

Automated Valve Detection in Piping and Instrumentation (P&ID) Diagrams

M. Gupta^a, C. Wei^a and T. Czerniawski^a

^aSchool of Sustainable Engineering and Built Environment, Arizona State University, USA
E-mail: mgupta70@asu.edu, cwei32@asu.edu, Thomas.Czerniawski@asu.edu

Abstract – For successfully training neural networks, developers often require large and carefully labelled datasets. However, gathering such high-quality data is often time-consuming and prohibitively expensive. Thus, synthetic data are used for developing AI (Artificial Intelligence) /ML (Machine Learning) models because their generation is comparatively faster and inexpensive. The paper presents a proof-of-concept for generating a synthetic labelled dataset for P&ID diagrams. This is accomplished by employing a data-augmentation approach of random cropping. The framework also facilitates the creation of a complete and automatically labelled dataset which can be used directly as an input to the deep learning models. We also investigate the importance of context in an image that is, the impact of relative resolution of a symbol and the background image. We have tested our algorithm for the symbol of a valve as a proof-of-concept and obtained encouraging results.

Keywords –

Piping and Instrumentation Drawings; Yolo; Symbol Detection; Convolution Neural Network; Engineering Drawings; Symbol Classification; Deep Learning

1 Introduction

A P&ID diagram depicts the logical flow of information about physical processes and plant components with help of lines and symbols. Lines of varying thicknesses are used to represent different pipelines and each symbol represents a unique item like pressure sensor, temperature sensor, gate valve, floor drain, etc. To exemplify, a sample P&ID diagram is shown in Figure 1. A typical P&ID diagram can have more than 30 different symbols and thus, are information-rich. These drawings are analyzed manually for the purpose of estimating the quantities of various equipment while placing a purchase order and even when project teams are planning their work schedule. This analysis is highly dependent on the subjective

knowledge of the person who is reviewing these drawings and thus, can be time-consuming and prone to human errors. This task can become even more challenging and complex when there are symbols on P&ID diagrams which are functionally different but visually similar, as shown in Figure 2. Thus, differentiating one symbol from another can become extremely important and challenging. Additionally, misinterpreting or overlooking any information can prove detrimental to a project's progress and can result in serious internal conflicts.

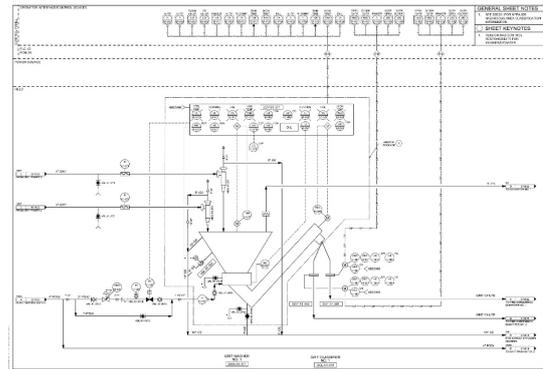


Figure 1. Sample P&ID diagram.



Figure 2. An example of similar symbols

It is safe to assume that companies have these drawings in legible electronic format for their ongoing projects which can be manipulated using state-of-the-art software. But there are still many companies who have these drawings in form of hard copies that is, paper-format or in scanned-format, especially for their older projects. So, digitization of these drawings in a format

which enables easy and user-friendly extraction of information can prove beneficial [1]. This can enable easy rectification of old drawings where the plant components have been replaced due to maintenance over time. Thus, with digitized and updated P&ID diagrams, it will be easier for the project teams to track their instrumentation inventory during the construction phase and develop an up-to-date drawings repository for the maintenance in the post-construction phase.

Currently, the construction industry does not possess a large dataset which is labelled and is publicly available. Creating such a large and real-world labeled dataset requires dedicated efforts from the experts to carefully annotate them. Thus, this process can be time-consuming and can become prohibitively expensive. Therefore, synthetic data can be used for training deep learning models.

In our research, we have created a synthetic dataset which was used for training our object detection algorithm. The approach has significance as it obviates the requirement of human annotation. We try to identify a particular symbol on a P&ID diagram which we define as the ‘Target Symbol’. While doing so, we investigate the importance of contextual information while developing an object detection deep learning model. Contextual information in this paper refers to the part of an image which does not include the object of interest (OOI). With our work, we also try to bring the applications of modern technology like machine learning and deep learning into the industries like construction, oil and gas which rely on engineering drawings for their operation and functioning. Deep Learning algorithms and frameworks like Convolutional Neural Network (CNN) [2], Histogram of Oriented Gradients (HOG) [3], You only Look Once (Yolo) [4], [5], etc. enable the processing of image data and hence, the engineering drawings. For our research project, we have applied Yolo version 2 which is an object detection algorithm.

