UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Malaysia-Japan
International Institute
of Technology
(MJIIT)

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.**
Tew Jia Jian	A19MJ0130
NAME OF LECTURER	**PROF. MADYA. IR. DR. ZOOL HILMI BIN ISMAIL**

CHAPTER 1

INTRODUCTION

In today's digital age, organizations are generating an enormous amount of data in various formats such as scanned documents, images, and PDFs. Extracting useful information from these unstructured sources can be a time-consuming and error-prone task. This is where Optical Character Recognition (OCR) and specific data extraction techniques come into play.

OCR is a technology that enables computers to interpret and convert scanned images or printed text into machine-readable text. By analyzing the shapes, patterns, and structures of characters, OCR algorithms can accurately recognize and convert the textual content into editable and searchable formats. OCR has revolutionized the way organizations handle documents, making it easier to extract valuable data and automate tedious manual processes.

One significant application of OCR is specific data extraction. It involves the extraction of specific information from structured or unstructured documents to be used in various business processes. Whether it's extracting customer details from invoices, capturing relevant information from resumes, or gathering data from medical records, specific data extraction using OCR has become an invaluable tool across industries.

The process of specific data extraction begins with preprocessing the document, which may involve noise reduction, skew correction, and image enhancement to optimize OCR accuracy. Next, the OCR engine analyzes the document, identifying characters, words, and sentences. By applying intelligent algorithms, it locates and extracts specific data elements based on predefined rules or patterns.

OCR technology can handle diverse data types, including printed or handwritten text, barcodes, tables, and even complex documents with multiple languages. By leveraging machine learning and artificial intelligence, OCR systems continuously improve their accuracy and adaptability, making them highly efficient in handling large-scale data extraction tasks.

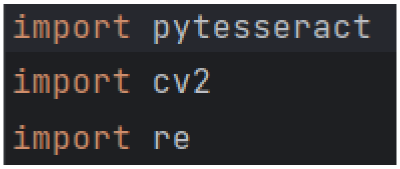
The benefits of specific data extraction using OCR are numerous. It significantly reduces manual effort and human errors associated with manual data entry, leading to improved productivity and cost savings. Moreover, it enables faster data processing, making information readily available for analysis and decision-making. With accurate and automated data extraction, organizations can streamline their workflows, enhance data governance, and unlock valuable insights hidden within unstructured documents.

CHAPTER 2

METHODOLOGY

2.1 Libraries Utilized

The code utilizes several powerful libraries to extract information from invoice images. PyTesseract, a Python wrapper for Google's Tesseract OCR engine, plays a central role in extracting text from images. It enables the code to perform optical character recognition on the invoice images, providing a foundation for text analysis. OpenCV, a widely-used computer vision library, aids in image preprocessing tasks such as loading and preparing the images for OCR. The combination of PyTesseract and OpenCV allows for efficient text extraction from invoices.To extract specific information, the code employs the 're' library for pattern matching using regular expressions. Regular expressions provide a flexible and efficient way to identify and extract invoice numbers, dates, and totals from the OCR-recognized text.



