MALAYSIA-JAPAN INTERNATIONAL INSTITUTE OF TECHNOLOGY
ELECTRONIC SYSTEMS ENGINEERING DEPARTMENT

SEMESTER 2 2022 /2023

SMJE4263 COMPUTER INTEGRATED MANUFACTURING

INDIVIDUAL ASSIGNMENT

Extract Information from Receipt or Invoice

NAME	MATRIC NO.
1. SHIM CHUNG SIONG	A19MJ0124
NAME OF LECTURER	**PROF. MADYA. IR. DR. ZOOL HILMI BIN ISMAIL**

CHAPTER 1

INTRODUCTION

In the realm of business operations, the extraction of vital information from receipts and invoices plays a pivotal role. However, manually processing a large volume of these documents can be a time-consuming and error-prone task. Fortunately, the power of Optical Character Recognition (OCR) technology, coupled with the versatility of Python programming, offers a transformative solution for automating the extraction of data from receipts and invoices. The objective of this project is to harness the capabilities of OCR in Python to efficiently extract crucial information such as invoice number, invoice date, and amount from a variety of receipt and invoice documents. By utilizing popular libraries like pytesseract, cv2, and re, developers can leverage the strength of OCR algorithms to convert images containing text into machine-readable data.

Python's versatility and extensive library support make it an ideal choice for implementing OCR-based information extraction systems. With the aid of pytesseract, the text extraction process is simplified, enabling the conversion of images into textual representations. By employing cv2, image preprocessing techniques can be applied to enhance the OCR accuracy, optimizing the extraction process even further. Additionally, the power of regular expressions (re) allows for pattern matching, aiding in the identification and extraction of specific data elements.

Through the amalgamation of OCR, Python, and associated libraries, this project seeks to streamline the process of extracting essential information from receipts and invoices. Once the data has been successfully extracted, it can be conveniently displayed, providing businesses with an organized overview of the extracted invoice number, invoice date, and amount. By automating this otherwise cumbersome task, businesses can save time, minimize errors, and make well-informed decisions based on accurate and readily available data.

CHAPTER 2

METHODOLOGY

2.1 Libraries Utilized

To achieve the objective of extracting invoice numbers, invoice date, and amount from receipts or invoices using OCR in Python, the project will utilize several essential Python libraries: pytesseract, cv2, re, and os as shown in Figure 2.1 below. The following methodology outlines the step-by-step process involved in implementing the OCR-based information extraction system.

```
import os
import cv2
import pytesseract
import re
```

Figure 2.1: Libraries used in this project

The project starts by leveraging the “cv2” library to load the receipt or invoice image. The image is read using the “cv2.imread()” function, allowing Python to access the image's pixel data. Next, the image is converted to grayscale using the “cv2.cvtColor()” function. Converting the image to grayscale simplifies subsequent processing steps and enhances the OCR accuracy.

For OCR text extraction, The project incorporates the “pytesseract” library, which interfaces with the Tesseract OCR engine, to extract text from the preprocessed image. The “pytesseract.image_to_string()” function is employed to perform OCR and obtain the extracted text from the image.

Regular expressions (regex) play a crucial role in identifying and extracting specific information from the extracted text. The “re” library is utilized to apply regex patterns and extract invoice numbers, invoice date, and amount. Regex patterns are designed to match and capture relevant data elements based on keywords such as "invoice no," "invoice date," "total," or "grand total" present in the extracted text. The “re.search()” function is utilized to locate and extract the desired information.

The “os” library in Python is used for file and directory operations, including reading images from a folder when performing OCR. It provides functions to interact with the underlying operating system and perform operations such as listing files in a directory, accessing file properties, and navigating the file system. The batch processing using the “os” library is to facilitate the extraction of information from multiple receipt or invoice images, the “os” library is employed to handle file operations and directory traversal. This project utilize “os.listdir()” to retrieve a list of files in a specified folder, enabling batch processing of multiple images.

