Finetuned LayoutLMv3 for Indonesian Receipts Extraction

I Made Andre Dwi Winama Putraa,\*, Ni Kadek Ayu Wirdiania, A. A. Kompiang Oka Sudanaa

*a Departement of Information and Technology, Universitas Udayana, Jimbaran, Badung, 80361, Indonesia*

*Corresponding author: andre002wp@gmail.com*

***Abstract***—**Shopping is a transaction that generates a record as a payment receipt. Typically, a receipt is given as a small piece of paper that can be easily lost. It is essential to store the transaction information in the receipt digitally. Keeping the information in a digital form will make it easily accessible and will overcome the problem of easily lost receipts. Currently, the process of transferring receipt information into digital form is still being done manually. Having a system that can extract this information helps speed up the digitalization process tremendously. This research proposes a method that applies finetuning to the LayoutLMv3 Model and, with the help of OCR from Google Vision, can be used to extract transaction information contained in the receipt. The system works by using Google Vision to parse and segment every word contained within the receipt and its bounding box** **The LayoutLMv3 model will then assign labels to each word, and important words will be extracted. The finetuned LayoutLMv3 Model successfully achieved an accuracy of 97.98% on training data and 90% accuracy on real-time test scenarios for extracting information on receipts written in the Indonesian language. However, variations in the diverse forms of receipts make extracting information on interrelated labels such as product names, quantities, and prices difficult. Furthermore, the current system is still unable to extract information from very long receipts. Based on these results, further experimentation needs to incorporate very long receipts in mind.**

***Keywords***—**LayoutLMv3, Optical Character Recognition, Receipt, Computer Vision**

1. Introduction

Nowadays, every transaction generates a payment receipt, whether shopping at a minimarket, restaurant, mall, hardware store, or any other establishment. These payment receipts are typically received as a small piece of paper containing the transaction summary. Keeping important financial information on a small piece of paper that can easily get lost is not ideal if we want to keep track of our spending. That's why extracting the information from the receipt and storing it digitally is crucial.

It's best to store the information in digital form to ensure that shopping receipts or notes are not easily lost. Currently, the extraction process is done manually for each transaction which takes a lot of time [1]. The existence of a system that can extract information from receipts and save it in digital format automatically will increase work efficiency. Storing it in digital form also makes it easier to see expenditure information for a certain period. One method that can be used to extract text information is Optical Character Recognition (OCR).

Optical Character Recognition (OCR) is the process of converting text in an image into machine-readable text format. The image used for text conversion can come from printed text or handwritten characters [2,3]. OCR technology is a part of artificial intelligence widely used in automation fields such as document and questionnaire scanning, license plate reading, and document verification, among others [4]. Several Model architectures that can be used as a basis for OCR are the Long Short Term Memory (LSTM) Model and Convolutional Neural Network (CNN). The OCR model is often combined with the Natural Language Processing (NLP) model to improve text reading accuracy [5]. The NLP architecture frequently used in this context is Bidirectional Encoder Representations from Transformers (BERT) [6]. The application of OCR technology can automatically extract information from receipts and notes.

Research into creating an automatic receipt extraction system has been conducted using various methods to detect and localize key information from receipts. Raoui et al. used localization and Deep Convolution Neural Networks (DCNN) to segment store signs [7]. Shi et al. and Lin et al. used template matching based on prior knowledge and specific signs of a given receipt to locate receipt information [8,9]. Meng et al. used a YOLOv3 model to segment key information on an invoice image with a standardized template [10]. However, some of these systems rely on receipts having a standardized form, which is only sometimes the case, and assume the receipt never changes its template.

Recently, progress in document extraction AI has been significant with the rise of a new topic: Document AI. Document AI is a field of study that aims to provide techniques for understanding, extracting, and classifying documents. Essentially it is an object detection task for document images [11]. An early iteration of the Document AI task is the Faster R-CNN model by Schreiber et al. [12], which achieved SOTA performance in the ICDAR 2013 Dataset. In recent years progresses on document AI have been remarkable with the rise of models such as Graph Convolution Network [13] and LayoutLM [14–16] as well as datasets for benchmarking such as PubLayNet [17], TableBank [18]. With these new models, we hope to build a system to extract information from a receipt without the need to recognize the exact template of the receipt.

The novelty of this research aims to create a system that can extract information without relying on the receipts template. The proposed system uses Google Vision OCR to segment and extract words within the receipt and uses a fine-tuned LayoutLMv3 model to recognize the content of the receipt and extract meaningful information.

1. Materials and method

## Model Finetuning Diagram

The workflow of the finetuning process starts with gathering and creating an Indonesian receipt dataset. The gathered receipt then goes into the preprocessing step to crop, annotate and combine the dataset. The next step is to train the model with the combined dataset and evaluate the result.

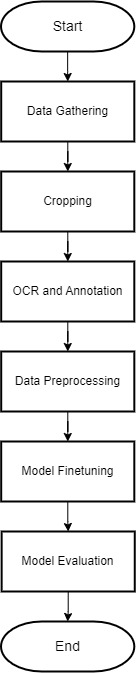


Fig. 1 Model Finetuning Diagram

## Data Gathering

The data gathering process involves manually gathering Receipts written in Indonesian, totaling 100 receipts from various vendors. Each of the receipts will then be captured with a camera and processed to separate it from the background.



