

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF INFORMATION AND TECHNOLOGY



FINAL REPORT
Course: Computational Thinking
CS117.L21

Name of project:

KEY INFORMATION EXTRACTION ON VIETNAMESE RECEIPTS

Instructor: PhD. Duc Thanh Ngo

Students:

TT	Full name	Role	Phone	Email
1.	Doanh C. Bui	Leader	0938237990	19521366@gm.uit.edu.vn
2.	Ngoc Dung T. Bui	Member	0762632004	19521385@gm.uit.edu.vn
3.	Minh D. Nguyen	Member	0816044072	19520164@gm.uit.edu.vn

Ho Chi Minh City – August 2021

TABLE OF CONTENTS

1. Introduction.....	3
1.1. Introduction to the problem	3
1.2. Challenges.....	4
1.3. Applying computational thinking	4
2. Related works.....	5
2.1. Previous approaches	5
2.2. Deep CNN architecture.....	6
3. Methodology	9
3.1. Data preparing	10
3.2. Text image pre-processing.....	10
3.3. Predictive location post-processing	11
3.4. Model configuration	11
4. Experiments.....	12
4.1. MC-OCR dataset	12
4.2. Metric.....	13
4.3. Results.....	14
4.4. Discussion.....	15
5. Future work.....	17
6. References.....	17

Assigning tasks to members

Order	Task	Assignee
1	Hold periodic meetings, agree on ideas of members	Doanh
2	Research about Faster R-CNN, train Faster R-CNN model for extracting locations of key information on MMDetection toolbox.	Doanh, Dung
3	Research about Transformer, find source code and train Transformer model for text recognition	Minh, Doanh
4	Convert data annotation to COCO format for training Faster R-CNN model.	Dung
5	Preparing data for training Transformer model.	Minh
6	Report, edit source code and push in Github	Doanh

1. Introduction

1.1. Introduction to the problem

Extracting information from receipts is an important element of the financial, accounting, and taxes fields. Companies that provide these services should collect pertinent data from receipts and keep it on computers for easy management. This activity is still done by hand, which will cost a lot of money, so it's critical to design a machine-based system for extracting data automatically. The topic we have discussed is known as the Receipts OCR challenge, which is the task of extracting useful information from receipts fundamentally mechanically.

We define the input and output of the problem as follows:

- Input: the raw image of Vietnamese receipt
- Output: the coordinates of four key information regions on image and extracted text on these regions. The four key information on receipts that we extract are SELLER, ADDRESS, TIMESTAMP and TOTAL COST.


