

项目 5



图像处理

本项目基于 Python 语言,借助 PyQt5 开发图形,调用 Stable-Diffusion-XL 模型进行智能图像处理的,实现 AI 作画功能。

5.1 总体设计

本部分包括整体框架和系统流程。

5.1.1 整体框架

整体框架如图 5-1 所示。



项目 5
教学资源

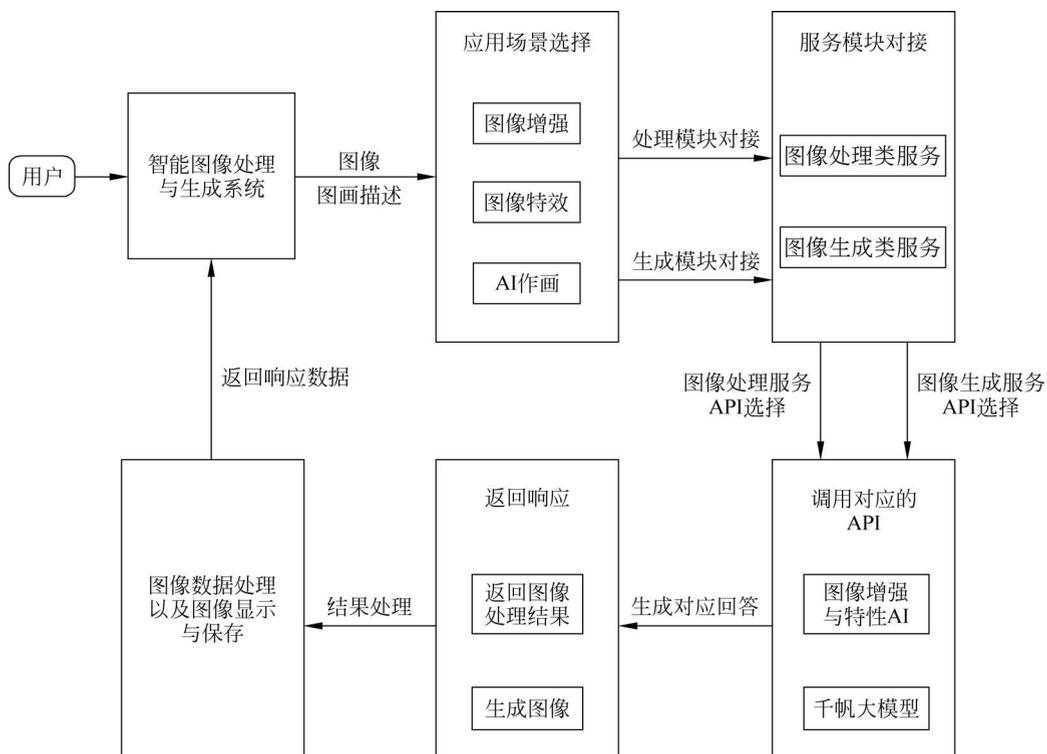


图 5-1 整体框架

5.1.2 系统流程

系统流程如图 5-2 所示。

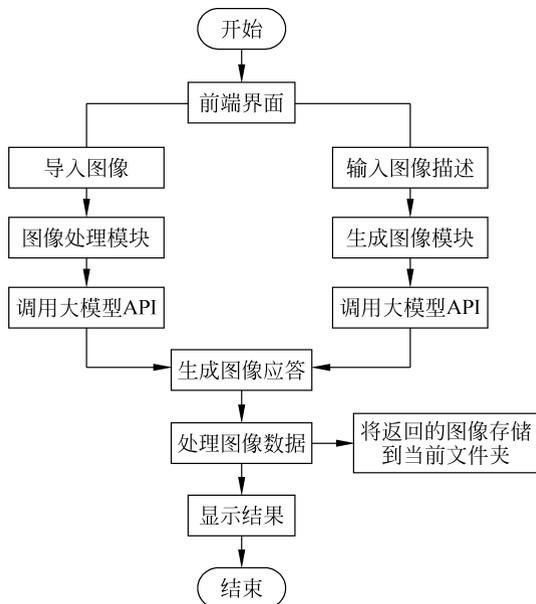


图 5-2 系统流程

5.2 开发环境

本节主要介绍 PyQt5 的安装过程,给出所需要的环境配置并介绍大模型 API 的申请步骤。

5.2.1 安装 PyQt5

在命令行中输入“pip install PyQt5”语句,使用 pip 命令在线安装 PyQt5,如图 5-3 所示。

在命令行中输入“pip install PyQt5-tools”语句,安装 PyQt5 的常用工具,如图 5-4 所示。

在命令行中输入“pip list”语句,检查是否已有 PyQt5,若存在于列表中则证明安装成功,如图 5-5 所示。

5.2.2 环境配置

在 PyCharm 中,依次单击 File→Settings→Tools→External Tools→Add,实现添加外部工具,如图 5-6 所示。

```
C:\Users\LENOVO>pip install PyQt5
Defaulting to user installation because normal site-packages is not writeable
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
Requirement already satisfied: PyQt5 in c:\users\lenovo\appdata\roaming\python\python39\site-packages (5.15.10)
Requirement already satisfied: PyQt5-sip<13, >=12.13 in c:\users\lenovo\appdata\roaming\python\python39\site-packages (from PyQt5) (12.13.0)
Requirement already satisfied: PyQt5-Qt5>=5.15.2 in c:\users\lenovo\appdata\roaming\python\python39\site-packages (from PyQt5) (5.15.2)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
[notice] A new release of pip is available: 23.1.2 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

图 5-3 安装 PyQt5

```
C:\Users\LENOVO>pip install PyQt5 -tools
