**中国执行信息公开网爬虫架构设计文档**

一、需求分析

1.扒取信息的网站：<https://zxgk.court.gov.cn/zhzxgk/>；

2.输入信息：输入查询人姓名，如：张晓东；

3.扒取信息：被执行人姓名/名称、身份证号码（不全）/组织机构代码、执行法院、立案时间、案号、执行标等。

二、开发环境

1.操作系统：Windows 10

2.开发语言：Python 3.6、JSON

3.开发工具：Visual Studio Code、Python 3.6 IDLE

4.依赖库或函数：

random, time, requests, urllib, BeautifulSoup, re, Image, path, pytesseract, webbrowser, json

threading

三、详细编程设计

1.获取查询人姓名：

name = input('请输入被执行人的姓名：')

2.定义url和headers，并作为参数使用requests向页面发出请求

url = "http://zxgk.court.gov.cn/zhzxgk/"

headers = {

'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3',

'Accept-Encoding': 'gzip, deflate',

'Accept-Language': 'en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7',

'Cache-Control': 'max-age=0',

'Connection': 'keep-alive',

'Cookie': 'JSESSIONID=1117B4BC0F5FBD9E0027A7C4542DCF05; \_gscu\_15322769=57994001imyfys20; SESSION=8a89a2ba-66fc-4700-916b-64cf745d6171; Hm\_lvt\_d59e2ad63d3a37c53453b996cb7f8d4e=1558834041,1558923029,1559009549,1559095425; \_gscbrs\_15322769=1; Hm\_lpvt\_d59e2ad63d3a37c53453b996cb7f8d4e=1559096204; \_gscs\_15322769=5909542610tvxi12|pv:4',

'Host': 'zxgk.court.gov.cn',

'Referer': 'http://zxgk.court.gov.cn/index.jsp',

'Upgrade-Insecure-Requests': '1',

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36'

}

web = requests.get(url, headers = headers).content.decode('utf-8')

3.获取页面信息并找到验证码的图片链接，并下载到本地

soup = BeautifulSoup(web, 'lxml')

imgSrc = soup.find('img', attrs = {'id' : 'captchaImg'})['src']

imgUrl = url + imgSrc + '/captcha.jpg'

urllib.request.urlretrieve(imgUrl, 'captcha.jpg')

4.依次使用fall、ganraoxian、two这三个函数对验证码图片进行去除噪点、去除干扰线（如果存在干扰线）、二值化（生成灰度图）的操作。

def fall (img1):#img：图片地址

white = (255,255,255,255)

black = (0,0,0,255)

img = Image.open(img1)

pixdata = img.load()

X = img.size[0]-1

Y = img.size[1]-1

def icolor(RGBA):

if RGBA == white:

return(1)

else:

return(0)

for y in range(Y):

for x in range(X):

if (x<1 or y<1):

pass

else:

if icolor(pixdata[x,y]) == 1:

pass

else:

if (

icolor(pixdata[x+1,y])+

icolor(pixdata[x,y+1])+

icolor(pixdata[x-1,y])+

icolor(pixdata[x,y-1])+

icolor(pixdata[x-1,y-1])+

icolor(pixdata[x+1,y-1])+

icolor(pixdata[x-1,y+1])+

icolor(pixdata[x+1,y+1])

)>6:

#如果一个黑色像素周围的8个像素中白色像素数量大于5个，则判断其为噪点，填充为白色

pixdata[x,y] = white

#填充白点

for y in range(Y):

for x in range(X):

if (x<1 or y<1):

pass

else:

if icolor(pixdata[x,y]) == 0:

pass

else:

if (

(icolor(pixdata[x+1,y]))+

(icolor(pixdata[x,y+1]))+

(icolor(pixdata[x-1,y]))+

(icolor(pixdata[x,y-1]))

)<2:

#如果一个白色像素上下左右4个像素中黑色像素的个数大于2个，则判定其为有效像素，填充为黑色。

pixdata[x,y] = black

#二次去除黑点

for y in range(Y):

for x in range(X):

if (x<1 or y<1):

pass

else:

if icolor(pixdata[x,y]) == 1:

pass

else:

if (

icolor(pixdata[x+1,y])+

icolor(pixdata[x,y+1])+

icolor(pixdata[x-1,y])+

icolor(pixdata[x,y-1])

