International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Aug-2023)

http://dx.doi.org/10.12785/ijcds/140132

Digitizing Receipts with OCR

Ooi Ming Yeung 1[1], Dr. Justtina Anantha Jothi AP C.M.E John 2[2] and Dr. Nursakirah Binti Ab Rahman Muton 3[3]

[1,2,3]*Department of Computing, UOW Malaysia KDU Penang University College, Pulau Pinang, Malaysia*

Received 02 Aug. 2022, Revised 06 May. 2023, Accepted 09 May. 2023, Published 01 Aug. 2023

Abstract: Receipt digitization carries numerous advantages over traditional paper receipts in terms of preservation. Saving images of receipts is a method of digitizing receipts, but managing (sorting, searching, editing, deleting, adding annotations) them is rather tedious. This paper analyzes current receipt managing applications in the market, and their approach in digitizing receipts. Other researchers' methods were also analyzed, and a pipeline to achieve receipt digitization is deduced. With a pipeline of various methods in each step, the strengths and flaws of each method were addressed. Improvements to those methods and to the overall usability of a receipt management system were proposed. The proposed methods were then developed into a usable product application.

Keywords: Optical Character Recognition, OCR, Field extraction, Information Extraction, Abbreviation Disambiguation, Image processing

1. INTRODUCTION

The preservation of paper documents by storing and managing them is a common practice albeit being tedious. This is no exception to receipts, where they are being printed abundantly every day even though they could be generated and sent electronically. By the small and disposable nature of receipts, storing or managing them is a hassle, despite carrying essential information. This issue is exacerbated by the low quality of paper receipts, making them unfit for long-term storage in the first place. The fading of ink, smudging of the paper and wrinkles will cause potential loss of information on the receipt. Following the same direction as e-books, archiving receipts digitally is a solution to those issues. Not only can they be stored indefinitely, managing them is made simpler, and various insights can be produced from the receipt data too. This benefits everyone from individuals to companies, for many different purposes such as budgeting, tax filing or consumer behavior analysis [1]. With the Optical Character Recognition (OCR) technology becoming more accessible and accurate, it can be applied for digitizing receipts while requiring minimal human intervention for tasks like data entry. This paper aims to investigate existing receipt digitizing applications in the market and the methods to produce those applications. Based on the results from the investigation, the best approach is used to develop a standalone receipt management system with built-in OCR, information extraction and abbreviation disambiguation features for the automation of receipt saving. The application is developed purely on Python programming language and can be deployed on any machine that is capable of running Python. The management component of this application includes searching, adding, deleting, sorting, and editing of receipts. The receipt data is saved locally on the machine, in a JSON file.

OCR is responsible for converting the text in images into digital text. Information extraction utilizes the digital texts and their corresponding coordinates to classify the text into fields like address, phone number, item names, prices, and dates. Lastly, abbreviation disambiguation is applied for unrecognizable item names, where they are queried in a search engine, and the results are saved to be used during searching.

The receipt management applications present in the current market have several fatal flaws. While some of them implement OCR to extract words from receipt images, their capability to classify them into fields is limited. Their field classification only extends to basic fields like address, total and date. To overcome this limitation, some applications just provide empty fields for users to manually input their receipt information. This demonstrates the inability of those applications to fully utilize OCR for field classification. In fact, no applications provided a field for individual item names and prices. Abbreviation disambiguation is naturally required with the inclusion of item names as a receipt field, because their names are printed in short due to the lack of horizontal receipt space. Since users do not typically remember short form item names, such a feature facilitates the searching of specific items within a plethora of saved receipts. The features included in this application will address and fill the gaps of existing applications.

This application is developed purely in the Python programming language. The tools used for each step in the application pipeline is specified. The graphical user

E-mail address: 0203712@student.kdupg.edu.my, justtina@kdupg.edu.my, nursakirah.rm@kdupg.edu.my

interface (GUI) is built using the Tkinter library. For image manipulation purposes as the preprocessing step, OpenCV and Pillow libraries are used. Google libraries are used to connect the application to the Google OCR engine API and Search Engine Results Page (SERP) API. From the OCR engine output, calculations involving the coordinates of texts use the Shapely library and Regex library is used to classify the text into receipt fields. For abbreviation disambiguation, before sending the item names to the SERP API as a query, the words are checked if they exist in the English dictionary using the Enchant library.

The development of this application benefits several parties, from general users to company employees. Being a general purpose receipt manager, it expedites any process of financing activities that require the use of receipts, especially with the abbreviation disambiguation feature. Meanwhile, features like OCR and information extraction are to speed up the uploading of a new receipt itself.

2. BACKGROUND

This chapter analyses current findings surrounding the pipeline of a receipt scanner application: optical character recognition (OCR), information extraction and abbreviation disambiguation. Firstly, evaluation is done on existing applications with similar concepts, where OCR is utilized on receipts. Then, possible methods and tools to produce each component from the pipeline are researched. Problems that may arise during the pipeline are also addressed with potential solutions. The feasibility of those solutions being implemented in this application are analyzed as well. Through such analysis, the knowledge gaps in this field can be identified which justifies the rationale of this application.

