Computers in Industry

Optimizing Image Format P&ID Recognition: Integrating Symbol and Text Recognition with a Single Backbone Architecture

--Manuscript Draft--

Manuscript Number:	
Article Type:	Research Paper
Keywords:	Deep learning, Piping and instrumentation diagram, Object detection, Text spotting, Text recognition
Corresponding Author:	Hyungki Kim Jeonbuk National University KOREA, REPUBLIC OF
First Author:	Junhyung Byun
Order of Authors:	Junhyung Byun
	Bonggu Kang
	Duhwan Mun
	Gwang Lee
	Hyungki Kim
Abstract:	Recent studies propose deep learning-based methods to recognize symbols and text, the primary components of Piping and Instrumentation Diagrams (P&ID). However, existing studies implement a complex process where the object detection model consists of two individual detection models for symbol and text, and the text recognition model is separated from the text detection model. Therefore, we propose an integrated model with a single symbol-text detection module and text recognition module by applying a text spotting method which utilizes the detected text features for recognition. Our proposed model extracts detected text regions' features which are encoded with local information of characters for text recognition, eliminating the need for multiple layers for encoding local information of characters in our text recognition module. Thus, our text recognition module, being lightweight, reduces the time required for text recognition. Furthermore, as our proposed model facilitates end-to-end learning between the symbol-text detection and the text recognition modules, it enables semantic information transmission between these modules, resulting in better text detection and recognition compared to the process where symbol-text detection and text recognition models are separated. Additionally, during the training phase of our proposed model, the text recognition module leverages text features, eliminating the need to generate and store text images for training. To identify the practical applicability of our proposed model, we tested our proposed model on P&ID images used in actual industries. The results for symbol-text detection/text recognition performance, with an IoU (Intersection over Union) threshold of 0.5, were evaluated as a maximum precision of 97.63%/95.27%, recall of 95.21%/90.75%, and F1 score of 96.40%/92.95%.
Suggested Reviewers:	Fazhi He Wuhan University fzhe@whu.edu.cn
	Jinwon Lee Gangneung-Wonju National University jwlee@gwnu.ac.kr

Highlights

- Proposed deep-learning architecture for image format P&ID (Piping and Instrumentation Diagram) recognition
- Extended state-of-the-art text spotting method for simultaneous symbol and text recognition
- Reuse of encoded image features improved inference time and performance
- Demonstrated that the reuse of encoded image features helps to reduce recognition error
- Achieved 96.40%/92.95% F1-score for detection/recognition, proving industrial applicability

Title Page (with Author Details)

Optimizing Image Format P&ID Recognition: Integrating Symbol and Text Recognition with a Single Backbone Architecture

Junhyung Byun[a], Bonggu Kang[a], Duhwan Mun[b], Gwang Lee[c], Hyungki Kim[a,*]

[a]Department of Computer Science and Artificial Intelligence/CAIIT, Jeonbuk National University, Jeonju, Republic of Korea

[b]School of Mechanical Engineering, Korea University, Seoul, Republic of Korea

[c]CCLSOFT Co., Ltd., Sejong, Republic of Korea

* Corresponding Author,

Email: hk.kim@jbnu.ac.kr

CRediT authorship contribution statement

Junhyung Byun: Methodology, Software, Investigation, Visualization, Writing – original draft, Writing – review & editing **Bonggu Kang:** Software, Data curation **Duhwan Mun:** Conceptualization, Methodology, Supervision **Gwang Lee:** Resources **Hyungki Kim:** Conceptualization, Methodology, Supervision, Validation, Writing – review & editing

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data that support the findings of this study are not publicly available due to intellectual property restrictions. However, the data may be available from the authors upon reasonable request and with permission from the property owner.

Acknowledgment

This work was supported by the BK21 FOUR (Fostering Outstanding Universities for Research) funded by the Ministry of Education(MOE, Korea) and the National Research Foundation of Korea(NRF), by the Basic Science Research Program (No. NRF-2022R1A2C2005879) through the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) and by the AI-based gasoil plant O&M Core Technology Development Program (No. 21ATOG-C161932-01) funded by the Korean government (MOLIT).

Abstract

Recent studies propose deep learning-based methods to recognize symbols and text, the primary components of Piping and Instrumentation Diagrams (P&ID). However, existing studies implement a complex process where the object detection model consists of two individual detection models for symbol and text, and the text recognition model is separated from the text detection model. Therefore, we propose an integrated model with a single symbol-text detection module and text recognition module by applying a text spotting method which utilizes the detected text features for recognition. Our proposed model extracts detected text regions' features which are encoded with local information of characters for text recognition, eliminating the need for multiple layers for encoding local information of characters in our text recognition module. Thus, our text recognition module, being lightweight, reduces the time required for text recognition. Furthermore, as our proposed model facilitates end-to-end learning between the symbol-text detection and the text recognition modules, it enables semantic information transmission between these modules, resulting in better text detection and recognition compared to the process where symbol-text detection and text recognition models are separated. Additionally, during the training phase of our proposed model, the text recognition module leverages text features, eliminating the need to generate and store text images for training. To identify the practical applicability of our proposed model, we tested our proposed model on P&ID images used in actual industries. The results for symbol-text detection/text recognition performance, with an IoU (Intersection over Union) threshold of 0.5, were evaluated as a maximum precision of 97.63%/95.27%, recall of 95.21%/90.75%, and F1 score of 96.40%/92.95%.

Keywords: Deep learning, Piping and instrumentation diagram, Object detection, Text spotting, Text recognition

1. Introduction

Piping and instrumentation diagrams (P&IDs) are detailed schematics that depict the process flow within a plant or industrial facility. P&IDs include various piping, process equipment, instruments, and control devices used to monitor and control the process, as well as identification numbers, labels, and annotations. P&IDs are crucial for the design, construction, operation, and maintenance of process systems. They serve as visual references for engineers, operators, and maintenance personnel to understand how systems operate, troubleshoot issues, and ensure that systems operate safely and efficiently within defined regulations.

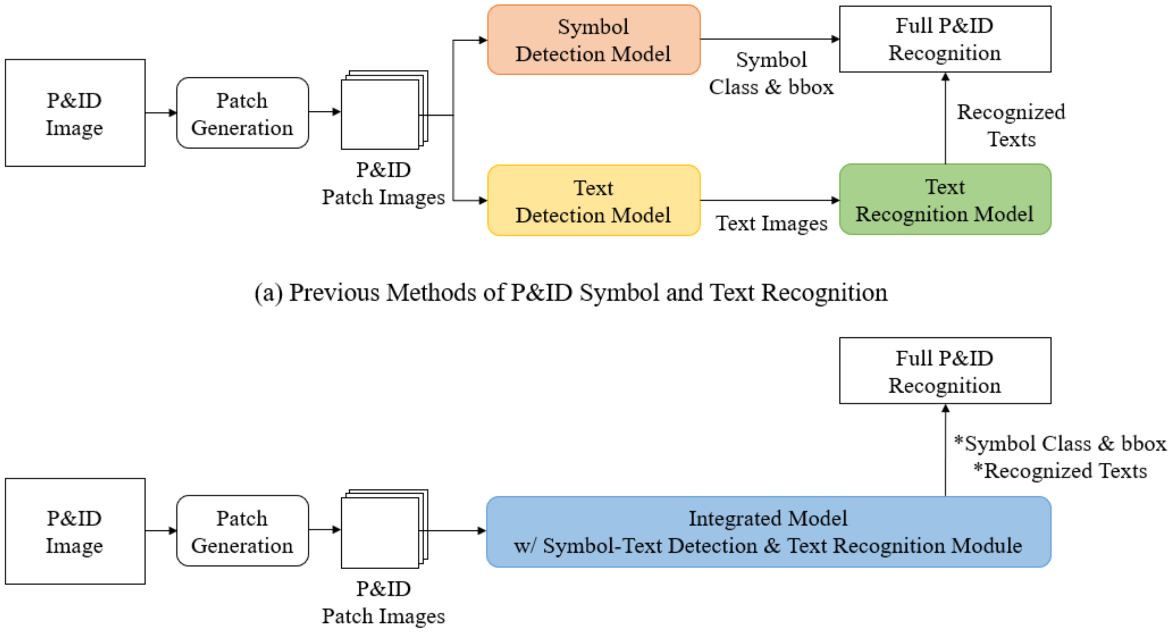
Digital P&IDs store information about objects in a database, making it easy for users to identify objects. Additionally, they are convenient because they can be modified using software. Furthermore, digital P&IDs are typically saved in electronic file formats, making them easy to share and store digitally. Therefore, in recent years, digital P&IDs, which allow easy access to, and modification of information represented in diagrams, have been widely adopted in the industry.

Due to the various advantages of digital P&IDs, EPC (Engineering, Procurement, and Construction) companies utilize them in project execution. However, during projects, EPC companies often receive P&IDs in image format from collaborating companies, which can hinder the utilization of digital P&IDs. Furthermore, older plants, in contrast to recently constructed ones, possess P&IDs in image format as they were produced through analog methods before the advent of digital P&IDs. Consequently, these aging plants are unable to leverage the benefits of digital P&IDs for plant improvement and expansion. To harness the advantages of digital P&IDs, it is necessary to recognize the components of non-digital P&IDs and undergo a process of digitization.

In P&ID drawings, there are main components consisting of symbols and text. Symbol types include piping, equipment, and instrumentation, while text is present to provide identification numbers or relevant information for the symbols. Typically, in industry, high-level objects existing in image-format P&IDs are manually identified by workers and then converted into digital P&IDs through manual labor. This method is time-consuming and incurs significant costs due to the use of specialized personnel. Additionally, the outcome of converting to digital P&IDs may vary depending on the worker's expertise.