Fig. 2 Data Processing : original (top left), greyscale (top right), blur and dilated (bottom left), contour edge (bottom right)

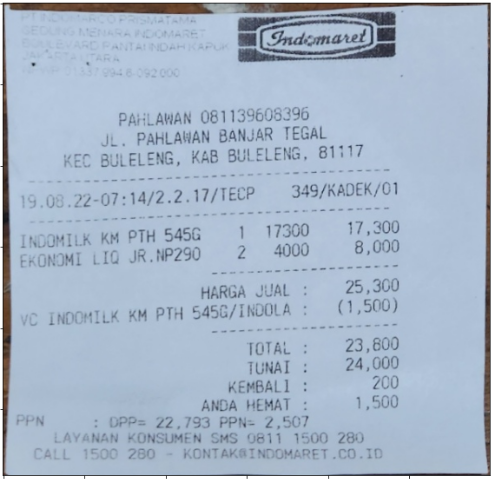


Fig. 3 Region of Interest Cropping Result

The general process of separating receipt pictures from their background uses Canny edge contour detection. Firstly, the image is converted to grayscale. Then, the image is blurred and dilated, so the text is not detected as edges. Finally, Canny is applied to detect the edges, and the image is cropped based on the detected edges.

## LayoutParser

Layout Parser is a tool with the capability of a document AI that is capable of extracting and segmenting information from a document image. Layout Parser serves as a tool to help extract information from a receipt using the Google Vision OCR agent and return each word and bounding box from the receipt. [15].

## Google Vision API

Google Vision API is a machine learning model trained to perform OCR through REST and RPC APIs. Google Vision API can annotate images and provide labels for each category detected in the image. This annotation process is called automatic image annotation [19]. The automatic image annotation from Google Vision API can extract content from an image to obtain visual information such as labeling images, detecting facial landmarks, OCR, etc. [20].

## Data Annotation

Data annotation involves the use of the Layout Parser library to extract and segment every word within a receipt using Google Vision API. The results are then manually annotated by assigning the corresponding labels listed in Table 1.

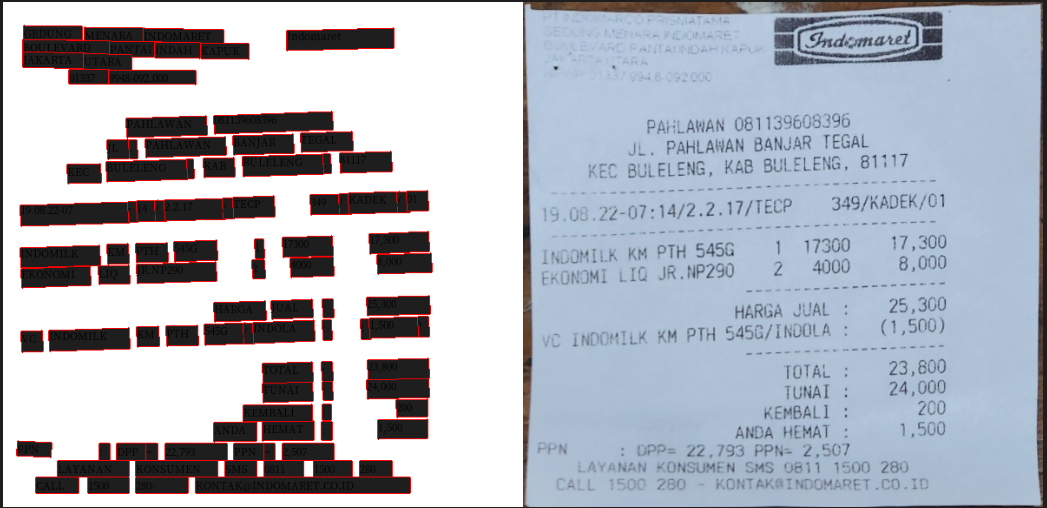


Fig. 4 Image Segmentation using Layout Parser and Google Vision API.

The segmentation process results in a list of bounding box coordinates and the detected word sequence within the image. These boxes and words then get labeled and saved for finetuning the LayoutLM model.

TABEL I  
Receipt Annotation Code.

|  |  |
| --- | --- |
| Label Name | Label Code |
| Ignore | 0 |
| Store\_name\_value | 1 |
| Date\_value | 2 |
| Time\_value | 3 |
| Prod\_item\_key | 4 |
| Prod\_item\_value | 5 |
| Prod\_quantity\_key | 6 |
| Prod\_quantity\_value | 7 |
| Prod\_price\_key | 8 |
| Prod\_price\_value | 9 |
| Subtotal\_key | 10 |
| Subtotal\_value | 11 |
| Total\_key | 12 |
| Total\_value | 13 |
| Others | 14 |

Table 1 consists of all the labels used in the annotation process. Eight important labels are going to be extracted, which are Store\_name\_value, Date\_value, Time\_value, Prod\_item\_value, Prod\_quantity\_value, Prod\_price\_value, Subtotal\_value, and Total\_value with the other six labels acting as an anchor point to helps determined each important label from the rest.