)>2:

pixdata[x,y] = white

img.save('captcha1.png')

def ganraoxian(img2):#img：图片地址

img = Image.open(img2)

width = img.size[0]

heigth = img.size[1] #获取长宽

smap={}

slist=[]

keylist=[]

for i in range(0,width):

for j in range(0,heigth):

argb = img.getpixel((i,j))

r = argb[0]

g = argb[1]

b = argb[2]

sum = r + g + b #得到每一点的rgb

if sum not in smap.keys(): #如果没有该sum值的点 进行添加 并且给值为1

smap[sum]=1

else:

num=smap[sum]

smap[sum]=num+1 #如果有了这个值 在原基础上+1

slist=sorted(smap.items(),key=lambda x:x[1],reverse = False)

if (len(slist) > 4):

num1 = slist[len(slist) - 5][1]

num2 = slist[len(slist) - 4][1]

num3 = slist[len(slist) - 3][1]

num4 = slist[len(slist) - 2][1] #获取像素点最多的四个点

for key in smap:

if smap[key] == num1 or smap[key] == num2 or smap[key] == num3 or smap[key] == num4 :

#if num1 in smap or num2 in smap or num3 in smap or num4 in smap :

keylist.append(key) #找到对应颜色的点

for x in range(0,width):

for y in range(0,heigth):

argb = img.getpixel((x,y))

r = argb[0]

g = argb[1]

b = argb[2]

ssum = r + g + b

flag = True

for i in range(1,3): #px+1

if y + i < heigth and y - i > 0 and x - i > 0 and x + i < width:

upargb = img.getpixel((x,y-i))

endargb = img.getpixel((x,y+i))

rightupargb = img.getpixel((x+i,y+i))

leftupargb = img.getpixel((x-i,y+i))

leftdownargb = img.getpixel((x-i,y-i))

rightdownargb = img.getpixel((x+i,y-i))

r1 = upargb[0]

g1 = upargb[1]

b1 = upargb[2]

sum1 = r1 + g1 + b1

r2 = endargb[0]

g2 = endargb[1]

b2 = endargb[2]

sum2 = r2 + g2 + b2

r3 = rightupargb[0]

g3 = rightupargb[1]

b3 = rightupargb[2]

sum3 = r3 + g3 + b3

r4 = leftupargb[0]

g4 = leftupargb[1]

b4 = leftupargb[2]

sum4 = r4 + g4 + b4

r5 = leftdownargb[0]

g5 = leftdownargb[1]

b5 = leftdownargb[2]

sum5 = r5 + g5 + b5

r6 = rightdownargb[0]

g6 = rightdownargb[1]

b6 = rightdownargb[2]

sum6 = r6 + g6 + b6

if sum1 in keylist or sum2 in keylist or sum3 in keylist or sum4 in keylist or sum5 in keylist or sum6 in keylist:

flag = False

if (ssum not in keylist and flag):

img.putpixel((x,y),(255,255,255))

for x in range(0,width):

for y in range(0,heigth):

if img.getpixel((x,y))==(255,255,255,255):

continue

else:

img.putpixel((x,y),(0,0,0,255))

#curImg.setRGB(x, y, Color.white.getRGB())

img.save('captcha2.png')

def two(img3): #img：图片地址

i = 0

#img = Image.open('/home/yang/png/0.png') # 读入图片

img = Image.open(img3)

img = img.convert("RGBA")

while i < 4:#循环次数视情况进行调整

i = i+1

pixdata = img.load()

#一次二值化

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y][0] < 90:#使RGB值中R小于90的像素点变成纯黑

pixdata[x, y] = (0, 0, 0, 255)

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y][1] < 190:#使RGB值中G小于90的像素点变成纯黑

pixdata[x, y] = (0, 0, 0, 255)

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y][2] > 0:#使RGB值中B大于0的像素点变成纯白

pixdata[x, y] = (255, 255, 255, 255)

#理论上的二值化代码只有上面那些，RGB值的调整阈值需要针对不同验证码反复调整。同时实际中一组阈值往往没法做到完美，后面的部分是视实际情况添加的类似部分

#二次二值化（除去某些R、G、B值接近255的颜色）

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y][0] < 254:

pixdata[x, y] = (0, 0, 0, 255)

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y][1] < 254:

pixdata[x, y] = (0, 0, 0, 255)

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y][2] > 0:

pixdata[x, y] = (255, 255, 255, 255)

#三次二值化，怼掉纯黄色（实际使用中发现很多图片最后剩几个纯黄色的像素点）

for y in range(img.size[1]):

for x in range(img.size[0]):

if pixdata[x, y] ==(255,255,0,255):

pixdata[x, y] = (0, 0, 0, 255)

img.save('captcha3.png')

5.使用pytesseract.image\_to\_string函数识别验证码图片中的字符串

info = Image.open('captcha3.png')

imageInfo = pytesseract.image\_to\_string(info)

print(imageInfo)

6.由于上一步识别到的验证码中的字符串的正确率几乎为0，因此决定使用Github上的验证码开源识别框架cnn\_captcha： <https://github.com/nickliqian/cnn_captcha>