Due to the drawbacks associated with manually converting image-format P&IDs to digital P&IDs, research on deep learning-based P&ID recognition for automatic digital P&ID conversion has been proposed. Process in Figure 1-(a) illustrates the method used by existing studies [1, 2, 3, 4, 5] to recognize P&ID symbols and text. The P&ID symbol and text recognition process in existing studies proceeds with separate object detection and text recognition models. The object detection model consists of two individual detection models for symbols and text, and the text recognition model recognizes characters of detected text images extracted from the P&ID drawing. When the text detection model and text recognition model are separated, the text recognition model cannot acquire features for recognition from the text detection model. Therefore, the text recognition model needs a large module consisting of multiple layers to encode local information of text images. It increases the size of the text recognition model and slows down text recognition speed. Additionally, when the text detection model and recognition models are separately trained, it is difficult to obtain better text detection and recognition results than when the two models are integrated. This is because semantic information transmission between the text detection model and the recognition model is not possible.

In this study, we propose a model where the symbol-text detection module and text recognition module are integrated, as illustrated in Figure 1-(b). The proposed model detects symbols and text with a single detection module [49], and then extracts features of detected text regions for text recognition. Extracted features for text recognition are already encoded with local information of characters, so the text recognition module doesn't need multiple layers to encode local information of text features. Consequently, text recognition is possible with a lightweight text recognition module. Furthermore, since the proposed model is an integrated model with the symbol-text detection and text recognition modules, they are trained together. Therefore, the proposed model is trained to facilitate semantic information transmission between the text detection and recognition results, enabling more accurate text detection and recognition results compared to the process where symbol-text detection and text recognition models are separated.



(a) Previous Methods of P&ID Symbol and Text Recognition

(b) Proposed Method P&ID Symbol and Text Recognition

Fig. 1. Comparison with proposed method and previous methods.

The academic contributions are as follows.

- First, to the best of the author's knowledge, this is the first study to conduct text spotting for both text and multi-class objects. The text spotting model is an integrated model with the text detection and recognition modules utilizing detected texts′ features for recognition, which have primarily been researched for text only. However, our proposed model consists of a single symbol-text detection module, not divided into symbol detection and text detection modules, also our proposed model′s lightweight text recognition module recognizes characters of text quickly by leveraging text features among multiple symbol classes and text classes.

- Second, unlike previous research [4], which required generating and storing text images for training the text recognition model, our proposed model trains the text recognition module without this step, thereby achieving maximum performance for recognizing texts within P&ID drawings. Since our proposed model trains the symbol-text detection and text recognition modules together, only the ground-truth text needs to be labeled in the detection dataset.
- Lastly, the feasibility of the proposed method for real-world applications was confirmed. Symbol and text detection performance, as well as text recognition performance, were measured for P&ID drawings used in industries. Each of the 20 test drawings, consisting of 82 classes, contains a dense presence of symbols and texts ranging from a minimum of 248 to a maximum of 1120 instances per drawing. For the 20 P&ID test drawings, the proposed model achieved a maximum precision of 97.63%/95.27%, recall of 95.21%/90.75%, and F1 score of 96.40%/92.95% for symbol-text detection/text recognition.

The remaining content of this paper is described as follows. Section 2 provides an overview of related works. Sections 3 and 4 contain detailed explanations of our proposed methodologies and experimental results, respectively. Finally, Section 5 includes conclusions and future works.

2. Related works

2.1. P&ID recognition

Many studies have proposed approaches to P&ID recognition based on deep learning. [1, 2, 3, 4, 5] address symbol and text detection as well as text recognition. [1, 5] utilize GFL [10] as the symbol detection model. [2, 3] employ FCN [11] and CNN (Convolutional Neural Network), respectively, for symbol detection. [4] detects symbols using template matching. For text detection, [2] uses CTPN [12], while [1, 5] uses CRAFT [13]. [3] utilizes EAST [7] for text detection. [4] extracts text contour regions as a pre-process before text recognition. [1, 2, 3, 4, 5] utilize Tesseract OCR [9] for text recognition, and [4] further enhances text recognition accuracy by storing text images which the text recognition model failed to predict in a database for training the text recognition model.

The existing P&ID recognition process utilizes individual models for symbol and text detection. Text regions acquired through the text detection model are extracted as images and then the text recognition model recognizes characters in text images. When the text detection and recognition models are separated, images are used instead of features acquired from the text detection model for recognition. Consequently, modules consisting of multiple layers for encoding local information of text images are required, leading to an increase in the size and recognition time of the text recognition model. Additionally, when the text detection and recognition models are trained separately, the lack of semantic information transmission between them makes it challenging to achieve better text detection and recognition. In this study, by proposing an integrated model with a single symbol-text detection module and a lightweight text recognition module, fast and accurate P&ID symbols and text recognition becomes possible.

2.2. Object detection

Object detection is a task in the field of computer vision, involving the classification and localization of objects within a given image. With the emergence of CNN [14], deep learning has become the primary method for object detection in various fields [6, 16, 33]. In deep learning-based object detection, when an image is inputted, the model independently extracts features and performs learning, allowing it to simultaneously determine the coordinates of bounding boxes representing the location of objects in the image, their classes, and confidence scores for detection. Methods of deep learning-based object detection can generally be distinguished concerning anchor boxes.

Anchor boxes are pre-defined bounding boxes used to detect objects in an image. During the training of anchor-based detectors, these anchor boxes are adjusted to ensure effective object detection. Single-stage detectors utilizing anchors such as RetinaNet [15] and DDOD [17] classify and locate objects in one step without region proposals. In contrast, two-stage detectors like Faster R-CNN [18] utilize a Region Proposal Network (RPN) [18] to first identify potential object locations and then extract features from these locations to determine the classes and positions of objects. Recently, research has extended beyond these two stages to multi-stage object detection methods, such as Cascade R-CNN [19] and Sparse R-CNN [20]. Cascade R-CNN [19] achieves refined detection by increasing the IoU threshold at each stage. Sparse R-CNN [20] provides a fixed number of candidate objects and refines the bounding boxes of these candidates in each stage to match the regions of interest for the objects being detected.

Among anchor-free methods, one approach is based on key points [21,22]. CornerNet [21] utilizes top-left and bottom-right corners for object detection, while CenterNet [22] uses the center of the bounding box as a key point. FCOS [23] and Varifocalnet [24] detect objects without using anchors, employing a structure consisting of backbone-FPN (Feature Pyramid Network) [26]-heads. FCOS is structured with ResNet [25]-FPN-Head and generates a centerness map to remove low-quality bounding boxes in addition to the object class and bounding box coordinates as output. Varifocalnet [24], combining FCOS [23] and ATSS [27], improves performance through a proposed loss function and star-shaped bounding boxes. Besides, studies that utilize transformer encoder-decoder [28] structures with object queries for object detection such as DETR (Detection Transformer) [29] and Deformable DETR [30] have been proposed.

Traditionally, object detection models have been categorized into single-stage and two-stage approaches. Single-stage object detection models are generally simpler in structure, allowing for fast detection but were considered to have lower performance compared to two-stage models. However, advancements in single-stage methods have enabled fast and accurate detection even with simple structures. Therefore, in most previous P&ID recognition studies, symbol detection models were selected through comparisons among single-stage object detectors. However, to select the most superior model for symbol and text detection, we compared single-stage, multi-stage, anchor-free, and transformer-based object detection models. The structure of Sparse R-CNN [20] which is a multi-stage but fast object detection model, is utilized as the detection module of our proposed model.

2.3. Text recognition

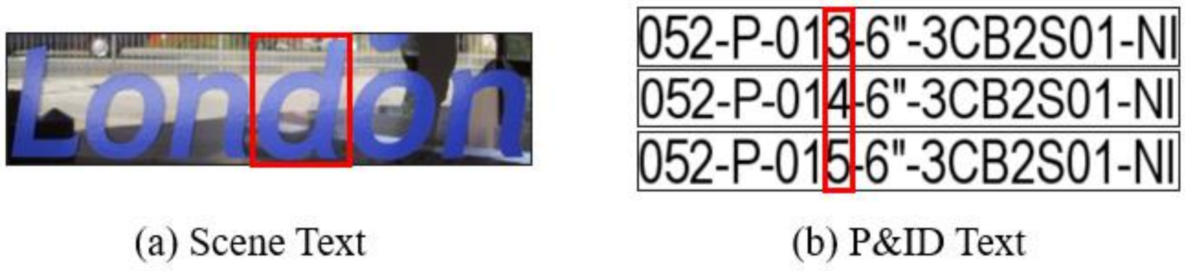
Text recognition is one of the tasks in the field of computer vision, aiming to recognize the text present in given images. Currently, the text recognition field is referred to as Scene Text

Recognition, which recognizes text appearing in natural images with diverse backgrounds. Various deep learning-based approaches with different structures have been proposed for Scene Text Recognition.

Rosetta [31] and SVTR [32] recognize text by distinguishing the visual characteristics of characters in text images. They can infer text quickly as they do not require additional tasks considering the context or relationships of characters. Rosetta [31] is a CNN-based text recognition model that uses ResNet-18 as its backbone. SVTR [32] is a vision transformer-based [48] text recognition model. SVTR [32] reduces the height of extracted features through patch embedding while encoding both local and global information of characters.

