IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 March 2024, accepted 25 March 2024, date of publication 1 April 2024, date of current version 10 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3383911

RESEARCH ARTICLE

Efficient Text Bounding Box Identification Using Mask R-CNN: Case of Thai Documents

PHANTHAKAN KIATPHAISANSOPHON[1], DITTAYA WANVARIE[2], (Member, IEEE), AND NAGUL COOHAROJANANONE[2], (Member, IEEE)
[1]Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand
[2]Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand

Corresponding author: Dittaya Wanvarie (dittaya.w@chula.ac.th)

This work was supported by the Higher Education for Industry Consortium (Hi-FI) under Experiential Learning Program, Office of the Permanent Secretary (OPS), Ministry of Higher Education, Science, Research and Innovation.

ABSTRACT Text detection is a fundamental task in computer vision, particularly for Optical Character Recognition (OCR) applications. This study focuses on text detection within an OCR application, encompassing text detection, text recognition, and information extraction, explicitly focusing on text detection. Character-Region Awareness for Text Detection (CRAFT), Pyramid Mask Text Detector (PMTD), and Scene Text Detection with Supervised Pyramid Context Network (SPCNET) have demonstrated promising results in bounding-box detection. However, it faces challenges related to post-processing and multiline text detection. A post-processing problem arises because of the need to reconfigure the model when new documents are introduced, which leads to inefficiencies and complexities. In addition, CRAFT tends to merge bounding boxes from consecutive lines by introducing multiline errors, especially for CRAFT. To address these challenges, this study proposes an adapted approach based on Mask R-CNN, an instance segmentation model that treats each text element as an individual object. By adopting the Mask R-CNN approach, post-processing issues were successfully eliminated. Moreover, the multiline problem is effectively resolved. Comparative experiments demonstrate that the proposed model achieves results comparable to those of these models while surpassing them in accuracy and versatility. The proposed model is extensively evaluated on various document types, including bankbooks, Thai ID cards (both front and back sides), invoices, car registrations, mobile banking slips, passports, Indonesian ID cards, driver licenses, and receipts. The results indicated the model's high performance and potential for real-world applications. Eliminating post-processing and multiline problems ensures the model's adaptability to a wide range of document structures and reduces both time inference and resource utilization.

INDEX TERMS Deep learning, text detection, optical character recognition (OCR).

I. INTRODUCTION

Text detection is a computer vision process that involves the identification of text within an image. It has gained popularity in various applications, such as Optical Character Recognition (OCR) [1], image text translation [2], and document classification [3], because text detection plays a vital role in these processes. Specifically, text detection aims to locate the areas in an image where text is present and define bounding boxes around the identified texts.

OCR, an application used to interpret text in images or documents, involves three primary steps: text detection [4], text recognition [5], and information extraction [6]. However, our focus is solely on text detection, specifically finding a bounding box around the text within an image.

We experimented with various methods for extracting bounding boxes, such as SPCNET the Supervised Pyramid Context Network (SPCNET) [7], PMTD (Pyramid Mask Text Detector) [8], and Character-Region Awareness for Text detection (CRAFT) [9]. Among these, CRAFT has proven to be the most efficient for detecting text bounding boxes in our OCR compared to the CRAFT method we used.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

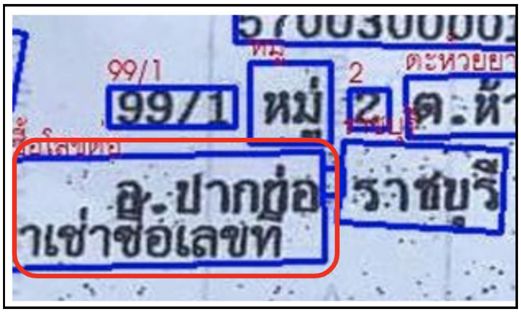


FIGURE 1. A multiline problem is shown in a red rounded rectangle. There are two lines of black text in one blue bounding box.

Although the CRAFT technique is highly accurate, some significant issues must be addressed. However, the CRAFT model cannot be effectively generalized for every document. When new services or document types are encountered, they must be adjusted to fit the new document type for the best result. This is the main issue related to post-processing problems, including the need to adjust or set new parameters for the post-processing stage whenever we have an improved model or encounter new document types. Thus, post-processing is necessary. However, conducting post-processing for each document type is challenging because of the diverse range of document structures, which requires time-consuming grid searches to determine optimal parameters. This approach is unsuitable for handling new document types. Moreover, during post-processing, we can overlook the parameters that require tuning for each service. However, accuracy may be affected by the size of the document and the text within it. This variability is the primary reason why we must engage in post-processing is required to define the final bounding boxes.

In addition, there is a problem in detecting multiline text, where bounding boxes may encompass multiple lines of text instead of separate boxes for each line. If two distinct lines are close to each other, the text-detection model might erroneously group them into a single bounding box (as shown in Figure 1.). This issue is commonly observed in various document types.

This problem arises because the model can become confused when texts are closely situated as it attempts to localize each object at the segmentation level. To overcome this challenge, we propose training a new model using an alternative approach known as instance-level object detection. This model identifies each word as an individual object, eliminating the need for post-processing of different document types. The final output of the model provided individual bounding boxes for each text element.

Furthermore, this approach solves the multiline problem by treating each text element as a separate object. This model recognizes each character as a distinct object, eliminating the necessity of post-processing different document types. The final output of the model yields separate bounding boxes for each text element. In addition, this approach resolves the multiline problem by treating each text element as an independent object.

In contrast to the CRAFT model, which requires post-processing and threshold-based box separation to obtain the final individual bounding boxes, our new approach generates bounding boxes for each object directly in the model's output. This means the model resolves the need for post-processing and effectively addresses multiline issues.

In this paper, we present a model for text detection in images of documents such as bankbooks, Thai ID cards (front and back), invoices, receipts, car registration books, mobile banking slips, passports, Indonesian ID cards, and driver's licenses. Building on this, our research proposes a Mask R-CNN-based method that effectively overcomes the inefficiencies associated with post-processing and multiline errors observed in the CRAFT model. The result is an improved level of accuracy, versatility, and adaptability across diverse document types, along with a reduced inference time and resource consumption during model deployment.

II. RELATED WORKS

A. REGION-BASED AND REGION-AWARENESS TEXT DETECTOR

Region-based techniques in computer vision are fundamental approaches that focus on analyzing and processing specific regions or areas of an image. These techniques are crucial for object detection, segmentation, and region-specific image processing tasks. Region-based methods enable the identification, classification, and detailed analysis of objects or regions of interest within images. In this approach, the CRAFT uses the region in the image to analyze characters and their relationships. Another approach for this type is Mask R-CNN [10], [11], [12], instance segmentation [13], extended from the Faster R-CNN [14], [15] architecture, which also enhances region-awareness that simultaneously identifies objects and generates high-precision pixel-level masks, effectively "masking" individual object instances within an image. By combining object detection and semantic segmentation, Mask R-CNN allows for the capture of fine-grained details and boundaries of objects. Pose estimation [16] is a basic example of an application that uses Mask R-CNN. In the context of text detection, several GitHub projects show individuals employing a text-detection model trained using Detectron2 [17], [18] as the underlying framework.

Another approach is the segmentation training technique YOLOv8 [19], which divides an image into a grid and predicts bounding boxes and class probabilities directly from the grid cells. The bounding boxes are then associated with the regions in the input image. This is in contrast to traditional object detection methods such as Faster R-CNN, which involve region proposal networks (RPNs) to generate potential regions of interest before classifying and refining bounding boxes. Most studies with YOLOv8 focus on real-time object detection, such as real-time orchard tree segmentation [20].

B. MULTIPLE LAYERS-BASED DETECTOR

Another common approach is based on studies dealing with the pyramid-based architecture, SPCNET, for such networks to be used for tasks such as image segmentation, object detection, and scene parsing, leveraging the multi-scale context information of the Feature Pyramid Network (FPN) to improve the precision and accuracy of these tasks. Another pyramid-based PMTD can be inferred as a type of object-detection framework that incorporates a pyramid-based approach. Pyramid structures are commonly employed to analyze objects on multiple scales, which can improve the accuracy and robustness of object detection tasks. The ''mask'' component in the Pyramid feature suggests that this method might be related to instance segmentation, where precise pixel-level masks are generated to delineate individual objects within an image.

For more information on the mask involved in the pyramid feature and Mask R-CNN, the mask refers to the pixel-wise segmentation mask generated by the model for each instance of an object in an image. The ''mask'' in Mask R-CNN is essentially a binary image where pixels inside the object boundary are set to 1, indicating the presence of the object, and pixels outside the boundary are set to 0. These masks provide fine-grained segmentation of objects, enabling the model to understand not only where objects are located in an image but also which pixels specifically belong to each object instance.

Multiple layers from an FPN are widely used to extract rich feature representations. Zhang et al. [21] proposed an active pedestrian detection system that operates on multiple layers to address the performance of detecting small pedestrians far from a camera. The key idea is that pedestrians of different sizes have distinct appearances, and multilayer neuronal representations capture these differences effectively.

Another work by Zhang et al. [22] used multiple layers from an FPN to enhance the performance of capturing the difference scale of a pedestrian. High-level convolutional layers, such as ResNet-50-C4 and ResNet-50-C5, are used to extract ROI features and perform feature aggregation for pedestrian detection at different scales.

More examples of using multiple layers from the FPN, Zhang et al. [23]. Utilizing multiple layers in this study allowed the network to acquire hierarchical representations. The lower layers capture basic features, like edges and textures, whereas the higher layers capture more complex and abstract features. By combining these features across various layers, the network enhances its abilities, leading to improved accuracy in both the classification and regression tasks, particularly in visual tracking.

C. HYBRID TEXT DETECTOR

This approach combines many features to obtain text detection results. TextFuseNet [24] extracts and captures richer fused features for text localization. It utilizes three branches: semantic segmentation [25] for global semantics, word and character detection, and masks, for instance, segmentation. These branches generated richer fused features for text detection. TextFuseNet incorporates a feature pyramid network (FPN) [26] for multi-scale feature extraction and a region proposal network (RPN) for text proposal generation. It employs a semantic segmentation branch to obtain global features. The detection branch predicts the categories and performs bounding box regression using word- and global-level features. The mask branch performs instance segmentation using character, word, and global-level features. A multipath fusion architecture aligns and merges extracted features for robust text detection, particularly for arbitrary shapes. The fusion process enhances text detection by combining text representation and fusion techniques. This approach enables more robust and accurate detection of text areas of various shapes.

D. DETECTRON2 AS A FRAMEWORK

Detectron2,[1] developed by the Facebook AI Research team, is a cutting-edge object detection framework renowned for its modular architecture, exceptional performance, and widespread adoption in computer vision applications. This method is notable for its versatility, ease of use, and top-tier accuracy on the benchmark datasets.

Detectron2 in comparison to other frameworks, offers a winning combination of accuracy, flexibility, and efficiency. Its widespread adoption is attributed to modular design, community support, and rich feature sets.

Researchers and developers can leverage Detectron2 to build, train, and deploy models for tasks, such as instance segmentation, object detection, and keypoint detection. With this, Detectron2 provides a powerful model, Mask R-CNN, for instance segmentation, capable of providing pixel-level segmentation masks for objects in an image.

Furthermore, the outputs generated by the Mask R-CNN model revealed four integral components that provided details pertaining to object detection and segmentation. These primary components encompass bounding boxes, denoted as boxes, comprising coordinates (x, y, width, and height) that encapsulate the identified bounding boxes surrounding objects. Additionally, the class labels, denoted as classes, represent the predicted classification for each identified object. The scores, referred to as confidence scores, are accompanied by the predicted bounding boxes and reflect the level of certainty associated with the predictions. Finally, masks, characterized as segmentation masks, delineate the detected objects and demarcate their spatial extent.

To derive the final bounding box for each text, we used segmentation masks to extract the bounding boxes because the segmentation masks provided the area of the text at the pixel level. We cannot use the bounding boxes directly because if the model faces the rotated images, the coordinates of the bounding boxes will give us strange boxes (as shown in Figure 2. and Figure 3.) owing to the structure of the bounding boxes (x, y, width, and height).

---

[1] https://github.com/facebookresearch/detectron2

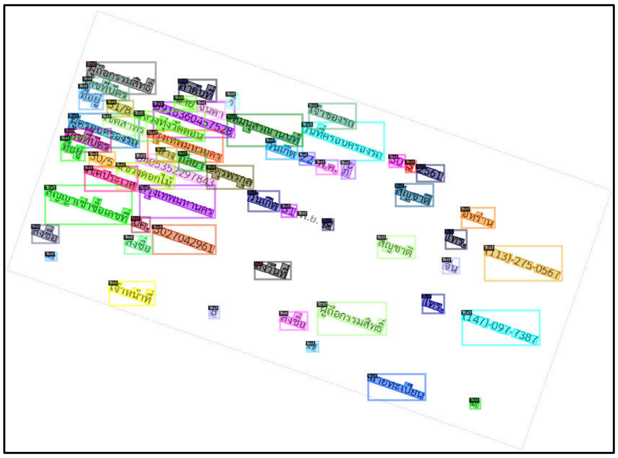


FIGURE 2. The rotated image with the ground-truth of bounding boxes and segmentation masks in COCO format.

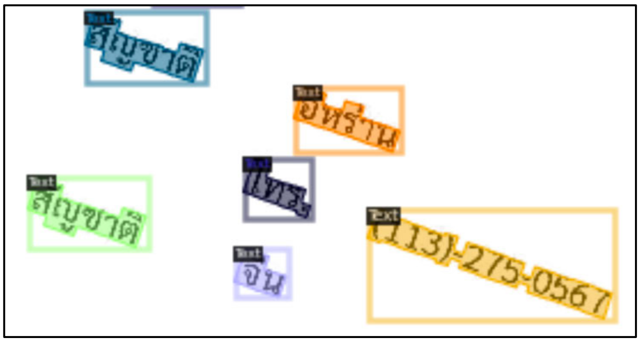


FIGURE 3. The image is cropped from the rotated image, there is a bounding box and mask represented in it. The bounding box is rectangle coordinate, but the mask is span along the characters area.

III. EVALUATION METRICS, TYPEFACE ANATOMY AND MODEL FUNDAMENTAL

A. TEDEVAL- TEXT DETECTION EVALUATION

TedEval is an evaluation metric designed specifically for scene-text detection models [27]. It also addresses common issues, such as granularity, multiline text, and character incompleteness. Unlike the traditional Intersection over Union (IoU) approaches, TedEval is suitable for detecting both single- and multi-character texts. It evaluates the results through instance-level matching and character-level scoring, focusing on granularity and completeness. By creating pseudo character centers (PCC) from word bounding boxes and lengths, TedEval computes matches and penalizes missing or overlapping characters.

TedEval was designed to evaluate the scene-text-detection model. It addresses common issues, such as granularity, multiline, and character incompleteness (as shown in Figure 4.). The IoU approach is a standard method for assessing an object detection model by determining the intersection between the union of the detected box and ground truth. However, this method is unsuitable for text detection because several characters exist in a single text. TedEval also addressed this problem by changing the detection level from the object or character level to the instance level. However, TedEval still reports character-level scoring with a focus on granularity and completeness. The granularity score was computed from the character alignment between the prediction and ground truth. There were one-to-one, one-to-many, and many-to-one character matches that were used as error cases. Because we do not have a character-level ground truth, TedEval creates pseudo-character centers (PCC) from a word (or text) bounding box and its length. TedEval can then find the IoU from the instance (or text) level while reporting a penalty for missing and overlapping at the character level.