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
Collecting PyQt5
  Downloading PyQt5-5.15.10-cp37-abi3-win_amd64.whl (6.8 MB)
    6.8/6.8 MB 3.5 MB/s eta 0:00:00
Collecting PyQt5-sip<13, >=12.13 (from PyQt5)
  Downloading PyQt5-sip-12.13.0-cp39-cp39-win_amd64.whl (78 kB)
    78.5/78.5 kB 4.3 MB/s eta 0:00:00
Collecting PyQt5-Qt5>=5.15.2 (from PyQt5)
  Downloading PyQt5-Qt5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
    50.1/50.1 MB 7.0 MB/s eta 0:00:00
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
Installing collected packages: PyQt5-Qt5, PyQt5-sip, PyQt5
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
spyder 5.1.5 requires pyqtwebengine<5.13, which is not installed.
pyqt5-plugins 5.15.9.2.3 requires pyqt5==5.15.9, but you have pyqt5 5.15.10 which is incompatible.
pyqt5-tools 5.15.9.3.3 requires pyqt5==5.15.9, but you have pyqt5 5.15.10 which is incompatible.
spyder 5.1.5 requires pyqt5<5.13, but you have pyqt5 5.15.10 which is incompatible.
Successfully installed PyQt5-5.15.10 PyQt5-Qt5-5.15.2 PyQt5-sip-12.13.0
WARNING: Ignoring invalid distribution -ensorflow-intel (c:\users\lenovo\appdata\roaming\python\python39\site-packages)
[notice] A new release of pip is available: 23.1.2 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

图 5-4 安装 PyQt5 常用工具

```
PyQt5 5.15.10
pyqt5-plugins 5.15.9.2.3
PyQt5-Qt5 5.15.2
PyQt5-sip 12.13.0
PyQt5-stubs 5.15.6.0
pyqt5-tools 5.15.9.3.3
```