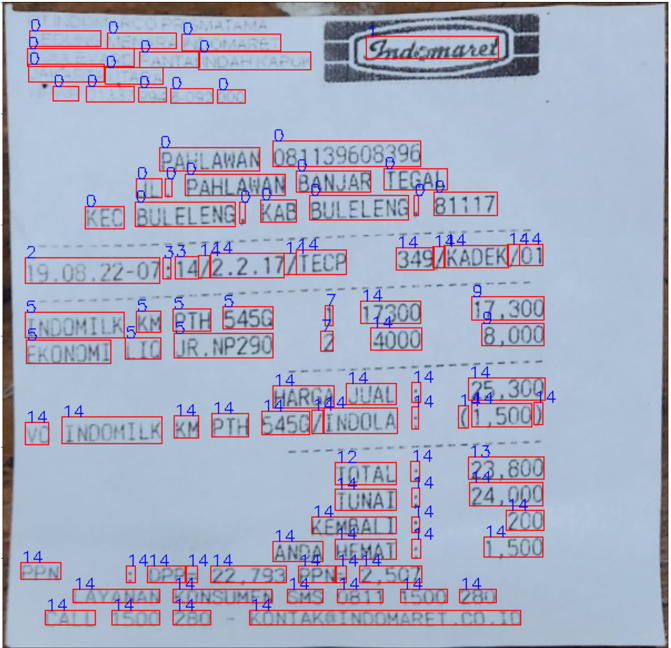


Fig. 5 Results of data manually labeled using table 1 as a reference.

Figure 4 shows the results of the annotation process. The figure displays labels for all the words detected following Table 1. The fine-tuned model will later be trained to predict labels on various receipts and assign labels accordingly.

## LayoutLM

LayoutLM is a document understanding model designed to comprehend the structure of a document. It was developed by considering the developments in Natural Language Processing (NLP), where every NLP model always focuses on text-level manipulation. LayoutLM [14–16] is created by utilizing the interaction between the text information within a document and its layout using BERT [6] as a reference. This model was developed using data from scanned documents from various categories, such as letters, memos, emails, invoices, news, articles, questionnaires, resumes, etc. [16]. With the recent development of the LayoutLMv3 model aims to analyze Visually-rich Document Understanding (VrDU) where structured information can be automatically extracted. LayoutLMv3 improves upon the original model by integrating Masked Image Modeling (MIM) from BEiT [21] to interpret visual content in the document. Inspired by ViT [22] and ViLT [23], LayoutLMv3 [16] uses linear projection features of image patches before feeding them into the multimodal transformer to remove the need to extract CNN grid features [15,24] or rely on an object detector like Faster R-CNN [25] to extract region features [14,26–28] for image embeddings which require heavy computation bottleneck or region supervision making LayoutLMv3 the first multimodal model in Document AI that does not rely on CNNs to extract image features.

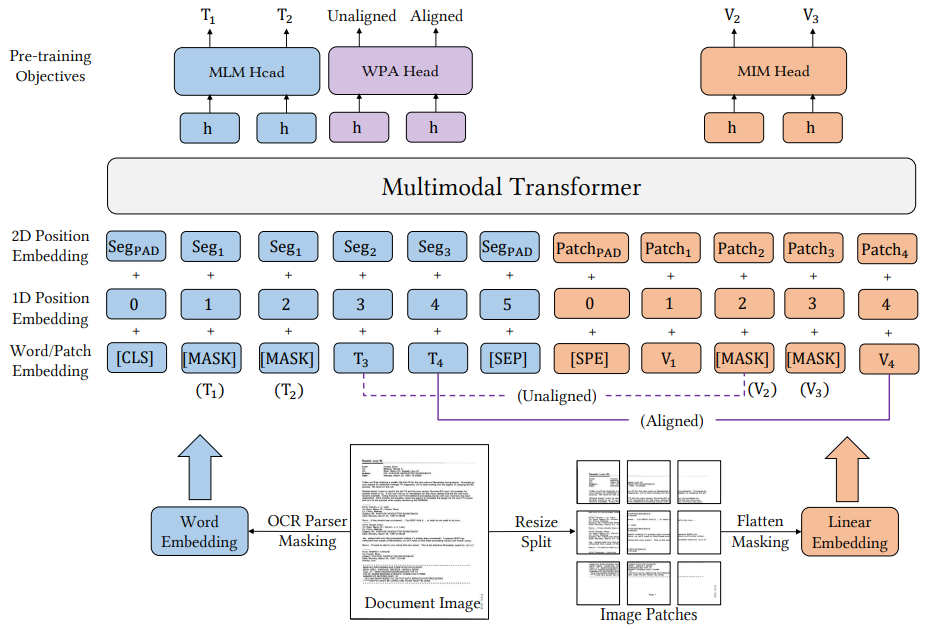


Fig. 6 LayoutLMv3 Architecture Overview [16].