A. *Existing Applications*

Six receipt scanning applications from the Google Play Store were evaluated: Foreceipt - Receipt Scanner & Expense Tracker Cloud [2], Expensify - Expense Reports [3], Receipt Scanner: Easy Expense [4], Veryfi Receipts OCR Expenses [5], Dext Invoice Expense Reports [6], and Receipt Scanner Expense Tracker by Saldo Apps [7]. While the optical character recognition (OCR) features of all the applications were accurate on average, each of them has different information extraction capabilities. The information which are automatically extracted from each application are summarized in the following table:

Fields that are provided but require users to manually input are not included in the table. Veryfi had the most extensive information extraction capabilities, with the most number of fields being automatically filled.

However, none of these applications have automatically extracted individual item names and prices, nor provided a field for them. This is a fatal flaw in a receipt management system because it does not satisfy an expected use case using item prices from past receipts for price comparison. Unless the user knows the specific date or store name, an impractical solution is by viewing a receipt image (a feature available in all evaluated applications) then searching for the item name and repeating until the item is found. This application provides a more ideal process. After uploading receipt images and saving them as receipt entries, the user can search for a particular receipt entry by an item name, select the receipt, then indicate the price of the item within the receipt entry. Although the exact methods of each application in extracting information are not shared publicly, their features and shortcomings serve as a benchmark for this application.

B. *Receipt Scanning Pipeline*

The receipt scanning pipeline is discussed in 3 separate sections: Optical Character Recognition (OCR), Information Extraction and Abbreviation Disambiguation (AD). Methodology proposed and experimented by past research are presented and compared in appropriate sections. Tools and issues related to each section are addressed as well. Basic components of a receipt management system like receipt entry saving, editing, searching, deletion and viewing are not included in this section.

1) *Optical Character Recognition (OCR)*

Optical Character Recognition (OCR) is used to extract texts from receipt images. Although an OCR can be built from scratch [8], the simpler and more practical approach is by using a pre-trained OCR engine. Google Cloud Vision (GCV) [9], [10], [11], [12] and Tesseract [13], [14], [15], [16] are two of the most prominent OCR engines implemented on receipts. Tesseract is a free open source engine whereas GCV provides a limited number of calls for free. According to an evaluation done by [17], GCV achieved a higher text detection accuracy on scanned documents compared to Tesseract. Similarly, GCV had a higher character accuracy on business documents than Tesseract, as presented by [18]:

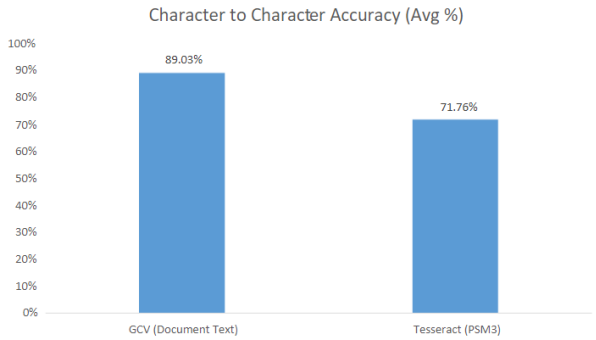


Figure 1. Comparison of GCV and Tesseract character accuracy [18]

This may be due to Tesseract's strict requirement of a minimum 300 dpi image and heavy reliance on preprocessing to improve the clarity of image before input (tesseract-ocr, n.d.).

Since there are many factors affecting the clarity of receipt images (e.g. thin fonts, lighting, image resolution, folds, wrinkles, smudges, faded inks), the images have to undergo preprocessing to improve OCR accuracy. Albeit proposed by different authors, the preprocessing steps of document images/scanning generally share the following stages and flow:

Tabelle I. Information extraction capabilities of different applications

	Foreceipt	Expensify	Easy Expense	Veryfi	Dext	Saldo Apps
Date	✓	✓	✓	✓	✓	✓
Time				✓		
Store name	✓	✓		✓	✓	✓
Address						
Item names						
Tax			✓	✓	✓	
Tip				✓		
Subtotal				✓		
Total	✓	✓	✓	✓	✓	✓

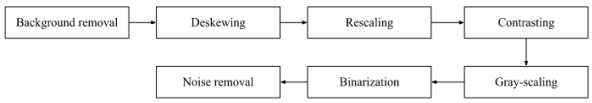


Figure 2. Preprocessing pipeline

Background removal eliminates irrelevant objects which may be picked up as characters by the OCR engine. Deskewing ensures that the words in the receipt are in the correct orientation. Both of these steps can either be done manually by the user, or an algorithm. The Canny edge detection algorithm removes noise using Gaussian filtering, then calculates the gradient intensity of every pixel for comparison with neighboring pixels. A large difference in gradient intensity indicates a potential edge. Based on the gradient value in the double-threshold detection, the real edge is determined if it exceeds the threshold [19]. This algorithm is typically followed by Hough transform for deskewing. The shape of a receipt is passed as a parameter, and lines in the image that match the shape are iteratively searched for [20]. From a successful iteration, the angle of rotation of shape is obtained. Using that information, the image is rotated so the words are perfectly upright. The image size after background removal may decrease which indirectly speeds up the OCR process [15].

