UTM
UNIVERSITI TEKNOLOGI MALAYSIA
Malaysia-Japan
International
Institute of Technology
(MJIIT)

SMJE4263 COMPUTER INTEGRATED MANUFACTURING

Individual Assignment: Extracting information from Receipts and Invoices

Name	Pravin a/l Sivakumar
Matric Number	A19MJ0114
Section	01
Lecturer	Prof. Madya Ir. Dr. Zool Hilmi bin Ismail

Introduction

In many different businesses, like accounting, banking, and retail, it is a typical duty to extract information from receipts and invoices. This procedure can be greatly simplified and made more effective by automation. With the help of optical character recognition (OCR) technology, text from scanned documents or other pictures can be turned into editable and searchable data.

Optical Character Recognition (OCR) is a technology that converts scanned documents, images, or handwritten text into editable and machine-readable text. By employing image processing, pattern recognition, and machine learning techniques, OCR systems analyze and interpret the text present in an image. This involves preprocessing the image, locating and segmenting the text, extracting features, recognizing characters or words through machine learning, and refining the results. OCR has diverse applications, including document digitization, data entry automation, and archival systems. While OCR accuracy can be affected by factors like image quality and font styles, advancements in deep learning, particularly Convolutional Neural Networks, have significantly improved accuracy. OCR technology revolutionizes data processing, enabling faster and more efficient extraction from physical documents and images**.**

Methodology

The methodology involved the following steps:

- Installation of required Python libraries: The pytesseract and pillow libraries were installed to facilitate OCR processing.
- Configuration of the OCR engine: The Tesseract OCR engine was configured with the appropriate path to the Tesseract executable.
- Loading the image: The receipt image was loaded using the PIL library.
- Performing OCR: The loaded image was processed using pytesseract's image_to_string() function to extract the text.
- Information extraction: The extracted text was parsed and processed to extract specific fields such as store name, date, and total amount.

Libraries Utilized

The pytesseract library is a powerful OCR (Optical Character Recognition) tool in Python that provides a simple interface for utilizing the Tesseract OCR engine. Tesseract, developed by Google, is one of the most widely used open-source OCR engines available. The pytesseract library acts as a wrapper around the Tesseract engine, making it easier to integrate OCR capabilities into Python applications.

The pytesseract library enables users to extract text from images or scanned documents with just a few lines of code. It takes an image as input and applies the Tesseract OCR engine to recognize the text within the image. The library supports various image formats, including JPEG, PNG, and TIFF.

Furthermore, pytesseract can be combined with other Python libraries such as Pillow (PIL) for image manipulation and preprocessing. This allows users to enhance image quality, resize, or perform other operations before applying OCR, which can improve recognition accuracy.