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

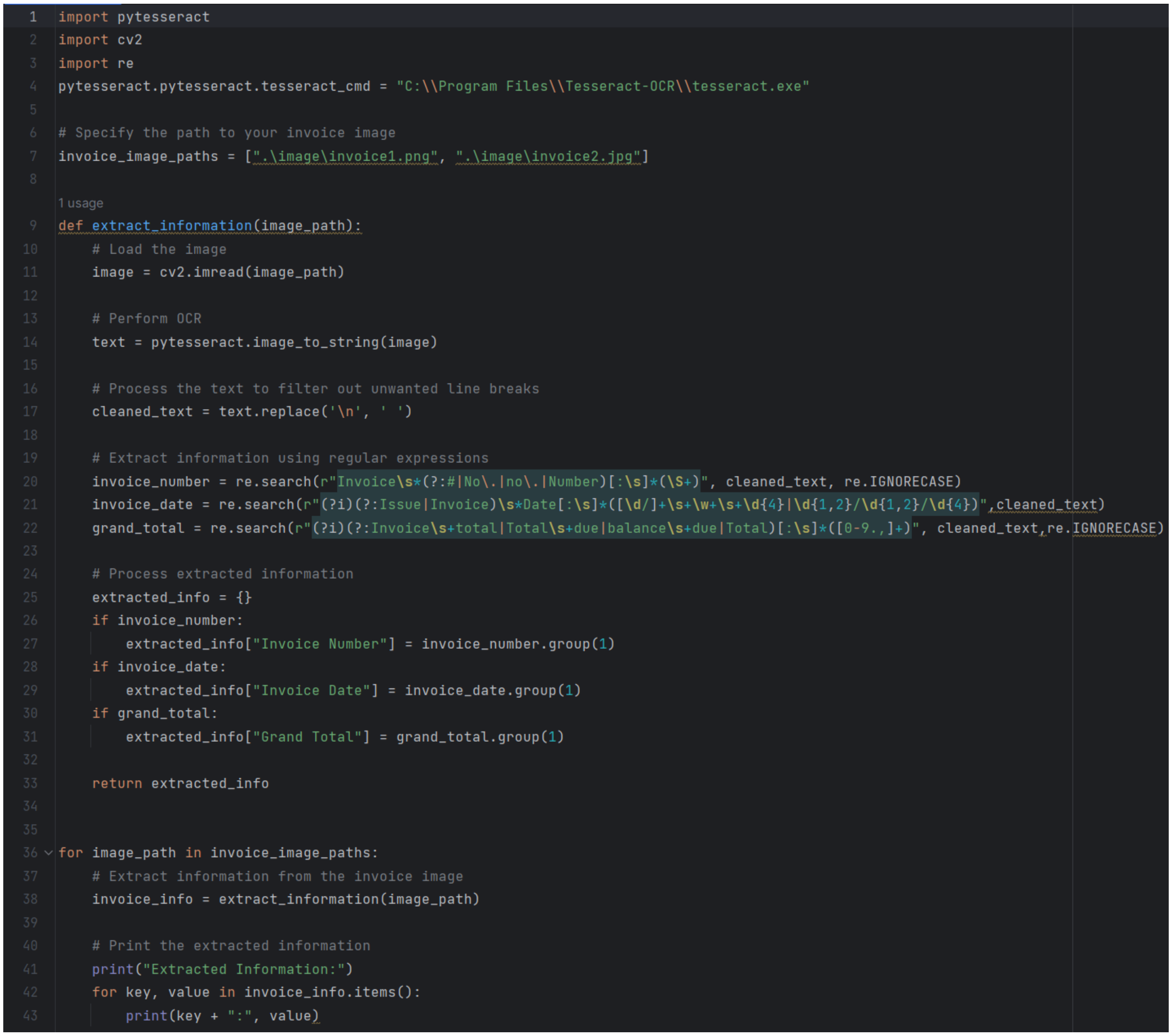
Figure 2.1: Libraries used in this project

PyTesseract is a Python wrapper for Google's Tesseract OCR (Optical Character Recognition) engine. It allows you to extract text from images or scanned documents. In the code, PyTesseract is used to perform OCR on the invoice images and extract the text.

OpenCV is a popular computer vision library that provides functions for image processing and analysis. In the code, OpenCV is used to load and preprocess the invoice images before passing them to the OCR engine.

The re module is a built-in Python library that provides support for regular expressions. Regular expressions are used to define patterns and perform pattern matching operations on text data. In the code, regular expressions are used to extract specific information (such as invoice numbers, dates, and totals) from the OCR-recognized text.