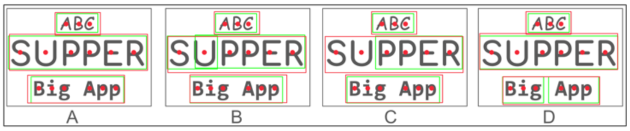


FIGURE 4. Examples of incomplete detections. Red dot: PCC. Red box: ground-truth, Green: detection. A: Complete detection, B: One-To-Many detection, C: Incomplete detection, and D: Split detection (Big App).

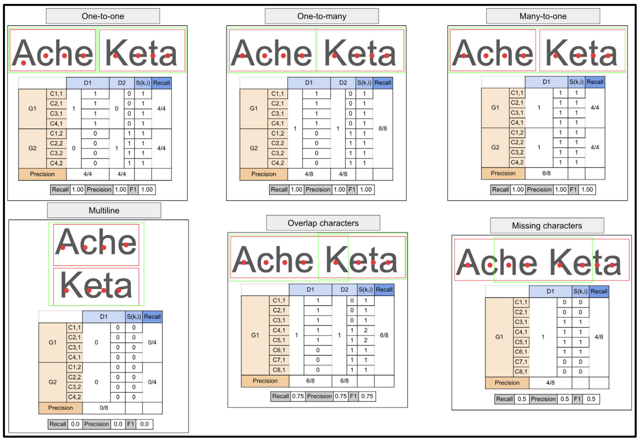


FIGURE 5. Examples of computing the score from each error case. Red dot: PCC. Red box: ground-truth, Green: detection.

TedEval provides recall and precision at the character level using PCC from word-level bounding boxes and word lengths (as shown in Figure 6.). It can compute the score of an entire match from a partial match by penalizing the missing or overlapping characters(as shown in Figure 5.). If a word is split into multiple detections, TedEval classifies it as a one-to-many error and penalizes the prediction by decreasing the

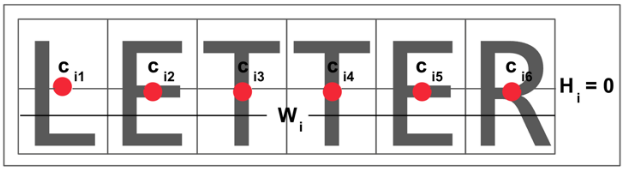


FIGURE 6. Example of computing PCC of Gi. Red dot: PCC. Red dash: pseudo character box. Grey: Ci (ground truth of image i).

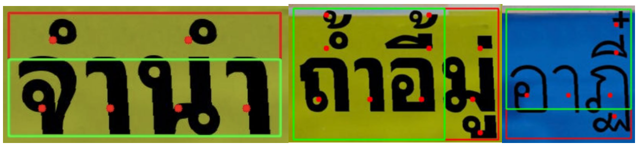


FIGURE 7. In the example images, there are the red dots appear upper and lower from the normal line. This assure that if the word has the upper or lower component, we can count the score accurately. The red boxes show the ground-truth annotation, the green boxes show the prediction result from the detection model.

total matching score. Apart from the model evaluation and error analysis, TedEval can ensure that the text detection result is sufficiently reliable for the subsequent text recognition process. Even though we do not have the ground truth at the character level, or the text-level ground truth needs to be cleaner, TedEval can still evaluate the detection result effectively.

For the Thai language, there are many special components that differ from those of the English language; we will discuss these components in the following sub-section. An example of a special component is the upper and lower vowels that appear in Thai (as shown in Figure 9).

Furthermore, we implemented TedEval to gain further insight into the specific errors that occurred during the evaluation. This includes identifying issues, such as missing characters, multiline text detection, missing single characters, or situations in which one-character maps to multiple or multiple-character maps to one. Precisely, for a language with upper and lower components, such as the Thai language, in this study, we can also count the number of error cases involving missing upper- or lower-case components.

Errors in the Thai language (as shown in Figure 7), the upgraded version of TedEval that we extended from the original code, will count if the upper or lower vowels are not detected, extended error cases with missing upper and lower characters.

B. ENGLISH AND THAI TYPEFACE ANATOMY

Typeface anatomy is a fundamental concept in typography that includes visual elements and components that make up a typeface [28]. Understanding these elements, such as baseline, cap height, x-height, serifs, stems, and counters (as shown in Figure 8), is essential for designers and typographers to create well-designed and legible texts. The complexity of these elements contributes to the overall aesthetics and readability of the written communication.

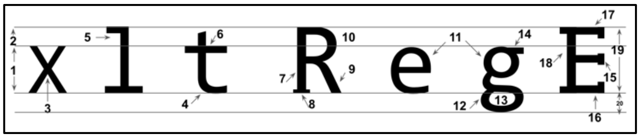


FIGURE 8. Typographic parts of a glyph: 1) x-height; 2) ascender line; 3) apex; 4) baseline; 5) ascender; 6) crossbar; 7) stem; 8) serif; 9) leg; 10) bowl; 11) counter; 12) collar/link/neck; 13) loop; 14) ear; 15) tie; 16) horizontal bar; 17) arm; 18) vertical bar; 19) cap height; 20) descender height.

In the field of technology, computer vision and understanding typeface anatomy are essential for tasks such as text detection. Knowledge of typeface anatomy is essential for building accurate detection algorithms that can identify text regions regardless of the typeface used, orientation, or layout variation.

Typeface anatomy is the basis of typography and influences design, communication, and technology. Its significance has spread across diverse languages and applications, from crafting culturally appropriate typefaces to developing advanced text-detection algorithms that support various aspects of our digital lives.

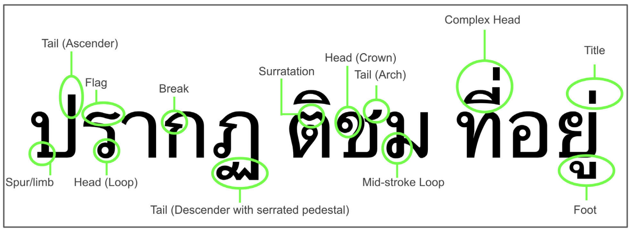


FIGURE 9. Variation in Thai Typeface Anatomy. This image shows diverse typographic elements in Thai fonts, including character height, head, and tail, highlighting the rich spectrum of design choices in Thai typography.

In our case, in terms of typeface anatomy, languages such as Thai were used (as shown in Figure 9), and there are unique considerations owing to the distinct script characteristics. Thai script features loops, curves, and stacking elements that influence the design of typefaces. In addition, the writing can go above and below normal lines, with or without connecting parts. This is important in text detection. We need to be careful about the upper and lower parts of the letters because these distinctions might confuse the model and might not capture all the text correctly.

Understanding how typefaces work, like building blocks for typography, affects design and technology. It is important across different languages and uses, from creating fonts that match different cultures to creating smart algorithms that form the foundation of text detection.

C. FUNDAMENTAL OF OUR PROPOSED MODEL

The mask region-based Convolutional Neural Network (R-CNN) is a state-of-the-art model for object instance segmentation. It extends the Faster R-CNN architecture by adding an additional branch to predict segmentation masks in parallel with the existing branches for object detection and bounding box regression.

For some of the fundamental concepts of Mask R-CNN, one needs to know before performing the model training:

1) BACKBONE NETWORK

The purpose of the backbone network is to extract hierarchical features from the image. We used several backbones to compare the results while tuning the hyperparameters. The results show that the more complex the backbone, the better the performance. The results of the backbone architecture are presented in Section VI.

2) REGION PROPOSAL NETWORK (RPN)

A Mask R-CNN utilizes an RPN to generate region proposals for potential object instances. RPN proposes candidate bounding boxes and assigns scores to them.

3) REGION-BASED ROI ALIGN

This operation enables precise pixel-to-pixel alignment between the extracted features and the output masks.

4) OBJECT BRANCH

This branch is used to predict class labels and refine the bounding box coordinates for each proposed region. In our work, we focus on mask prediction because it is more versatile than the object branch.

5) MASK BRANCH

It is a parallel branch with an object branch to predict the segmentation masks for the identified objects. Unlike the object-detection branch, the mask branch produces an output at the pixel-to-pixel level.

6) LOSS FUNCTIONS

The model is trained using multiple loss functions, including classification loss for object detection, bounding box regression loss, and mask segmentation loss. These losses are combined to form a comprehensive training objective. The option of the loss function that detectron2 provides for us: Smooth_L1, GIoU, DIoU, and CIoU. This is particularly true in the training of bounding-box regression models. These loss functions aim to quantify the difference between the predicted bounding boxes and the ground truth bounding boxes. The choice of the loss function to use may depend on the specific characteristics and requirements of the object detection task. The results of the loss function are used in Section V.

IV. EXPERIMENT SETTING

Our main goal is to use an instance-level object detection model with the Detectron2 framework to accurately locate the bounding box within an image. To this end, the experiments were divided into three phases. This first phase obtains the best dataset by training the model using default hyperparameters. The second phase used the best dataset from the first phase to perform a grid search and fine-tuning to determine the best parameters for enhancing the accuracy of the model. The final phase uses the final model to integrate the detection model with an end-to-end pipeline to extract and recognize text from images in various applications. Examples include Thai ID cards (both front and back), passports, and car registration. The overall training process is illustrated in Figure 10.

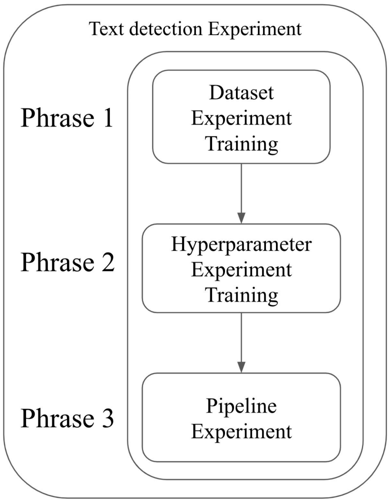

Text detection Experiment
Phrase 1
Dataset Experiment Training
Phrase 2
Hyperparameter Experiment Training
Phrase 3
Pipeline Experiment

FIGURE 10. Diagram of the training process with step-by-step instructions on how to acquire the optimal model for the purpose of testing the entire end-to-end pipeline.

The detection model is a Mask-RCNN for instance, segmentation. To facilitate the training process, we utilized Detectron2, developed by the Facebook research team.

Initially, we trained the model on Google Collaboratory Pro with a GPU that provided us with preinstalled libraries and configurations, such as CUDA, for GPU acceleration. The workstation specifications include up to 27 GB of RAM, more than 200GB of storage capacity (the storage capacity varies), Tesla T4, and P100. Unfortunately, this was difficult because the training process took a long time, sometimes more than a day, owing to the difference in epochs we had trained, that is, 18 hours once when training with 100000 epochs. However, Google Collaboratory sessions can only run for up to 12 hours. Therefore, we initially trained only for a few epochs to obtain training. Subsequently, we switched to a workstation. The workstation specifications included 128GB of RAM, 2TB storage capacity, and 24GB GPU. We also used this workstation to train YOLOv8 using Ultralytics.[2]

V. DATASETS EXPERIMENT

In this section, we present a comprehensive investigation to determine the optimal dataset for our study. Three distinct datasets–Simple, Simple+, and Simple++–were created to examine how the composition of the dataset affected our analysis. The primary difference between these datasets was the number and type of documents contained. The overall training phase is illustrated in Figure 11.

We have a diverse set of datasets, including passports, Thai ID cards (both front and back), and car registrations. However, this quantity was insufficient for training large-scale models. Because we only have a small amount of data for each type of document, training the model solely with these data may not yield highly accurate results. To overcome

---

[2] https://github.com/ultralytics/ultralytics

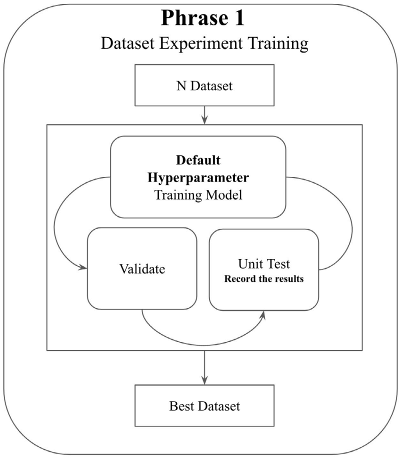


FIGURE 11. Diagram of the training process of phrase 1 which is a dataset experiment training to obtain the optimal dataset for the next phrase.

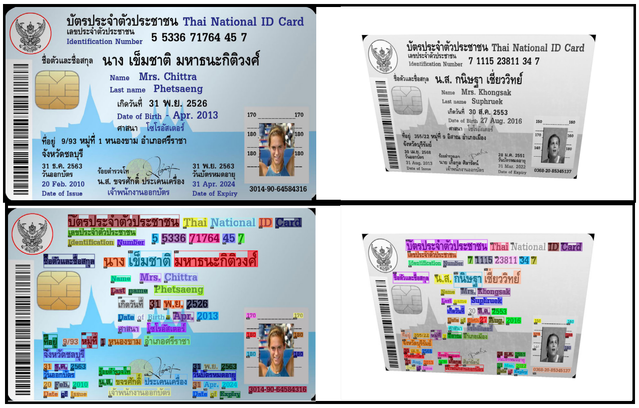


FIGURE 12. Examples of synthetic Thai ID card with ground truth COCO-instance annotation in normal cases (on the left) and augmented cases (on the right). The top upper rows are synthetic images and the bottom rows are ground-truth annotations in word-level ground-truth. And the box filled with color is the character-level ground-truth.

this limitation, synthetic data can be used to train the model instead of relying solely on real-world datasets. By generating synthetic data that resembles real documents, we can increase the overall amount of training data and enhance the accuracy of the model. Note that the training experiments were conducted using default hyperparameters.

We employed a synthesizer tool to create images, in which the tool mimicked real images during the generation process. Each document comprises template backgrounds, field positions, and a diverse pool of words that are specific to the document. During image synthesis, the tool randomly assigns synthetic words to their respective field positions (Table 1). Augmentation techniques include image rotation or conversion of the image to black and white (as shown in Figure 12.) and were applied after the initial image was generated.

TABLE 1. Field position for thai ID card synthesizer.

x	y	text	tag	size	font_color	render
252	40	บัตรประจำตัวประชาชน	lbl_id_name_th	80	(0,0,0)	static
730	40	Thai National ID Card	lbl_id_name_en	80	(39,45,35)	static
252	130	Identification Number	lbl_id_number_en	50	(39,45,35)	static
600	105	@id_number	id_number	80	(0,0,0)	regex

Table 1, an example of the position fields on the front side of the Thailand ID Card that shows the position of the field, font size, font color, and the type of the word to fill in, which is static or randomly picked using regex.

Two additional outputs are obtained from a single synthesis iteration: the ground truth at the word and character levels. These two ground truths were then utilized to construct a ground truth in the COCO format [29], facilitating the training of the detectron2 model.

The synthetic data consists of three components:

Image: The synthetic images include normal cases as well as augmented variations (as shown in Figure 12.). These variations can include faded images, distorted images, rotated images, or images with added noise. This approach allows us to expose the model to a wider range of document variations, improving its ability to handle real-world scenarios.

Word-level ground-truth: This component provides rectangular bounding box positions for each word text within the synthetic image. This allows the model to learn the spatial relationships and positions of the words within a document.

Character-level ground-truth: This component provides rectangular bounding box positions for each character text within the synthetic image. This enables the model to understand the precise positioning of individual characters within the words.

