# Integrating deep learning-based object detection and optical character recognition for automatic extraction of link information from piping and instrumentation diagrams

| | | | |
|---|---|---|---|
| 東京大学 | 董　飛艶 | Feiyan　DONG | Student-member |
| 東京大学 | 陳　実 | Shi　CHEN | Member |
| 東京大学 | 出町　和之 | Kazuyuki　DEMACHI | Member |
| 日本原子力研究開発機構 | 橋立　竜太 | Ryuta　HASHIDATE | Non-member |
| 日本原子力研究開発機構 | 高屋　茂 | Shigeru　TAKAYA | Member |

Piping and Instrumentation Diagrams (P&IDs) contain information about the piping and process equipment together with the instrumentation and control devices, which is essential to the design and management of Nuclear Power Plants (NPPs). There are abundant complex objects on P&IDs, with imbalanced distribution of these objects and their linked information across different diagrams. The complexity of P&IDs thus is increased which make automatic identification difficult. Therefore, the content of P&IDs is generally extracted and analyzed manually, which is time consuming and error prone. To efficiently address these issues, we integrate state-of-the-art deep learning-based object detection and Optical Character Recognition (OCR) models to automatically extract link information from P&IDs. Besides, we propose a novel image pre-processing approach using sliding windows to detect low resolution small objects. The performance of the proposed approach was experimentally evaluated, and the experimental results demonstrate it capable to extract link information from P&IDs of NPPs

**Keywords**: Piping and Instrumentation Diagrams (P&IDs), Nuclear Power Plants (NPPs), Object Detection, Deep Learning, Optical Character Recognition (OCR)

## 1. Introduction

In nuclear power plants (NPPs), piping and instrumentation diagrams (P&IDs) are commonly used as key components to depict the process equipment and their control devices. Typically, it is illustrated as symbols and accordingly text information, representing layout and connection relationship for every instrument via P&IDs. The extraction of such information is the basis for safety analysis as well as Operation and Maintenance (O&M) of NPPs. However, the complexity of P&IDs is naturally increased by an abundance of symbols on each entity. What is worse, the text information of each equipment, such as, the location and control device, is presented many times more than symbols, caused the imbalanced distribution. Both pose obstacles in effective recognition. Furthermore, the paucity of sufficient real datasets for this domain makes it more difficult to implement an automatic extraction on P&IDs. For decades, this extraction process is implemented and analyzed manually due to the complexity, assessed as time-consuming and error-prone tasks. Hence, it would be of the value if the content of each P&ID can be extracted automatically.

To alleviate above issues, as an advanced application of object detection in P&IDs, we integrate a state-of-the art deep learning-based object detection model, YOLOv4, to the Optical character recognition (OCR). Additionally, it is proposed an adaptive data pre-process approach to augment the datasets for training and testing. By doing those, it can be achieved the main objective that extract the information from P&IDs automatically.

The remainder of this paper is organized as follows: Section 2 briefly states the related work in extraction the information from P&IDs. Section3 provides the overview and theoretical details of proposed algorithms. Based on the proposal, the experiment implement details and results are discussed in Section 4. Section 5 concludes this paper and discusses the future work.

## 2. Related works

Even though the high demand of automatic digitalization for P&IDs, there still have been a few studies related. At present, the

連絡先: 董　飛艶、〒113-8656　東京都文京区本郷７－３－1、東京大学大学院工学系研究科原子力国際専攻
E-mail: dongfeiyan@g.ecc.u-tokyo.ac.jp

automatic extraction information from P&IDs approaches can be divided into two types: conventional and machine learning-based approaches.

In the conventional approaches, automatic information extraction from P&IDs has been studied as graphic problem. Yan et al. [1] introduced a case-based study by listed the exact examples manually at first and iteratively searched for similar lines. Based on this, graphical attributes, such as distances with constraints between different component graphs on P&IDs, are used to complete the recognition.