Each OCR engine has their own minimum image size: 300 dots-per-inch (dpi) for Tesseract [21] and 1024x768 pixels for GCV [22]/ During rescaling, the image is resized to the minimum recommended size if it is below that, while maintaining its aspect ratio. Optionally, images with extremely large resolution can shrinked to speed up the OCR process. While the process of image resizing is straightforward, a suitable algorithm must be chosen to preserve the image quality, so no information is lost.

Contrasting, grey-scaling and binarization are stages used to increase the clarity of receipt texts in different lighting conditions. Contrasting brightens the lighter regions, and darkens the darker regions, facilitating the differentiation between the receipt paper and ink. Grey-scaling is an intermediate stage between contrasting and binarization. Grey-scaling eliminates harsh colour variations produced by contrasting. It improves the efficiency of binarization in terms of speed and output quality [8].

Binarization or thresholding converts the image to black and white. A pixel is converted to white if the pixel value is above the algorithm-produced threshold or converted to black if the pixel value is below the threshold. The three types of thresholding algorithms are simple, adaptive, and Otsu's thresholding. Although some OCR engines like Tesseract have built-in Otsu thresholding, thorough preprocessing is still recommended. Simple thresholding uses a global threshold value chosen by the user, while adaptive thresholding calculates different threshold values for different areas of the pictures. On the other hand, Otsu's thresholding automatically obtains the most suitable global threshold value to separate the background and foreground [23]. The choice of algorithm is situational, but the fact that lighting conditions within a receipt tend to vary, has to be considered.

There are three processes within the noise removal stage: dilation, erosion, and blurring. Dilation thickens the fonts on receipts, while erosion makes them thinner. These processes are essential to separate characters that are too close together, or to identify glitchy fonts due to improper printing [24]. Noises are very detrimental to the accuracy of OCR on receipts because they may be detected as a symbol. Hence, denoising using Gaussian blurring or median blurring are applied.

An image of a long receipt will naturally have small fonts. Instead of enlarging the image during preprocessing, separating the receipt into multiple images can preserve the clarity of receipt. They can then be recombined in the image stitching stage.

Depending on the accuracy of the OCR engine used, some preprocessing stages can be skipped or the order of executing them may be different. This is to prevent the image from being over-processed, causing the loss of important information. The algorithm used at each stage must be carefully selected, to strike a balance between image quality, OCR accuracy, processing time and power, while accommodating various image conditions. For similar reasons, the parameters for each algorithm must be optimally tuned as well.

To compare the OCR accuracy between Tesseract (Tes) and GCV, three receipts have been selected. The receipts are manually cropped and deskewed, then they are all preprocessed with the same algorithm. The raw and preprocessed versions of the receipts are input into both engines and their accuracies are tabulated. The metric used is Levenshtein distance which counts the total transformation needed for the tested output to mimic the

ground truth. Types of transformations include deletion, insertion and substitution and each transformation has a weight of [25]. Higher number of total transformations indicates a higher degree of inaccurate OCR detection. The Levenshtein distance by comparing each engine output and each image type to the ground truth is:

Tabelle II. Total Levenshtein distance of Tesseract and GCV

Receipts	Normal	Wrinkled	Smudged
Tes without preprocessing	245	429	163
Tes with preprocessing	196	120	252
GCV without preprocessing	15	16	5
GCV with preprocessing	29	28	94

Figure 3. Normal receipt

Figure 4. Preprocessed normal receipt

Figure 5. Smudged receipt

Figure 6. Preprocessed smudged receipt

Figure 7. Wrinkled receipt

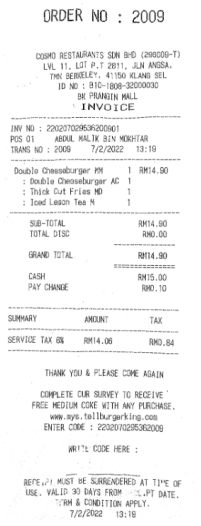


Figure 8. Preprocessed wrinkled receipt

On average, the best performing engine is GCV without any preprocessing. GCV performs better without

preprocessing because GCV does its own preprocessing internally [26]. Performing preprocessing multiple times on an image may erase important information. On the other hand, the only preprocessing done by Tesseract is Otsu's thresholding [21]. Tesseract performs better with preprocessing than without, but its accuracy still lacks behind GCV. For all cases, the ECE % of preprocessed images are higher than their raw versions. This is due to the preprocessing algorithm failing to completely eliminate the speckles and noises on receipts. Hence, they are incorrectly recognized as characters by the OCR engines. For preprocessed images with edges present between the receipt paper and the background, Tesseract recognized them as random characters.

As a cloud API, GCR naturally requires an internet connection and a longer time to produce an output compared to the offline engine, Tesseract. Although the formatting of words is not considered as any error, the formatting of Tesseract was better than GCV. Correct formatting is essential for the information extraction step. Tesseract's outputs were perfectly arranged into lines, whereas GCV struggled to do so in cases where there were multiple 'columns'. Such cases are common in the item names and price section of receipts. Therefore, to implement GCV, a separate algorithm has to be deployed to rearrange the detected words as presented in the receipt.