2.2 Codes



```python
import pytesseract
import cv2
import re
pytesseract.pytesseract.tesseract_cmd = "C:\\Program Files\\Tesseract-OCR\\tesseract.exe"

# Specify the path to your invoice image
invoice_image_paths = [".\image\invoice1.png", ".\image\invoice2.jpg"]

1 usage
def extract_information(image_path):
    # Load the image
    image = cv2.imread(image_path)

    # Perform OCR
    text = pytesseract.image_to_string(image)

    # Process the text to filter out unwanted line breaks
    cleaned_text = text.replace('\n', ' ')

    # Extract information using regular expressions
    invoice_number = re.search(r"Invoice\s*(?:#|No\.|no\.|Number)[:\s]*(\S+)", cleaned_text, re.IGNORECASE)
    invoice_date = re.search(r"(?i)(?:Issue|Invoice)\s*Date[:\s]*([\d/]+\s+\w+\s+\d{4}|\d{1,2}/\d{1,2}/\d{4})",cleaned_text)
    grand_total = re.search(r"(?i)(?:Invoice\s+total|Total\s+due|balance\s+due|Total)[:\s]*([0-9.,]+)", cleaned_text,re.IGNORECASE)

    # Process extracted information
    extracted_info = {}
    if invoice_number:
        extracted_info["Invoice Number"] = invoice_number.group(1)
    if invoice_date:
        extracted_info["Invoice Date"] = invoice_date.group(1)
    if grand_total:
        extracted_info["Grand Total"] = grand_total.group(1)

    return extracted_info


for image_path in invoice_image_paths:
    # Extract information from the invoice image
    invoice_info = extract_information(image_path)

    # Print the extracted information
    print("Extracted Information:")
    for key, value in invoice_info.items():
        print(key + ":", value)
```

Figure 2.1: Python Code

Figure 2.1 shows the Python code to extract the necessary information from image. The code involves several key steps to extract information from invoice images. Firstly, the invoice image is loaded using the OpenCV library, providing the input for the OCR process. PyTesseract, a Python wrapper for the Tesseract OCR engine, is then employed to perform optical character recognition on the image, converting the text within the image into machine-readable format. To enhance the text processing, the OCR-recognized text is preprocessed to remove unwanted line breaks, resulting in a cleaner and more manageable text format.

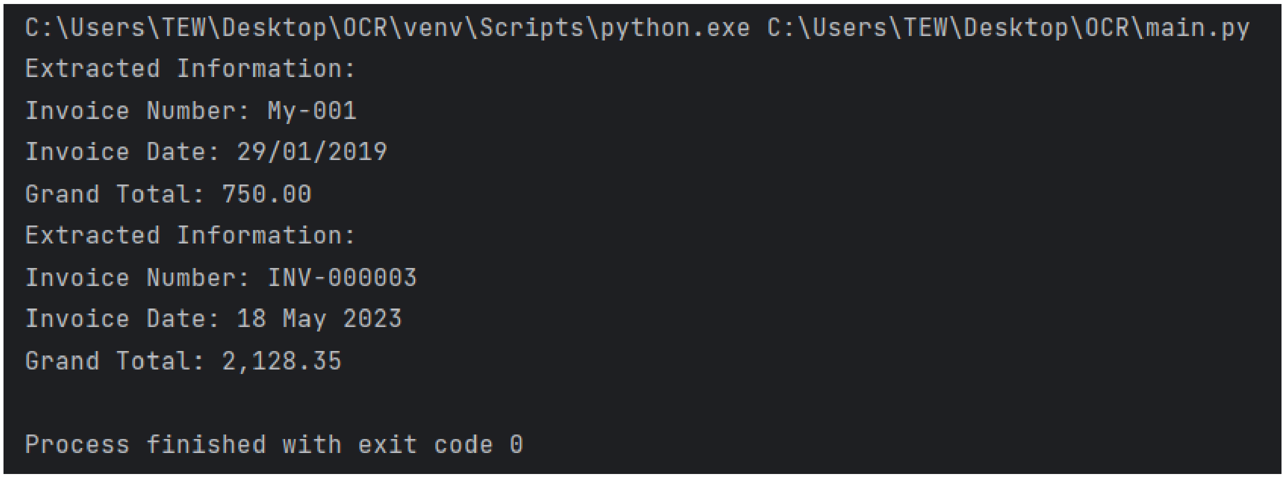
Next, the code utilizes regular expressions to extract specific information from the processed text. Regular expression patterns are crafted to identify and capture relevant data such as the invoice number, invoice date, and grand total. These patterns account for variations in keywords and associated values, ensuring flexibility in extracting the desired information accurately.