图 5-5 检查 PyQt5 安装

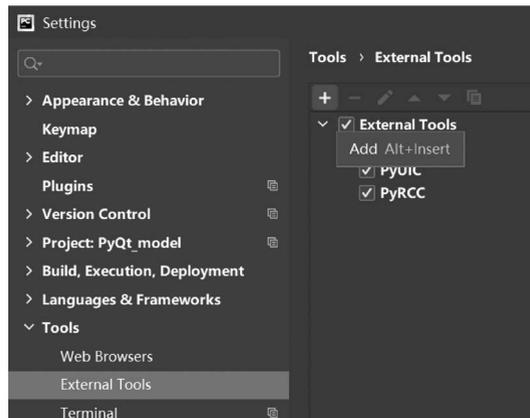


图 5-6 添加外部工具

根据安装资源在磁盘中的具体位置,分别完成对 Qt Designer、PyUIC 和 PyRCC 外部工具的配置,如图 5-7~图 5-9 所示。

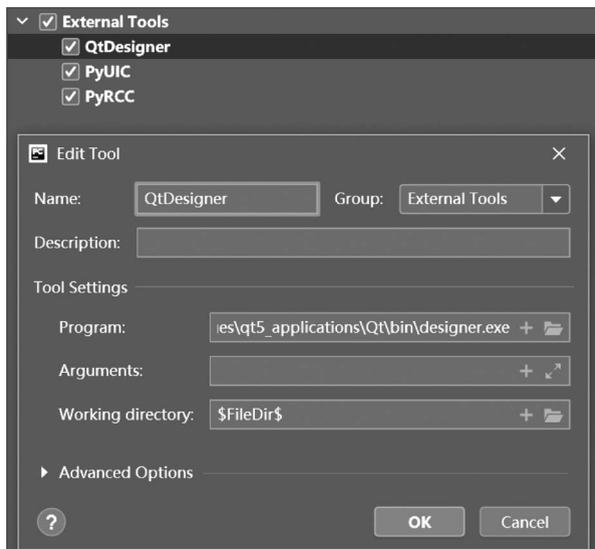


图 5-7 配置 Qt Designer

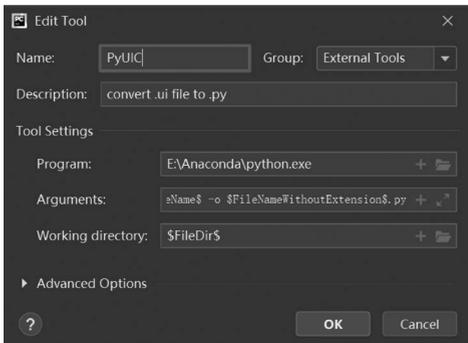


图 5-8 配置 PyUIC

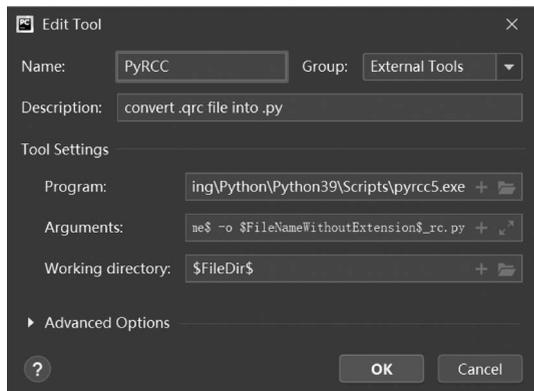


图 5-9 配置 PyRCC

依次单击 Tools→External Tools→Qt Designer,打开 PyQt5 界面,如图 5-10 所示;PyQt5 设计界面如图 5-11 所示。

5.2.3 大模型 API 申请

百度智能云千帆大模型 API 申请参见 1.2.4 节。

应用创建成功后,默认开通所有 API 的调用权限,无须申请授权。然后根据 APIKey 和 SecretKey,使用 Python 方法获取 access_token,相关代码如下。