2.2 Codes

```
1  import os
2  import cv2
3  import pytesseract
4  import re
5
6  pytesseract.pytesseract.tesseract_cmd = "C:\\Program Files\\Tesseract-OCR\\tesseract.exe"
7
8  def extract_information_from_image(image_path):
9      # Load the image using OpenCV
10     image = cv2.imread(image_path)
11
12     # Convert the image to grayscale
13     gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
14
15     # Perform OCR using pytesseract
16     extracted_text = pytesseract.image_to_string(gray_image)
17
18     # Process the extracted text to extract relevant information
19     lines = extracted_text.split('\n')
20     invoice_number = ''
21     date = ''
22     amount = ''
23
24     # Iterate over each line and extract the relevant information
25     for line in lines:
26         if not invoice_number:
```

Figure 2.2: Coding from line 1 to 26

```
27                # Extract invoice number (look for "invoice no" keyword)
28                match = re.search(r'(?i)invoice\s*no[\s:]+([a-zA-Z0-9]+)', line)
29                if match:
30                    invoice_number = match.group(1).strip()
31            if not date:
32                # Extract date (look for "invoice date" keyword)
33                match = re.search(r'(?i)invoice\s*date[\s:]+([\w\s]+)', line)
34                if match:
35                    date = match.group(1).strip()
36            if not amount:
37                # Extract amount (look for "total" or "grand total" keyword and number on the same line)
38                if re.search(r'(?i)total|grand total', line):
39                    words = line.split()
40                    for word in words:
41                        if word.replace('.', '').isdigit():
42                            amount = word.strip()
43
44        # Return the extracted information
45        return invoice_number, date, amount
46
47    # Folder containing invoice images
48    folder_path = "invoice"  # Update with your folder path
49
50    # Iterate over images in the folder
```

Figure 2.3: Coding from line 27 to 50

```
51    for filename in os.listdir(folder_path):
52        if filename.endswith(".jpg") or filename.endswith(".png"):
53            # Full path of the image file
54            image_path = os.path.join(folder_path, filename)
55
56            # Extract information from the image
57            invoice_number, date, amount = extract_information_from_image(image_path)
58
59            # Print the extracted information
60            print("Invoice Number:", invoice_number)
61            print("Date:", date)
62            print("Amount: RM", amount)
63            print("-------------------------------------")
```

Figure 2.4: Coding from line 51 to 63

CHAPTER 3

RESULT AND DISCUSSION

With 2 invoice pictures provided, the Python code is able to be executed and extracts invoice number, invoice date and amount from the picture. Figure 3.1 below shows the result in terminal after the code is executed.