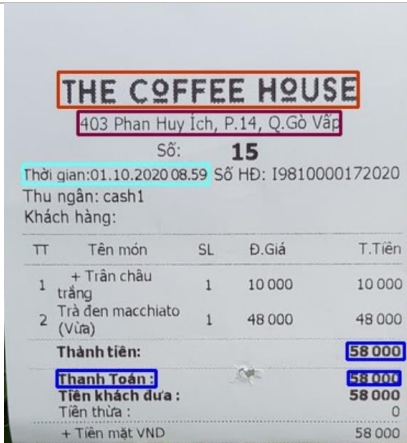
																																																																																	
<p>THE COFFEE HOUSE 403 Phan Huy Ích, P.14, Q.Gò Vấp Số: 15 Thời gian:01.10.2020 08.59 Số HĐ: I9810000172020 Thu ngân: cash1 Khách hàng:</p> <table><tr><th>TT</th><th>Tên món</th><th>SL</th><th>Đ.Giá</th><th>T.Tiền</th></tr><tr><td>1</td><td>+ Trân châu trắng</td><td>1</td><td>10 000</td><td>10 000</td></tr><tr><td>2</td><td>Trà đen macchiato (Vừa)</td><td>1</td><td>48 000</td><td>48 000</td></tr><tr><td colspan="4">Thành tiền:</td><td>58 000</td></tr><tr><td colspan="4">Thanh Toán :</td><td>58 000</td></tr><tr><td colspan="4">Tiền khách đưa :</td><td>58 000</td></tr><tr><td colspan="4">Tiền thừa :</td><td>0</td></tr><tr><td colspan="4">+ Tiền mặt VND</td><td>58 000</td></tr></table>	TT	Tên món	SL	Đ.Giá	T.Tiền	1	+ Trân châu trắng	1	10 000	10 000	2	Trà đen macchiato (Vừa)	1	48 000	48 000	Thành tiền:				58 000	Thanh Toán :				58 000	Tiền khách đưa :				58 000	Tiền thừa :				0	+ Tiền mặt VND				58 000	<p>THE COFFEE HOUSE 403 Phan Huy Ích, P.14, Q.Gò Vấp Số: 15 Thời gian:01.10.2020 08.59 Số HĐ: I9810000172020 Thu ngân: cash1 Khách hàng:</p> <table><tr><th>TT</th><th>Tên món</th><th>SL</th><th>Đ.Giá</th><th>T.Tiền</th></tr><tr><td>1</td><td>+ Trân châu trắng</td><td>1</td><td>10 000</td><td>10 000</td></tr><tr><td>2</td><td>Trà đen macchiato (Vừa)</td><td>1</td><td>48 000</td><td>48 000</td></tr><tr><td colspan="4">Thành tiền:</td><td>58 000</td></tr><tr><td colspan="4">Thanh Toán :</td><td>58 000</td></tr><tr><td colspan="4">Tiền khách đưa :</td><td>58 000</td></tr><tr><td colspan="4">Tiền thừa :</td><td>0</td></tr><tr><td colspan="4">+ Tiền mặt VND</td><td>58 000</td></tr></table>	TT	Tên món	SL	Đ.Giá	T.Tiền	1	+ Trân châu trắng	1	10 000	10 000	2	Trà đen macchiato (Vừa)	1	48 000	48 000	Thành tiền:				58 000	Thanh Toán :				58 000	Tiền khách đưa :				58 000	Tiền thừa :				0	+ Tiền mặt VND				58 000
TT	Tên món	SL	Đ.Giá	T.Tiền																																																																													
1	+ Trân châu trắng	1	10 000	10 000																																																																													
2	Trà đen macchiato (Vừa)	1	48 000	48 000																																																																													
Thành tiền:				58 000																																																																													
Thanh Toán :				58 000																																																																													
Tiền khách đưa :				58 000																																																																													
Tiền thừa :				0																																																																													
+ Tiền mặt VND				58 000																																																																													
TT	Tên món	SL	Đ.Giá	T.Tiền																																																																													
1	+ Trân châu trắng	1	10 000	10 000																																																																													
2	Trà đen macchiato (Vừa)	1	48 000	48 000																																																																													
Thành tiền:				58 000																																																																													
Thanh Toán :				58 000																																																																													
Tiền khách đưa :				58 000																																																																													
Tiền thừa :				0																																																																													
+ Tiền mặt VND				58 000																																																																													
<p>Input</p>	<p>Extracted text:</p> <p>THE COFFEE HOUSE 403 Phan Huy Ích, P.14, Q. Gò Vấp Thời gian: 01.10.2020 09.59 Thanh toán: 58.000</p> <p>Output</p>																																																																																

Figure 1 An example of input and output of the problem

1.2. Challenges

This problem is extremely challenging because the structure of receipts is very diverse leading to the difficulty of extracting relevant information. Moreover, receipts might be crumpled or the content might be blurred and the quality of photos taken with mobile devices is very diverse because of being captured in different environments: indoor, out-door, complex background and so on [7]. Some previous studies proposed deep learning methods to handle this task, but almost for non-Vietnamese receipts or not achieving high accuracy on Vietnamese receipts.

1.3. Applying computational thinking

1.3.1. Abstraction

To solve the problem, we abstract the problem by two main tasks: Object detection and Text recognition. Because our problem is detecting key information, which means we must find the coordinates of “key information” on an image, the “key information” are now objects, so this is Object detection. After detecting, we crop the predicted key information regions and recognize text from these cropped images, so this is Text recognition.