To train the model, it was necessary to convert the ground-truth data into COCO instance format. The training process requires this format to train the model.

By utilizing both word- and character-level ground-truth data, we can generate a polygon that follows the curves of the characters and serves as a mask bounding box (as shown in Figure 12.). This approach allows for more precise and accurate localization of text within an image.

A. TRAIN DATASET

We generated multiple datasets for training purposes, focusing on the three primary datasets to report the results. The remaining datasets were used for trial and error, and some contained incorrect ground truths.

1) SIMPLE

The synthetic data include Thai ID cards (both front and back), Indonesian ID cards, six types of car registration, and passports. Initially, we trained this model to ensure the correctness of the training process and ground truth. Subsequently, we generated additional samples for each document, resulting in a final dataset of 24,900 images. We used this dataset as the foundational dataset and incorporated additional documents.

2) SIMPLE+

Once the model is trained for a certain period, more document types are added to create a more challenging dataset. This new dataset includes driver licenses and bankbooks. Building on the previous dataset (Simple), this led to improved performance, particularly for driver licenses, bankbooks, and all previously included documents. The final dataset size was 30,900 images.

3) SIMPLE++

We included additional document types such as receipts, invoices, and statements in the dataset. The reason for this is that these new document types pose a higher level of difficulty owing to their numerous fields and diverse structural templates. Our assumption was that training the model on challenging data would become more robust. Subsequently, upon training the model, we observed an improved performance, particularly when using these document types. Hereafter, we will use this dataset as the primary training source for moving forward. The final dataset was comprised of 44,641 images.

Note that, for each document type, we generated varying numbers of documents because the fields in each document differed. In addition, some documents contain different sub-templates, typically with each template containing approximately 3000 images per template in each document, including normal and augmented images.

TABLE 2. Training, validation and test dataset.

	Train			Validation	Test
Document	**Simple**	**Simple+**	**Simple++**	**Test_01**	**Test_02**
Thai ID Front	5300	5300	5300	40	45
Thai ID Back	4000	4000	4000	40	45
Indo ID Card	3600	3600	3600	40	-
Car Books	9000	9000	9000	240	45
Passport	3000	3000	3000	40	45
Driver License	-	3000	3000	40	-
Bankbook	-	3000	3000	40	45
Receipt	-	-	3600	-	45
Invoice	-	-	5000	-	45
Statement	-	-	5000	-	-
Government Doc	-	-	-	-	45
Mobile Slip	-	-	-	-	45
Total	24900	30900	44500	480	405

B. VALIDATION DATASET

The purpose of this validation dataset is to assess the performance of the proposed model on unseen data during training. This helps prevent overfitting [30], fine-tune the hyperparameters, select the best model, and evaluate the generalization before deploying the model in real-world scenarios.

This dataset was generated using the synthesizer generator tool, Test_01. It resembles the training dataset but is created with a different seed, resulting in variations between the datasets. The document consists of a bank book, car registration, driver's license, Indonesian ID card, and Thai ID cards (both front and back). We generated a separate synthetic dataset for the model validation. This validation dataset consists of approximately 40 images per document template to measure the performance of the model. A total of 480 validation datasets were used: 480.

In model validation, Detectron2 offers a script to evaluate the model using mean average precision (mAP). However, in our case, we compared the accuracy of the new models with that of the CRAFT model. To ensure a fair and consistent comparison, using a central metric that produces comparable output is important. By doing so, we can measure accuracy and make meaningful comparisons.

We used TedEval as the central measure to compare the models. TedEval was specifically designed to evaluate the scene-text detector models. TedEval provides metrics, such as recall, precision, and F1 scores, which were used to assess the performance of our models. These metrics help us to understand how well our models detect text. As we needed to compare the outcomes with the earlier model, utilizing the mAP for comparison was not possible because the CRAFT model was not evaluated using the mAP. Instead, TedEval offers a more effective way to display accuracy and highlight instances of error.

The results in Table 3 show the default model's performance for the validation dataset. This outcome indicates enhancements in the model compared to the preceding training iterations and ensures that the model is not susceptible to overfitting.

TABLE 3. Score from each dataset evaluated with validation dataset.

No.	Dataset	Avg-Recall	Avg-Precision	Avg-F1
1	Simple	90.1970	90.3945	90.2940
2	Simple+	91.5323	91.4657	91.4920
3	Simple++	91.7937	92.2707	92.0271

C. TEST DATSET

We prepared a test dataset called Test_02, which is comprised of real-world images for unit testing and evaluation of the detection models. This dataset included nine document types: bankbooks, Thai ID cards (both front and back), invoices, car registration books, mobile banking slips, passports, government documents, and receipts. All models were evaluated and compared using this dataset, which has 405 real test datasets.

Table 4 shows the results of the default model for the real-world test dataset, Test_02. The results show that adding different types of data boosts all evaluation metrics.

In the Simple dataset, we initially included fundamental document types such as Thai ID cards (both front and back),

TABLE 4. Score from each dataset evaluated with test dataset.

No.	Dataset	Avg-Recall	Avg-Precision	Avg-F1
1	Simple	77.0991	75.2205	76.1482
2	Simple+	82.9550	79.1290	80.9968
3	Simple++	84.6604	81.4970	83.0486

Indonesian ID cards, car registrations, and passports. This served as our baseline dataset, providing a foundational set of documents for the analysis. We used this dataset to train the first model, and we believe that we can train a more accurate model if we add more documents to the dataset.

By expanding upon a simple dataset, we created a simple +. We added more document categories in this extended version, including driver licenses and bankbooks. This expansion aims to improve the diversity and comprehensiveness of a dataset by covering a wider range of document types.

To further explore the impact of dataset composition, we created the Simple++ dataset. In this iteration, we added receipts, invoices, and statements from Simple+ while retaining the remaining document categories. All additional documents are quite challenging; we hope that they can enhance the model performance, and these documents have been proven to enhance the model (as shown in Table 4).

We established a reliable and consistent dataset by selecting the best hyperparameters for the model. We obtained the Simple++ dataset, with 44500 images containing bank books, car registrations, passports, driver licenses, Indonesian ID cards, Thai ID cards (both front and back sides), receipts, bank statements, and invoices, as the best dataset for the next experiment, which is a hyperparameter-tuning phrase [31].

D. ADDITIONAL TRAINING DATASET RESULT

Considering the details of adding additional datasets to this experiment, we were curious about its impact on the CRAFT model when subjected to an increased number of datasets. We trained the CRAFT models using our datasets and found that augmenting the dataset did not enhance the performance of the model (Figure 13). To ensure this, we included our experiment with the proposed model; adding more data to the experiment enhanced the model's performance. As shown in the graph, the number of red dots increases significantly when the dataset is added. At the last point, which is the orange star, we also include the hyperparameter tuning results to ensure that the ability of the proposed model can overcome the CRAFT model, measured with an F1-score from the Test_02 dataset, when we perform hyperparameter tuning, which is discussed in the next section.

E. LOSS FUNCTION FOR OUR TASK

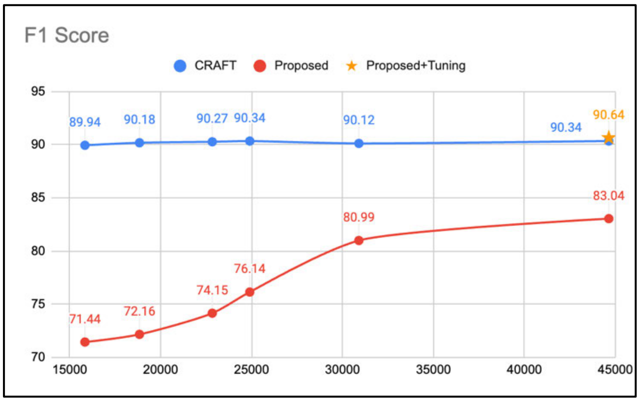


FIGURE 13. The image shows the trend of adding more amount of data into the training dataset can improve the performance in proposed model, but it does not improve in the case of CRAFT model. Note that the orange star is the result from hyperparameters tuning section. To ensure the ability of the proposed model can overcome the CRAFT.

At the early step of the experiment, we implemented the training pipeline involving data synthesis, created the ground truth for training, and then trained the model. Once we confirm that our training pipeline can execute the training accurately, we proceed to the next step, which involves identifying an appropriate loss function for our task. To achieve this, we conducted some model training, employing default hyperparameters and the dataset "Simple" while varying the loss function across the four options. The results are presented in Table 5.

TABLE 5. Score from each dataset evaluated with test dataset.

No.	Loss Function	Recall	Precision	F1
1	Smooth_L1	77.10	75.22	76.14
2	GIoU	76.84	75.02	75.92
3	DIoU	76.66	75.29	75.96
4	CIoU	76.94	74.31	75.60

The results in Table 5 indicate that the Smooth L1 loss function achieves the highest Recall and Precision values of 77.10% and 75.22%, respectively, resulting in an overall F1 score of 76.14%. While GIoU, DIoU, and CIoU showed competitive performance with relatively close metrics, the choice of the most suitable loss function may depend on the specific requirements. For our task, we decided to choose Smooth L1 because the result show it as the best performance among the others.

VI. HYPERPARAMETERS TUNING EXPERIMENT

In this section, we move from the dataset experiment (section V.) to the phase of fine-tuning the model. With a robust and stable dataset in place, our focus shifted to the critical task of optimizing hyperparameters that significantly affect the model performance. The overall training phase is illustrated in Figure 14.

This hyperparameter tuning process encompasses a thorough evaluation of various parameters, including the base model, learning rate adjustments, minimum and maximum image sizes, anchor choices, aspect ratio configurations, angle parameter fine-tuning, Intersection over Union (IoU)

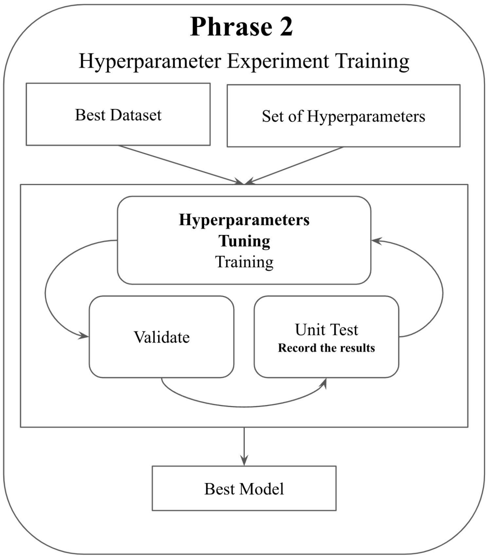


FIGURE 14. Diagram of the training process of phrase 2 which is a hyperparameter tuning experiment training to obtain the optimal hyperparameters for the model.

threshold adjustments, and the potential to freeze specific layers of the model's backbone. The values of each hyperparameter are listed in Table 6.

TABLE 6. Hyperparameters configuration.

No.	Config	Choice	Note
1	Base Model	R50-FPN-1X	Default
		R50-FPN-3X	-
		R101-FPN-3X	-
		X101-FPN-3X	-
2	Learning Rate	0.001	Default
		0.005	-
		0.01	-
		0.0125	-
		0.025	-
3	MinTrainSize	(800,)	Default
		(1200,)	-
4	MaxTrainSize	1333	Default
		1888	-
5	Anchor Size	[[8], [16], [32], [64], [128]]	-
		[[16], [32], [64], [128], [256]]	-
		[[32], [64], [128], [256], [512]],	Default
6	Aspect Ratio	[[0.5, 1.0, 2.0]]	Default
		[[0.5, 1.0, 2.0, 3.0]]	-
		[[0.25,0.5, 1.0, 2.0, 3.0]]	-
		[[0.25,0.5, 1.0, 2.0, 3.0, 4.0]]	-
		[[0.125,0.25,0.5,0.75, 1.0,1.25,1.5,1.75,2.0,2.5,3.0]]	-
7	Angles	[[-45, 0, 45]]	-
		[[-90, 0, 90]]	Default
8	IoU	[0.2,0.8]	-
		[0.3,0.7]	Default
9	Backbone	FREEZE_AT = 0	-
		FREEZE_AT = 2	Default

A. MODEL HYPERPARAMETERS

Along with the existing configurations in the notebook that detectron2 offered, we were able to edit additional configurations to fine-tune the model. For further details on the available configurations and options, please refer to the Detectron2 GitHub repository.[3] This allowed us to customize and optimize the model according to specific requirements. Some important configurations are as follows:

These hyperparameters are discussed in detail in Section III. This provides a better understanding of hyperparameter tuning.

- Base Model:
  There are four backbone choices for the model architecture: R50-FPN-1X [32], R50-FPN-3X [33], R101-FPN-3X (ResNet) [34], and X101-FPN-3X (ResNeXt) [35]. These architectures differ in depth, computational complexity, and the number of layers.
- General Model Hyperparameters:
  The general model hyperparameters are the learning rate and the number of training iterations. The learning rate determines the size of the optimization step. The number of training iterations specifies the number of times the model was trained. This hyperparameter was set to 200000 iterations for every experiment, and the model was saved every 1000 iterations.
- Advanced Model Hyperparameters:
  Advanced hyperparameters include the anchor size, aspect ratio, and angle. The anchor sizes are predefined bounding box shapes used for region proposal generation. We configured the anchor sizes and aspect ratios to match the objects and the text in the dataset. Hyperparameters such as anchor sizes, aspect ratios, and angles for text detection might be different from the settings for normal object detection, because texts are typically smaller than normal objects. Consequently, we decreased the values of these parameters to smaller than the default settings. For example, the default value of the anchor size is [[32], [64], [128], [256], [512]], and we adjust this value to be smaller [[16], [32], [64], [128], [256]] and hope that this new value will be able to detect the text object better. The aspect ratios help the anchor to have various shapes; for example, if the anchor size is 32 and the aspect ratio is [[0.5, 1.0, 2.0]], the anchor box will be [32 × 16], [32 × 32], and [32 × 64], respectively.
- Enable Model Components:
  In the backbone, we can include or exclude specific feature pyramid levels, adjust the number of heads in the model, and enable different feature-fusion strategies. These choices can influence the performance of the model and the computational requirements.

Following the training of the models with the default parameters for selecting the final dataset, Simple++, from Section V, we enhanced our model through hyperparameter tuning.

Hyperparameter tuning is a training loop in which we experiment with different settings to determine the best set for our detection model. The goal was to determine the

---

[3] https://github.com/facebookresearch/Detectron

optimal combination of hyperparameters for improving the performance of the model.

B. HYPERPARAMETERS TUNING RESULTS

1) BEST HYPERPARAMETERS FROM GRID SEARCH

After conducting a grid search to fine-tune the hyperparameters, we identified the optimal hyperparameters that yielded the best results when evaluated for both validation and test datasets. These hyperparameters are particularly relevant for our text detection task. For instance, when considering the anchor size in Table 7, the specified value proved to be appropriate and effective for detecting text within an image. The default value, [[32, 64, 128, 256, 512]], may be well suited for larger objects in images but may be less effective for the specific text we aim to detect. Another notable hyperparameter is the angle, where the chosen value is more suitable for document images containing text than the default value for real-world or scene images.

TABLE 7. Best hyperparameters from grid search.

