

# Python程序设计报告

设计题目：爬取百度图片

学生姓名：曹辰鹏

专 业：计算机科学与技术

班 级：19-4

学 号：2019218117

指导教师：马学森

完成日期：2021.1.3

# （一）需求和规格说明

**问题描述：**

这是一款爬取百度图片的一个小爬虫，不仅能爬取图片，还能根据关键词选取你可能喜欢的其他关键词

**编程任务：**

1. 在百度图片网站上获取信息，获得相应的信息。
2. 下载图片
3. 获得推荐信息

# 设计

## 1．设计思想

这道题用的就是爬虫。我们先用爬虫将网站上的内容爬下来，之后根据内容分析得到图片的地址，再根据这个地址获得图片，写到本地的图片中。在最后用函数再次得到网站的信息，之后根据这个信息得到百度推荐的内容，显示到屏幕上。

## 2. 设计表示

除了采用如下方式（注意：下面的描述是C++类成员描述，要根据Python语言的相应的修订），还可以给出流程图，IPO等各种设计表示。

1. 函数：

|  |  |  |
| --- | --- | --- |
| **数据类型** | **函数名称** | **描述** |
| int | Search(url,r) | 获得和关键字匹配的信息，得到最大的下载数量（默认最多是1020） |
| 列表 | recom(url) | 爬取网站上的推荐内容 |
| void | downloa | 得到网站的图片地址后，按照这些地址爬取，得到图片 |

1. 全局变量：

|  |  |
| --- | --- |
| 变量名 | 作用 |
| numPicture = 0 | 存储需要图片的数量 |
| fileName = '' | 得到创建文件的名字 |
| l = [] | 存放图片的url |
| num = 0 | 表示正在下载图片的张数 |

## 核心算法

核心算法主要在最后识别主函数的那一块

在最后的主函数中，整个程序聚合到一起运行。首先，我们从百度上得到百度图片网站的地址，但是有一点就是那个关键字的位置需要根据用户的需要放入对应的字符串。所以在地址上，我们要在关键的检索词的那一块放入用户的关键字。当然关键字从输入端获得。之后我们调用Search函数得到检索图片的数量，当然这个过程也将所有图片的地址存放到了l中。之后再得到需要下载的数量和存储文件的文件夹名字。创建文件夹。通过一个循环不断访问图片地址，调用download不断下载。之后再调用函数recommend来得到推荐内容输出。

在Search函数中，每次加60不断访问百度图片的地址，记录图片数量的同时也将这些地址的放到列表l中。这一步给后来的下载图片提供了前提。每一个地址实际上都是相隔60个单位的。

在download函数中，根据传入的网页内容找到所有图片地址并放入列表。之后遍历列表，以列表的元素为地址，得到每张图片，并进入到那个创建好的文件夹中，创建文件以二进制的形式输入，进而实现下载图片。

在recommend函数中，我们还爬到Search的那个网址上。得到网址内容之后，找到推荐栏的地方，将这上边的内容取出。再在这个基础上找到推荐的内容，将这些内容放到列表中，最后返回存放推荐列表。

# 用户手册

按照提示输入相关信息，但是每一次使用程序都会认为是第一次使用这个程序，每次下载文件都要建立新的文件夹。

首先输入想要爬取图片的名字，可以是人名，地名等。就算是一堆无意义的数字，百度上也会自动解析并检索出它认为你可能想找的图片，这个时候就会出现一个网址，我们爬取的就是这个网址上的内容。当然，为确保能够精准找到你想找的，建议用有意义的名字，比如说是明星、名胜区什么的。

爬虫会计算图片的数量，但是最大只计算到1020。虽然这个数量可以通过网页的信息获得，但是为了您的计算机存储空间着想，不会下载过多的图片。

接下来就是输入存储的文件夹名称了。程序会在程序的目录下生成一个专属的文件夹来存放下载的图片。之后就是下载了。

# （四）调试及测试

## 1. 测试数据：



## 屏幕截图 2021-01-06 1108192. 测试结果：

## 3. 进一步改进：

原本是没有推荐功能的，但是发现了网站上有推荐的栏目，所以我就想着这个内容爬下来。本来已经可以了。但是这个推荐内容只有内容，没有连接。有链接的话可以让使用者找到对应的网站预览，看是否要爬，之后再这个程序爬取。

还有就是，每次使用这个程序，程序并不会记录之前爬取结束的位置，所以每一次都是找到网站后从头开始爬取内容。当然，我们可以用文件记录每次爬取的关键词和每次结束的位置，去掉重复，并排序，让用户选择相应的内容。

# 感想

实际上，写这个程序的时候，还是遇到一些困难的。就比如说，在最后的推荐函数的内部，实际上刚开始是和现在内容差不多的。但是程序总是有问题，总是没有办法得到标签的内容。之后，我就在Search函数中，每次得到地址内容后，我就用相同的方法来获取标签内容，但是最后的结果都是正确的。所以到最后，我干脆志杰把原来recommend的内容删掉了，重新写了一遍。最后的结果终于正确了。

