UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SEMESTER 1 SESSION 2022/2023

SMJE4383- ADVANCED PROGRAMMING

ASSIGNMENT 2

SCREEN SCRAPING & OCR TEXT RECOGNITION USING PYTHON SCRIPT

NAME	MATRIC NO.
AFIQ FIRDAUS BIN MUHAMAD ROSDI	A19MJ0007
WAN DZAFIRUL HAKIMI BIN WAN MAZRI	A19MJ0134
Section : 01 Lecturer's name: Assoc. Prof. Ir. Dr. Zool hilmi Bin Ismail Github link: https://github.com/WanDz03/SMJE4383/tree/main/Assignment_2	

Background study

Screen scraping is a technique used to extract data from a website or a computer program's user interface. This can be achieved by parsing the HTML code of a web page or by using software that can recognise the user interface elements of an application and extract the data from them.OCR (Optical Character Recognition) is a technology used to identify and convert written or typed text characters into machine-encoded text. This is useful for converting scanned documents, PDFs, or images into editable text. OCR can be used to extract text from an image and then feed it into another process, such as screen scraping.

The main objective of this assignment is to extract text from an image and convert it into a string. The main output is the string in the terminal. The programming code will execute an end-to-end process of screen scraping and OCR test recognition using Python Script. The side feature of this assignment is converting the one-page PDF file into a PNG file. This topic is selected because it is part of the academia-industrial collaboration MJIIT and aims to solve the industrial-based problem.

Problem Statement

A company want to automate checking machine information and the vision of the computer. Every computer has different machine information depending on the updated vision. It became a hassle to find the machine information manually. By having this code, the programmer can check the machine's information in the terminal by running the code in the terminal. A feature like this is really helpful for the programmer to check their machine information in the terminal.

Objective

The objective of the assignment is to develop a programming code for the extraction of images and get the required information by using Screen Scraping and OCR Text Recognition in Python.

Methodology

The programming code workflow is shown below:

1. The ‘platform’ library is used to extract the machine information and the ‘FPDF” library is used to convert the machine information into a PDF file.
2. The ‘pdf2image’ library is used to convert the PDF file into a PNG file.
3. OCR Text Recognition: Using an OCR library like ‘pytesscract’ to extract text from the PNG file.
4. The extraction from the image is displayed in the terminal.

Result and Discussion

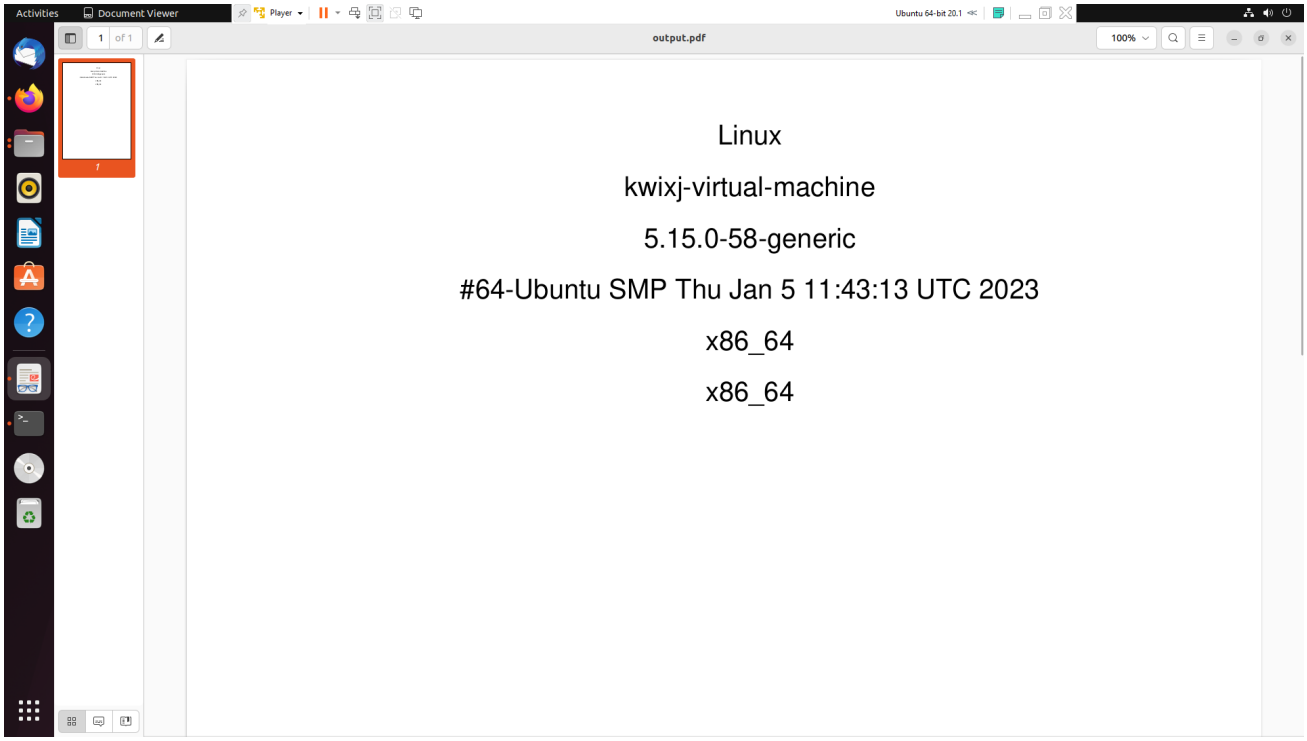

Figure 1 The machine information in the PDF file

```
Installing collected packages: fpdf
Successfully installed fpdf-1.7.2
kwixj@kwixj-virtual-machine:~/Desktop/KWIXJ/Assignemnt2$ python3 manchine.py
uname_result(system='Linux', node='kwixj-virtual-machine', release='5.15.0-58-generic', version='#64-Ubuntu SMP Thu Jan 5 11:43:13 UTC 2023', machine='x86_64')
kwixj@kwixj-virtual-machine:~/Desktop/KWIXJ/Assignemnt2$ python3 manchine.py
uname_result(system='Linux', node='kwixj-virtual-machine', release='5.15.0-58-generic', version='#64-Ubuntu SMP Thu Jan 5 11:43:13 UTC 2023', machine='x86_64')
kwixj@kwixj-virtual-machine:~/Desktop/KWIXJ/Assignemnt2$ 0
```

The platform library gets the machine's information and stores the information in a PDF file. However, the ‘pytesseract’ library did not have the capability to read the test in the PDF file. A process to convert the PDF file into a PNG file is required. This is useful to use this code to check the machine information every time needed. Some libraries can be used for specific visions. This code also can read other text in the PDF file after converting it into a PNG file.

Code

22 lines (15 sloc) | 334 Bytes

```
1   #import os
2   #os.system("settings")
3   #import subprocess
4   #subprocess.Popen([file],shell=True)
5   from fpdf import FPDF
6   import platform
7
8   print(platform.uname())
9   pdf = FPDF()
10
11  pdf.add_page()
12
13  pdf.set_font("Arial", size = 15)
14
15  f = platform.uname()
16
17  for x in f:
18          pdf.cell(200, 10, txt=x, ln=1, align='C')
19
20  pdf.output("output.pdf")
21
22  import test
```

```
import cv2
import pytesseract
from pdf2image import convert_from_path

pages = convert_from_path('output.pdf', 500)
for page in pages:
        page.save('out.png', 'PNG')

def ocr_core(img):
        text = pytesseract.image_to_string(img)
        return text

img = cv2.imread('out.png')

print(ocr_core(img))
```