Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale, Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

A Novel Symbol Recognition Framework for Digitization of Piping and Instrumentation Diagrams

Zhiyuan Li[a,b], Jinsong Zhao[a,b]*, Huahui Zhou[c], and Xiaoxin Hu[c]

[a] Tsinghua University, Department of Chemical Engineering, Beijing, China
[b] State Key Laboratory of Chemical Engineering, Beijing, China
[c] Sinopec Ningbo Engineering Co., Ltd, Ningbo, China
* Corresponding Author: jinsongzhao@tsinghua.edu.cn.

ABSTRACT

Piping and Instrumentation Diagrams (P&IDs) play a crucial role in the chemical industry, yet they are often stored as scanned images or computer-aided design (CAD) drawings, limiting their seamless integration into modern digital workflows. Consequently, the task of automating P&ID digitization has attracted significant attention from the computer-aided design (CAD) research community. Traditional approaches, which typically rely on conventional object detection techniques, often demand extensive manual annotations to accurately identify and classify the various symbols in P&IDs. In this paper, we proposed a novel framework for automating the recognition of P&IDs. Our method first extracts key features of geometric primitives in CAD drawings through an automated process. Subsequently, a Transformer-based model is employed to predict the layer assignment of these primitives. Following this, an unsupervised clustering method, guided by predefined rules and geometric distances, is applied within each layer type to facilitate the automatic identification of devices, pipes, instrumentation, valves, insulation, and utilities in P&IDs. Extensive evaluation on an annotated dataset comprising 336 authentic P&IDs demonstrates that the proposed semantic annotation method achieves an overall accuracy exceeding 92%. The results of the unsupervised clustering are visualized, demonstrating promising performance and supporting the effectiveness of the proposed method. This study provides valuable insights into the intelligent digitization of P&IDs and lays the groundwork for further advancements in intelligent chemical process design.

Keywords: Intelligent Systems, Piping and Instrumentation Diagram, Computer Aided Design

1 INTRODUCTION

Piping and Instrumentation Diagrams (P&IDs) play a central role at every stage of chemical engineering design. They serve as an essential tool for communication among engineers, operators, and maintenance personnel by depicting how piping systems connect with industrial processing equipment and how processes are controlled through automated systems. By indicating critical information such as isolation valves and emergency shutdown devices, P&IDs aid operators in executing safe operations and responding effectively in emergencies.

Over the years, millions of P&IDs have been manually generated, scanned, and stored in image formats, encapsulating a wealth of valuable engineering design knowledge [1]. However, analyzing the complex information within these diagrams—due to the intricacy of devices, text, and pipelines—has traditionally required extensive manual effort by engineers. This manual analysis is not only time-consuming but also prone to errors. Therefore, the ability to automatically extract information from P&IDs would significantly enhance productivity and minimize human error. Moreover, since P&IDs are a primary source of design and safety information in chemical engineering, the extracted data can be integrated with intelligent algorithms for process simulation, optimization, and safety analysis, promoting more effective chemical process management and design.

To support digital storage and interoperability, the Digital Exchange of Process and Instrumentation Diagrams (DEXPI) standard [2] was introduced. By defining a comprehensive set of structured data formats based on eXtensible Markup Language (XML), DEXPI enhances the efficiency and accuracy of engineering processes. In recent years, numerous deep learning-based image recognition algorithms have been developed to convert image-format P&IDs into drawings that comply with the DEXPI standard. For example, Kim et al. proposed a P&ID digitization system that employs two independent Generative Flow Networks (GFL) for recognizing equipment symbols of varying sizes [3]. Despite advances in deep learning-based image recognition, digitization of P&IDs remains challenging due to the high irregularity of ordering and uncertain orientations [4].

Transitioning from these challenges, most traditional P&ID files are created using computer-aided design (CAD) tools, which means that beyond the visible image data, these drawings inherently contain detailed information about the shapes, positions, and layer properties of various geometric primitives (such as lines, arcs, polygons). This fine-grained semantic data can greatly enhance the effectiveness of P&ID recognition by addressing the limitations of traditional methods that rely solely on images. Building on this insight, in this paper we propose a novel intelligent symbol recognition framework that tokenizes various geometric primitives found in CAD drawings, then a Transformer-based model is employed to automatically annotate the layer assignment of these primitives. After that, we apply unsupervised clustering to group the isolated primitives into meaningful symbols (including devices, valves, etc.). Validation on 336 authentic P&IDs demonstrates that our framework achieves an overall accuracy of 0.927 in annotating geometric primitives. The visualization of the clustering results further demonstrates the effectiveness of our approach.