In machine learning-based approaches, pattern recognition was introduced in symbol-classification on P&IDs, as in [2], where k-means clustering method was applied. Nowadays, with the flouring of the deep learning, end-to-end deep neural network-based object detection have heralded remarkable performance as features extracted using deep learning models are more precise and finer than those done by humans. Those object detection tasks are mainly composed of detecting and classifying objects, as well as localizing them in a complex background. Most of the recent successful object detection methods are based on Convolutional Neural Networks (CNNs), which are one of the popular architectures in deep learning. As demonstrated in Fig. 1, an ordinary object detection model can be divided into four modules: input, backbone, neck and head. After feeding the image into the model, a backbone consisting of multiple CNNs is used first to extract the feature maps of each part. Some of the commonly used backbone networks are VGG [3], ResNet [4] and EfficientNet [5]. Next, the neck module (e.g., FPN [6], NAS-FPN [7], Fully-connected FPN [8]) is deployed to better utilize the features extracted from the backbone. Those intensified features will indeed do better at the latter task completing the task of interference. That is to say, the head module would detect the object on the input picture. Whether the head module contains the region proposal function, which makes separate prediction for each region that may contain an object, divides the head module into so-called one-stage detector and two-stage detector. The most common examples of one-stage detectors are YOLO [9] and SSD [10]. The output of neck will be passed through the head only once and directly predict all the bounding boxes. Based on all bounding boxes of one-stage detectors that the two-stage detectors like Faster R-CNN [11] contain an extra region proposal function combining the output of neck to predict again.

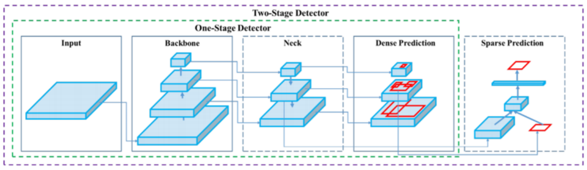


**Fig. 1 The procedure of object detector using CNNs**

In automatic extraction the information from P&IDs, the deep learning-based object detection models present suitable applicability. Nevertheless, due to the lack of benchmark datasets, this application task has received relatively low attention. Yun et al [12] implemented R-CNN architecture with customized region proposal to address this task. As the abundant components on P&IDs, the meaningless regions are inevitable. To deal with this and improve the performance, three types of datasets were used for the classification problem including positive (P model), positive with negative through k-means (PN-Kmeans model), and positive with negative through DAC (PN-DAC model). However, the multiple computations in this work reduce the engineering efficiency. To take the noise of text message and color into account, Rahul et al [13] proposed using the CTPN [14] to complete the detection of all individual components. Then the Connected Component analysis was used to form the save data structure. During the detection, as the ineluctable random noise for P&IDs like line markings and overlaid diagrams, to better recognize the association of pipeline, the Ramer-Douglas algorithms was adopted as supplement. In the field of nuclear engineering, Gao et al [15] deployed FRCNN model for the detection task and proposed the comparison standards for different datasets on P&IDs digitalization application. To achieve better performance, we propose a new detection framework and adopt the state-of-the-art deep learning-based model, YOLOv4 [16] with OCR for automatic extraction of link information from P&IDs.

## 3. Method

In this section, to extract the information from P&IDs, we propose a deep learning-based framework that consists of three steps, as illustrated in Fig. 2. Firstly, the original P&IDs were pre-processed to construct the inputs. Subsequently, the processed data are fed into the object detection network for training and testing to detect bounding boxes of linking arrow and characters. Finally, for all the character bounding boxes near the linking arrows, we convert them from images to text by OCR model.