## Model Finetuning

Finetuning of the LayoutLM model uses the version LayoutLMv3 as the base model. Firstly, the previously gathered data are combined with the Wild Receipt dataset and split using 80% data for training and 20% for validation, with the total combined data used for training being 1348, and 492 is used for validation. The data is then processed using LayoutLMv3 auto processor to obtain the embedding and used for finetuning the base model.

TABEL II  
dataset split value

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Training** | **Validation** |
| WildReceipt | 1268 | 472 |
| Indonesian Receipt | 80 | 20 |

## Model Evaluation

Model performance evaluation is carried out using a typical classification evaluation problem, as presented in the confusion matrix table below.

TABEL III  
Receipt Annotation Code.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted** | | |
| **Actual** |  | **Negative** | **Positive** |
| **Negative** | TN | FP |
| **Positive** | FN | TP |

The confusion matrix consists of four main values, representing total counts from the predicted and actual values. These values would then be used to calculate the accuracy, precision, recall, and F-measure of a given problem [29].

(1)

Accuracy is one of the most commonly used metrics for classification. Accuracy is calculated using the sum of the true values over the sum of the possible values.

(2)

(3)

Precision and recall are other important metrics that provide valuable information regarding how well the model performs [30]. Precision measures how accurate the model is in predicting positive values, while recall measures its strength in predicting positive values.

(4)

F-measure is calculated using a weighted harmonic mean between precision and recall. F measure helps to further understand the tradeoff between improving recall and its effect on precision [29].

## System Deployment and Evaluation Diagram

The system deployment evaluates the model's performance in a real-time scenario. Firstly, the finetuned model will be deployed on a web server that takes an image as input from a smartphone. The image will then be processed, and important information will be extracted using the labels assigned by the LayoutLM model as a reference. Finally, the extracted information is returned to the smartphone, and the results will be evaluated.

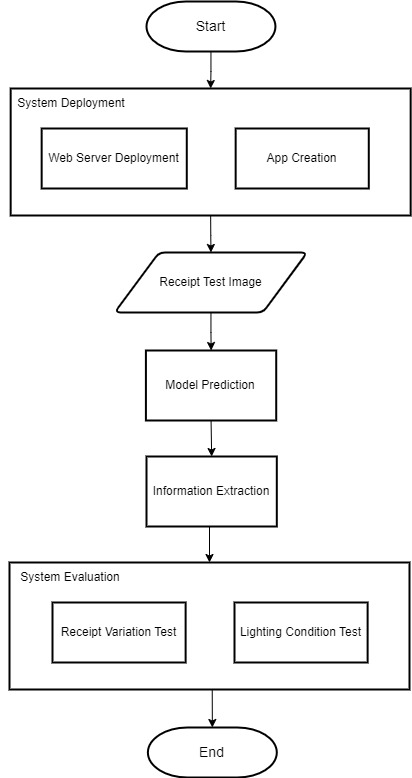


Fig. 8 Model Deployment and System Evaluation Diagram

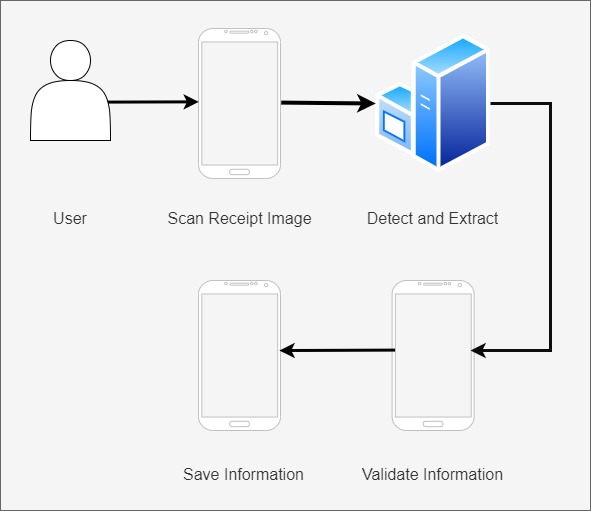


Fig. 9 Receipt Detection System

## Information Extraction

Information extraction is done in 2 different ways. Labels with multiple words attached and interrelated label extraction involve calculating the height of the bounding box's midpoint. In contrast, a single-worded label is extracted without the need to be processed.