1.3.2. Decomposition

This problem has lots of way to decompose, but in this project, we decompose the problem into subproblems as follows:

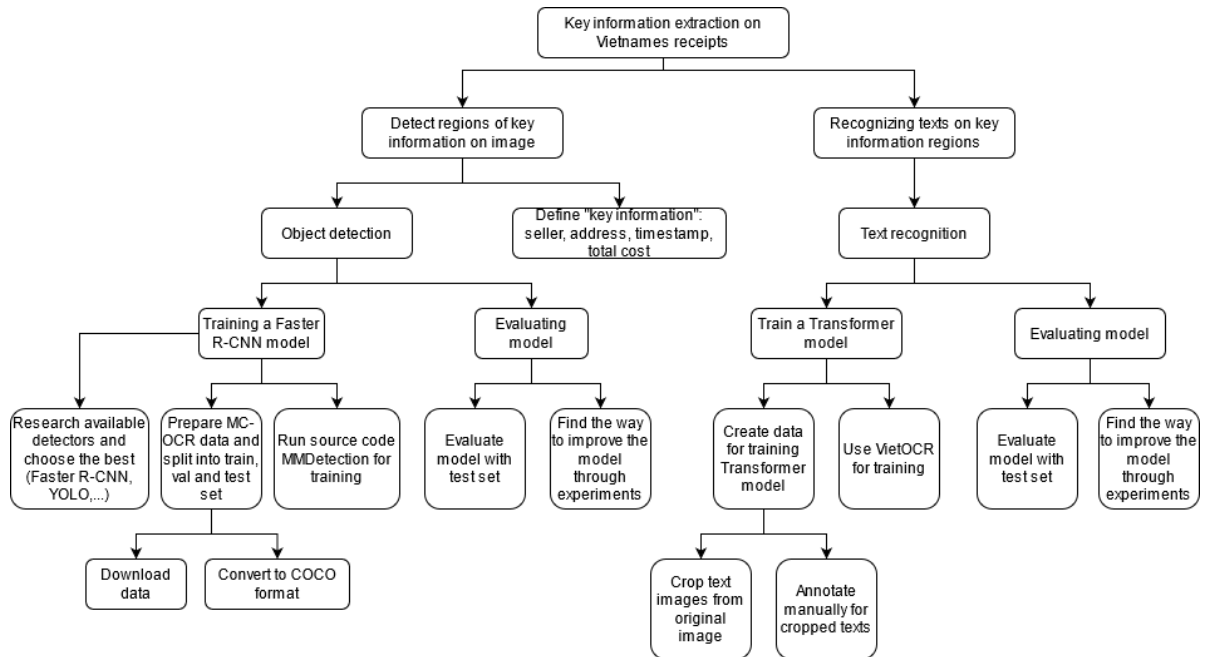


Figure 2 Process of decomposition

Figure 2 show our hierarchical structure for decomposition. In overall, we divide the problem into 2 sub tasks: localization (which means detecting regions of key information on image) and recognition (which means recognizing texts on key information regions). In both the localization task and recognition task, we divide it into more sub-tasks that can not be divided anymore for convenience in solving the process.

1.3.3. Pattern recognition

Localization: we recognize that this task is an object detection problem, so research on some available detectors and choose Faster R-CNN, a two-stage detector because this task requires and accepted accuracy. We train a Faster R-CNN model using ResNet-50 architecture as the backbone for detecting key information. The MC-OCR data is used and converted to COCO format for training.

Recognition: in this sub-task, we train a Transformer model using VietOCR source code. The dataset used for training this task is created from the original MC-OCR dataset.

1.3.4. Algorithm design

After solving all the sub-tasks, we propose our pipeline for the “key information extraction on Vietnamese receipt” problem. The detail of our pipeline will be described more clearer in the Method section.