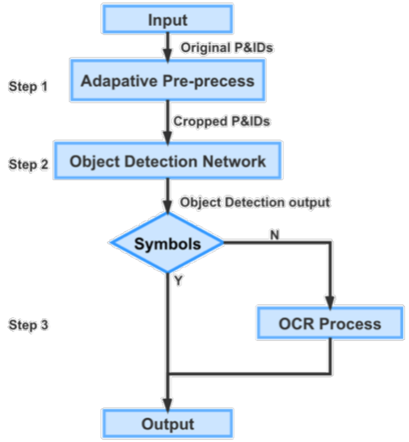


**Fig. 2 The procedure of proposed framework**

## 3.1 Objection detection

The extraction task is implemented on the deep learning-based object detection model. To achieve the best trade-off between accuracy and speed, the state-of-the-art model YOLOv4 [16] is adopted. Nowadays, the YOLOv4 model is modified with scaling technique to achieve the better accuracy in detection task. The keyword scaling refers to compound scaling model for composed convolutional layers. In every convolution block, there are three separate properties: width (the number of the parameters), depth (the layers) and height (the resolution of the input image). In conventional works, such as, VGGNet and ResNeXt [17], those methods are usually ineffective since they only focus on a single or limited scaling dimensions. Recent work like EfficientNet and EfficientDet [18] both represent remarkable performance on object detection task by jointly scaling up all dimensions of network.

In YOLOv4 model, it focuses more on analysis the relationship between different parameters to complete synergistic compound scaling. It is defined the qualitative factors including model inference time, precision, etc. that will have different gain effects according to equipment and dataset implemented on. Hence, based on the YOLOv4, design the scaling object detector on low-end, high-end and general devices respectively called YOLOv4-tiny, YOLOv4-large and CSP-ized YOLOv4 accordingly. The structure is shown in Fig. 3. Those models are designed based on CSPNet and CSPOSANet for model scaling. By using 1*1 convolutional filter primarily, the features can be split and pyramided so that the amount of computation will be cut down to 40% without damage of output accuracy. For high-end devices, it is designed the structure with fully CSP-ize model YOLOv4-P5 and scaling it up to YOLOv4-P6 and YOLOv4-P7.

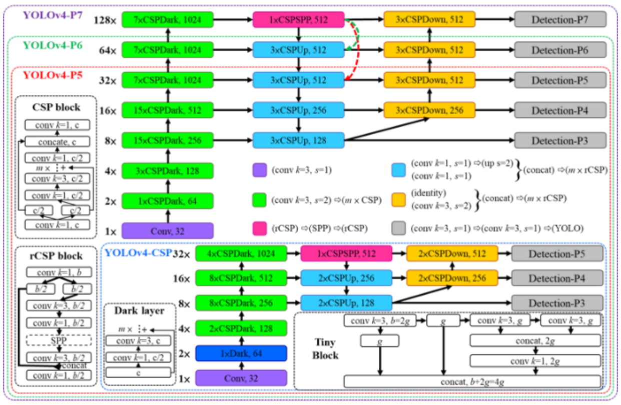


**Fig. 3 Overview Architecture of YOLOv4 [19]**

## 3.2 Sliding windows

Nevertheless, in addition to the challenge of limited resources available for P&IDs posed in Section 2, the size of P&IDs in real NPP is usually too large to handle the computation for proposed network, typically 4961*3508 pixels due to the limitation of computation resources. If those input figures are fed to the detection model directly, the model will resize the input image to smaller one automatically before training, resulting in serious low precision issues because directly loss of pixel will result in vanishing features. In P&IDs, both the symbols and characters are trivial compared to the size of original image. Hence, even loss bits of pixels presume loss of many object information. Generally, the quality of inputting fed into the proposed network has great influence on the final object detection precision. Thus, it is pivotal to pre-process the data and split into the appropriate format. To solve these problems, it is proposed an adaptive data preparation approach named sliding windows. The implementation details are shown in Fig. 4 following the subsequent steps:

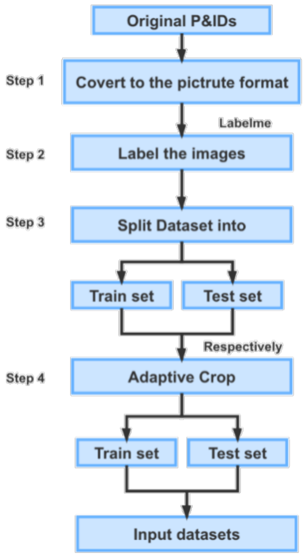


**Fig. 4 The procedure of sliding window for inputs pre-processing**

(1) The original P&IDs are converted to image format file (*.jpg).

(2) The original P&IDs images are manually annotated.

(3) The annotated images are split into the training and testing datasets in proportion.

