



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**MALAYSIA-JAPAN INTERNATIONAL INSTITUTE OF  
TECHNOLOGY  
(DEPARTMENT OF ELECTRONIC SYSTEMS ENGINEERING)**

**ADVANCE PROGRAMMING  
(SMJE 4383)**

**Assignment 2**

<b>NAME</b>	:	<b>Cheng Wei Ping Chong Ming Chuen</b>
<b>MATRIX NO</b>	:	<b>A19MJ0021 A19MJ0024</b>
<b>YEAR/PROGRAM</b>	:	<b>4 SMJE</b>
<b>SECTION</b>	:	<b>01</b>
<b>LECTURER'S NAME</b>	:	<b>DR. ZOOL HILMI ISMAIL</b>
<b>DATE</b>	:	<b>7/2/2022</b>
<b>Github Link</b>	:	<b><a href="https://github.com/chuench/AdvancedProgramming-Assignment/tree/Assignment_2">https://github.com/chuench/AdvancedProgramming-Assignment/tree/Assignment_2</a></b>

## **Table of Contents**

<b>Topic</b>	<b>Page</b>
<b><u>CHAPTER 1 INTRODUCTION</u></b>	
1.1 Introduction	2
1.2 Project Framework and Interface	2
1.3 Working Principle	3
1.4 Existing System	4
1.5 Problem Statements	4
1.6 Objectives	4
<b><u>CHAPTER 2 METHODOLOGY</u></b>	
2.1 Software Required	5
2.2 Modules Used	6
2.3 Procedures Explanation	8
<b><u>CHAPTER 3 RESULTS AND DISCUSSION</u></b>	
3.1 Results	10
3.2 Discussion	11
<b><u>CHAPTER 4 CONCLUSION</u></b>	
4.1 Conclusion	12
References	13

## **CHAPTER 1: INTRODUCTION**

### **1.1 Introduction**

Screen scraping, also known as web scraping, is the process of automatically extracting data from a website or other source of information that is displayed on a screen. Screen scraping involves making HTTP requests to a website, parsing the HTML or other data format, and extracting the relevant information. The extracted data can then be cleaned, processed, and stored for use in other applications [1].

Screen scraping is commonly used in a variety of applications, including data mining, data analysis, and data integration. By automating the process of extracting data from websites, screen scraping can greatly improve efficiency and reduce the risk of human error. It can also make previously inaccessible information usable and searchable [2].

OCR (Optical Character Recognition) text recognition is a technology that enables the conversion of scanned images and handwritten text into editable and searchable digital text [3]. It works by analyzing the patterns and shapes of the characters in an image and comparing them to a database of known characters [3]. The technology can recognize a wide range of text styles and languages, including both printed and handwritten text.

OCR text recognition is commonly used in a variety of applications, such as document scanning, data entry, and digital archiving. By automating the process of text recognition, OCR technology can greatly improve efficiency and reduce the risk of human error [2]. It can also make previously inaccessible information, such as handwritten notes or scanned documents, usable and searchable.

### **1.2 Project Framework and Interface**

There are few software frameworks and interfaces included in this project. To build end-to-end process for Screen Scraping OCR Text Recognition, the implementations include Visual Studio Code, pytesseract, pyautogui and PIL.

Visual Studio Code is an efficient code editor designed for tasks such as debugging, executing tasks, and version management. Its purpose is to offer the essential tools for a developer's rapid coding, building, and debugging cycle, while leaving more advanced

workflows to more comprehensive Integrated Development Environments like Visual Studio IDE [4].

Python-tesseract (pytesseract) is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images. pytesseract is a tool that acts as a bridge to connect Google's Tesseract-OCR Engine. It can also be used as a standalone program to run tesseract and extract text from various image formats, such as jpeg, png, gif, bmp, tiff, and others, which are supported by the Pillow and Leptonica image libraries. When utilized as a script, Python-tesseract will display the recognized text on the screen instead of saving it in a file [5].

PyAutoGUI (pyautogui) is essentially a Python package that works across Windows, MacOS X and Linux which provides the ability to simulate mouse cursor moves and clicks as well as keyboard button presses [6]. pyautogui is a cross-platform GUI automation Python module for human beings and usually used to programmatically control the mouse and keyboard [7].