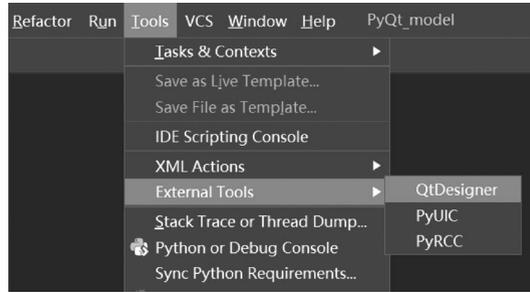


图 5-10 打开 PyQt5 界面

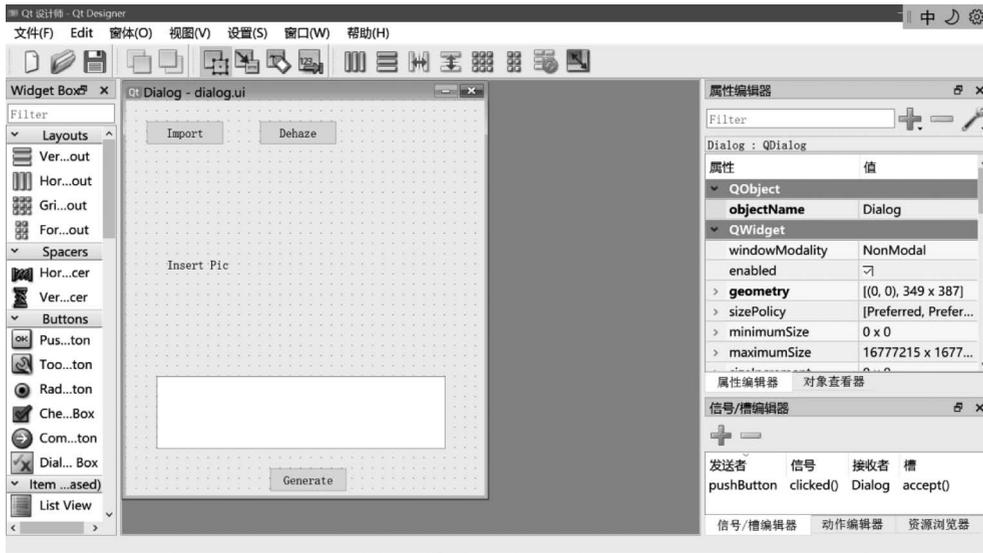


图 5-11 PyQt5 设计界面

```

import requests
import json
def main():
    url = "https://aip.baidubce.com/oauth/2.0/token?client_id = 【ApiKey】&client_secret =
【SecretKey】&grant_type = client_credentials"
    payload = json.dumps("")
    headers = {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    }
    response = requests.request("POST", url, headers = headers, data = payload)
    return response.json().get("access_token")
if __name__ == '__main__':
    access_token = main()
print(access_token)

```

预期功能根据用户输入的文本生成图像,所以需要调用 Stable-Diffusion-X 大模型 API,相关代码如下。

```

import requests
import json
API_KEY = "x0G0pviGV6M878f9ITBVz5dH"
SECRET_KEY = "ZhrsqsldZEj6NiSBQ1BTLRAYHGpf9rT3"
def main():
    url = "https://aip.baidubce.com/rpc/2.0/ai_custom/v1/wenxinworkshop/text2image/sd_xl?
access_token = " + get_access_token()
    payload = json.dumps({
        "size": "1024x1024",
        "n": 1,
        "steps": 20,
        "sampler_index": "Euler a"
    })
    headers = {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    }
    response = requests.request("POST", url, headers = headers, data = payload)
    print(response.text)
def get_access_token():
    url = "https://aip.baidubce.com/oauth/2.0/token"
    params = {"grant_type": "client_credentials", "client_id": API_KEY, "client_secret":
SECRET_KEY}
    return str(requests.post(url, params = params).json().get("access_token"))
if __name__ == '__main__':
    main()

```

5.3 系统实现

本项目使用 PyCharm 和 PyQt5 构建后端与前端系统,文件结构如图 5-12 所示。

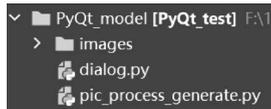


图 5-12 文件结构

5.3.1 PyQt5 组件初始化与绑定机制

定义 PyQt5 的 QWidget 类之后完成对界面中控件的布局及绑定设置(初始化在 dialog.py 中完成),相关代码见“代码文件 5-1”。

5.3.2 PyQt5 槽函数的定义

当程序触发某种状态或者发生某种事件时会发出一个信号,若程序想捕获这个信号,需要执行相应的逻辑代码,这个过程会用到“槽”,“槽”实际上是一个函数,当信号发射后,执行与之绑定的槽函数,故在槽函数中完成与信号相对应的功能设置,实现参数的传递。

图像处理槽函数(以图像去雾功能为例)相关代码见“代码文件 5-2”。

5.3.3 主函数

主函数相关代码如下。

```
if __name__ == "__main__":  
    app = QtWidgets.QApplication(sys.argv)           # 有且只有一个 QApplication 对象  
    my = Master()                                   # 实例化 Master 对象  
    my.show()                                       # 展示窗口  
    sys.exit(app.exec_())                          # 程序进行循环等待状态,直到关闭窗口
```

