AUTOMATED SCANNED RECEIPT PROCESSING WITH OPTICAL CHARACTER RECOGNITION AND MACHINE LEARNING

HOR ZHANG NENG

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITI MALAYA
KUALA LUMPUR

2022

AUTOMATED SCANNED RECEIPT PROCESSING WITH OPTICAL CHARACTER RECOGNITION AND MACHINE LEARNING

HOR ZHANG NENG

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITI MALAYA
KUALA LUMPUR

2022

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: HOR ZHANG NENG

Matric No: WOA190005

Name of Degree: Master of Computer Science (Applied Computing)

Title of Dissertation: Automated Scanned Receipt Processing with Optical Character Recognition and Machine Learning

Field of Study: Machine Learning

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the Universiti Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 30/01/2022

Subscribed and solemnly declared before,

Witness's Signature Date: 30.01.2022

Name:

Designation:

AUTOMATED SCANNED RECEIPT PROCESSING WITH OPTICAL CHARACTER RECOGNITION AND MACHINE LEARNING

ABSTRACT

Text detection and recognition in parsing optical character recognition (OCR) receipts are less studied than other popular OCR tasks. Study for post-OCR parsing of receipts is scarce, which opens up the opportunity to explore extracting key information from receipts and classifying them. This dissertation explores how the OCR and machine learning (ML) techniques can optimize and automate receipt handling for reimbursement purposes. Automating the reimbursement process keeps faulty reimbursement expense reporting behaviour to a minimum and speeds up employee claims. The dataset prepared for this work consists of one hundred receipts commonly found in Malaysia's employee expense reimbursement report. The receipts are organized into six categories: meals, groceries, petrol, accommodation, telecommunication, and transportation fares. The receipts are of Malaysian origin, and the language of receipts is restricted to only containing English text. This work does not consider parsing handwriting on the receipt nor addresses text ambiguity. The text processing accuracy follows the accuracy of the OCR tool selected. This dissertation proposes three objectives; developing an image processing framework in improving receipt quality pre-parsing, recognizing text and extracting key information from receipts using the OCR technique, and evaluating the ML classifiers in improving receipt classification post-parsing. The overall text extraction is 90.72% and 78.51% accurate at character and word level, with harmonic mean of the precision and recall, F1 score of 0.89 and 0.78. Overall accuracy for key information extraction is 74.33%, with an F1 score of 0.74. Seven ML classifiers, Naïve Bayes, maximum entropy, Support Vector Machine (SVM), linear Support Vector Classifier (SVC), k-nearest neighbours (KNN), decision tree and random forest, were compared.

They perform between 52% and 80% overall, with F1 scores between 0.55 and 0.79. Interestingly, the linear SVC has the highest score and accuracy for its searching capability in finding the best dividing field that separates high-dimensional text data into classes.

Keyword: Understanding receipts, OCR parsing, machine learning classification, reimbursement process

PEMPROSESAN RESIT DIIMBAS AUTOMATIK DENGAN PENGIKTIRAFAN KARAKTER OPTIK DAN PEMBELAJARAN MESIN

ABSTRAK

Pengesanan dan pengecaman teks dalam penghuraian pengecaman aksara optik (OCR) adalah masalah yang kurang dikaji berbanding permasalahan popular OCR yang lain. Jarang kajian penghuraian resit pasca OCR dijumpai, membuka peluang meneroka pengekstrakan maklumat penting dari resit dan mengklasifikasikannya. Disertasi ini meneroka bagaimana teknik-teknik OCR dan pembelajaran mesin (ML) dapat mengoptimum dan mengautomasikan pengendalian proses tuntutan bayaran balik. Proses tuntutan bayaran balik berautomasi boleh meminimumkan ralat dalam pelaporan perbelanjaan sekaligus mempercepatkan tuntutan pekerja. Set data yang disediakan untuk kerja ini terdiri daripada seratus resit yang biasa didapati dalam laporan pembayaran perbelanjaan pekerja di Malaysia. Resit-resit disusun mengikut enam kategori: makanan, barang runcit, petrol, penginapan, telekomunikasi, dan pengangkutan. Hanya resit berbahasa Inggeris dan berasal dari Malaysia dikira dalam set data tersebut. Kajian ini tidak mempertimbangkan resit yang mengandungi tulisan tangan atau akan menangani permasalahan kekaburan teks. Ketepatan pemprosesan teks juga adalah bergantung kepada ketepatan alat OCR yang dipilih. Disertasi ini mencadangkan tiga objektif; membangunkan kerangka gambar pra-pemprosesan dalam meningkatkan kualiti resit pra-penguraian, mengecam teks dan mengekstrak maklumat penting dari resit menggunakan teknik OCR, dan menilai pengelasan-pengelasan ML dalam meningkatkan pengelasan resit pasca-penguraian. Ketepatan keseluruhan kerangka pra-pemprosesan imej dalam pengekstrakan teks adalah 90.72% pada tahap aksara dan 78.51% pada tahap perkataan, dengan min harmonik untuk ketepatan dan ingatan semula, skor F1 masing-masing 0.89 dan 0.78. Ketepatan keseluruhan pengekstrakan maklumat utama pula adalah 74.33%,

dengan skor F1 0.74. Tujuh pengkelasan ML, *Naïve Bayes*, Entropi Maksimum, *Support Vector Machine (SVM)*, *Linear Support Vector Classifier* (*SVC)*, *k-nearest neighbours (KNN)*, pepohon keputusan dan hutan rawak, dibandingkan. Ketepatan pengelasan adalah diantara 52% dan 80% dengan skor F1 masing-masing antara 0.55 dan 0.79. Menariknya, *Linear SVC* mempunyai skor dan keupayaan ketepatan tertinggi dalam mencari medan pemisah terbaik yang memisahkan data teks dimensi tinggi kepada kelas.

Kata kunci: Memahami resit, penghuraian OCR, klasifikasi pembelajaran mesin, proses tuntutan pembayaran balik

ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude to my dissertation's supervisor, Dr Zati Hakim Azizul Hasan, who gave me a lot of guidance and support throughout my study. Without her guidance, passion, patience and immense knowledge, this dissertation would be impossible to complete.

Next, I would like to thanks my friends for sharing their time and knowledge. I am grateful because my friends have always been there to support me. Then, I want to thank my beloved family for their morale support that encourages and gives me strength in completing this dissertation.

Thank you all,

Aug 2021,

Hor Zhang Neng

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

API	:	Application Programming Interface
CBOW	:	Continuous bag-of words
CNN	:	Convolutional Neural Network
DPI	:	Dots Per Inch
DT	:	Decision Tree
ICDAR	:	International Conference on Document Analysis and Recognition
JPEG	:	Joint Photographic Experts Group
KNN	:	K-Nearest Neighbors
L.SVC	:	Linear Support Vector Classifier
ME	:	Maximum Entropy
ML	:	Machine learning
NB	:	Naïve Bayes
NLP	:	Natural Language Processing
OCR	:	Optical Character Recognition
PDF	:	Portable Document Format
PIL	:	Python Imaging Library
RCNN	:	Region-Based Convolutional Neural Network
REST	:	Representational State Transfer
RF	:	Random Forest
SME	:	Small and Medium-sized Enterprises
SROIE	:	Scanned Receipt OCR and Key Information Extraction
SVC	:	Support Vector Classifier
SVM	:	Support Vector Machine
TF-IDF	:	Term Frequency–Inverse Document Frequency
VOC	:	Visual Object Class
Word2Vec	:	Word to Vector

CHAPTER 1: INTRODUCTION

1.1 Background

Keeping tabs on employee expenses is compulsory for businesses. When employees log travel expenses, they are often required to manually input data from receipts into a reimbursement report sheet. The reimbursement report sheet usually comprises several categories, such as mode of transport, fuel, meals, accommodation, and more. Receipts are attached to the report as proof of purchase. Filing up such a report is usually a chore done upon return to the office.

Manual handling of receipts can be daunting, and the task of entering data into an expense database is often subjected to human error and bias. The time and effort in transferring information from physical objects are better spent on something more productive. A more attractive solution is to convert the manual process into an automatic one. The first step towards automating the process is parsing scanned receipts to a program for information, often text, and extraction. For an expenditure report, the parsing must consider specific dataset statistics such as store information like name and address, payment information like date and time of purchase, additional details like mileage for taxi or menu for food, and the total amount paid.

Parsing information from receipts and invoices can be done using the Optical Character Recognition (OCR) tool. The OCR is a character recognition tool that can turn characters on a page into live text. The OCR is widely used in practical commercial tasks such as business card recognition, vehicle number plate recognition and recognizing handwriting on paper (Hwang et al., 2019; Arif & Javed, 2020; Kaur & Banga, 2013; Kakani et al., 2017; Tejas et al., 2017; Adriano et al., 2019; Yin et al., 2019). However, receipt OCR has a better performance compared to the commercial OCR tasks. The main reason is that

scanned receipts are sometimes susceptible to low scan quality. Therefore, businesses still rely on human supervision in the scanned receipt OCR and key information extraction (SROIE) area.

Research works and publications on the SROIE topic are surprisingly modest. A global competition, the International Conference on Document Analysis and Recognition (ICDAR), was hosted in 2019, addressing the challenges regarding SROIE (Huang et al., 2019; Rigaud et al., 2019). One of their discussions centres on the direct correlation between the text extraction accuracy and the quality and condition of the objects or characters in the scanned image. The interdependency leads to the possibility of preprocessing the image to improve the quality of the immutable attributes.

Preprocessing an image is recommended as it may increase the character-level precision and word-level precision. Suponenkovs et al. (2017) proposed classical image processing tools to make character blocks more defined for extraction. A character-level accuracy of 99% describes a probability that 1 in 100 characters is untrustable. In comparison, an accuracy of 99.9% shows that 1 out of 1000 characters is uncertain. There are about five OCR tools available, including the Tesseract, Azure Computer Vision API, Google Drive REST API, Text Fairy, and ABBYY FineReader. Among these tools, the Tesseract is highly regarded as the best open-source options.

1.2 Motivation

For receipt detection and OCR tasks, text localization refers to an annotated dataset. Each receipt image has a bounding box (or bbox) marking, which denotes the location of a key field text. A key field text is used to locate, for example, the position of text on the receipt that depicts the goods name, the merchants' name, the unit price, or the total cost. These

positions are called rectangles with four vertices arranged clockwise from the top of the image throughout the ICDAR2019 competition. The goal for receipt parsing with text localization is to localise texts with four vertices accurately. The text localization ground truth is determined to match at words level following the notional taxonomy of the Visual Object Class (Everingham et al., 2015). Table 1.1 shows an example of the VOC representations.

Table 1.1: The VOC classes from Everingham et al. (2015)

Vehicles	**Household**	**Animals**	**Other**
Airplane	Bottle	Bird	Person
Bicycle	Chair	Cat	
Boat	Dining table	Cow	
Bus	Potted plant	Dog	
Car	Sofa	Horse	
Motorbike	TV/monitor	Sheep	
Train			

It is noteworthy to learn that finding instances of text in an image makes recognition a more challenging task than classification. Guessing the suitable class is reportedly not easy and very demanding too. Performing 'objectness' analysis or score, i.e., drawing a detector's attention to specific points within an image, has been enhanced by the advent of deep learning algorithms such as CNN and Faster-RCNN (Nagaoka et al., 2017; Gomez et al., 2017).

The ICDAR2019 competition provides datasets that can be downloaded from Google Drive and Baidu Pan. For receipt recognition and OCR tasks, each image in the data record is annotated with a text frame (bbox) and the transcript of each bbox. If an annotated dataset is not available, parsing OCR receipt needs to be carried out without localization information. Although, providing a list of words recognized in the receipt image can improve text recognition performance. Without bbox to guide, string

manipulations such as tokenization and splitting is required to increase ground truth matching. The CountVectorizer is a vectorization and tokenization tool by Scikit-learn for the task. This tool can tokenise and divide the text string into several tokens.

The CountVectorizer takes all tokens and creates a matrix-vector with the tokens based on their frequency. Discarding unwanted text using filters, databases, and thesaurus can also be considered in this step so that the extracted text makes sense and does not require further inspection (Ullah et al., 2018). Regardless of whether text localization is present, the OCR receipt parsing should output a collection of the raw text of the receipt (see Fig. 1.1 for an example).

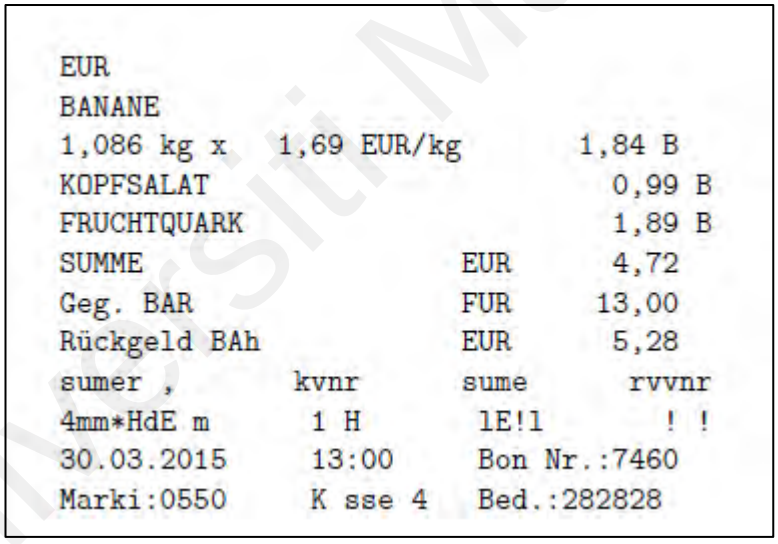


Figure 1.1: Excerpt of raw text supermarket receipt after OCR parsing (Ziegaus, 2016)

Figure 1.1 shows the raw text from the OCR parsing is not particularly meaningful to understand scanned receipt. Also, not all characters are essential, especially when a specific use case is considered. For this matter, it is crucial to revisit the use case and make explicit essential information regarding that receipt, which matches the use case at hand. Once the explicit keywords for the use case have been decided, the machine learning models can build the classifier model. This task aims to extract relevant

keywords for the use case accurately and then match the extracted keywords' content and category to match the ground truth. For example, the key field 'total' is paired with numbers that describe the correct payment amount, and ' merchants name' matches the data accordingly.

Machine learning techniques, namely, k-nearest neighbours, random forests, and Naïve Bayes, are proposed in solving text classification for post-OCR parsing (Hadi et al., 2018; Xu, 2018; Jiang et al., 2016; Singh et al., 2019; Sun et al., 2020; Hazra et al., 2017). Machine learning (ML) aims to allow software programs to learn from experience. Text classification aims to learn an *objective function* from a set of a function called the *hypothesis space*. The hypothesis space maps text to suitable categories. A set of pre-classified documents is essential for a classifier to learn an objective function. A human operator is required to mark up this set, and this is the only human input needed to operate a classifier. The learning and subsequent classification which come next can be done automatically. It is also possible to learn from other document sets that have not been classified, termed *unsupervised learning*. However, for text classification, the performance is reportedly significant with supervised learning.

1.3 Problem statement

According to ICDAR2019, text detection and recognition in parsing receipt OCR are much less studied than other popular OCR tasks such as name card recognition, license plate recognition, and handwritten text recognition. Receipt OCR in general also requires a much higher accuracy to meet expense reporting requirements. A study for post-OCR parsing of receipts is also scarce, opening the opportunity to explore extracting key information from receipts. This study interested in exploring how the OCR and ML techniques can optimise and automate receipt handling for reimbursement purposes in this work.

1.4 Research questions

It is crucial to organize and direct the dissertation to meet the problem statement. Therefore, four research questions are proposed in shaping the work. They pay attention to discriminators of quality scanning, parsing methods, what information to look for and how to extract from the receipt, and what machine learning techniques can classify receipts. The research questions are as follows:

1. What are the discriminators that distinct a quality scanned receipt?
2. How can text detection and recognition be improved for parsing OCR receipts?
3. What is the key information relevant to extract for a reimbursement use case?
4. How do statistical ML techniques perform in classifying receipt categories?

1.5 Objectives of the study

Receipt parsing is central to automating reimbursement processing, and its performance is heavily influenced by the pre-receipt parsing quality and the receipt classification performance post-parsing. Different requirements require specific tools and techniques to address the research questions at each of the three stages. Therefore, the objectives must direct proper framework and model selection to address the research challenges. The following objectives are determined for the study:

1. To develop an image processing framework in improving receipt quality pre-parsing.
2. To recognize text and extract key information from receipt using the OCR technique.
3. To train and test ML classifiers in improving receipt classification performance post-parsing.

1.6 Scopes of the study

Malaysian businesses manage employee travel claims by manually inputting data from receipts into a reimbursement report sheet. The reimbursement report sheet usually comprises several categories, such as mode of transport, fuel, meals, accommodation, and more. The receipts to complement expense claims can be of different formats and structures. The meaningful data from receipts are not always in the typeset text; they can be handwritten. Handwriting characteristics are different compared to typeset text. Handwritten text is often characterized by line quality, alignment, size, spacing, connecting strokes, pen lifts, pen pressure and angle (Chaudhari & Thakkar, 2019).

The varied and ambiguous styling from person to person makes extracting handwritten text on receipt challenging. Therefore, handwritten receipts are out of the scope of this dissertation. However, this dissertation considers receipts with the poor quality of the source document/image due to degradation over time. Briefly, the following outlines the scopes of this study:

1. The set of receipts will consist of 100 receipts
2. Six classification categories proposed are meals, groceries, petrol, accommodation, telecommunication and fares. The classification categories must be expected in the employee expense reimbursement report in Malaysia.
3. The set of receipts will be of Malaysian origin.
4. The language of receipts is restricted to containing English text.
5. Not able to parse handwriting on receipt
6. Text ambiguity is not addressed; the work is limited by the accuracy of the OCR tool selected.

1.7 Research Mapping

Table 1.2 shows the mapping between research questions, research objectives, research methodology, and the research outcome:

Table 1.2: Mapping of research questions, objectives, methods and outcomes

Research Questions	Research Objectives	Methodology	Research Outcome
What are the discriminators that distinct a quality scanned receipt?	To determine relevant image preprocessing techniques to increase the accuracy of text extraction.	a) Literature search b) Systematic literature review	a) Core criteria for quality scanned receipt b) The metrics to quantise individual criteria into measurable data
How can text detection and recognition be improved for parsing OCR receipts?	To develop an image processing framework in improving receipt quality pre-parsing.	a) Framework development	a) An image processing framework to improve receipt quality
What is the key information relevant to extract for a reimbursement use case?	To recognize text and extract key information from receipt using the OCR technique.	a) Framework development	a) An OCR framework to recognize text and extract key information from receipts
How do statistical ML techniques perform in classifying receipt categories?	To train and test ML classifiers in improving receipt classification performance post-parsing.	a) Experiment b) Validating and fine-tuning	a) Results of performance on the accuracy of text classification

1.8 Significance or Impact of the Study

ICDAR2019 outlines SROIE as the critical task in rationalizing document-intensive practices and business automation. SROIE has huge commercial potentials drawing massive attention from big analytic players like Google, Baidu and Alibaba. Detecting text and OCR are two vital processes of the SROIE task in extracting key information from scanned receipts. The biggest challenge in text detection and OCR is the different document structures and the probability of text ambiguity of the key information. The rising trend of combining OCR and ML techniques can boost SROIE research and development in real office applications, such as automating the reimbursement process and expediting expense auditing. The utility of SROIE can open up opportunities for businesses to adopt digital transformation faster.

1.9 Dissertation organization

This dissertation is divided into five chapters. Chapter 1 introduces the research topic, including the background and motivation, problem statement, research questions, objectives and scopes. Chapter 2 provides a critical overview of the literature for topics of interest in this dissertation. Learning from other researchers on state of art in combining OCR and ML techniques is central to the chapter. The methodology adopted in this dissertation is presented in Chapter 3. Included in the description are the proposed research framework and the overall process flow of the research. Chapter 4 presents an analysis of the performance of the proposed method, while Chapter 5 forward the final remark of the work done, concluding the dissertation.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

This chapter deals with the literature review related to this research. Overviews of expenses reimbursement and automated document processing are discussed. Then, some existing receipt imaging software such as Rydoo, Certify Expenses and Expensify are reviewed. Next, the most common OCR tools, such as Tesseract, ABBBY FineReader and Google Cloud Vision that uses images to recognise text, are discussed. The literature review pays special attention to text classification techniques like feature extraction and discusses popular text classifiers algorithms. A comparison between OCR tools and text extraction performance is also provided.

