

**SEMESTER 1 2022/2023**

SMJE 4383

ADVANCED PROGRAMMING

**ASSIGNMENT 2**

Screen Scraping & OCR Text Recognition

|  |  |  |
| --- | --- | --- |
| **NAME** | **Matric No.** | **Signature** |
| 1. LI DAHUA | A19MJ5039 | li |
| 1. TEW JIA JIAN | A19MJ0130 | tew |
| **Name of Lecturer** | | **SECTION** |
| ASSOC PROF IR DR ZOOL HILMI ISMAIL | | 01 |

**1.0 INTRODUCTION**

* 1. **Problem Background**

Screen scraping is a technology that involves automatically extracting data from websites, applications, or other digital sources. It is done by parsing the HTML, XML, or other structured data formats of these sources to extract the relevant information.

OCR stands for Optical Character Recognition, which is a technology that enables the recognition of text from images and scanned documents. It works by analyzing an image and converting the characters within it into editable and searchable text. This process is useful in many applications, such as document digitization, data extraction, and information retrieval. OCR software typically uses pattern recognition, machine learning, and other advanced technologies to accurately recognize characters, even in images that contain noise, distortion, or other anomalies. OCR technology has made it easier to access and process information stored in image or scanned format, making it a valuable tool for organizations of all sizes.

These technologies have numerous applications in areas such as data mining, web scraping, document digitization, and many more. By using these tools, organizations can quickly and easily extract relevant information from large amounts of data sources, saving time and effort.

**1.2. Problem Statement**

Student need to carry out a programming task that can execute end-to-end process for Screen Scraping & OCR Text Recognition using Python Script. This topic is selected because it is part of the academia-industrial collaboration at MJIIT.

**1.3 Project Objective**

* To extract the information of text recognition and display the data to user by using Tesseract OCR engine
* To automate text recognition without requiring the user to perform several actions

**2.0 METHODOLOGY**

**2.1 Usage of modern tool**

Tools that involve in this project are PyCharm, Python 3. Library that used in this project are Tesseract and OpenCV.

Tesseract OCR (Optical Character Recognition) engine is an open-source OCR software developed by Google. It is one of the most popular OCR engines used for recognizing text from images and scanned documents.

Tesseract OCR uses advanced algorithms and machine learning techniques to accurately recognize characters, even in images that contain noise, distortion, or other anomalies. The software is highly customizable, allowing users to train the OCR engine to recognize specific fonts, character sets, and languages.

**2.1 Procedure**

1. PyCharm was install in the computer.

2. Tesseract OCR software was install from github : <https://github.com/UB-Mannheim/tesseract/wiki>.

3. Tesseract and OpenCV library package was downloaded by using command line:

“pip install pytesseract”

“pip install opencv-python”