2. Related works

In this section, we survey some previous approaches of Receipts OCR on Vietnamese receipts. In detail, we survey from studies of top participants in the RIVF2021 MC-OCR Competition. Besides, we also describe the theory of object detectors, Transformer, ResNet, ResNeXt and VGG-19 architecture which are used in our study.

2.1. Previous approaches

In MC-OCR competition 2021, which is the first challenge on Vietnamese receipts, the approaches of participants can be divided into two main groups:

- Detect key information first, recognize texts later: this approach is similar to ours. They also divide the problem into 2 steps: localization and recognition. The localization step will extract all key information regions on an image, and the recognition step will recognize text on these cropped images.

- Recognize all texts first, detect key information later: this approach is the accurate opposite of our approach. These teams use this approach to first extract all texts are contained in receipt images, then they train a deep learning model for detecting the key information in these texts.

2.2. Deep CNN architecture

In this section, we introduce some backbone architectures that are utilized in Faster R-CNN model and the Transformer model.

2.2.1. VGG-19

VGG is a classical convolutional neural network architecture. The authors [8] do several experiments with different depths of CNN networks for analyzing the impact of depth on performance. The depth is increased through convolutional layers with the kernel size is 3×3 , combined with pooling layers, activations. Experiments showed that it is possible to achieve higher performance by pushing the depth to 16-19 weight layers. In this study, we use VGG-19 as the backbone for producing feature map input of the Transformer model. The illustration of VGG-19 we show in Figure 3.

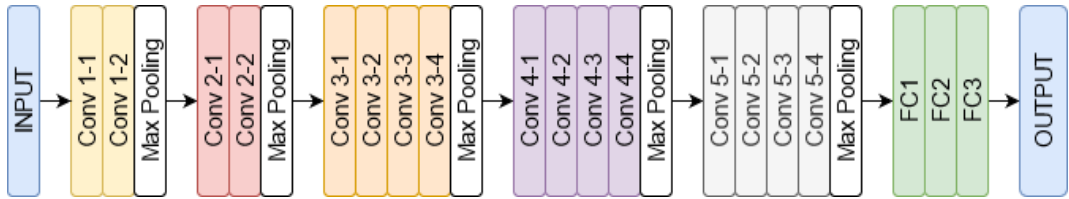


Figure 3 VGG-19 architecture

2.2.2. ResNet-50

Previous deep CNN architectures existed the limitations that the weights of last layers are almost not be updated, this is called the "vanishing gradient". ResNet architecture is proposed by [1] for controlling the vanishing gradient problem by skip connection. For more detail, the authors proposed the refined residual block and a pre-activation variant of the residual block so that the gradients can flow through the skip connections to any previous layers unlimitedly. The residual block includes convolutional layers, its feature map input is added with the output of the last layer in the residual block, then goes through a ReLU activation for producing the final feature map output, this is called skip connection.

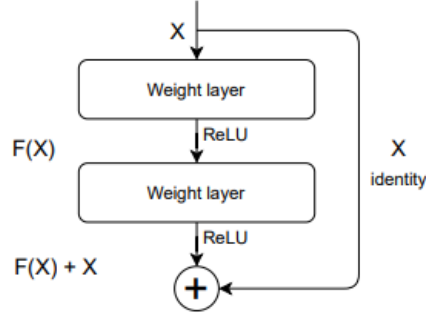


Figure 4 Residual Block

ResNet50 is used as the backbone architecture of our study when doing an experiment on Faster R - CNN. ResNet50 uses a residual block that includes three convolutional layers (two layers use 3×3 kernel and one layer uses 1×1 kernel). The first two layers have the same output dimension but the last is four times deeper. The illustration of ResNet50 is shown at Figure 5.