2 METHODOLOGY

2.1 Data Preprocessing

This paper primarily uses two data sources: CAD drawings and rasterized images. The CAD drawings are in Drawing Exchange Format (DXF), exported from AutoCAD. DXF files employ a vector-based data format, providing detailed attribute information for various geometric primitives and defining the blocks and layers within the drawing. The rasterized images are in Portable Network Graphics (PNG) format, converted from the DXF files. Thus, there is a one-to-one correspondence between the CAD drawings and the rasterized images.

We use the Python library ezdxf to extract the types and layer information of geometric primitives from CAD drawings. The layer information for these geometric primitives is manually created by the designer and typically conveys the specific meaning of the primitives in the P&ID. Therefore, we use the layer information as the semantic label for the primitives. In P&ID drawings, the layers are primarily categorized into 11 distinct categories, the details of which are shown in Table 1.

Next, we utilize the Shapely Python library to convert various types of geometric primitives from the DXF file into Shapely geometric objects. Specifically, straight lines and open polylines are directly converted into LineString objects, while closed polylines are transformed into Polygon objects. For curves in the DXF file, we approximate them with a polyline consisting of 72 endpoints, converting them into either LineString or Polygon objects based on whether they are closed. Subsequently, we employ functions from Shapely to calculate the centroids of the geometric primitives and the minimum distances between each pair of primitives. For each primitive, we identify the 128 nearest geometric primitives as its neighboring nodes and construct the adjacency matrix.

Table 1. Common layer definitions in P&IDs.

Layer name	Definition
Device	Contour of Physical devices used in the system, such as pumps
Instrument	Symbol of measuring and controlling instruments
Valve	Symbol of valves in the system
In-out Signal	Communication signals for system input and output
Pipeline	Symbol of the pipework that are used to transport process materials
Control Line	Symbol of the control lines that are used to carry signals
Insulation	Symbol of the insulation around piping or equipment
Hot Utility	Symbol of systems that provide heat
Cold Utility	Symbol of systems that provide cooling
Arrow	Symbol of the flow of materials

2.2 Model for Layer Assignment Prediction

2.2.1 Backbone Architecture

The backbone architecture of the predictive model is illustrated in Figure 1. PNG images derived from CAD drawings are processed using the pre-trained HRNetV2-W48 model [5] to automatically extract features. Additionally, the previously discussed feature extraction method is used to extract the normalized center coordinates of geometric primitives from the CAD drawings and to construct an adjacency matrix.

Using these normalized coordinates, we project each geometric primitive from the CAD drawing to its corresponding position in the PNG image. We then apply bilinear interpolation-based upsampling to align the feature maps with the positions of the geometric primitives, calculating the feature vectors at each primitive's location. These interpolated feature vectors undergo further linear transformations and ReLU operations to generate the