5.4 功能测试

本部分包括图像处理功能测试及图像生成功能测试。

5.4.1 图像处理功能测试

运行主程序,进入系统初始界面,如图 5-13 所示。



图 5-13 系统初始界面

单击“导入图像”按钮,在本地导入 .png 或 .jpg 格式的图像,如图 5-14 所示。

色彩增强原始图像如图 5-15 所示。

单击“色彩增强”按钮,处理结果显示在原始图像的位置,图像如果发生变化,说明色彩

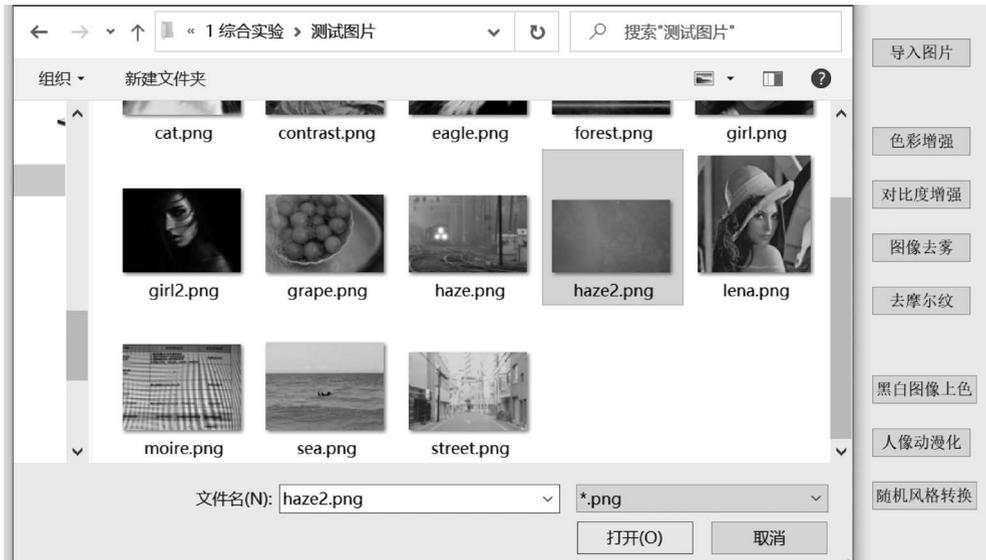


图 5-14 导入图像



图 5-15 色彩增强原始图像

增强功能有效果,处理后的图像将自动保存在 images 文件夹中,如图 5-16 所示。

图像对比度增强原始图像如图 5-17 所示。

单击“对比度增强”按钮,处理结果显示在原始图像的位置,图像如果发生明显的变化,说明对比度增强功能有效果,处理后的图像将自动保存在 images 文件夹中,如图 5-18 所示。



图 5-16 色彩增强处理结果



图 5-17 图像对比度增强原始图像

图像去雾原始图像如图 5-19 所示。

单击“图像去雾”按钮,处理结果显示在原始图像的位置,图像如果发生明显的变化,说明去雾功能的效果较为明显,处理过的图像自动保存在 images 文件夹中,如图 5-20 所示。



