**Module 8: Computer Vision OpenCV and Visualisation using Bokeh**

Case Study

1. Write a program to fetch hyperlinks from any website which user enters.

**Solution:**  
from bs4 import BeautifulSoup

import requests

url = input("Enter the URL to scrap: ")

source = requests.get(url).text

soup = BeautifulSoup(source, 'html5lib')

for link in soup.find\_all('a', href=True):

print(link['href'])

2. Write a program to download all the videos from youtube.com for django from the hyperlink given below

https://www.youtube.com/playlist?list=PLxxA5z-8B2xk4szCgFmgonNcCboyNneMD

**Solution:**

import bs4 as bs

import sys

import urllib.request

from PyQt5.QtWebEngineWidgets import QWebEnginePage

from PyQt5.QtWidgets import QApplication

from PyQt5.QtCore import QUrl

import pytube

class Page(QWebEnginePage):

def \_\_init\_\_(self, url):

self.app = QApplication(sys.argv)

QWebEnginePage.\_\_init\_\_(self)

self.html = ''

self.loadFinished.connect(self.\_on\_load\_finished)

self.load(QUrl(url))

self.app.exec\_()

def \_on\_load\_finished(self):

self.html = self.toHtml(self.Callable)

print('Load finished')

def Callable(self, html\_str):

self.html = html\_str

self.app.quit()

links = []

def exact\_link(link):

vid\_id = link.split('=')

# print(vid\_id)

str = ""

for i in vid\_id[0:2]:

str += i + "="

str\_new = str[0:len(str) - 1]

index = str\_new.find("&")

new\_link = "https://www.youtube.com" + str\_new[0:index]

return new\_link

url = "https://www.youtube.com/playlist?list=PLxxA5z-8B2xk4szCgFmgonNcCboyNneMD"

page = Page(url)

count = 0

soup = bs.BeautifulSoup(page.html, 'html.parser')

for link in soup.find\_all('a', id='thumbnail', href=True):

# not using first link because it is

# playlist link not particular video link

if count == 0:

count += 1

continue

else:

vid\_src = link['href']

# print(vid\_src)

# keeping the format of link to be

# given to pytube otherwise in some cases

new\_link = exact\_link(vid\_src)

# error might occur due to this

# print(new\_link)

# appending the link to the links array

links.append(new\_link)

for link in links:

yt = pytube.YouTube(link)

# Downloaded video will be the best quality video

stream = yt.streams.filter(progressive=True,

file\_extension='mp4').order\_by(

'resolution').desc().first()

try:

stream.download()

# printing the links downloaded

print("Downloaded: ", link)

except:

print('Some error in downloading: ', link)

3. Create a csv file with name and hyperlink after fetching it from the web page

http://bioguide.congress.gov/biosearch/biosearch.asp

Select any of the option from it as in the below screenshot and click on search

Later download the page source, save it in html file and then perform scraping.

**Solution:**

from \_\_future\_\_ import division, unicode\_literals

import codecs

from bs4 import BeautifulSoup

f=codecs.open("congress.gov\_biosearch\_biosearch1.html", 'r', 'utf-8')

document= BeautifulSoup(f.read()).get\_text()

document1 = BeautifulSoup(document,"html5lib")

import csv

csv\_file = open('biosearch.csv', 'w')

csv\_writer = csv.writer(csv\_file)

csv\_writer.writerow(['Name', 'BiographyLink'])

for name in document1.find\_all('a', href=True):

Name = name.text

BiographyLink = name['href']

csv\_writer.writerow([Name, BiographyLink])

csv\_file.close()

4. from the question above, fetch only the hyperlinks

**Solution:**

from bs4 import BeautifulSoup

filename = r"C:/Users/BNP/Documents/PythonLearning/congress.gov\_biosearch\_biosearch1.html"

pageopen = open(filename,'r').read()

soup = BeautifulSoup(pageopen, "html.parser")

for link in soup.find\_all('a', href=True):

print(link['href'])

5. Write Perform the web scraping on the following page

<html> <head> <title> Page title </title> </head> <body> <p id=**"firstpara"** align=**"center"**> This **is** paragraph <b> one </b> </p> <p id=**"secondpara"** align=**"blah"**> This **is** paragraph <b> two </b> </p> </body> </html>

i) Read the page using BeautifulSoup and show it in well formatted indented manner.

**Solution:**

from bs4 import BeautifulSoup

pageread = open("Page.html",'r').read()

soup = BeautifulSoup(pageread, "html.parser")

print(soup.prettify())

ii) Print the b tag from the page

**Solution:**

soup.find('b')

iii) Print all the tags that starts from b

**Solution:**

import re

for tag in soup.find\_all(re.compile('^b')):

print(tag.name)

iv) Print text from the tags having 'title' and 'p'. by using lists

**Solution:**

for tag1 in soup.find\_all(["title", "p"]):

print(tag1.text)

v) Print text from the tags having 'title' and 'p'. by using dictionaries

**Solution:**

for tag2 in soup.find\_all({"title", "p"}):

print(tag2.text)

vi) Print all the tag names present in the page

**Solution:**

for tag3 in soup.find\_all():

print(tag3.name)

vii) Print the complete tag that have two, and only two, attributes

**Solution:**

soup.find\_all(attrs={'id': ['firstpara', 'secondpara'], 'align':['center', 'blah']})

viii) Print the tags that have one-character names and no attributes

**Solution:**

for char in soup.find\_all('b', attrs=None):

print(char)

ix) Print all the tags which have a value of "center" for their "align" attribute

**Solution:**

results = soup.findAll("p", {"align":"center"})

for result in results:

print(result)

x) From the xml content

'<person name="Bob"><parent rel="mother" name="Alice">'

Print the attributes having "name" as "Alice"

**Solution:**

soupXml.select('parent', attrs={'name':'Alice'})