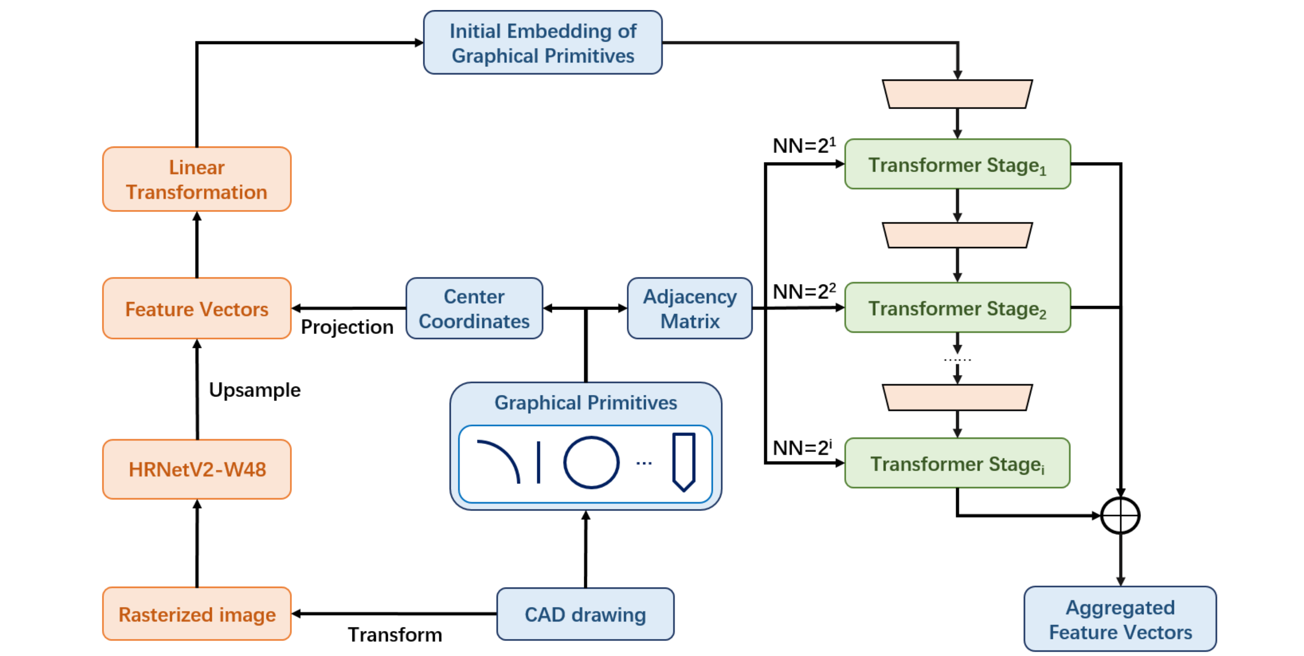


Figure 1. Backbone architecture for predicting layer assignment of the geometry primitives.

initial embedding for each geometric primitive. The updated vectors are subsequently processed using a Transformer model that consists of various stages, each stage applies self-attention with spatially localized k-nearest neighbors, enabling the refinement of each primitive's representation using its local features. The features across different stages are then fused by calculating the mean of the output of the Transformer stages.

2.2.2 Layer Assignment Prediction Task

The updated vector representations of geometric primitives are then processed using a multi-layer perceptron (MLP) to predict the layer to which each primitive belongs. The MLP model outputs a predicted score for each geometric primitive corresponding to each layer, which is then compared to the actual layer properties of the primitives. The loss function employed is based on the cross-entropy loss function:

$$L_{sem} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{N} y_{i,m} log(p_{i,m}) \quad (1)$$

where M is the total number of geometric primitives, and N is the number of layer categories, $y_{i,m}$ represents whether the m-th geometric primitive actually belongs to the i-th layer; it is equal to 1 if it does, and 0 if it does not. Additionally, p_i represents the predicted score for the model's prediction of the geometric primitive belonging to the i-th layer. In order to overcome the impact of class imbalance, we choose AM-Softmax loss function [6], which introduces an additive margin to the target class's logit. During the inference process, the predicted semantic label is the layer with the highest predicted probability.

2.3 Clustering for Symbol Recognition

After assigning layer for the geometric primitives, in order to extract higher-level features, such as the contours of devices, and to better understand the physical process layout, we perform unsupervised clustering for each layer to automatically group isolated geometric primitives into meaningful symbols.

For instances with closed contours, such as equipment, instruments, valves, insulation, and inlet/outlet labels, we apply a Depth First Search (DFS) algorithm to cluster all geometric primitives within the same layer. We assume that if the distance between two geometric primitives is less than the specified threshold, they are considered to intersect. By utilizing a stack to simulate the recursive depth-first search, we group all intersecting geometric primitives into the same cluster, while non-intersecting ones are assigned to separate clusters.

For instances without closed contours, such as pipelines, control lines, cold utilities, and hot utilities, we distinguish instances based on predefined rules. For example, when two perpendicular pipelines intersect, it typically signifies that a main pipeline is branching into two directions. In such cases, the algorithm generates three clusters: one for the main pipeline and two for the branch pipelines. Conversely, if two perpendicular pipelines cross without an intersection point, it suggests that the two pipelines are physically offset. In this scenario, one pipeline may be represented as two straight lines in the drawing. The algorithm identifies collinear segments with distances that are consistent with common knowledge and merges them into a single cluster.