The model CRNN [34] is structured by combining CNN and RNN (Recurrent Neural Network). CRNN initially extracts visual features of characters in text images by using CNN and then recognizes text by considering the context of visual features with RNN. Subsequently, many studies proposing text recognition through encoder-decoder structures such as SEED [35] and ASTER [36] have emerged. These studies utilize attention mechanisms to consider the relationships between characters for recognition. Recently, research has also proposed methods such as ABINet [37] and SRN [38], which fuse the visual information of characters with linguistic rules considering the relationships between characters. The studies mentioned above have improved text recognition performance by considering the continuity of characters within the text and the relationships between characters. However, as the size of the models increases, the recognition speed may decrease.

In this study, we utilized partial structure of text recognition model which rely solely on the visual features for text recognition as our text recognition module, because it is challenging to perceive relationships between characters in the text of P&ID. As depicted in the left image of Figure 2, the scene text consists of words existing in the surrounding environment, thus exhibiting regularity. Therefore, it is possible to infer the next character based on the relationships between characters. However, as depicted in the right image of Figure 2, P&ID text represents identification numbers or process information, making it difficult to perceive relationships between characters. Thus, partial structure of text recognition model capable of recognizing characters without considering the context or relationships between characters was utilized as our text recognition module.



(a) Scene Text (b) P&ID Text

Fig. 2. Comparison with scene text and P&ID text.

2.4. Text spotting

Text spotting is a method that enables an end-to-end manner from text detection to text recognition. There are two approaches to achieving end-to-end manner from text detection to recognition. The conventional approach utilizes a process where the text detection and recognition model are separated. The process recognizes characters of text images that are extracted from the detected regions. [39, 40] are studies that utilize a process where text detection and recognition models are separated. [39, 40] constitute a series of studies, where the text detection model is structured with convolution layers added to the backbone. For text recognition, CRNN [34] was utilized.

Recently, studies like [41, 42], where features extracted from detected text regions are utilized for recognition. [41] employs an MLP (Multi-Layer Perceptron) for the classification and regression of proposed regions, and utilizes a Region Feature Encoder [41] for text feature extraction. An encoder-decoder is used for recognizing the extracted features. [42] utilizes a CNN-based detection module to detect text, and then extracts features from detected regions using a Text-Alignment layer [42]. The extracted features are then recognized through an encoder-decoder module.

Research on P&ID recognition typically employs a process where the text detection and recognition models are separated. Consequently, when the text detection and recognition models are trained separately, it becomes difficult to transmit semantic information between them, making it challenging to obtain better results for text detection and recognition. On the other hand, we propose a new integrated model with symbol-text detection and text recognition modules. The proposed model not only detects text but also detects symbols and text together, and then utilizes features of detected text regions for recognition.

3. Methodology

In this study, we propose a model that integrates a single symbol-text detection module with a text recognition module by applying a text-spotting method. The proposed model is capable of end-to-end learning from symbol-text detection to text recognition modules. The description of the proposed model proceeds as follows: Section 3.1 describes the overall structure of the proposed model. Sections 3.2 and 3.3 explains details of the single symbol-text detection module and the text recognition module, respectively. Finally, Section 3.4 describes the training method of the proposed model.

3.1. Overall Architecture

Figure 3 illustrates the overall structure of the proposed model. The P&ID patches are images sliced from drawings using a sliding-window method. The P&ID patches are utilized as inputs, and a shared backbone generates four feature maps reduced in size by 1/4, 1/8, 1/16, and 1/32 from the input image. The shared backbone, serving as the backbone for both the symbol-text detection module and the text recognition module, employs ResNet50-FPN (Feature Pyramid Network). Subsequently, the symbol-text detection module detects symbols and texts for each patch. For the detected text regions, features for recognition are extracted from the feature maps using RoI Align [43]. After symbol-text detection and text feature extraction for all patches are completed, the detection results consist of confidence scores, class numbers, and coordinates of detected regions for symbols and texts. Additionally, extracted features are added for texts. Before applying Adaptive NMS (Non-maximum Suppression) [45] to the detection results, the coordinate values representing the detection regions of symbols and texts in the patch images are converted to coordinates within the entire P&ID drawing. Then, Adaptive NMS, as described in [1], is applied to remove duplicate detection results. Duplicate detection results are removed because during the step of converting the P&ID drawing into patch images using the sliding window method, objects might be redundantly included in the patch images. After removing duplicated symbols and texts through Adaptive NMS, the features retained from text detection results among the remaining detection results of symbols and texts are used as inputs for the text recognition module. After the text recognition stage, the text results contain text detection results with their recognized texts. Finally, the entire P&ID drawing is recognized based on the text results and symbol results.

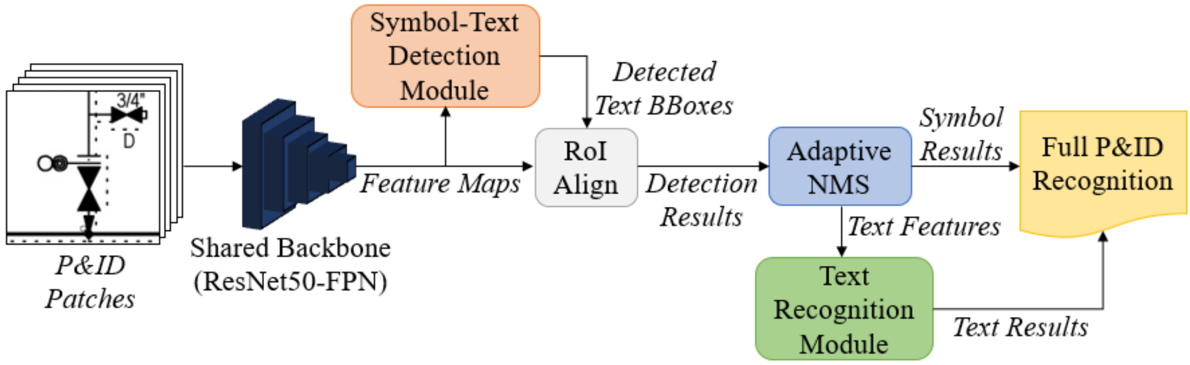


Fig. 3. Over architecture of proposed method.

3.2. Symbols and Text Detection Module

Prior research has been conducted to select the symbol-text detection module of the proposed model. Table 1 compares the performance of object detection models' P&ID symbols and text detection.

Table 1.

Object detection models' performance on P&ID images.

Model	Deformable DETR	**Sparse R-CNN**	DDOD	VarifocalNet
Schedule	3x	**1x**	1x	2x
Training Time	2d 5h	**12h**	12h	23h
Precision	0.9679	**0.9765**	0.9642	0.9241
Recall	0.9452	**0.9492**	0.9188	0.8972
AP	0.308	**0.383**	0.235	0.347
AP50	0.365	**0.417**	0.257	0.384
AP75	0.334	**0.404**	0.256	0.379

To select the symbol-text detection module of the proposed model, representative models were selected from overall deep learning-based object detection methods. The chosen object detection models include Deformable DETR [30] with an encoder-decoder structure, Sparse R-CNN [20] from the multi-stage family, DDOD [17] as a single-stage model, and VarifocalNet

[24] based on anchor-free methods. According to the experimental results shown in Table 1, Sparse R-CNN exhibited the best performance for symbol-text detection in P&ID drawings while requiring shorter training time. Sparse R-CNN first identifies a fixed number of potential locations where objects might exist and then performs classification and regression, leading to better detection results. Therefore, the proposed model utilized the structure of Sparse R-CNN as the symbol-text detection module.

The symbol-text detection module in the proposed model provides a fixed number of proposal regions for each P&ID patch image to detect symbols and text. Unlike other object detection models that typically provide proposal regions through RPN, our symbol-text detection module utilizes an Nx4-sized lookup table trained on the training dataset to provide a fixed number of proposal regions. Here, N represents the number of proposal regions, set to 100. The 4 denotes the normalized center (x, y), height, and width of the bounding box representing the proposal region. The bounding boxes representing proposal regions of size Nx4 indicate potential locations of symbols and texts regardless of the input image. Additionally, an Nxd-sized lookup table trained on the training dataset is provided. The Nxd (=256) sized lookup table complements the information provided by Nx4-sized proposal regions. When Nx4-sized proposal regions and Nxd-sized features are provided together, the symbol-text detection module classifies and refines the bounding boxes of proposal regions gradually to match the regions of symbols and text in the input image through the Dynamic Instance Interactive Head [20] present at each of the I (=6) stages.

The advantage of utilizing the structure of Sparse R-CNN for the symbol-text detection module is that it identifies unnecessary regions through 6 stages without the need for the NMS (Non-Max Suppression) stage, as it uses a fixed number of proposed regions. Thus, despite being a multi-stage detection module, it enables fast and accurate symbol-text detection.

Finally, symbols detected through 6 stages for each patch image retain their confidence score, class number, and coordinates of the detected region. The detected text retains its confidence score, class number, coordinates of the detected region, and corresponding regions′ features extracted from the feature maps generated by the shared backbone using RoI Align. These extracted text features are later used for recognition in the text recognition module.