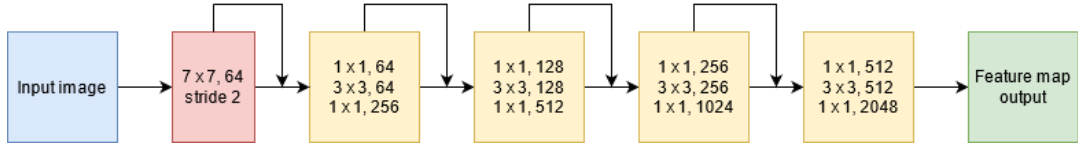


Figure 5 ResNet50 architecture

2.2.3. Faster R-CNN

Proposed by [2], Faster R - CNN is the improvement of Fast R - CNN. This is the classical object detector but still, achieves good performance on problems related to object detection. The main contribution of the authors is the Regional Proposal Network (RPN) and mechanism for sharing convolutional layers between Fast R - CNN and RPN.

First, the input image is passed into the backbone architecture, then the feature map output is fed into RPN. The RPN produces some region of interest (RoI) may include object, these ROIs are rescaled to have the same resolution by the RoI Pooling. Then these samples are fed into a detection network (Fast R - CNN) for localization and classification. The general architecture of Faster R-CNN is shown in Figure 6.

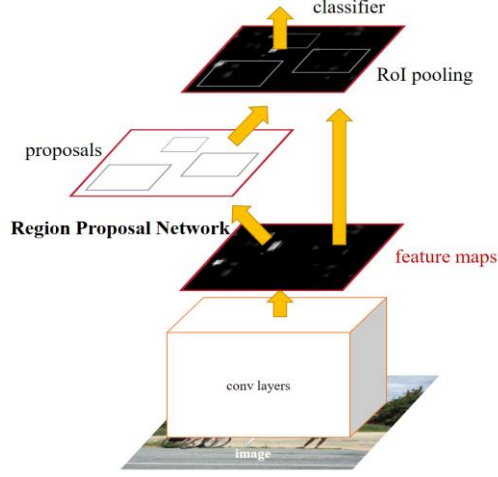


Figure 6 Diagram describes general architecture of Faster R-CNN

To train RPN for proposing ROIs, the authors use the anchor generator, which uses a sliding window size $n \times n$ on a feature map produced by backbone architecture. At each position the window slides through, there are k anchor boxes are generated. In the original paper, the authors use 1 square, 2 rectangles with a width and length ratio of 1-2, 2-1, along with 3 different sizes, so we have $k = 9$. These anchor boxes then are calculated IoU with ground truths, just keep samples that qualify the condition below:

$$\text{Label} = \begin{cases} 1 & \text{if } \max \text{IoU}(b, G) > 0.7 \\ 0 & \text{if } \max \text{IoU}(b, G) < 0.3 \end{cases}$$

After labeling, the anchor boxes will be fed into the RPN for training. The loss function used to train the RPN is a combination of a binary classification loss function with or without an object and a bounding box coordinate regression loss function, defined as Equation 1:

$$\mathcal{L} = \frac{1}{N_{\text{cls}}} \sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{reg}}} \sum_i p_i^* \mathcal{L}_{\text{reg}}(t, t_i^*) \quad (1)$$

The special thing of Faster R - CNN is that both RPN and Fast R - CNN networks use the same few convolution layers and use the same entire image features. In the original paper, Ren and his colleagues used the alternate training strategy to conduct their experiments.

2.2.4. Transformer architecture

In this section, we survey the Transformer model, which we use for OCR task. The Transformer includes two parts: the CNN backbone network for producing feature

map of an image and the Transformer architecture. As the original paper [3], the Transformer architecture includes two parts: encoder and decoder.