3 EXPERIMENTS

3.1 Datasets and Evaluation Metrics

The dataset utilized in this study consists of 336 real P&ID drawings stored in DXF format, each labeled with its corresponding layer assignment. These drawings represent 17 process flows designed for the deep desulfurization of industrial flue gas, providing detailed information about the piping systems, instruments, valves, process topology, and utilities. Although the original versions of these drawings include layer annotations, the naming conventions for the layers are inconsistent. Consequently, we standardized the layer names across the drawings into 11 categories, as detailed in Table 1. Additionally, we converted the drawings from DXF format to PNG format. Prior to being input into the model, each PNG image was cropped into square blocks, each measuring 14m × 14m. The training, validation, and test sets were divided in an 8:1:1 ratio, resulting in 268, 33, and 35 P&ID drawings, respectively.

The Transformer-based layer prediction model we employed has an input dimension of 384 and consists of four layers of self-attention. The number of neighboring nodes utilized by each self-attention layer increases from 2 in the first layer to 4 in the second, 8 in the third, and 16 in the fourth layer. The output dimension is 11, corresponding to the 11 layer categories. During training, the initial learning rate is set to 0.00025, and the Adam optimizer is used with β_1 = 0.9 and β_2 = 0.999. Every 20 epochs, the learning rate is reduced to 50% of its previous value. The original momentum is set to 0.1, with momentum decay set to 0.5. The model is trained for 300 epochs. The batch size is fixed to 6, and we trained our model with 6 NVIDIA RTX 4090 GPUs with 1 training sample on each GPU.

We evaluated the effectiveness of the proposed model using the following classic classification metrics:

- Precision: The proportion of correctly predicted instances of a specific layer among all instances predicted as that layer.
- Recall: The proportion of correctly predicted instances of a specific layer among all instances that actually belong to that layer.
- F1 Score: The harmonic mean of precision and recall, considering the balance between these two metrics.
- Accuracy: The proportion of correctly predicted samples out of the total number of samples.

Additionally, we employed a confusion matrix to analyze which geometric elements of the layers are more likely to be mistaken for one another.

3.2 Results on Layer Assignment Prediction

The quantitative evaluation results on the test set are presented in Table 2. Figure 2 shows the detailed prediction results for each class in the form of a confusion matrix. The diagonal elements of the matrix represent the number of correctly classified samples, while the off-diagonal elements indicate misclassifications, reflecting the model's ability to distinguish between different categories. It can be seen that our model performs well in overall layer classification for all geometric primitives, achieving an F_1 score of 0.9224, with precision of 0.925, recall of 0.9224, and accuracy of 92.7%.

Table 2. Evaluation results of the layer prediction model.

Layer name	F_1	Precision	Recall
Device	0.9249	0.9051	0.9457
Instrument	0.9643	0.9611	0.9676
Valve	0.9608	0.9606	0.9611
In-out Signal	0.9845	0.9772	0.9919
Pipeline	0.8693	0.8487	0.8910
Control Line	0.7844	0.7977	0.7716
Insulation	0.9349	0.9496	0.9207
Arrow	0.9290	0.9290	0.9290
Hot Utility	0.3042	0.5192	0.2151
Cold Utility	0.6973	0.7765	0.6328
Total	0.9244	0.9250	0.9224

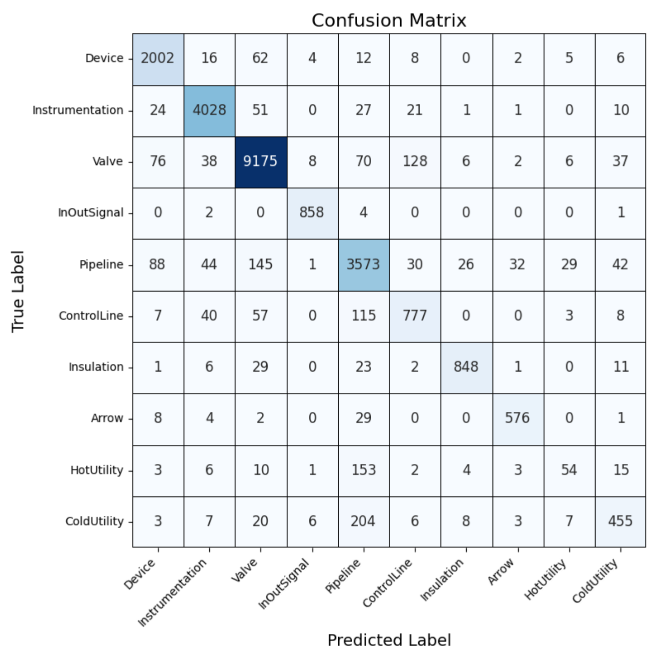


Figure 2. Confusion Matrix of Classification Results.