We believe the method described in our paper can be applied to the cases which have class-imbalance [6], [7] problem in an image dataset. In the context of our paper, the class-imbalance problem would mean that there could be a symbol on P&ID sheets whose total number of occurrences is far less than the occurrence of another symbol. In such cases, data augmentation strategies can often help in improving a neural network’s performance. As our method allows user to control the number of occurrences of less frequent symbols in an artificially generated dataset hence the distribution of the minority class can be balanced. This can result in increased performance while training a neural network.

2 Literature Review

In this section, we will discuss some of the previously published papers related to processing of the engineering drawings and their proposed frameworks. Papers such as [8] present work in symbol detection and [9] in symbol classification.

[10] highlights the challenges in the successful classification of symbols. The biggest challenge is the unavailability of a labelled public dataset. They discussed intra-class and inter-class similarity amongst symbols as a major obstruction for detection algorithms. Their results found that class-decomposition helped in increasing the classification accuracy. Class-decomposition is the process of breaking down labelled datasets to a larger number of subclasses by means of clustering the instances that belong to one class at a time. [10] , [11] developed heuristics-based rules for extraction and localizing symbols. Generally, these methods are highly data-dependent and could not be used if there are slight variations in symbols or P&ID diagrams as a whole.

[9], [10] present works in symbol classification and [8], [12] in symbol detection. [12] performed symbol detection on a dataset which had class-imbalance problem. The dataset had 29 different symbols in P&ID diagrams and the distribution of these symbols was non-uniform. They first applied Yolo model for symbol recognition and found 3 least occurring symbols were missed by their network. Then, they performed data augmentation for 8 minority classes (that is, 8 least occurring symbols) using MFC-GAN (Multi Fake Class Generative Adversarial Network). Results showed that MFC-GAN improved the accuracy of their model in detecting even the least occurring symbols. They stated Yolo is a simple framework consisting of a single convolutional neural network which can be used for detecting multiple bounding boxes for objects belonging to different classes. That is why we have also implemented Yolo for our research.

For an extensive analysis of P&ID diagrams, [11], [13], and [14] developed frameworks to recognize symbols, texts, and lines. In an inspiring work stated in [13], they presented a proof-of-concept for identifying symbols, interpreting component connections, and representing those connections graphically for a P&ID drawing. Their model required data in a vectorized format and had an assumption that P&ID symbols are all ‘blocked’ in DXF files. Similarly, [11] developed a methodology for a complete analysis of P&ID diagrams encompassing symbol recognition, pipeline identification, and text localization. They used a pre-trained CTPN (Connectionist Text Proposal Networks) network for text detection. Shape properties like, the number of sides in a polygon, length to width ratio, etc. were used as features to detect a pipe’s inlet and outlet.

Identified pipelines, tags, symbols, and texts were associated with each other based on the Euclidean distance. However, they annotated symbols by completely masking their pixel values. So, essentially annotations were based on the shape of a symbol's boundary or outline. We believe this approach would not be able to give accurate results when 2 or more symbols have the same shape or the same 'outer boundary'. For instance, the 3 symbols shown in Figure 3 are distinct while their outline/ boundary is the same. Nevertheless, the work in [11] is significant for the complete analysis of P&ID diagrams.

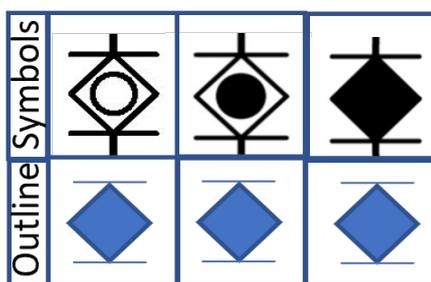


Figure 3. Showing that 'Target Symbol' has same boundary as its close neighbors

3 Experiment and Problem Setup

An industry partner provided us with P&ID diagrams, an example of which has been shown in Figure 1. The sheets have various symbols and some of them are shown in Figure 4. The main goal is to generate an algorithm that can correctly identify all the instances of these symbols. However, the current work has been performed for one symbol as a proof-of-concept and will be extended in the future for all the different symbols.

