被自动填入了图像识别部分识别出的数独初始内容；
2. 若用户发现图像识别部分识别出的数独初始内容不准，可在右侧九宫格中手动输入正确数字；

图5-5 用户点击“确定选择”后的界面显示，九宫格6行8列为手动输入

1. 点击“逐步求解”按钮或“直接求解”按钮，九宫格下方会分别显示两种方法解数独所花时间。若点击“逐步求解”按钮，九宫格中会每隔1.2秒填入一个数字，表示按照解数独的方法当前步骤需在此位置填入此数字。若点击“直接求解”按钮，九宫格中会直接显示此数独的解。若该数独无解，界面会显示“数独无解！”；

图5-6 用户点击“逐步求解”，自动逐步求解数独，标红为当前步骤要填的数字

1. 若数独有多个解，点击“下一组解”按钮，九宫格中会直接显示此数独的下一组解；

图5-7 求解完毕的数独

1. 若想要选择新的图片继续求解，点击“重新选择”按钮会清空九宫格与图片内容，同时跳转到“选择数独”页面以供用户重新选择一张图片。

（三）GUI界面的代码实现

由于App Designer的人性化设计，在添加组件时，组件的位置、大小、内容、样式（如按钮大小、九宫格底色等）代码可由App Designer自动生成，开发者只需手动输入几个参数，如此大大简化了设计界面布局的难度。因此相比于界面布局的设计，开发者更需关心的是各个组件行为的设计。

在App Designer设计视图，右键点击某个组件即可为其添加或指定某个动作的回调函数。例如对于“选择图片”按钮，我们定义行为是“点击该按钮后”可选择一张图片并显示在界面上，则可为其添加“ButtonPushedFcn”回调，表示按钮被点击时，回调函数会被执行。

图5-8 点击“选择图片”按钮后会执行的回调函数

诸如此类，为每个按钮添加“ButtonPushedFuc”指定其被点击后的行为、为九宫格的每个格子（实际每个格子是一个文本编辑字段）添加“ValueChangedFuc”指定其中数值被用户修改后的行为，即可定义整个GUI界面中所有组件的行为。

在鲁棒性控制方面，主要是控制用户不合法的按钮点击。由于界面中各个按钮存在逻辑上的先后关系（如只有点击“选择图片”之后才能点击“确定选择”），需在点击某个按钮后确认其是否合法，只需为按钮指定逻辑变量并进行简单检查即可。