The Python Imaging Library (PIL) gives the Python interpreter the ability to edit images. It has a module called Image, which offers a class of the same name that is utilized to represent a PIL image. The module also offers a series of factory functions, including functions for loading images from files and generating new images. This process is performed that the function identifies the file but keeps it open and the image data is not actually read from the file until it is needed for processing [8].

### **1.3 Working Principle**

This project is started by preparing Visual Studio Code, downloading all the required materials such as pytesseract, pyautogui and PIL before started the procedures. Steps and codes that are available in the Chapter 2. The screenshot will be done when the code run. Then, the text in the screenshot will be analysis and write into text file.

## **1.4 Existing System**

In this Advanced Programming subject, students are required to set up their Ubuntu operating system, however, in this project, the Ubuntu operating system is not suitable, so Visual Studio Code is selected to use in this project.

## **1.5 Problem Statements**

An easy and quick way is provided by Python to build an end-to-end process for Screen Scraping OCR Text Recognition. Students are requested to study and learn the way to design a fully end-to-end process for Screen Scraping OCR Text Recognition by using the Python programming language, Visual Studio Code and other framework or interface such as pytesseract, pyautogui and PIL.

## **1.6 Objectives**

To develop an end-to-end process for Screen Scraping OCR Text Recognition using Python Script, the objectives of this project are as below:

- a) To capture a region screenshot by using Python module.
- b) To analysis and extract the text from the screenshot.
- c) To save the extracted text and write into a text file.

## **CHAPTER 2: METHODOLOGY**

### **2.1 Software Required**

#### **1. Visual Studio Code**

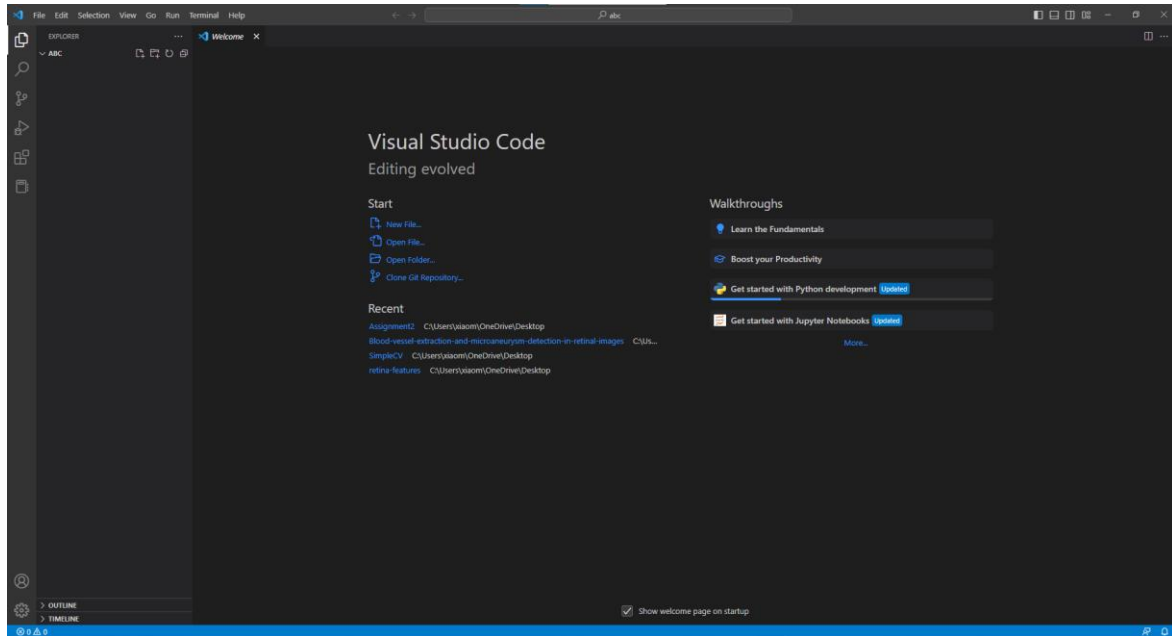


Figure 2.3: GUI for Visual Studio Code

Visual Studio Code (VSCode) is a free, open-source, cross-platform code editor developed by Microsoft. It is designed to be lightweight and fast, while still providing a rich set of features for developers. Some of the features of VSCode include: syntax highlighting and code formatting for many programming languages, intelligence code completion, code debugging and testing tools, integrated version control support (Git), code navigation and search, customizable user interface and color themes, built-in terminal and extensibility through the use of plugins and extensions. We use the visual studio code in Window 10 operating system due to some screenshot issue happen in Ubuntu software.