No	Config	Choice	Note
1	Base Model	X101-FPN	-
2	Learning Rate	0.005	-
3	MinTrainSize	(1200,)	-
4	MaxTrainSize	1888	-
5	Anchor	[[16], [32], [64], [128], [256]]	-
6	Aspect Ratio	[[0.125,0.25,0.5,0.75,1.0, 1.25,1.5,1.75,2.0,2.5,3.0]]	-
7	Angles	[[-45, 0, 45]]	-
8	IoU	[0.3,0.7]	Default
9	Backbone	FREEZE_AT = 0	-

2) VALIDATION TEST WITH TEDEVAL

As shown in Table 8, the proposed model appears to be effective, particularly for documents such as Thai ID backs, fronts, passports, driver's licenses, and Indonesian ID cards. It achieved a good balance between recall and precision. However, for documents such as car registration books and bankbooks, there is room for improvement, particularly in terms of both recall and precision.

TABLE 8. Validation score from TedEval.

	Proposed Model		
Document	Recall	Precision	F1
Thai ID Front	99.22	99.25	99.23
Thai ID Back	100.00	100.00	100.00
Passport	96.94	96.81	96.87
Car Books	87.92	87.86	87.89
Bankbook	86.60	90.13	88.33
Driver License	95.15	95.23	95.19
Indo ID Card	97.72	97.61	97.67
Average	94.79	95.27	95.03

3) ACCURACY FROM UNITTEST WITH TEDEVAL

In Table 9, 10, and 11. We compared the performance of PMTD, SPCNET, CRAFT, YOLOv8, and the proposed model using TedEval. The proposed method tends to perform well in terms of precision, whereas the CRAFT may have a higher recall in some cases. The overall performance of the proposed method is comparable to or superior to that of CRAFT, depending on the specific document type. The

TABLE 9. Recall score of unit test score from TedEval.

	Recall				
Document	PMTD	SPCNET	CRAFT	YOLOv8	Proposed
Thai ID Front	94.60	96.19	97.00	85.40	97.21
Thai ID Back	73.12	71.65	75.36	66.77	72.68
Passport	84.32	88.49	94.71	64.47	88.85
Car Books	86.94	85.54	91.47	61.89	92.27
Invoice	85.49	83.63	96.58	43.89	84.88
Bankbook	82.40	80.19	86.07	63.60	85.30
Mobile Slip	83.86	94.47	86.74	54.89	96.72
Receipt	90.04	91.02	97.20	55.05	91.17
Government	90.83	92.87	97.24	63.22	93.11
AVG	85.73	87.12	91.37	62.13	89.13

TABLE 10. Precision score of unit test score from TedEval.

	Precision				
Document	PMTD	SPCNET	CRAFT	YOLOv8	Proposed
Thai ID Front	94.34	94.29	95.82	86.02	97.45
Thai ID Back	80.47	82.87	73.03	68.79	93.31
Passport	81.63	86.23	85.43	73.87	89.19
Car Books	89.54	83.87	87.74	62.32	92.86
Invoice	89.39	90.17	96.74	52.50	87.78
Bankbook	84.21	83.57	85.63	67.91	88.66
Mobile Slip	83.47	86.96	84.98	28.86	95.86
Receipt	88.11	86.18	90.11	56.46	92.36
Government	87.15	93.12	96.08	62.43	94.70
AVG	86.48	87.47	88.40	62.13	92.46

TABLE 11. F1 score of unit test score from TedEval.

	F1				
Document	PMTD	SPCNET	CRAFT	YOLOv8	Proposed
Thai ID Front	94.47	95.23	96.40	85.71	97.33
Thai ID Back	76.62	76.86	74.18	67.76	81.71
Passport	82.95	87.35	90.93	68.85	89.03
Car Books	88.22	84.70	89.57	62.11	92.56
Invoice	87.40	86.78	96.66	47.81	86.30
Bankbook	83.30	81.85	88.55	65.68	86.95
Mobile Slip	83.66	90.56	85.85	37.83	96.29
Receipt	89.06	88.53	93.52	55.75	91.76
Government	88.95	92.99	96.65	62.82	93.90
AVG	86.07	87.21	90.26	61.59	90.65

results show that, in most services, CRAFT and our proposed model have better accuracy than the other three models.

4) ERROR CASES FROM UNITTEST WITH TEDEVAL

Table 12 shows the error cases from the TedEval. The proposed method tends to have fewer multiline errors (as shown in Figure 17), split detection, one-to-many, and many-to-one (Figure 16), and duplicate detections (Figure 18) and miss the lower characters. However, CRAFT has fewer missing upper characters (as shown in Figure 19), missing characters, and missing characters (Figure 15). These results suggest that the proposed method may have advantages over the CRAFT in various cases. The proposed model, which treats individual objects independently, resulted in a significant decrease in multiline errors. Similarly, for other cases where the model outperforms CRAFT, its output provides suitable bounding boxes for use. However, in the case of missing upper characters, the results may indicate a higher error rate compared with CRAFT. This issue could be attributed to the model having encountered fewer instances of upper characters during the training. For further training, we may prioritize incorporating more examples of this character type to enhance the model's performance and achieve better results.

TABLE 12. Error cases from TedEval.

Cases	PMTD	SPCNET	CRAFT	YOLOv8	Proposed
Multiline	35	41	91	44	38
Splitted Detection	48	43	30	877	18
One To Many	532	491	433	4602	207
Many To One	1742	1530	1310	1048	1024
Duplicate Detection	203	179	118	448	128
Miss Upper Character	1282	1319	799	2183	1158
Miss Lower Character	120	148	51	298	90
Miss Character	1219	1083	917	6510	970
Miss Single Character	499	563	483	840	512

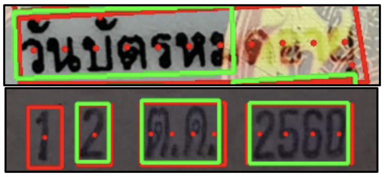


FIGURE 15. The model's predictions are represented by green boxes, while the actual ground-truth boxes are red boxes. The red dot within a box, approximates the character's position. In the above figure, it is a missing character case. There are some dots that do not intersect with the green boxes. The TedEval will count as a missing character. In the below figure, it is a missing single character case. In the first red box, there is no overlap with any green boxes, the TedEval will count as a missing single character and also count as missing character too.

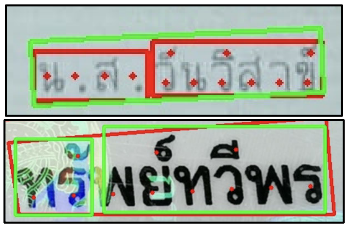


FIGURE 16. The above figure, there are two red boxes as ground-truth, but the model predicted these two boxes grouped together in one green box. In this case, TedEval will count as many-to-one case. The below figure, there is one red box as ground-truth, but the model predicted two boxes separately, TedEval will count as one-to-many case.

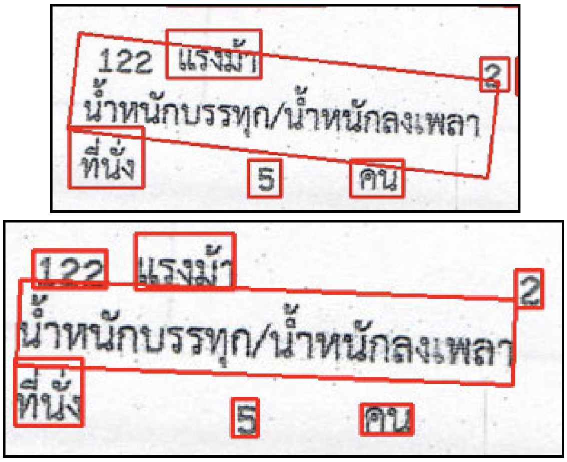


FIGURE 17. Example of the multiline case from CRAFT model and our proposed model. The upper image is the bounding box from CRAFT and the lower image is from our proposed model. In the image above, observe that in CRAFT, the red bounding boxes include a box that extends across distinct horizontal lines. This is called multiline. However, in the image below, our proposed model doesn't generate that kind of box.

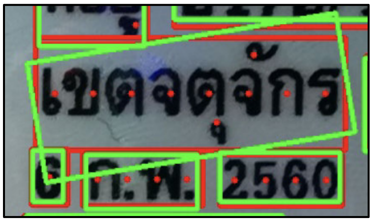


FIGURE 18. Example of the duplicate detection in our model that does not affect the result from TedEval. As you can see in the left below of the image, the number "6" is detected by 2 times which are the biggest box and its own box, TedEval will count as duplicate detection.

We experimented with YOLOv8 using the Simple++ dataset to assess its performance. It appears that it is not particularly effective for text detection, so we decided not to test it using the E2E pipeline.

C. ANALYSIS OF THE MODEL COMPLEXITY

In the model training, various aspects of the model must be addressed. For example, we should consider the model's architecture, the number of parameters, and how these factors impact the time it takes for the model to make inferences (more details are provided in Section VII).

In this section, we examine the architecture of our proposed model. We experimented with different backbone

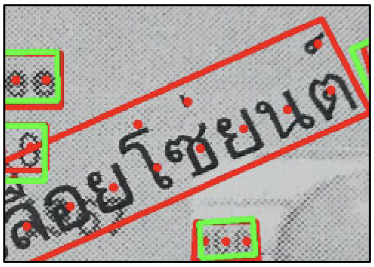


FIGURE 19. Example of the missing upper char from proposed model.

TABLE 13. Results evaluated in each backbone of model.

No.	Model	NumParams	Recall	Precision	F1
1	R50-FPN-1X	~44M	74.13	75.91	75.01
2	R50-FPN-3X	~44M	77.09	75.22	76.14
3	R101-FPN-3X	~63M	79.17	78.28	78.72
4	X101-FPN-3X	~107M	84.66	81.49	83.04

architectures (Table 13). Using the optimal dataset from Section V. Dataset Experiment and default parameters, we observed how each backbone model influenced accuracy in terms of Recall, Precision, and F1 score.

1) RECALL

recall values generally increase as the backbone model becomes more complex or deeper. X101-FPN-3X had the highest recall, indicating a better performance in capturing true positives.

2) PRECISION

The precision values show a slight variation across different backbone models.

3) F1 SCORE

X101-FPN-3X had the highest F1 score, suggesting a good balance between precision and recall.

4) MODEL COMPLEXITY

This trend indicates that as the backbone model becomes more complex (from ResNet-50 to ResNet-101 to ResNeXt -101), the performance generally improves.

In summary, the primary concern is the overall performance, with a good balance between precision and recall, and the X101-FPN-3X backbone is the most suitable choice. However, the backbone choice may depend on specific requirements or constraints in the application domain. In our task, selecting X101 as the backbone model yielded the best performance among available options. The grid search results further confirm that the X101-FPN is the chosen final backbone model when tuning the hyperparameters.

D. CONCLUSION OF HYPERPARAMETERS TUNING

In conclusion, based on the results in Tables 11 and 12, our performance matched the CRAFT baseline. However, it is important to note that our approach showed significant improvements in certain critical documents, such as enhancing the front side of the Thai ID card. Moreover, our use of the proposed model effectively overcomes the post-processing challenges, which is the focus of our research. This implies that we no longer need extensive post-processing each time a new service is introduced. The multiline detection problem was reduced, as presented in Table 12.

Once we obtained the best model from the training phase, we directly utilized it to detect bounding boxes in the images without any additional post-processing steps. The detected bounding boxes are then passed on to the next stages, which is text recognition and matching of the key and value in the information extraction part.

E. EXAMPLE OF BBOX PREDICTION FROM MODEL

See Figure 20 to 25.

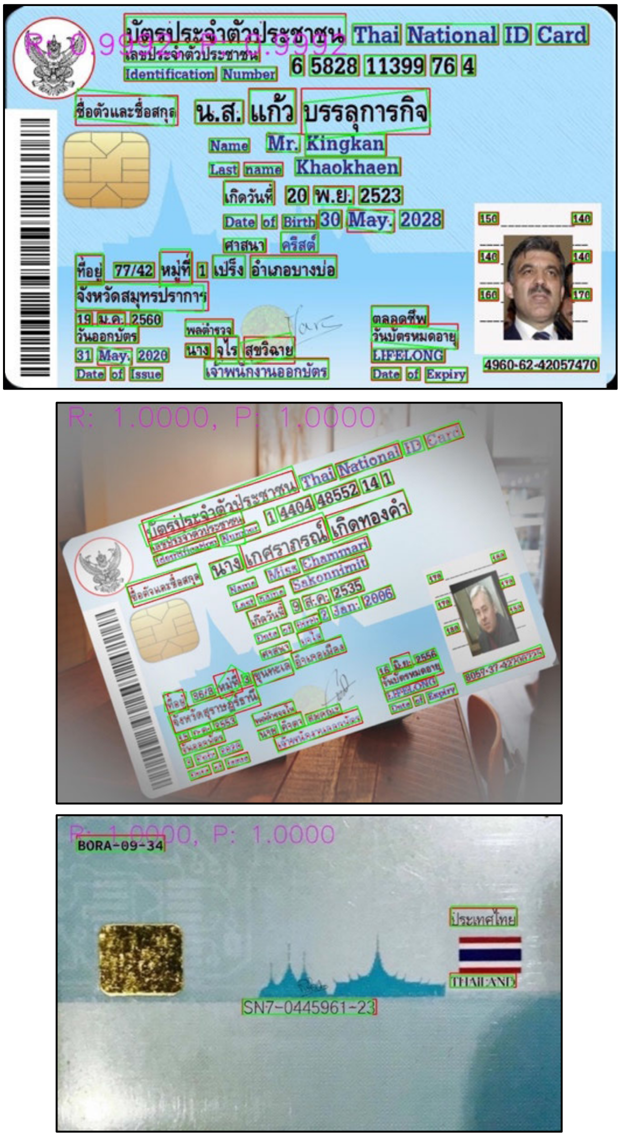


FIGURE 20. Example of the bounding box prediction from the proposed model of both front and back size of Thai ID Card with synthetic images. The green one represents the prediction bounding box and the red one represent the ground-truth bounding box.

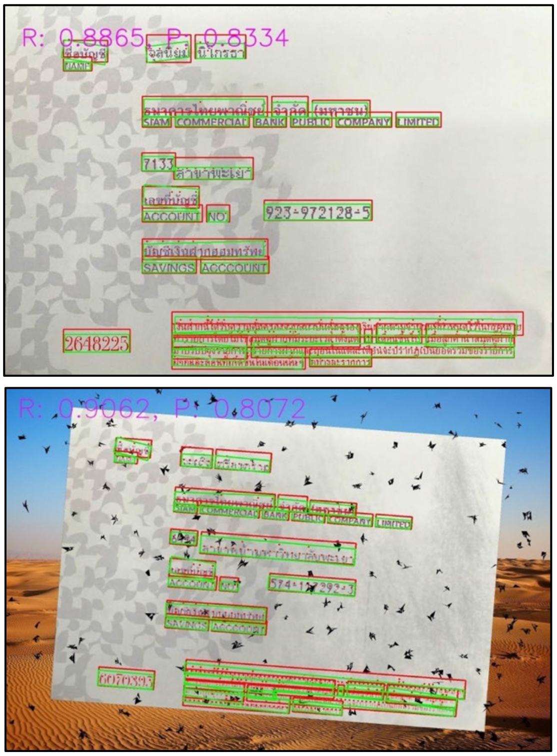


FIGURE 21. Example of the bounding box prediction from the proposed model of bank book with synthetic images. The green one represents the prediction bounding box and the red one represent the ground-truth bounding box.