图 5-18 对比度增强处理结果



图 5-19 图像去雾原始图像

黑白图像上色原始图像如图 5-21 所示。

单击“黑白图像上色”按钮,处理结果显示在原始图像的位置,图像如果发生明显的变化,说明黑白图像上色功能有效果,处理后的图像将自动保存在 images 文件夹中。



图 5-20 图像去雾处理结果



图 5-21 黑白图像上色原始图像

随机风格转换原始图像如图 5-22 所示。

单击“随机风格转换”按钮,处理结果显示在原始图像的位置,展示风格转换的结果自动保存在 images 文件夹中,如图 5-23~图 5-25 所示。



图 5-22 随机风格转换原始图像

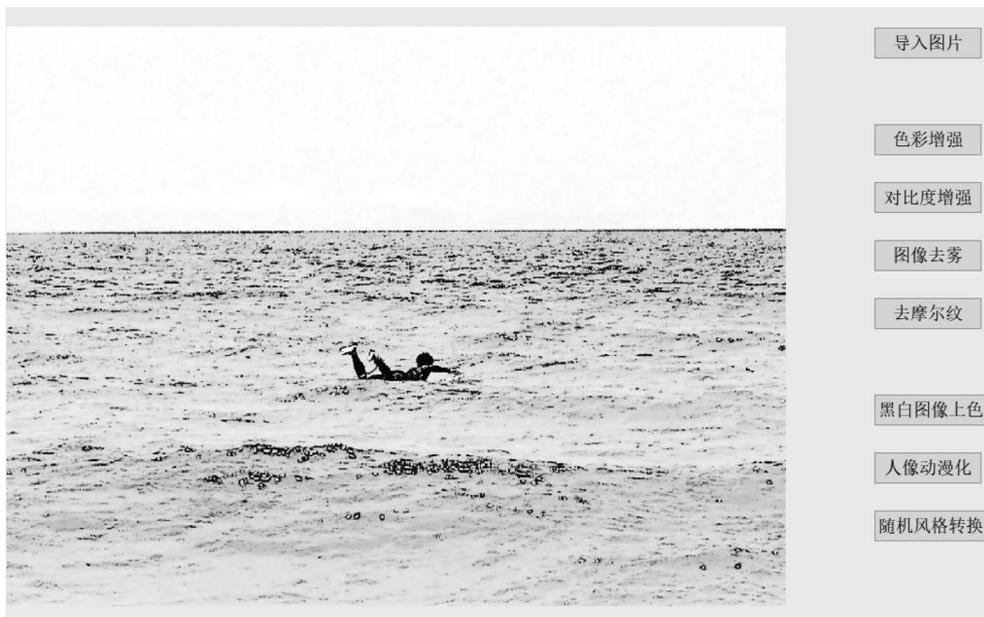


图 5-23 随机风格转换结果(风格: color_pencil)



图 5-24 随机风格转换结果(风格: lavender)

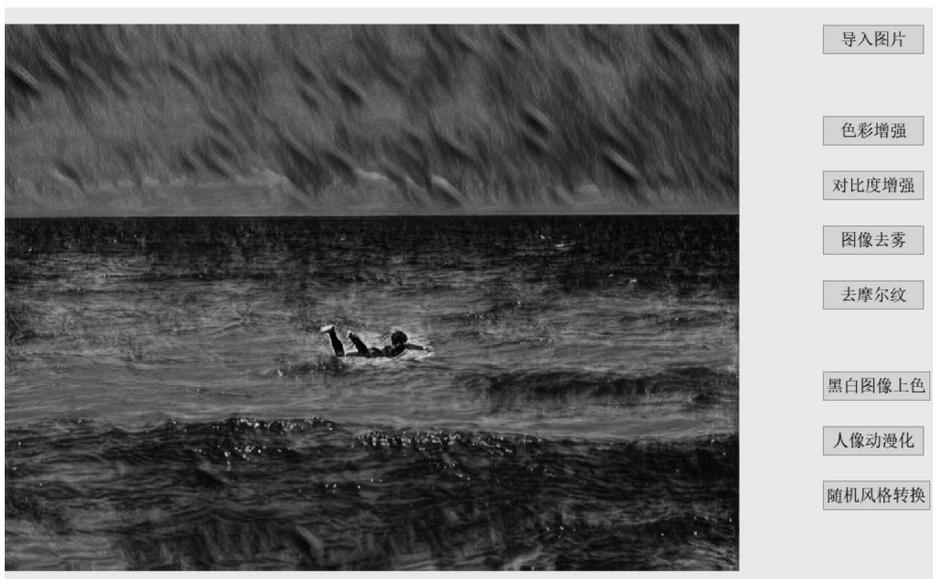


图 5-25 随机风格转换结果(风格: scream)

5.4.2 图像生成功能测试

在界面下方的输入框中,输入图像描述,然后单击“AI 作画”按钮,调用大模型生成的图像显示在画面上方的空白区域中,AI 作画结果如图 5-26~图 5-28 所示。