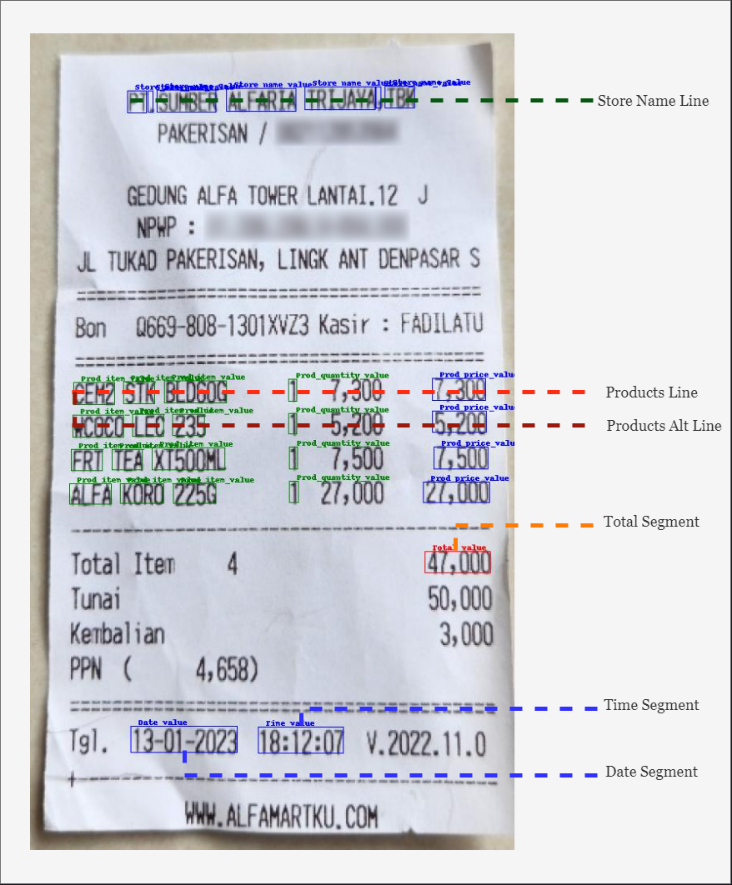


Fig. **7** Information Extraction Illustration

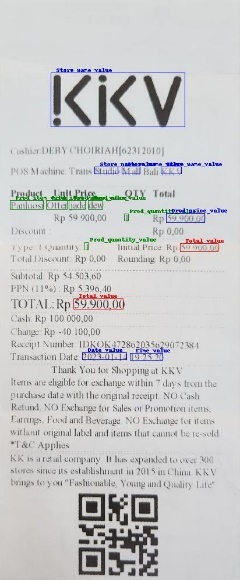
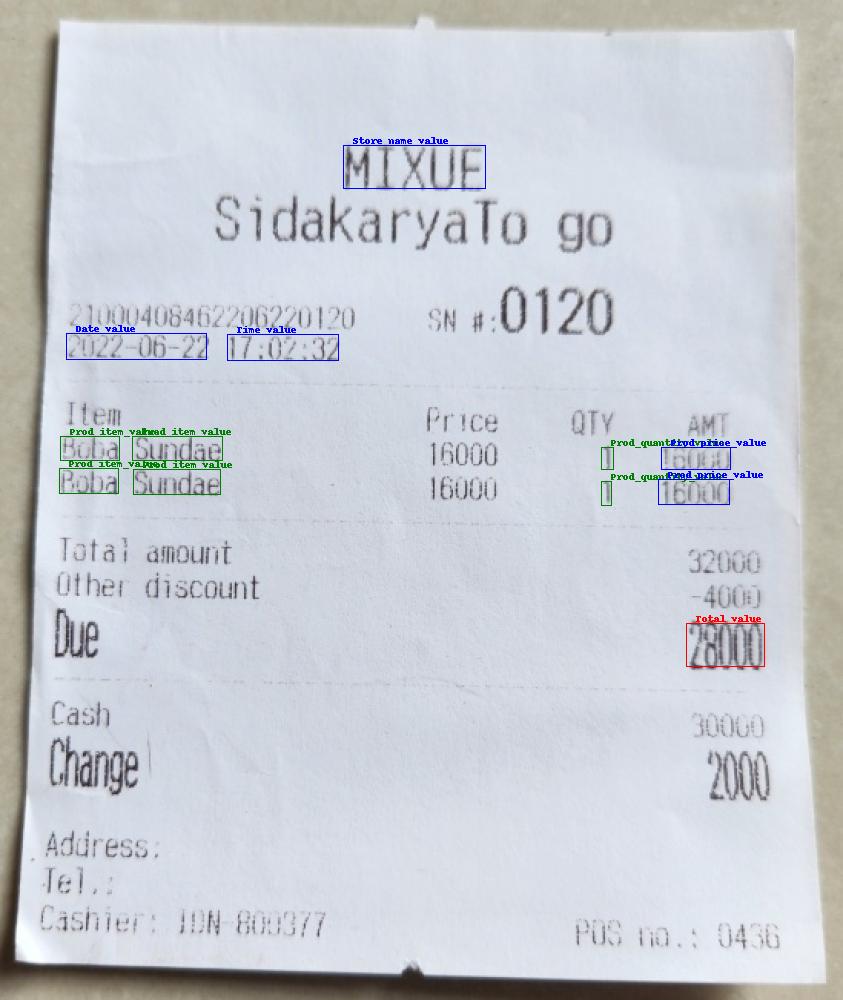
The product extraction process checks for whether a product name, quantity, and price labels exist in a single line. The products will be extracted if all three are found in a single line. If at least one of the corresponding labels is missing, the system will check for the missing label on the alternate line for receipts that uses multiple lines for each product item.

## System Tests

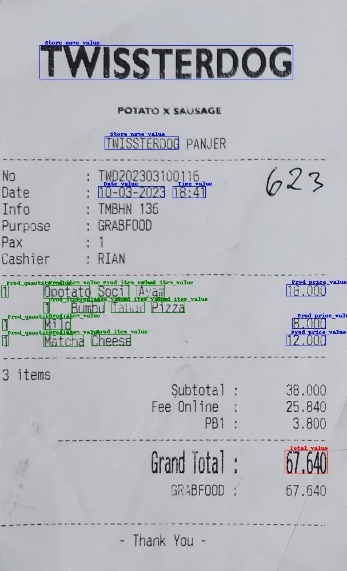
The system test uses 2 test scenarios to test the limits of the proposed system. The test will use a different accuracy calculation based on the amount of important information contained within the receipt compared to the amount of information successfully extracted.