整个程序实际上每个函数之间的联系还是蛮大的。Search函数为后来的download打好基础。但中间借用了全局变量，所以两者的是通过这个变量联系的。这算是一点不足之处吧，但是代码是很清晰的。在写这个程序的时候，我对与字符串的应用，爬虫的基本格式还有用程序对我们的电脑操作更加熟悉。也发现了自己对于正则表达式不太熟悉。在写取网站地址的时候不清楚这个应该怎么写，最后还是上网查到，看着理解的。

在学习python期间，真正感受到了python的强大，虽然python的计算速度没有C/C++那么快，但是这门语言其庞大的第三方库为很多的开发都提供了轮子，不需要自己造轮子。我们只需要调用这些函数就行。我们想实现的很多东西，很多有意思的东西都可以通过python实现。还记得我的一个朋友是学习设计的，他电脑上的一些设计软件就是用python写的，他还给我演示了在那个软件中怎么用自定义的python程序画图。还有就是在实现人工智能的时候，python也提供了很多的库。写一个简单的衣服识别只需要几行代码。这对没接触过的人是想都不敢想的。没接触之前，都以为人工智能是很复杂的项目，但是真正接触之后，才发现用python写可以很简单。这个爬虫只是python应用的冰山一角，但是关于爬虫延伸出来的爬虫和反爬虫的斗智斗勇却可以至少写一本书的。

最后，在写这次大作业的时候，是一边对应网站的html格式，一边写的。刚开始，还没找到对应内容的位置，只不过最后一个一个部分翻看找到了需要的内容。当然，这个只是爬取静态网页的，难度也比较低。之后在假期的时候会对更深层的内容去学习。

# 附录：

import re

import time

import requests

from urllib import error

from bs4 import BeautifulSoup

import os

numPicture = 0 #存储需要图片的数量

fileName = '' #得到创建文件的名字

l = [] #存放图片的url

num = 0 #表示正在下载图片的张数

#找出图片的网址信息

def Search(url,r):

global l

print("正在搜索图片，请稍等")

t = 0

i = 1

s = 0

while t < 1000:

Url = url + str(t)

try:

Result = r.get(Url,timeout = 7, allow\_redirects = False)

except BaseException:

t += 60

continue

else:

result = Result.text

pictureUrl = re.findall('"objURL":"(.\*?)",', result, re.S) #利用正则表达式找到图片的url

le = len(pictureUrl)

s += le

if le == 0:

break

else :

l.append(pictureUrl)

t+=60

return s

def recommend(url):

R = []

Url = url + str(0)

try:

Result = r.get(Url,timeout = 7, allow\_redirects = False)

except BaseException:

return

else:

result = Result.text

soup =BeautifulSoup(result,'html.parser')

a = soup.find(id = 'topRS')

la = a.find\_all('a')

for i in la:

if i is not None:

R.append(i.get\_text())

return R

def download(html,keyword):

global num

pictureUrl = re.findall('"objURL":"(.\*?)",', html, re.S) #先用正则表达式找到图片的url

print("找到关键词："+keyword+'的图片，即将开始下载...')

for i in pictureUrl:

print("正在下载第"+str(num+1)+'张图片')

try:

if i is not None:

pic = requests.get(i,timeout = 7)

else:

continue

except BaseException:

print("出现错误，当前图片无法下载")

continue

else :

Str = fileName + r'\\'+keyword + '\_' + str(num) + '.jpg'

fp = open(Str,'wb')

fp.write(pic.content) #

fp.close()

num += 1

if num >= numPicture:

return

#主函数

if \_\_name\_\_ == '\_\_main\_\_':

headers = {

'Accept-Language': 'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2',

'Connection': 'keep-alive',

'User-Agent': 'Mozilla/5.0 (X11; Linux x86\_64; rv:60.0) Gecko/20100101 Firefox/60.0',

'Upgrade-Insecure-Requests': '1'

}

r = requests.Session()

r.headers = headers

name = input("请输入关键词")

url = 'https://image.baidu.com/search/flip?tn=baiduimage&ie=utf-8&word=' + name + '&pn='

a=url

Sum = Search(url,r)

print('检测出{}类图片，有{}张'.format(name,Sum))

numPicture = int(input("请输入想要下载的数量"))

fileName = input("请输入将要建立的专门存储图片的文件夹名称")

x = os.path.exists(fileName)

if x==1:

print("该文件已经存在，请重新输入")

file = input("请输入将要建立的专门存储图片的文件夹名称")

os.mkdir(fileName)

else:

os.mkdir(fileName)

t = 0

tmp = url

while t < numPicture:

try:

url = tmp + str(t)

result = r.get(url,timeout = 10,allow\_redirects = False)

except error.HTTPError :

print("网络错误，稍后重试")

t += 60

else:

download(result.text,name)

t+=60

print("下载结束，感谢使用")

R = recommend(a) #记录推荐

print("你可能喜欢")

for i in R:

print(i,end = ' ')