(4) Other than directly resize, the original large P&IDs are first cropped to the ideal size for training and testing set respectively. This step completes the work of sliding windows. The original image is traversed by the ideal size window $w \times h$, where $w$ and $h$ are the width and height of window, respectively. After windowing, the original P&IDs will be transferred into many window-sized cropped images. It is equivalent to every windowed image is fed to the network parallelly and iteratively. To refine detections at the edge of the cropped images, it is necessary to preserve the overlapping intersection parts on adjacent images, controlled by the parameter interception ratio $\alpha$. $\alpha$ is computed by the proportion of overlapping area in the preset window area $w \times h$. In addition, to make good use of every annotations on cropped images so that better augment the datasets, including as many integrated samples as possible on each cropped figure, while cropping, the size will adaptively extend to the widest components all around as well as the accordingly annotation files will be updated. Fig. 5 demonstrates an example of sliding window. Fig. 5(a) represents the original P&IDs and the two cropped samples, the Fig. 5(b) shows the copped adaptive ones. After adaptation, every cropped image size will reshape to the $w\_adaptive$ and $h\_adaptive$ differently.

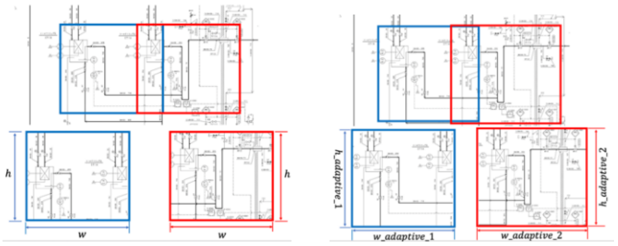


**Fig. 5 The procedure of sliding window for inputs pre-processing**

In summary, the benefits of the sliding window pre-process are as follows: 1) by cropping the original P&IDs adaptively, the number of total datasets will be increased significantly without vanishing any pixels; 2) in every cropped image, the intact samples will be contained as many as possible; 3) it can be reversed to the original large picture for the labels is examined at the same time; 4) The related coordinate on each image will be distributed more homogeneously; 5) more importantly, on each cropped image, the related size of both the symbols and text message will be enlarged.

## 3.3 Character recognition

OCR has been widely used in converting the printed or scanned text into editable one devoting to mining the text information. This intent is not so exceptional, but rather is universal and profound since based on converted information, further processing can be applied in various fields. With the flourishing of computer intelligence, it has been extensively developed for over 60 years. It is illustrated the main steps in the following Fig. 6. The workflow consists of three parts: preprocessing, feature extraction and classification. It extracts the characters and according to the semantic information, organize them into several text blocks.

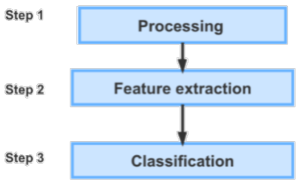


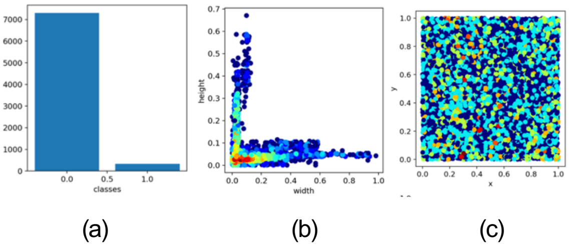
**Fig. 6 The workflow of OCR**

Thus, the accuracy of OCR depends more on the text preprocessing and segmentation algorithms. Sometimes, it might occur the problem to retrieve text for different size, angles, complex background and others from the original files. Besides, there is little visible distance between categories especially in some letters and digits for machine to understand, such as, digit "0" and letter "o", increasing the difficulty. What is more, the recognized text blocks from OCR are usually discrete lacking context order. Compared to conventional OCR methods, nowadays, it is proposed the deep learning-based algorithms to better handle the above problems and Tesseract is the one of them. Tesseract [20] is implemented on LSTM [21] algorithm with remarkable performance and thus being adopted in this work for OCR task.