(5)

Extraction accuracy refers to the overall amount of important information successfully extracted from all the important information available in the receipt. In this case, important information refers to the store name labels, products, date and time of the transaction, and total payment. At the same time, error accounts for the sum of important labels missed and labels misclassified as important.

(a)



(b)

Fig. 10 (a) Receipt Variation Tests Sample (b) Lighting Condition Tests Sample

# Results and Discussion

The initial finetuning process on the LayoutLMv3 model shows promising results for classifying important labels on a given receipt. The model is able to differentiate which word is important information and which word should be ignored.

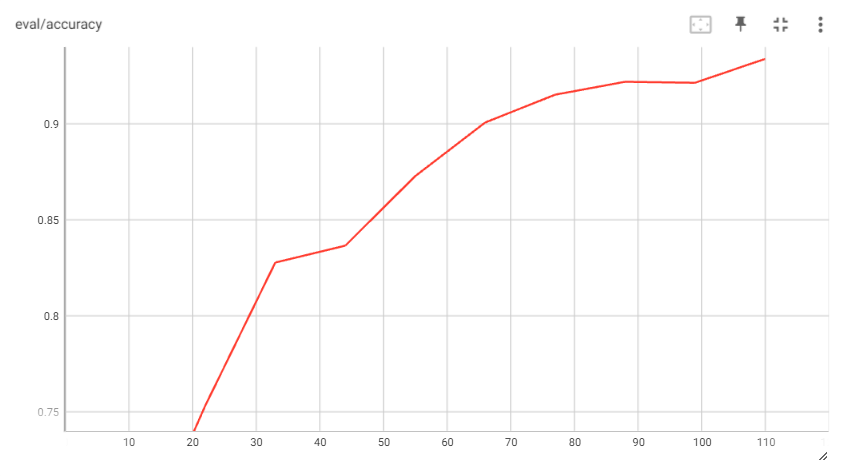


Fig. 11 LayoutLMv3 Evaluation Accuracy after 10 epochs

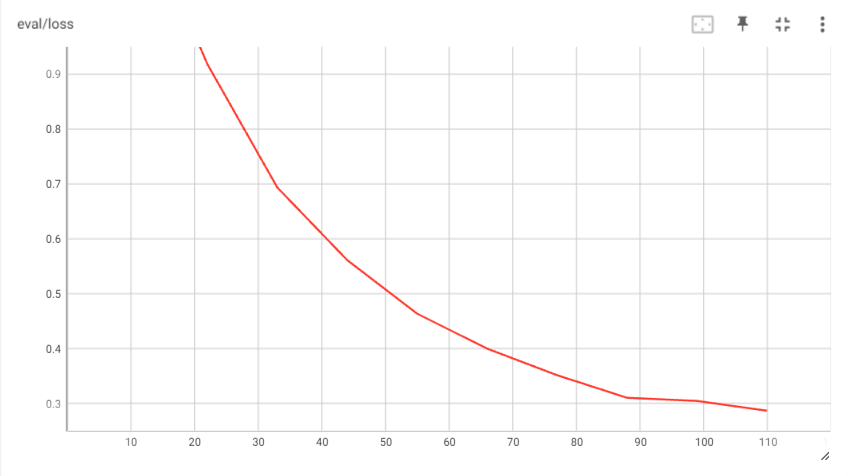


Fig. 12 LayoutLMv3 Evaluation Loss after 10 epochs

Figure 11 and 12 shows the result of finetuning the LayoutLMv3 Base model. The finetuning process on the combined data achieves a training accuracy of 99,2% and an evaluation accuracy of 97.98% after ten epochs. The publicly available WildReceipt dataset, which includes receipts written in multiple languages, proves to help improve the model's accuracy for extracting information from Indonesian receipts compared to using only 100 Indonesian receipts, as shown in Table IV below.

TABLE IV  
Training Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score |
| Indonesian | 0.8889 | 0.9276 | 0.9064 |
| Combined | 0.9799 | 0.9798 | 0.9797 |

Table IV compares the result of training after ten epochs. Based on the results, combined data from 100 gathered Indonesian receipts combined with the Wildreceipt dataset performed better on all three weighted evaluation categories.