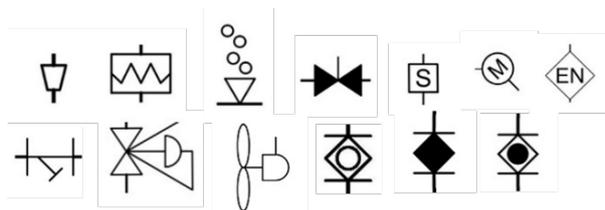


Figure 4. Few symbols in the P&ID diagrams

Thus, in this project, we are detecting one symbol representing a valve which we call as the 'Target symbol', and is shown below in Figure 5. The reason for choosing this symbol as our 'Target symbol' is that it has look-alike symbols which are the last 2 symbols

shown in Figure 4. The existence of look-alike symbols makes our symbol recognition a relatively challenging case-study, and so representative of one of the more difficult cases. So, we want our algorithm to identify the Target symbol successfully on a P&ID diagram which can also have other similar looking (but, functionally different) on it.



Figure 5. The 'Target Symbol'

Visual assessment of given P&ID diagrams reveals that the Target Symbol occurs in different orientations and has a slight variation in its appearance across the P&ID drawings. These changes in appearance are attributed to the pipeline connections and rotation of the symbol. All the possible variations of the Target Symbol have been shown in Figure 6. These configurations of the Target Symbol are stored as 8 different images in a folder whose name is 'Target_Symbol_folder' (suppose). This forms the first step of our experimental setup. Thereafter, we develop a labeled synthetic data through random cropping.

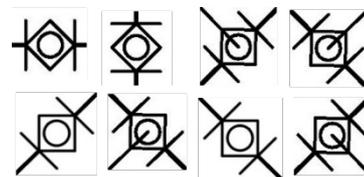


Figure 6. 'Target Symbol' and its different configurations in the P&ID diagrams.

3.1 Data Augmentation

We received 3 P&ID sheets from our industry partner. We performed data augmentation using 2 of these 3 sheets and kept 1 sheet for testing/inferencing. Original sheets have a resolution of 10100-by-6600 pixels. Thus, we decided to split it into sub-images of a smaller size of 256x256 pixels. The sub-image/ 'crop' is extracted randomly as shown in Figure 7, and the number of sub-images can be defined by the user. Script for processing the P&ID sheets to generate random crops was written in Matlab where the user also has an option to control the total number of crops to be generated from the 2 original sheets.

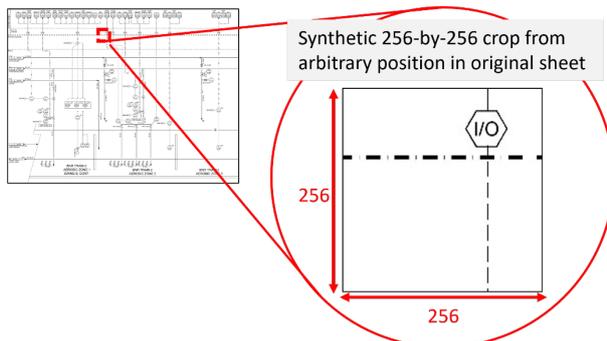


Figure 7. Sample of a 256-by-256 random crop

The idea is to generate several thousand sub-images from 2 P&ID sheets which can then be used for training the Yolo v2 neural network.

3.2 Labelled Data Generation

To create a labelled dataset, we need to have information about the Object of Interest (OOI). Because we are using Yolo v2 for object detection, therefore, we would require the location information that is, the bounding box coordinates of the Target symbol in each sub-image. To assess the impact of the context on our symbol detection we have partitioned our project into 2 phases namely, Phase-1 and Phase-2. In Phase-1, a neural network was trained on images which only had our Target symbol while in Phase-2 the input images also had one of the look-alike symbols in addition to the 'Target symbol'. We wanted to assess the impact of this change and compare the performance of these two networks. We believe this can enable us to draw insights about strategies to create an effective synthetic dataset. Detailed methodology for Phase-1 and Phase-2 are discussed below.