For training the detection module, a multi-loss used in Sparse R-CNN is utilized, along with the application of set prediction loss [29]. The set prediction loss computes the loss through optimal bipartite matching between a fixed number of proposal regions and ground-truth objects. Hence, each loss within L_{det} is normalized by the number of matched pairs (=M) between ground-truth and predicted objects in the training batch.

$$L_{det} = \lambda_{cls}L_{cls} + \lambda_{giou}\mathrm{L}_{giou} + \lambda_{L1}L_{L1} \quad (1)$$

$$L_{cls} =-\frac{1}{M}\sum_{i=1}^{M}\alpha(1-p_i)^{\gamma}log(p_i) \quad (2)$$

$$\mathrm{L_{giou}} = \frac{1}{M}\sum_{i=1}^{M}(1-GIoU) \ \ (\mathrm{GioU} = \frac{|b_i \cap \overset{\wedge}{b_i}|}{|b_i \ U \ \overset{\wedge}{b_i}|} - \frac{|s_i\backslash(b_i \ U \ \overset{\wedge}{b_i})|}{|s_i|}) \quad (3)$$

$$\mathrm{L}_{L1} = \frac{1}{M}\sum_{i=1}^{M}(\sum_{j=1}^{4}|y_{ij} - y^{\wedge}{}_{ij}|) \quad (4)$$

L_{cls} is the loss function for classification, employing focal loss [15]. Here, p_i represents the predicted probability value, while α and γ serve as balancing and focusing parameters. α is used to address class imbalance, while γ adjusts the down-weighting for easily classified cases. L_{giou} and L_{L1} are loss functions related to the regions of detected and ground-truth bounding boxes. L_{giou} denotes the generalized IoU loss [46]. In the equation, s_i represents the minimum-sized bounding box containing both the ground-truth b_i and predicted $\overset{\wedge}{b_i}$ bounding boxes. By using L_{giou}, even if the ground-truth and detected bounding boxes do not overlap, the extent to which they are separated is reflected in the loss, resulting in better detection results. L_{L1} loss refers to the difference in center (x, y), height, and width between ground-truth and detected bounding boxes. The predicted value $\hat{y_{ij}}$ trained to closely match the value of the ground-truth y_{ij}. In our experiment, we fixed λ_{cls} = 2, λ_{L1} = 5, λ_{giou} = 2 as default values and conducted the experiments.

Once the symbol-text detection and text feature extraction for all patch images of the entire P&ID drawing is completed, the next step is to remove duplicate objects. When generating patch images from the entire P&ID drawing using a sliding window manner, an object may be redundantly included in different patch images. Therefore, to output the results for the entire P&ID drawing based on the detection results of P&ID patch images, duplicate objects included in different patch images must be removed.

To remove duplicate objects, the coordinates of the detected symbols and texts in P&ID patch images need to be converted to coordinates in the entire P&ID drawing. By converting the coordinates of the detection results for P&ID patch images to coordinates in the entire P&ID drawing, detected symbols are composed of (confidence score, class number of the symbol, coordinates of the symbol region in the entire P&ID drawing), and detected texts are composed of (confidence score, text class number, coordinates of the text region in the entire P&ID drawing, feature extracted from the feature maps).

Subsequently, Adaptive NMS is employed for removing duplicate detections. By utilizing Adaptive NMS as a method for removing duplicate detections, different IoU thresholds are applied for each class. This allows for setting higher thresholds for densely clustered classes to retain duplicate detections~~,~~ while lowering the threshold for sparsely clustered classes to remove duplicate detections.

3.3. Text Recognition Module

After removing duplicate objects through the Adaptive NMS, the features of the detected texts are used as inputs to the text recognition module among the remaining detected symbols and texts. As a text recognition module, CNN-based and ViT (Vision Transformer)-based text recognition modules are compared. The CNN-based text recognition module utilizes part of the structure of Rosetta [31], a CNN-based text recognition model, while the ViT-based text recognition module utilizes part of the structure of SVTR [32], a ViT-based text recognition model. Since the features used as input for the text recognition module were extracted from the feature maps generated by a shared backbone, the text features are encoded with local information of characters. Therefore, our text recognition module doesn't need multiple layers for encoding local information of text features because it uses text features encoded with local information of characters as input. However, a text recognition model typically requires modules composed of multiple layers to encode local information, taking text images as input. Hence, our text recognition module can recognize text through a simpler structure than its original model.

First, the left image of Figure 4 depicts the structure of our CNN-based text recognition module. It consists of three convolution blocks and one convolution layer. The inputs to the CNN-based text recognition module are detected text regions′ features of size Nx8xWxC extracted from the feature maps generated by the shared backbone using RoI Align. Here, N, 8, W, and C respectively represent the number of text features remaining after Adaptive NMS, the height, the width, and the channel of text features. Each convolution block consists of a 3x3 convolution layer, batch norm, and ReLU. Shortcut connections are used in each convolution block to prevent the gradient vanishing phenomenon. In the cases of convolution blocks 1 and 2, since the sizes of input features and output features are different during shortcut connection, the input features are passed through a 1x1 convolution layer with a stride of (2,1) and batch norm before being added to the output features of the block (dashed line in the image). After passing through the three convolution blocks and a 2x1 convolution layer, the final output feature size becomes NxWxC'. W is set to 62 because the longest text to be recognized in the P&ID dataset used in the experiment is 62 characters long. C' represents the number of classes of characters, which is 87 in this case. The character classes include uppercase and lowercase alphabets, numbers, and special characters.

Next, here's an explanation of the ViT-based text recognition module. The ViT-based text recognition module utilizes part of the structure of SVTR [32]. SVTR has four structures depending on size, and preliminary research has been conducted to select one of these structures. Table 2 presents the comparison results of P&ID text recognition among SVTR models, using cropped text images that match the ground-truth regions on the P&ID drawings as the test set.

Table 2.
SVTR models' text recognition performances on P&ID text.

	Epoch	Training Time	WEM	Params(M)
SVTR-T	20	1h 31m	87.43	**4.20**
SVTR-S		2h 9m	88.55	8.51
SVTR-B		3h 16m	88.89	22.75
SVTR-L		4h 33m	**92.85**	38.93

All models were trained with the same number of epochs, and the performance of the trained models was measured using WEM (Word Exactly Matching). WEM determines correct recognition when all characters in the recognized text match those in the ground-truth text. Although there is a difference in the number of parameters between Tiny, Small, and Base models, there was not a significant performance variation. SVTR-L, with 38.93M parameters, was the largest model and exhibited the best text recognition performance. However, in this study, verifying the benefits of integrating the symbol-text detection module and the text recognition module is important, so a part of the structure of SVTR-T, which requires the least training time, is used as the text recognition module for efficiency in experiments.

The ViT-based text recognition module proposed in this study only utilizes the last 3 global mixing blocks of the original SVTR-T model. SVTR-T consists of a patch embedding stage to generate text images at the patch level, 6 local mixing blocks, and 6 global mixing blocks to capture internal and external patterns of characters, ultimately generating a one-dimensional feature through a 1x1 linear layer. The local mixing blocks, which capture internal patterns of characters, apply a self-attention mechanism by moving a pre-defined size mask in a sliding-window manner to encode local information of characters. However, in our ViT-based text recognition module, local mixing blocks were removed since input text features extracted from the feature maps generated by the shared backbone are already encoded with local information of characters. Therefore, our text recognition module utilizes the global mixing blocks of the multi-head self-attention mechanism to distinguish between character and non-character parts of the text features. Instead of using all 6 global mixing blocks, only 3 global mixing blocks are used in our ViT-based text recognition module, reducing text recognition time without compromising text recognition performance.

The structure of the ViT-based text recognition module is depicted in the right image of Figure 4. The inputs to the ViT-based text recognition module are detected text regions′ features of size Nx2xWxC extracted from feature maps generated by the shared backbone using RoI Align. Here, N, 2, W, and C denote the number of remaining text features after Adaptive NMS, the height, the width, and the channel of text features, respectively. Since there is no patch embedding module in the ViT-based text recognition module, the Nx2xWxC text features are flattened to Nx2*WxC dimensions to form a patch-like shape. Each global mixing block consists of layer norm, global mixing, MLP, and shortcut connection. In the global mixing step, multi-head self-attention operations are performed to differentiate the importance of text and non-text parts in the patch-shaped features. Subsequently, the features passed through the 3 global mixing blocks undergo pooling, a 1x1 convolution layer, a hard swish activation function [47], and a dropout layer, resulting in features with a height of 1. Features with a height of 1 pass through a linear layer and features of size NxWxC' are output. W and C' represent the length of the text and the number of character classes. W and C′ set to 62 and 87.