2.2 Expense reimbursement

Expense reimbursement is one of the critical elements in business process models. It provides a reasonable and timely mechanism for employees to compensate for approved travel, living, registration, and other business-related expenses. Expense reimbursement is part of the company's responsibility for repaying employees who have spent their own money on business-related expenses. The received expenses reimbursement is not counted as an employee's wage or income.

Company policies and employment contracts play essential roles in maintaining trust between employees and the organization. The limitation of spending has been mentioned in company policies, so employees are clear on travel allocation, booking, and purchasing and process approval more efficiently (Tripathy & Moorthy, 2020). The expenses reimbursement process may differ for each company depending on their policy, but the people involved in the reimbursement process are usually similar. Based on Arigliano et

al. (2011), employees and administration departments are the main parties in the reimbursement process.

The primary responsibility of employees is to send reimbursement requests with support documents. In contrast, the administration department is responsible for checking document validity, and expenses limit for the employee before executing the reimbursement. Some companies have involved the employee's reporting lines, such as the supervisor or manager, who must review and approve the reimbursement application before sending the request to the administration department.

Some small and medium-sized enterprises (SMEs) manually execute their reimbursement process by asking employees to fill out a monthly expense form (Maslova et al., 2019). The monthly expense form is submitted to the administration department for approval. Supporting documents such as original receipts and bills are submitted together with the application. Usually, the administrative personnel will key in data into a software database to record the expenses details. Error is unavoidable during manual data entry, which can cause trouble verifying expenses details, and sometimes the original receipts can get lost. The reimbursement form that the employee filled up may also be incorrect due to typos or personal fault. Consequently, reviewing the completeness of the reimbursement form and verifying all supporting documents can be time-consuming and costly (Zhu et al., 2007; Saldivar et al., 2016).

Although some applications and business process models have been proposed to increase the efficiency of the reimbursement process, there are only 70% of all reimbursement requests have been processed on time (Saldivar et al., 2016). Fraudulent expense reporting behaviour may happen during the expense reporting phase. Relationships between

employees, supervisors, and the company can increase the risk of fraud, and system security may provide an opportunity for fraud behaviour. Fraud behaviour can threaten the company's bottom line (or net income) and employee morale (Peng & Ford, 2014). Without a software system or application to monitor and manage, the manual submission method may pose some problems to the reimbursement process. Potentially, the reimbursement process can become slow, inefficient, or result in communication errors between the handling parties delaying claim update status to employees (Palawancha, 2012). Although company policies usually include clear and comprehensive guidelines regarding the reimbursement processes and well-defined roles and responsibilities for the officers in charge, employees still find their claims coming in slowly (Triparty & Moorthy, 2020).

To summarize, there are three main stages in the manual reimbursement process. First, the employee must fill-up the expense reimbursement form during the submission of the reimbursement application. Second, the administrator has to review the completeness of the reimbursement form and verify the support document manually. Third, the approver has to approve the reimbursement expenses. Figure 2.1 shows an outline of the manual reimbursement process.

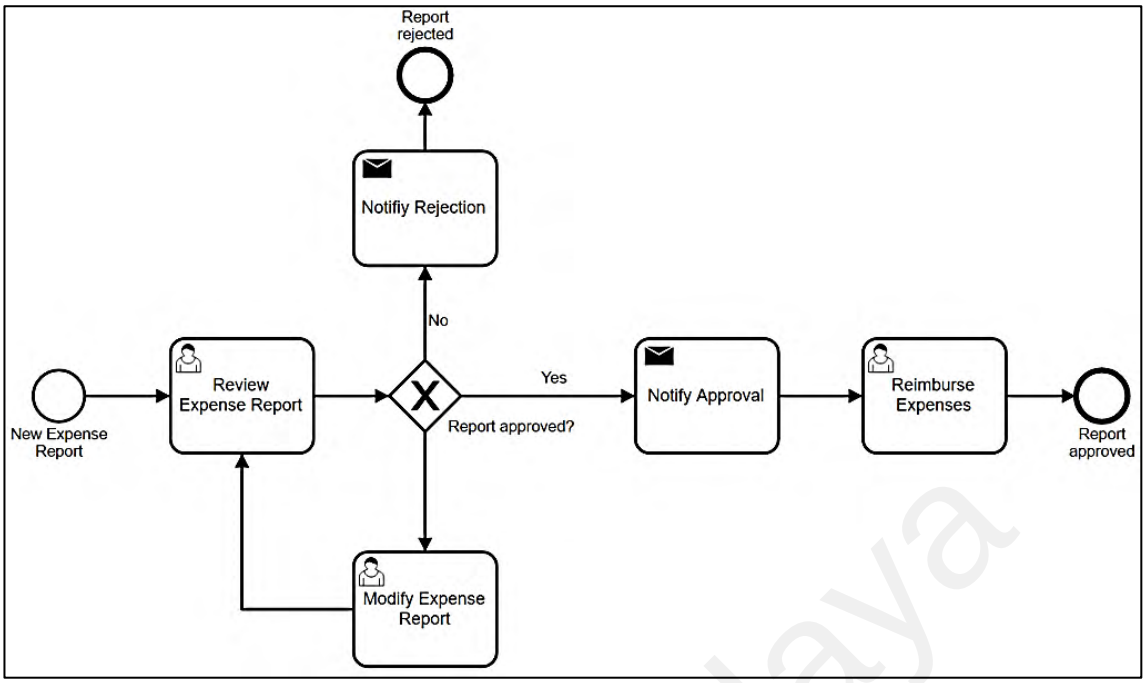


Figure 2.1: General flow of the reimbursement process. Taken from https://www.flokzu.com/blog/en/process-templates/expense-reimbursement/

2.3 Automated document processing

Central to the reimbursement process is document processing. Automated document processing involves, in the beginning, extracting specific text content from files and populating the data into a digital medium (Torres, 2017). The data is then sorted and organised, following classification and clustering rules. Finally, a decision about a specific task is made based on the classification outcome. Automated document processing sometimes considers generating a final report for auditing purposes. Auditing employees expense claims has been a manual task for many companies and often involves many entities fulfilling auditing requirements. However, many organizations are limited in audit expense reports because examining incoming receipts and judging their accuracy requires dedicated auditors (Zhu et al., 2007; Triparthy & Moothy, 2020).

Recently, many companies are looking into technologies to automate the reimbursement process. The aim is to complete the reimbursement process flow with minimum time and less personnel involved. Saldivar et al. (2016) analyzed business process flow and recommended software tools are the way forward in automatic expense validation with minimum human interaction. The characteristics of the software tools must feature text recognition from image-based sources, extract critical information and organise them into itemized categories and classes, and match the correct cost to each item.

Optical Character Recognition (OCR) gets famous for text extraction as it can transform scanned images into text formats that software can comprehend. The OCR can recognize handwritten scripts and printed texts, but the accuracy is directly dependent on the input documents quality. It is not possible to achieve 100% accuracy, but it can help to increase the business process (Dhiman & Singh, 2013). Many different OCR tools were available in the market, such as Tesseract, ABBYY FineReader, and CuneiForm. Their accuracy rate varies from 71% to 98%, and only a few of them are free, open-source software packages (Patel et al., 2012). The OCR has been widely used in the banking and legal industry for cheque scanning and clearance (Srivastava et al., 2019; Jha et al., 2019; Dhanawade et al., 2020). It has five main components to recognise text: pre-processing, scanning, segmentation, feature extraction, and recognition (Mithe et al., 2013).

Text classification is critical in content management, contextual search, opinion polling, product review analysis, spam cleaning, and text sentiment mining. The efficiency of classification and retrieval of relevant content is essential to overcome information overload. There are four steps involved in text classification. First, data preprocessing is proposed to control the size of the input text document. The data preprocessing includes sub-steps to shorten the sentence boundary by filtering stop-words and stem words to

their base form. Second, feature extraction and selection are recommended to identify attributes usually represented by significant words or the frequency of phrases present in the text document. Third, performing text classification using machine learning models. Naïve Bayes, support vector machine and k-nearest neighbours are among the most popular algorithms used for text classification. The final step of text classification is most important, i.e., model training and classifier testing (Xie & Bailey, 2020). A classifier model is usually the outcome of text classification.

2.4 Image preprocessing

According to ICDAR2019, factors that determine the quality of a receipt pre-parsing include sharp character borders, high contrast, well-aligned characters, and as little pixel noise as possible. Korobacz & Tabedzki (2018) propose that image preprocessing techniques such as scaling, contrast enhancement, binarization, denoising, skew correction and layout analysis are usually considered to improve receipt quality.

2.4.1 Scaling of image

OCR engines are now complemented with guidelines concerning input image quality and the relation to its size. Image scaling modifies the proportion of a digital image, so its resolution is redefined to a new value. Resizing an image has several advantages to its visual appearance through changing the quantity of information in an image (Johansson, 2019). For this reason, scaling becomes standard practice in early multi-stage image processing whenever the consequent processing pipeline requires the image at a specific scale. A 300 DPI (dots per inch) has been recommended as the optimised scaling size for OCR purposes. Scaling higher than the intended DPI does not mean the receipt quality gets better. For example, scaling up an image at 600 DPI and above only adds up the size

but is stagnant at image quality. Meanwhile, scaling down to 200 DPI and lower reduces size and image information resulting in an unclear and incomprehensible output.

Interpolation is the method used to resize or distort an image. The method takes advantage of a known pixel point to predict where it can add new pixel points. The approximation allows increment of pixel numbers, and more importantly, contributing to overall pixel distribution for the image. The challenge for interpolation is projecting correctly the location of new pixel points given a known point that enhances the appearance of the foreground information, text included. The projection usually works in 1-D and 2-D directions and is heavily influenced by the positions of existing neighbouring pixels. Figure 2.1 shows 1-D and 2-D projections examples based on a known pixel point.

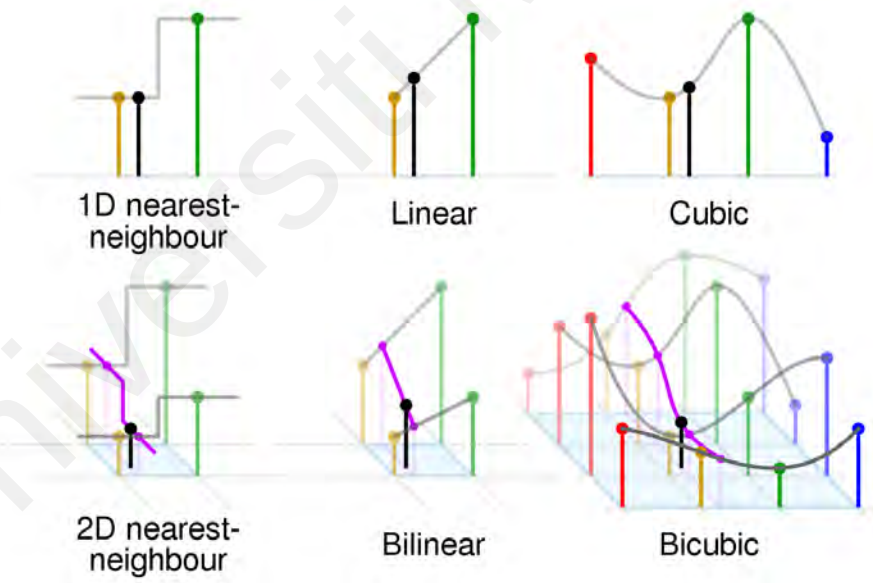


Figure 2.2: Example of projections for 1-D and 2-D interpolation. Taken from https://commons.wikimedia.org/wiki/File:Comparison_of_1D_and_2D_interpolation.svg

Several popular algorithms for interpolation differs by how they manage the projection. The simplest projection method is called the Nearest Neighbour interpolation. The method copies known pixel information and duplicates it at several locations surrounding the original pixel (Aizan et al., 2016; Li & Luo, 2011). Thus, giving the original pixel a

more prominent appearance visually. A 200% resize can make one pixel grow into a 2 x 2 area of four pixels. Copying the pixel information means copying its colour value, resulting in *aliasing*. Aliasing makes edges seem jagged, especially at curves. The image can seem distorted upon close examination.

Bilinear interpolation observes the four pixels closest to a goal (known pixel) and takes their weighted average, and copies that average to the nearest 2 x 2 neighbours of the goal. Taking the average colour value has an anti-aliasing impact which significantly reduces distortion to the edges. Edges and curves appear smooth even up close (Sakila & Vijayarani, 2017; Wang & Yang, 2008).

Following the performance of bilinear interpolation, researchers recommend bicubic interpolation towards smoother edges output. Instead of looking at 2 x 2 neighbours, the bicubic proposed using sixteen pixels in the nearest 4 x 4 neighbourhood of the goal (pixel). Such projection produces excellent resampling images; the method becomes widely adopted in commercial digital imaging and editing software (Huang & Cao, 2020; Li et al., 2019; Rangsikunpum et al., 2017). Figure 2.3 shows the difference interpolation results.

Figure 2.3: Interpolation methods result performed on the same image. Taken from https://slideplayer.com/slide/9248005/

2.4.2 Increase in contrast

The primary reason for applying image contrast is to enhance the foreground (text) density compared to the background. Increasing contrast means increasing the difference in visual properties that makes an object distinguishable from the background. Such enhancement is a priority, especially when the physical condition of the receipt has deteriorated. Exposure to the sun or wear and tear is usually damaging to the image intensity across the image. There are two steps to contrast enhancement; contrast stretch and tonal enhancements. The former uniformly increases brightness differences across the image, and the latter improves specific brightness at dark, grey or bright regions.

The base of contrast enhancement begins with the grey-level histogram (Arici et al., 2009). The grey-level histogram finds the number of grey-level values pixels over the total pixels in the image and charting their percentage distributions for the image. Figure 2.4 shows a histogram with several peaks indicating the distribution of grey-level value percentages across the image. The percentage distributions show the image's relative brightness and darkness, which can be manipulated, so their appearance becomes more pronounced. Reassigning the grey-level values higher or lower using a mapping function can affect the tone and contrast of the objects. Figure 2.5 shows the transformation, where the mapping function reassigns the grey-level to new values.

Grey-level values span from 0 to 255. Any image that is high in contrast will show a histogram that has grey level values across the spectrum, i.e., a higher standard deviation. An image that is low in contrast, on the other hand, will not. However, one can *stretch* the low-contrast image's grey values to fit the entire spectrum. Contrast stretching remaps the entire grey values of low-contrast images to span full range on the histogram, making it appear high-contrast (Sangwine & Horne, 2012). A linear transform performs the most

straightforward dynamic range adjustment by stretching the lowest grey value to zero and the highest grey value to 255. Consequently, the histogram profile retains its shape, but the shape gets stretched to fill the spectrum. The expansion creates evenly distributed gaps between the grey-level values increasing the contrast and tone of the image. Figure 2.6 shows an example of an image stretching on a low-contrast image. See the evenly distributed gaps between the grey-level values.

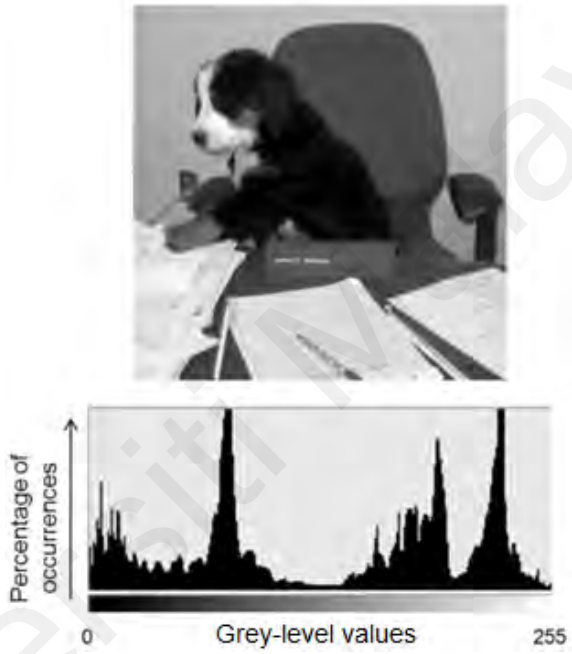


Figure 2.4: Grey-level histogram with peaks of grey-level values. Taken from https://spie.org/samples/TT92.pdf.

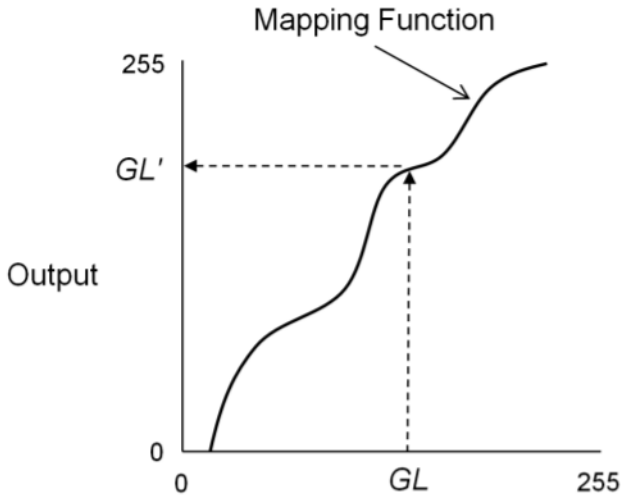


Figure 2.5: Reassigning grey-level (GL) values to new ones (GL') using a mapping function. Taken from https://spie.org/samples/TT92.pdf.

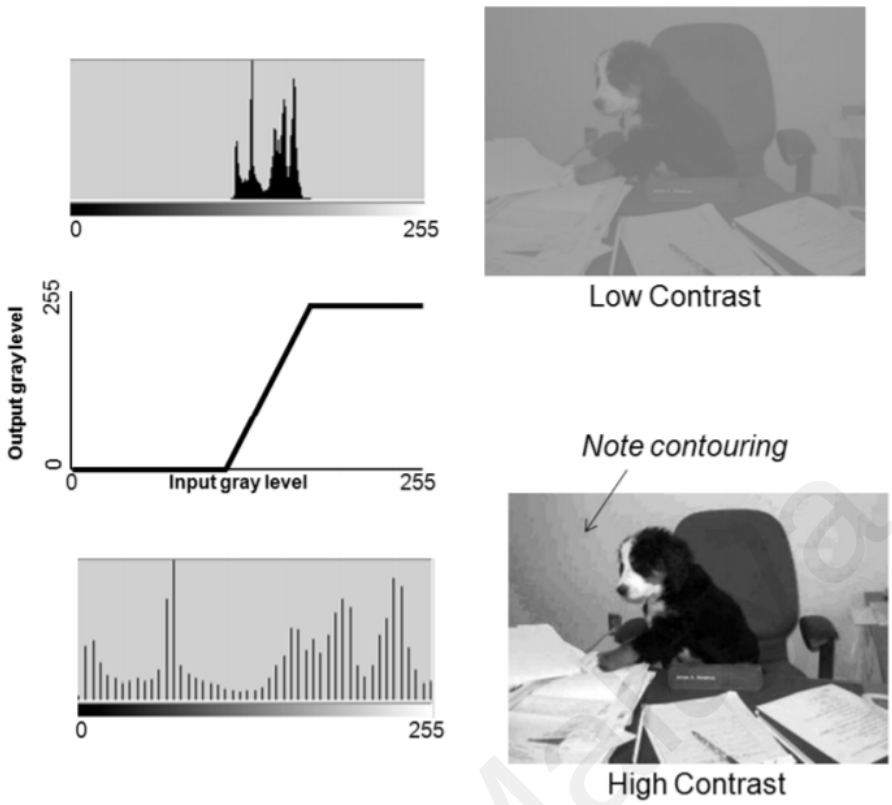


Figure 2.6: A low-contrast image stretched with a linear transformation resulting in a high-contrast image. Taken from https://spie.org/samples/TT92.pdf.

2.4.3 Binarization

This step further improves the clarity of characters in the image by keeping the image black and white (monochrome). It usually includes converting from colour (RGB) to grayscale to monochrome. A straightforward method to perform binarization is through thresholding. A threshold functions to filter the foreground and background according to pixel intensities resulting in a binary grey-scale image. The binary image has pixels with intensities lower than a specific threshold reassigned to zero, while those that pass are reassigned to one. On documents, flawed thresholding methods can mean blotches, deformed text and even erased text.