2) Information Extraction

Information extraction is a unique problem for structured documents such as receipts. Texts extracted using an OCR engine may contain noise and lack any meaning or context. Therefore, an algorithm is required to extract relevant information based on the specific application. Proposed algorithms can be categorized into rule-based or machine learning.

Templating, proposed by [15] is an example of a rule-based system. Based on the receipt layout structure from each store, a template containing individual field coordinates and their relative positions is saved. To predict fields on a new receipt from a known store, the length and height ratio of each field in the overall receipt from the template is used. Because the receipt structures are known beforehand, these rules can be as specific as possible, so high accuracy of field extraction is easily achieved. However, scalability is an issue because not all receipts have the same structure. So, this method would not work if the system encountered a never-before-seen receipt as the template does not exist in the database.

Albeit different names, Convolutional Universal Text Information Extractor (CUTIE) [27], Skip-Rect Embedding (SRE) [28] and Chargrid [29] are machine learning models which share a similar concept, where the relationship of spatial information and textual formats are utilized. These models first derive spatial information of each word using their bounding boxes, or by mapping them onto a grid. The words are encoded (using Char2Vec, Word2vec or one-hot encoding) along with their spatial information then input into a Convolutional Neural Network. The final step is decoding, where the output are all the words being categorized as a receipt field. BERTgrid, proposed by [30], improved these models by passing the words through a language model to obtain their contextualized vectors before continuing the similar encoding-decoding pipeline. Another variation of such model is BIO tagging. All the words present in the receipt are serialized into traditional text formatting, to preserve their contextual information [31]. The words are then tagged with their line-group embedding, coordinates and positions for encoding.

A machine learning model using a Graph Convolution Network (GCN) was proposed as well [32]. All the words in the receipt are first categorized as boolean features based on their potential fields: dates, known cities, known countries, or zip codes. They are also categorized based on their attributes: only contains numbers, only contains alphabets, contains alphabets and numbers, contains decimal point, or any combination. All the words are encoded, and the relative distance of every word to the nearest word in four directions (top, bottom, left, right) is also calculated. To model the graph, the nodes represent each word, and the edges represent their relative positions. Instead of connecting all the words to each other, only the nearest words in four directions are connected because it is less computationally expensive. Then, the graph is passed into a GCN where the output are nodes being classified as receipt fields.

The major advantage to machine learning approaches is being able to extract information from any generic receipts but deploying them involves multiple layers of complexity. It requires a large dataset of receipts during training and testing, but even then, the accuracy is not guaranteed as it depends on how well-designed and tuned the machine learning algorithm is.

[33] has done extensive comparison of these machine learning and rule-based algorithms:

Class/Model	Rule Based	LSTM	BERT	GCN	Oracle
VENDOR	0.455	0.078	**0.677**	0.222	0.791
DATE	**0.923**	0.814	0.842	0.297	0.916
ADDRESS	**0.427**	0.075	0.344	0.072	0.574
TAX RATE	0.697	0.801	**0.985**	0.410	0.985
PRICE	**0.833**	0.655	0.818	0.022	0.900
CURRENCY	**0.926**	0.875	0.885	0.628	0.895
PRODUCTS	**0.127**	0.037	0.027	0.067	0.404
MICRO AVG	**0.515**	0.278	0.455	0.167	0.660
MACRO AVG	**0.710**	0.520	0.653	0.305	0.781

Figure 9. Accuracy comparison of different models in field extraction [33]

Oracle represents the accuracy of an error-prone OCR output with perfect field extractions, compared to an errorless OCR field extraction. This is used to measure the percentage of incorrect field extractions caused by OCR errors. Surprisingly, rule-based models achieved the highest accuracy in extracting all the fields except tax rate and vendor. The machine learning algorithms yielded poor accuracies despite being trained with 790 receipts. This disputes the effectiveness of deploying a machine learning algorithm for information extraction purposes.

3) Abbreviation Disambiguation (AD)

To conserve horizontal space in a receipt, item names are sometimes abbreviated. Therefore, abbreviation disambiguation (AD) is a necessary feature for this application as users rarely remember the exact item names as printed on receipts. [34] addressed this issue by matching short form names with their long form counterparts using a dictionary, given that the naming conventions of stores are known. [35] tackles AD with a rather unconventional yet complicated approach. It attempts to generate a digital receipt on the online store, which mimics the purchases made on the physical receipt. Then, using web scraping, the item names on the physical receipt are matched with the item names in the digital receipt for AD.

Although aimed for text-dense documents, the acronym disambiguation approach proposed by [36] has potential. A dictionary is created by scraping multiple websites which are dense in acronyms and their corresponding long forms (e.g. arXiv, Reddit, Wikipedia). With such a large dictionary and ambiguity, it is inevitable that many different long forms can be obtained from a single acronym. To better understand the context of the text, a supervised model is trained by inputting the whole text and the target acronym position. Then, the most probable long forms are obtained as the output.

[37] proposed two machine learning models for AD in medical texts: Support Vector Machines (SVM) and Convolution Neural Network (CNN). For training, both models use part-of-speech (POS) embeddings, and encoded words as their features; the SVM model encodes words using one-hot encoding while the CNN uses specifically trained word embeddings. A feature unique to the SVM model is the previous and next three words of an abbreviation. Meanwhile, other features of the CNN model include positional and section embeddings.