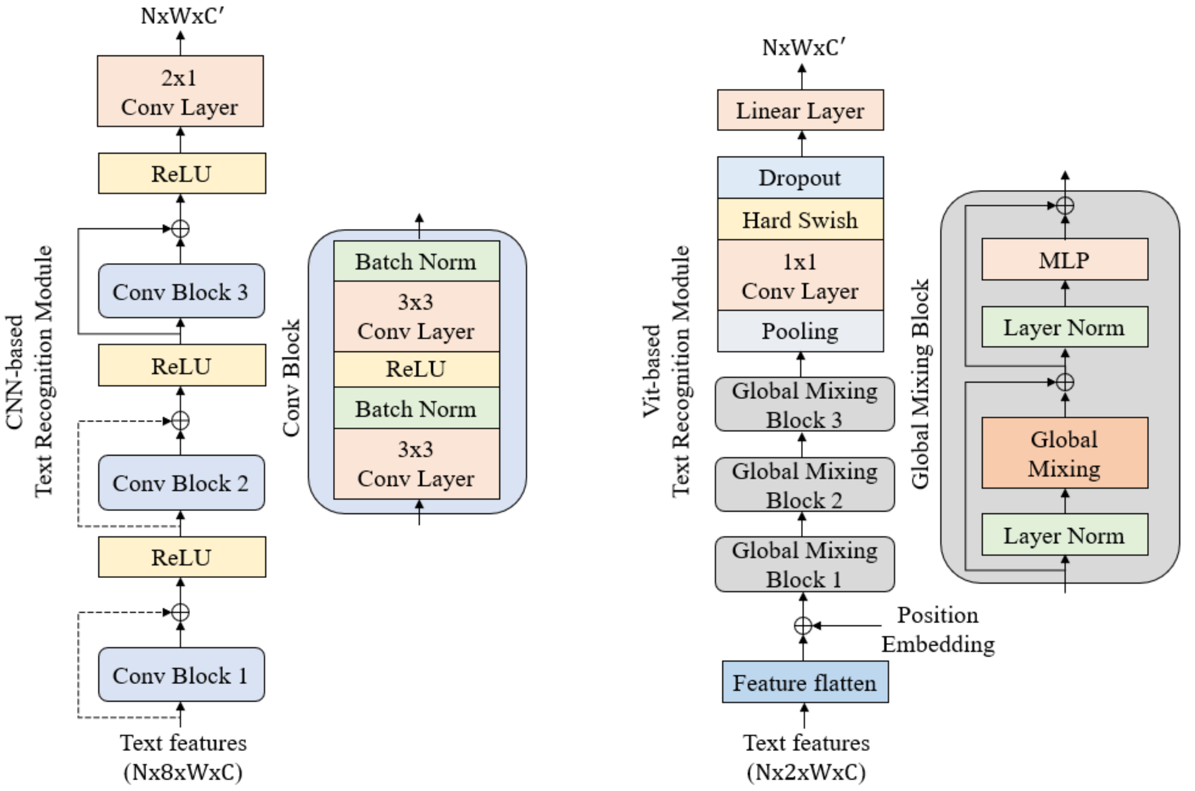


Fig. 4. Structure of CNN-based (left) and ViT-based (right) text recognition module.

Finally, both CNN-based and ViT-based text recognition modules pass each final output feature of size NxWxC' through a CTC (Connectionist Temporal Classification) decoder [44] respectively to recognize text with the best combination of characters. After the text recognition stage, recognized texts are matched with their corresponding detected region. Then, lastly, the matched results (=text results) and symbol results are used for recognizing the entire P&ID drawing.

The training loss for the text recognition module is the CTC (Connectionist Temporal Classification) loss [44], and the formula is as follows. First, it calculates the sum of probabilities of all paths π that match the ground-truth label sequence L, when the blank B is omitted in the predicted sequence of length W.

$$\mathrm{P(L|W)} = \sum_{\pi \in B^{-1}(L)} p(\pi|W) \quad (5)$$

$$L_{rec} = -\frac{1}{N}\sum_{i=1}^{N} \log(p(L_i|W_i)) \quad (6)$$

The training strategy involves maximizing the log likelihood of $p(L_i|W_i)$, where N denotes the number of text regions present in the training images.

3.4. Training Details

Figure 5 illustrates the training structure of the proposed model. The proposed model takes patch images of P&ID drawings as input and training proceeds from the detection module to the text recognition module in a single step. For patch images in the training dataset, feature maps are generated through the shared backbone, and symbols and texts are detected through the symbol-text detection module. The detection loss incurred during symbol and text detection consists of focal loss, generalized IoU loss, and L1 loss, as described in section 3.2.

The text recognition module utilizes ground-truth text regions' features extracted from the feature maps generated by the shared backbone using RoI Align. The reason why the text recognition module utilizes features of the ground-truth text regions during training is that the predicted regions cannot represent objects as precisely as the ground-truth regions, especially in the early stages of training. Therefore, when training the text recognition module using features extracted from predicted text regions, it would be influenced by noise from the detection module. To prevent this, features from ground-truth text regions are used during the training for the text recognition module. The text recognition loss, as described in section 3.3, utilizes CTC loss, and to train both the detection module and the text recognition module together, the sum of the detection loss and the CTC loss is used as the final loss. The formula for the final loss is as follows.

$$L = L_{Det} + \lambda_{rec} L_{rec} \quad (7)$$

By adjusting the constant value multiplied by the text recognition loss, we can control the training of both the detection module and the text recognition module.

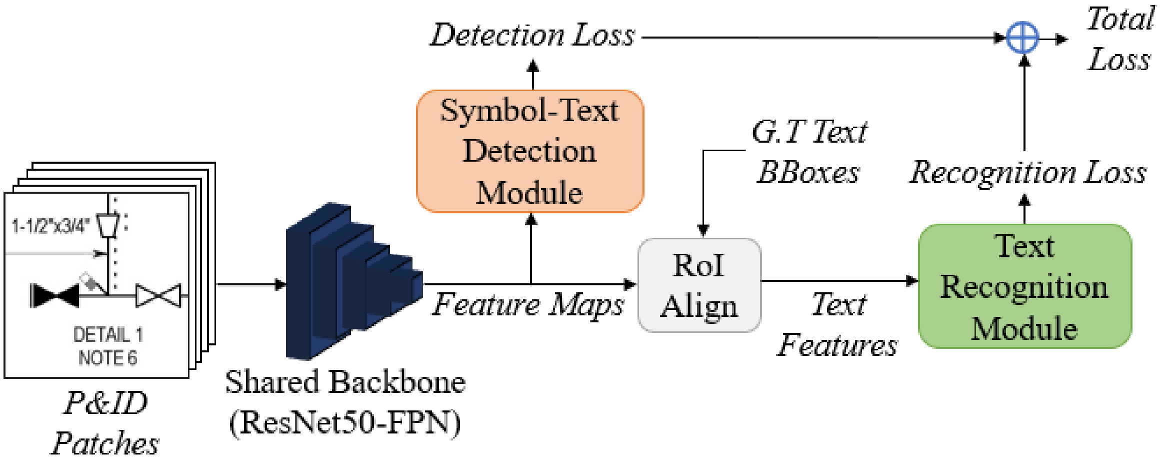


Fig. 5. Training structure of proposed method.

4. Experiments

4.1. Experimental data

This study utilized 200 P&ID drawings used in the industry. Out of 200 P&ID drawings, each 20 drawings were randomly chosen for validation and testing purposes. The overall size of the P&ID drawing is 9933x7016. The train/validation/test datasets were created by reducing the size of the entire drawings by half and then dividing them into 800x800 patch images using a sliding window approach with a stride of 300, resulting in a total of 204 patch images per drawing. For the training of the proposed model, a total of 32,640 patch images were used and the number of classes including symbols and text was 145 (144 symbols + 1 text). In the patch images, there were 174,524 symbols and 489,616 text instances. Additionally, during the training of the text recognition module of the proposed model, text images were not required since the features of the ground-truth text regions were utilized. For generating ground-truth labels for training the proposed model, the task simply involves adding the ground-truth characters to instances with text class in the detection dataset.

In the experimental phase of this study, our proposed model is compared with the process where the symbol-text detection model and the text recognition model are separated. For the process where the symbol-text detection model and the text recognition model are separated, 32,640 patch images are also used for training. However, for the training of the text recognition model, 489,616 text regions from the 32,640 patch images need to be held in image format, requiring 1.77GB of storage space. Therefore, unlike the proposed model, the process where the symbol-text detection model and the text recognition model are separated requires image cropping and storage space for training the text recognition model.

4.2. Experimental results

During the testing phase, symbol-text detection and text recognition performances are evaluated with the 20 P&ID test drawings consisting of 82 classes. For the evaluation of detection performance, precision, recall, and F1 score are used. Precision refers to the ratio of detected instances over IoU threshold among the overall detected instances, while recall signifies the ratio between detected instances over IoU threshold and the ground-truth instances. The value used for the IoU threshold was 0.5. F1 score, as the harmonic mean of precision and recall, considers both metrics together, as they are inversely related.

Since the model proposed in this study is capable of end-to-end manner from symbol-text detection to text recognition, the performance measurement of text recognition is also calculated using precision, recall, and F1 score. The method for measuring text recognition performance is as follows.

$$\text{Precision} = \frac{\#\ of\ correctly\ recognized\ texts}{\#\ of\ detected\ texts\ over\ IoU\ threshold} \quad (8)$$

$$\text{Recall} = \frac{\#\ of\ correctly\ recognized\ texts\ in\,\text{detected}\,texts\ over\ IoU\ threshold}{\#\ of\ ground{-}truth\ texts} \quad (9)$$

$$\text{F1} = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \quad (10)$$

In the measurement of text recognition performance, precision refers to the proportion of correctly recognized texts among the instances detected as text over IoU threshold. The criterion for determining if the recognized text matches the ground-truth text is WEM, where correct recognition is determined if all characters in the recognized text match those in the ground-truth text. When comparing the recognized text with the ground-truth text, no lexicon is utilized. A lexicon is a list containing ground-truth texts, and it is used to match the recognized text with the most similar text from the lexicon-defined list to compare it with the ground-truth text. Therefore, text recognition evaluation utilizing a lexicon typically yields higher performance compared to when a lexicon is not used. However, in this study, to assess how well the proposed model can recognize the text present in industrial-grade P&ID drawings, the recognized text is directly compared with the ground-truth text using the most stringent text recognition criterion, WEM, without the assistance of a lexicon. Subsequently, recall represents the ratio between correctly recognized texts in the instances detected as text over IoU threshold and the instances of ground-truth texts, and it also evaluates whether the recognized text is correct using WEM without utilizing a lexicon. The value used for the IoU threshold was also 0.5. Finally, the F1 score is employed to measure text recognition performance, ensuring that both Precision and Recall are considered together.

To evaluate the proposed model, we conducted a comparative experiment with the process where the symbol-text detection model and the text recognition model are separated. The experimental results are shown in Table 3. In the table, Ours-C and Ours-V are our proposed models respectively with CNN-based and ViT-based text recognition modules. * indicates CNN-based text recognition model and module. * indicates ViT-based text recognition model and module.