The extracted information, including the invoice number, invoice date, and grand total, is stored in a dictionary data structure. Finally, a report is generated, presenting the extracted details for each processed invoice image.

By following this methodology, the code effectively extracts essential information from invoice images, streamlining the invoice processing workflow and enabling efficient analysis of invoice data.

CHAPTER 3

RESULT AND DISCUSSION

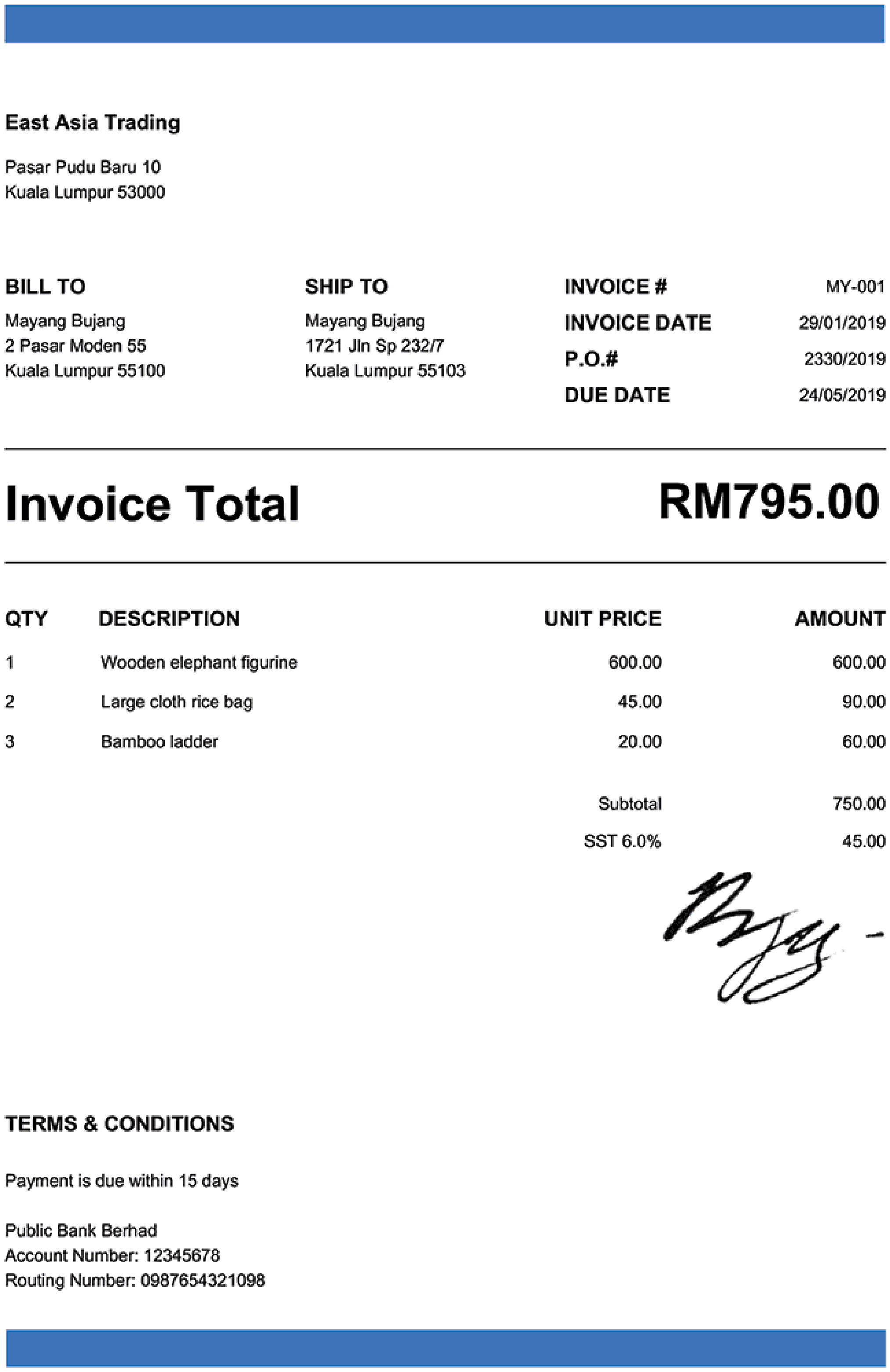


```
C:\Users\TEW\Desktop\OCR\venv\Scripts\python.exe C:\Users\TEW\Desktop\OCR\main.py
Extracted Information:
Invoice Number: My-001
Invoice Date: 29/01/2019
Grand Total: 750.00
Extracted Information:
Invoice Number: INV-000003
Invoice Date: 18 May 2023
Grand Total: 2,128.35

Process finished with exit code 0
```