TABLE V  
Receipts variation test

|  |  |  |
| --- | --- | --- |
| **Store Id** | **Sample Size** | **Average Extraction Accuracy** |
| 1 | 5 | 91,7% |
| 2 | 3 | 93,3% |
| 3 | 3 | 86,9% |
| 4 | 3 | 90,9% |
| 5 | 2 | 100% |
| 6 | 2 | 94,4% |
| 7 | 4 | 90,9% |
| 8 | 4 | 86% |
| 9 | 5 | 88% |
| 10 | 1 | 88,3% |

Table V shows the model's accuracy in detecting labels based on a real-time scenario using various receipts under optimal lighting conditions. Receipts used in this test came from a diverse set than the one used in the training and validation. Moreover, the test also includes a receipts template that the model has never seen. Store id refers to a unique store, and sample size refers to the number of receipts collected as a sample from that store. Some significant problems that are found during this test include:

* The model sometimes detects farewell messages (“Goodbye”, “Thank you”, etc) and receipts coupons on the edges of the receipts as the store name.
* Part of the receipts identification code is labeled as date information, with a serial number format that resembles date and time information.
* The model detects total payment twice on some receipts with a nonstandard total key name such as “Due,” “Net Sales,” “Total Sale,” etc.

The results suggest that while the model is able to detect key information perfectly on receipts with moderate spacing between each item, it still struggles to accurately detect information on some receipts with different layouts, designs, and fonts. Nevertheless, the proposed system achieved an average extraction accuracy of 90%.

TABLE VI  
various lighting condition test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Store id** | **Label name** | **Optimal Condition** | **30% shade** | **80% shade** |
| 1 | Store Name | Detected | Detected | Detected |
| Date | Detected | Detected | Detected |
| Time | Detected | Detected | Detected |
| Products | Partially Detected | Partially Detected | Partially Detected |
| Total | Detected | Detected | Detected |
| 2 | Store Name | Detected | Detected | Detected |
| Date | Detected | Detected | Detected |
| Time | Detected | Detected | Detected |
| Products | Detected | Detected | Partially Detected |
| Total | Detected | Detected | Detected |
| 3 | Store Name | Detected | Detected | Detected |
| Date | Not Detected | Not Detected | Not Detected |
| Time | Detected | Detected | Detected |
| Products | Detected | Detected | Detected |
| Total | Not Detected | Not Detected | Not Detected |
| 4 | Store Name | Detected | Detected | Detected |
| Date | Detected | Detected | Detected |
| Time | Detected | Detected | Detected |
| Products | Detected | Detected | Partially Detected |
| Total | Detected | Detected | Detected |
| 5 | Store Name | Not Detected | Not Detected | Not Detected |
| Date | Detected | Detected | Detected |
| Time | Detected | Detected | Detected |
| Products | Detected | Partially Detected | Partially Detected |
| Total | Detected | Detected | Detected |

Table VI shows the model's performance in detecting under different lighting conditions. The result suggests that lighting condition affects the system's segmentation process. Some items cannot be appropriately segmented, which does affect the system's overall accuracy.

After further inspection, it is later found that the problem lies in the OCR system rather than the LayoutLM Model. Specifically, the word and bounding box extraction using Vision API sometimes failed to extract information in a dimly lit environment and shaded region.

# Conclusion

This study suggests an alternative method to extract key information from receipts using LayoutLMv3. The system was trained using gathered Indonesian receipts and the publicly available WildReceipt dataset, which includes receipts written in multiple languages. Using the combined dataset proves to be significant in increasing the accuracy of the model for extracting information from Indonesian receipts compared to using only 100 Indonesian receipts. The best-performing model is able to achieve an accuracy of 97.98% for predicting keywords on Indonesian receipts after being trained for ten epochs.

The LayoutLMv3 Model combined with OCR from Google Vision has successfully extracted information from notes without recognizing the template of each note to be read. Based on the three tests, the model accuracy obtained during the initial evaluation of 97.98% still has issues. The model still cannot work optimally on several note variations with closely located information and too small fonts. These variations cause the system's extraction accuracy to decrease to 90%. In addition, the model has yet to be able to extract information correctly from very long notes.

Another problem found during the system testing is that the system has not been able to detect notes optimally in dark conditions or notes covered by shadows. This problem is caused by the OCR extraction process through Google Vision, which sometimes fails to extract important information from notes. For future work, more receipts sample is needed for the model to understand differences in the receipts layout and incorporate a better preprocessing method.

# REFERENCES

[1] F., A.J., S.M., & G.P. Mohammad, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2088-2090, IEEE, 2014.

[2] J. Memon, M. Sami, R.A. Khan, Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR), (2019). http://arxiv.org/abs/2001.00139.

[3] V. Kumar, P. Kaware, P. Singh, R. Sonkusare, S. Kumar, Extraction of information from bill receipts using optical character recognition, Proceedings - International Conference on Smart Electronics and Communication, ICOSEC 2020. (2020) 72–77. https://doi.org/10.1109/ICOSEC49089.2020.9215246.

[4] A. Qaroush, A. Awad, M. Modallal, M. Ziq, Segmentation-based, omnifont printed Arabic character recognition without font identification, Journal of King Saud University - Computer and Information Sciences. 34 (2022) 3025–3039. https://doi.org/10.1016/j.jksuci.2020.10.001.

[5] M. Hajiali, J.R. Fonseca Cacho, K. Taghva, Generating Correction Candidates for OCR Errors using BERT Language Model and FastText SubWord Embeddings, in: Lecture Notes in Networks and Systems, Springer Science and Business Media Deutschland GmbH, 2022: pp. 1045–1053. https://doi.org/10.1007/978-3-030-80119-9\_69.