Table 3.

Model performance comparison between the proposed model and process where the symbol-text detection and the text recognition models are separated.

Method	Symbol-Text Detection			Method	Text Recognition		
	Precision	Recall	F1		Precision	Recall	F1
Sparse R-CNN	97.73	95.23	96.46	Rosetta*	88.81	84.59	86.65
				Rosetta* (aug)	90.42	86.12	88.22
				SVTR*	84.28	80.27	82.23
				SVTR* (aug)	95.18	90.66	92.87
				Tesseract	66.17	63.03	64.56
Ours-C	97.66	95.31	96.47	Ours-C*	93.26	88.93	91.04
Ours-V	97.63	95.21	96.40	Ours-V*	95.27	90.75	92.95

Ours-C is a model that integrates a CNN-based text recognition module with a symbol-text detection module. The detection performance of Ours-C was measured with a precision of 97.66%, recall of 95.31%, and F1 score of 96.47%. When compared with Sparse R-CNN, it showed similar detection performance. Subsequently, when evaluating Ours-C's recognized results of the detected text based on the WEM criteria, the precision, recall, and F1 score were measured at 93.26%, 88.93%, and 91.04%, respectively. When compared with Rosetta, a CNN-based text recognition model, Rosetta exhibited lower performance with precision, recall, and F1 score measured at 88.81%, 84.59%, and 86.65%, respectively. Rosetta (aug), as mentioned in [31], is a model trained with data augmentation, where the text images are randomly shifted during training to account for cases where the detected text regions may not align well with the ground truth regions. Data augmentation includes translation as well as color jitter, Gaussian noise, and motion blur. The text recognition performance measured on Rosetta (aug) trained with the augmented data is the best performance for Rosetta's text recognition performance improvement. However, it can still be observed that Our-C demonstrates superior text recognition performance even without data augmentation.

Ours-V is a model that integrates a ViT-based text recognition module with a symbol-text detection module. Since Ours-V utilizes part of the structure of SVTR-Tiny for its ViT-based text recognition module, the comparison target SVTR refers to SVTR-Tiny and is abbreviated as SVTR throughout. Ours-V achieved precision, recall, and F1 score of 97.63%, 95.21%, and 96.40%, respectively. When compared with Sparse R-CNN, similar performance was measured in symbol-text detection. Subsequently, when evaluating the Ours-V's recognized results of the detected texts based on the WEM criteria, precision, recall, and F1 score were measured at 95.27%, 90.75%, and 92.95%, respectively. Comparing the text recognition performance of Ours-V with SVTR, it can be observed that the F1 score is 10.72% (92.95% - 82.23%) higher. SVTR (aug) utilized data augmentation during training, including perspective distortion, motion blur, Gaussian noise, and rotation, as used in [32]. Since SVTR is a text recognition model, it only uses text images aligned perfectly with the ground truth regions for training. Therefore, translation was added to SVTR (aug) with the data augmentation methods of SVTR to account for cases where the detected text regions may not align well with the ground truth regions. The text recognition performance of SVTR (aug) trained on augmented data was measured at precision, recall, and F1 score of 95.18%, 90.66%, and 92.87%, respectively. Although there was performance improvement compared to SVTR, it did not surpass the text recognition performance of Ours-V. As an additional experiment, we applied a pretrained Tesseract based on LSTM (Long Short-Term Memory) [8], commonly used in most P&ID recognition studies, to our P&ID text recognition. The text recognition performance showed a low recognition rate with precision, recall, and F1 score of 66.17%, 63.03%, and 64.56%, respectively.

In Table 4, the size, and the inference time for text recognition of our proposed model's text recognition module and the text recognition model are recorded. In the table, Ours-C and Ours-

V are proposed models respectively with CNN-based and ViT-based text recognition modules. * indicates CNN-based text recognition model and module. * indicates ViT-based text recognition model and module.

Table 4.

Size and inference time comparison between the proposed model's text recognition module and text recognition model.

Type	Method	Params (M)	Inference Time
Model	Rosetta*	11.27	1,442.56ms
	SVTR*	4.20	1,209.93ms
	Tesseract	-	970.52s
Module	Ours-C*	9.73	247.67ms
	Ours-V*	2.47	322.56ms

When comparing the text recognition time of Ours-C and Rosetta, it can be observed that Ours-C, utilizing a smaller CNN-based text recognition module (9.73M), achieves text recognition much faster, with only 247.67ms required. Similarly, comparing Ours-V and SVTR, Ours-V, leveraging only part of SVTR's structure (2.47M), achieves much faster text recognition in 322.56ms. The reason the text recognition module can recognize text faster than the text recognition model is due to the utilization of features encoded with local information of characters extracted from feature maps generated by a shared backbone. It eliminates the need for our text recognition module to be designed as a large module for encoding local information of characters. Therefore, when integrating the symbol-text detection module and the text recognition module, a lighter-weight text recognition module can be designed compared to its original model, enabling faster text recognition. Furthermore, the LSTM-based pretrained Tesseract, commonly used as a text recognition model in most P&ID recognition studies, recognizes images one by one and operates on the CPU, resulting in a delayed text recognition speed of 970.52s.

According to the experimental results, the model integrated with the symbol-text detection and the lightweight text recognition modules demonstrates high text recognition performance even without data augmentation. It's because our proposed model is capable of end-to-end learning from symbol-text detection to the text recognition module. As detection performance is measured based on IoU, there is no difference in symbol-text detection performance. However, when visualizing the text detection and recognition results, our proposed model has three differences compared with the visualized text detection and recognition results of the process where the symbol-text detection and text recognition modules are separated. The proposed model's robustness against merge, obstacle, and omitted phenomena leads to better text recognition. The proposed model's robustness against merge, obstacle, and omitted phenomena is explained in Figure 6.

Figure 6 compares the text detection and recognition results between the proposed model and the process where the symbol-text detection module and the text recognition module are separated. The merge occurs when two texts on the same line are detected as one text due to a single-space gap between them. During end-to-end learning of our proposed model, it recognizes characters of texts, so it knows two texts on the same line are not one text with a space. Therefore, our symbol-text detection module of the proposed model can detect them in two words. Next is the obstacle issue, where non-text entities are detected alongside text, causing failure in recognition. During end-to-end learning of our proposed model, it recognizes characters of texts, so it can distinguish between foreground characters and background, allowing it to ignore non-text entities during recognition. Thus, it makes our proposed model more robust against the obstacle problem. Lastly, omitted refers to the problem of detecting part of the text, so all characters of the text can’t be recognized. However, our proposed model, having learned that the recognized characters influence detection can set the region of the detected text based on the length of the recognized characters. Therefore, our proposed model is more robust against the omitted problem by detecting without missing characters. Through Figure 6, we can confirm that our proposed models, Ours-C and Ours-V, exhibit better text detection and recognition results for the merge, obstacle, and omitted phenomenon compared to sparse R-CNN + Rosetta and sparse R-CNN + SVTR. Visualizations of Sparse R-CNN+Rosetta and Sparse R-CNN+SVTR show the results of the process when the text recognition model is not trained on augmented data. While the process where the symbol-text detection and the text recognition models are separated improves text recognition performance through data augmentation, it only addresses text recognition results, leaving issues in detection unresolved. Therefore, when converting the recognized results into digital P&ID, the process where the symbol-text detection and the text recognition models are separated needs more corrections for the wrongly detected text results than the proposed model. However, the proposed model can achieve better text detection and recognition results by training the symbol-text detection module and the lightweight text recognition module together without data augmentation.