Figure 3.1: Output after running Python file

From the results shown for the first invoice, the information extracted is My-001, 29/01/2019 and RM750.00 for invoice number, invoice date and amount respectively. For the second invoice, the information extracted is INV-000003, 18 May 2023, RM2128.35 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.



East Asia Trading

Pasar Pudu Baru 10
Kuala Lumpur 53000

BILL TO	**SHIP TO**	**INVOICE #**	MY-001
Mayang Bujang	Mayang Bujang	**INVOICE DATE**	29/01/2019
2 Pasar Moden 55	1721 Jln Sp 232/7	**P.O.#**	2330/2019
Kuala Lumpur 55100	Kuala Lumpur 55103	**DUE DATE**	24/05/2019

Invoice Total **RM795.00**

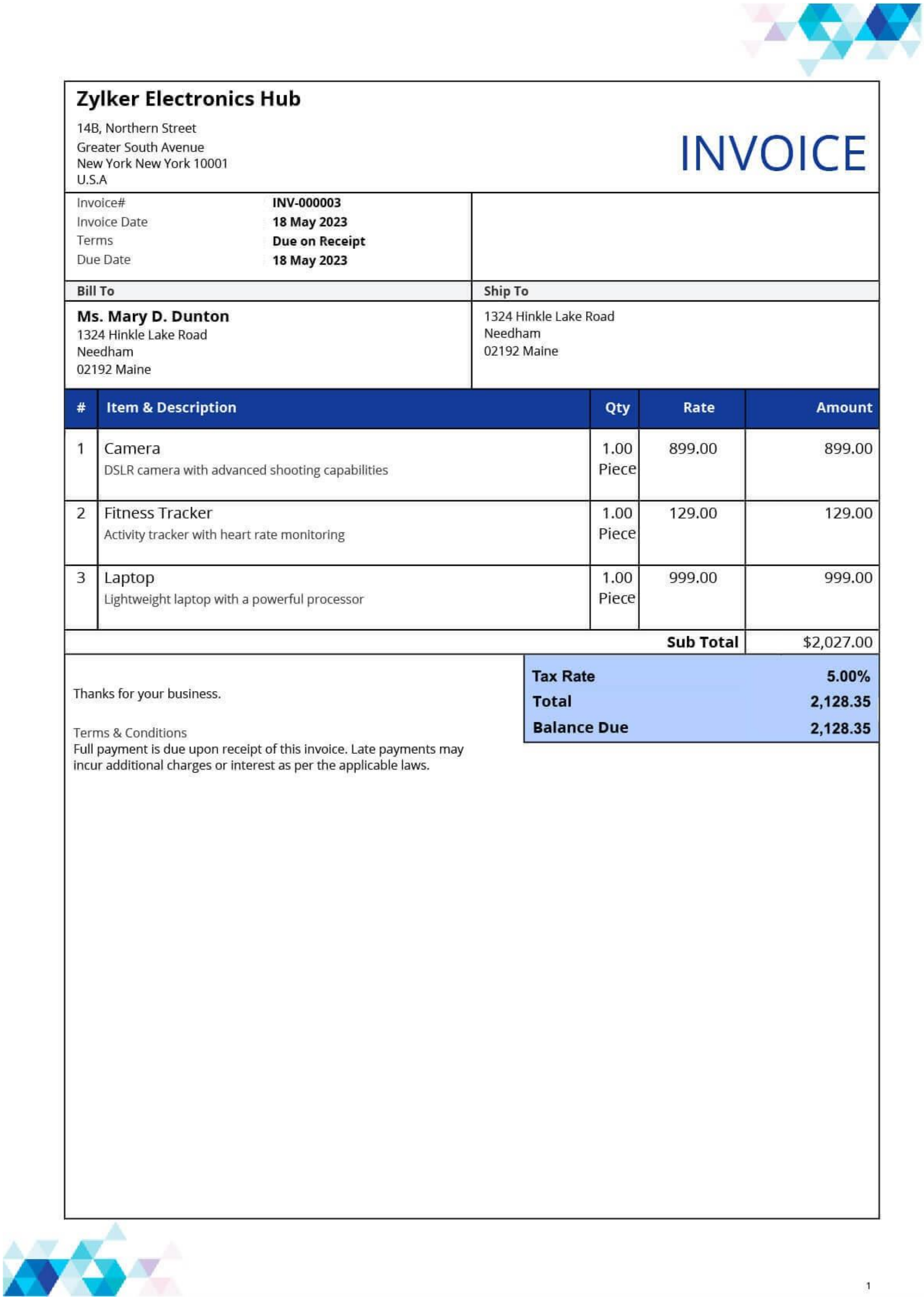
QTY	DESCRIPTION	UNIT PRICE	AMOUNT
1	Wooden elephant figurine	600.00	600.00
2	Large cloth rice bag	45.00	90.00
3	Bamboo ladder	20.00	60.00
		Subtotal	750.00
		SST 6.0%	45.00