[6] J. Devlin, M.-W. Chang, K. Lee, K.T. Google, A.I. Language, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, n.d. https://github.com/tensorflow/tensor2tensor.

[7] R. Raoui-Outach, C. Million-Rousseau, A. Benoit, P. Lambert, Deep learning for automatic sale receipt understanding, Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications, IPTA 2017. 2018-Janua (2018) 1–6. https://doi.org/10.1109/IPTA.2017.8310088.

[8] C.J. Lin, Y.C. Liu, C.L. Lee, Automatic Receipt Recognition System Based on Artificial Intelligence Technology, Applied Sciences (Switzerland). 12 (2022). https://doi.org/10.3390/app12020853.

[9] S. Shi, C. Cui, Y. Xiao, An Invoice Recognition System Using Deep Learning, in: 2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS), 2020: pp. 416–423. https://doi.org/10.1109/ICICAS51530.2020.00093.

[10] Y. Meng, R. Wang, J. Wang, J. Yang, G. Gui, IRIS: Smart Phone Aided Intelligent Reimbursement System Using Deep Learning, IEEE Access. 7 (2019) 165635–165645. https://doi.org/10.1109/ACCESS.2019.2953501.

[11] L. Cui, Y. Xu, T. Lv, F. Wei, Document AI: Benchmarks, Models and Applications, (2021). http://arxiv.org/abs/2111.08609.

[12] S. Schreiber, S. Agne, I. Wolf, A. Dengel, S. Ahmed, DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017: pp. 1162–1167. https://doi.org/10.1109/ICDAR.2017.192.

[13] X. Liu, F. Gao, Q. Zhang, H. Zhao, Graph Convolution for Multimodal Information Extraction from Visually Rich Documents, (2019). http://arxiv.org/abs/1903.11279.

[14] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, LayoutLM: Pre-training of Text and Layout for Document Image Understanding, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2020) 1192–1200. https://doi.org/10.1145/3394486.3403172.

[15] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang, L. Zhou, LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding, (2020). http://arxiv.org/abs/2012.14740.

[16] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei, LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking, in: Association for Computing Machinery (ACM), 2022: pp. 4083–4091. https://doi.org/10.1145/3503161.3548112.

[17] X. Zhong, J. Tang, A. Jimeno Yepes, PubLayNet: Largest Dataset Ever for Document Layout Analysis, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019: pp. 1015–1022. https://doi.org/10.1109/ICDAR.2019.00166.

[18] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Z. Li, TableBank: A Benchmark Dataset for Table Detection and Recognition, (2019). http://arxiv.org/abs/1903.01949.

[19] P. Baker, L. Collins, Creating and analysing a multimodal corpus of news texts with Google Cloud Vision’s automatic image tagger, Applied Corpus Linguistics. 3 (2023) 100043. https://doi.org/10.1016/j.acorp.2023.100043.

[20] K.D. Saputra, D.A. Rahmaastri, K. Setiawan, D. Suryani, Y. Purnama, Mobile financial management application using google cloud vision API, in: Procedia Comput Sci, Elsevier B.V., 2019: pp. 596–604. https://doi.org/10.1016/j.procs.2019.09.019.

[21] H. Bao, L. Dong, S. Piao, F. Wei, BEiT: BERT Pre-Training of Image Transformers, (2021). http://arxiv.org/abs/2106.08254.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, (2020). http://arxiv.org/abs/2010.11929.

[23] W. Kim, B. Son, I. Kim, ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision, (2021).

[24] S. Appalaraju, B. Jasani, B.U. Kota, Y. Xie, R. Manmatha, DocFormer: End-to-End Transformer for Document Understanding, (2021). http://arxiv.org/abs/2106.11539.

[25] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, (2015). http://arxiv.org/abs/1506.01497.

[26] J. Gu, J. Kuen, V.I. Morariu, H. Zhao, R. Jain, N. Barmpalios, A. Nenkova, T. Sun, UniDoc: Unified Pretraining Framework for Document Understanding, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, J.W. Vaughan (Eds.), Adv Neural Inf Process Syst, Curran Associates, Inc., 2021: pp. 39–50. https://proceedings.neurips.cc/paper\_files/paper/2021/file/0084ae4bc24c0795d1e6a4f58444d39b-Paper.pdf.

[27] P. Li, J. Gu, J. Kuen, V.I. Morariu, H. Zhao, R. Jain, V. Manjunatha, H. Liu, SelfDoc: Self-Supervised Document Representation Learning, (2021). http://arxiv.org/abs/2106.03331.

[28] R. Powalski, Ł. Borchmann, D. Jurkiewicz, T. Dwojak, M. Pietruszka, G. Pałka, Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer, (2021). http://arxiv.org/abs/2102.09550.

[29] A. Kulkarni, D. Chong, F.A. Batarseh, Foundations of data imbalance and solutions for a data democracy, in: Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering, Elsevier, 2020: pp. 83–106. https://doi.org/10.1016/B978-0-12-818366-3.00005-8.

[30] P. Bruce, A. Bruce, P. Gedeck, an O.M.Company. Safari, Practical Statistics for Data Scientists, 2nd Edition, n.d.