There are many binarization methods, and one of the most popular is Otsu's thresholding (Otsu, 1979). It is a clustering-type method based on the bimodal histogram profile of an image. Otsu's method assumes that the bimodal profile represents the foreground on one peak and the background on the other. The algorithm transforms the grey-level image to a binary and calculates the optimum threshold separating the two-pixel intensities to minimize their inter-class deviation (Yousefi, 2011). The algorithm continues to search for a threshold aiming at minimizing the inter-class deviation. Otsu's method performs best when the pixels belonging to each cluster are close to one another. Otsu's method has inspired many improvisations over the years (Saddami et al., 2019; Mansoor & Olson, 2019; Latib et al., 2021). Figure 2.7 shows the effect of Otsu's thresholding on the non-bimodal and bimodal histogram profiles.

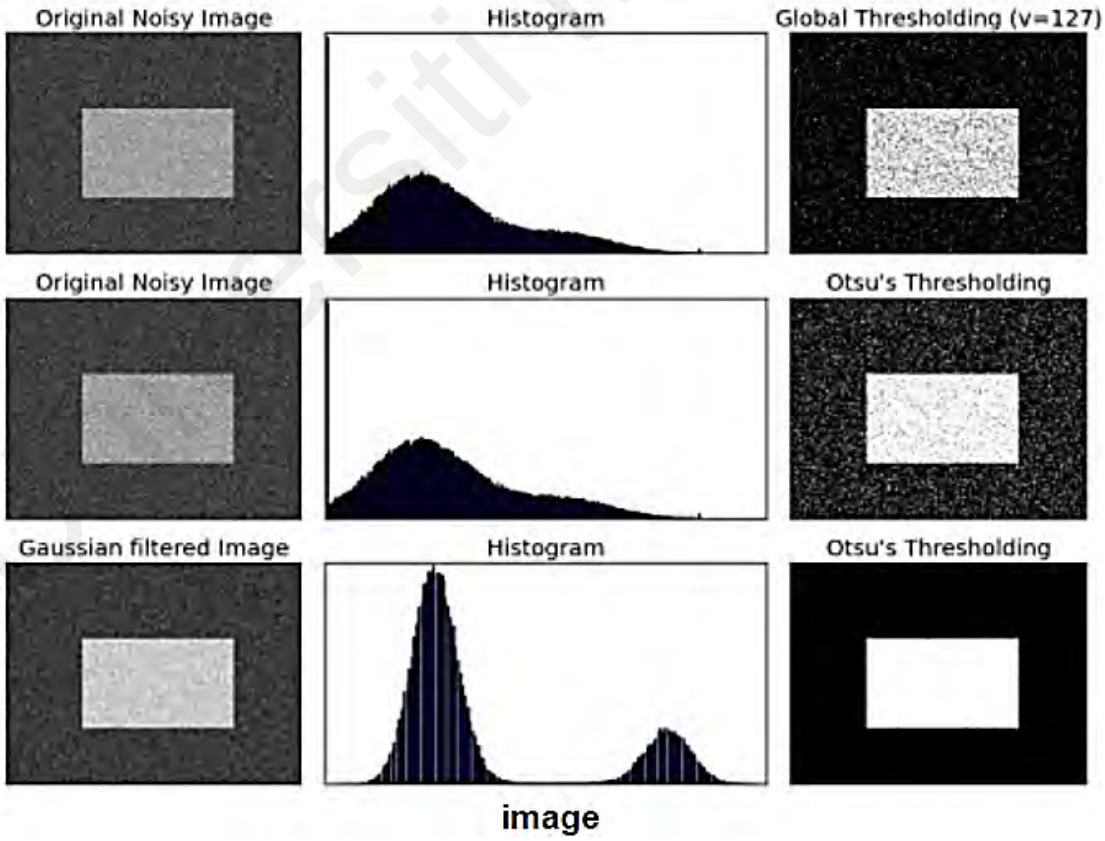


Figure 2.7: Otsu's thresholding on non-bimodal and bimodal images. Taken from https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

2.4.4 Noise and artefact removal

The foreground and background noise can be present if the original scanning is not done right. Noise can be in the form of random colour or brightness at the pixel level of an image. Some of the typical noise include the salt and pepper noise and the Gaussian noise. Image denoising aims to eliminate noise, but it is not easy to remove the artefacts as they can also be parts of the high-frequency components like edge and texture. Retrieving loss information due to denoising is an open challenge (Milanfar, 2012; Jain & Tyagi, 2016; Diwakar & Kumar, 2018). High-quality denoising for documents, according to these researchers, is to remove noise while maintaining (see Figure 2.8 for example):

- the flat region of the image is smooth,
- the edges are protected and sharp,
- the text is intact, and
- no new artefacts appeared.

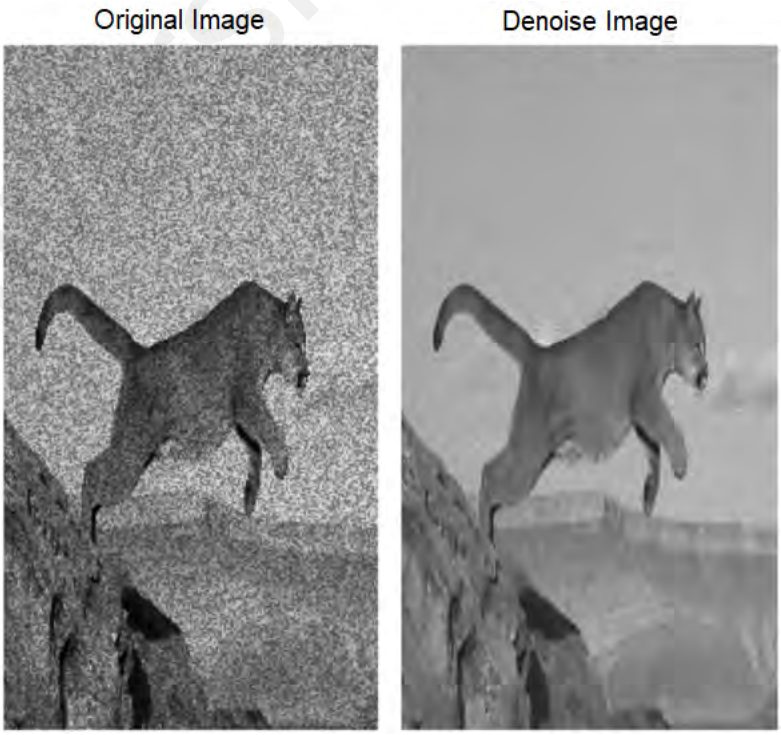


Figure 2.8: Image denoising with wavelet thresholding. Taken from https://www.mathworks.com/help/wavelet/ug/wavelet-denoising.html

The filtering parameter is the critical component in denoising methods. However, at the same time, the parameter has to be an estimation of the noise difference. In other words, the filter has to learn the white noises to safely deletes them. Several classical approaches have shaped researches in this area, including Gaussian smoothing (Feichtinger & Strohmer, 2012), semantic filtering (Yang, 2016), Bayesian neighbourhood filters (Huang, 2015) and wavelet thresholding (Hsia et al., 2015). They show excellent performance in reducing signal to noise ratio, improving the quality of a document image.

2.4.5 Skew detection and correction

Skew is also referred to as rotation. Deskewing means rotating the document image clockwise or anti-clockwise to bring the text to the correct format and shape. For example, the text should appear horizontal and not tilted at any angle. Deskewing includes detecting the text block with a skew in the image, calculating the rotation angle, and finally, rotating the image to correct the skew. Figure 2.9 shows an example of skew correction. Skew detection is performed first. Conceptually, skew detection projects skewed pages at several angles and compare the number of black pixels (texts) per projected line. The angle with the slightest difference to the ground truth (normal text alignment on paper) has the highest probability of correcting the skew.

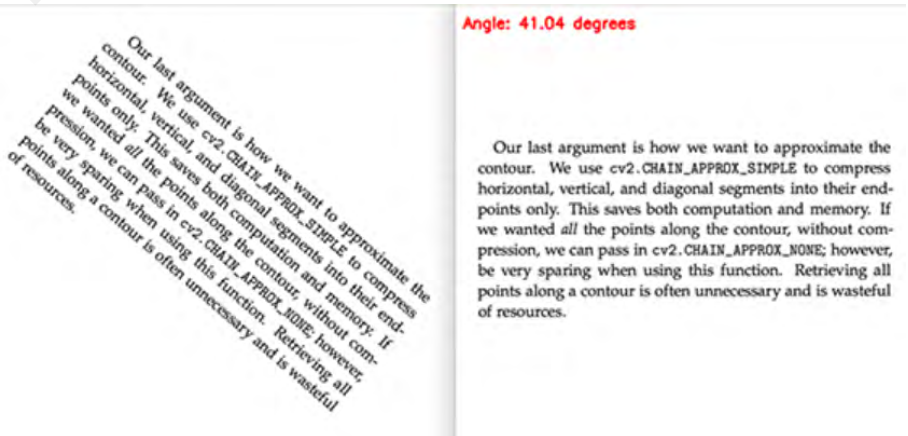


Figure 2.9: Deskewing detection and correction example. Taken from https://www.pyimagesearch.com/2017/02/20/text-skew-correction-opencv-python/

2.4.6 Layout analysis

The previous step, deskewing, deals with bringing the document image to the correct *shape* where the text appears horizontal and not tilted at any angle. The next stage is to locate where the text layout is on the document. The layout analysis is performed, referring to any possible interesting areas that contain text for extraction. The areas of interest on document images can be paragraphs, tables, columns, and captions inside the image. If the program misses any layout or zone, words may be cut off and fail the detection. Ideally, the layout analysis should output the text region of interest with bounding markers or blobs.

Locating text in the document begins by searching for in-between characters or words spaces. Connecting the nearest neighbours using linear regression should produce line segments. Next is to search for vertical spaces between the line segments. If the vertical distance passes the line stacking threshold, the lines are marked as properties of the same text block. Bounding boxes can be computed to represent the groups of line segments. The document layout analysis is completed when the bounding boxes separate the text groups. Figure 2.10 shows an example of a bounded text block.

Figure 2.10: Bounding text for document layout analysis. Taken from https://nanonets.com/blog/ocr-with-tesseract/

It is interesting to note that open-source image processing libraries such as OpenCV with bindings for C++, C, Python, and JAVA can be utilised towards computational efficiency in a real-time environment. For binarization, the OpenCV has libraries to perform Simple Thresholding, Adaptive Thresholding, and the Otsu's Binarization. The Python community has also proposed deep learning-based for the layout or zone analysis, such as the EAST text detector initiative (Rosebrock, 2018).

2.5 Receipt imaging software

Today, there are many receipt imaging software available. This receipt imaging software allows employees to scan receipts for more accessible storage in the long term. Some software is also given mobile access, so employees have the option to scan on the go. Examples of such software include Rydoo, Certify Expenses, Expensify, Wave, Taggun and Shoeboxed (Maslova et al., 2019). Some of this software, such as Rydoo, allows integration with online report sheets.

However, there is still a black box behind with some limitations. They are usually limited to general information extraction, such as total transaction amount and date, but do not extract the details of purchased items. Extracting necessary detail is a difficult task, especially in ensuring the accuracy of extracted information. Besides that, most of them required an extended processing time and sometimes even as long as several minutes. Table 2.1 shows the list of existing receipt parsing applications with their key features and limitations.

Table 2.1: List of existing receipt parsing applications

Application	Web / Mobile based	Key Feature	Parsing Technique	OCR Parsing Accuracy	Limitation
Rydoo	Mobile	Manage expenses in real-time and allow integrated with online report sheet.	OCR	High	Limited information extracted
Certify Expenses	Mobile	It is integrated with other solutions and an online report sheet.	OCR	High	
Expensify	Mobile	It has an automatic approval workflow and accounting sync.	OCR	High	Long processing time and limited information extracted
Wave	Mobile	Able to process submission without internet.	OCR	Medium	
Taggun	Web	Short processing time	OCR	82%	Limited information extracted
Shoeboxed	Mobile	Data is verified twice by OCR and humans.	OCR/ Manual verification	Medium	

2.6 Optical character recognition

The application of Tesseract OCR as character blocks extractors has become widespread in document processing. Figure 2.11 depicts the design and modules flow of the Tesseract OCR. The flow begins with scanning an image that may be grey-scaled or coloured, converted into a binary image by an adaptive thresholding algorithm. In the second step, line segments are extracted and organised into text lines by connected component analysis. Then, text lines are broken into words by the helping of fuzzy spaces. Finally, two passes of recognition are tried to maximise the word recognised and text extracted from the image after resolving fuzzy space (Patel et al., 2012; Brisinello et al., 2017; Mithe et al., 2013).

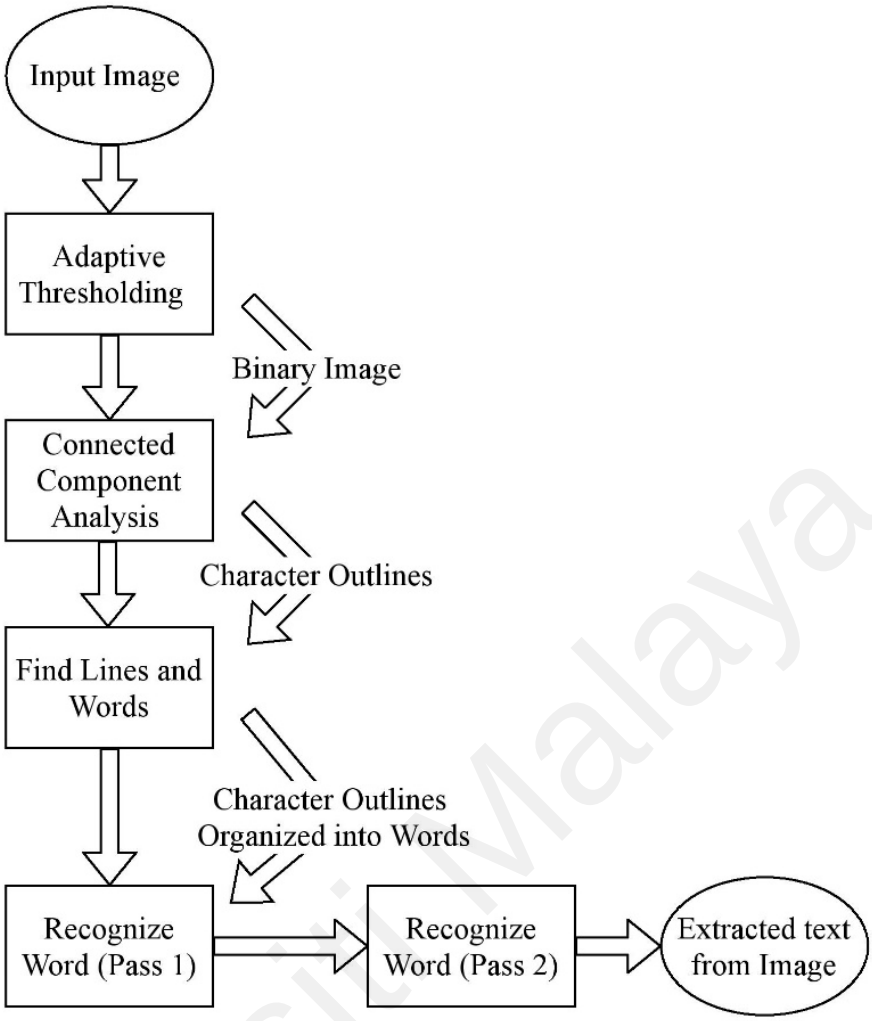


Figure 2.11: Tesseract OCR flow system (Patel et al., 2012)

ABBBY FineReader is an OCR application developed by ABBYY. Three numerical classifiers based on different numerical attributes, raster, Omnifont and the contour classifier, makes up ABBYY. The recognition process is terminated if the level of confidence is reached. The training sample is clustered separately and organised by using the nearest neighbour classification. The calculation of Mahalanobis distance provides the distance between test set objects and the training set clusters, producing a hypothesis list with a good confidence level. After the classifiers complete the classification process, the differential classifier rearranges the hypothesis list using modified bubble sort to compare the hypothesis with their linear classifier (Itskovich & Itskovich, 2011). Figure 2.12 shows the ABBYY OCR recognition schema.

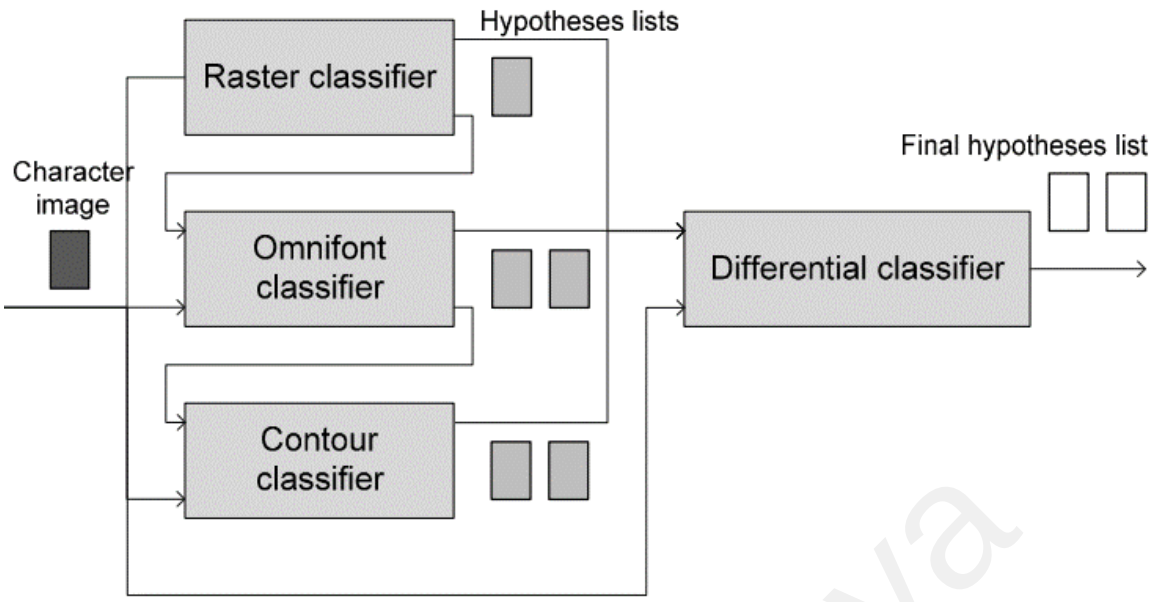


Figure 2.12: ABBYY OCR recognition schema (Itskovich & Itskovich, 2011)

For many years, the document processing community favours the Google Cloud Vision to analyse and retrieve the document image's content and main features. The application programming interface (API) called the Google Cloud Vision API can be used in pre-trained models or AutoML Vision in building custom models. The API is accessible remotely in processing images' content and extract visual data such as handwriting recognition, image attributes, and OCR.

After uploading the image to Google Drive, Google Cloud OCR proceeds with text feature, page layout and font analysis. Then, it proceeds with character segmentation and recognises the word, checking for language, and extracting the text. The Google Cloud OCR scores a 100% output against the Tesseract OCR whenever the testing environment includes multilingual (Johansson, 2019; Vaithiyanathan & Muniraj, 2019). Table 2.2 shows the comparison of OCR tools, advantages and limitations between these tools are compared.

Table 2.2: Comparison of OCR tools

OCR Tools	Sources	Advantages	Limitation
Tesseract	Patel et al., 2012; Brisinello et al., 2017; Mithe et al., 2013	• Open source and free tool • High accuracy on a clean page	• Easy affect by noise
ABBYY	Itskovich & Itskovich, 2011	• High accuracy on noised page	• The only commercial version available • Require more effort to train
Google Cloud Vision	Johansson, 2019; Vaithiyanathan & Muniraj, 2019	• Remotely process the content of images • 100% output in a dynamic environment	• Require internet connection

2.7 Supervised learning in machine learning

Humans have limited capacity in processing data; thus, the recent trend in artificial intelligence and machine learning has boosted research that requires collecting vast amounts of data. The machine learning approach is advantageous in different areas such as data analysis, natural language processing and text classification. The technique is constructed by statistics-based learning methods that thrive in scrutinizing data and classifying them. Due to its statistical nature, machine learning is data-hungry. Therefore, a database or dictionary with relevant data is required to feed the algorithms. Otherwise, the technique cannot show desirable data classification performance.