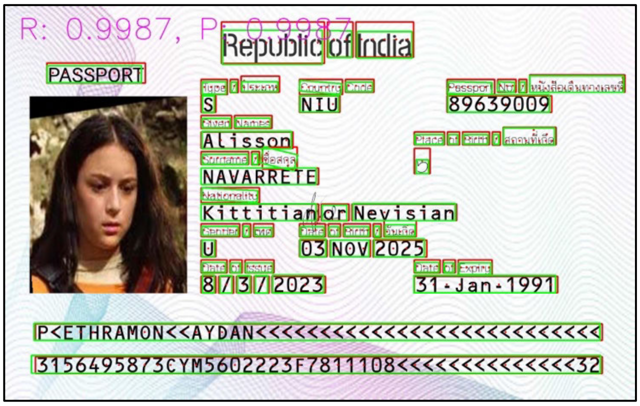


FIGURE 22. Example of the bounding box prediction from the proposed model of passport with synthetic images. The green one represents the prediction bounding box and the red one represent the ground-truth bounding box.

VII. PIPELINE EXPERIMENT

In this section, we describe the integration of the proposed model into the (OCR) pipeline. The OCR pipeline comprises three fundamental stages: text detection, text recognition, and information extraction. The proposed model is placed in the text detection part.

We performed end-to-end tests for all of our services and compared the performance of our proposed model with the baseline from the CRAFT. The overall training phase is illustrated in Figure 26.

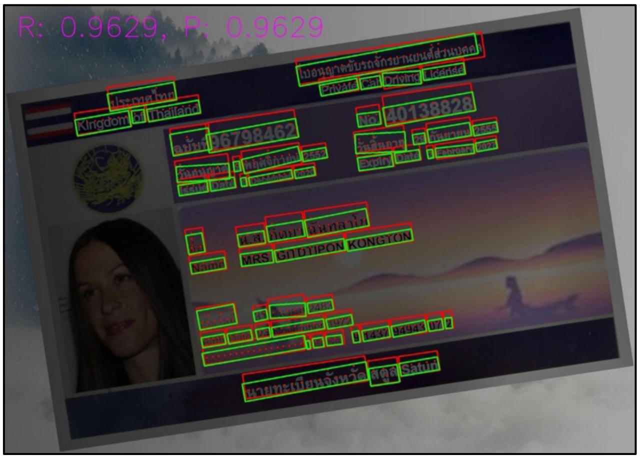


FIGURE 23. Example of the bounding box prediction from the proposed model of Thai driver license with synthetic images. The green one represents the prediction bounding box and the red one represent the ground-truth bounding box.

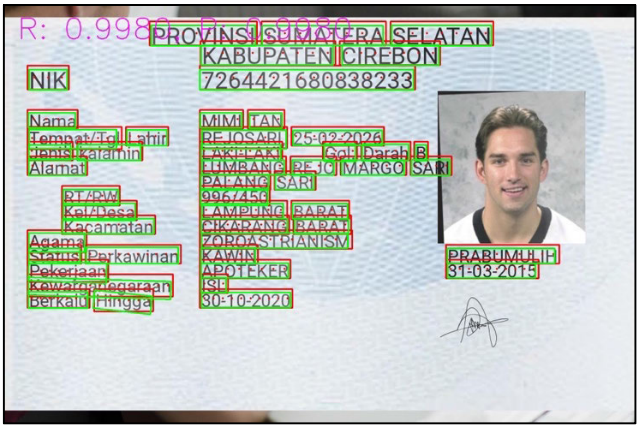


FIGURE 24. Example of the bounding box prediction from the proposed model of Indonesian ID Card with synthetic images. The green one represents the prediction bounding box and the red one represent the ground-truth bounding box.

A. TEST DATASET FOR END-TO-END TESTING

In order to compare the predicted results of the two different models, we need to use the same dataset that was used for the CRAFT model, and that dataset is this one.

This dataset consists of real images randomly selected from an end-to-end dataset.

This dataset serves as an evaluation benchmark for the OCR pipeline (Table 14). There are multiple datasets that are aligned with multiple pipelines. These datasets encompass real images and exhibit various characteristics such as noise, black-and-white, and distorted images.

B. END-TO-END RESULTS

1) END-TO-END RESULTS FROM ALL SERVICES

The end-to-end results (as shown in Table 15) demonstrate significant improvements in OCR and text detection for specific document types, such as the front side of the Thai ID

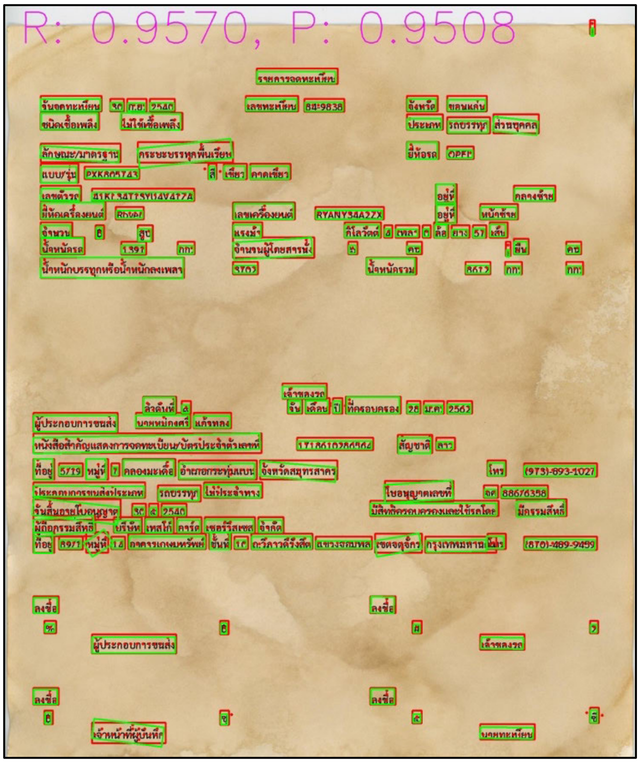


FIGURE 25. Example of the bounding box prediction from the propsed model of car registration with synthetic images. The green one represents the prediction bounding box and the red one represent the ground-truth bounding box.

TABLE 14. End-to-end dataset.

No.	Document	E2E
1	Thai ID Front	1262
2	Thai ID Back	2011
3	Car Books 1	732
4	Car Books 2	732
5	Car Books 3	732
6	Car Books 4	217
7	Car Books 5	217
8	Receipt 1	52
9	Receipt 2	376
10	Bankbook	1308
11	Mobile Slip	410
12	Passport	917
13	Indo ID Card	54
14	Driver License	160
	Total	9180

card, car registration, invoice, receipt, and Indonesian ID card (see the example results in Figures 20, 25, and 22.). The model exhibited superior performance for the accurate extraction of text from these documents.

TABLE 15. End-to-end results.

No.	Documents	PMTD	SPCNET	CRAFT	Proposed
1	Thai ID Front	93.16	94.88	96.70	96.97
2	Thai ID Back	89.19	91.67	95.93	91.27
3	Indo ID Card	60.58	58.14	55.82	70.33
4	Car Books 1	84.54	81.11	83.03	83.62
5	Car Books 2	81.74	79.49	84.19	82.89
6	Car Books 3	71.83	71.98	74.29	74.19
7	Car Books 4	72.18	73.25	74.53	74.60
8	Car Books 5	65.38	64.11	69.86	70.07
7	Passport	92.83	94.18	95.66	95.17
10	Driver License	90.22	88.43	91.25	88.50
11	Bankbook	80.38	79.54	84.82	78.79
12	Mobile Slip	80.11	82.58	85.84	85.14
13	Receipt 1	80.53	78.84	82.77	79.61
14	Receipt 2	58.23	55.60	62.12	64.81
	AVG	78.64	78.12	81.20	81.14

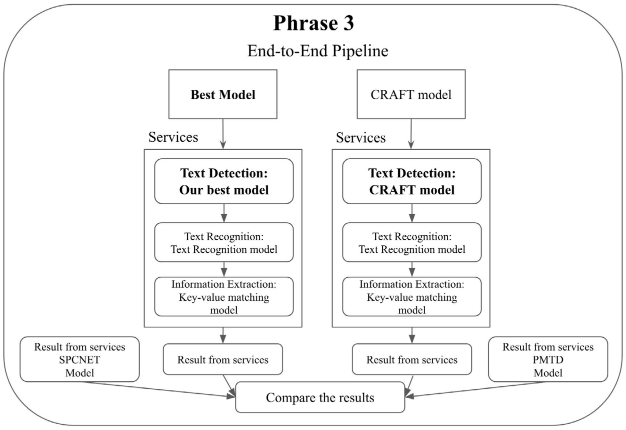

Phrase 3
End-to-End Pipeline
Best Model
CRAFT model
Services
Services
Text Detection: Our best model
Text Detection: CRAFT model
Text Recognition: Text Recognition model
Text Recognition: Text Recognition model
Information Extraction: Key-value matching model
Information Extraction: Key-value matching model
Result from services SPCNET Model
Result from services
Result from services
Result from services PMTD Model
Compare the results

FIGURE 26. Diagram of the pipeline process of phrase 3 which is an End-to-End pipeline. To test our proposed model with the real world cases compare to the CRAFT model.

The proposed model performed well for the important services. Achieving optimal accuracy for other document types remains a challenge. The model's performance was relatively lower for less important documents, such as passports, bank books, and driver's licenses (see examples in Figures 22, 21, and 23) because of the differences in document templates, data availability, data diversity, and complexities specific to each document type.

The average overall accuracy of the proposed model matched that of the CRAFT model. However, the key advantage of the proposed model is its ability to decrease both inference time and resource requirements for deploying the model. This reduces the costs associated with the training process and the maintenance of the model in the cloud.

2) PIPELINE TIME INFERENCE, RESOURCE CONSUMING AND MODEL ARCHITECTURE

The average inference time for each model is less than 1 s, which is quite for the OCR pipeline because we have

TABLE 16. Comparing the results, resources and model architecture.

No.	Documents	PMTD	SPCNET	CRAFT	Proposed
1	Accuracy AVG	81.59	81.50	81.20	81.14
2	Total Time Inference AVG (s)	0.61	0.56	0.84	0.52
3	Detection Time Inference AVG (s)	0.53	0.49	0.64	0.52
4	Post-process Time Inference AVG (s)	0.08	0.07	0.20	0.00
5	CPU-Core	4	4	4	3
6	CPU-RAM	7	7	8	6
7	GPU	5	5	10	6
8	GPU Node (~24Gb)	1	1	1	1
9	Model Architecture	R50-FPN	R50-FPN	VGG-16	ResNeXt-101
10	Parameters	~44M	~44M	~138M	~107M

two more parts to perform (text recognition and information extraction) before sending the results to the users. It is desirable for this time to be minimized as much as possible. Consider a scenario in which numerous users simultaneously request OCR services, which usually occurs. In such a case, the processing time of the pipeline may increase. This is particularly evident in scenarios, such as processing a car registration book with numerous text fields for detection. Our proposed model outperformed the other models in handling such tasks.

Utilizing Mask R-CNN, built on Detectron2, resulted in faster inference times than the CRAFT model. We can reduce resources when deploying our proposed model. Additionally, eliminating post-processing reduces the time gaps in the process by approximately 0.20 s.

3) EXAMPLE OF FIELDS FROM PASSPORT

From the passport result in Table 17, we exclusively extracted information from the MRZ lines located at the bottom of the passport (see the sample in Figure 22). Although the overall accuracy aligns closely with CRAFT's, we will continue improving this service.

TABLE 17. End-to-end results.

No.	Fields	CRAFT	Proposed
1	mrz1	96.14	96.62
2	mrz2	95.17	93.72
3	AVG	95.66	95.17

4) EXAMPLE OF FIELDS FROM FRONT SIDE OF THAI ID CARD

From the graph in Figure 27, this service reads Thai ID card documents and is vital for handling the highest number of daily transactions. Maintaining and enhancing this service is crucial, even if it results in a mere 0.50% increase in accuracy. These improvements extend beyond accuracy to include reduced inference time and decreased resource utilization.

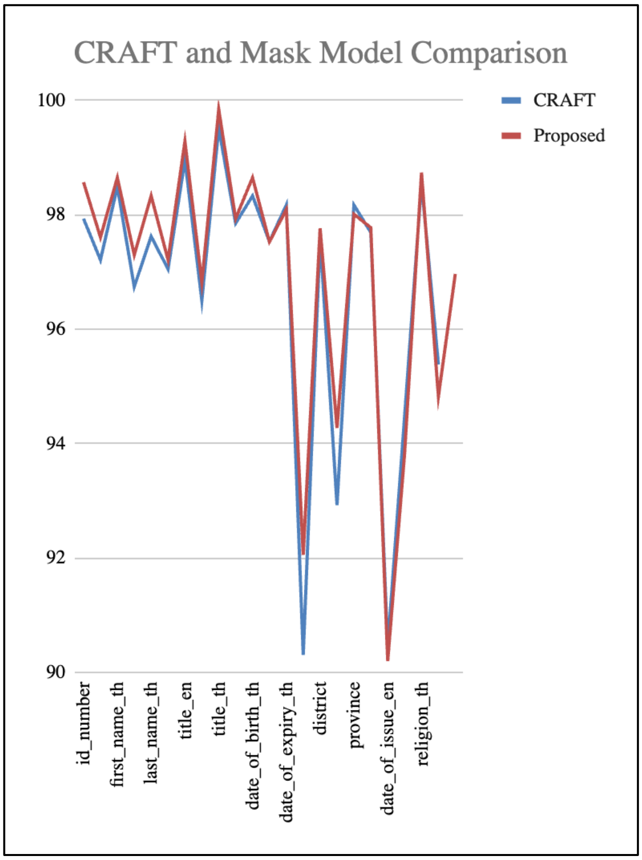


FIGURE 27. Show the E2E results from Thai ID Card pipeline comparing between CRAFT and proposed model. The fields are ordered by the most important one.

5) EXAMPLE OF FIELDS FROM CAR REGISTRATION

The graph in Figure 28 shows that the OCR pipeline service focuses on car registration. Although significant improvements have been made in the most important fields, some fields of the proposed model still need to catch up to CRAFT. Efforts are ongoing to enhance these specific fields further to improve overall performance.

C. ANALYSIS OF THE RESULT PREDICTION

For result prediction, we conducted an E2E test for real-world case testing. Once we obtain results from the E2E pipeline experiment, it is beneficial to analyze individual images. We went through images in which the predictions differed from the ground truth. We evaluate the key-value pairs and inspect the bounding boxes defined by the proposed model. This detailed analysis enabled us to understand why certain predictions were incorrect.