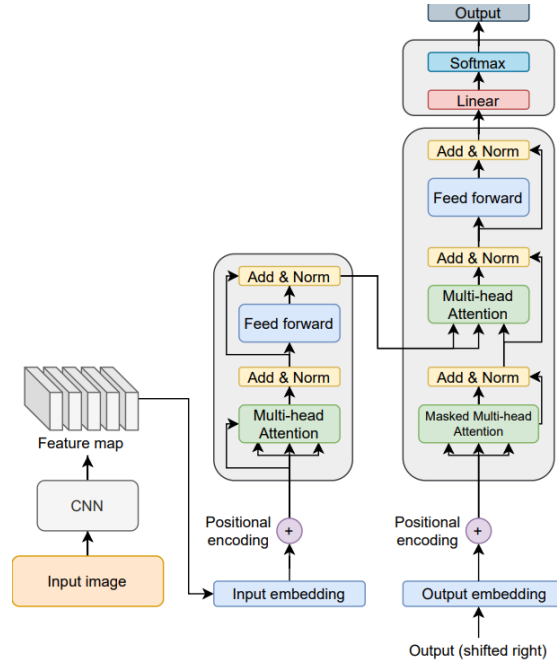


Figure 7 Diagram describes Transformer Encoder and Decoder architecture

- Encoder: The encoder includes N stacked identical layers. Each of them contains two sub-parts: Multi-head mechanism and Position-wise Feedforward network. These layers are followed by a Norm layer, which is quite similar to Batch normalization. Between these two parts, the authors use a residual connection, decreases the impact of vanishing gradients. For each stacker layer, the input vector from positional encoding is fed into Multi-head, the output is added with the input and continues to be passed through the norm layer. It is passed through a feed-forward network which is applied to each position separately and identically.
- Decoder: Similar to Encoder, Decoder has N stacked identical layers which have three sub-parts in each of them. The three sub-parts are Masked Multi-head mechanism, 2D Multi-head mechanism, and position-wise feed-forward network. For each stacked layer, the input vector from positional encoding is passed through Masked Multi-head, the output is added with the input vector, then fed into a Norm layer. The output continues to be passed to the next 2D multi-head mechanism.

3. Methodology

As Figure 3, an image is fed into an object detector that is already trained. But the raw predictions are still not used because of location confusion. Here we do a heuristic

based on our observation of the common structure of receipts. Then these predictions are cropped and apply binarization and two different text skew correction methods. Finally, we fed these solved cropped images to the trained Transformer model of the corresponding type of information to receive the final results.

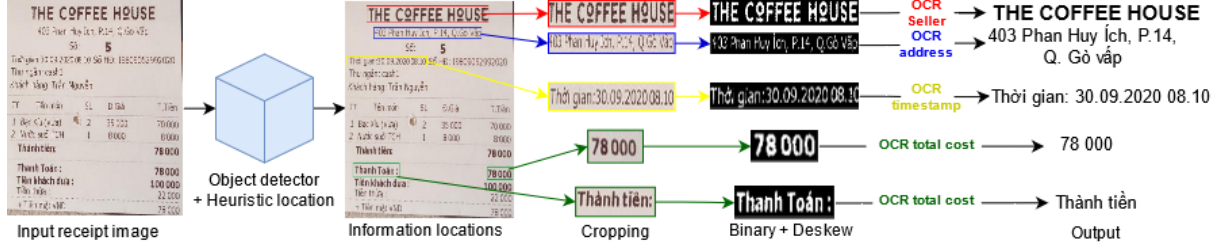


Figure 8 Visual of pipeline for solving "key information extraction" problem

3.1. Data preparing

3.1.1. Training detector

The original MC-OCR dataset (we will introduce in Section 3) annotate bounding boxes of information and its text. We use bounding boxes annotation for training object detectors. However, we have to convert the original annotation to COCO format for training. In the training set which is provided by MC-OCR organizer, we cut 20% of this data as test set for evaluating performance of object detection.

3.1.2. Training Transformer model

For the training Transformer model, we create a new dataset that just contains the pair of text images with its text ground truth, split into four types of information. We also do data augmentation this dataset by combining binary images with original images, with the expectation that the model can learn both general semantic of text images and its binary version.

3.2. Text image pre-processing

We apply the Projection profile algorithm to calculate the skew angle of a text image. This task includes following steps:

- Convert to grayscale image by the Formula 2:

$$I' = B \times 0.1140 + G \times 0.5870 + R \times 0.2989 \quad (2)$$

- Convert grayscale image to binary image by using Otsu threshold
- We rotate the image at various angles in range of $[-5;5]$ and generate a histogram of pixels in each iteration. To determine the skew angle, we compare the

maximum difference between peaks and using this skew angle, rotate the grayscale image to correct the skew.