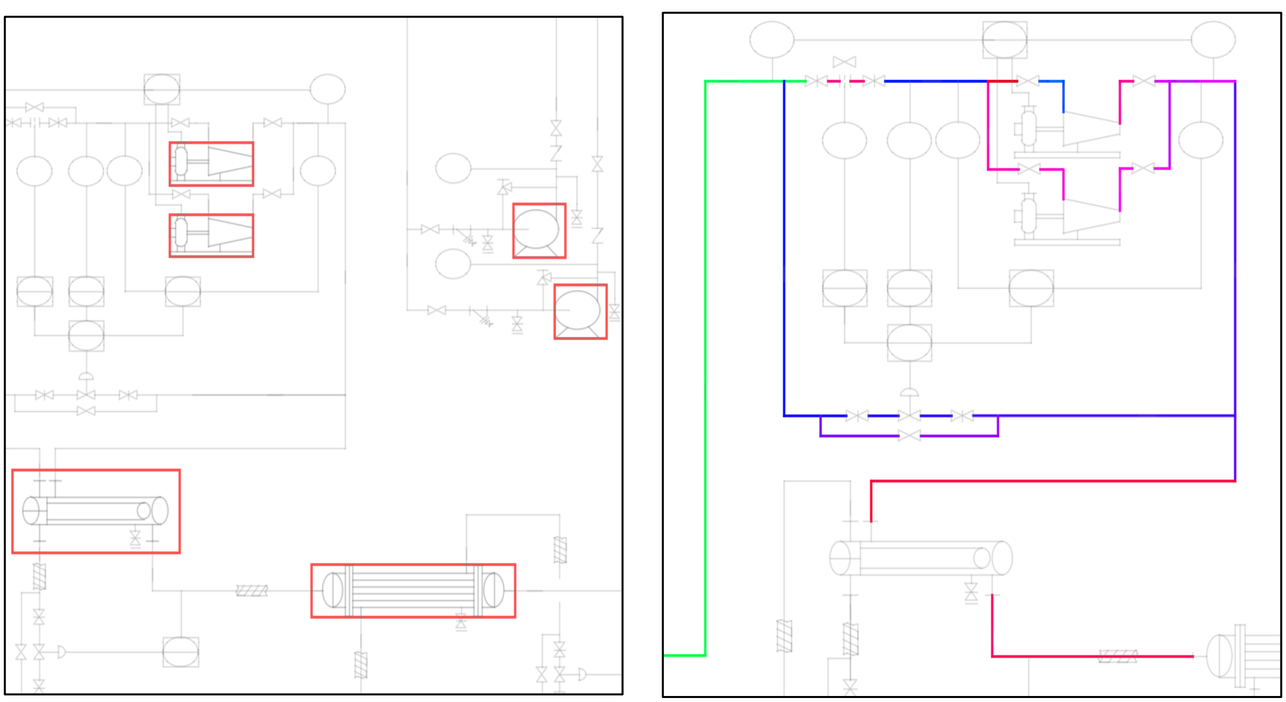
Among the geometric primitives in each layer, the model demonstrates high accuracy in recognizing devices, instruments, valves, inlet/outlet markers, insulation layers, and arrows, with F1 scores exceeding 0.9. We hypothesize that this is due to the relatively standardized shapes of the symbols in these layers and the large number of training samples available.

In contrast, there is room for improvement in the recognition of pipes, control lines, and utilities. This is likely because these layers consist mainly of multiple line segments, which can form highly variable combinations. As indicated by the confusion matrix, some pipelines are misclassified as equipment outlines or valves. Furthermore, the geometric primitives in the control lines, cold utilities, and hot utilities layers are difficult to distinguish from those in the pipeline layer. We believe this is not only because these layers contain fewer geometric primitives than the pipeline layer, but also because they cannot be easily differentiated based solely on the positional information of the geometric primitives. Specifically, the key distinction between control lines and pipes is whether both endpoints of the line segments (or polylines) are connected to control instruments. Similarly, the fundamental criterion for distinguishing between utilities and process material pipelines is whether the transported substance is process material, which requires additional textual information, such as piping codes, from the P&IDs.