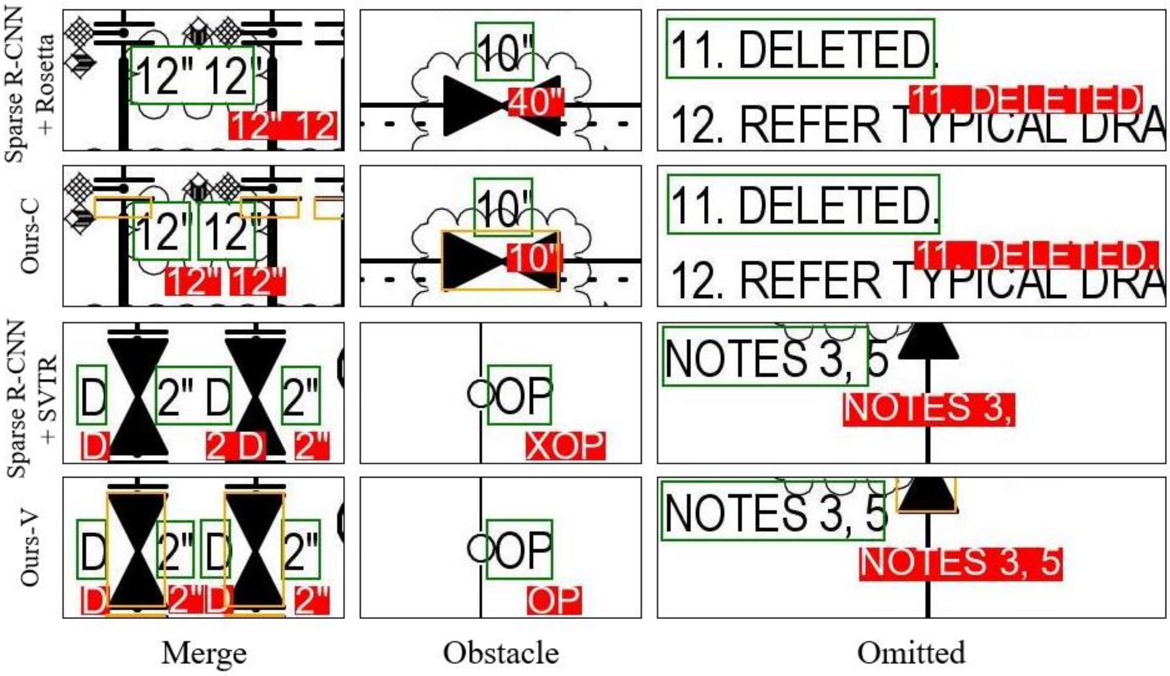
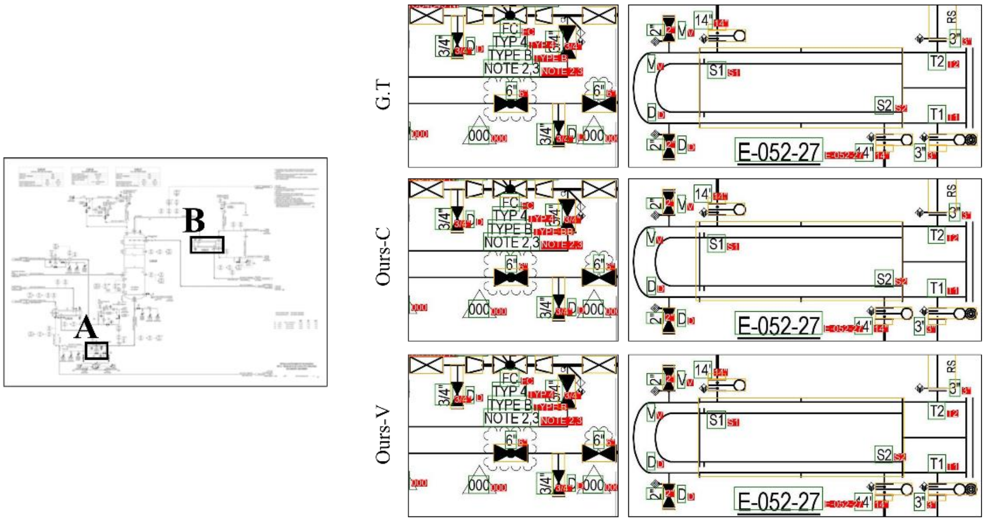
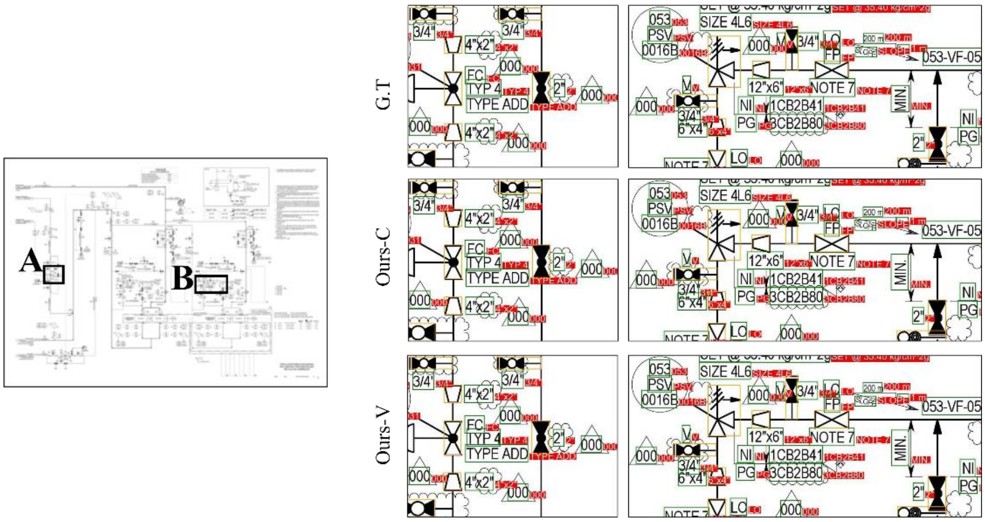


Fig. 6. Comparison of text detection and recognition results between the proposed model and the process where the symbol-text detection and the text recognition models are separated.

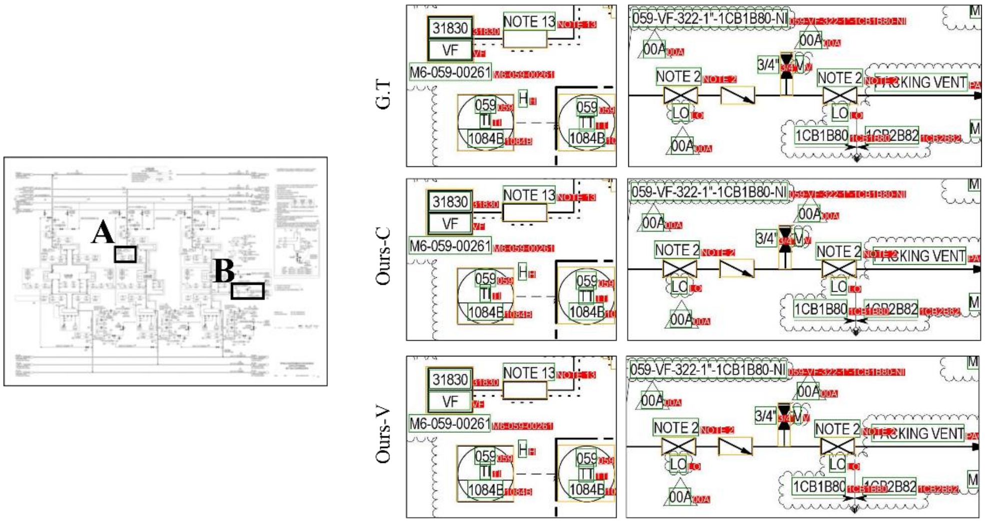
Figure 7 compares the P&ID recognition results of Ours-C and Ours-V with the ground-truth objects. The orange boxes represent symbols' bounding boxes, while the green boxes represent text bounding boxes. The red text indicates the ground-truth and recognized characters.



(a) A G.T and predictions (b) B G.T and predictions



(a) A G.T and predictions (b) B G.T and predictions



(a) A G.T and predictions (b) B G.T and predictions

Fig. 7. Ours-C and Ours-V P&ID recognition results.

4.3. Ablation study

In the ablation study, balancing the training of the symbol-text detection and the text recognition modules is conducted by increasing the training proportion of the text recognition module.

Table 5.
Ablation study of Ours-C.

λ_{rec}	Detection			End-to-End		
	Precision	Recall	F1	Precision	Recall	F1
1	**97.66**	**95.31**	**96.47**	93.26	88.93	91.04
100	97.49	94.38	95.91	95.65	**90.76**	**93.14**
200	97.24	94.09	95.64	95.58	90.20	92.81
300	97.58	94.36	95.94	95.40	90.34	92.80
400	97.58	94.42	95.97	94.53	89.65	92.03
500	97.56	94.03	95.76	**95.70**	90.54	93.05

Table 5 presents the results of balancing the training between the symbol-text detection module and the text recognition module by varying the constant value of the text recognition module's loss in Ours-C. When the constant value is fixed at 1, the detection performance is measured to be the highest based on the F1 score. Regarding text recognition performance, when the constant value is 500, the precision is measured to be the highest at 95.70%, while at 100, recall and F1 score are measured to be the highest. Comparing between the constant value of 100 and 1, it can be observed that the text recognition performance improved by 2.1% (93.14% - 91.04%) based on the F1 score. The other constant values also showed higher text recognition performance than 1. Through the experiment, it was confirmed that the training between the two modules can be balanced.

Table 6.
Ablation study of Ours-V.

λ_{rec}	Detection			End-to-End		
	Precision	Recall	F1	Precision	Recall	F1

1	**97.63**	**95.21**	**96.40**	95.27	90.75	92.95
100	97.37	94.49	95.91	**96.56**	**91.67**	**94.05**
200	97.33	94.54	95.91	96.24	91.51	93.82
300	97.57	94.66	96.09	96.11	91.34	93.66
400	97.58	94.42	95.97	94.53	89.65	92.03
500	97.46	94.17	95.79	96.16	91.20	93.62

Table 6 shows the results of balancing the training between the symbol-text detection module and the text recognition module in Ours-V by varying the constant value of the text recognition module's loss. While there is a slight decrease in detection performance compared between constant values of 1 and 100, the text recognition performance is increased by 1.1% (94.05% - 92.95%) which is the most significant margin based on the F1 score. Except for 400, for the remaining constant values, although symbol-text detection performances decrease compared to 1, it can be observed that text recognition performances improve.

5. Conclusions and future work

We proposed a new model that integrates a single symbol-text detection module and a text recognition module, enabling the recognition of symbols and text in P&ID with one model. By applying the text spotting method, which utilizes text features for recognition, we were able to integrate the symbol-text detection module and the text recognition module. The proposed model facilitates end-to-end learning from the symbol-text detection module to the text recognition module, so it can get better text detection and recognition results than the process where the symbol-text detection and text recognition modules are separated. Additionally, during the training of the proposed model, there is no need to generate and store text images for training, as the text recognition module utilizes text features. Furthermore, since detected text regions' features extracted from the feature maps generated by the shared backbone are encoded with local information of characters, it is possible to achieve text recognition with a lightweight text recognition module. To assess the practical applicability of the proposed model, we measured its performance on P&ID test drawings used in actual industries. The symbol-text detection performance achieved a maximum precision of 97.63%, recall of 95.21%, and F1 score of 96.40%. For text recognition performance, without using a lexicon, the model achieved a maximum precision of 95.27%, recall of 90.75%, and F1 score of 92.95%. The experimental results demonstrate that the proposed model can achieve high-performance symbol-text detection and text recognition even without data augmentation. Additionally, through experimentation, we confirmed that the training of the symbol-text detection module and the text recognition module of the proposed model can be balanced.