## 4. Experiments and Results

### 4.1 Datasets

Five pages of P&IDs from one real world NPP are used in the current experiments, four of which are used for training and the remaining one are reserved for validating the performance of the proposed model. The resolution of each diagrams is around 4961*3508 pixels after converting from the original *.pdf file to the *.jpg file. Those original P&ID images are first be annotated manually with bounding boxes for two categories: linking arrows and characters, using the tool Labelme [22]. Subsequently, those images are split into designed series batches together with the according annotations. After comprehensive consideration, it is designed the window size as width $w = 640$, height $h = 640$ and $\alpha = 0.3$ for interception rate. The detailed distribution of the datasets is shown in Fig. 7. Fig. 7(a) represents the annotating sample number for each category. The number of characters is over 10 times more than arrows, which creates an unbalanced distribution among categories. Fig. 7(b) shows the distribution of the location of the annotated bounding boxes on each cropped image. The probabilities of the locations are uniformly distributed over the cropped images. The relative size of bounding boxes to the cropped images are illustrated in the Fig. 7(c).



(a) (b) (c)

**Fig. 7 Detailed distribution of the training dataset: (a) the sample number for each category, (b) the location distribution of bounding boxes, (c) the relative size of bounding boxes**

### 4.2 Object detection results

In the automatic extraction information from P&IDs task, taken the precision as the most important consideration, we employ the YOLOv4-P5 as the training model and set batch size as 16. We train the model for 200 epochs using SGD optimizer with initialized learning rate of 0.001 and the momentum of 0.937.

The results of the experiments are reported in Fig. 8 to Fig. 10. In YOLOv4, as it belongs to the anchor-based object detection model, it will output many bounding boxes and corresponding confidence scores, each can be presented as $(x, y, w, h, c)$, where $(x, y)$ represents the center point coordinate of the bounding box and $c$ stands for the scores for predicted bounding box to the ground truth one while $(w, h)$ represents for the relative width and height for the bounding box. Therefore, the most important criteria are related to the detected classification and bounding box. As for the total loss of the model, it is composed by three loss function: localization loss, confidence loss and classification loss. The localization loss represents for the sum-squared error between the detected bounding box and the ground-truth bounding box. The IoU (Intersection over Union) loss function [23] has been widely used for bounding box regression, which takes the integrity of the object into consideration other than treat the coordinate of the bounding box independently and simply compared to the maximum using MSE (Mean Square Error) when estimate the coordinate values of each detected bounding box. However, when the intersection area is zero, the IoU loss function cannot be differentiable for gradient computation. This can be best handled by the improved CIoU loss function implemented in YOLOv4 algorithm. CIoU additionally introduces the overlapping area, the distance between center points, and the aspect ratio for better convergence speed and accuracy on the bounding box regression problem.

Each inputting image can be split into many small square grid cells and on each cell, if the center of the object is included, the classification of this cell is set to the category of object and the bounding box will be predicted many times. First to delete the ones lower than the IoU threshold. The confidence loss consists of two situations, one is the object included cell and another is non-object included cell controlled by the parameter of ground truth. If non-object includes, the ground truth of it will be 0. And for object include situation, the ground truth of each cell is 1 and the confidence loss function will be calculated as the $1 * IoU_{Pred}^{Truth}$, of which the truth represent for the ground truth of the label and the prediction bounding box is changed when training to find out the maximum IoU one as this predicting bounding box for calculating the confidence loss function. The classification loss represents by the cross entropy for evaluating whether the bounding box contains the center of objects or not, the loss will be updated only for the object including cell. The trends of three loss function shows the performance of the detection model. It is supposed that the less of the error, the higher accuracy of prediction has been made. Finally, those three functions will be integrated to the total loss function by different weight. Fig. 8 illustrates the training and validating losses. With the time went by, the loss of four loss function is decreased, showing the effectively regression of detection model. The detection model can be quantitative analysis by the of the mean Average Precision (mAP). Fig. 9 shows the mAP of the threshold of IoU is 0.5, the mAP can be achieved around 85%. However, when it comes to the mean average precision from the threshold from 0.5 to 0.95 (interval as 0.05 for each), the mAP is under 0.5. The reason why it happens can be resulting from the imbalanced distribution of the number of two categories in the training dataset. Thus, the minority category, arrow, will have a lower confidence score than the other. When the threshold increased, the mAP will be decreased. In future work, this problem will be one of the key points to deal.