TERMS & CONDITIONS

Payment is due within 15 days

Public Bank Berhad
Account Number: 12345678
Routing Number: 0987654321098

Figure 3.2: First invoice picture to be extracted



Zylker Electronics Hub

14B, Northern Street
Greater South Avenue
New York New York 10001
U.S.A

INVOICE

Invoice#	**INV-000003**
Invoice Date	**18 May 2023**
Terms	**Due on Receipt**
Due Date	**18 May 2023**

Bill To

Ms. Mary D. Dunton
1324 Hinkle Lake Road
Needham
02192 Maine

Ship To

1324 Hinkle Lake Road
Needham
02192 Maine

#	Item & Description	Qty	Rate	Amount
1	Camera DSLR camera with advanced shooting capabilities	1.00 Piece	899.00	899.00
2	Fitness Tracker Activity tracker with heart rate monitoring	1.00 Piece	129.00	129.00
3	Laptop Lightweight laptop with a powerful processor	1.00 Piece	999.00	999.00
			Sub Total	$2,027.00

Tax Rate	**5.00%**
Total	**2,128.35**
Balance Due	**2,128.35**

Thanks for your business.

Terms & Conditions
Full payment is due upon receipt of this invoice. Late payments may incur additional charges or interest as per the applicable laws.

1

Figure 3.3: Second invoice picture to be extracted

CHAPTER 4

CONCLUSION

In conclusion, the OCR data extraction process is a valuable technique for extracting information from invoice images. By leveraging libraries such as PyTesseract, OpenCV, and regular expressions, the code successfully retrieves critical details such as invoice numbers, invoice dates, and grand totals from the OCR-recognized text. The combination of image preprocessing, OCR, and text processing techniques ensures accurate extraction of relevant information from invoice images, regardless of variations in keyword usage, date formats, or currency symbols. This automated data extraction process reduces manual effort, enhances efficiency, and facilitates the analysis and processing of invoice data. The reliability and flexibility of the code make it a valuable tool for organizations seeking to streamline their invoice processing workflows and gain valuable insights from their invoice data.