Machine learning is commonly divided into supervised or unsupervised learning. Learning is considered supervised when the classes are already tagged on the data used for training (Lilleberge et al., 2015). The machine is given the data classes at the training stage. Meanwhile, the machine is not provided with the data classes when training data for unsupervised learning (Triparthy et al., 2015). Supervised learning commonly handles regression and classification-type use cases, whereas unsupervised learning works well

with clustering problems. Document images contain an abundance of information meaningful to humans. Mining the information and processing them into categories and classes is usually described as a supervised learning task. For example, Lilleberg et al. (2015) show that supervised learning can identify receipts accurately following the probability suggested by the data that completed training with specific corpus labelling.

Since text classification models can only be processed in numerical values, some natural language processing (NLP) techniques are used to retrieve information from the document and convert text to numbers before classification (Triparthy et al., 2016; Kowsari et al., 2019). Pranckevičius & Marcinkevičius (2017) show the efficiency of their feature vectors in converting words to integer values and counting the word frequency. Scikit-learn has the library required to vector and tokenize words for data cleaning and analysis (Hackeling, 2017; Pedregosa et al., 2011). This free and open-sourced machine learning software library also has several pre-trained models for text classification. Some of the popular classifier models for classification are the Naïve Bayes and Support Vector Machine.

CountVectorizer is a word vectorization technique used to convert the text data into a number represented as vectors. It replaces words as the vector-based frequency of occurrence of each word. Besides that, Word2Vec is a new method introduced by Google that brings extra semantic features that help in text classification. It is trained by a neural network to find the similarity vector by considering the frequency of each word (Lilleberg et al., 2015). CountVectorizer is only suitable for text without semantic meaning, while Word2Vec can handle text with semantic meaning (Bogach & Kovenko, 2020).

Continuous bag-of-words (CBOW) and Term Frequency–Inverse Document Frequency (TF-IDF) are the most basic word vectorization techniques that map words from a corpus dictionary to an equivalent real numbers vector. CBOW can convert text to numbers by disregarding grammar and word order, but it keeps the multiplicity. It does not take into account the fact that words frequently appear in documents. However, TF-IDF combined Term Frequency and the Inverse Document Frequency to count precisely the regular words in a document and cut their frequency down (Lilleberg et al., 2015).

A classification model can be produced by feeding it with training data consisting of feature sets pairs into a text classifier. Different text classifiers have different algorithms and characteristics. The classifiers have the highest chance to produce a more accurate prediction if given more training samples. Furthermore, the Naïve Bayes and Support Vector Machine classifiers are often considered straightforward, considering their many applications. This section discusses the Naïve Bayes, SVM and several other classifiers.

2.7.1 Naïve Bayes

The Bayes' probability rules define the Naïve Bayes classifier decision making. The rules determine the likeliness of what is about to happen based on prior facts about the situation. In the context of the text document, the probabilistic classification calculates several training data set, and the probability is tested with the testing data set. The presence or absence of features in the text document can affect the classifier's probability. After completing the probabilities derivation, the new text document can be classified following the probabilities score of each feature category.

The statistical nature of the Naïve Bayes classifier can be implemented quickly and is attractive when there is little time and data for training. However, the classifier's performance is reportedly less efficient when it comes to high dimensional data. The features were less than other classifiers (Dalal & Zaveri, 2011; Khairnar & Kinikar., 2013; Patra & Singh, 2013; Tripathy et al.,2015; Singh et al., 2017). This supervised learning classifier will assume all features occurred independently from each other. The naïve Bayes classifier was based on Bayes' rule that is defined in Equation (1).

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \quad (1)$$

In this equation, 'c' represents the class, and 'd' represent features. $P(d)$ is the probability of picking randomly a text document containing features, whereas $P(c)$ is the probability of picking randomly a text document belonging to a related class. $P(c|d)$ is the conditional probability for each category. Since the number of possible text documents is high, calculating $P(d|c)$ is not straightforward. Therefore, the term $P(d|c)$ can be decomposed by assuming conditional independence of features fi's given d's class with Equation (2).

$$P_{NB}(c|d) = \frac{P(c)(\prod_{i=1}^{m}(f_i)|c)^{n_i d}}{P(d)} \quad (2)$$

Naïve Bayes classifier has four main steps. First, keywords in the text document are checked and stored in a map. Second, the frequency of yes and no of each keyword in the text document is counted. Third, the probability of each keyword of the text document is computed using Equation (2). Finally, a text document is classified into various classes based on the probability calculated (Bijalwan et al., 2014; Sharma & Dey, 2012).

2.7.2 Maximum entropy

Maximum entropy (ME) is another probabilistic classifier. The supervised learning method is controlled by a constraint calculated from training data using conditional probability. In other words, the characteristics of training data have been represented by constraints. Conditional probability *(P(c|d))* can be defined as Equation (3).

$$P_{ME}(c|d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right) \quad (3)$$

Where $Z(d)$ is a normalizing factor, $F_{i,c}$ is the feature or class function for feature f_i and class c is defined in Equation (4).

$$F_{i,c}(d,c') = \begin{cases} 1 \; n_i(d) > 0 \; and \; c' = c \\ 0 \; otherwise \end{cases} \quad (4)$$

The ME algorithm has a better performance compared to Naïve Bayes because the former assumes the dependence of features, whereas the Naïve Bayes assumed the independence of features. Therefore, if a keyword frequently occurred in a class, the weight will be increased, and the probability of the class selected will be higher (Sharma & Dey, 2012; Khairnar & Kinikar., 2013; Tripathy et al., 2016).

2.7.3 Support vector machine

Another supervised learning method is the support vector machine (SVM). The SVM classifier is based on non-probabilistic binary linear strategies that are advantageous in classifying linear and non-linear data. Non-linear formulization has been used in converting training data into a high dimension diagram. The optimal linear searching, on the other hand, identifies hyper-planes. Vectorised data and the key idea behind the

training model have been analysed thoroughly to get the maximum margin of hyper-plane represented by vector w. Hyper-plane separates document vectors between classes, and the separation is proposed to be kept as large as possible (Sharma & Dey, 2012). The equation of vector *w* can be defined as Equation (5).

$$w = \sum_j a_j c_j \, d_j, a_j \geq 0 \tag{5}$$

Where c_j is class (positive or negative) for a document d_j and the value a_j solves the dual optimization problems. All d_j is termed support vectors when a_j is greater than 0 as it was the only document vector contributing to *w* (Khairnar & Kinikar., 2013; Tripathy et al., 2015). After the classifier is trained, the test data will examine the model. The prediction of the class will be determined by which side of the hyperplane the test data falls into. SVM can perform two-class classification problems such as positive or negative effectively and accurately. The classifier can deal with a limited amount of training data to perform the supervised training process. However, SVM performs poorly on multi-class classification problems. The reason is SVM requires input in the form of a vector of numbers; the value of the text document has to convert to a numeric value. When it goes through the scaling process, the classifier has to keep the vectors in range of predefined class (Dalal & Zaveri, 2011; Tripathy et al., 2016). Supervised learning

2.7.4 Linear support vector classifier

Linear support vector classifier (SVC) is a type of SVM which apply linear kernel function. It has more flexibility as additional parameters such as penalty normalization. Compared to SVM, Linear SVC minimises the squared hinge loss and converge quicker on massive amounts of knowledge (Kong et al., 2015). The supervised learning strategy

is mainly used for classification problems. The hyper-plane splits training sets into multiple classes in many different ways. The linear SVC will maximise the margin as wide as possible between the hyperplane and the support vectors. The wider the margin, the more accurate is the classification performance (Odd & Theologou, 2018). However, one limitation of the linear SVC is dealing with extensive data set because the hyperplane is challenging to generate correctly, which can easily cause overfitting.

2.7.5 Decision tree

The decision tree is a recursive partition classifier formed by a node representing an attribute and edge. A tree-based classifier is initialized by a root node with no incoming edge but branched out with nodes connecting precisely to a single incoming edge. All the nodes except the root node are called leaves; they are also known as the terminating or deciding nodes. Nodes with outgoing edges are called the internal or test nodes. They split the instance space into at least two sub-spaces with the discrete function of the input attributes values (Sharma & Dey, 2012). Traversing the decision tree is fast and is widely considered for document categorization (Kowsari et al., 2019).

The greedy strategy builds the supervised learning decision tree to improve overall error on the document training data. The document classification is obtained by examining the tree branches started from the root node following the test data features scores. The decision tree is simple, easy to understand and interpret for its graphical representation. The classifier learns the document training data in no time. Document data can be classified by casting a vote down the tree branches at a fast rate. However, a decision tree is vulnerable to minor variations in the data, causing overfitting. The tree is also hard to manage when dealing with too many attributes (Dalal & Zaveri, 2011; Patra & Singh, 2013, Kong et al., 2015).

2.7.6 Random forest

A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms. Similar to a decision tree, its primary goals are regression and classification problems. The random forest is an outcome of connecting random and unrelated decision trees in one feature space. True to its name, the diversity of the tree influences the random forest performance. During testing, the data passing through all forest trees are given a unit vote to predict the result (see Figure 2.13). The random forest is robust to noise and a fast method to identify non-linear patterns in the data. Handle both the numerical and categorical data is its main strength. Even if more trees are added to the forest, it will not suffer from overfitting (Agrawal et al., 2013; Chaudhary et al., 2016). However, the main issue with having *that* many branches are dimensionality, instigating computing time complexity. Chances to append new correlated trees are also inevitable if features with large weights are selected repeatedly (Ye et al., 2013; Kowsari et al., 2019).

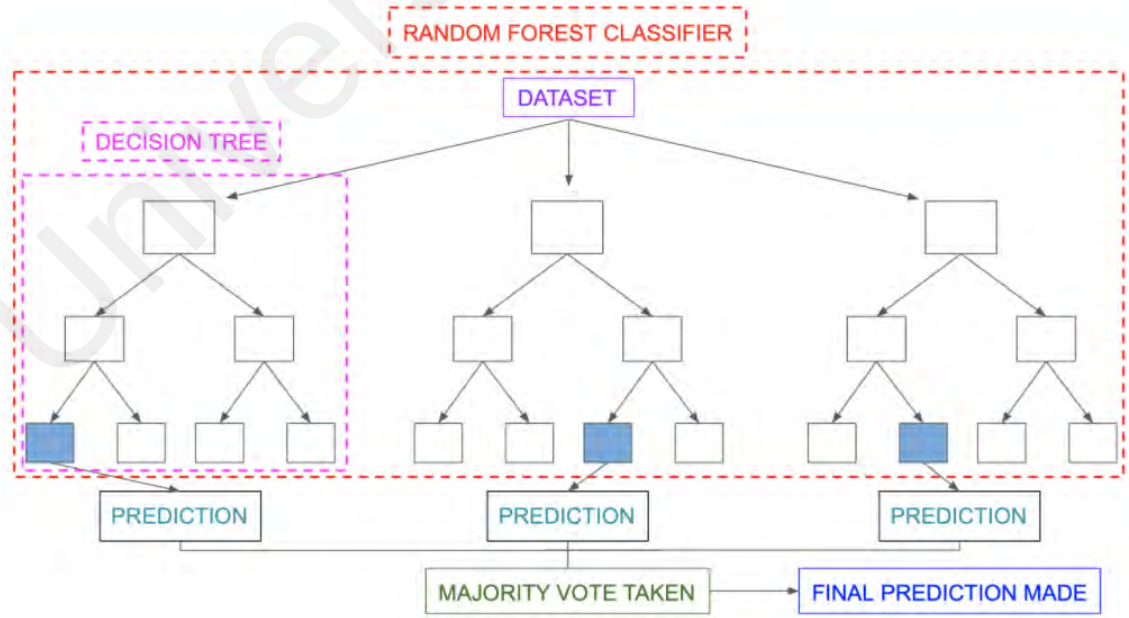


Figure 2.13: A random forest casting votes down the branches for prediction. Taken from https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

2.7.7 K-nearest neighbours

The k-nearest neighbours (KNN) classifier is another supervised learning method suitable for classification and regression problems. The instance-based classifier infers prediction based on the class vector of training and testing data. After the classifier has been trained, test data is classified by the similarity-based vote of its neighbours. The prediction performance relies on voting several labelled training examples with their smallest distance from each object (). Therefore, it will be classified into the class with the smallest distance among its k nearest neighbours where k is a positive integer number. For example, if $k = 1$, the test data will be assigned to the nearest neighbour's class. Else, it will be comparing the distance and class between k nearest neighbour to select the target class. The distance k-nearest neighbours are represented by similarity score, which is defined as Equation (6).

$$Score(d, c_i) = \sum_{d_j \epsilon KNN(d)} sim(d, d_j)\delta(d_j, c_i) \quad (6)$$

Where $KNN(d)$ represents the test data set of k nearest neighbours of document d. The $\delta(d_j, c_i)$ will equal to 1 if d_j belongs to c_i, else it will equal 0. Then, test data will be assigned to the class with the highest resulting similarity score (Sharma & Dey, 2012). There are three main steps of KNN. First, identity vector for every data and select centroid vector for each class. Second, the similarity score between vectors has been calculated. Then, the test data will be classified based on the calculated similarity score (Bijalwan et al., 2014). The advantage of KNN was based on its simplicity and competitiveness for implementing the text classification model. However, similarity score as its primary function in classifying objects required high computing cost and time (Patra & Singh, 2013).

2.7.8 Comparison of text classification algorithm

Table 2.3 compares the text classification algorithms like the Naïve Bayes, maximum entropy, SVM, linear SVC, decision tree, random forest and KNN neighbours, showcasing their advantages and limitations.

Table 2.3: Comparison of text classification algorithms

Text Classification Algorithm	References	Advantages	Limitation
Naïve Bayes	Dalal & Zaveri, 2011; Khairnar & Kinikar., 2013; Patra & Singh, 2013; Tripathy et al.,2015; Singh et al., 2017	• Easy to implement • Less training time	• Inadequate in managing interrelated feature
Maximum Entropy	Sharma & Dey, 2012; Khairnar & Kinikar., 2013; Tripathy et al., 2016	• High efficiency • Short execution time	• Features and classes have a high impact on conditional probability
Support Vector Machines	Sharma & Dey, 2012; Khairnar & Kinikar., 2013; Tripathy et al., 2015; Dalal & Zaveri, 2011; Tripathy et al.,2016	• Perform effectively and accurately • Require small amount of training data	• Challenging to deal with a multi-class classification problem • Slow training speed
Linear Support Vector Classifier	Kong et al., 2015; Odd & Theologou, 2018	• More flexibility as additional parameters • Minimises the squared hinge loss • Converge quicker on massive amounts of knowledge	• Not suitable for large data sets
Decision Tree	Sharma & Dey, 2012; Dalal & Zaveri, 2011; Patra & Singh, 2013	• Simple and easy to understand • Less training time	• The shape of the tree is vulnerable to change even with minor variations of data

			• Challenging to manage a large number of features
Random Forest	Agrawal et al., 2013; Chaudhary et al., 2016; Ye et al., 2013; Kowsari et al., 2019	• Robust to noise and fast method to identify non-linear patterns • Able to handle numerical and categorical data • Not suffer from overfitting	• High time complexity in prediction • Not suitable for high dimensional feature spaces
K-Nearest Neighbors	Sharma & Dey, 2012; Bijalwan et al., 2014; Patra & Singh, 2013	• Simple to compute • Competitiveness for implementation	• High computing cost and time

2.8 Concluding remark

Expense reimbursement is part of the company's operations. Employees need to submit reimbursements requests on time, while administrative staff need to review, and some need to be approved by supervisors. However, only 70% of reimbursement requests can be completed on time. Automated document processing is used to speed up expense reimbursement procedures. OCR is responsible for extracting the text in the receipt. Before the raw receipt is proceeded by OCR, some image preprocessing steps need to be performed to increase receipt quality.

OCR tools such as Tesseract, ABBBY FineReader and Google Cloud Vision convert the text in the receipt into machine-readable text formats. It has five main components to recognise text: scanning, segmentation, preprocessing, feature extraction, and recognition. After that, the OCR tool is processing to extract the key information in the receipt image, such as the merchants' name, date and total price following the reimbursement auditing requirements. The last procedure of document processing is organizing the receipts into the proper categories.

Machine learning is classifying the receipt into different classes. Text classification is an excellent challenge to supervised learning strategies of machine learning. The building blocks of machine learning is statistical learning methods, powerful in analysing data. It can be on supervised learning tasks, especially identifying categories of receipts based on probability suggested by different classifiers. Training data is required to train the classifier, and it must be in the numeric value because the classifier cannot process text value. Word vectorization techniques such as CountVectorizer, Word2Vec, CBOW and TF-IDF of the NLP can convert the text data into vector numbers format.

After the classifier is trained, it can provide predictions based on the input. Different classifiers have different algorithms; Naïve Bayes and maximum entropy are probabilistic-typed classifiers, whereas the SVM and linear SVC are non-probabilistic binary linear classifiers. Each classifier has different characteristics and behaviour; thus, the prediction performance should be different.

2.9 Chapter summary

This chapter reviews the literature relevant to the work following the research questions identified. The following chapter presents the methodology to address the problem statement, exploring the indicators to improve receipt pre-parsing, working through the text detection and recognition in parsing OCR receipt and finally exploring machine learning techniques for receipt classification post-parsing.

CHAPTER 3: METHODOLOGY

3.1 Overview

This chapter describes the research methodology, which helps in achieving the goal and objectives of this dissertation. Data acquisition is discussed at the beginning of the chapter. An overview of the modules developed in this chapter is presented next, covering the considerations taken during pre-receipt parsing, the OCR processing and the post-OCR parsing.

3.2 Data acquisition

The receipt is the primary dataset of this work. The receipt collection is retrieved from two different sources, daily compilation and online databases. A total of 100 receipts are accumulated, where 60% is from the everyday collection and the rest are from the ICDAR2019 database. All the receipts are from Malaysian businesses and divided into six categories: meals, accommodation, petrol, transport, telecommunication, and grocery. All receipts are physical receipts obtained after each purchase and captured by a camera or scanner in the daily collection section. Some merchants will distribute electronic receipts (e-receipt). These receipts are in portable document format (PDF) with colourful content such as logo and advertisement.

Some of the electronic receipts are collected through screenshots of the phone or computer. The merchants' receipts are often received through specific apps or emails. For example, the fare of tolls can only be obtained through Touch & Go's web pages or applications. Although these receipts may affect the performance of OCR, they are not ignored. The 40 physical receipts are downloaded free from the ICDAR2019 database (ICDAR, 2019). Appendix A shows some examples and the GitHub link to download all 100 receipts. Table 3.1 shows the total receipts collected in different categories.

Table 3.1: Total receipts collected in different categories

Category	Daily Collection	ICDAR Database	Total
Screenshot of online receipts (e.g. Grab, Touch n Go, etc.)	11	0	11
PDF/ E-bills (e.g. AirSelangor, TNB, etc.)	19	0	19
Scanned physical receipt (e.g. restaurants, lodgings, etc.)	30	40	70
Total	**60**	**40**	**100**

In addition, training data is needed to train the classifier. A dictionary with 1150 lines of knowledge representation is built as training data. The dictionary comprises the merchants' names and products and some related terms to the receipt's categories. This dictionary refers to some online data such as Data World (see URL: https://data.world/) and Wikipedia (see URL: https://www.wikipedia.org/), and related words will be included in this dictionary.

3.3 Development overview

The proposed development is completely done in the Python programing language for its massive support in software libraries for image and text analysis. The development's three stages begin with image processing. Stage one improves the quality of receipt images before OCR parsing, while stage two is OCR parsing deciding the text layout localization, recognition and extraction. As a result, the stage two output is text in a machine-readable format. Following the extracted text is learning regular expressions and receipt classification in stage three. Overall, stage two is critical because getting the receipts' key information affects the classifiers' performance at post-OCR parsing. An evaluation protocol examines the performance of supervised ML classifiers on the test data. Figure 3.1 shows the flow chart of the proposed model.