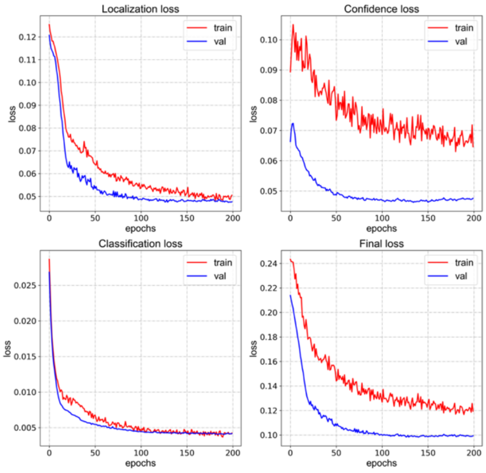


**Fig. 8 Training and validating losses**

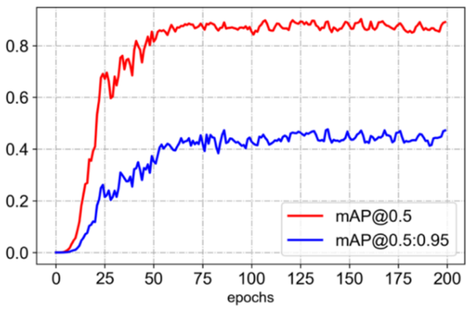


**Fig. 9 The validating mAP**

The model detection outputs on cropped images are shown in Fig. 10. The two target categories, namely arrows and characters, are represented by blue and yellow bounding boxes, respectively.

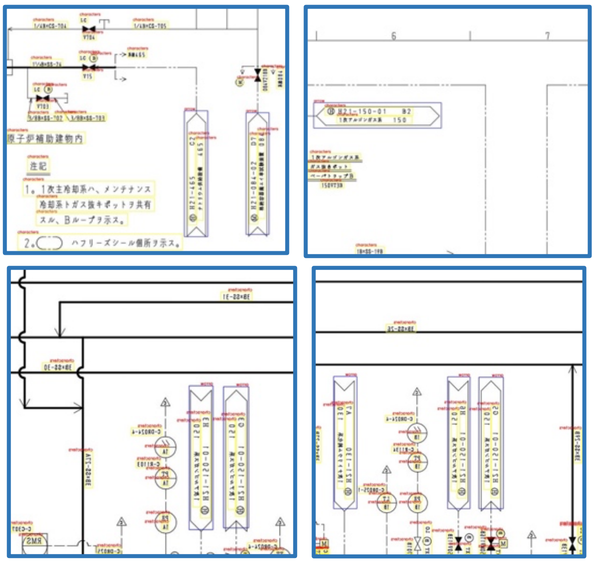
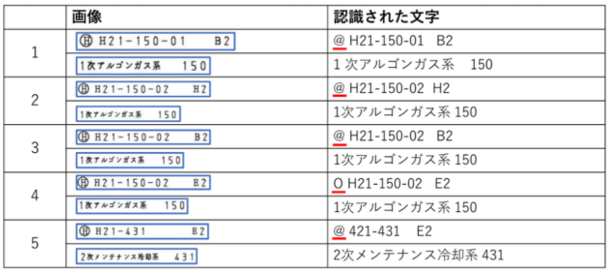


**Fig. 10 The sample results of object detection on cropped P&IDs**

All experiments are performed on a machine with Intel Xeon W-2125 (4 cores, 4.0GHz), 128GB DDR4 RDIMM RAM, NVIDIA Quadro RTX 6000 GPU (24GB of GDDR6 memory and 4608 CUDA cores).