The images were examined by grouping the key values of the results. For example, when reviewing the front side of a Thai ID card, we inspect keys such as id_number, full_name_th, full_name_en, or date_of_birth, the additional keys are listed in Table 18. We analyzed the bounding boxes from our proposed model and identified the differences from the actual ground truth. Most incorrect predictions are from

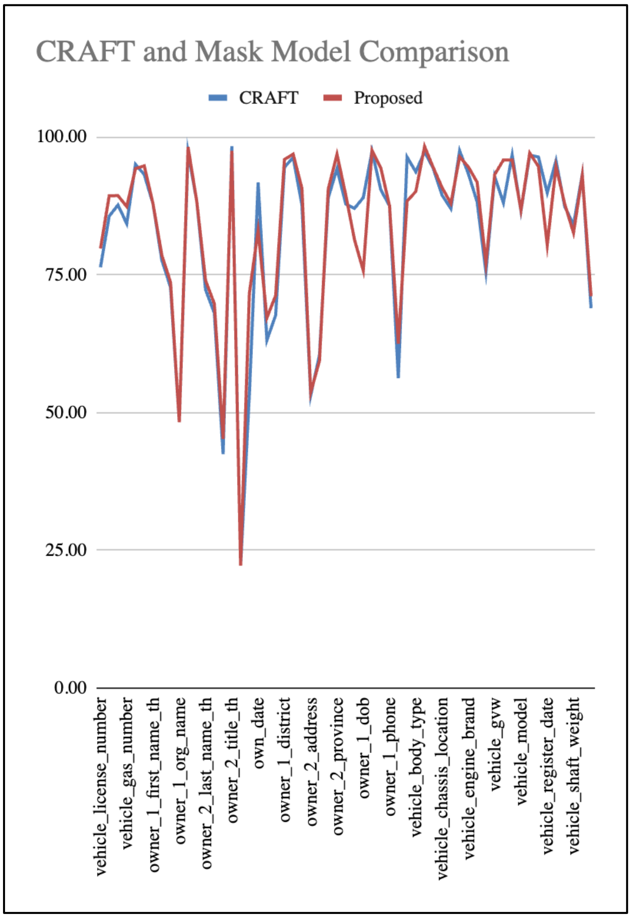


FIGURE 28. Show the E2E results from car registration pipeline comparing between CRAFT and proposed model. The fields are ordered by the most important one.

bounding boxes that are too tight around the text; the bounding boxes cannot cover the entire text, potentially missing a single character, or having correct bounding boxes that, when passed to the next module (text recognition), cannot be read because of issues with the text recognition itself. An example of the prediction analysis is shown in Table 18.

After analyzing the images, we can identify and address common issues in each key and find ways to enhance the model in the future. We not only analyzed the results for the front side of the Thai ID card but also for other services. This process is time consuming because of the multiple number of services and images available for review. This creates an overload of information.To make it more manageable, we chose to showcasethe results for only one serviceas a concrete example.

D. ANALYSIS OF INFERENCING IN MODEL PIPELINE

We conducted experiments in a consistent environment for all parts of the inference step during testing. For example, in the E2E part, we conducted a testing-in-development (DEV) environment to compare and analyze the results.

TABLE 18. Analysis of the result prediction.

Fields	Analysis
address_no	Most case of the incorrect prediction is missing single character in address number.
address_th	This field is composed of other address components such as address_no or districts. Incorrect prediction are likely to occur alongside these fields since they are part of each other.
date_of_expiry_en	The presence of light reflex, and hologram elements makes it more challenging to read this field (Recognition problem).
date_of_expiry_th	
date_of_issue_en	
date_of_issue_th	
district	The majority of cases occur because the information extraction process fails to locate this field.
province	
sub_district	
first_name_en	Some words are split, and there is a hologram reflection positioned within this field.
middle_last_name_en	
title_first_name_en	
first_name_th	Some lower vowels are missing due to the bounding box inadequately capturing it, occurring in only 2-3 cases.
full_name_th	
last_name_th	
id_number	In most cases, the last digit is missing.
religion_th	Some lower vowels are missing due to the bounding box inadequately capturing it. Especially in the case of word “พุทธ”
serial_number	Recognition service problem, due to overly tight bounding boxes in the detection process.
date_of_birth_en	-
date_of_birth_th	-
title_en	-
title_th	-

We explored why the proposed model performed faster when inferring bounding boxes. We examined different aspects of the model, such as the parameters of the model architecture and post-processing. We excluded factors such as the environment and hardware acceleration because we conducted the test in the same setting.

Regarding the parameters, we referred to each model's architecture and found that the CRAFT model had more parameters than our proposed model. In general, models with more parameters often require more computational time for inferences. For post-processing, unlike the CRAFT model, our model does not require an additional step to extract the bounding boxes. This difference contributes to an increase in the detection process time in the CRAFT. For PMTD and SPCNET, although the detection inference time is slightly less than that of our proposed model (approximately 0.1 s, a post-processing step is required before finalizing the bounding boxes. Consequently, the total inference time remains higher than the proposed model's.

E. ANALYSIS OF THE MODEL RESOURCE CONSUMPTION

Variations in the model architecture necessitate distinct resource allocations. In the case of neural network models, both CPU and GPU are essential for constructing the model.

CPUs are more versatile and better suited for general-purpose computing. Some parts of the inference process, such as data preprocessing or post-processing tasks, may be handled more efficiently by the CPU. Because the CRAFT model requires an additional post-processing step, unlike our proposed model, an extra 2 gigabytes was incorporated to facilitate its execution.

The GPU has a greater ability to perform parallel processing. Both CRAFT and Mask R-CNN involve a large number of matrix operations that can be parallelized. GPUs are well suited for handling these parallel computations and can significantly speed up the inference process compared to only a CPU. From Table 16, our proposed model is smaller than the CRAFT model, possibly in terms of the number of parameters. The CRAFT requires more computational work, resulting in a higher GPU consumption of approximately 4 gigabytes more.

F. CONCLUSION OF END-TO-END PIPELINE

In conclusion, we selected the best model from the previous section (Section VI.) to conduct end-to-end tests on the OCR pipeline for each service. We combined our proposed model, which is a text detection model, with other services, text recognition, and information extraction.

1) PIPELINE

The average accuracy of the proposed model surpassed that of the CRAFT model for the most important services. It is important to note that some less essential services, such as driver's licenses and bank books, may not be currently utilized by users. We can set aside these services for further improvement. In addition, we focused on other services that are currently used services.

2) POST-PROCESS

We successfully resolved the post-processing issue by eliminating the need to configure each service individually to obtain the bounding box information. Our model's default process allows for the easy retrieval of bounding boxes.

3) MULTILINE

We also addressed the problem of multiline or closely positioned bounding boxes by treating objects individually, ensuring that such situations occur less frequently with our proposed model.

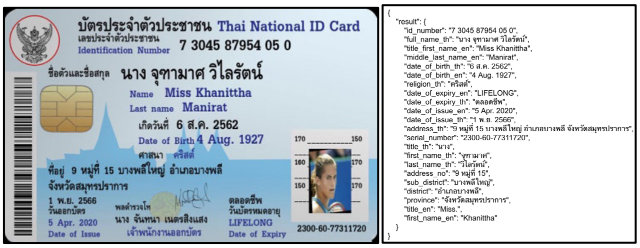
4) TIME INFERENCE AND RESOURCES

By employing our proposed model, we can reduce the inference time while simultaneously reducing the utilization of the CPU, GPU, and RAM resources. This leads to a decrease in the overall cost of maintaining a model in the cloud.

G. EXAMPLE OF RESPONSES FROM OCR PIPELINE WITH OUR PROPOSED MODEL

1) RESPONSE FROM THAI ID CARD FRONT SIDE

See Figure 29.

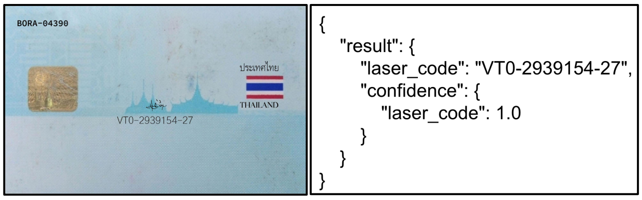


```
{
  "result": {
    "id_number": "7 3045 87954 05 0",
    "full_name_th": "นาง จุฑามาศ วิไลรัตน์",
    "title_first_name_en": "Miss Khanittha",
    "middle_last_name_en": "Manirat",
    "date_of_birth_th": "6 ส.ค. 2562",
    "date_of_birth_en": "4 Aug. 1927",
    "religion_th": "คริสต์",
    "date_of_expiry_en": "LIFELONG",
    "date_of_expiry_th": "ตลอดชีพ",
    "date_of_issue_en": "5 Apr. 2020",
    "date_of_issue_th": "1 พ.ย. 2566",
    "address_th": "9 หมู่ที่ 15 บางพลีใหญ่ อำเภอบางพลี จังหวัดสมุทรปราการ",
    "serial_number": "2300-60-77311720",
    "title_th": "นาง",
    "first_name_th": "จุฑามาศ",
    "last_name_th": "วิไลรัตน์",
    "address_no": "9 หมู่ที่ 15",
    "sub_district": "บางพลีใหญ่",
    "district": "อำเภอบางพลี",
    "province": "จังหวัดสมุทรปราการ",
    "title_en": "Miss.",
    "first_name_en": "Khanittha"
  }
}
```

FIGURE 29. **Response from Thai ID card front side pipeline. The left is the image, and the right is the response of key and value in JSON format.**

2) RESPONSE FROM THAI ID CARD BACK SIDE

See Figure 30.

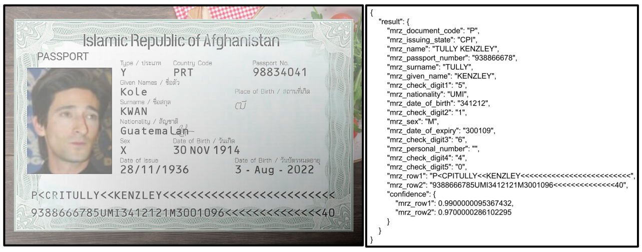


```
{
  "result": {
    "laser_code": "VT0-2939154-27",
    "confidence": {
      "laser_code": 1.0
    }
  }
}
```

FIGURE 30. **Response from Thai ID card back side pipeline. The left is the image, and the right is the response of key and value in JSON format.**

3) RESPONSE FROM PASSPORT

See Figure 31.

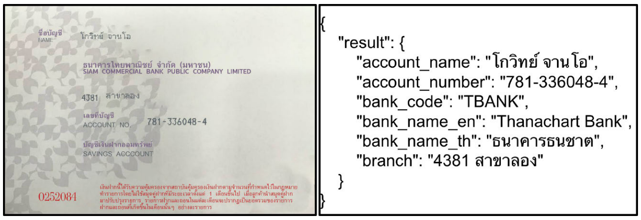


```
{
  "result": {
    "mrz_document_code": "P",
    "mrz_issuing_state": "CPI",
    "mrz_name": "TULLY KENZLEY",
    "mrz_passport_number": "938866678",
    "mrz_surname": "TULLY",
    "mrz_given_name": "KENZLEY",
    "mrz_check_digit1": "5",
    "mrz_nationality": "UMI",
    "mrz_date_of_birth": "341212",
    "mrz_check_digit2": "1",
    "mrz_sex": "M",
    "mrz_date_of_expiry": "300109",
    "mrz_check_digit3": "6",
    "mrz_personal_number": "",
    "mrz_check_digit4": "4",
    "mrz_check_digit5": "0",
    "mrz_row1": "P<CPITULLY<<KENZLEY<<<<<<<<<<<<<<<<<<<<<<<<<",
    "mrz_row2": "9388666785UMI3412121M3001096<<<<<<<<<<<<<<<40",
    "confidence": {
      "mrz_row1": 0.9900000095367432,
      "mrz_row2": 0.9700000286102295
    }
  }
}
```

FIGURE 31. **Response from passport pipeline. The left is the image, and the right is the response of key and value in JSON format.**

4) RESPONSE FROM BOOK BANK

See Figure 32.

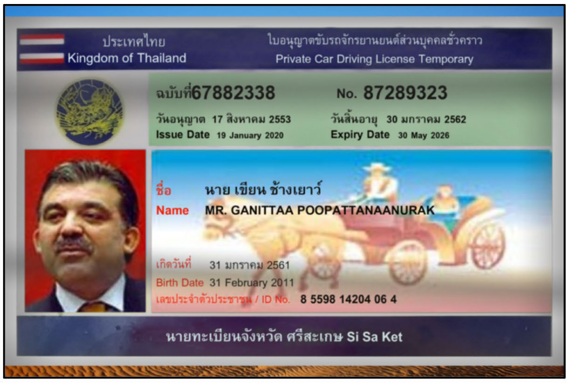


```
{
  "result": {
    "account_name": "โกวิทย์ จานโอ",
    "account_number": "781-336048-4",
    "bank_code": "TBANK",
    "bank_name_en": "Thanachart Bank",
    "bank_name_th": "ธนาคารธนชาต",
    "branch": "4381 สาขาลอง"
  }
}
```

FIGURE 32. **Response from book bank pipeline. The left is the image, and the right is the response of key and value in JSON format.**

5) RESPONSE FROM DRIVER LICENSE

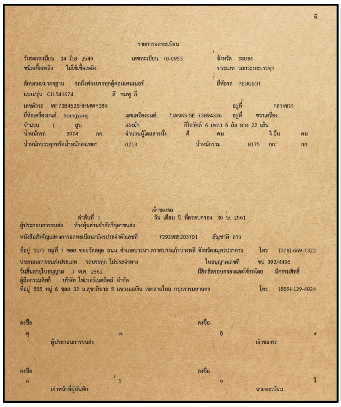
See Figure 33.



ประเทศไทย
Kingdom of Thailand
ใบอนุญาตขับรถจักรยานยนต์ส่วนบุคคลชั่วคราว
Private Car Driving License Temporary
ฉบับที่67882338 No. 87289323
วันอนุญาต 17 สิงหาคม 2553
Issue Date 19 January 2020
วันสิ้นอายุ 30 มกราคม 2562
Expiry Date 30 May 2026
ชื่อ นาย เขียน ช้างเยาว์
Name MR. GANITTAA POOPATTANAANURAK
เกิดวันที่ 31 มกราคม 2561
Birth Date 31 February 2011
เลขประจำตัวประชาชน / ID No. 8 5598 14204 06 4
นายทะเบียนจังหวัด ศรีสะเกษ Si Sa Ket

```
{
    "result": {
        "address_no": "",
        "branch_en": "Bangkok",
        "branch_th": "กรุงเทพมหานคร",
        "date_of_birth_en": "31 February 2011",
        "date_of_birth_th": "31 มกราคม 2561",
        "date_of_expiry_en": "30 May 2026",
        "date_of_expiry_th": "30 มกราคม 2562",
        "date_of_issue_en": "17 January 2020",
        "date_of_issue_th": "17 สิงหาคม 2553",
        "district": "",
        "first_name_en": "GANITTAA",
        "first_name_th": "เขียน",
        "full_name_en": "MR. GANITTAA POOPATTANAANURAK",
        "full_name_th": "นาย เขียน ช้างเยาว์",
        "id_number": "8 5598 14204 06 4",
        "last_name_en": "POOPATTANAANURAK",
        "last_name_th": "ช้างเยาว์",
        "middle_name_en": "",
        "middle_name_th": "",
        "number": "",
        "number_en": "87289323",
        "number_th": "67882338",
        "province": "",
        "sub_district": "",
        "title_en": "MR.",
        "title_th": "นาย",
        "type_en": "Private Motorcycle Driving Licence (Temporary)",
        "type_th": "ใบอนุญาตขับรถจักรยานยนต์ส่วนบุคคลชั่วคราว"
    }
}
```