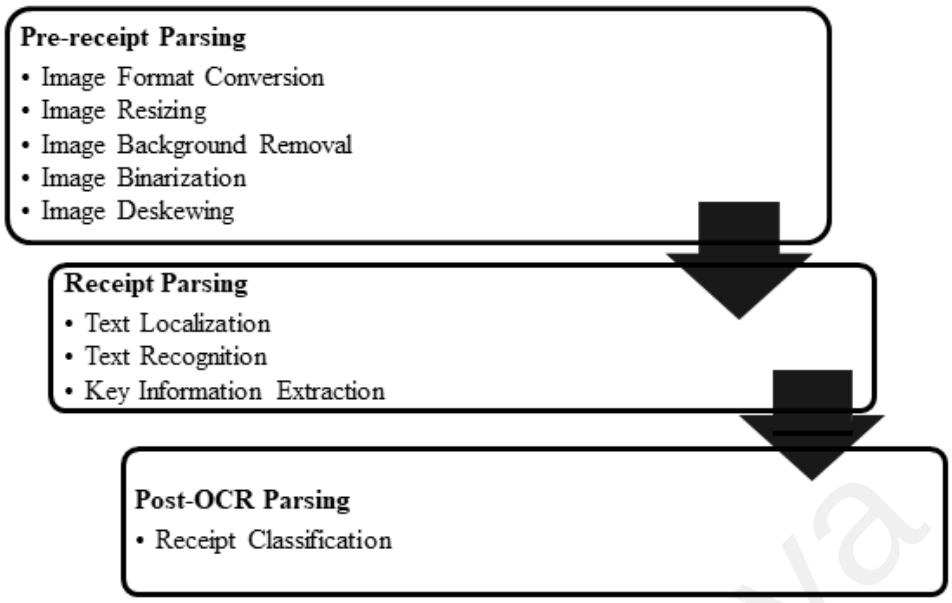


Figure 3.1: Flow Chart of Proposed Model

3.4 Pre-receipt parsing

The quality of receipt mainly influences the accuracy of the OCR tool. Before the receipt is scanned by the OCR tool, some steps of image preprocessing are required. The image format is converted to a standardised JPEG format. The PDF files are also converted to image format through the *pdf2image* library. Next, OpenCV as an open-source image processing library is used to preprocess the image with image resizing, background removal, binarization, and image deskewing.

3.4.1 Image format conversion

The purpose of the OCR tool is to extract text from an image. Thus the file, before processing by the OCR tool, needs to convert into image format. Then, all image objects are converted into JPEG format because most of the images captured by the camera and scanner are in JPEG format. Standardization of image format will make the data cleaner and unified. A Python library, *pdf2image,* is installed to convert PDF to image object. The Python Imaging Library (PIL) is an open-source library for image format convergence and resizing.

3.4.2 Image resizing

High-resolution images increase the processing time for the OCR tool, and it did not increase the accuracy. The optimal value of image resolution DPI is 300 for the most scenario. In this dissertation, an image is resized to less than 1200 pixel height and fixed the resolution DPI to 300. Large photos are shrunk, and small photos will be enlarged to reduce OCR processing time.

3.4.3 Image background removal

The image taken by the mobile camera may contain noisy background, and it will increase OCR processing time and results in garbage text. The uncertain part is removed, and one can pay attention to the part with key information. After the background is removed, OCR can process faster and accurate. Candy edge detection is used here. Figure 3.2 shows the before and after results for image background removal.

3.4.4 Image binarization

Image binarization is a process of converting coloured images to black and white image. Otsu's thresholding is selected for a reliable result. This step plays a significant role to ensure OCR operates in the best environment. An image taken by a mobile camera has a high chance to be noisy, shaded and dirty. This process can clean it. Figure 3.3 shows the before and after of image binarization.

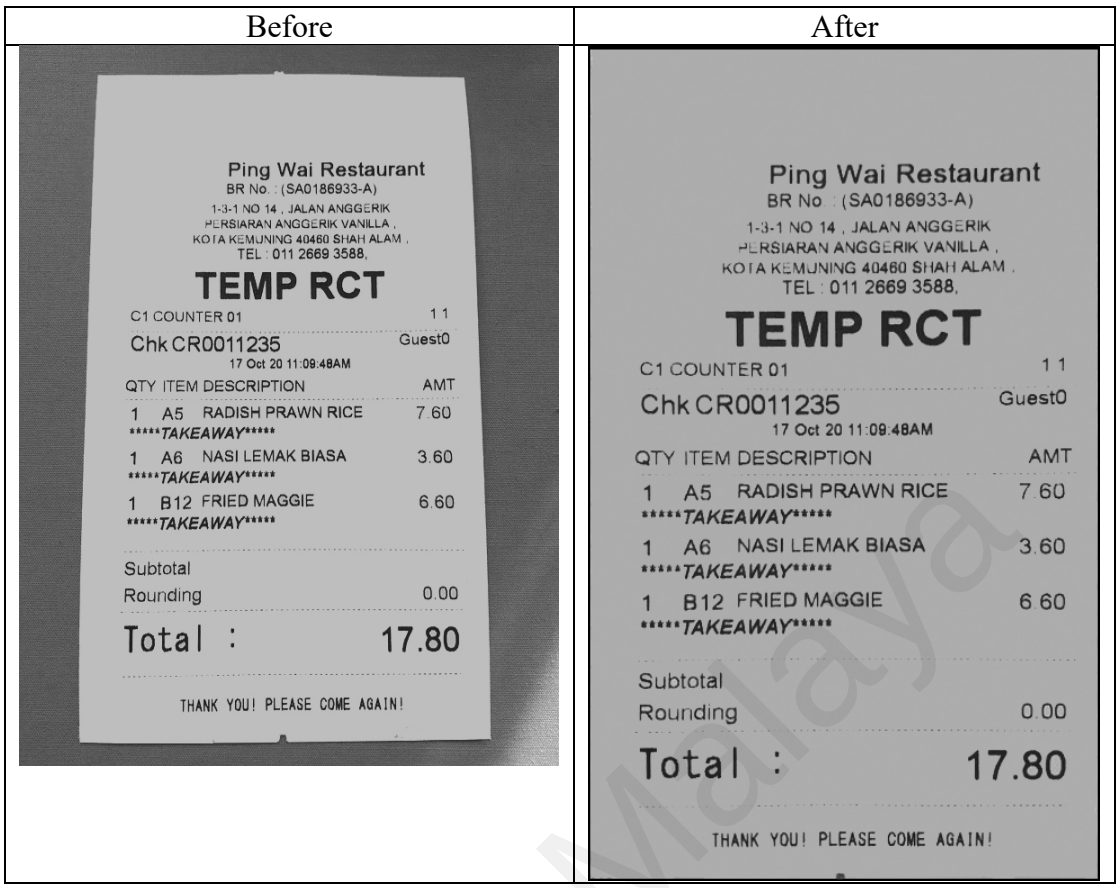


Figure 3.2: Before and after of image background removal

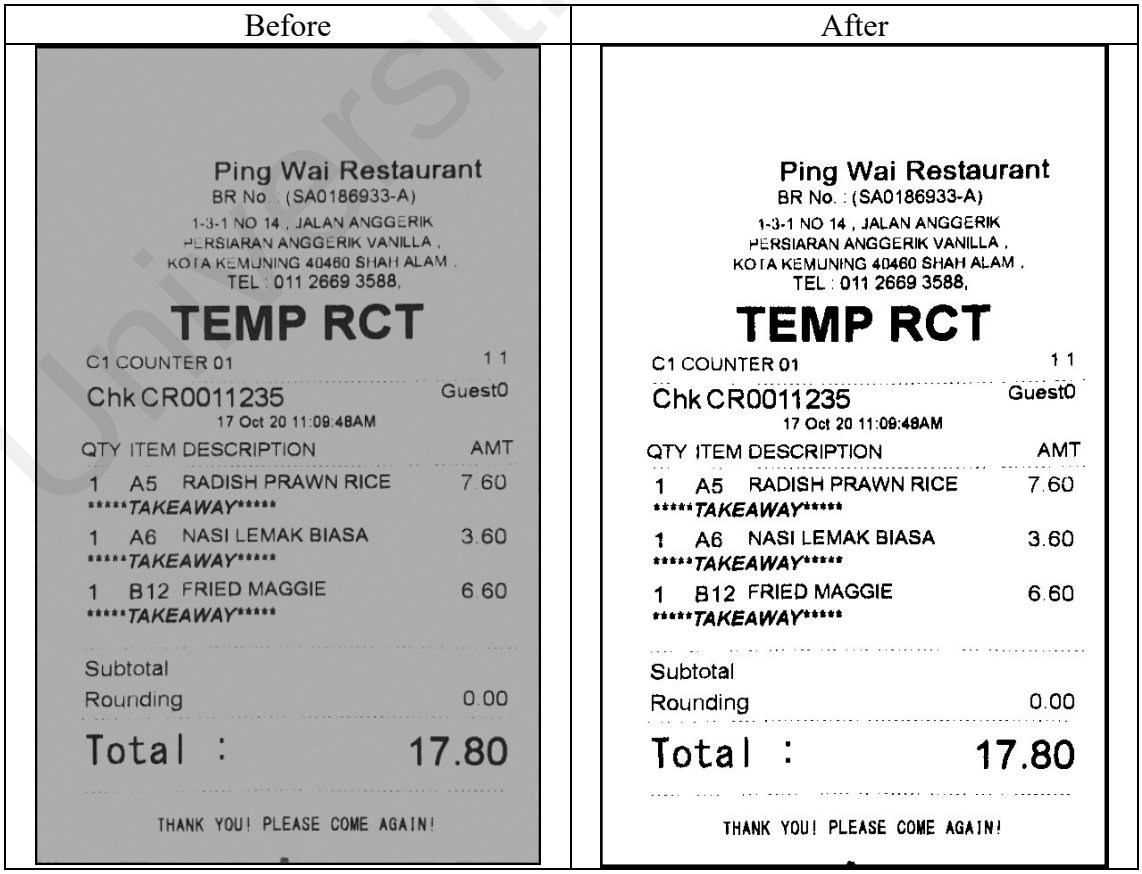


Figure 3.3: Before and after of image binarization

3.4.5 Image Deskewing

The text in the receipt can sometimes have layout orientation issues. The OCR will perform poorly in detecting and recognizing text for extraction without correcting the text layout angle. Therefore, image deskewing is applied to get the text correctly and horizontally aligned. The de-skewed text increases the accuracy of the OCR tool because it is much closer to what the OCR tool is supported to encounter when performing image analysis. Figure 3.4 shows the before and after of image deskewing with OCR's result. The the image preprocessing steps are written in Python using library such as numpy, cv2, imutils and PIL.

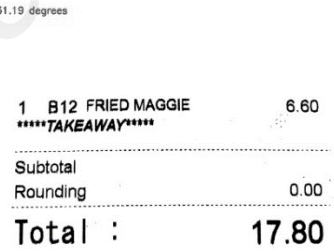
	Before	After
Image		
OCR's Result	**Undefined**	1 B12 FRIED MAGGIE 6.60 wees TAKEAWAY SOE . Subtotal Rounding 0.00 Total : 17.80

Figure 3.4: Before and after of image deskewing with OCR's result

3.5 Receipt parsing

Receipt parsing is the most critical and challenging phase. The performance and accuracy of the OCR tool will directly affect the overall performance of this research. After the image preprocessing in the previous stage, the words in the receipt will be more recognizable. Text localization and recognition are used to localized and recognise words in the receipt. After text extraction from the image, key information is required to extract.

3.5.1 Text localization & recognition

Receipt parsing can perform with or without text localization. In this research, text localization is used to obtain text containing the image region. A bounding box is used to locate key field texts such as merchants names, goods names, unit price and total cost in the receipt. The texts are localised and annotated as rectangles with four verticals accurately. The goal of receipt parsing is to detect and compute the bounding box for every region of text in the receipt, then label and determine a class label for each computed bounding box.

In this research, the Tesseract OCR tool is used for text detection and localization. Tesseract OCR engine and pytesseract are install into Python. Some parameters are defined to control the algorithm, such as minimum confidence threshold, *–min-conf*, which filters weak text detection. By default, the *–min-conf* value is set to 0, and all detections are returned. If the *–min-conf* value is too high, some low-quality text regions may be filtered. A bounding box with recognised text and coordinates for the box's centre, width, and height (c_x, c_y, w, h) is generated, then draw on the image around detected text by using OpenCV. Figure 3.5 shows the image with a bounding box and text recognised by the Tesseract OCR tool.

The text recognised by the Tesseract OCR tool is saved into a text file. These texts contain much important information; however, the discriminators will be extracted in the following steps after filtering and extracting only the key information. The accuracy of the Tesseract OCR tool will also be evaluated, which will be discussed in the following section.

Image After Text Localization	OCR's Result
	Ping Wai Restaurant BR No. :(SA0186933-A) 4-3-1 NO 14, JALAN ANGGERIK PERSIARAN ANGGERIK VANILLA , KOTA KEMUNING 40460 SHAH ALAM , TEL : 011 2669 3588, TEMP RCT C1 COUNTER 01 11 ChkCROO11235 Guest0 17 Oct 20 11:08:48AM QTY ITEM DESCRIPTION AMT 1 AS RADISH PRAWN RICE 7.60 880i TAKE AWAY**OFE 1 A6 NASI LEMAK BIASA 3.60 #8883 TA KEAWAY*OEEE 1 812 FRIED MAGGIE 6.60 seeneTAKEAWAY*** Subtotal Rounding 0.00 Total : 17.80 THANK YOU! PLEASE COME AGAIN!

Figure 3.5: Image with bounding box and text recognised by Tesseract OCR tool

3.5.2 Key information extraction

The merchant's name, transaction date and total price are the key information in receipt and need to be extracted. A goods name is also extracted because it is required during receipt classification. The key information can be determined through the positioning of the text, keywords and string pattern. For example, a merchant's name is usually at the header of a receipt. The total price is often printed alongside keywords such as "Total" or "Subtotal". Another typical pick up is the transaction date consists of the "DD/MM/YYYY" pattern.

Text extracted by the Tesseract OCR tool is saved line by line in a top to bottom order of a text file. Since the merchant's name is typically at the top of the receipt, the first line from the text file is usually the merchant's name. Sometimes the top of the receipt is not the merchant's name but the title or page number. This situation is also handled using a regular expression to identify the date and price in the receipt. For example, expression "^\d{1,2}\/\d{1,2}\/\d{4}$" can be used to identify date format such as 21/09/2020 and it usually represent transaction date of the receipt. However, there are many different date formats observed in the receipt. Therefore, different expressions will be included in the key information extraction algorithm to cover different date formats, such as 21-09-21 and 21 Sep 2021.

On the other hand, the expression "(\d{1,3},?)*\d{1,3}\.\d{2}" is used to identify pricing with at least one digit before and two digits after the decimal place. All line with pricing is filtered. Keywords such as "Total", "Subtotal", and "Nett" are used to identify the actual total price in the receipt. Some of the receipts are partly in Malay, where the word "*Jumlah*" with total in Malay is included as a keyword to identify total price. Besides that, other filtered lines with prices are considered as goods used for receipt classification. The accuracy of key information will be evaluated, which will be discussed in the following section. Figure 3.6 shows the example of key information extracted.

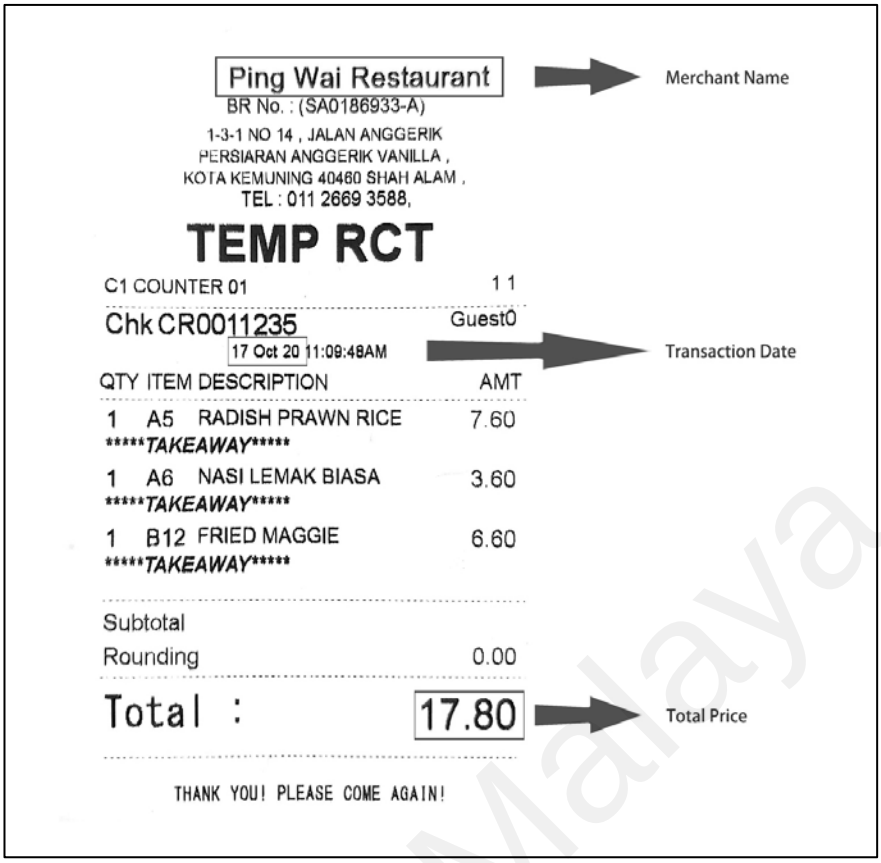


Figure 3.6: Example of key information extracted

3.6 Post-OCR parsing

The ICDAR2019 observed that careful selection of use case related patterns or keywords might improve post-OCR receipt parsing. However, they did not consider text classification. Their effort stops at the SROIE problem only. Inspired by this observation, a dictionary is developed to be specific to the use case in this dissertation. This dictionary is also valuable for the data classification when looking to match the extracted keywords' content and category to the ground truth.

3.6.1 Dictionary development

The dictionary is developed based on the behaviour of Malaysian origin receipt, which included words related to a Malaysian employee expenditure reimbursement. The Malaysian context is essential because the culture gives unique names for food, drinks

and places. For example, a receipt with *Air Kedah* is not the same category as *Air Batu Campur*, even though both begin with "*Air*", which means water. *Air Kedah* is a utility bill, whereas *Air Batu Campur* is the name of a drink. Figure 3.7 shows part of the dictionary, which is separated by name and label. The six categories are given a label each—for example, M for meals, G for groceries, T for telecommunication and so forth.

	name	label
2	3G	T
3	4G	T
4	5G	T
5	7 ELEVEN	G
6	99 SPEEDMART	G
7	AA PHARMACY	G
8	ABC	M
9	ACAR	M
10	ACCOMMODATION	A
11	AEON	G
12	AEON BIG	G
13	AEON GROUP	G
14	AEON MAXVALUE	G
15	AEON WELLNESS	G
16	AGAR AGAR	M
17	AGAR-AGAR	M
18	AGODA	A
19	AIR	A
20	AIR BATU CAMPUR	M
21	AIR BNB	A
22	AIR KEDAH	A
23	AIR KELANTAN	A

Figure 3.7: Example of Dictionary Content

3.6.2 Receipt classification

The receipts are classified into six categories based on its characteristic. For example, a receipt purchase for a burger or fried chicken will be categorised as meals, purchased from the supermarket will be categorised as grocery. The merchant's name and the goods extracted in the previous step are used as input to the classifier. Before passing the input to the classifier, all inputs must be converted to vectors of numbers because the classifier can only be processed in numerical values. Seven popular classifiers are chosen to classify

the receipt because there are the most basic and commonly used in various text classification studies.

The classifier is selected from one of the supervised learning approaches, which require labelled training data to learn. The training data preparation has been discussed in the previous section. The NLP model converts text to number before training data and input are passed to the classifier because the classifier can only handle numerical data. These data are often incomplete and inconsistent—data preprocessing is an effective method in getting better classification results.

Tokenization and word lemmatization are techniques used for data preprocessing. Tokenization filters the text into word chunks or tokens used as input for further processing. Function *word_tokenize* and *sent_tokenize* from NLTK as python library break the text into a list of words. Word lemmatization is used to reduce inflectional forms of a word into root forms or as nouns. For example, "walking" is converted to the word "walk". The list of words is converted to uppercase, and all single-character word is removed to make data more standard and unified. Besides that, stop words and non-alpha words are removed so that the data is cleaner for classification.

Next, word vectorization is used to turn the words into numerical feature vectors. TF-IDF is the most popular method adopted to assign scores based on the frequency appearance of each word. TF-IDF builds a vocabulary of words through learning the corpus data, and a unique integer number is assigned to each of these words. Parameter *max_features* is set to 5000 mean maximum of 5000 unique words in the vocabulary. Label encoder is used to transform predefined receipt's category into numerical values for classification purposes.