Some approaches reviewed may not be feasible in generic receipt parsing due to these factors:

- Local dictionaries do not work on receipts by a never-before-seen store. For example, an abbreviation dictionary for store A will only work for store A but not on any other stores.
- Heterogenous item naming conventions. There is not one universal convention shared between different stores.
- Lack of receipt item abbreviation and long form databases for training.
- The store generating physical receipts does not have an online store, making it hard to find online generated versions of the receipt.
- Words in receipts do not obey grammatical rules, making POS tagging feature unusable.

3. DECISION OF ALGORITHM

Many state-of-art approaches involved in the receipt scanning pipeline are reviewed. They provide a guideline for the choice of algorithms or models implemented in this application. The ultimate goal is to propose an approach which capitalizes on the benefits while mitigating the drawbacks as much as possible. There are many factors in the receipt pipeline that have to be considered when proposing an approach: the initial receipt condition, the lighting of receipt image, the effectiveness of preprocessing algorithms, the OCR engine, and finally the information extraction and abbreviation disambiguation algorithm deployed. These factors may be the bottlenecks of the system if algorithms are not carefully selected and implemented.

Possible modifications on the reviewed algorithms are proposed for improvements and specialization in the use case of this application. This application emphasizes on the use of OCR to minimize human intervention in digitizing receipts, making accuracy an important element. Thus, referring to the accuracy comparisons in Table b, GCV is selected to be applied on minimally preprocessed images. Likewise, the selection of information extraction algorithm is based on the accuracy comparison in Figure 9. Being the algorithm with the most promising accuracy, a generic rule-based algorithm is implemented to extract information from a variety of receipts.

Because AD is context-dependent, there is no definitive solution proposed that can be directly implemented into this application. Despite that, the web-scraping approach by [36] is modified to be specialized for receipt item names. Since AD in receipts is only used in the search function, the disambiguated words are not visible or displayed to the user. As a back-end function, the ability to successfully retrieve potentially relevant (and irrelevant) receipts based on the searched term is prioritized, rather than not yielding any results when the searched term is completely relevant. In other words, the extent of irrelevant long form words associated with a receipt is forgiving.

After detecting item names in the information extraction stage, each word is checked with an English dictionary. If the word does not exist in the dictionary, it is automatically input into a Search Engine Results Page (SERP) API. The modification lies in utilizing SERP API to selectively scrape a search engine results only when it is needed, instead of scraping multiple predefined web pages to obtain a large dictionary. This approach exploits the wide scope of search engines along with their efficient autocorrect features, and result relevancy rankings. With the wide-spread accessibility of internet connection in devices, the requirement of internet connection for GCV OCR API and SERP API would not pose issues.

4. IMPLEMENTATION

This chapter discusses how each component of this application is implemented in terms of code. The components include image processing, OCR, information extraction, abbreviation disambiguation, and database.

A. Image Processing

Since GCV has achieved a higher accuracy without preprocessing, only the first three stages in the proposed preprocessing pipeline are implemented as recommended on GCV's documentation [22].

The first and second preprocessing steps, rotating and cropping, are achieved through the GUI which is based on the judgement of the user. The user is allowed to flip (horizontally and vertically) using the buttons and rotate the image using the scale provided. It is followed by image cropping where the user is required to move the sizable red rectangle to crop out the receipt background. The rotating and cropping of image are shown in Figure 10 and Figure 11 respectively:

Figure 10. Rotate image

Figure 11. Crop image

The relative coordinates of the rectangles in the GUI canvas are converted into actual coordinates (on original image) by calculating the ratio of the original image resolution to the canvas image resolution, which are then used for cropping on the original image. The date and time when the new receipt image is saved, is used as a unique identifier. With the GCV documentation listing the minimum image size as 1024 x 764 pixels, images smaller than that are enlarged accordingly.

B. *Optical Character Recognition (OCR)*

Since it is not practical to design and train an OCR model from scratch, a pre-trained OCR engine is selected. The engine selected in this application is GCV, which is built on the Tensorflow framework [38]. The API returns a dictionary containing the detected text and the coordinates of each detected character, word, block, paragraph, and page.

As highlighted before, the text from GCV is not completely organized. An image snippet in Figure 12 and the corresponding GCV output in Figure 13 demonstrates that.

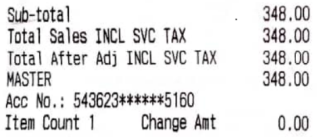


Figure 12. Receipt image snippet

```
Sub-total
Total Sales INCL SVC
TAX
Total After Adj INCL
SVC TAX
MASTER
Acc No.:
543623******5160
Item Count 1 Change Amt
348.00
348.00
348.00
348.00
0.00
```

Figure 13. GCV output snippet

To solve this, an algorithm using the coordinates of each word is designed as illustrated in Figure 14. For each word, a line (red) is defined from the centre of the word to the midpoint of the top left and bottom left coordinates. Then, the line (red) is extrapolated from the centre of the word to the left edge of the receipt image. Another line (blue) from the word's top right and bottom right coordinate is defined. The shortest intersection between the blue and red line indicates that the two words are side to side with each other.