该项目针对字符型图片验证码，使用tensorflow实现卷积神经网络，进行验证码识别，项目封装了比较通用的校验、训练、验证、识别、API模块，

涉及的技术：

（1）图像处理：前处理（灰度化、二值化）、图像分割、裁剪（去边框）、图像滤波、降噪、去背景、颜色分离、旋转

（2）KNN、SVM、卷积神经网络

7.使用收费的验证码识别API

聚合数据验证码识别API：<https://www.juhe.cn/docs/api/id/60>

示例代码：

#!/usr/bin/python

# -\*- coding: utf-8 -\*-

import json, urllib

from urllib import urlencode

#----------------------------------

# 验证码识别调用示例代码 － 聚合数据

# 在线接口文档：http://www.juhe.cn/docs/60

#----------------------------------

def main():

#配置您申请的APPKey

appkey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#1.识别验证码

request1(appkey,"POST")

#2.查询验证码类型代码

request2(appkey,"GET")

#识别验证码

def request1(appkey, m="GET"):

url = "http://op.juhe.cn/vercode/index"

params = {

"key" : appkey, #您申请到的APPKEY

"codeType" : "", #验证码的类型，&lt;a href=&quot;http://www.juhe.cn/docs/api/id/60/aid/352&quot; target=&quot;\_blank&quot;&gt;查询&lt;/a&gt;

"image" : "", #图片文件

"dtype" : "", #返回的数据的格式，json或xml，默认为json

}

params = urlencode(params)

if m =="GET":

f = urllib.urlopen("%s?%s" % (url, params))

else:

f = urllib.urlopen(url, params)

content = f.read()

res = json.loads(content)

if res:

error\_code = res["error\_code"]

if error\_code == 0:

#成功请求

print res["result"]

else:

print "%s:%s" % (res["error\_code"],res["reason"])

else:

print "request api error"

#查询验证码类型代码

def request2(appkey, m="GET"):

url = "http://op.juhe.cn/vercode/codeType"

params = {

"key" : appkey, #您申请到的APPKEY

"dtype" : "", #返回的数据的格式，json或xml，默认为json

}

params = urlencode(params)

if m =="GET":

f = urllib.urlopen("%s?%s" % (url, params))

else:

f = urllib.urlopen(url, params)

content = f.read()

res = json.loads(content)

if res:

error\_code = res["error\_code"]

if error\_code == 0:

#成功请求

print res["result"]

else:

print "%s:%s" % (res["error\_code"],res["reason"])

else:

print "request api error"

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

main()

8.分页抓取信息并保存成json文件，每一页包含10个人的信息，再生成每个人的信息页面的请求，将信息页面的信息都扒取下来并保存成json文件。

for page in range(1, 113):

data = {

'pName': name,

'pCardNum' : '',

'selectCourtId': '0',

'pCode': pCode,

'captchaId': captchaId,

'searchCourtName': '全国法院（包含地方各级法院）',

'selectCourtArrange': '1',

'currentPage': str(page)

}

web2 = requests.post(url + 'searchZhcx.do', data = data, headers = headers).content.decode('utf-8')

#print(type(web2))

web2 = web2.encode()

with open('json\jsonArray' + str(page) + '.json', 'wb') as jsonFile:

jsonFile.write(web2)

jsonFile.close()

with open('json\jsonArray' + str(page) + '.json', "r", encoding='utf-8') as jsonFile:

loadDict = json.load(jsonFile)

jsonData = json.loads(web2)

#print(type(jsonData))

jsonPage = []

for case in range(0, len(loadDict[0]['result'])):

#for case in range(0, len(jsonData[0]['result'])):

#caseCodeNewDel = jsonData[0]['result'][case]['caseCode']

caseCodeNewDel = loadDict[0]['result'][case]['caseCode']

#print('caseCodeNewDel: ' + caseCodeNewDel)

url2 = 'http://zxgk.court.gov.cn/zhzxgk/detailZhcx.do?pnameNewDel=' + name + '&cardNumNewDel=&j\_captchaNewDel=' + pCode + '&caseCodeNewDel=' + caseCodeNewDel + '&captchaIdNewDel=' + captchaId

#webbrowser.open(url2)

web3 = requests.get(url2, headers = headers).content.decode('utf-8')

soup = BeautifulSoup(web3, 'lxml')

trs = soup.find\_all('td')

#index = -1

dictD = {}

for i in range(0, len(trs)):

if i % 2 == 0:

item = trs[i].strong.string

#print(trs[i].string + '\t', end='')

else:

val = trs[i].string

#print(trs[i].string)

dictD.update({item: val})

jsonPage.append(dictD)

print('Page: ' + str(page))

with open('json\jsonPage' + str(page) + '.json', 'wb') as jsonFile:

jsonFile.write(str(jsonPage).encode())

jsonFile.close()

9.使用多线程的机制，将扒取个人页面信息的那部分代码封装成一个函数，并把函数实例化为一个线程对象，加入线程池。

threads = []

for page in range(1, 113):

t = Thread(target=getPage, args=(name, pCode, captchaId, page, url, headers))

t.start()

threads.append(t)

for i in threads:

i.join()