In P&ID drawings, there are numerous horizontal symbols and texts. Therefore, this study aimed to simplify the complex process of recognizing symbols and texts by integrating a single symbol-text detection module and a text recognition module into one model using the text spotting method, with the goal of accurately recognizing horizontal symbols and texts. Therefore, the rotation of oriented symbols and texts with relatively fewer occurrences was not considered. Thus, to get a more accurate representation of the oriented symbols and text regions and improve oriented text recognition performance, future work will focus on an integrated model with symbol-text detection and text recognition modules capable of recognizing the rotation information of symbols and texts.

References

[1] Kim, H., Lee, W., Kim, M., Moon, Y., Lee, T., Cho, M., & Mun, D., 2021. Deep-learning-based recognition of symbols and texts at an industrially applicable level from images of high-density piping and instrumentation diagrams. Expert Systems with Applications, 183, 115337. https://doi.org/10.1016/j.eswa.2021.115337

[2] Rahul, R., Paliwal, S., Sharma, M., & Vig, L., 2019. Automatic information extraction from piping and instrumentation diagrams. In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods. https://doi.org/10.5220/0007376401630172

[3] Mani, S., Haddad, M. A., Constantini, D., Douhard, W., Li, Q., & Poirier, L., 2020. Automatic digitization of engineering diagrams using deep learning and graph search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp.176-177). https://doi.org/10.1109/CVPRW50498.2020.00096

[4] Kang, S. O., Lee, E. B., & Baek, H. K., 2019. A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (P&ID). Energies, 12(13), 2593. https://doi.org/10.3390/en12132593

[5] Kim, B. C., Kim, H., Moon, Y., Lee, G., & Mun, D., 2022. End-to-end digitization of image format piping and instrumentation diagrams at an industrially applicable level. Journal of Computational Design and Engineering. https://doi.org/10.1093/jcde/qwac056

[6] Wang, Z., Cai, Z., & Wu, Y., 2023. An improved YOLOX approach for low-light and small object detection: PPE on tunnel construction sites. Journal of Computational Design and Engineering, 10(3), 1158-1175. https://doi.org/10.1093/jcde/qwad042

[7] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J., 2017. East: an efficient and accurate scene text detector. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp.5551-5560). https://doi.org/10.48550/arXiv.1704.03155

[8] Hochreiter, S., & Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[9] Smith, R., 2007. An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp.629-633). IEEE. https://doi.org/10.1109/ICDAR.2007.4376991

[10] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., & Yang, J., 2020. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems, 33, 21002-21012. https://doi.org/10.48550/arXiv.2006.04388

[11] Long, J., Shelhamer, E., & Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp.3431-3440). https://doi.org/10.48550/arXiv.1411.4038

[12] Tian, Z., Huang, W., He, T., He, P., & Qiao, Y., 2016. Detecting text in natural image with connectionist text proposal network. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14 (pp.56-72). Springer International Publishing. https://doi.org/10.1007/978-3-319-46484-8_4

[13] Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H., 2019. Character region awareness for text detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.9365-9374). https://doi.org/10.1109/CVPR.2019.00959

[14] Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25. https://doi.org/10.1145/3065386

[15] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P., 2017. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp.2980-2988). https://doi.org/10.48550/arXiv.1708.02002

[16] Zhang, D., Hao, X., Liang, L., Liu, W., & Qin, C., 2022. A novel deep convolutional neural network algorithm for surface defect detection. Journal of Computational Design and Engineering, 9(5), 1616-1632. https://doi.org/10.1093/jcde/qwac071

[17] Chen, Z., Yang, C., Li, Q., Zhao, F., Zha, Z. J., & Wu, F., 2021. Disentangle your dense object detector. In Proceedings of the 29th ACM international conference on multimedia (pp.4939-4948). https://doi.org/10.1145/3474085.3475351

[18] Ren, S., He, K., Girshick, R., & Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28. https://doi.org/10.48550/arXiv.1506.01497

[19] Cai, Z., & Vasconcelos, N., 2018. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp.6154-6162). https://doi.org/10.48550/arXiv.1712.00726

[20] Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., & Luo, P., 2021. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.14454-14463). https://doi.org/10.48550/arXiv.2011.12450

[21] Law, H., & Deng, J., 2018. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European conference on computer vision (ECCV) (pp.734-750). https://doi.org/10.48550/arXiv.1808.01244

[22] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q., 2019. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF international conference on computer vision (pp.6569-6578). https://doi.org/10.48550/arXiv.1904.08189

[23] Tian, Z., Shen, C., Chen, H., & He, T., 2020. FCOS: A simple and strong anchor-free object detector. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(4), 1922-1933. https://doi.org/10.1109/TPAMI.2020.3032166

[24] Zhang, H., Wang, Y., Dayoub, F., & Sunderhauf, N., 2021. Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.8514-8523). https://doi.org/10.48550/arXiv.2008.13367

[25] He, K., Zhang, X., Ren, S., & Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp.770-778). https://doi.org/10.1109/CVPR.2016.90

[26] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S., 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp.2117-2125). https://doi.org/10.48550/arXiv.1612.03144

[27] Zhang, S., Chi, C., Yao, Y., Lei, Z., & Li, S. Z., 2020. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.9759-9768). https://doi.org/10.48550/arXiv.1912.02424

[28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30. https://doi.org/10.48550/arXiv.1706.03762

[29] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S., 2020. End-to-end object detection with transformers. In European conference on computer vision (pp.213-229). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_13

[30] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J., 2020. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159. https://doi.org/10.48550/arXiv.2010.04159

[31] Borisyuk, F., Gordo, A., & Sivakumar, V., 2018. Rosetta: Large scale system for text detection and recognition in images. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (pp.71-79). https://doi.org/10.1145/3219819.3219861

[32] Du, Y., Chen, Z., Jia, C., Yin, X., Zheng, T., Li, C., Du, Y., & Jiang, Y. G., 2022. Svtr: Scene text recognition with a single visual model. arXiv preprint arXiv:2205.00159. https://doi.org/10.48550/arXiv.2205.00159

[33] Choi, M., Kim, C., & Oh, H., 2022. A video-based SlowFastMTB model for detection of small amounts of smoke from incipient forest fires. Journal of Computational Design and Engineering, 9(2), 793-804. https://doi.org/10.1093/jcde/qwac027

[34] Shi, B., Bai, X., & Yao, C., 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence. https://doi.org/10.1109/TPAMI.2016.2646371

[35] Qiao, Z., Zhou, Y., Yang, D., Zhou, Y., & Wang, W., 2020. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.13528-13537). https://doi.org/10.48550/arXiv.2005.10977

[36] Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., & Bai, X., 2018. Aster: An attentional scene text recognizer with flexible rectification. IEEE transactions on pattern analysis and machine intelligence, 41(9), 2035-2048. https://doi.org/10.1109/TPAMI.2018.2848939

[37] Fang, S., Xie, H., Wang, Y., Mao, Z., & Zhang, Y., 2021. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7098-7107). https://doi.org/10.48550/arXiv.2103.06495

[38] Yu, D., Li, X., Zhang, C., Liu, T., Han, J., Liu, J., & Ding, E., 2020. Towards accurate scene text recognition with semantic reasoning networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.12113-12122). https://doi.org/10.48550/arXiv.2003.12294

[39] Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W., 2017. Textboxes: A fast text detector with a single deep neural network. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1). https://doi.org/10.1609/aaai.v31i1.11196

[40] Liao, M., Shi, B., & Bai, X., 2018. Textboxes++: A single-shot oriented scene text detector. IEEE transactions on image processing, 27(8), 3676-3690. https://doi.org/10.1109/TIP.2018.2825107

[41] Li, H., Wang, P., & Shen, C., 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. In Proceedings of the IEEE international conference on computer vision (pp.5238-5246). https://doi.org/10.48550/arXiv.1707.03985

[42] He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., & Sun, C., 2018. An end-to-end textspotter with explicit alignment and attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp.5020-5029). https://doi.org/10.48550/arXiv.1803.03474

[43] He, K., Gkioxari, G., Dollár, P., & Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp.2961-2969). https://doi.org/10.48550/arXiv.1703.06870

[44] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J., 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning (pp.369-376). https://doi.org/10.1145/1143844.1143891

[45] Liu, S., Huang, D., & Wang, Y., 2019. Adaptive nms: Refining pedestrian detection in a crowd. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.6459-6468). https://doi.org/10.1109/CVPR.2019.00662

[46] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.658-666). https://doi.org/10.1109/CVPR.2019.00075

[47] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q., & Adam, H., 2019. Searching for mobilenetv3. In Proceedings of the IEEE/CVF international conference on computer vision (pp.1314-1324). https://doi.org/10.1109/ICCV.2019.00140

[48] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. https://doi.org/10.48550/arXiv.2010.11929

[49] Lee, W., Kim, M., Mun, D., & Kim, H., 2021. Image Format P&ID Recognition Technique Using Synthetic Data and Text-symbol Integrated Detection. Korean Journal of Computational Design and Engineering, 26(4), 355-365. http://dx.doi.org/10.7315/CDE.2021.355

Conflict of Interest

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Duhwan Mun reports financial support was provided by Korean government (MSIT). Duhwan Mun reports financial support was provided by Korean government (MOLIT). Hyungki Kim reports a relationship with National Research Foundation of Korea that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.