3.2.1 Phase-1

In Phase-1 of the project, we generated a total of 1000 random crops of size 256-by-256 pixels through the same approach as defined in Section 3.1. Then, our custom-built Matlab function will access the 'Target_Symbol_folder', and then, randomly select any one of the 8 Target symbol images and, place it randomly on 256-by-256 crop as shown in Figure 8. Our Matlab function allows us to save the position of the Target Symbol in the $[x,y,w,h]$ format. 'x' and 'y' represent the coordinates for the upper left corner, and 'w' represents the width and 'h' represents the height of the Target Symbol crop. Essentially, this is the bounding box information for the Target Symbol on each sub-image. Thus, we are able to produce a labelled dataset which can be used for training. The format of

$[x,y,w,h]$ is selected because this is the default input format for Matlab. Hence, the dataset generated can be used directly as an input for training the neural network in Matlab.

However, minor manual updates were required in a special case where the crop already had the 'Target Symbol' on it. Thus, when our algorithm pasted one more instance of it, the sub-image would have 2 Target symbols on it. An example is shown in Figure 9. Hence, bounding box information for the pre-existing symbol was added manually to the list where our Matlab program was storing the position coordinates $[x,y,w,h]$ of all the Phase-1 crops.

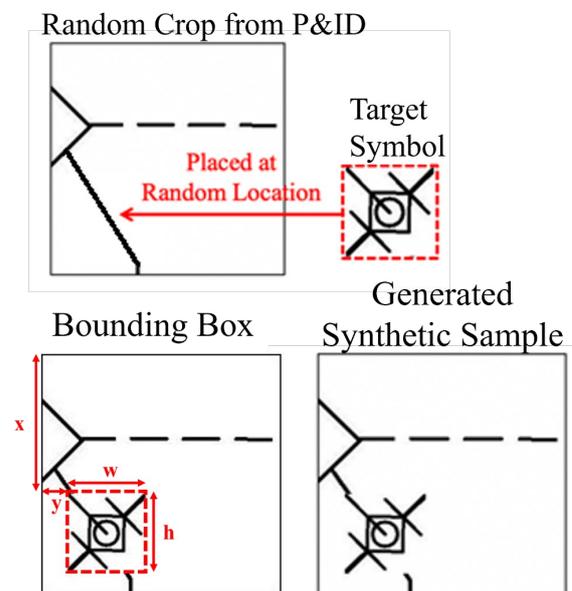


Figure 8. Sample of a 256-by-256 random crop with 'Target symbol'

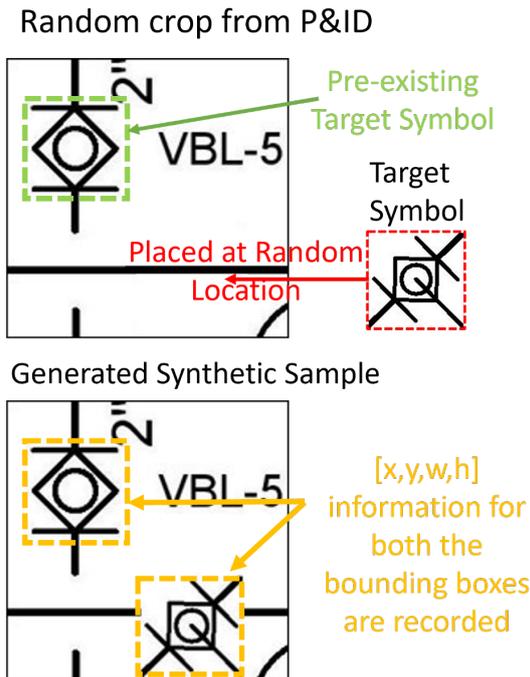


Figure 9. Crop having both the pre-existing and newly pasted 'Target Symbol'

Now this labeled dataset can be used as input to our object detection model. Similar to [12], we also used Yolo algorithm as the choice of our object detection model. Our Yolo v2 network has Resnet-50 [15] as a classifier for feature extraction and a CNN as our detection network.

It is to be noted that a detector trained as described above can only be used for testing on 256-by-256 images. Hence, we could not use this detector for testing directly on the third original sheet whose size is 10100-by-6600 pixels. So, after training the Yolo detector, we developed a custom pipeline that would run our trained detector in a sliding window manner of size 256-by-256 pixel with a stride of 256 to test any P&ID diagram whose resolution is more than 256-by-256. Results of Phase-1 are summarised in Section 4.1.