1. A sample input image was prepared.
2. The code was developed and recorded.
3. The outcome of the code was discussed.

**3.0 RESULT AND DISCUSSION**

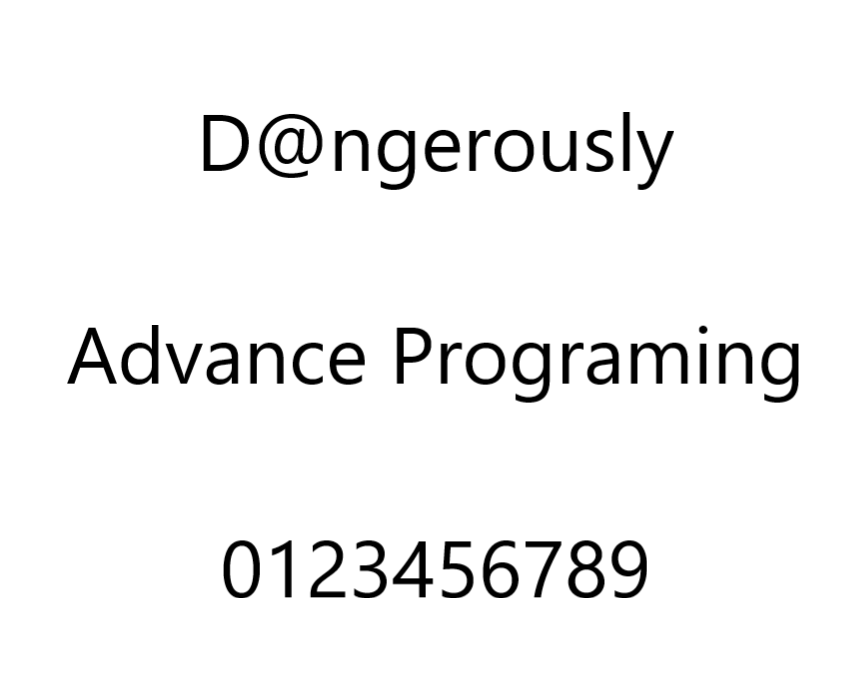


Figure 1: Sample input image

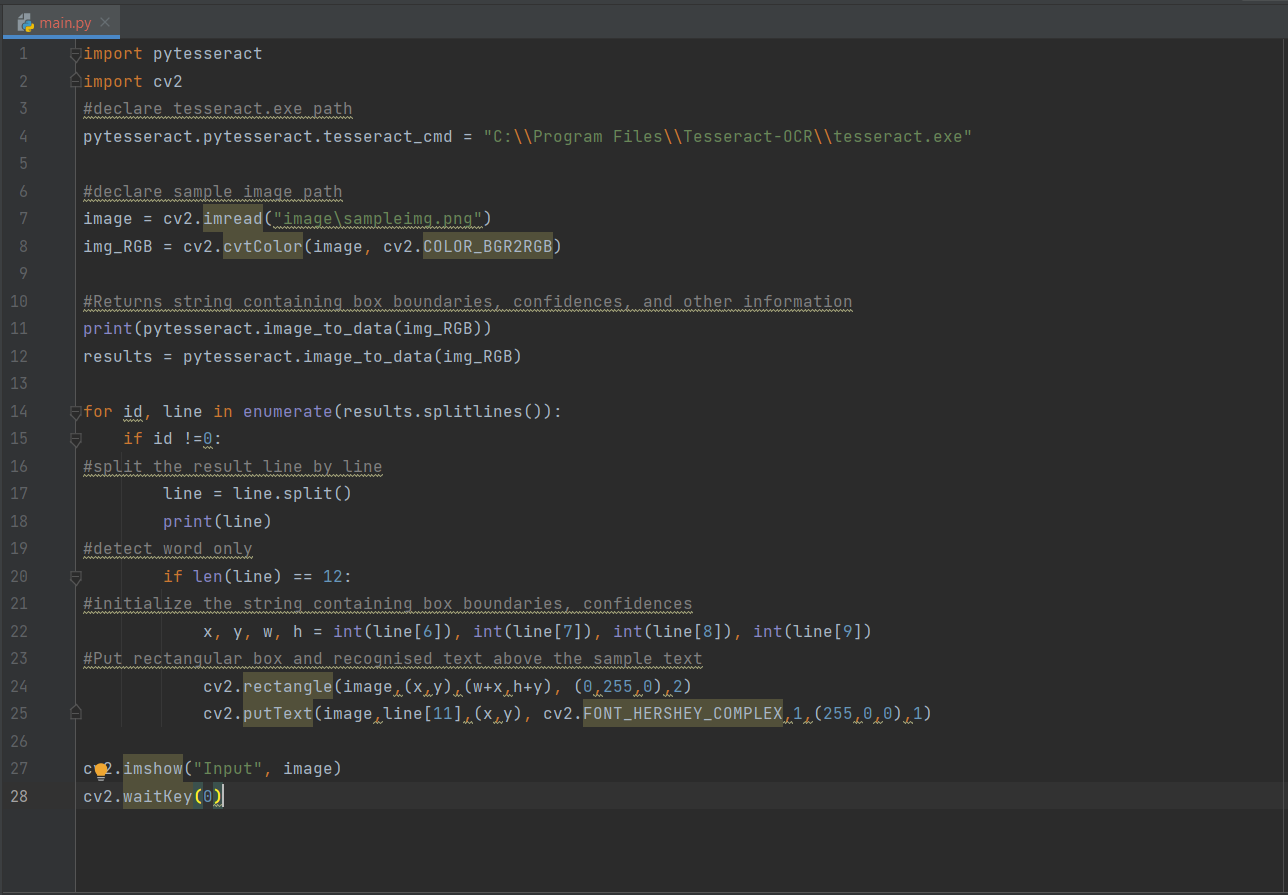


Figure 2: Python Script in PyCharm

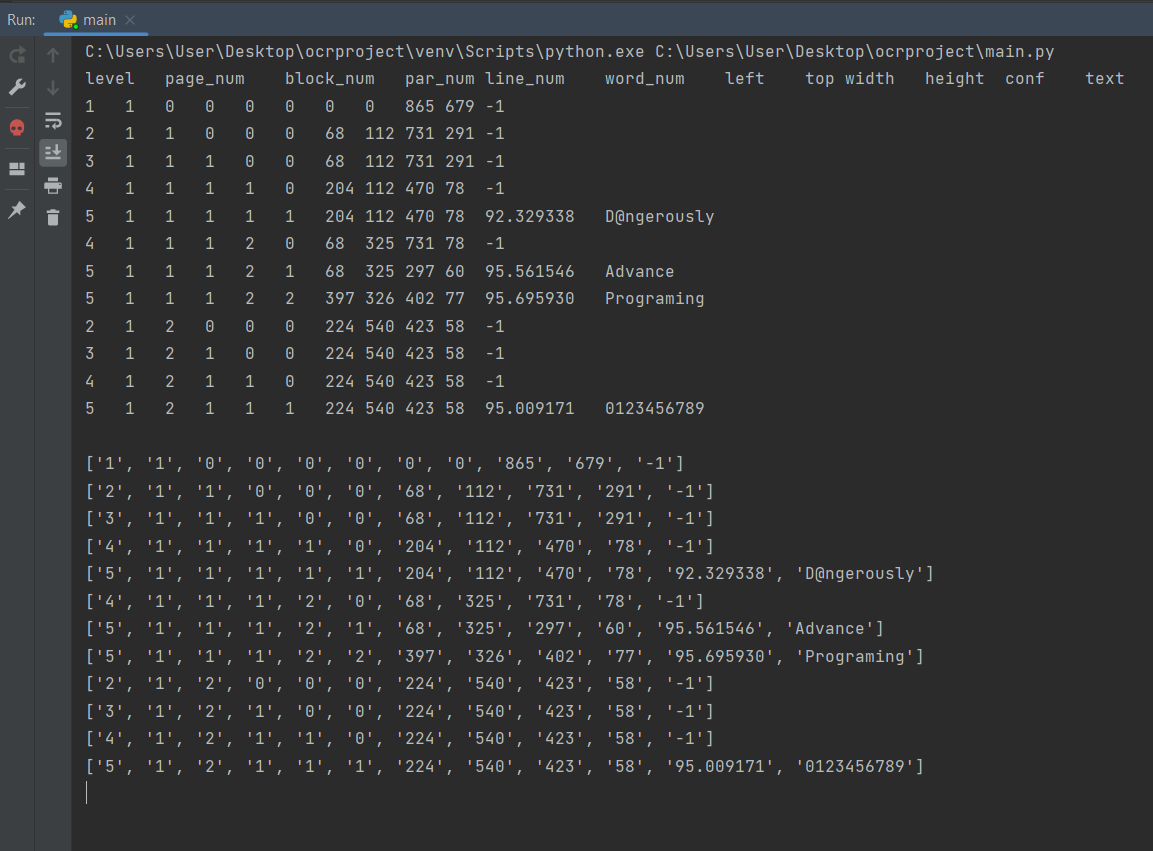


Figure 3: Result from terminal after ran the code

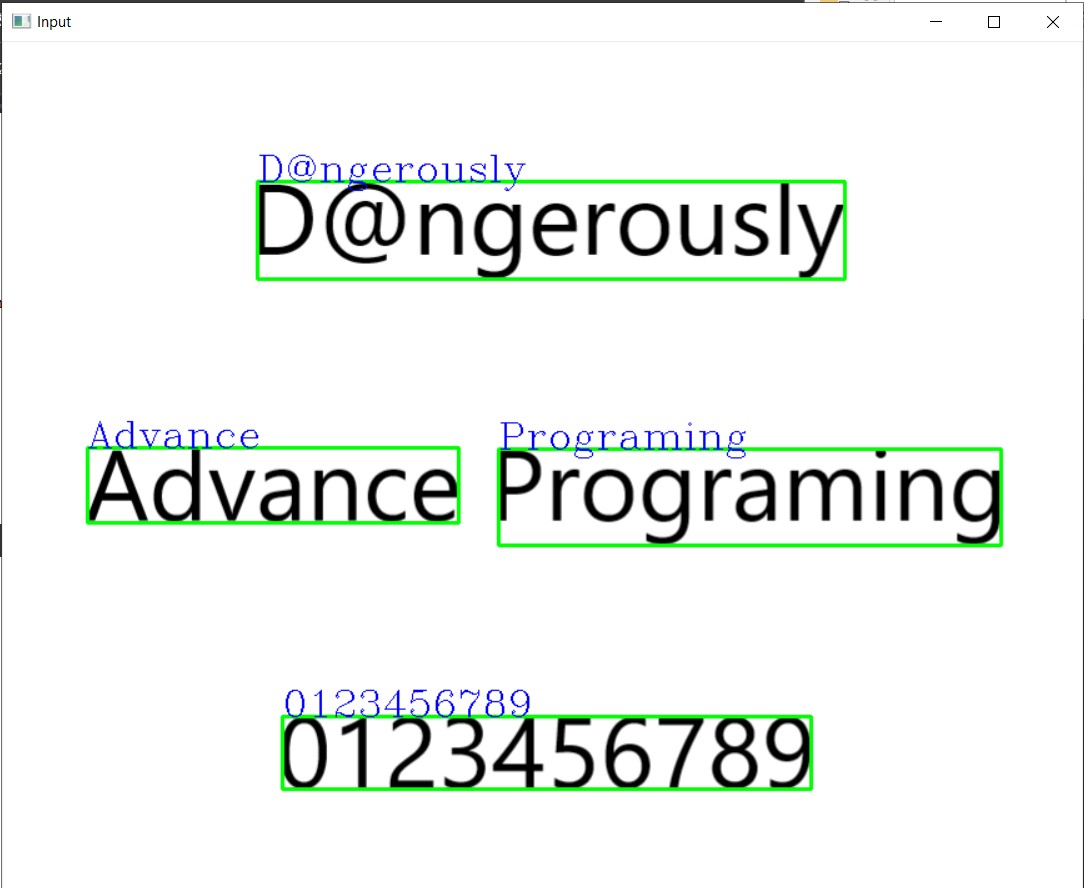


Figure 4: Result of the sample image after ran the code

**3.1 Discussion**

The result of the code is that the PyCharm able to display the text that Tesseract OCR software recognized from the sample input image on the terminal of the PyCharm and also output an image with boxes and recognized text.

Figure 1 was a png format picture that written “ D@ngerously Advance Programing 0123456789”.

In Figure 3, the information that displaying on the terminal including the level, page number, block number, parameter number, line number, word number, left, top, width, height, configuration and text. All the information was split into array so that the python code able to extract the needed information for text recognization.

In Figure 2, line 21 code was utilized the information that return from Figure 3 to make the rectangular boxes on the recognized text from the input image. The information used was left, top, width, height.

Figure 4 shows the recognized text with rectangular boxes and recognized word on the input image.

**4.0 CONCLUSION**

In conclusion, the information of text recognition was extracted by using Tesseract OCR engine and the data was displayed to user by using PyCharm. Text recognition was completed without requiring the user to perform several actions.