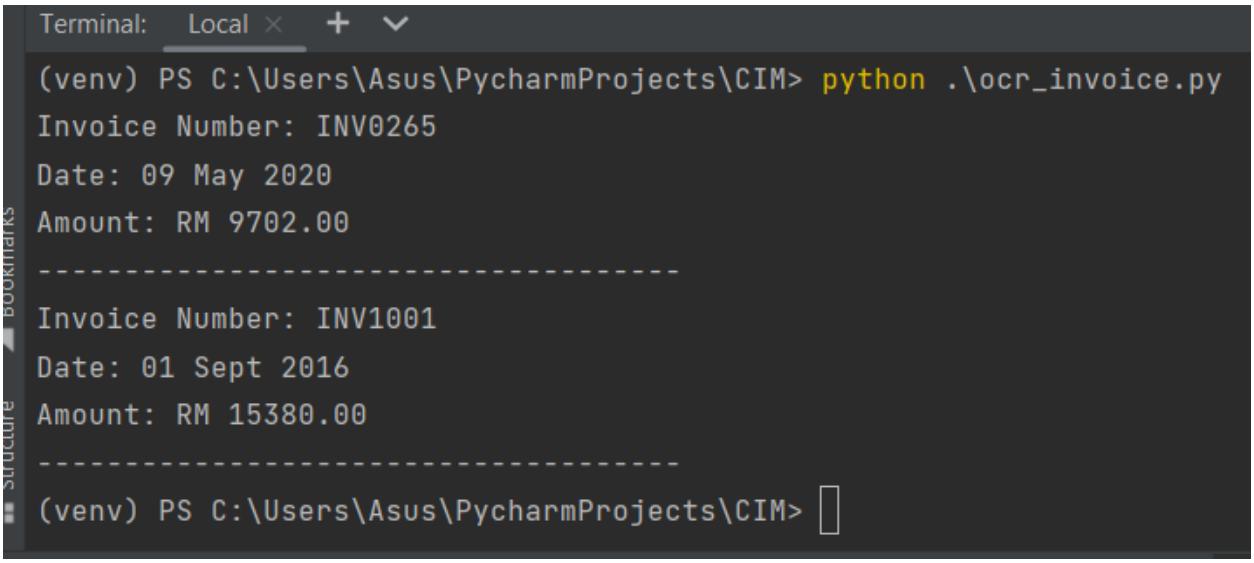

Figure 3.1: Output after running Python file

From the results shown for the first invoice, the information extracted is INV0265, 09 May 2020 and RM9702.00 for invoice number, invoice date and amount respectively. For the second invoice, the information extracted is INV1001, 01 Sept 2016, RM15380.00 for invoice number, invoice date and amount respectively. By comparing the result with the invoice in the picture, it can be shown that all the information is extracted correctly. Figure 3.2 below shows the first invoice to be extracted whereas Figure 3.3 below shows the second invoice.

Biztory Cloud Accounting Software

CO3 Social Office,
2-3, Jalan Merbah 1,
Bandar Puchong Jaya,
47170 Puchong, Selangor.
T: 012-3803369
E: ask@biztory.com.my

BILL TO

Biztory Cloud Accounting Software
2-3, Jalan Merbah 1, Bandar Puchong, Jaya, Bandar Puchong Jaya, Puchong, 47170 Selangor
T: 0123803369
E: ask@biztory.com.my

DELIVER TO

INVOICE DETAILS

SALES INVOICE

Invoice No	INV0265
Invoice Date	09 May 2020
DELIVERY DATE	09 May 2020
SALES PERSON	snenterprise

NO.	DESCRIPTION	QTY	UNIT	PRICE (MYR)
1	Name card	1		1.50
2	Apple iMac with Retina 5K display 27-inch - 8GB 2666MHz DDR4 memory - 1TB Fusion Drive storage - Magic Mouse 2	1		7,699.00
3	new ipad 2018 - RoseGold	1		2,000.00
4	Name card	1		1.50

PAYMENT TERM	Cash
MYR 9,702.00	Due on 09 May 2020

Total	9702.00
DUE NOW	9,702.00

Figure 3.2: First invoice picture to be extracted

INVOICE//INV1001

Reg. No
Sales Tax ID No.:
Services Tax ID No.:

CUSTOMER
syafrie shafie
Service Tax Reg No.:
Financio Sdn Bhd
KA3-2-13, Kuchai Avenue
39 Jalan Kuchai Maju 13
Kuchai Lama,
58200 Kuala Lumpur

Invoice No	INV1001
Invoice Date	01 Sept 2016

#	ITEM	TAX	QTY	RATE (RM)	TOTAL (RM)
1	**Red T-shirt**	ST10	1	1,000.00	1,000.00
2	**Blue T-shirt**	ST05	1	2,000.00	2,000.00
3	**Green T-shirt**	SV06	1	3,000.00	3,000.00
4	**Printing services**	TD06	1	4,000.00	4,000.00
5	**Packaging services**	ESC5	1	5,000.00	5,000.00

SUB TOTAL	RM 15,000.00
SST 6%	RM240.00
B2B EXEMPTION	(RM 240.00)
Grand Total	15380.00

Figure 3.3: Second invoice picture to be extracted

CHAPTER 4

CONCLUSION

In conclusion, using OCR in Python to extract information from receipts and invoices is a powerful and efficient solution. The objective of this project was achieved by extracting the invoice number, invoice date, and amount. By leveraging Python libraries such as pytesseract, cv2, re, and os, the project successfully automated the extraction process.

OCR in Python, through the “pytesseract” library, enabled the extraction of text from receipt and invoice images, while “cv2” facilitated image preprocessing for improved accuracy. The regular expressions “re” library played a vital role in pattern matching and extracting specific data elements, while the “os” library provided the means for batch processing multiple images.

By harnessing OCR in Python, businesses can streamline their operations, reduce manual effort, and minimize errors during data entry. The automated extraction of invoice information allows for efficient financial analysis, inventory management, and decision-making. Displaying the extracted information provides a concise summary for quick reference.

Overall, the use of OCR in Python for extracting invoice information presents significant benefits for businesses. By integrating this technology into existing workflows, businesses can optimize their processes, save time, and make data-driven decisions based on accurate and accessible information. With ongoing advancements in OCR and Python, the potential for further improvements in extraction accuracy and efficiency is promising.