## 2. Python 3.10



Figure 2.2: Logo of Python 3.10 software

Python 3.10 is a version of the Python programming language. Python is a high-level, interpreted, and general-purpose programming language. It is known for its readability, ease of use, and wide variety of libraries and modules for various tasks, such as web development, scientific computing, and data analysis. Python 3.10 was released in October 2021 and includes features such as string methods for removing prefixes and suffixes, and improvements in type annotations, among others. Python is an open-source language, which means that it is free to use and the source code is publicly available. We used python3 as the programming language in this project.

### 2.2 Modules Used

#### 1. pyautogui (import pyautogui)

The pyautogui module is a Python module for automating GUI interactions, such as clicking and typing, on a computer. It can be used to automate tasks such as filling out forms, logging into websites, or clicking buttons on a graphical user interface (GUI). The module works by taking screenshots of the GUI and searching for specific images or text to identify elements on the screen, which it can then interact with. We use pyautogui to perform screen capturing to capture screenshots of the screen and save them as image files.

#### 2. pytesseract (import pytesseract)

The pytesseract module is a Python wrapper for the Tesseract OCR (Optical Character Recognition) engine. The function is to extract text from images and convert it into a machine-readable format. Tesseract is an open-source OCR engine that is capable of

recognizing text in more than 60 languages. The pytesseract module provides a simple and convenient interface for using Tesseract from within a Python program. Key features of pytesseract are image to text conversion where pytesseract can extract text from images and convert it into a machine-readable format. The pytesseract module is commonly used for tasks such as extracting text from scanned documents, recognizing text in images, and automating data entry. We use the pytesseract module to perform image to text recognition & conversion.

### 3. PIL (from PIL import Image)

PIL (Python Imaging Library) is a library in Python that allows you to work with images. PIL can perform a wide range of image processing tasks, such as opening and saving images in different formats, cropping and resizing images, applying filters and transformations, and creating image thumbnails. PIL provides a comprehensive set of image manipulation functions and supports many image formats, including JPEG, PNG, BMP, and GIF. The library is easy to use, and its high-level API makes it possible to perform complex image processing tasks with just a few lines of code. The key features of PIL are:

- Image manipulation: PIL provides a wide range of functions for manipulating images, including crop, resize, rotate, and flip.
- Image enhancement: PIL provides functions for enhancing images, such as colour correction, contrast adjustment, and noise reduction.
- Image file handling: PIL makes it easy to open and save images in different file formats.



## 2.3 Procedure Explanation

Below shows the full coding in the project. The detail of the code will we upload in the GitHub and also will be attached in the appendix below. The procedures are summarized in below:

```
ass2.py  X
ass2.py > ...
1  import pyautogui
2  import pytesseract
3  from PIL import Image
4
5  # Specify the region for screenshot width with 1000pixels and 700pixels height
6  x, y, width, height = 0,200,1000,700
7
8  # Region Screenshot capturing using pyautogui
9  def screen_capture(image_filename):
10     screenshot = pyautogui.screenshot(region=(x, y, width, height))
11     # Save the screenshot as png
12     screenshot.save('RegionScreenshot.png')
13
14     # OCR Text Recognition using pytesseract
15     def ocr_text_recognition(image_filename):
16         text = pytesseract.image_to_string(Image.open(image_filename))
17         return text
18
19     # Text recognition and text file generate based on Screenshot image
20     def screen_scraping_ocr_generation(screenshot_input, text_output):
21         screen_capture(screenshot_input)
22         text = ocr_text_recognition(screenshot_input)
23         with open(text_output, 'w') as file:
24             file.write(text)
25
26     # Main function here
27     if __name__ == "__main__":
28         screen_scraping_ocr_generation("RegionScreenshot.png", "Output.txt")
```