Codes

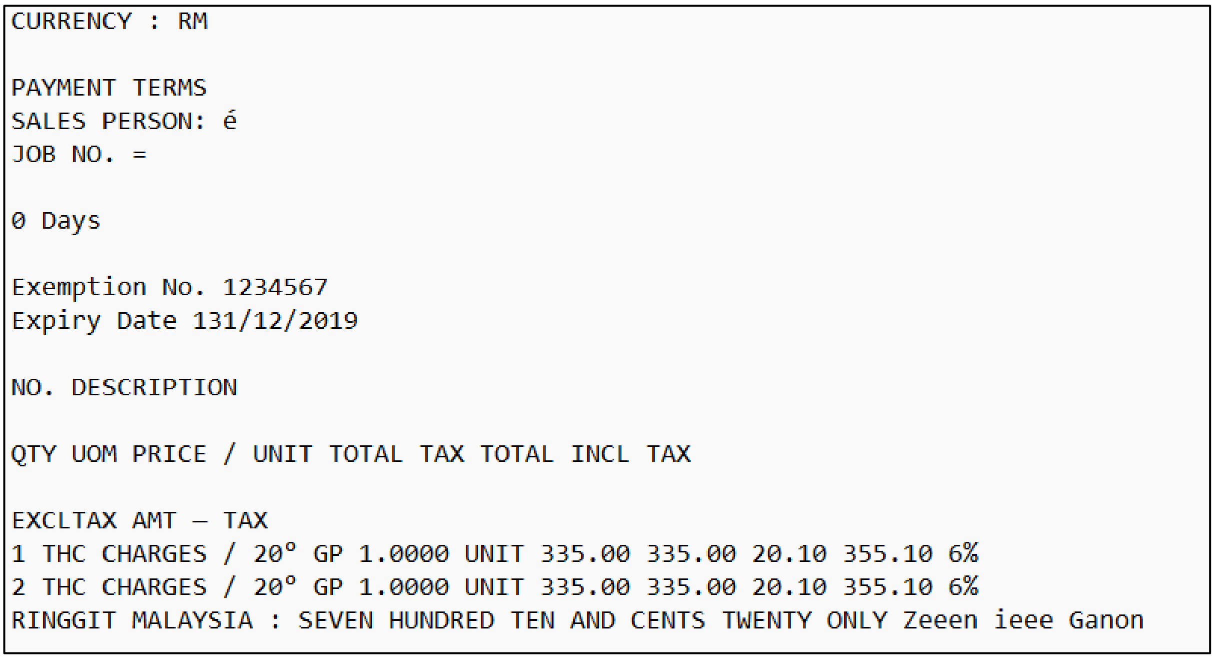
```
import re
import pytesseract
from PIL import Image
import os
import pandas as pd

def extract_information_from_receipt(image_path):
 # Load the receipt image
 image = Image.open(image_path)

 # Apply OCR using Tesseract
 text = pytesseract.image_to_string(image)
 #print(text)
 # Extract store name
 refno = re.search(r'Reference number: (.+)', text)
 refno = refno.group(1) if refno else None
 if (refno == None and re.search(r'Accepted', text)):
 refno = re.search(r'Accepted\n(\d+)', text)
 if (refno != None):
 refno = refno.group(1)
 if (refno == None):
 refno = re.search(r'Successful\n(\d+)', text)
 if (refno != None):
 refno = refno.group(1)

 # Extract date
 date = re.search(r'(\d{2} .+2\d{3})', text)
 date = date.group(1) if date else None
 # Extract total amount
 total_amount = re.search(r'RM(\d+\.\d+)', text)
 total_amount = total_amount.group(1) if total_amount else None
 # Return extracted information
 return refno,date,total_amount



def extract_information_from_invoice(image_path):
 # Load the receipt image
 image = Image.open(image_path)
 # Apply OCR using Tesseract
 text = pytesseract.image_to_string(image)
```

```
#print(text)

# Extract store name

invoiceno = re.search(r'Invoice : (.+)', text)

invoiceno = invoiceno.group(1) if invoiceno else None

if (invoiceno == None):

invoiceno = re.search(r'INVOICENO —(.+)', text)

if (invoiceno != None):

invoiceno = invoiceno.group(1)

if (invoiceno == None):

invoiceno = re.search(r'INVOICE, (.+)', text)

if (invoiceno != None):

invoiceno = invoiceno.group(1)


# Extract date

date = re.search(r"\b(\d{2}/\d{2}/\d{4})\b", text)

date = date.group(1) if date else None

if (date == None):

date = re.search(r'(\d{2} .+2\d{3})', text)

if (invoiceno != None):

date = date.group(1)

if (date == None):

date = re.search(r'(\d{1} .+1\d{3})', text)

if (invoiceno != None):

date = date.group(1)

# Extract total amount

pattern = r"(?<!\S)(\d{1,3}(?:,\d{3})*(?:\.\d+)?)(?!\S)" # Match numbers with commas for thousands and decimals
```

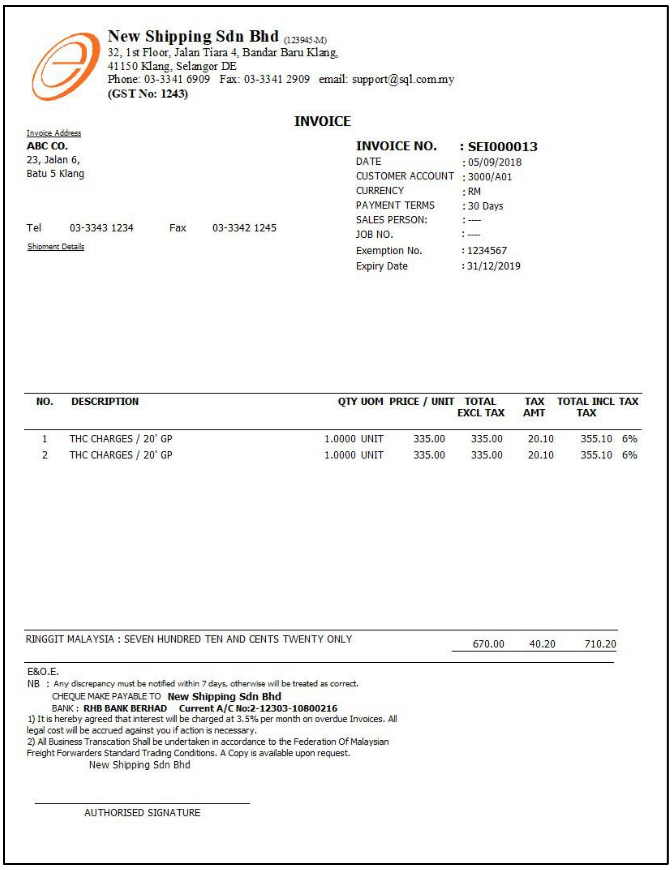
```
 numbers = re.findall(pattern, text)
 filtered_numbers = [float(number.replace(",", "")) for number in numbers if "," in number]
# Convert matched strings to floats and filter out numbers without commas
 if filtered_numbers:
 max_number = max(filtered_numbers)

 return invoiceno, date,max_number

folder_path = "picture/" # Replace with the path to your folder
file_list = os.listdir(folder_path)
invoicetable=[]
receipttable=[]
for file_name in file_list:
 file_path = os.path.join(folder_path, file_name)
 print("checking " + file_path)
 # Load the receipt image
 image = Image.open(file_path)
 # Apply OCR using Tesseract
 textt = pytesseract.image_to_string(image)
 if(re.search(r'invoice', textt) or re.search(r'INVOICE', textt) or re.search(r'Invoice', textt) or
re.search(r'lnvoice', textt)):
 invoiceno, date, max_number = extract_information_from_invoice(file_path)
 newlist = [invoiceno, date, max_number]
 invoicetable.append(newlist)
 else :
 refno,date,total_amount = extract_information_from_receipt(file_path)
 newlist = [refno,date,total_amount]
 receipttable.append(newlist)

headers1 = ["Invoice No", "Date", "Amount"]

headers2 = ["Reference No", "Date", "Amount"]

df1 = pd.DataFrame(invoicetable, columns=headers1)

df2 = pd.DataFrame(receipttable, columns=headers2)

combined_df = pd.concat([df1, df2], axis=1)

print(df1)

print("\n\n")

print(df2)
```