Finally, the data is ready to feed into a different classifier. This study experimented with seven supervised type classifiers: the Naïve Bayes, maximum entropy, SVM, linear SVC, decision tree, random forest, and KNN. These classifiers are well known for regression and classification problems. Their statistical nature requires labelled or annotated training data but is not vulnerable to small-sized data. The training data is a group of data in the knowledge representation, whereas test data is the merchant's name and goods extracted from receipt. The training data will be first fed to the classifiers, followed by test data. Function *classification_report* from the Scikit-learn module is used to calculate accuracy and F1 score, discussed in the following section. Figure 3.8 shows the process of receipt classification.

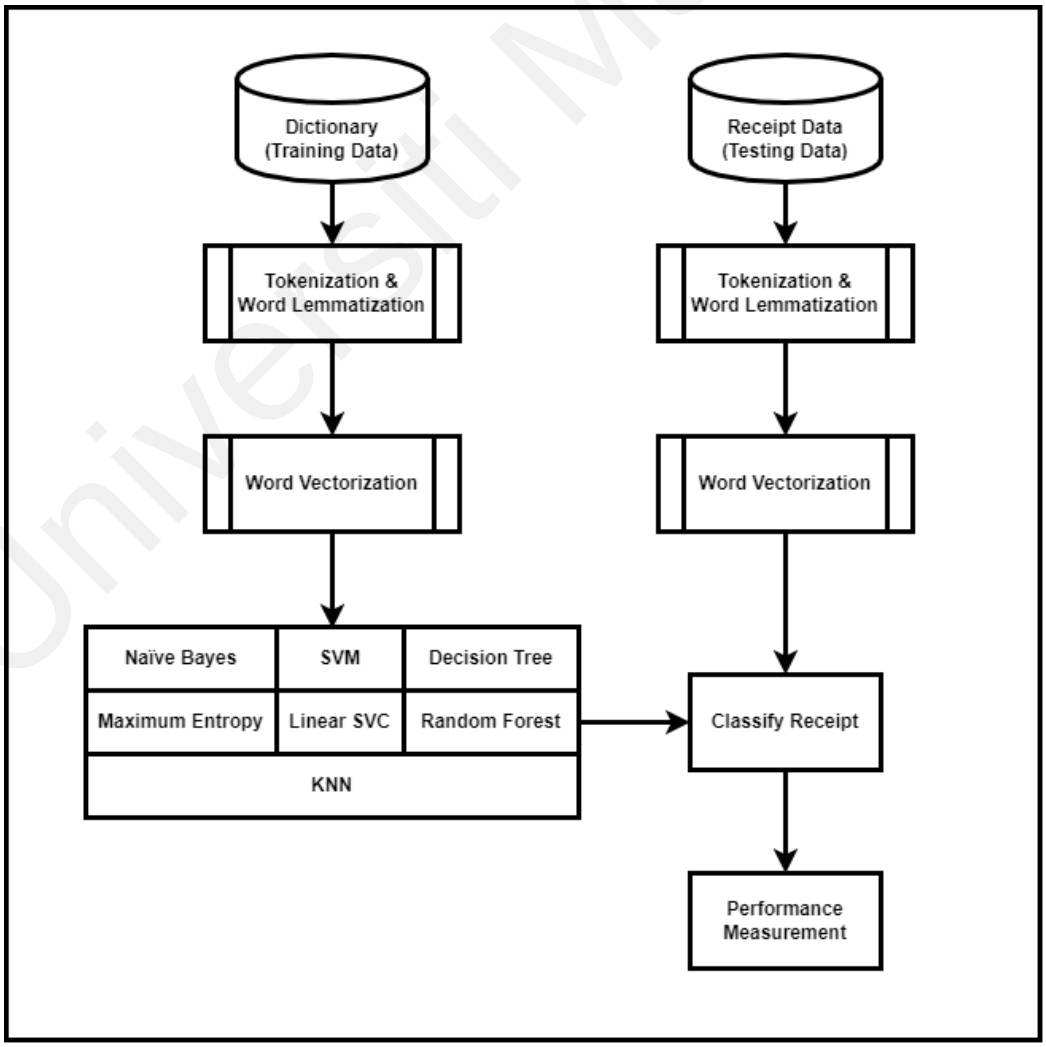


Figure 3.8: Process of receipt classification

3.7 Performance Measurement

The performance of the proposed model is evaluated according to ICDAR2019's standard. F1 score is recommended as the metrics derived from average precision, mAP and recall, representing overall accuracy. The equation of the F1 score can be defined as Equation (7).

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision \cdot recall} \tag{7}$$

Where precision represented the number of correct matches over the number of detected words and recall is the number of correct matches over the number of ground truth words. F1 score is selected as a metric in this research because it can seek a balance between precision and recall. Also, the F1 score can handle the uneven distribution of categories. Besides that, accuracy, representing the number of correct matches over the total number, is considered the second metric.

The performance measurement is separated into three parts. Firstly, the performance of the Tesseract OCR tool is evaluated by comparing the output of Tesseract and text in the receipt. The number of characters and words recognised correctly will be recorded to calculate accuracy. The accuracy of OCR is divided into two-level, which are character level and word level. Character level accuracy can be defined as Equation (8), whereas word-level accuracy can be defined as Equation (9).

$$Chatacter\ Level\ Accuracy\ = \frac{Number\ of\ characters\ recognized\ correctly}{Total\ of\ characters\ in\ receipt} \tag{8}$$

$$Word\ Level\ Accuracy\ = \frac{Number\ of\ words\ recognized\ correctly}{Total\ of\ words\ in\ receipt} \quad (9)$$

The number of characters and words recognised correctly were compared manually between the OCR's output and the receipt's content. The manual comparison is because most receipts are collected daily. Without a definite actual text from receipt, the manual is the most direct and fast method to identify the number of characters and words recognised correctly. In this part, precision is calculated by the number of correct matches over the number of detected words. In contrast, recall is calculated by the number of correct matches over ground truth words.

Secondly, the performance of key information extraction is evaluated. The key information extracted included the merchant's name, transaction date and total price. The measurement standard is based on the requirement of a reimbursement request, where the transaction date and total price must be accurate. On the other hand, minor typos on the merchant's name are allowed. The measurement for this part is very straightforward. A misrecognition at OCR parsing may cause the merchant's name extracted with typos. Regardless of spelling error, the merchant's name is added to the frequency count so long there is a successful receipt extraction.

On the other hand, if the date and price are wrong, regardless of whether or not it is an OCR misrecognition issue, they will be counted as an error. The frequency count will not increase unless the correct data and price details are extracted from the receipt. The mAP is computed over all the extracted key information of all the receipt images. F1 score is computed based on mAP and recall, and the score makes up the ranking.

Third, the performance of all classifiers is evaluated. After the classifiers successfully predict the receipt categories, the Python function *classification_report* is called to generate a classification report. The predicted and actual results are imported into the Python function, then overall accuracy and F1 score will be calculated. In the generated classification report, precision, recall and F1 score for each class are shown. The overall accuracy, an average of macro and weighted F1 score can also be found in the report. Average of macro and weighted are the function to compute the F1 score for each label. Still, the macro returns the average without considering the proportion for each label, whereas weighted returns the average by considering the proportion for each label. In this research, the weighted average is used because the proportion of labels needs to be considered. The receipts classified as meals are often the most from the collected dataset. Following the classifiers performance evaluation, the results obtained are compared with previous related research. The comparison of OCR tools' performance, key information extraction, and all selected classifiers are presented using a table. The comparison concludes that the performance of the proposed model can extract key information accurate and classify the receipt efficiently to meet reimbursement requirements.

3.8 Chapter summary

This chapter presented three development stages at pre-receipt parsing, during OCR receipt parsing, and post-OCR parsing. The data used are a combination of Malaysian everyday receipts and ICDAR2019's scanned receipt in the online database (see Appendix A for more details of the receipt collected). Seven supervised type classifiers are considered in the last stage. They are observed with dissimilar performances due to the difference in their statistical approaches. In the next chapter, the results are presented and discussed.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Overview

This chapter presents the result of OCR, key information extraction and classifier. These results are analysed based on the performance analysis mentioned in the previous chapter. The accuracy and F1 score are used as metrics to measure the performance of the proposed model. Then, the comparative analysis is done by comparing the result obtained with other previous related works or research. The comparison is based on the performance of OCR with key information extraction and classifier. OCR is the main component to extract information from the receipt. Therefore related research must include OCR to be used for comparison.

4.2 OCR result and discussion

The number of characters and words recognised correctly are measured by comparing OCR's output with text in the receipt. Accuracy and F1 scores in character and word levels are calculated based on the equations discussed in Chapter 3. See Table 4.1, showing the result summary of the tesseract OCR tool. The summary shows different categories like overall, physical receipt and electronic receipt that included screenshot and PDF. 70 “Physical” receipts are captured by the scanner or camera, while the 30 e-receipts are captured from screenshots or PDFs. The "Electronic" category is further divided into the category "Screenshot" and category "PDF".

Table 4.1: Tesseract OCR tool result

	Accuracy (%)		F1 Score	
Category	**Character Level**	**Word Level**	**Character Level**	**Word Level**
Overall	90.72457	78.51078	0.89	0.78
Physical	92.15914	80.23667	0.90	0.79
Electronic	87.99889	75.23158	0.89	0.77
Screenshot	99.80522	99.08303	0.99	0.99
PDF	81.16364	61.42284	0.82	0.64

The Tesseract OCR tool result shows an overall 90.72% accuracy at the character level and 78.51% performance at the word level. The accuracy for character level passing the 90% mark is encouraging. However, the performance is considered relatively low for the word level, which may affect the performance of key information extraction and classifiers. The F1 scores of the character and word levels are 0.89 and 0.78, respectively, which means that the accuracy rate is not affected after considering false-positive cases.

The OCR processing on screenshot receipts has the highest accuracy, achieving 99.81% and 99.08% for the character and word levels. The performance means that there is less than one error in 100 characters. The primary reason is that the screenshot receipts get a high resolution and have very little noise. Referring to Table 4.1, the OCR accuracy for PDFs is lower in terms of character and word level accuracies, scoring 81.16% and 61.42%, respectively. Although it is also an electronic-typed receipt, it includes many images for advertisement and logos with borders. These other entities that are not text-based on the PDF bills limit the OCR text recognition capacity for the document type. Figure 4.1 shows the tesseract OCR tool result in graph.

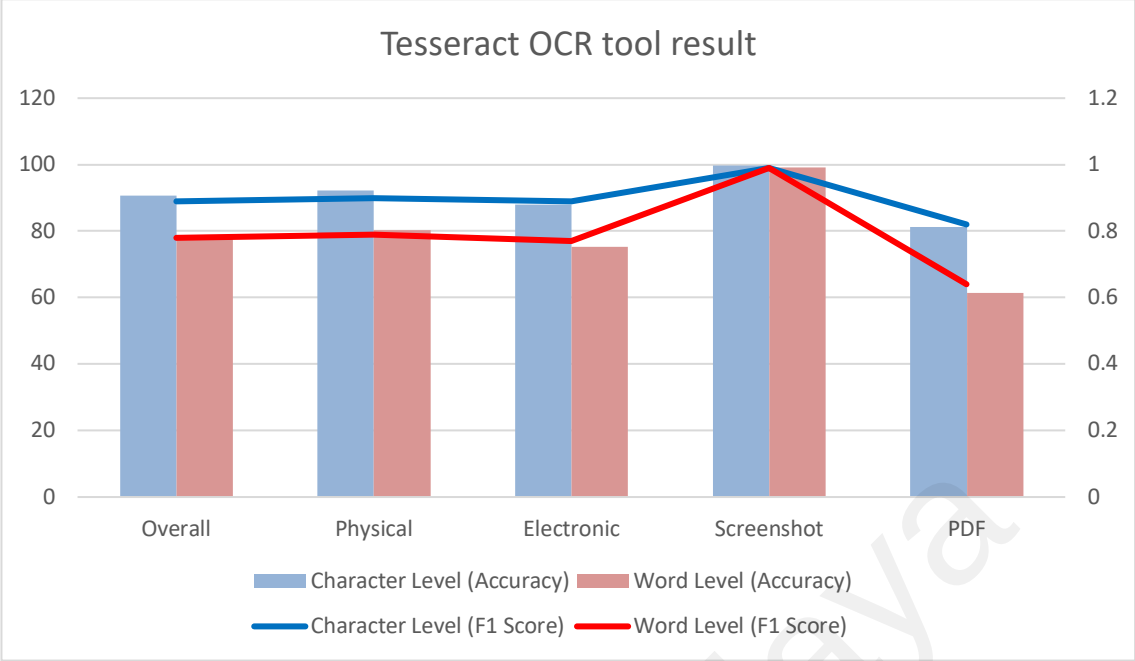


Figure 4.1: Tesseract OCR tool result

The Tesseract OCR tool is highly susceptible to noise. Although the receipt is preprocessed, the Tesseract still has trouble fully recognising the text in the receipt to achieve 100% accuracy. Several factors are discussed that affected the accuracy of OCR, including quality of the original document, quality of scan process, and common misrecognised issues.

4.2.1 Quality of original document

First of all, the quality of the original document is the most important for OCR recognition. If the receipt is wrinkled and torn, the OCR cannot obtain accurate text recognition and extraction. A wrinkled receipt will cause some of the text to be misaligned, and a torn receipt will lose a large part of the information. Some physical receipts are printed on low-quality paper, decreasing crispness and contrast between the background and foreground. Low contrast colour ink such as red, blue or purple will cause the font to be blurred, or some receipts are not printed clearly. Although the naked eye can recognise it, it is easily ignored by the OCR. Most of the actual receipts are printed on carbon paper,

which can display all fonts, but if the receipt is stored for a long time or stored in a bad environment, the fonts will be blurred or disappear.

4.2.2 Quality of scan process

Secondly, the quality of the scan is one of the factors which affected the OCR's performance. Good quality receipt but inferior scanning technology will lead to poor overall text recognition. Scanners and cameras are the primary devices to capture receipts into images. If their DPI is set lower than 200, the scanning devices will produce incomprehensible results.

Furthermore, the resolution of the scanned images affects the distinctness of the font. Although these scanned images will undergo image preprocessing, it sometimes has a reversed effect; images with lower scan quality will not be better due to image preprocessing but will worsen. Figure 4.2 shows an example of a low-quality scanned image that has followed through all the image preprocessing methods. Due to the unbalanced contrast, the binarization converts a large part of the image to black damaging the receipt quality and potentially the receipt OCR results.

Figure 4.2: Low-quality scanned image after image preprocessing

4.2.3 Common misrecognised issues

Thirdly, some common misrecognised issues may affect the performance of OCR. One of the most common is similar words such as the character "0", "O", "D", and "o". The appearance of these words is very similar; therefore, OCR will often misrecognise them. Although this problem is common, it has a significant impact on the extraction of key information. The wrong identification of OCR will lead to errors in identifying the date and price based on the pattern. Due to the previous factors, part of the character may be slightly blurred than other parts, such as the character "8" will often be mistaken for "6" or "9", which dramatically reduces the accuracy of OCR. Some non-standard fonts are used in the receipt too, and OCR may ignore or misrecognise these fonts. Figure 4.3 shows some non-standard fonts that often appear in receipts.

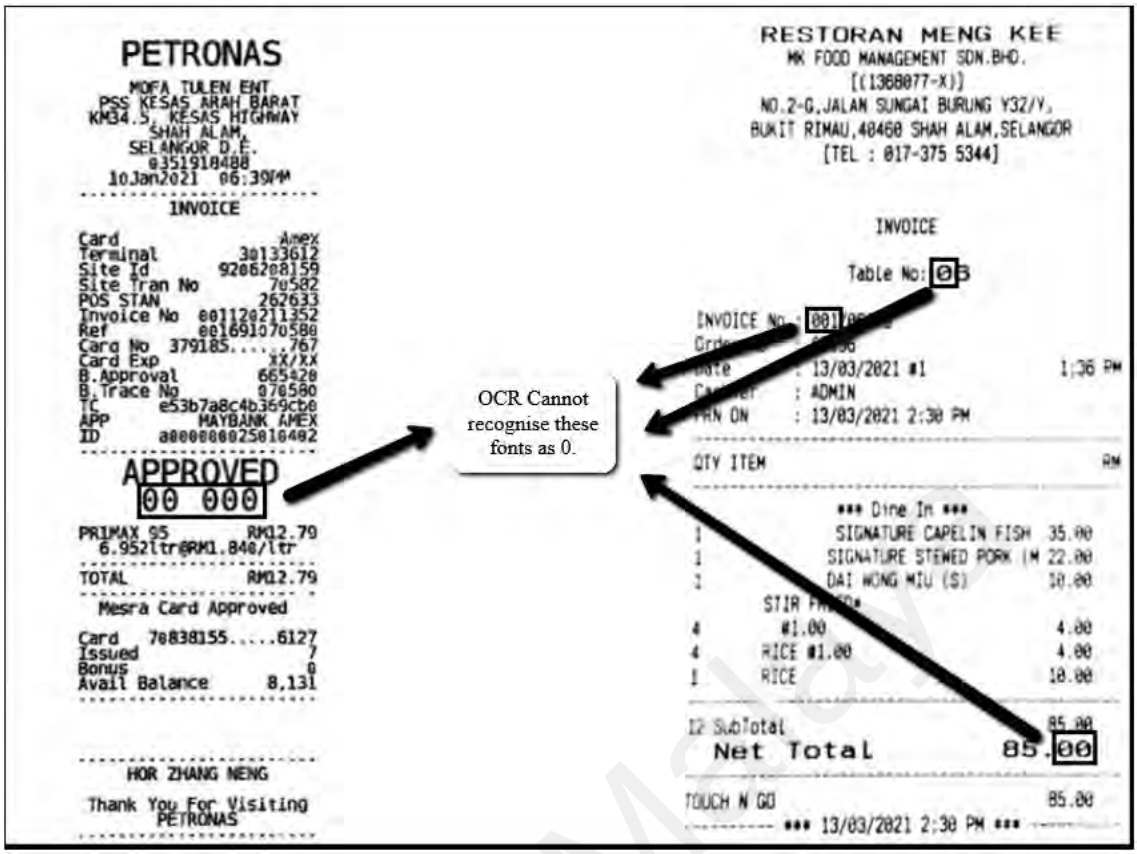


Figure 4.3: Non-standard fonts that often appear in receipts

Some electronic receipts carry advertisements and logos with text covered by background colour. The OCR perform recognition poorly with these elements. Image preprocessing and OCR cannot identify whether these elements are unwanted parts or noise, and OCR cannot recognise the text covered by the background colour. In addition, some separator lines, such as dotted lines or double lines, can confuse OCR and ignore the text around the separator line. The same also happens to text covered by borderlines. Therefore the text in the table with borderlines will often be ignored. Figure 4.4 shows that the text example is ignored by OCR, near separator lines or covered by borderlines.