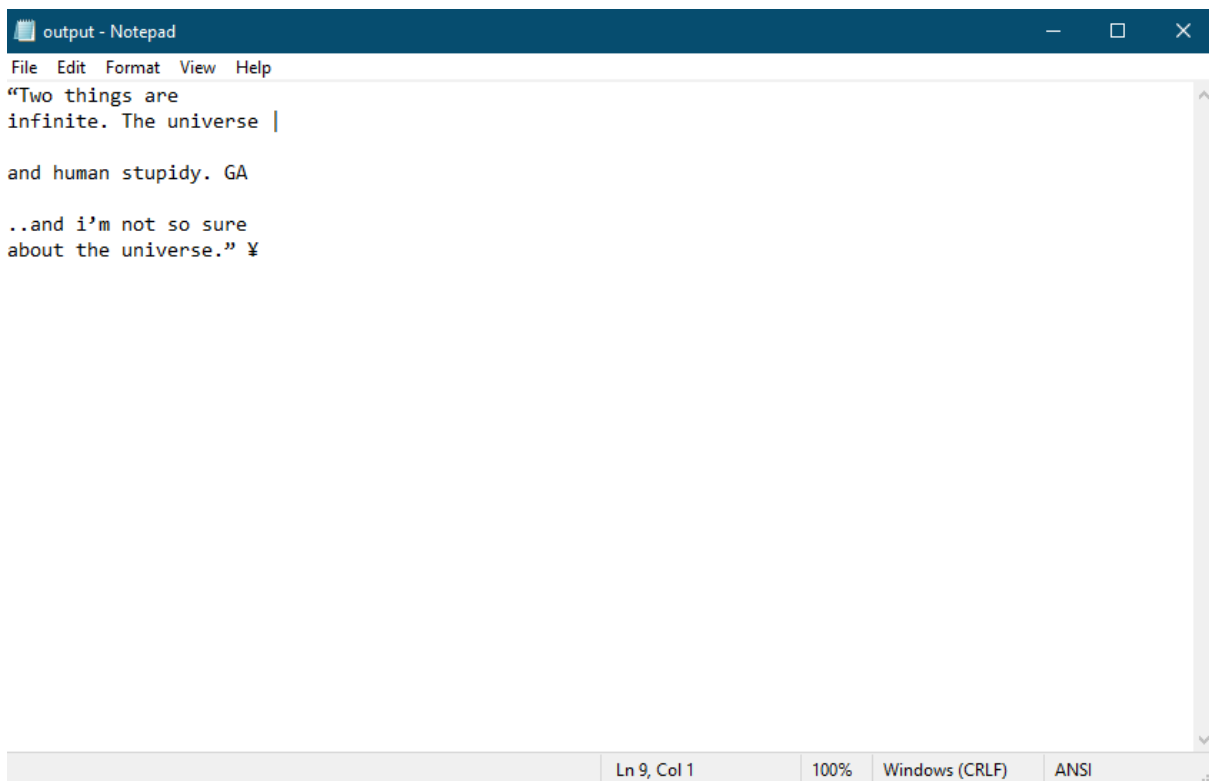
Figure 2.3: Full code of the project.

### Steps for implementation

1. Install Visual Studio Code in the Windows 10 OS.
2. Install the required module (pytesseract, pyautogui, PIL)
3. Put the pytesseract installation file in the same directory of the python script file.
4. Start implementation of the python script file with the file name “ass2.py”.
5. The code imports the required libraries pyautogui, pytesseract, and Image from PIL library.
6. The region for screenshot is specified with width of 1000 pixels and height of 700 pixels, x and y axis of 0, 200 respectively.
7. screen\_capture function captures a screenshot of the specified region using pyautogui and saves the screenshot as a png file.
8. ocr\_text\_recognition function uses pytesseract to perform OCR (Optical Character Recognition) on the image, converts it to text and returns the text.
9. screen\_scraping\_ocr\_generation function performs the text recognition and generation of a text file based on the screenshot image by calling screen\_capture and ocr\_text\_recognition.
10. The main function calls screen\_scraping\_ocr\_generation with “RegionScreenshot.png” as screenshot input and “Output.txt” as the text output.