3.2.2 Phase-2

In Phase-2, we seek to improve the performance of the system presented in Phase-1 by (1) increasing the amount of contextual information in each sample and (2) adding adversarial symbols (i.e., look-alike symbols) into the training symbols. We think one way to achieve (1) is by decreasing the resolution of the original P&ID sheet which in turn increases the amount of 'background' for the Target symbol. For (2) we modified the crops so that our sub-image comprised of one Target Symbol and one of its look-alike symbols on the same 256-by-256 image crop. We wanted to study how these variations from Phase-1 affects the model performance. As already

mentioned, our Target Symbol has a close resemblance with 2 other symbols. The various configurations in which these 2 symbols occur in the drawings set are shown in Figure 10. These symbols are saved as 14 different images in a folder which is named as 'Similar_to_Target' (suppose). In other words, the images in Similar_to_Target folder form a close neighbour group for our Target Symbol.

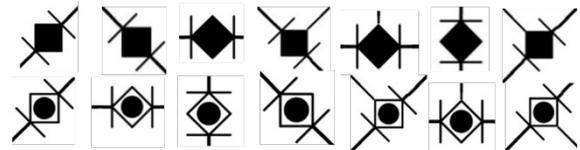


Figure 10. Collection of symbols configurations that resemble the 'Target Symbol' (referred to as close neighbors)

Hence, for Phase-2 we formulated a few changes in Phase-1 which are listed below:

1. Resize the original P&ID sheets from 10100-by-6600 pixels to 3800-by-2450 pixels. The reason for doing this is that a crop of the 256-by-256 crop out of a 10100-by-6600 pixels sheet didn't capture the background (contextual) information well. Thus, all the 1000 crops of Phase-1 didn't have a well-representative background. Whereas crops generated from 3800-by-2450 pixels sheets fairly capture the background information in them. Thus, these new crops are more representative of the original sheets. This is demonstrated in Figure 11. It can be observed from the figure that very less background is captured in crop (a) while crop (b) has significantly more context/ information about the background.

Crop from 10100-by-6600 pixels sheet

Crop from 3800-by-2450 pixels sheet

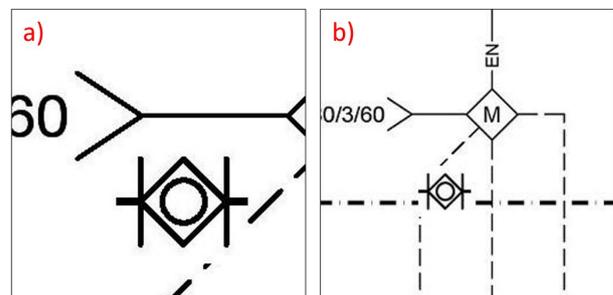


Figure 11. Comparison of background information in crops generated from a) 10,100-by-6,600 pixels sheet and b) 3,800-by-2,450 pixels sheet

- In addition to pasting only the Target Symbol, we also randomly pasted one of the 14 images of the “look-alike” symbols from Similar_to_Target folder. An example is shown in Figure 12 This is done to increase the ability of the model to effectively differentiate amongst similar-looking symbols.

The process of manually updating the bounding box information for the crops having pre-existing Target symbol is the same as Phase-1. However, on average only 1 in 70 crops had a pre-existing Target symbol and thus, our dataset requires minor manual updates. Hence, Phase-2 has background information-rich 1500 sub-images of size 256-by-256. The results are discussed in Section 4.2

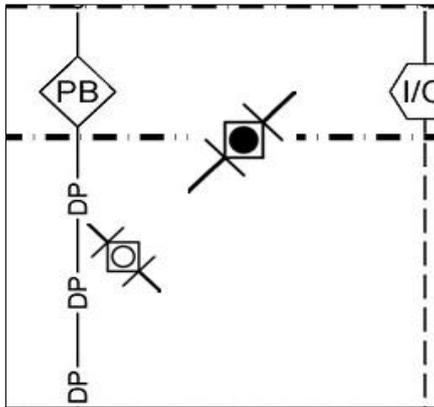


Figure 12. 256-by-256 crop in Phase-2 having both the ‘Target Symbol’ and one of its close neighbours

4 Results

As mentioned, 2 out of 3 original sheets are used for data augmentation and training while the remaining one sheet is kept reserved for testing our model and assessing its performance. The trained model is run in the sliding window manner on the third original sheet for inferencing and the results are discussed below.