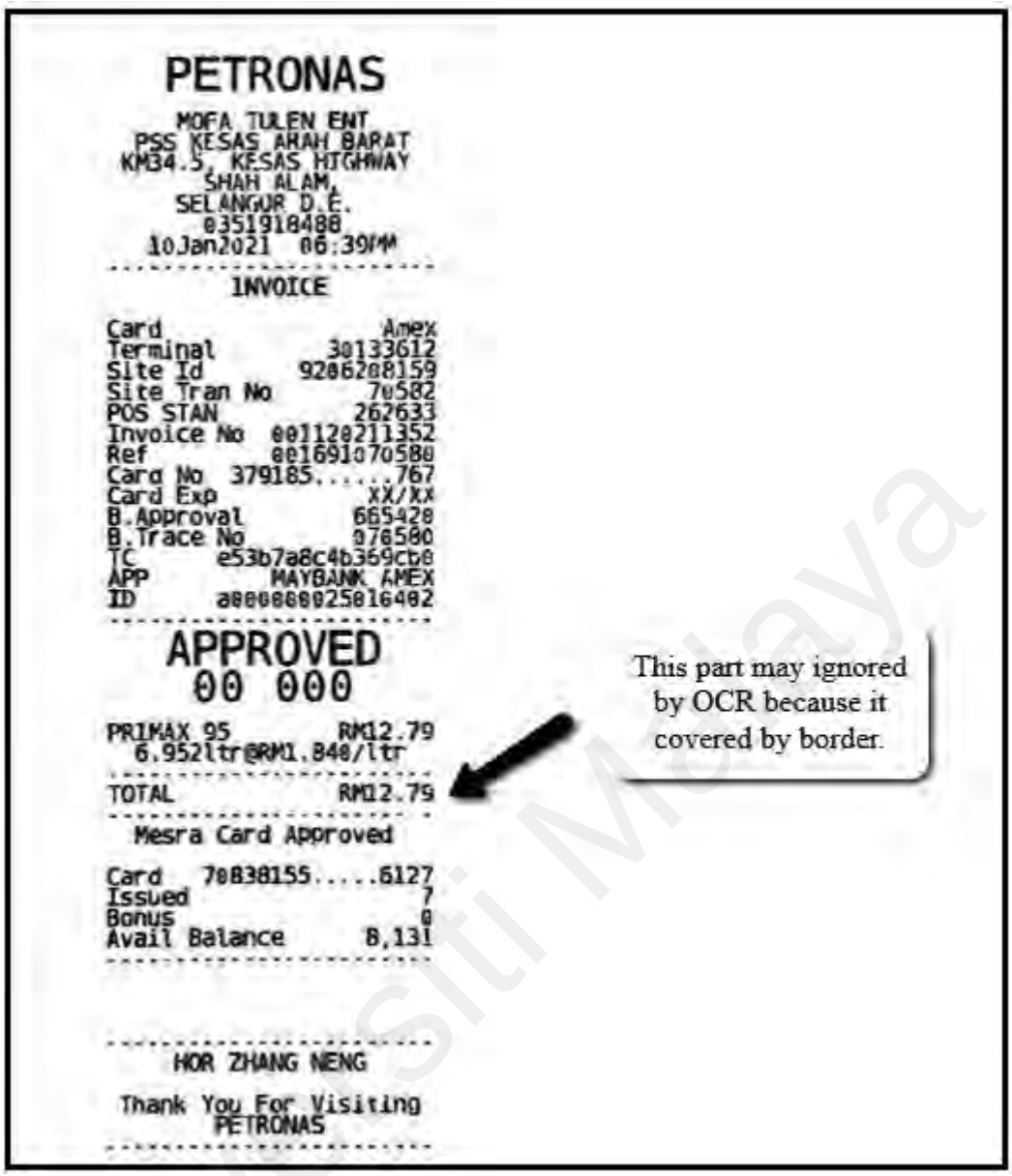


Figure 4.4: Example of Text is ignored by OCR

4.3 Key information extraction result and discussion

The extracted key information is marked as correct if matched with the ground truth. Otherwise, it will be marked as incorrect. The measurement method follows the steps discussed in Chapter 3. The F1 score is computed based on mAP and recall, following the accurate key information extraction over total test receipt images. Table 4.2 shows the key information extraction result for the different categories: merchant's name, transaction date, and total price. Category "Overall" includes the other three categories, and the value is computed based on the mean of these categories.

Table 4.2: Key information extraction result

Category	Accuracy (%)	F1 Score
Overall	74.33	0.74
Merchant's Name	93.00	0.93
Transaction Date	70.00	0.70
Total Price	60.00	0.60

The key information extraction result shows the overall accuracy is 74.33%, with an F1 score of 0.74. This result is acceptable, showing a success rate of more than 70% in effectively extracting the key information from the receipt. Among all the categories, the successful extraction rate of business names reached 93%, showing that most merchant names are in the header part. The remaining 7% error rate is factored by the merchant's name missing in the header part, or the OCR cannot recognize any merchants' names represented by their company logos.

The accuracy rate for extracting the transaction date is only 70%, which is barely acceptable. Regular expressions are used to extract the transaction date. Although most of the pattern is covered by expression, there are still a small number of patterns that cannot be covered. The expression defined is DDMMYYYY, denoting the expression follows the day, month and year format. The day can be single or double digits, whereas the year can be double digits or four. A single-digit or double-digit can represent the month, and in the description written such as January or short form, Jan. There can be a space, hyphen or other delimiters like "/" between day and month or month and year. However, it will fail to extract if the transaction date is not in this format, for example, YYYYMMDD.

The error rate of OCR also leads to low accuracy of key information extraction. Some symbols were misrecognised, causing the model to fail to recognise the correct date. It has also happened on misrecognition of digits, such as the correct data is extracted, but the value is wrong. From the data collection and analysis, nine receipts returned the correct date but the wrong value, and 15 receipts returned empty dates due to mismatch date patterns from the OCR result. Another six receipts returned the wrong date because of the uncovered date format and limitation of the model to identify the correct date.

In addition, the accuracy rate of extracting the total price is only 60%, and there is still a lot of space for improvement. The main reason for the failure of extracting a total price from receipt is the performance of OCR. Pattern for extracting a price from receipt is any value with two decimal points, and the total price is identified by keywords such as “Total” or “Nett”. Two conditions must be met to extract the total price from receipt successfully, but one of them will fail in most cases. The most common occurrence is that the decimal point is misrecognized for a comma or space. In this case, the model cannot extract the actual price based on the pattern. Then, OCR returns the wrong value for the price, similar to transaction date extraction, decreasing the accuracy of key information extraction. Based on the analysis, 12 receipts’ total price is extracted wrongly because OCR results return the wrong value. Figure 4.5 shows the key information extraction result in graph.

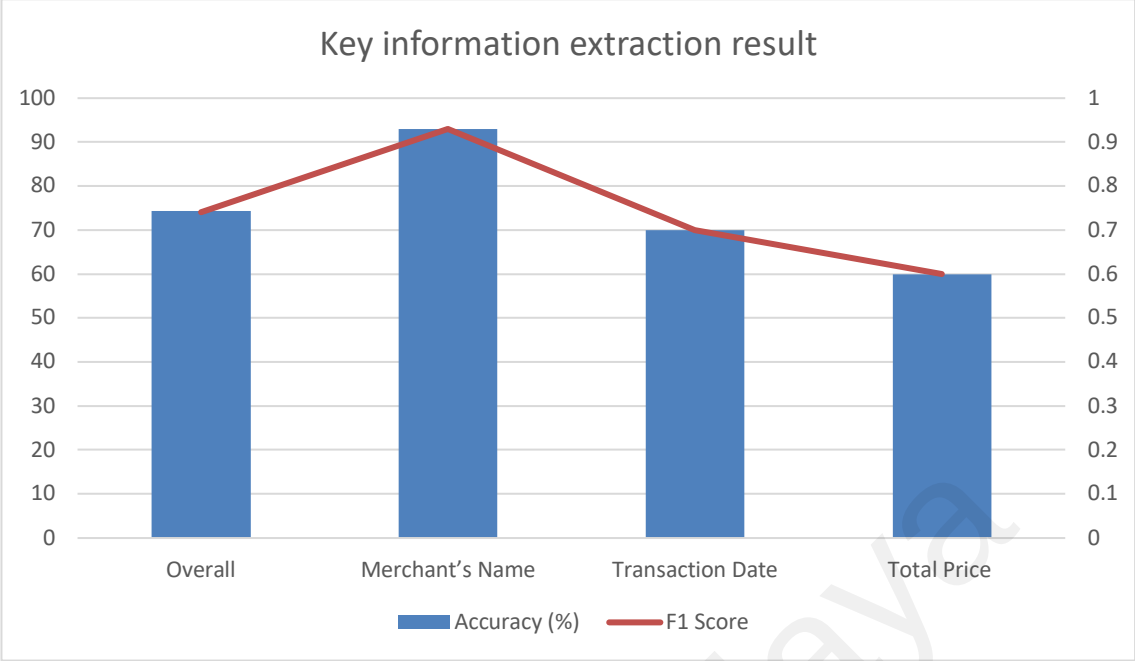


Figure 4.5: Key information extraction result

4.4 Classifier result and discussion

The results from the classification task for classifying the receipt into the correct category is provided. The classifier is evaluated by the python function *classification_report*, which returns the accuracy and F1 score, and enters the merchant's name and good name. These inputs come from the key information extraction step without any correction of the wrong information or value. Results predicted by classifiers are compared with pre-label data to compute the accuracy. F1 score is computed based on the precision and recall for each class. Table 4.3 shows the classifier's result, which included Naïve Bayes, maximum entropy, SVM, linear SVC, KNN, decision tree, and random forest.

Table 4.3: Supervised-based classifiers results

Type	Accuracy (%)	F1 Score
Naïve Bayes	**63.0**	**0.55**
Maximum Entropy	**63.0**	**0.57**
SVM	**65.0**	**0.61**
Linear SVC	**80.0**	**0.79**
KNN	**56.0**	**0.59**
Decision Tree	**52.0**	**0.57**
Random Forest	**56.0**	**0.61**

The classifiers result shows the accuracy between 52% to 80% and the F1 score between 0.55 to 0.79. The methodology used in this research for classification is slightly different compared to other research. Other research will usually divide their receipts into two sets, which are 80% for training and 20% for testing. The classifiers in this research are trained by knowledge representation and tested by key information extracted from receipt. It isn't easy to prepare the knowledge representation because it may not cover all related words for different receipt categories. Figure 4.6 shows the supervised-based classifiers result in graph.

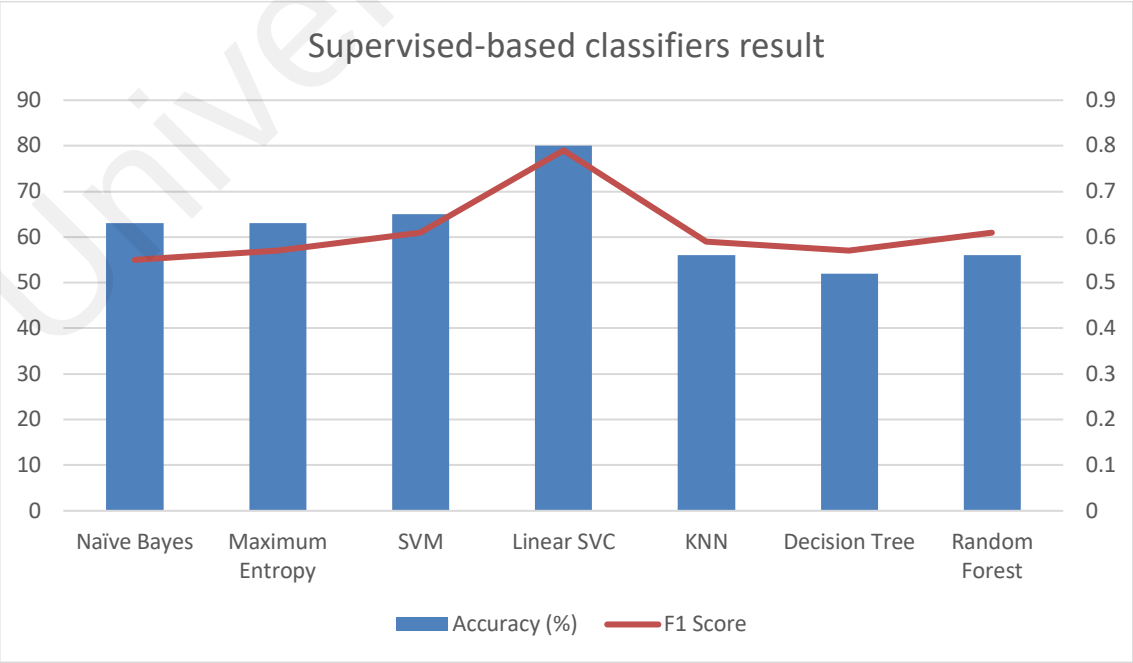


Figure 4.6: Supervised-based classifiers result

A customised knowledge representation can reflect a practical business system because increasing the number of knowledge representations will increase the receipt's category. The classifier's performance is not limited to the current receipts but can classify receipts from different establishments. On the other hand, if the prepared receipts are trained and tested separately, it can only mean that the classifier can classify through the current receipts, although the accuracy of the classifier may be improved. The classifier must be retrained if the receipt has a new source entry; otherwise, the recognition accuracy is not guaranteed.

Retraining a classifier is not a straightforward task and is highly time-consuming. Therefore, frequent classifier retraining is not practical or sustainable for enterprises dealing with new image document formats as receipt sources. The classifiers must learn all formats and forms of the receipt images; otherwise, the prediction result gets affected. Therefore, updating the knowledge representation to substitute frequent training is more sensible for a practical system.

4.4.1 Naïve Bayes and maximum entropy

Naïve Bayes and maximum entropy are probabilistic-based classifiers with 63% accuracy, whereas the F1 scores are 0.55 and 0.57, respectively. The Naive Bayes' classification opinion is based on the maximum value of the posterior probability of each keyword in the training data. The built of knowledge representation included related words and merchant's name from the different receipt categories. Therefore, some words may group into different classes, such as "egg" grouped into meals or groceries. When the input consists of "egg", it gets the same probability fall on class meal and class grocery.

On the other hand, the maximum entropy classifier predictions are feature dependent. If a word frequently occurred in a class, the weight will be increased, and the probability of the class selected will be higher. For example, if the word "egg" frequently occurred in class meals, “egg” gets a higher probability of falling into the class meals. Unlike the Naïve Bayes classifier, the overall probability is computed based on all words in the input. The class with the highest overall probability will be returned. Although their accuracy rates are 63%, the maximum entropy classifier considers the dependence of features, which is why the maximum entropy classifier gets a higher F1 score.

4.4.2 SVM and linear SVC

SVM classifier and linear SVC classifier are non-probabilistic binary linear classifiers. SVM classifier reported 65% accuracy with an F1 score of 0.61, whereas the linear SVC classifier reported 80% accuracy with an F1 score of 0.79. These classifiers show the best accuracy and F1 scores among the classifiers. The high F1 score gives good results on imbalanced classification problems showcasing the classifiers higher precision and recall.

The classify option for SVM classifier and linear SVC classifier are searching for the best dividing field that separates high-dimensional text data perfectly into classes. The training data are plot into a graph. Then a hyperplane is computed to separate into different classes. The larger the number of training data for the class, the larger areas to represent the class. For example, in 1150 rows of training data, 523 items belonging to the class meal, which is the highest number of representatives from the training dataset. The class meal has the largest area in the SVM classifier. The hyperplane plots area data into the graph before returning a prediction according to the class for the area.

Linear SVC classifier is an SVM classifier, but it has more flexibility and additional parameters to minimise the squared hinge loss. It is converged quicker on massive amounts of knowledge using the linear kernel function, which is relatively faster than the non-linear classifiers. The result of the linear SVC classifier is the best. Besides the strengths just discussed, it can better handle dense and sparse input and handle multi-class support according to the "one-to-many rest" scheme with limited training data.

4.4.3 k-nearest neighbors

KNN classifier is a non-parametric classifier. The KNN classifier reported 56% accuracy with an F1 score of 0.59, the second-lowest score among the group of classifiers. The KNN classifier will rank the nearest neighbours of the labelled examples from the training set and use the highest-ranked neighbours' classes to derive a class assignment. The parameter k has been set to 3, which means a total of three nearest neighbours is selected. The parameter k value is crucial because the probability result will be computed from the number of k samples. Based on the analysis, most of the test data are located in the location around class transport. Therefore there is a higher probability of being classified in the class. However, it is classified into the wrong class resulting in low accuracy.

4.4.4 Decision tree and random forest

The decision tree classifier and random forest classifier are recursive partition classifiers. The decision tree classifier reported 52% accuracy with an F1 score of 0.57, whereas the random forest classifier reported 56% accuracy with an F1 score of 0.61. The decision tree is built by using a greedy strategy to depict each possible result of decision making. It is divided into branches from root representing the difference option and end by a leaf node representing a key node for the prediction. The test data will go branch by branch until it reaches a leaf. The value of the leaf will return a prediction of that decision tree.

The random forest classifier building blocks are multiple random decision trees. The forest leverages when the trees are sparsed and highly uncorrelated. The prediction follows random feature selection and randomly chosen data subsets from a grown decision tree. The number of estimators is set to 100, which means a total of 100 decision trees in the forest. The performance of the random forest classifier is better than the decision tree classifier because it gets the precision based on a group of decision trees and uses averages to improve prediction accuracy and control overfitting. However, it is not suitable for high dimensional feature space because it is too difficult to identify the key node to build the tree. Therefore, the accuracy of the decision tree classifier and random forest classifier returned low accuracy.

4.5 Comparative Analysis

The results obtained are compared with previous related research as mentioned in Chapter 3. The OCR, key information extraction, and classifier results are compared with existing research related to receipt parsing. These researchers may not use Tesseract as an OCR tool, and they may have adopted other classifiers. However, their goal is the same; therefore, their results are competent for comparison.

4.5.1 Comparison of Accuracy and F1 Score

Previous research uses Google Drive REST API, Azure Computer Vision API and deep learning to extract text from receipt. Their results would be interesting to compare with the Tesseract OCR in this dissertation. Table 4.4 compares the accuracy and F1 score between the Tesseract OCR and other OCR tools.

Table 4.4: Comparison of accuracy and F1 score obtained from Tesseract with other methods

Method	Reference	Accuracy (%)	F1 Score
Tesseract	-- This Research--	90.72	**0.89**
Google Drive REST API	Odd & Theologou, 2018	88.79	0.88
Azure Computer Vision API		52.55	0.53
Tesseract	Yasser, 2016	89.3	
FineReader		**97.3**	
Deep Learning	Le et al., 2019	72.3	0.71
ICDAR2019 (Average of Top 10 Result)			
	Huang et al., 2019	92.39	0.92

From Table 4.4, FineReader shows the highest accuracy, and the OCR Tesseract from this dissertation shows the highest F1 Score. The most significant difference between these two methods is that FineReader is commercial, while Tesseract is free and open source. FineReader has many built-in features, including image preprocessing and the ability to handle different languages. However, Tesseract can only handle English and requires image preprocessing in advance. Google Drive REST API also provides high accuracy, which ranked after Tesseract and FineReader. However, the Google Drive REST API required an internet connection, which may not be suitable for implementing a standalone model.

According to an ICDAR2019 report, their top 10 results shows accuracy and average F1 score of 92.39% and 0.92, respectively, higher than the results of this study. Most participants in ICDAR2019 are using deep learning to extract text from receipts, which requires a large number of original receipts to obtain higher results. In addition, an expensive graphics processing unit (GPU) is required to train the data model. Deep learning is not in the scope of this dissertation.

4.5.2 Key Information Extraction

Past research proposed pattern recognition, NLP and multi-level information extraction methods to extract key information from receipts (Odd & Theologou, 2018; Yue, 2020; Majumder et al., 2020; Sun et al., 2021). Table 4.5 compares the F1 score of SROIE between these methods and the performance in this dissertation. The highest score of SROIE is highlighted in the table.

Table 4.5: Comparison of F1 Score of SROIE between different methods

Method	Reference	F1 Score		
		Merchant	Transaction Date	Total Price
Pattern Recognition	-- This Research--	**0.93**	0.70	0.60
	Odd & Theologou, 2018		0.51	0.72
NLP	Yue, 2020			0.72
Neural Scoring Model	Majumder et al., 2020		0.85	0.81
Spatial Dual-Modality Graph Reasoning method	Sun et al., 2021	0.75	**0.92**	**0.82**
ICDAR2019 (Average of Top 10 Result)				
	Huang et al., 2019	0.85		

From Table 4.5, the spatial dual-modality graph reasoning method shows the highest F1 score for transaction date and total price. In contrast, the pattern recognition used in this research shows the highest F1 score for the merchant. The spatial dual-modality graph reasoning and neural scoring method are multi-level information extractors, which combine pattern recognition and deep learning to extract the key information from the receipt. Deep learning iterates and searches through all locations for the key information more effectively, improving extraction accuracy and obtaining higher scores.

Besides that, pattern recognition and NLP show acceptable results for key information extraction. The F1 score for transaction date is around 0.51 to 0.70, whereas the F1 score for a total price is around 0.60 to 0.72. The challenges for pattern recognition and NLP are the text extraction from OCR and the expression defined for a specific pattern. If the extracted text is wrong, it will not recognise whether it is the transaction date or the total price by its pattern.

On the other hand, according to the ICDAR2019 report, the average F1 score of the top 10 results is 0.85, which is slightly higher than the results of this study. There are other approaches and ideas used for key information extraction, including NLP and deep learning. Based on the report, more than half of the submitted methods achieve an F1 score of less than 80%. The lack of performance among the variety of approaches, including one in this dissertation, shows the huge gap in improving key information extraction.