Below shows the output of the txt file for the image to text recognition.



```
output - Notepad
File Edit Format View Help
"Two things are
infinite. The universe |
and human stupidity. GA
..and i'm not so sure
about the universe."  
Ln 9, Col 1 100% Windows (CRLF) ANSI
```

Figure 3.3: Output of the txt file.

### 3.2 Discussion

The python script perform screen scraping and optical character recognition (OCR) generation. It uses the PyAutoGUI module to capture a region screenshot of the screen with specified dimensions (width of 1000 pixels and height of 700 pixels), and saves it as a PNG image. It then uses the PyTesseract module to perform OCR on the screenshot image and extract the text. Finally, the text is saved in a text file using the built-in “open” function in Python.

The script is organized into several functions, which allows for modularity and easy maintenance. The “screen\_capture” function captures the screenshot of the specified region, while the “ocr\_text\_recognition” function performs the OCR on the screenshot image. The “screen\_scraping\_ocr\_generation” function ties everything together by calling the “screen\_capture” function to obtain the screenshot and then calling the “ocr\_text\_recognition” function to extract the text. The main function at the end of the script calls the “screen\_scraping\_ocr\_generation” function to generate the final output. The

output of the code will be a text file containing the extracted text from the region screenshot. The accuracy of the OCR results will depend on the quality of the screenshot image and the training data used by the OCR engine.

For the output discussion, the “PyTesseract” module will need to install in the same directory with the python script file in order for the module to run. Region Screenshot is selected instead of full-screenshot due to full screen shot having a lot of into to capture and conversion from image to text where it make the output become lengthy and hardly to recognized what it required. Based on Figure 3.3, the output of the txt file generate by the extracted text from the region screenshot are not accurate as it contain some extra symbol where it shouldn’t exist. This may due some limitation of the “PyTesseract” module where it makes some mistake due to low-quality images or text in non-standard fonts.

## **CHAPTER 4 CONCLUSION**

### **4.1 Conclusion**

In this assignment, all the objectives are achieved. The region screenshot has successfully captured by using Python module with Visual Studio Code. The text has successfully analysis and extract from the screenshot. The text also has been saved and wrote into a text file.

## References

- [1] What Is Screen Scraping. (2022). Retrieved from: <https://dzone.com/articles/screen-scraping>
- [2] RPA Technical Insights, Part 10: Why Screen Scraping is Essential to the RPA Toolkit. (2016). Retrieved from: <https://www.symphonyhq.com/rpa-technical-insights-part-10/>
- [3] OCR Screen Scraping. (2016). Retrieved from: [https://ui.vision/rpa/x/desktop-automation/screen-scraping#:~:text=Text%20Recognition%20\(also%20called%20Screen%20Scraping%2C%20OCR\)&text=Optical%20Character%20Recognition%20\(OCR\)%20works,code%20and%20document%20browser%20object.](https://ui.vision/rpa/x/desktop-automation/screen-scraping#:~:text=Text%20Recognition%20(also%20called%20Screen%20Scraping%2C%20OCR)&text=Optical%20Character%20Recognition%20(OCR)%20works,code%20and%20document%20browser%20object.)
- [4] What is the difference between Visual Studio Code and Visual Studio IDE?. (2015). Retrieved from: <https://code.visualstudio.com/docs/supporting/FAQ#:~:text=Visual%20Studio%20Code%20is%20a,such%20as%20Visual%20Studio%20IDE.>
- [5] pytesseract 0.3.10. Retrieved from: <https://pypi.org/project/pytesseract/>
- [6] Automate UI Testing with PyAutoGUI in Python. (2019). Retrieved from: <https://towardsdatascience.com/automate-ui-testing-with-pyautogui-in-python-4a3762121973#:~:text=PyAutoGUI%20is%20essentially%20a%20Python,well%20as%20keyboard%20button%20presses.>
- [7] PyAutoGUI 0.9.53. Retrieved from: <https://pypi.org/project/PyAutoGUI/>
- [8] Python PIL | Image.open() method. (2019). Retrieved from: <https://www.geeksforgeeks.org/python-pil-image-open-method/>