3.3 Visualization of Clustering Results

The clustering results of the equipment, and pipelines layers are shown in Figure 3, which not only allows us to observe the performance of target recognition for each layer, but also helps assess the clustering precision of different layers. The clustering outcomes for the equipment layers are illustrated in Figures 3(a), respectively. Due to variations in size and distribution density between the equipment and other identified objects, we applied distinct distance thresholds for each layer. For the equipment layer, a larger threshold of 6 units effectively groups adjacent primitives together. While for all other layers, a smaller threshold of 0.001 units ensures more precise clustering by grouping close geometric primitives together, minimizing redundancy and incorrect clustering. From the clustering visualization, we can see that the contours of most equipment and instruments, including instrument interface pipes, are accurately recognized and clearly distinguishable. This demonstrates the strong potential of using a DFS algorithm for closed-contour target recognition.

The clustering results for the pipelines layers are presented in Figures 3(b), respectively. It can be observed that the various pipelines were effectively distinguished according to the established rules. However, some pipelines were not successfully recognized by the layer assignment predictive model, and as a result, their clustering results were not reflected in the drawing. What's more, since the current rule definitions are relatively simplistic, certain specific scenarios, such as the presence of bypass valves on pipelines have not been adequately addressed. In these cases, the bypass pipe should merge with the main pipeline. Consequently, the



(a) (b)

Figure 3. (a) Clustering visualization for the equipment layers; (b) Clustering visualization for the pipelines layers.

redundancy in the pipeline clustering results can be reduced by refining the rules.

4 CONCLUSION

In this study, we propose a novel intelligent symbol recognition framework for P&ID based on CAD drawings. During the data preprocessing stage, CAD drawings are divided into fine-grained geometric primitives, and an adjacency matrix is constructed based on the distances between these primitives. The symbol recognition framework consists of two stages: layer assignment prediction and symbol recognition. In the layer assignment prediction stage, the classic image recognition framework HRNetV2-W48 is employed to generate feature vectors from the converted PNG images of the CAD drawings, enabling the positional encoding of each geometric primitive. Then, a Transformer-based architecture is utilized to update the encoding of each geometric primitive by leveraging information from the nearest neighbors in the spatial domain. Finally, the membership probability of each geometric primitive to different layers is determined through a linear transformation. In the symbol recognition stage, the symbols with closed contours and open contours are recognized by unsupervised clustering using depth-first search and predefined rule, respectively.

Experimental results showed that the model for layer assignment prediction achieved a multi-class accuracy of 92.7% and an F_1 score of 0.9224 on the test set. This indicates the potential of using the Transformer architecture for layer assignment prediction tasks. Unlike traditional object detection frameworks, which perform regression to detect bounding boxes and classify targets within the boxes, this study directly utilizes fine-grained geometric primitive from CAD drawings. It extracts essential geometric features and spatial topology, further refining the embedding representation of geometric primitives using the self-attention mechanism. This approach allows for improved handling of diverse combinations and randomly orientation of the symbols. The effectiveness of unsupervised clustering is demonstrated through visualization, where symbols are accurately clustered according to predefined rules and geometric distances.

Although the experimental results are generally satisfactory, several areas require further improvement. Firstly, the current classification model tends to confuse geometric primitives of pipes, control lines, and utilities across different layers, and some pipes are misidentified as equipment contours or valve components. Secondly, the clustering of pipes, control lines, and utilities may result in redundancy. Finally, the extraction of grounding lines and textual information, and their interrelationships were not addressed in this study. In future work, we plan to develop a model to establish the matching relationships among equipment, instruments, and textual descriptions, thereby automatically forming a process topology network with node descriptions. We believe this study will provide a novel approach to the digitalization of P&ID and contribute to the advancement of intelligent process design, risk analysis, and large language models in the field of chemical engineering.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge support from the National Key R&D Program of China (2024YFC3013600).

REFERENCES

1. Paliwal, S., Jain, A., Sharma, M., & Vig, L. (2021). Digitize-PID: Automatic digitization of piping and instrumentation diagrams. In Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings 25 (pp. 168-180). Springer International Publishing.
2. Data Exchange in the Process Industry. https://dexpi.org/static/pid_specification_1.4/
3. 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, 9(4), 1298-1326.
4. Fan, Z., Chen, T., Wang, P., & Wang, Z. (2022). Cadtransformer: Panoptic symbol spotting transformer for cad drawings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10986-10996).
5. Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5693-5703).
6. Wang, F., Cheng, J., Liu, W., & Liu, H. (2018). Additive margin softmax for face verification. IEEE Signal Processing Letters, 25(7), 926-930.