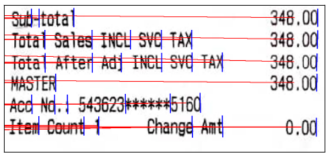


Figure 14. Algorithm to reorganise GCV output

C. *Information Extraction*

The receipt phone number, date and time are extracted purely by regular expression (regex) pattern matching. The address uses regex too but with a database of places in Malaysia. If the word 'Malaysia' is not found, the regex searches for a match with a database of Malaysian states and if it is not found, the regex searches for a match with a database of Malaysian cities. If a match is found, the line of where the word is found is the last line of the address. If the last line lies in the top half of the receipt, the first line of address is the first line of the whole receipt. If the last line lies at the bottom half of the receipt, the first line

of the address is the first line of the 'block' (containing the last line) defined by the GCV output. The full address is all the words from the first to the last line, but any lines that overlap with the phone number's line, date's line or time's line are skipped.

In the first step to identify prices, the maximum price is searched using regex, where the value is the highest, it lies at the bottom right quadrant of the receipt, and the last digit is a zero (final prices are always rounded up to the nearest tenth in Malaysia). The leftmost x-coordinate of the maximum price is used as a threshold line (blue) to avoid misidentifying quantities and item ids as prices. The x-coordinate is further reduced by 20% of the image width as a margin of error in case the receipt is not perfectly upright.

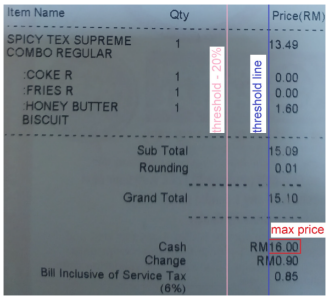


Figure 15. Algorithm to detect prices

Any price detected by regex that falls beyond the threshold (pink) line becomes an item price, and the words to its left is the item name. To handle cases where an item occupies more than one line, by default, items without any price directly to their right, belong to the nearest price at the top right. To check if an item belongs to the bottom right price, the application searches for a potential price header. If there is a gap between the header line and the first price line, items that do not have prices to their right belong to the closest bottom right price.

1) Information Extraction Accuracy Testing

The metric proposed by [33] is used for information extraction accuracy calculation. The accuracy of OCR is not included in this section as it has been addressed in section 2.3.1. It should be noted that the sample size is small (17 receipts) and testing on more receipts with different types of structure is required to better reflect the information extraction accuracy. For all fields except price and items, their accuracies are binary: 1 if it matches the ground truth, 0 otherwise. The individual accuracies for prices and items are calculated in such manner: $\frac{\textit{total correct prices}}{\textit{total prices in ground truth}}$. The accuracies from each receipt and fields are then averaged and tabulated:

Tabelle III. Average Field Extraction Accuracy

Fields	Accuracy (%)
Address	88.235
Phone number	61.538
Date	88.235
Time	100.00
Price	96.172
Item	80.756

The time field has the highest accuracy due to its unique formatting using colons, making it easily recognizable. With similar reasons, the date field has a high accuracy because of its distinct separators like dashes and forward slashes. Address field has a high accuracy too, given that the user follows the guideline provided during cropping. The phone number field has the lowest accuracy as there is a large variety of phone number formats in Malaysia. The application had difficulty in differentiating between phone numbers and long number strings such as item ids and service tax ids. The price field accuracy depends on the recognition of prices and the correct number of total prices detected. In contrast, the item field requires the item names and their corresponding prices to be correct, which explains the accuracy gap between the item field and price field. The dynamic formatting of item names in receipts further reduced the item field accuracy. Item names which occupy more than one line is the major cause of the ambiguity of whether they belong to the price above or below them.

D. Abbreviation Disambiguation (AD)

For each word in item names that is not found in the English (Enchant library) dictionary, the word is passed to the SERP API. The title, link, and snippet of the first 5 search results are appended to the full text of receipt (only used for searching).

5. DISCUSSION

With the consideration that the extracting capabilities are not 100 percent accurate at all times, the final application allows users to edit the prefilled receipt entries. While the OCR, information extraction and abbreviation disambiguation accuracies are satisfactory, there is big room for improvement. The field extraction rules implemented in the final application do not encompass all types of receipts. Namely, localized rules such as currency name, phone numbers and addresses. More research into engines specifically trained for receipts can be done. Possibly, the engine can carry out OCR and information extraction simultaneously. Although this recommended approach was proven unsuccessful, the use of different models can be explored. Such exploration is beneficial as an alternative for the rigid rule-based field extraction. Meanwhile, the abbreviation disambiguation feature is a hit or miss as it relies on the search engine and its results ranking. Irrelevant words may also be associated with a receipt because of inaccurate search results.