Figure 9 Visual of text image pre-processing

3.3. Predictive location post-processing

The predictions of object detectors are not the final because of the location confusion. We must apply our rule-based solution for order the correct of information positions of each class based on our observation.

- **SELLER and ADDRESS:** these two pieces of information are normally top-to-bottom on the invoice, so we order them based on the y-value of the top-left coordinate of the bounding boxes.
- **TIMESTAMP and TOTAL COST:** these two pieces of information are normally left-to-right on the receipt, so we order them based on the x-value of the top-left coordinate of the bounding boxes.



Figure 10 Examples of located bounding boxes

3.4. Model configuration

3.4.1. Faster R-CNN detector

We use the MMDetection toolbox [4] for training two object detectors Faster R-CNN with ResNet-50 backbone. In general, we train 30 epochs, use an SGD optimizer,

the learning rate is $2e-2$. Besides, we apply $1e-4$ weight decay. All images are resized to the resolution of 1333×800 and we apply RandomFlip data augmentation. Other hyperparameters we set as default.

3.4.2. Transformer model

We use the VietOCR toolbox [5] for training four Transformer models corresponding to each type of information. To achieve better performance, we use a pre-trained model on this toolbox, this model is trained on a dataset of 10 million images, including many different types of images such as self-generated images, handwriting, and actually scanned documents. We train 10000 iterations on our data, other hyperparameters are set as default.

4. Experiments

4.1. MC-OCR dataset

In this study, we do experiments on a public dataset published in the "RIVF2021 MC-OCR"¹. The dataset includes 1149 images of receipts contain 6550 bounding boxes that are annotated in 4 classes: seller, address, timestamp, total cost; each image also has texts as ground truth. The distribution of information types of datasets and some samples of dataset are shown in Figure 11 and Figure 12:

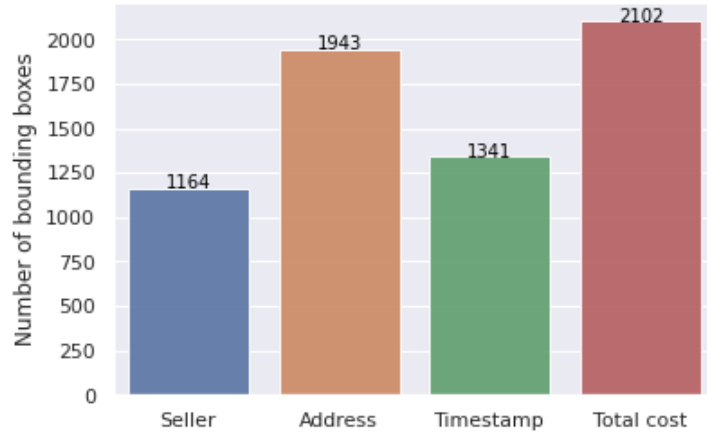


Figure 11 Chart describes the distribution of information types of datasets

¹ <https://competitions.codalab.org/competitions/27798>



Figure 12 Some samples of dataset

4.2. Metric

4.2.1. Mean Average Precision (mAP)

To evaluate object detection method, we compute Average Precision AP for each class which is the average of AP with IoU threshold i in $[0.5, 0.95]$. Besides, we also compute mAP which is the average of AP of all classes with different IoUs:

- mAP : average of mAP with IoU threshold i in $[0.5, 0.95]$
- mAP_{50} : mAP with IoU threshold = 0.5
- mAP_{75} : mAP with IoU threshold = 0.75

4.2.2. Character Error Rate (CER)

We use the CER score as a metric for evaluation, which is the same metric as Task 2 in the MC-OCR competition.