FIGURE 33. Response from driver license pipeline. The left is the image, and the right is the response of key and value in JSON format.



```
{
    "result": {
        "instalment_contract": "",
        "instalment_contract_date": "",
        "own_date": "30 2561",
        "owner_1_address": "15/3 หมู่ 7 ซอย ซอยวัดสลุด ถนน อำเภอบางนา-ตราดบางแก้วบางพลี จ.สมุทรปราการ",
        "owner_1_address_no": "15/3 หมู่ 7 ซอย ซอยวัดสลุด ถนน อำเภอบางนา-ตราดบางแก้วบางพลี",
        "owner_1_district": "",
        "owner_1_dob": "",
        "owner_1_first_name_th": "",
        "owner_1_last_name_th": "",
        "owner_1_name": "",
        "owner_1_nationality": "ลาว",
        "owner_1_org_name": "ห้างหุ้นส่วนจำกัดวิชุดาขนส่ง",
        "owner_1_phone": "378-669-7322",
        "owner_1_province": "จ.สมุทรปราการ",
        "owner_1_sub_district": "",
        "owner_1_thai_id": "7292985203701",
        "owner_1_title_th": "",
        "owner_2_address": "555 หมู่ 6 ซอย 32 ถ.สุขาภิบาล 5 แขวงออเงิน เขตสายไหม กรุงเทพมหานคร",
        "owner_2_address_no": "555 หมู่ 6 ซอย 32 ถ.สุขาภิบาล 5",
        "owner_2_district": "เขตสายไหม",
        "owner_2_dob": "",
        "owner_2_first_name_th": "",
        "owner_2_last_name_th": "",
        "owner_2_name": "",
        "owner_2_nationality": "",
        "owner_2_org_name": "รถบรรทุก ไม่ประจำทาง",
        "owner_2_phone": "",
        "owner_2_province": "กรุงเทพมหานคร",
        "owner_2_sub_district": "แขวงออเงิน",
        "owner_2_thai_id": "บริษัทไซเบอร์เนตติคส์จำกัด",
        "owner_2_title_th": "",
        "owner_no": "",
        "vehicle_body_type": "ประเภท รถกระบะบรรทุก รถกึ่งพ่วงบรรทุกตู้คอนเทนเนอร์ ชมพู ภ",
        "vehicle_brand": "PEUGEOTCJL541674",
        "vehicle_car_weight": "",
        "vehicle_chassis_location": "ขวาเครื่อง",
        "vehicle_chassis_number": "WF73B452SHHMWY38K",
        "vehicle_color": "",
        "vehicle_displacement": "",
        "vehicle_engine_brand": "SSANGYONG",
        "vehicle_engine_location": "",
        "vehicle_engine_number": "",
        "vehicle_fuel_type": "",
        "vehicle_gas_number": "",
        "vehicle_gvw": "",
        "vehicle_hp": "",
        "vehicle_license_number": "",
        "vehicle_license_province": "ระยอง",
        "vehicle_model": "",
        "vehicle_model_year": "",
        "vehicle_piston": "9974",
        "vehicle_register_date": "14 มิ.ย. 2549",
        "vehicle_seat_number": "",
        "vehicle_shaft": "- เส้น",
        "vehicle_shaft_weight": "8175",
        "vehicle_short_type": "",
        "vehicle_type": "ไม่ใช้เชื้อเพลิง"
    }
}
```

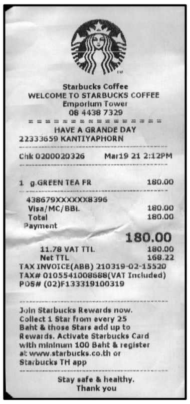
FIGURE 34. Response from car registration pipeline. The left is the image, and the right is the response of key and value in JSON format.

6) RESPONSE FROM CAR REGISTRATION

See Figure 34.

7) RESPONSE FROM RECEIPT

See Figure 35.



Starbucks Coffee
WELCOME TO STARBUCKS COFFEE
Emporium Tower
08 4438 7329
================
HAVE A GRANDE DAY
22333659 KANTIYAPHORN

Chk 0200020326 Mar19 21 2:12PM

1	g.GREEN TEA FR	180.00
	438679XXXXXX8396	
	Visa/MC/BBL	180.00
	Total	180.00
	Payment	
		180.00
	11.78 VAT TTL	180.00
	Net TTL	168.22

TAX INVOICE(ABB) 210319-02-15520
TAX# 0105541008688(VAT Included)
POS# (02)F133319100319

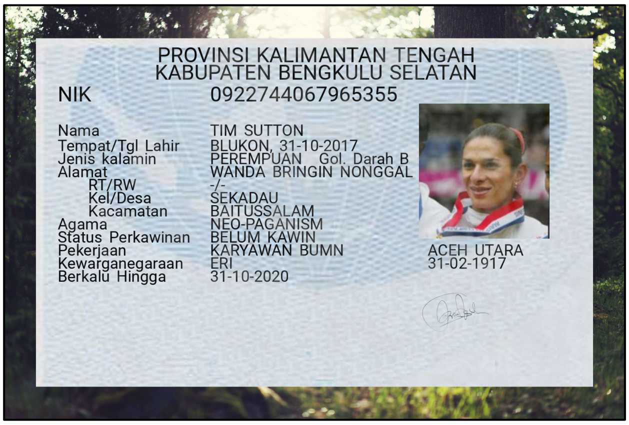
Join Starbucks Rewards now.
Collect 1 Star from every 25
Baht & those Stars add up to
Rewards. Activate Starbucks Card
with minimum 100 Baht & register
at www.starbucks.co.th or
Starbucks TH app

Stay safe & healthy.
Thank you

```
{
    "result": {
        "address": "",
        "branch": "Emporium Tower",
        "company_name": "",
        "date": "Mar19 21",
        "net_price": "180.00",
        "store_name": "Starbucks Coffee",
        "tax_id": "0105541008688",
    }
}
```

```
"lineitem": [
        {
            "product_code": {
                "text": "",
                "confidence": 0.0
            },
            "product_description": {
                "text": "g.GREEN TEA FR",
                "confidence": 1.0
            },
            "quantity": {
                "text": "1",
                "confidence": 1.0
            },
            "unit_price": {
                "text": "",
                "confidence": 0.0
            },
            "sub_total_price": {
                "text": "180.00",
                "confidence": 1.0
            },
            "discount": {
                "text": "",
                "confidence": 0.0
            },
            "subitem": []
        }
    ]
```

FIGURE 35. Response from receipt pipeline. The left is the image, the middle is the response from the head of the receipt. The right is the information in the table of key and value in JSON format.



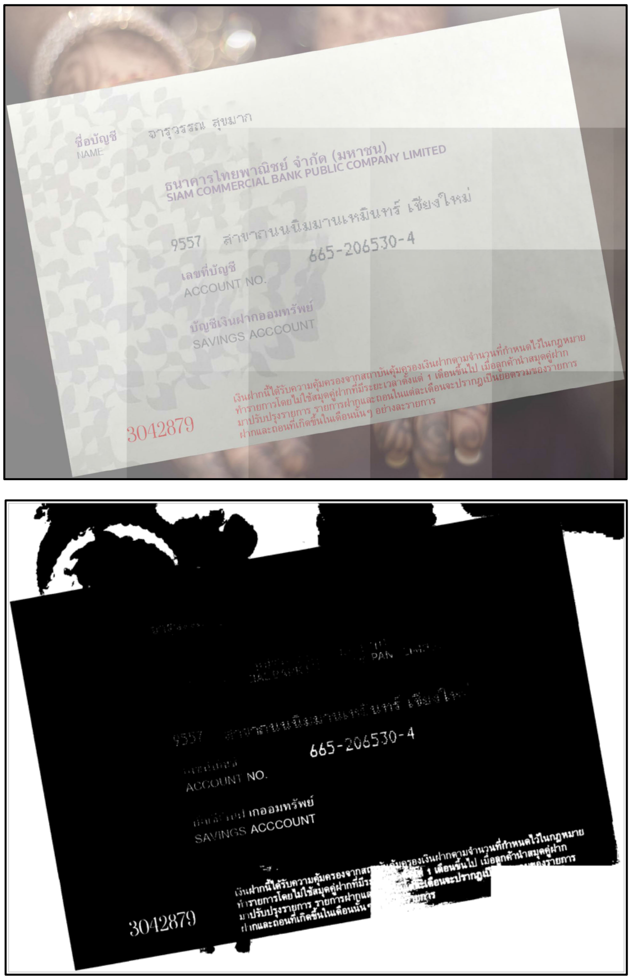
PROVINSI KALIMANTAN TENGAH
KABUPATEN BENGKULU SELATAN

NIK	0922744067965355
Nama	TIM SUTTON
Tempat/Tgl Lahir	BLUKON, 31-10-2017
Jenis kalamin	PEREMPUAN Gol. Darah B
Alamat	WANDA BRINGIN NONGGAL
RT/RW	-/-
Kel/Desa	SEKADAU
Kacamatan	BAITUSSALAM
Agama	NEO-PAGANISM
Status Perkawinan	BELUM KAWIN
Pekerjaan	KARYAWAN BUMN
Kewarganegaraan	ERI
Berkalu Hingga	31-10-2020

ACEH UTARA
31-02-1917

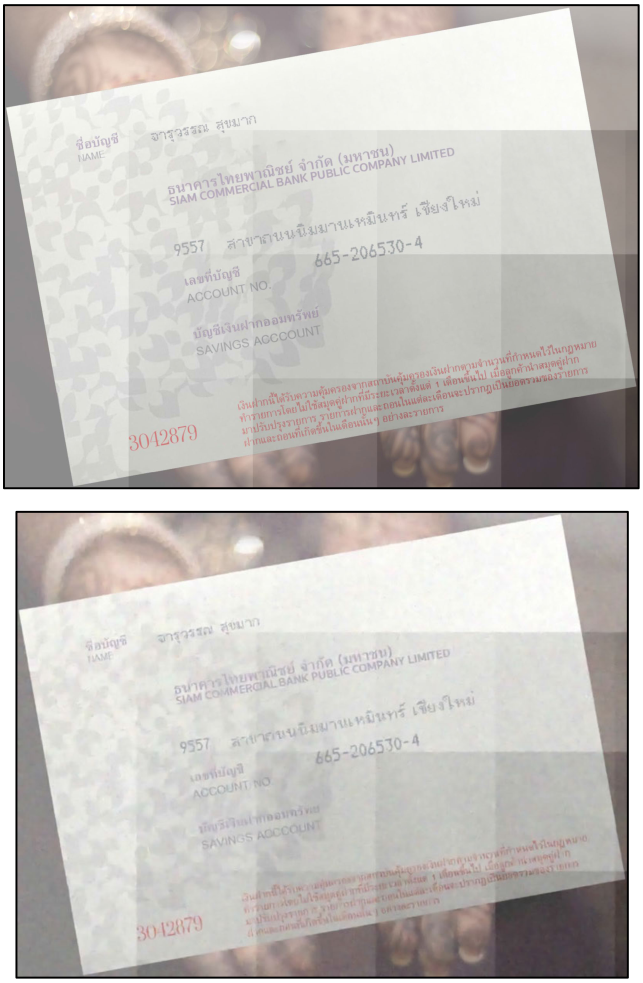
```
{
    "result": {
        "province": "PROVINSI KALIMANTAN-TENGAH",
        "regency_city": "KABUPATEN BENGKULU-SELATAN",
        "id_number": "0922744067965355",
        "name": "TIM SUTTON",
        "place_date_of_birth": "BLUKON, 31-10-2017",
        "gender": "PEREMPUAN",
        "blood_type": "B",
        "address": "",
        "rtrw": "-/-",
        "district": "SEKADAU",
        "sub_district": "BAITUSSALAM",
        "religion": "NEO_PAGANISM",
        "marital_status": "BELUM KAWIN",
        "occupation": "KARYAWAN BUMN",
        "nationality": "ERI",
        "date_of_expiry": "31-10-2020",
        "place_of_issue": "ACEH UTARA",
        "date_of_expiry": "31-02-1917",
    }
}
```

FIGURE 36. Response from Indonesian ID card pipeline. The left is the image, and the right is the response of key and value in JSON format.



ชื่อบัญชี
NAME
จารุวรรณ สุขมาก
ธนาคารไทยพาณิชย์ จำกัด (มหาชน)
SIAM COMMERCIAL BANK PUBLIC COMPANY LIMITED
9557 สาขาถนนนิมมานเหมินทร์ เชียงใหม่
665-206530-4
เลขที่บัญชี
ACCOUNT NO.
บัญชีเงินฝากออมทรัพย์
SAVINGS ACCCOUNT
3042879

FIGURE 37. **The upper image represents the original image, while the lower image is the binary image that pass through the preprocess function.**



ชื่อบัญชี
NAME
จารุวรรณ สุขมาก
ธนาคารไทยพาณิชย์ จำกัด (มหาชน)
SIAM COMMERCIAL BANK PUBLIC COMPANY LIMITED
9557 สาขาถนนนิมมานเหมินทร์ เชียงใหม่
665-206530-4
เลขที่บัญชี
ACCOUNT NO.
บัญชีเงินฝากออมทรัพย์
SAVINGS ACCCOUNT
3042879

FIGURE 38. **The upper image represents the original image, while the lower image pass thought the DnCNN deep learning model to eliminate noise in the image.**

8) RESPONSE FROM INDONISIAN ID CARD

See Figure 36.

VIII. ABLATION EXPERIMENTS

Our focus is on enhancing the OCR accuracy, not only through model training but also by refining the image preprocessing before feeding it into the OCR steps. We hypothesized that effective preprocessing could enhance the performance of both text detection and text recognition models. These preprocessing steps may involve denoising, contrast enhancement, binarization, or angle correction. However, it has been observed that preprocessing can lead to information loss in the image (as shown in Figure 37.).

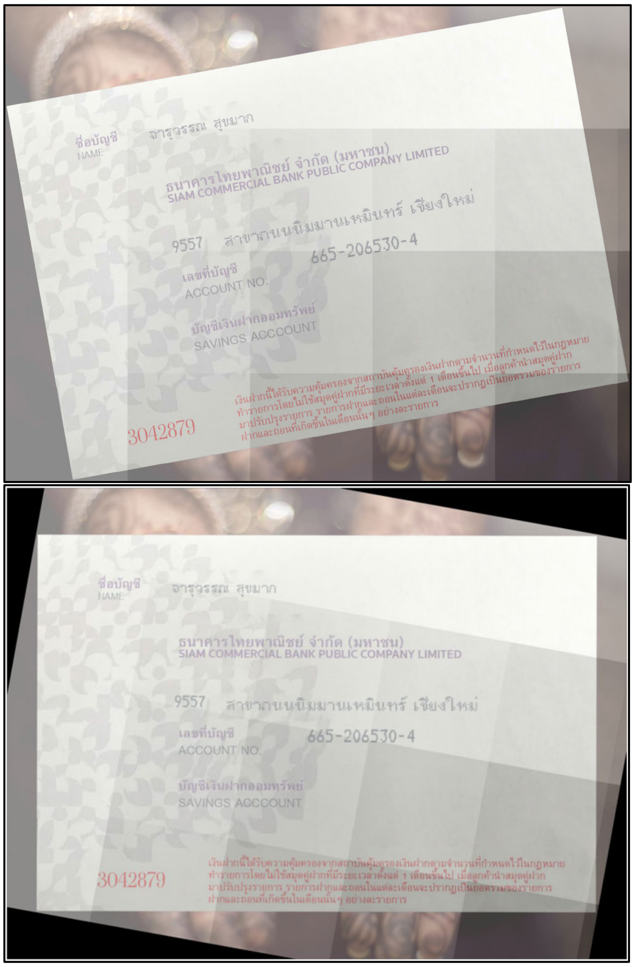
A. BINARIZATION

B. We applied pre-processing to enhance our model's performance by attempting to convert the original image from the RGB format to a binary format [36], where pixels are either black or white. This simplifies the text-detection process, making it easier to identify text regions. However, using a binary format can result in the loss of information from an image. This is shown in Figure 37, particularly in the pale-colored text. We cannot integrate this function into our pipeline directly, but it is possible for users to activate it by setting the ''pre-process'' flag to use this function.