4.1 Phase-1 results

The detection results for Phase-1 are shown in Figure 13 and Figure 14. It is observed that out of 5 occurrences of our Target Symbol on this sheet, our model is able to correctly identify 3 occurrences while wrongly classifying a different symbol 2 times and completely missing it 2 times as well.

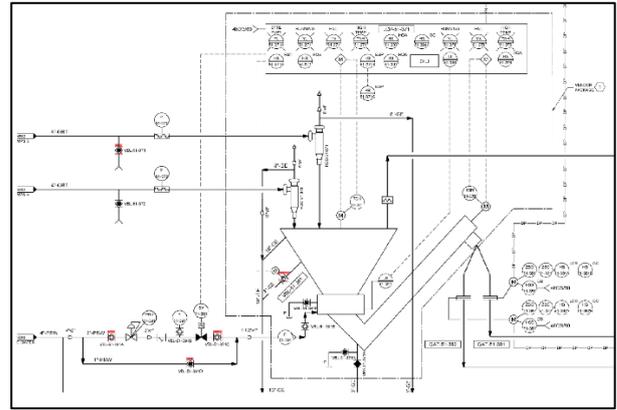


Figure 13. Phase-1 detection results on 10,100-by-6600 pixels sheet

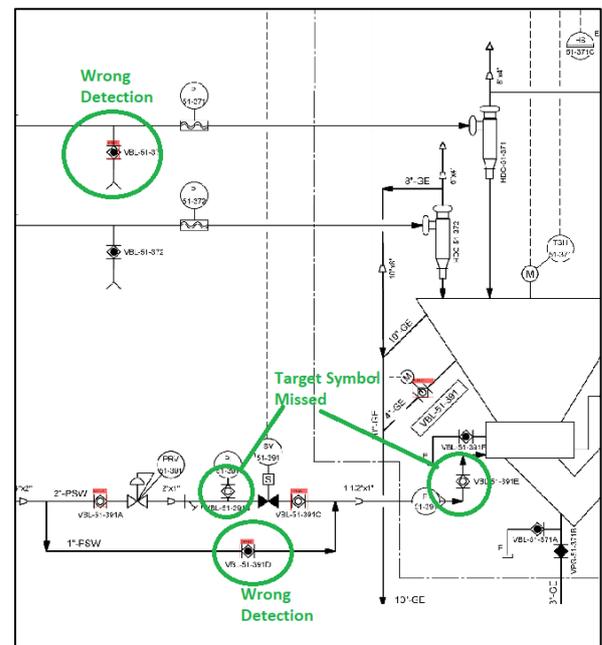


Figure 14. Zoomed-in view of Phase-1 results

Thus, our results of Phase-I have both the Type-I error and Type-II error [16]. The type-I error refers to the False positives which in our case is the “Wrong Detection” as highlighted in Figure 14. And, Type-II error is the False-negative which in our case is “Target Symbols missed” as highlighted in Figure 14. It can be noted that the wrongly classified symbol for both the occasions is one of the close resembling symbols as identified in Figure 10. This is closely related to the argument made in [10] regarding the presence of look-alike symbols.

4.2 Phase-2 results

Results for Phase-2 are shown in Figure 15 and Figure 16. It can be verified that all the 5 instances of the Target Symbol are now correctly identified without any error. It means that increasing the contextual information and introducing the look-alike symbols in 256-by-256 sub-images enhanced the discriminative ability of the neural network.