### 4.3 Character recognition results

The sample results from character recognition are shown in Fig. 11. As it stated, most characters and numbers can be recognized well. However, when it comes to some special marks, such as the H with a circle shown in the sample, there are some recognition errors. This will be addressed in the future work by comparing it with the set of common terms used in NPPs.



| | 画像 | 認識された文字 |
|---|---|---|
| 1 | Ⓗ H21-150-01 B2 | @ H21-150-01 B2 |
| | 1次アルゴンガス系 150 | 1 次アルゴンガス系 150 |
| 2 | Ⓗ H21-150-02 H2 | @ H21-150-02 H2 |
| | 1次アルゴンガス系 150 | 1次アルゴンガス系 150 |
| 3 | Ⓗ H21-150-02 B2 | @ H21-150-02 B2 |
| | 1次アルゴンガス系 150 | 1次アルゴンガス系 150 |
| 4 | Ⓗ H21-150-02 B2 | O H21-150-02 E2 |
| | 1次アルゴンガス系 150 | 1次アルゴンガス系 150 |
| 5 | Ⓗ H21-431 B2 | @ 421-431 E2 |
| | 2次メンテナンス冷却系 431 | 2次メンテナンス冷却系 431 |

**Fig. 11 The sample results of character recognition on cropped P&IDs**

## 5. Conclusion and future work

In this paper, a state-of-the-art deep learning-based object detection model combined with OCR has been proposed to automatically extract the information from the P&IDs in NPPs. As mentioned above, due to the nature of the large number of small objects on each P&IDs, novel preprocessing methods are crucial for the final detection results. In addition, these simple objects can be easily overfitted during training. To avoid this, a novel image pre-processing approach is proposed using sliding windows and YOLOv4-P5 model is adopted to detect low resolution small objects. According to the experimental results, the applicability of extracting link information from the P&IDs of NPPs can be verified.

Nevertheless, there still exist some issues can be optimized in the future work. The unbalanced distribution of each category of data has a significant impact on the detection results, thus how to balance the number distribution of each category to obtain more accurate output is an important consideration in future work. In addition, constructing the set of common terms in NPPs will be performed to improve the performance of the character recognition.

## References

[1] L. Yan and L. Wenyin, Engineering drawings recognition using a case-based approach. 2003.

[2] E. Elyan, C. Moreno-García, and C. Jayne, Symbols Classification in Engineering Drawings. 2018.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[5] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 36th Int. Conf. Mach. Learn. ICML 2019, vol. 2019-June, pp. 10691–10700, 2019.

[6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," Dec. 2016.

[7] G. Ghiasi, T. Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 7029–7038, 2019, doi: 10.1109/CVPR.2019.00720.

[8] Y. Wu et al., "Rethinking Classification and Localization for Object Detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 10183–10192, 2020, doi: 10.1109/CVPR42600.2020.01020.

[9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.

[10] W. Liu et al., "SSD: Single shot multibox detector," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, Jun. 2015, doi: 10.1109/TPAMI.2016.2577031.

[12] D.-Y. Yun, S.-K. Seo, U. Zahid, and C.-J. Lee, "Deep Neural Network for Automatic Image Recognition of Engineering Diagrams," Appl. Sci., vol. 10, p. 4005, Jun. 2020, doi: 10.3390/app10114005.

[13] R. Rahul, S. Paliwal, M. Sharma, and L. Vig, Automatic Information Extraction from Piping and Instrumentation Diagrams. 2019.

[14] Z. Tian, W. Huang, H. Tong, P. He, and Y. Qiao, Detecting Text in Natural Image with Connectionist Text Proposal Network, vol. 9912. 2016.

[15] W. Gao, Y. Zhao, and C. Smidts, "Component detection in piping and instrumentation diagrams of nuclear power plants based on neural networks," Prog. Nucl. Energy, vol. 128, p. 103491, Oct. 2020, doi: 10.1016/j.pnucene.2020.103491.

[16] Bochkovskiy, C.-Y. Wang, and H. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020.

[17] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 5987–5995, 2017, doi: 10.1109/CVPR.2017.634.

[18] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 10778–10787, 2020, doi: 10.1109/CVPR42600.2020.01079.

[19] Chien-Yao Wang and Alexey Bochkovskiy and Hong-Yuan, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," arXiv, 2021. https://arxiv.org/abs/2011.08036.

[20] "Tesseract." [Online]. Available: https://github.com/tesseract-ocr/tesseract.

[21] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[22] K. Wada, "labelme: Image Polygonal Annotation with Python," 2016. https://github.com/wkentaro/labelme.

[23] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, vol. 34. 2020.