网络爬虫和图形用户界面

Jonathon Chow *

（中国科学技术大学数学科学学院）

2022/4/24

摘要

本次实验主要实现了一个与网络相关的软件，主要实现了两个功能，分别是爬取相关网页的信息以及对图片进行一些简单的处理。同时，利用图形用户界面将这些功能展示出来。本次实验主要目的是熟悉 Python 语法、相关库的使用、相关软件的操作，并锻炼获取网络资源的能力。

本次实验的代码地址参见https://github.com/JONATHONCHOW/webcrawler

1 任务说明

首先，为实现爬取豆瓣 TOP250 排行榜的相关数据，包括电影信息以及海报图片。我们使用 requests 库获取网页的 url，然后利用 xpath 语法对网页解码，最后将获取到的数据下载到本地。

其次，为实现一些简单的图像处理的功能，包括模糊、灰度、轮廓。我们调用 cv2 库，读取目标图像并利用相关的函数进行处理。

最后，为将实现的功能交互的展示出来，包括输入信息、点击按钮、显示结果。我们利用 pyqt5 库，并使用 QtDesigner 设计图形用户界面，利用控件的信号与槽的功能实现交互。

2 代码实现

我们主要分为两部分代码，分别是爬虫部分（webcrawler.py）和图形用户界面部分（gui.py&main.py）。

*E-mail: jonathonchow23@gmail.com

2.1 爬虫部分：webcrawler.py

我们定义了三个函数，分别实现了获取源代码、解析源代码、下载网页信息的功能。我们使用 requests 库获取网页的 url，然后利用 xpath 语法对网页解码，最后将获取到的数据下载到本地。我们将这些程序存放在

```
1    def get_html(url):
2    def parse_html(html):
3    def downloading(url,movie):
```

2.2 图形用户界面部分：gui.py&main.py

首先，我们使用 QtDesigner 设计图形用户界面，包括控件的属性、信号与槽。然后使用 PyUIC 自动生成 gui.py，其中包括一个类。

```
1    class Ui_form(object):
2        def setupUi(self, form):
3        def retranslateUi(self, form):
```

其次，我们在 main.py 中调用 webcrawler.py 和 gui.py，并且使用库 PyQt5。

```
1    from PyQt5 import QtWidgets, QtGui
2    from webcrawler import *
3    from gui import Ui_form
```

再次，我们在 main.py 中定义了一个类，实现文字和图像信息的显示的功能。

```
1    class mywindow(QtWidgets.QWidget,Ui_form):
2        def __init__(self):
3        def showtext(self):
4        def showpicture(self):
```

最后，我们调用 cv2 库，在上述同一个类中定义了三个函数，实现一些简单的图像处理的功能，包括模糊、灰度、轮廓。

```
1    class mywindow(QtWidgets.QWidget,Ui_form):
2        def blur(self):
3        def gray(self):
4        def edge(self):
```