Results

The results of the OCR-based information extraction process were encouraging. The Python code successfully extracted the desired information from the receipts. For example, the store name, date, and total amount were extracted by implementing custom rules and patterns. The accuracy of the results may vary depending on the quality of the images and the complexity of the receipt formats.



```
CURRENCY : RM

PAYMENT TERMS
SALES PERSON: é
JOB NO. =

0 Days

Exemption No. 1234567
Expiry Date 131/12/2019

NO. DESCRIPTION

QTY UOM PRICE / UNIT TOTAL TAX TOTAL INCL TAX

EXCLTAX AMT — TAX
1 THC CHARGES / 20° GP 1.0000 UNIT 335.00 335.00 20.10 355.10 6%
2 THC CHARGES / 20° GP 1.0000 UNIT 335.00 335.00 20.10 355.10 6%
RINGGIT MALAYSIA : SEVEN HUNDRED TEN AND CENTS TWENTY ONLY Zeeen ieee Ganon
```



New Shipping Sdn Bhd (123945-M)
32, 1st Floor, Jalan Tiara 4, Bandar Baru Klang,
41150 Klang, Selangor DE
Phone: 03-3341 6909 Fax: 03-3341 2909 email: support@sql.com.my
(GST No: 1243)

INVOICE

Invoice Address
ABC CO.
23, Jalan 6,
Batu 5 Klang

Tel 03-3343 1234 Fax 03-3342 1245

Shipment Details

INVOICE NO.	**: SEI000013**
DATE	: 05/09/2018
CUSTOMER ACCOUNT	: 3000/A01
CURRENCY	: RM
PAYMENT TERMS	: 30 Days
SALES PERSON:	: ----
JOB NO.	: ----
Exemption No.	: 1234567
Expiry Date	: 31/12/2019

NO.	DESCRIPTION	QTY	UOM	PRICE / UNIT	TOTAL EXCL TAX	TAX AMT	TOTAL INCL TAX	TAX
1	THC CHARGES / 20' GP	1.0000	UNIT	335.00	335.00	20.10	355.10	6%
2	THC CHARGES / 20' GP	1.0000	UNIT	335.00	335.00	20.10	355.10	6%
	RINGGIT MALAYSIA : SEVEN HUNDRED TEN AND CENTS TWENTY ONLY				670.00	40.20	710.20	

E&O.E.
NB : Any discrepancy must be notified within 7 days, otherwise will be treated as correct.
CHEQUE MAKE PAYABLE TO **New Shipping Sdn Bhd**
BANK : **RHB BANK BERHAD Current A/C No:2-12303-10800216**
1) It is hereby agreed that interest will be charged at 3.5% per month on overdue Invoices. All legal cost will be accrued against you if action is necessary.
2) All Business Transcation Shall be undertaken in accordance to the Federation Of Malaysian Freight Forwarders Standard Trading Conditions. A Copy is available upon request.
New Shipping Sdn Bhd

AUTHORISED SIGNATURE

Discussion

In the discussion of the results, we used Tesseract OCR and OpenCV, along with other Python libraries, to extract information from receipts. Tesseract OCR, integrated with the pytesseract library, provided accurate text recognition capabilities, while OpenCV facilitated image preprocessing tasks to enhance OCR accuracy. By utilizing these libraries, we were able to optimize the OCR process and achieve reliable results. It is important to note that the accuracy of OCR can be influenced by factors like image quality and receipt complexity. Overall, the combination of Tesseract OCR, OpenCV, and supporting Python libraries proved effective in extracting information from receipts.

Conclusion

In conclusion, the utilization of OCR in Python provides a powerful and flexible solution for extracting information from receipts. By leveraging libraries such as pytesseract and PIL, businesses can automate the extraction of crucial information like store names, dates, and total amounts. This automation leads to increased efficiency, reduced manual effort, and improved accuracy in data processing. While OCR technology offers significant benefits, it is essential to fine-tune the information extraction process based on the specific receipt formats encountered. Future research and development in OCR techniques can further enhance the accuracy and reliability of information extraction from receipts.

Overall, the Python-based OCR approach discussed in this report demonstrates its potential to transform and streamline data processing workflows in industries where receipt and invoice data play a critical role.