图 5-26 高耸大楼

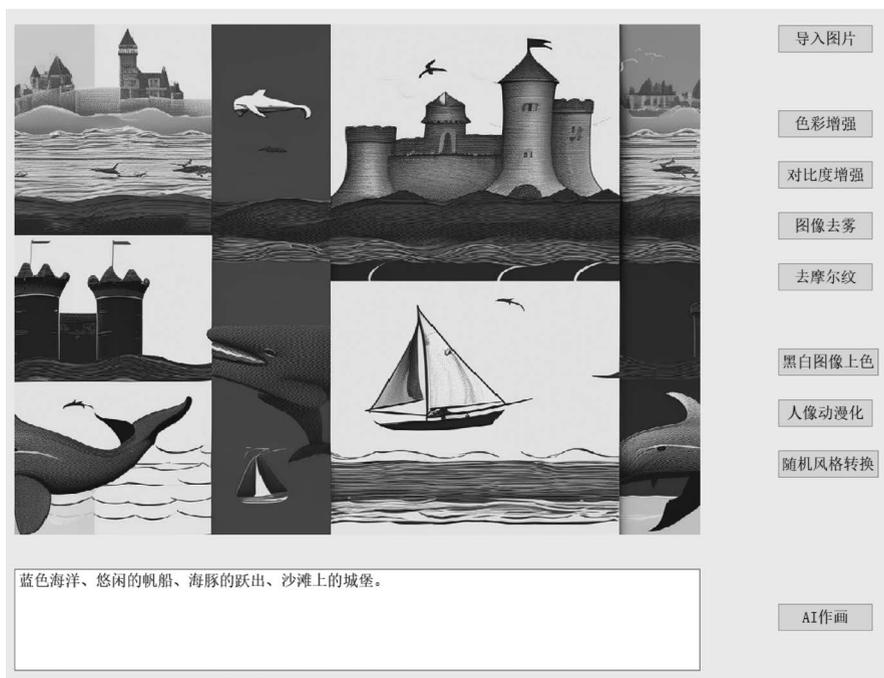


图 5-27 蓝色海洋

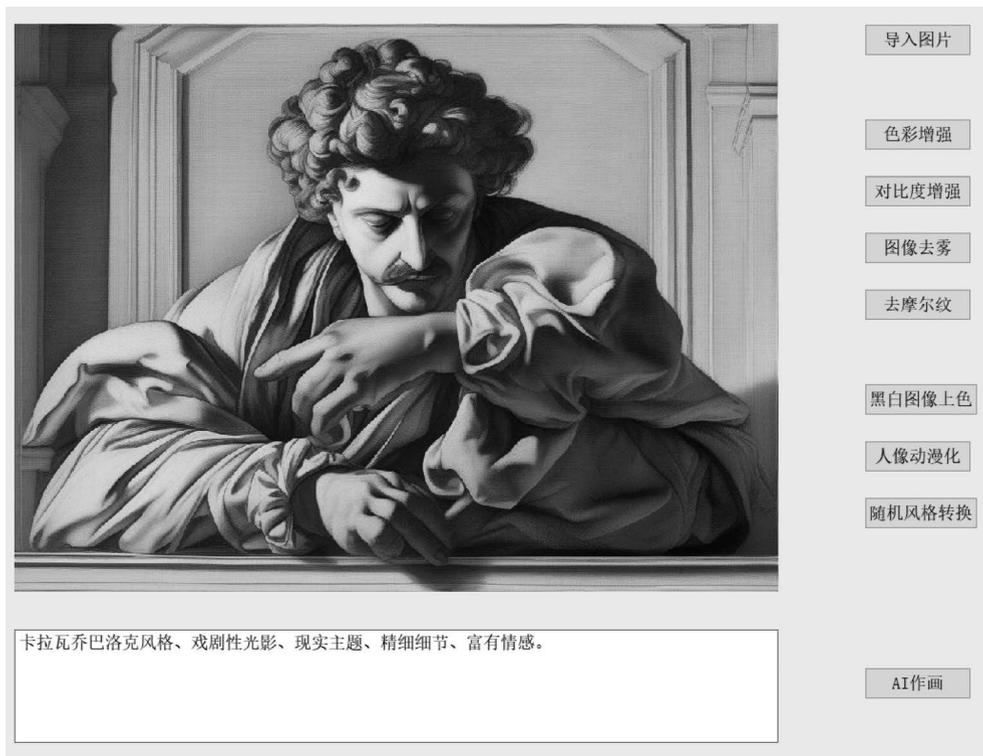


图 5-28 卡拉瓦乔巴洛克风格