First, Levenshtein distance [6] of all fields is computed. Then, the normalized score of the Levenshtein distance of all key information is calculated as the final score. The formula for the CER score is represented by the formula below:

$$CER = \frac{1}{L} \sum_{i=1}^N (i + s + d)$$

Where:

- $L = \sum_1^N l_i$: the total length of all reference texts of the test set.
- l_i : the length of i^{th} document.
- N : total numbers of text samples.
- $(i + s + d)$: corresponds to the minimal numbers of character insertions i , substitutions s , and deletions d required to transform the reference text into the OCR output.

4.3. Results

After the experiment, we receive the performance of object detection Table 1 and Table 2. The final result is shown in Table 3. We also do a comparison with the top three participants.

Table 1 Performance of object detection of each class

Class	AP[50-95] (%)
SELLER	66.5
ADDRESS	67.3
TIMESTAMP	67.2
TOTAL_COST	60.9

Table 2 Mean average precision on different IoU threshold

IoU threshold	AP (%)
[50:95]	65.5
50	94.6
75	80.7
FPS	12fps

Table 3 Result and comparison with top three team participants

Method	CER (%)
DataMining VC	22
SDSV_AICV	23
Our pipeline	25
SUN-AI	26

4.4. Discussion

As can be seen in Table 3, our result is competitive if we compare it to the top three team participants. Our result is 1% higher than the 3rd team and just 2% and 3% lower than the 2nd and 1st team. The performance of object detection is also well when mAP is recorded at 65.5% on the test set. The AP_{50} is very high, which is recorded at 94.6%. But there are still some mistakes in our model, we use the TIDE toolkit for analyzing to determine the main type of error.

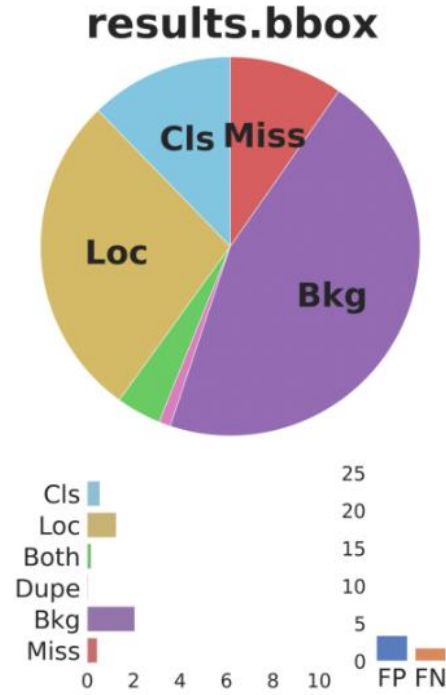


Figure 13 Mistakes are existing in detection model

Observing the pie chart in Figure 13, we can see that the Bkg part occupies the most space. It means the number of “non-information” regions are classified as a piece of key information is too high. The reason causes this may be in the Region Proposal Network module of Faster R-CNN, in the future this module should be improved for getting better performance.



Figure 14 A visualization of a prediction of our pipeline



Figure 15 Some limitations of our pipeline

The github repo of our project is available at [here](#).

5. Future work

Through experiments, we recognize that using the traditional detector Faster R – CNN is not enough, because it defines an object by two coordinates top-left and bottom-right. We think it should be at least four coordinates for the polygon to cover the pieces of key information. In the future, we will try to apply Mask R-CNN on this problem for improvement.

6. References

- [1] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [2] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015): 91-99.
- [3] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
- [4] <https://github.com/open-mmlab/mmdetection>.
- [5] <https://github.com/pbcquoc/vietocr>.
- [6] Levenshtein, Vladimir I. "Binary codes capable of correcting deletions, insertions, and reversals." Soviet physics doklady. Vol. 10. No. 8. 1966.
- [7] X.-S. Vu, Q.-A. Bui, N.-V. Nguyen, T.-T.-H. Nguyen, and T. Vu, "Mc-ocr challenge: Mobile-captured image document recognition for vietnamese receipts," in Proceedings of the 15th IEEE-RIVF International Conference on Computing and Communication Technologies, ser. RIVF '21, IEEE, 2021
- [8] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).