B. DENOISING WITH DEEP LEARNING MODEL

Various denoising models have recently been developed, and some have been utilized with object- or text-detection models to boost accuracy [37]. We experimented with a pre-trained denoising deep neural network known as DnCNN [38]. As shown in Figure 38, the lower image was denoised using the DnCNN model. However, the denoised image appears blurrier than the original one. This could occur because of the large amount of noise in the image or because the model is unsuitable for the document images.

Consequently, the results did not meet our expectations. We decided not to use a denoising model in our pipeline. The focus is on incorporating a broader range of diverse noise



ชื่อบัญชี NAME จารุวรรณ สุขมาก

ธนาคารไทยพาณิชย์ จำกัด (มหาชน) SIAM COMMERCIAL BANK PUBLIC COMPANY LIMITED

9557 สาขาถนนนิมมานเหมินทร์ เชียงใหม่

เลขที่บัญชี ACCOUNT NO. 665-206530-4

บัญชีเงินฝากออมทรัพย์ SAVINGS ACCCOUNT

3042879

เงินฝากนี้ได้รับความคุ้มครองจากสถาบันคุ้มครองเงินฝากตามจำนวนที่กำหนดไว้ในกฎหมาย ทำรายการโดยไม่ใช้สมุดคู่ฝากที่มีระยะเวลาตั้งแต่ 1 เดือนขึ้นไป เมื่อลูกค้านำสมุดคู่ฝากมาปรับปรุงรายการ รายการฝากและถอนในแต่ละเดือนจะปรากฏเป็นยอดรวมของรายการฝากและถอนที่เกิดขึ้นในเดือนนั้นๆ อย่างละรายการ

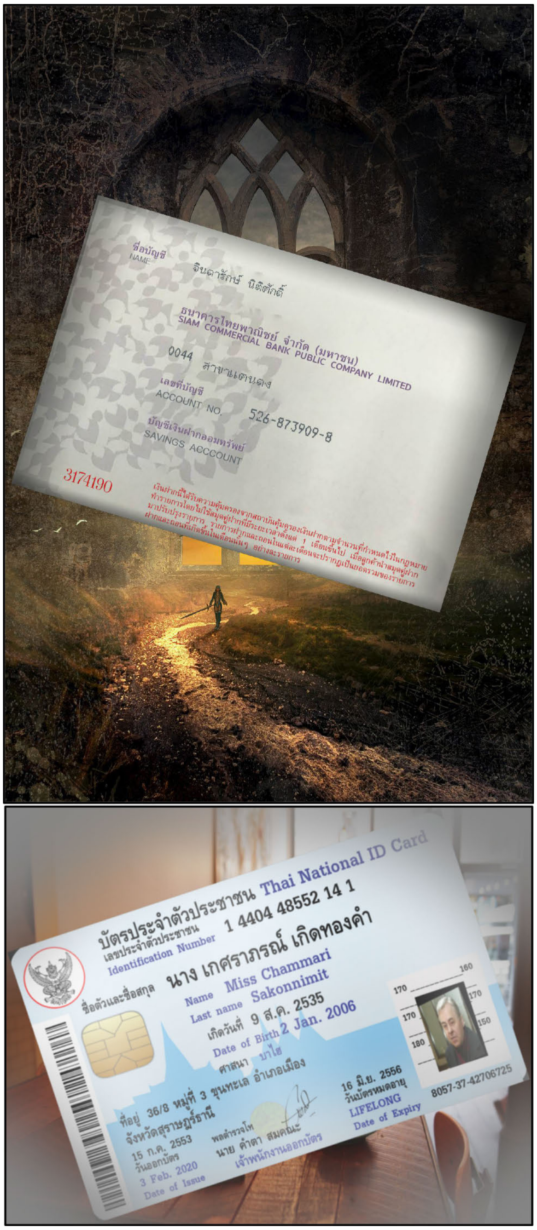
FIGURE 39. **The upper image represents the original image, while the lower image is rotated to the angle from the angle correction function.**

types into the model to enhance its ability to become a robust model.

C. ANGLE CORRECTION

Angle correction is another algorithm that can boost the performance of an OCR pipeline. This process aligns the text in the correct orientation to address potential challenges posed by skewed or rotated text in the input image. OCR models such as text detection and text recognition are typically trained on horizontally aligned texts. If the input image contains skewed or rotated text, the model may struggle to recognize the characters accurately. Angle correction ensures that the text is aligned, thereby improving the chances of correct character recognition.

Angle correction is performed by iterating over a range of degrees (−10 to 10) and rotating the input binary image by each degree. For each rotation, the projection was calculated from the projection of a binary image by counting the number of white pixels in each row. It then counts the rows with fewer white pixels than or equal to 0.01 times × width of the image (width). The final count was then returned. The degree of rotation (best degree) yielded the highest row count with low white pixel counts was stored. This function then refines the best degree by considering a smaller range around the initially determined best degree. Finally, it returns to the optimal degree of rotation.



ชื่อบัญชี NAME จินดารักษ์ นิติศักดิ์

ธนาคารไทยพาณิชย์ จำกัด (มหาชน) SIAM COMMERCIAL BANK PUBLIC COMPANY LIMITED

0044 สาขาแคนดง

เลขที่บัญชี ACCOUNT NO. 526-873909-8

บัญชีเงินฝากออมทรัพย์ SAVINGS ACCCOUNT

3174190

เงินฝากนี้ได้รับความคุ้มครองจากสถาบันคุ้มครองเงินฝากตามจำนวนที่กำหนดไว้ในกฎหมาย ทำรายการโดยไม่ใช้สมุดคู่ฝากที่มีระยะเวลาตั้งแต่ 1 เดือนขึ้นไป เมื่อลูกค้านำสมุดคู่ฝากมาปรับปรุงรายการ รายการฝากและถอนในแต่ละเดือนจะปรากฏเป็นยอดรวมของรายการฝากและถอนที่เกิดขึ้นในเดือนนั้นๆ อย่างละรายการ

บัตรประจำตัวประชาชน Thai National ID Card

เลขประจำตัวประชาชน Identification Number 1 4404 48552 14 1

ชื่อตัวและชื่อสกุล นาง เกศราภรณ์ เกิดทองคำ

Name Miss Chammari

Last name Sakonnimit

เกิดวันที่ 9 ส.ค. 2535

Date of Birth 2 Jan. 2006

ศาสนา บาไย

ที่อยู่ 36/8 หมู่ที่ 3 ขุนทะเล อำเภอเมือง จังหวัดสุราษฎร์ธานี

15 ก.ค. 2553 วันออกบัตร

3 Feb. 2020 Date of Issue

พลตำรวจโท นาย คำตา สมคณะ เจ้าพนักงานออกบัตร

16 มิ.ย. 2556 วันบัตรหมดอายุ

LIFELONG Date of Expiry

8057-37-42706725

FIGURE 40. **These images cannot perform angle correction by using our angle rotation function due to the background.**

The algorithm performs effectively on images such as books, document images with explicit lines of text, and images without a background.

In our experiment, we integrated this function into our pipeline and observed that some images could not be rotated, as shown in Figure 40. The function captures white pixels not only from the document but also from the background in the image, leading to confusion and incorrect degree values.

Consequently, we chose not to integrate this algorithm into the pipelines. Instead, we plan to enhance the OCR accuracy by training the text-detection model with diverse rotation angles, various noise levels, and backgrounds. In addition, for text recognition, we will augment the training data with more skewed images to improve the recognition capabilities of such image types.

Further enhancements of this angle-correction function are required to ensure that our function performs optimally across all types of images.

IX. CONCLUSION

In conclusion, our research focuses on advancing OCR and text detection accuracy across various services, including Thai ID cards (front and back), car registrations, mobile banking slips, book banks, passports, receipts, Indonesian ID cards (front side), and driver licenses (see the sample results in Figures 29, 30, 31, 32, 33, 34, 35, and 36.). The experimental results demonstrated significant progress, outperforming the CRAFT model, particularly for Thai ID card fronts, passports, receipts, car registrations, mobile banking slips, and Indonesian ID cards (front side).

While acknowledging these achievements, we recognize the need for further enhancements, particularly in accuracy, for less commonly used document types such as Thai ID cards back, bank books, and driver's licenses; despite their infrequent use, these documents remain significant in specific applications. Addressing the challenges associated with these document types has the potential to contribute to the development of more comprehensive and versatile OCR and text detection systems.

Our proposed model successfully improved the extraction of bounding boxes for important services, effectively resolving the post-processing issues encountered in the CRAFT model. This enhancement results in improvements in both time inference and resource utilization. Notably, errors, including those related to multiline detection that could affect recognition services, were significantly reduced.

In summary, this study contributes significantly to the advancement of OCR and text-detection technology, specifically in the domain of document services. As we continue to refine and expand our research, future efforts may lead to more robust OCR solutions that can handle a broader range of documents with a higher accuracy and efficiency.

ACKNOWLEDGMENT

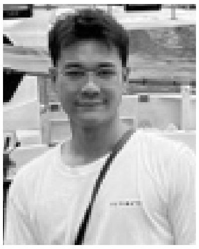
Phanthakan Kiatphaisansophon would like to thank Thanapoom Charoensiri for their unwavering support and valuable input, which significantly contributed to the success of this endeavor. Dittaya Wanvarie would also like to extend her heartfelt thanks to her dedicated co-workers, Rathanon Suwanteerangkul, Duangdao Thasorn, and Tanapat Ratchatorn, for their contributions and collaborative efforts during the various stages of this project. Their enthusiasm and teamwork greatly enriched their research process and outcomes.

The authors extend their gratitude to OpenAI for the instrumental role of ChatGPT and Grammarly in refining this manuscript's grammar and writing style.

REFERENCES

[1] A. P., "Optical character recognition," *Int. J. Sci. Res. Manage.*, vol. 2, no. 1, pp. 72–75, Jul. 2016.
[2] S. K. Gorti and J. Ma, "Text-to-image-to-text translation using cycle consistent adversarial networks," 2018, *arXiv:1808.04538*.
[3] H. Borko and M. Bernick, "Automatic document classification," *J. ACM*, vol. 10, no. 2, pp. 151–162, 1963.
[4] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 161–184, Jan. 2021.
[5] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3304–3308.
[6] J. Cowie and W. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, 1996.
[7] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 9038–9045.
[8] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid mask text detector," 2019, *arXiv:1903.11800*.
[9] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9357–9366.
[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[11] A. O. Vuola, S. U. Akram, and J. Kannala, "Mask-RCNN and U-Net ensembled for nuclei segmentation," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 208–212.
[12] Q. Zhang, X. Chang, and S. B. Bian, "Vehicle-damage-detection segmentation algorithm based on improved mask RCNN," *IEEE Access*, vol. 8, pp. 6997–7004, 2020.
[13] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9156–9165.
[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–14.
[15] V. Pham, C. Pham, and T. Dang, "Road damage detection and classification with Detectron2 and faster R-CNN," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 5592–5601.
[16] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 483–499.
[17] A. V. S. Abhishek and S. Kotni, "Detectron2 object detection & manipulating images using cartoonization," *Int. J. Eng. Res. Technol.*, vol. 10, 2021, pp. 322–326.
[18] A. B. Abdusalomov, B. M. S. Islam, R. Nasimov, M. Mukhiddinov, and T. K. Whangbo, "An improved forest fire detection method based on the Detectron2 model and a deep learning approach," *Sensors*, vol. 23, no. 3, p. 1512, Jan. 2023.
[19] X. Yue, K. Qi, X. Na, Y. Zhang, Y. Liu, and C. Liu, "Improved YOLOv8-Seg network for instance segmentation of healthy and diseased tomato plants in the growth stage," *Agriculture*, vol. 13, no. 8, p. 1643, Aug. 2023.
[20] Z. Khan, Y. Shen, H. Liu, and X. Zeng, "Improved YOLOv8 deep learning instance segmentation algorithm with dilated convolution: Real time precise segmentation of orchard canopies in natural environment," *SSRN J.*, 2023. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4604445
[21] X. Zhang, L. Cheng, B. Li, and H.-M. Hu, "Too far to see? Not really!—Pedestrian detection with scale-aware localization policy," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3703–3715, Aug. 2018.
[22] X. Zhang, S. Cao, and C. Chen, "Scale-aware hierarchical detection network for pedestrian detection," *IEEE Access*, vol. 8, pp. 94429–94439, 2020.
[23] X. Zhang, L. Li, H. Liu, P. Yang, and Y. Gao, "Disentangling classification and regression in Siamese-based network for visual tracking," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 27, p. e7246, Dec. 2022.
[24] J. Ye, Z. Chen, J. Liu, and B. Du, "TextFuseNet: Scene text detection with richer fused features," in *Proc. IJCAI*, 2020, pp. 516–522.

[25] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiseNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. ECCV*, 2018, pp. 325–341.
[26] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[27] C. Y. Lee, Y. Baek, and H. Lee, "TedEval: A fair evaluation metric for scene text detectors," in *Proc. ICDARW*, Sep. 2019, pp. 14–17.
[28] A. S. Lawson, *Anatomy of a Typeface*. Boston, MA, USA: David R. Godine Publisher, 1990.
[29] S. Rostianingsih, A. Setiawan, and C. I. Halim, "COCO (creating common object in context) dataset for chemistry apparatus," *Proc. Comput. Sci.*, vol. 171, pp. 2445–2452, Jan. 2020.
[30] X. Ying, "An overview of overfitting and its solutions," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 022022.
[31] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecol. Model.*, vol. 406, pp. 109–120, Aug. 2019.
[32] Detectron2. (2019). *mask_rcnn_R_50_FPN_1x*. [Online]. Available: https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml
[33] Detectron2. (2019). *mask_rcnn_R_50_FPN_3x*. [Online]. Available: github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml
[34] Detectron2. (2019). *mask_rcnn_R_101_FPN_3x*. [Online]. Available: github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml
[35] Detectron2. (2019). *mask_rcnn_X_101_32x8d_FPN_3x*. [Online]. Available: github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_X_101_32x8d_FPN_3x.yaml
[36] A. Fateh, M. Rezvani, A. Tajary, and M. Fateh, "Persian printed text line detection based on font size," *Multimedia Tools Appl.*, vol. 82, no. 2, pp. 2393–2418, Jan. 2023.
[37] A. Fateh, M. Fateh, and V. Abolghasemi, "Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection," *Eng. Rep.*, Dec. 2023, Art. no. e12832.
[38] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.



PHANTHAKAN KIATPHAISANSOPHON was born in Pak Phanang, Nakhon Si Thammarat, Thailand, in 1999. He received the B.S. degree in software engineering from Kasetsart University, Thailand, in 2022. He is currently pursuing the master's degree with the Faculty of Science, Chulalongkorn University, Thailand.

From 2018 to 2021, he was a Teacher Assistance of computer programming I&II, database and data analytics courses with the Computer Engineering Faculty, Kasetsart University.

DITTAYA WANVARIE (Member, IEEE) received the Ph.D. degree from Tokyo Institute of Technology, in 2011. She is currently an Assistant Professor with the Department of Mathematics and Computer Science, Chulalongkorn University. Her current research interests include information retrieval, natural language processing, and cloud computing.

NAGUL COOHAROJANANONE (Member, IEEE) received the B.S. degree in computer science from Mahidol University, and the M.Eng. and Ph.D. degrees in information and communication engineering from The University of Tokyo. He is currently an Associate Professor with Chulalongkorn University, Thailand. His research interests include computer vision, machine learning, multimedia technology, and user interface design.

• • •