Literatur

[1] A. Duc Le, D. Van Pham, and T. Anh Nguyen, "Deep learning approach for receipt recognition," 2019. [Online]. Available: https://www.researchgate.net/publication/333505533_Deep_Learning_Approach_for_Receipt_Recognition

[2] F. Inc, "Foreceipt - receipt scanner expense tracker cloud," Google.com, 2013. [Online]. Available: https://play.google.com/store/apps/details?id=com.foreceipt.android.cloud&hl=en_US&gl=US

[3] E. Inc, "Expensify - expense reports," Google.com, 2013. [Online]. Available: https://play.google.com/store/apps/details?id=org.me.mobiexpensifyg&hl=en_US&gl=US

[4] E. E. Tracker, "Receipt scanner - easy expense," Google.com, 2013. [Online]. Available: https://play.google.com/store/apps/details?id=com.easyexpense&hl=en_US&gl=US

[5] Veryfi.com, "Veryfi receipts ocr expenses," Google.com, 2013. [Online]. Available: https://play.google.com/store/apps/details?id=com.iqboxyinc.iqboxy&hl=en_US&gl=US

[6] Dext, "Dext invoice expense reports," Google.com, 2013. [Online]. Available: https://play.google.com/store/apps/details?id=com.receiptbank.android&hl=en_US&gl=US

[7] S. Apps, "Receipt scanner expense tracker by saldo apps," Google.com, 2013. [Online]. Available: https://play.google.com/store/apps/details?id=saldo.receiptscanner.app&hl=en_US&gl=US

[8] K. A. Hamad, "An android based receipt tracker system using optical character recognition," Ph.D. dissertation, 2017. [Online]. Available: https://openaccess.firat.edu.tr/xmlui/bitstream/handle/11508/18162/477923.pdf?sequence=1&isAllowed=y

[9] O. Maslova, L. Klein, D. Dabernat, A. Benoit, and P. Lambert, "Receipt automatic reader," *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, 09 2019. [Online]. Available: https://sci-hub.se/10.1109/CBMI.2019.8877407

[10] K. D. Saputra, D. A. Rahmaastri, K. Setiawan, D. Suryani, and Y. Purnama, "Mobile financial management application using google cloud vision api," *Procedia Computer Science*, vol. 157, pp. 596–604, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919311779

[11] V. Robert and H. Talbot, "Does super-resolution improve ocr performance in the real world? a case study on images of receipts," *2020 IEEE International Conference on Image Processing (ICIP)*, 10 2020. [Online]. Available: https://sci-hub.se/10.1109/CBMI.2019.8877407

[12] B. Sainz-De-Abajo, J. M. García-Alonso, J. J. Berrocal-Olmeda, S. Laso-Mangas, and I. De La Torre-Díez, "Foodscan: Food monitoring app by scanning the groceries receipts," *IEEE Access*, vol. 8, p. 227915–227924, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9300155

[13] *Optical character recognition applied on receipts printed in Macedonian language*. 11th International Conference on Informatics and Information Technologies, 2014. [Online]. Available: http://dejan.gjorgjevikj.com/papers/CIIT2014.59.pdf

[14] A. Yue, "Automated receipt image identification, cropping, and parsing," pp. 1–8, 2018. [Online]. Available: https://www.cs.princeton.edu/courses/archive/spring18/cos598B/public/projects/System/COS598B_spr2018_ReceiptParsing.pdf

[15] R. Ullah, A. Sohani, A. Rai, F. Ali, and R. Messier, "Optical character recognition engine to extract food-items and prices from grocery receipt images via templating and dictionary-traversal technique," *KIET Journal of Computing Information Sciences*, vol. 2, pp. 59–73, 01 2019. [Online]. Available: http://kjcis.pafkiet.edu.pk/index.php/kjcis/article/view/21/15

[16] *Extraction of information from bill receipts using optical character recognition*. International Conference on Smart Electronics and Communication (ICOSEC 2020), 09 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9215246

[17] G. Jaume, H. K. Ekenel, and J.-P. Thiran, "Funsd: A dataset for form understanding in noisy scanned documents," *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, 09 2019. [Online]. Available: https://sci-hub.se/10.1109/ICDARW.2019.10029

[18] C. d. Jager and M. Nel, "Business process automation: A workflow incorporating optical character recognition and approximate string and pattern matching for solving practical industry problems," *Applied System Innovation*, vol. 2, p. 33, 10 2019. [Online]. Available: https://www.researchgate.net/publication/336801449_Business_Process_Automation_A_Workflow_Incorporating_Optical_Character_Recognition_and_Approximate_String_and_Pattern_Matching_for_Solving_Practical_Industry_Problems

[19] X. Wang, X. Zhang, S. Lei, and H. Deng, "A method of text detection and recognition from receipt images based on craft and crnn," *Journal of Physics: Conference Series*, vol. 1518, p. 012053, 04 2020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1518/1/012053/meta

[20] W. Korobacz and M. Tabedzki, "Preprocessing photos of receipts for recognition," Advances in Computer Science Research, p. 87–103, 2018. [Online]. Available: https://bibliotekanauki.pl/articles/88364

[21] "Improving the quality of the output," tesseract-ocr. [Online]. Available: https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html

[22] "Supported images — cloud vision api — google cloud," Google Cloud, 2022. [Online]. Available: https://cloud.google.com/vision/docs/supported-files

[23] A. Tsimpiris, D. Varsamis, and G. Pavlidis, "Tesseract ocr evaluation on greek food menus datasets," *International Journal of Computing and Optimization*, vol. 9, pp. 13–32, 2022. [Online]. Available: http://www.m-hikari.com/ijco/ijco2022/ijco1-2022/p/varsamisIJCO1-2022.pdf