注意到，gui.py 和 main.py 是分离的，这也就 PyQt5 库的优点，实现了有关图形界面的代码和有关功能实现的代码的分离。

3 效果展示

3.1 爬虫部分

我们运行 webcrawler.py 获取相关网页的文字信息和海报图片，分别存放在文件 movie.csv 和文件夹 movieposter 中。

3.2 查询部分

我们接下来运行 main.py 生成图形用户界面。输入你想查询的电影排名，点击查询，就会反馈对应的电影信息以及海报图片。

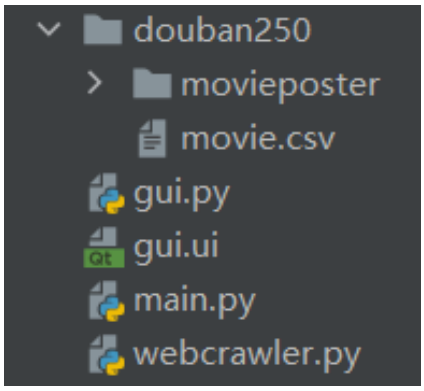


图 1: 爬虫

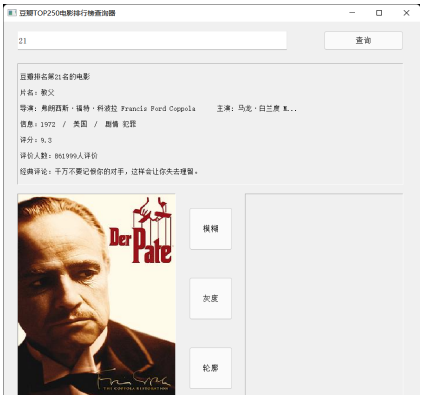


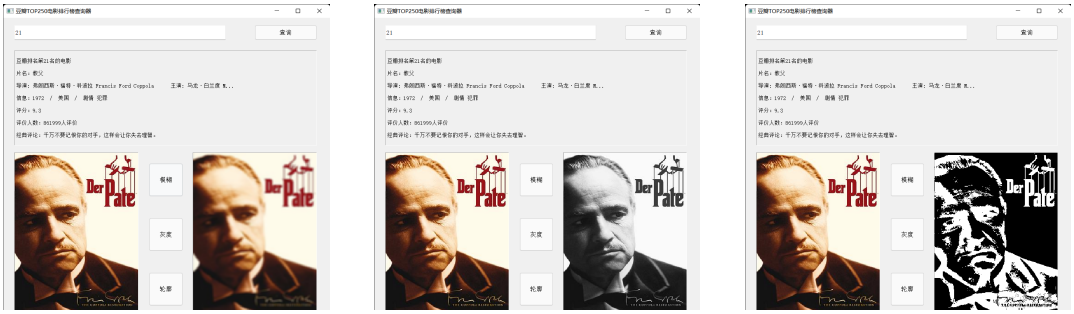
图 2: 查询

3.3 图像处理部分

我们对查询到的海报图片，点击模糊、灰度、轮廓的按钮，就会相应的得到经过模糊、灰度、轮廓处理后的海报图片。

4 实验难点

此次实验遇到了一些困难，主要得到两点经验。第一是多用网络资源不要自己造轮子，第二是多练习才会进步。我们将一些典型的 tips 列在下面。



(a) 模糊　(b) 灰度　(c) 轮廓

图 3: 图像处理

4.1 有关爬虫

（1）UA 代理不能开 VPN
（2）网址的规律，熟悉 xpath 语句结构

4.2 有关 PyQt

（1）控件：Qlabel（超级牛逼），单行文本框，多行文本框，按钮，Qmessagebox
（2）属性：Frame，字体字号，PlaceholderText，Title
（3）信号与槽：clicked connect，新建槽函数，可以一 click 多 slot
（4）注意控件命名很重要：区分名字、文件名、显示的文本，前两者在代码中体现，后一者在图形界面显示
（5）转化成 py 文件的方法很有用
（6）图片显示方法

4.3 代码基础

（1）类的封装
（2）全局变量，局部变量
（3）多个函数，多个类，多个 py 文件
（4）代码可读性，注释，书写习惯
（5）软件安装问题

4.4 一些库

（1）cv2 库的使用：图片读取只能英文路径，cv 图片转换成 qt 图片，cv2 的图片处理函数

（2）os 库的使用：路径问题

（3）csv 文件的操作：用 pandas，numpy 库

5 实验总结

通过本次实验，笔者较好的完成了实验任务，熟悉了 Python 语法，了解了一些外部库，比如 numpy、requests、pyqt5、cv2，使用了一些软件，比如 Anaconda，Jupyter，Spyder，PyCharm，QtDesigner，锻炼了对网络资源获取的能力，比如 CSDN、Github，受益匪浅。

参考文献

[1] 张越一：Python 与深度学习基础课件

[2] Python 爬虫教程（从入门到精通）：http://c.biancheng.net/python_spider/

[3] 图形界面程序：https://www.byhy.net/tut/py/gui/qt_01/

附录

历史版本记录

由于笔者疏忽，在本人的 Github 上直接提交了最终的版本，故将版本的迭代过程记录如下。

01 爬取网页的文字信息：只实现了从网页爬取豆瓣 TOP250 排行榜的文字信息。

```
52      os.chdir('douban250/')
53      moviedata = pd.DataFrame(movies1)
54      moviedata.to_csv('movie.csv')
```

图 4: 第一版

02 建立查询文字信息的图形用户界面：建立一个供用户查询电影文字信息的图形用户界面，利用 Qmessagebox 弹窗显示文字信息。

图 5: 第二版 1

图 6: 第二版 2

```python
def downloading(url,movie):
    if 'movieposter' in os.listdir('douban250/'):
        pass
    else:
        os.mkdir('douban250/movieposter/')
    os.chdir('douban250/movieposter/')
    img = requests.get(url).content
    with open(movie['rank']+'.jpg','wb') as f:
        print("正在下载：%s" % url)
        f.write(img)
    os.chdir('..')
    os.chdir('..')

```

图 7: 第三版

03 爬取网页的图片并下载：添加了从网页爬取海报图片的功能。

04 建立显示文字信息和海报图片的图形用户界面：利用 Qlabel 将文字信息和海报图片显示在图形用户界面上。

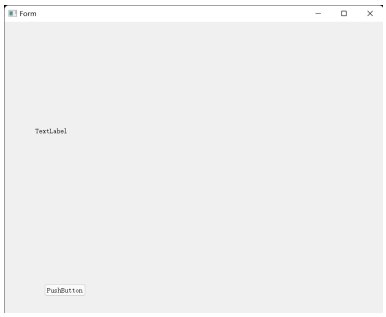


图 8: 第四版 1

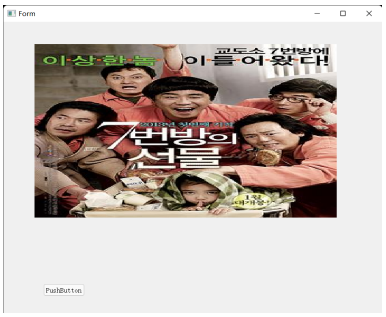


图 9: 第四版 2

05 最终版：添加了图像处理的功能。具体介绍参见正文。