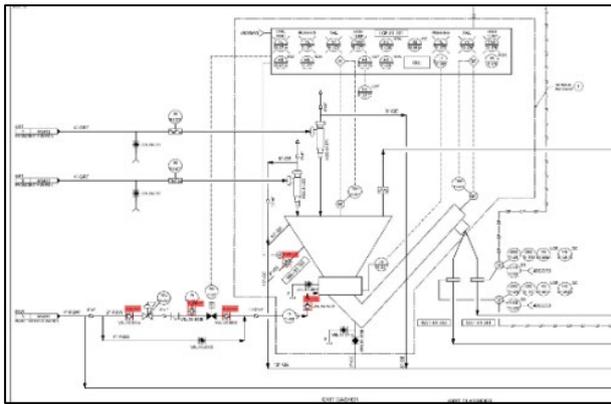


Figure 15. Phase-2 detection results

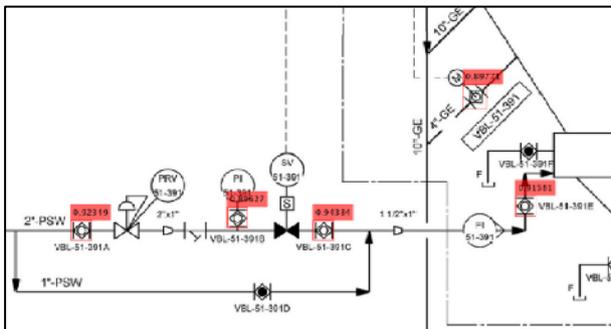


Figure 16. Zoomed-in view of Phase-2 results

5 Conclusion

In this paper, we investigated the effect of contextual information in boosting the discriminative ability of a neural network among similar-looking objects. Our developed model can differentiate among close resembling symbols and is able to find out all the instances of the desired symbol. The method for labeled data generation is simple and time efficient. It gave promising results on our dataset along with bypassing the requirement of an expert annotating the data. Therefore, it is cost-efficient too. We opine that with our approach users can generate labeled synthetic data with minimal effort. This project can also find applications in electronic circuit designing.

However, there are a few limitations to our work that we will be overcoming in our future endeavors. First, the model is trained for detecting only one symbol. In the future, we will be scaling up the project to detect all the symbols. Second, current data is from a single contractor hence, highly subjective to their 'style' of P&ID drawings. To make the program more robust and ready-to-use by contractors we will be collecting data from multiple companies to account for variations in designing layout and styles. Third, we will be increasing the scope to detect the pipelines and associated tags to graphically represent these connections. We will be developing an application to enable exporting the schedules of various pipes/symbols in a .csv format. This would be helpful in preparing BOQs (Bill-of-Quantities) and efficiently keeping track of the inventories by the project team.

Acknowledgment

We would like to thank Sundt Construction Company & General Contractor for providing us with the P&ID sheets for analysis.

References

- [1] S. v Ablameyko and S. Uchida, "RECOGNITION OF ENGINEERING DRAWING ENTITIES: REVIEW OF APPROACHES," 2007. [Online]. Available: www.worldscientific.com
- [2] K. Fukushima, "Biological Cybernetics Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," 1980.
- [3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005, vol. I, pp. 886–893. doi: 10.1109/CVPR.2005.177.
- [4] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Dec. 2016, [Online]. Available: <http://arxiv.org/abs/1612.08242>
- [5] X. Huang *et al.*, "PP-YOLOv2: A Practical Object Detector," Apr. 2021, [Online]. Available: <http://arxiv.org/abs/2104.10419>
- [6] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies."
- [7] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," Oct. 2017, doi: 10.1016/j.neunet.2018.07.011.

- [8] L. P. Cordella and M. Vento, “Symbol recognition in documents: a collection of techniques?”
- [9] J. Lladós, E. Valveny, G. Sánchez, and E. Martí, “Symbol recognition: Current advances and perspectives,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002, vol. 2390, pp. 104–128. doi: 10.1007/3-540-45868-9_9.
- [10] E. Elyan, C. M. Garcia, and C. Jayne, “Symbols Classification in Engineering Drawings,” in *Proceedings of the International Joint Conference on Neural Networks*, Oct. 2018, vol. 2018-July. doi: 10.1109/IJCNN.2018.8489087.
- [11] R. Rahul, S. Paliwal, M. Sharma, and L. Vig, “Automatic Information Extraction from Piping and Instrumentation Diagrams,” Jan. 2019, [Online]. Available: <http://arxiv.org/abs/1901.11383>
- [12] E. Elyan, L. Jamieson, and A. Ali-Gombe, “Deep learning for symbols detection and classification in engineering drawings,” *Neural Networks*, vol. 129, pp. 91–102, Sep. 2020, doi: 10.1016/j.neunet.2020.05.025.
- [13] C. Howie, J. Kunz, T. Binford, T. Chen, and K. H. Law, “Computer interpretation of process and instrumentation drawings.”
- [14] L. Boatto *et al.*, “Detection and Separation of Symbols Connected to Graphics in Line Drawings.”
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2016, vol. 2016-December, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [16] Dekking, Michel. A modern introduction to probability and statistics: understanding why and how. Springer, London. 2005.