[24] R. F. Rahmat, D. Gunawan, S. Faza, N. Haloho, and E. B. Nababan, "Android-based text recognition on receipt bill for tax sampling system," *2018 Third International Conference on Informatics and Computing (ICIC)*, 10 2018. [Online]. Available: https://sci-hub.se/10.1109/IAC.2018.8780416

[25] B. Berger, M. S. Waterman, and Y. W. Yu, "Levenshtein distance, sequence comparison and biological database search," *IEEE Transactions on Information Theory*, vol. 67, pp. 3287–3294, 06 2021. [Online]. Available: https://web.archive.org/web/20210717232409id_/https://ieeexplore.ieee.org/ielx7/18/9437276/09097943.pdf

[26] A. , "Effectiveness of image pre-processing with google cloud vision ocr," 2017. [Online]. Available: https://groups.google.com/g/google-cloud-dev/c/lDBCIzBqPUs/m/82MQteLcAwAJ

[27] X. Zhao, E. Niu, Z. Wu, and X. Wang, "Cutie: Learning to understand documents with convolutional universal text information extractor," 2019. [Online]. Available: https://arxiv.org/pdf/1903.12363.pdf

[28] *Visual-Linguistic Methods for Receipt Field Recognition.*

Computer Vision – ACCV 2018, 2019. [Online]. Available: https://sci-hub.se/10.1007/978-3-030-20890-5_35

[29] *Chargrid: Towards Understanding 2D Documents*, Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2018. [Online]. Available: https://aclanthology.org/D18-1476.pdf

[30] T. I. Denk and C. Reisswig, "Bertgrid: Contextualized embedding for 2d document representation and understanding," 2019. [Online]. Available: https://arxiv.org/pdf/1909.04948v2.pdf

[31] W. Hwang, S. Kim, M. Seo, J. Yim, S. Park, S. Park, J. Lee, B. Lee, and H. Lee, "Post-ocr parsing: building simple and robust parser via bio tagging," 2019. [Online]. Available: https://openreview.net/pdf?id=SJgjf695UB

[32] D. Lohani, A. Belaïd, and Y. Belaïd, "An invoice reading system using a graph convolutional network," *Computer Vision – ACCV 2018 Workshops*, pp. 144–158, 2019. [Online]. Available: https://sci-hub.se/10.1007/978-3-030-21074-8_12

[33] M. Lazic, "Using natural language processing to extract information from receipt text," Ph.D. dissertation, 2020. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1458900/FULLTEXT01.pdf

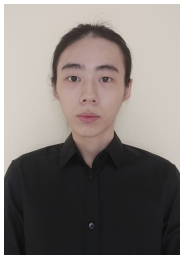
[34] E. Melz, "Understanding scanned receipts," 2020. [Online]. Available: https://arxiv.org/pdf/2005.01828.pdf

[35] R. Ullah, A. Sohani, A. Rai, F. Ali, and R. Messier, "Ocr engine to extract food-items, prices, quantity, units from receipt images, heuristics rules based approach," *International Journal of Scientific Engineering Research*, vol. 9, pp. 1334–1341, 02 2018. [Online]. Available: https://www.ijser.org/researchpaper/OCR-Engine-to-Extract-Food-Items-Prices-Quantity-Units-from-Receipt-Images-Heuristics-Rules-Based-Approach.pdf

[36] A. Pouran, B. Veyseh, F. Dernoncourt, W. Chang, and T. Nguyen, "Maddog: A web-based system for acronym identification and disambiguation," 2021. [Online]. Available: https://arxiv.org/pdf/2101.09893.pdf

[37] V. Joopudi, B. Dandala, and M. Devarakonda, "A convolutional route to abbreviation disambiguation in clinical text," *Journal of Biomedical Informatics*, vol. 86, pp. 71–78, 10 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046418301552?via%3Dihub

[38] R. Ramanathan, "Google cloud vision api available to all," Google Cloud Blog, 2016. [Online]. Available: https://cloud.google.com/blog/products/gcp/google-cloud-vision-api-available-to-all



Ooi Ming Yeung is a Bachelor of Computer Science graduate from UOW Malaysia KDU Penang University College with specialization in Artificial Intelligence. His study interests include machine learning and data science.

Dr. Justtina John has been with UOW Malaysia KDU Penang University College since March 2011 as a lecturer in the Department of Computing in the School of Engineering, Computing, and Built Environment. She had more than five years of experience serving private higher education institutions in teaching and administrative work prior to joining this institution. Now, she teaches Object Oriented System Analysis and Design for the Diploma in Computer Studies programme and System Analysis and Design, Business Information Systems, Information System Development, and Strategic Systems Management for the Bachelor of Information Systems(Hons) programme.

Dr Nursakirah is a computing lecturer at UOW Malaysia KDU Penang University College. In 2014, she earned a Master's degree in computer science with a major in Information Systems. She finished her Doctor of Philosophy in International Studies because of her keen interest in cross-cultural management and the switching behaviours of people from various cultures in global virtual team context. She has worked with her supervisor to publish articles on topics related to her area of expertise, including cross-cultural code switching, global virtual teams, and information systems on digital platforms. She is also very interested in using qualitative research methods because they enable her to fully explore and explain her research.