4.5.3 Receipt Classifiers

There are seven classifiers used in this research for receipt classification. Some results of receipt classification from previous research are used to compare with the result of this research. Table 4.6 shows the comparison of the F1 score of receipt classifier between different research. Surprisingly, there is only one related research that adopted a machine learning algorithm to classify receipts. Table 4.6 present the scores.

Table 4.6: Comparison of F1 score among ML classifier

Reference	F1 Score						
	NB	**ME**	**SVM**	**L. SVC**	**KNN**	**DT**	**RF**
-- This Research--	0.63	0.63	**0.65**	0.80	0.56	0.52	0.56
Odd & Theologue, 2018	**0.69**		~0.40	**0.94**	**~0.90**	**~0.81**	**~0.81**

From Table 4.6, almost all F1 score of receipt classifiers from other research is higher than F1 score of this research except SVM. The F1 score difference between the research for Naïve Bayes, SVM and linear SVC is tiny, while KNN, decision tree and random forest are large. As discussed in the previous section, the methodology used by this research with other research is slightly different. Other research divides their dataset into training and testing, while the classifier in this research is trained by customising knowledge representation. The customise knowledge representation reflects the practical system for business, which the coverage of receipt's category will increase when increasing the number for knowledge representation. Compared to other research methodology, it required more receipt to train the classifier. If the receipt is limited, then the accuracy will decrease.

4.6 Chapter Summary

In this chapter, the result of OCR, key information extraction and classifier are presented. All results are summarised and presented in a table format, and the results are analyzed and discussed. The chapter also compares the performance of modules developed in this dissertation with reported work in the literature. The next chapter forwards a conclusion and shares some ideas for future work.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion & Limitation

This dissertation explores practical receipt handling for Malaysian employee expense reimbursement through automatic documentation, which involves text extraction, parsing, and classification. The text extraction, parsing and classification problems are commonly known as the SROIE problem. The ICDAR2019 highlighted many observations in the SROIE problem. The first observation is that the quality and condition of the characters in the scanned image affects text extraction accuracy. Their second observation regards identifying the text in the document image by text localization and recognition is not straightforward.

ICDAR2019 outlines a third observation, i.e., understanding scanned receipt is best achieved with a careful use case. Interestingly, the ICDAR2019 have not considered text classification in solving receipt parsing. For example, extract relevant keywords or patterns for the use case accurately. In this dissertation, this gap is explored. The hypothesis is, with text classification, Naïve Bayes, SVM and random forest which are the supervised learning algorithms should be helpful to classify receipts.

In testing the hypothesis, 100 receipts fitting the use case of expense reimbursement for Malaysian employee is collected. The receipts are categorized into six classes, meals, groceries, petrol, accommodation, telecommunication and fares. The receipts are all collected inside Malaysia and written in English. This dissertation does not consider hand-writing on receipt nor address text ambiguity.

Three modules have been developed to address the objectives. Module 1 focuses on improving receipt quality before parsing. Module 2 works on detecting text layout and parsing key information from the receipt into a machine-readable format. The Tesseract OCR is used to fulfil this module. Module 3 addresses the last objective, i.e., classifying the extracted text information to predict the proper receipt categories for the reimbursement report. Module 3 explore seven supervised learning classifiers.

The methodology used in this work for text classification is different compared to other works. Other works usually divide their receipts into two sets, one for training and another one for testing. However, classifiers in this work are trained by a dictionary or knowledge representation. The testing is done against the key information extracted from the receipt. The limitation of this work is the need to carefully prepare the dictionary because it may not cover all related words for different receipt categories. However, updating the dictionary is easy compared to frequently retraining the dataset when a new receipt format enters the dataset. An enterprise without know-how in machine learning can well benefit from this approach.

In conclusion, this dissertation contributed two new datasets, the collection of Malaysian receipts from various resources and the dictionary. This dissertation achieved 90% accuracy at the character level and 78% accuracy at the word level. This work achieves 73% accuracy for the key information extraction and shows that the linear SVC performs best among the seven classifiers in classifying receipt.

In practice, using a dictionary is beneficial. A dictionary is not costly to build, modular and scalable. Businesses do not require special technical skills in extending a dictionary. Significantly, the utility of SROIE in an actual business application, such as in automating the expense reimbursement process, is critical in decreasing error in expense reporting and bring employees claim process up to speed.

5.2 Future work

The idea of proposing a dictionary or knowledge representation in receipt classification is beneficial for a specific use case like automating expense reimbursement for Malaysian employees. Evaluating various receipts in the Malaysian context shows the receipts are getting fancier with advertisements and logos. Most of the local receipts are not as plain as the ones available in the ICDAR2019 database. The Malaysian high humidity weather does not help in preserving physical receipt quality. Often the receipts are susceptible to sun exposure and rainwater damage adding to receipt fading.

The knowledge dictionary is easy to grow compared to retraining a classifier each time a new receipt format is encountered. For example, crowdsourcing can support the growth of the knowledge base for a Malaysian use case. Developing a competent dictionary or corpus will significantly serve the classification performance, especially since the Malaysian receipt is sometimes multilingual and heavy with text ambiguity.

In addition, it is worth exploring the unsupervised learning to receipt classification. Deep learning can handle the diversification of receipts and cluster different types of receipts. Deep learning can overcome the problem that the knowledge dictionary is growing too slow or late. By combining knowledge dictionaries and unsupervised learning, it is believed that the accuracy of receipt classification can be further improved.

REFERENCES

Adriano, J. E. M., Calma, K. A. S., Lopez, N. T., Parado, J. A., Rabago, L. W., & Cabardo, J. M. (2019, February). Digital conversion model for hand-filled forms using optical character recognition (OCR). In IOP Conference Series: Materials Science and Engineering (p. 012049).

Agrawal, R., Gupta, A., Prabhu, Y., & Varma, M. (2013, May). Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In Proceedings of the 22nd international conference on World Wide Web (pp. 13-24).

Aizan, J., Ezin, E. C., & Motamed, C. (2016). A face recognition approach based on nearest neighbor interpolation and local binary pattern. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 76-81). IEEE.

Arici, T., Dikbas, S., & Altunbasak, Y. (2009). A histogram modification framework and its application for image contrast enhancement. *IEEE Transactions on image processing*, *18*(9), 1921-1935.

Arif, H., & Javed, A. (2020, February). An Effective Card Scanning Framework for User Authentication System. In 2020 3rd International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-5). IEEE.

Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based machine learning approach for text and document mining. International Journal of Database Theory and Application, 7(1), 61-70.

Bogach, I. V., & Kovenko, V. A. (2020). Recommendation system based on NLP techniques (Doctoral dissertation, ВНТУ).

Brisinello, M., Grbić, R., Pul, M., & Anđelić, T. (2017, September). Improving optical character recognition performance for low quality images. In 2017 International Symposium ELMAR (pp. 167-171). IEEE.

Bui, Q. A., Mollard, D., & Tabbone, S. (2017, November). Selecting automatically preprocessing methods to improve OCR performances. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 169-174). IEEE.

Chaudhari, K., & Thakkar, A. (2019). Survey on handwriting-based personality trait identification. *Expert Systems with Applications*, *124*, 282-308.

Chaudhary, A., Kolhe, S., & Kamal, R. (2016). An improved random forest classifier for multi-class classification. Information Processing in Agriculture, 3(4), 215-222.

Dalal, M. K., & Zaveri, M. A. (2011). Automatic text classification: a technical review. International Journal of Computer Applications, 28(2), 37-40.

Dhanawade, A., Drode, A., Johnson, G., Rao, A., & Upadhya, S. (2020, March). Open CV based Information Extraction from Cheques. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 93-97). IEEE.

Dhiman, S., & Singh, A. (2013). Tesseract vs gocr a comparative study. *International Journal of Recent Technology and Engineering*, *2*(4), 80.

Diwakar, M., & Kumar, M. (2018). A review on CT image noise and its denoising. *Biomedical Signal Processing and Control*, *42*, 73-88.

Etter, D., Rawls, S., Carpenter, C., & Sell, G. (2019, September). A synthetic recipe for OCR. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 864-869). IEEE.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98-136.

Feichtinger, H. G., & Strohmer, T. (Eds.). (2012). *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media.

Gomez, R., Shi, B., Gomez, L., Numann, L., Veit, A., Matas, J., Belongie, S. & Karatzas, D. (2017, November). Icdar2017 robust reading challenge on coco-text. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 1435-1443). IEEE.

Hackeling, G. (2017). *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd.

Hadi, W. E., Al-Radaideh, Q. A., & Alhawari, S. (2018). Integrating associative rule-based classification with Naïve Bayes for text classification. Applied Soft Computing, 69, 344-356.

Hazra, T. K., Singh, D. P., & Daga, N. (2017, August). Optical character recognition using KNN on custom image dataset. In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) (pp. 110-114). IEEE.

Huang, C. T. (2015). Bayesian inference for neighborhood filters with application in denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1657-1665).

Huang, Z., & Cao, L. (2020). Bicubic interpolation and extrapolation iteration method for high resolution digital holographic reconstruction. *Optics and Lasers in Engineering*, *130*, 106090.

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. V. (2019, September). ICDAR2019 competition on scanned receipt ocr and information extraction. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1516-1520). IEEE.

Hsia, C. H., Hoang, H. G., & Tu, H. Y. (2015, June). Document image enhancement using adaptive directional lifting-based wavelet transform. In *2015 IEEE International Conference on Consumer Electronics-Taiwan* (pp. 432-433). IEEE.

Hwang, W., Kim, S., Seo, M., Yim, J., Park, S., Park, S., ... & Lee, H. (2019, September). Post-OCR parsing: building simple and robust parser via BIO tagging. In Workshop on Document Intelligence at NeurIPS 2019.

ICDAR (2019). The International Conference on Document Analysis and Recognition database is available for download from URL: https://rrc.cvc.uab.es/?ch=13&com=downloads

Itskovich, L., & Kuznetsov, S. (2011, June). Machine learning methods in character recognition. In International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (pp. 322-329). Springer, Berlin, Heidelberg.

Jain, P., & Tyagi, V. (2016). A survey of edge-preserving image denoising methods. *Information Systems Frontiers*, *18*(1), 159-170.

Jha, K., Doshi, A., Patel, P., & Shah, M. (2019). A comprehensive review on automation in agriculture using artificial intelligence. *Artificial Intelligence in Agriculture*, *2*, 1-12.

Jiang, L., Li, C., Wang, S., & Zhang, L. (2016). Deep feature weighting for naive Bayes and its application to text classification. Engineering Applications of Artificial Intelligence, 52, 26-39.

Johannsen, G. (1982). A threshold selection method using information measures. In *Proc. 6th Int. Conf. on Pattern Recogn, Munich, 1982*.

Johansson, E. (2019). Separation and extraction of valuable information from digital receipts using Google Cloud Vision OCR.

Kakani, B. V., Gandhi, D., & Jani, S. (2017, July). Improved OCR based automatic vehicle number plate recognition using features trained neural network. In 2017 8th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-6). IEEE.

Kapur, J. N., Sahoo, P. K., & Wong, A. K. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, *29*(3), 273-285.

Kaur, E. K., & Banga, V. K. (2013). Number plate recognition using OCR technique. *International Journal of Research in Engineering and Technology*, *2*(09), 286290.

Khairnar, J., & Kinikar, M. (2013). Machine learning algorithms for opinion mining and sentiment classification. International Journal of Scientific and Research Publications, 3(6), 1-6.

Kong, A., Nguyen, V., & Xu, C. (2015). Predicting international restaurant success with yelp.

Korobacz, W., & Tabędzki, M. (2018). Preprocessing Photos of Receipts for Recognition. Advances in Computer Science Research.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

Latib, S., Saha, D., & Giri, C. (2021). Retinal Vessel Segmentation Using Unsharp Masking and Otsu Thresholding. In *Proceedings of International Conference on Frontiers in Computing and Systems* (pp. 139-147). Springer, Singapore.

Le, A. D., Van Pham, D., & Nguyen, T. A. (2019, November). Deep learning approach for receipt recognition. In International Conference on Future Data and Security Engineering (pp. 705-712). Springer, Cham.

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015, July). Support vector machines and word2vec for text classification with semantic features. In 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 136-140). IEEE.

Li, Z., & Luo, J. (2011, June). Resolution enhancement from document images for text extraction. In *2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering* (pp. 251-256). IEEE.

Li, Y., Qi, F., & Wan, Y. (2019, December). Improvements on bicubic image interpolation. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Vol. 1, pp. 1316-1320). IEEE.

Majumder, B. P., Potti, N., Tata, S., Wendt, J. B., Zhao, Q., & Najork, M. (2020, July). Representation Learning for Information Extraction from Form-like Documents. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 6495-6504).

Mansoor, K., & Olson, C. F. (2019, December). Recognizing Text with a CNN. In *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)* (pp. 1-6). IEEE.

Maslova, O., Klein, L., Dabernat, D., Benoit, A., & Lambert, P. (2019, September). Receipt automatic reader. In 2019 International Conference on Content-Based Multimedia Indexing (CBMI) (pp. 1-6). IEEE.

Milanfar, P. (2012). A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, *30*(1), 106-128.

Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. International journal of recent technology and engineering (IJRTE), 2(1), 72-75.

Nagaoka, Y., Miyazaki, T., Sugaya, Y., & Omachi, S. (2017, November). Text detection by faster R-CNN with multiple region proposal networks. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 6, pp. 15-20). IEEE.

Nirmala, J. S., Banerjee, R., & Bharadwaj, R. S. (2020). Automatic Vehicular Number Plate Recognition (VNPR) for Identification of Vehicle Using OCR and Tesseract. In Micro-Electronics and Telecommunication Engineering (pp. 403-411). Springer, Singapore.

Odd, J., & Theologou, E. (2018). Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms.

Otsu, N. (1979). A threshold selection method from gray level histogram, IEEE Transactions in Systems, Man, and Cybernetics, vol. 9, pp. 62-66

Palawancha, N. (2012). Online employee expense management application.

Pandey, A., Sharma, V., Paanchbhai, S., Hedaoo, N., & Zade, S. D. (2017). Optical character recognition (ocr). International Journal of Engineering and Management Research (IJEMR), 7(2), 159-161.

Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019, September). CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In Workshop on Document Intelligence at NeurIPS 2019.

Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: A case study. International Journal of Computer Applications, 55(10), 50-56.

Patra, A., & Singh, D. (2013). A survey report on text classification with different term weighing methods and comparison between classification algorithms. International Journal of Computer Applications, 75(7).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Peng, J., & Ford, C. O. (2014). Fraudulent expense reporting: impact of manager responsiveness and social presence. *Journal of Applied Accounting Research*.

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, *5*(2), 221.

Rangsikunpum, A., Leelarasmee, E., & Pumrin, S. (2017, June). A design of sign video image expander for hdmi source using bicubic interpolation. In *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 171-174). IEEE.

Rigaud, C., & Burie, J. C. (2019, September). What do we expect from comic panel extraction?. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)* (Vol. 1, pp. 44-49). IEEE.

Rosebrock, A. (2018). OpenCV text detection (EAST text detection). Extracted from URL: https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/

Sabab, S. A., Islam, S. S., Rana, M. J., & Hossain, M. (2018, September). eExpense: A smart approach to track everyday expense. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 136-141). IEEE.

Saddami, K., Munadi, K., Away, Y., & Arnia, F. (2019). Improvement of binarization performance using local otsu thresholding. *International Journal of Electrical and Computer Engineering*, *9*(1), 264.

Sakila, A., & Vijayarani, S. (2017). Image Enhancement using Morphological Operations. *International Journal of Scientific Research in Science, Engineering and Technology*, *3*(2), 685-698.

Saldivar, J., Vairetti, C., Rodríguez, C., Daniel, F., Casati, F., & Alarcón, R. (2016). Analysis and improvement of business process models using spreadsheets. *Information Systems*, *57*, 1-19.

Sangwine, S. J., & Horne, R. E. (Eds.). (2012). *The colour image processing handbook*. Springer Science & Business Media.

Sharma, A., & Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. In Proceedings of the 2012 ACM research in applied computation symposium (pp. 1-7).

Singh, G., Kumar, B., Gaur, L., & Tyagi, A. (2019, April). Comparison between multinomial and bernoulli Naïve Bayes for text classification. In 2019 International Conference on Automation, Computational and Technology Management (ICACTM) (pp. 593-596). IEEE.

Singh, J., Singh, G., & Singh, R. (2017). Optimization of sentiment analysis using machine learning classifiers. Human-centric Computing and information Sciences, 7(1), 32.

Srivastava, S. K., Singh, S. K., & Suri, J. S. (2019). Effect of incremental feature enrichment on healthcare text classification system: A machine learning paradigm. *Computer methods and programs in biomedicine*, *172*, 35-51.

Suponenkovs, A., Sisojevs, A., Mosāns, G., Kampars, J., Pinka, K., Grabis, J., ... & Taranovs, R. (2017, November). Application of image recognition and machine learning technologies for payment data processing review and challenges. In *2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (pp. 1-6). IEEE.

Sun, H., Kuang, Z., Yue, X., Lin, C., & Zhang, W. (2021). Spatial Dual-Modality Graph Reasoning for Key Information Extraction. arXiv preprint arXiv:2103.14470.

Sun, Y., Li, Y., Zeng, Q., & Bian, Y. (2020, April). Application Research of text classification based on random forest algorithm. In 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (pp. 370-374). IEEE.

Tafti, A. P., Baghaie, A., Assefi, M., Arabnia, H. R., Yu, Z., & Peissig, P. (2016, December). OCR as a service: an experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym. In International Symposium on Visual Computing (pp. 735-746). Springer, Cham.

Tejas, B., Omkar, D., Rutuja, D., Prajakta, K., & Bhakti, P. (2017, June). Number plate recognition and document verification using feature extraction OCR algorithm. In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1317-1320). IEEE.

Torres, P. M. (2017, September). Text recognition for objects identification in the industry. In *International Conference of Mechatronics and Cyber-Mixmechatronics* (pp. 126-131). Springer, Cham.

Tripathy, A., Agrawal, A., & Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques. Procedia Computer Science, 57, 821-829.

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117-126.

Tripathy, A., & Moorthy, K. (2020). Satisfaction Level of Employees about Expense Reimbursement. Studies in Indian Place Names (UGC Care Journal), 40(13), 1232–1238.

Ullah, R., Sohani, A., Ali, F., & Rai, A. OCR Engine to extract Food-items and Prices from Receipt Images via Pattern matching and heuristics approach.

Vaithiyanathan, D., & Muniraj, M. (2019, December). Cloud based Text extraction using Google Cloud Vison for Visually Impaired applications. In 2019 11th International Conference on Advanced Computing (ICoAC) (pp. 90-96). IEEE.

Wang, S., & Yang, K. J. (2008). An image scaling algorithm based on bilinear interpolation with VC++. *Techniques of Automation and Applications*, *27*(7), 44-45.

Xie, D., & Bailey, C. P. (2020, April). Novel receipt recognition with deep learning algorithms. In Pattern Recognition and Tracking XXXI (Vol. 11400, p. 114000B). International Society for Optics and Photonics.Journal of Engineering and Management Research (IJEMR), 7(2), 159-161.

Xu, S. (2018). Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science, 44(1), 48-59.

Yang, Q. (2016). Semantic filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4517-4526).

Yasser, A. (2016). Classifying receipts and invoices in visma mobile scanner.

Ye, Y., Wu, Q., Huang, J. Z., Ng, M. K., & Li, X. (2013). Stratified sampling for feature subspace selection in random forests for high dimensional data. Pattern Recognition, 46(3), 769-787.

Ye, Y., Zhu, S., Wang, J., Du, Q., Yang, Y., Tu, D., Wang, L. & Luo, J. (2018, December). A unified scheme of text localization and structured data extraction for joint OCR and data mining. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2373-2382). IEEE.

Yousefi, J. (2011). Image binarization using otsu thresholding algorithm. *Ontario, Canada: University of Guelph*.

Yue, A. (2020). Automated receipt image identification cropping and parsing. Princeton. edu.

Yin, Y., Zhang, W., Hong, S., Yang, J., Xiong, J., & Gui, G. (2019). Deep learning-aided OCR techniques for Chinese uppercase characters in the application of Internet of Things. IEEE Access, 7, 47043-47049.

Ziegaus, M. (2016). Optical Character Recognition on supermarket receipts. Thesis submitted to The University of Passau, Germany.

Zhu, G., Bethea, T. J., & Krishna, V. (2007, August). Extracting relevant named entities for automated expense reimbursement. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1004-1012).