Deep Neural Network for Automatic Image Recognition of Engineering Diagrams

Dong-Yeol Yun 1, Seung-Kwon Seo 1, Umer Zahid 2* and Chul-Jin Lee 1*

1 School of Chemical Engineering and Materials Science, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul 06974, Korea; dy7585@cau.ac.kr (D.-Y.Y.); sungkseo@cau.ac.kr (S.-K.S.)
2 Chemical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
* Correspondence: uzahid@kfupm.edu.sa (U.Z.); cjlee@cau.ac.kr (C.-J.L.); Tel: +966-13-860-7360 (U.Z.); +82-2-820-5941 (C.-J.L.)

Received: 11 March 2020; Accepted: 04 June 2020; Published: 9 June 2020

Abstract: Piping and instrument diagrams (P&IDs) are a key component of the process industry; they contain information about the plant, including the instruments, lines, valves, and control logic. However, the complexity of these diagrams makes it difficult to extract the information automatically. In this study, we implement an object-detection method to recognize graphical symbols in P&IDs. The framework consists of three parts—region proposal, data annotation, and classification. Sequential image processing is applied as the region proposal step for P&IDs. After getting the proposed regions, the unsupervised learning methods, k-means, and deep adaptive clustering are implemented to decompose the detected dummy symbols and assign negative classes for them. By training a convolutional network, it becomes possible to classify the proposed regions and extract the symbolic information. The results indicate that the proposed framework delivers a superior symbol-recognition performance through dummy detection.

Keywords: convolutional neural network; object detection; piping and instrument diagram; unsupervised learning

1. Introduction

Engineering diagrams (EDs) are schematic drawings describing process flow, circuit construction, and engineering device information. Among the many types of EDs, piping and instrument diagrams (P&IDs) are broadly used in the production plant industry because they contain key information of the plant, including piping, valves, instruments, control logic, and annotations. Moreover, extracting this information—for example, where or of what type the objects are—is the first step in estimating the number of elements and managing the project during its operational period. Most of the plant industries, such as oil and gas production plants, have employed large teams of engineers to manually count these entities and digitalize the information into their internal systems because there is no module available to automatically extract such information from the diagrams. For decades, these tasks have been considered inefficient and time-consuming tasks. Consequently, the demand for a module enabling an automatic engineering diagram digitalization has increased as such procedures can improve productivity and gain a competitive edge for the company in the global market.

However, there are obstacles to be overcome before applying this technology in real-world scenarios. Firstly, the symbols of P&IDs come in a diverse range of forms, with approximately 100 different types for each entity [1]. Furthermore, there is an inter-similarity between these symbols themselves. This requires the person interpreting and counting the symbolic entities to know the P&ID symbols and legend sheets. In some diagrams, it is difficult to identify—through image
processing alone—a target symbol without confusing it with another symbol because there are so many objects. Moreover, in diagrams, text information such as notes, as well as line number or size information, is presented near the symbols. This information is also often also written across the symbol; therefore, it can present another obstacle in the effective recognition of diagrams. These key challenges need to be overcome to enhance the capabilities of P&ID digitalization procedures in real-world scenarios; however, there have been few studies applied to develop an object-detection algorithm to overcome these limitations and present suitable applicability.

Within the field of machine vision for EDs, there have been a few studies that have sought to extract specific information from the diagrams. In [2], the authors present new trends on machine vision to extract various information from EDs, such as binarization, contextualization, segmentation, and recognition. One of the most popular preprocessing methods is binarization; by adopting a threshold value, this method converts an image into a binary representation, thereby removing noise and improving entity identification in the diagram. There are several methods of applying such image binarizations, including global thresholding [3], local thresholding, and adaptive thresholding [4]. For line detection in the diagrams, canny edge detection [5], Hough transformations [6], and morphological dilations have been discussed in the literature. Probabilistic Hough transform (PHT) [7], which uses a random sampling of the edge points to detect lines in images, has been applied for the robust detection of lines in engineering diagrams [8]. A shape-detection procedure, employing a consistency attributed graph (CAG) with a sliding window, was used by [9] to construct a symbol-detection procedure. As a comparable method, a recursive model of the morphological opening was implemented by [10] to identify symbols by the empty fraction of their area. For text/graphics segmentation (TGS), a connected component (CC) analysis [11] was used with size constraints in an engineering diagram [12]. In one study [13], a procedure to realize pixel-wise classification into text, graphics, and background was performed using filter banks and estimations of the descriptor sparseness. To find a specific shape in the diagrams, template matching was also used with a symbol shape incorporated as part of the prior information [14]. In [15], a threshold-based object-detection algorithm was proposed for binary images.

For graphical symbol recognition using a machine learning approach for engineering diagrams, there is very little previous research. In the past, the template matching method [16] has been famously used to find a specific shape within an image by sliding the template across the entire window. In all slides, the template calculated the similarity using convolutional estimation. However, this method has an inherent disadvantage; when it estimates the similarity within a specific region, it uses the Euclidian distance. This metric is intuitive, and the method is convenient to apply, but it cannot consider the images with large numbers of dimensions. It judges similarity using only quantitative calculations. Consequently, when it sets a high threshold value, it misses the shapes in the image owing to the stringent criteria. In contrast, when it sets a low threshold value, it detects unsuitable shapes because of its naïve criteria. This makes it difficult to find a suitable threshold value and improve its detection performance. Furthermore, almost all images in the real world contain noise or resolutions inadequate for the implementation of machine vision methods. In terms of industrial scenarios, the template matching-based method is not suitable for object detection in engineering diagrams.

In electrical EDs, circuit symbols can be recognized with morphological operations and geometric analyses [17]. A convolutional neural network can be used to recognize the symbols in hand-sketched engineering diagrams and convert them to a computer-aided design (CAD) program [18]. Adapting the Hopfield model, an iterative neural network model was implemented for symbol recognition by employing a prototype [19]. One symbol-classification method applied to P&IDs considered class decomposition using k-means clustering [20]. Fully convolutional networks (FCN), which are an end-to-end network for pixel-wise prediction, were first applied as object detection for P&IDs in [21]. In [22], the author proposed a method to extract various objects, including symbol, characters, lines, and tables in a P&ID, using a machine vision method containing deep learning architecture. To reduce human effort while validating the CAD documents, the P&IDs attributed by the graph form were trained by a neural network and predicted the components vector which
represents the diagram flow in [23]. However, these previous approaches toward symbol recognition for P&IDs could not control the detection quality effectively for the symbols in the diagrams. Their applicability to real diagrams could not be confirmed because unpredictable objects were detected during the region proposal. This makes it difficult to recognize the target symbol owing to the characteristics of engineering diagrams made up of quasi-binary components.

In object detection, certain algorithms have delivered remarkable performances in recent years. The appearance of convolution networks has heralded significant improvements in image-classification problems. Even so, unlike basic image classification, object detection requires the solution not only of multi-labeled classification problems but also the bounding boxes for proposed regions in a digital image. To achieve this, networks for object detection employ a region proposal network (RPN), which plays the role of finding the symbol-region candidates, before classifying images. In a region-based convolutional neural network (R-CNN) [24], a selective search algorithm [25] was used for the RPN, and the boxed images were fed to the network for classification. RPN extracts numerous boxes from an image, considering the colors, scale, boundary, etc., of the object. The proposed regions are reshaped before being fed into the convolutional network for image classification. However, R-CNN has inherent limitations; it is expensive and slow. All processes in R-CNNs, from the RPN to the convolutional network, render the model inefficient and slow to detect objects in an image in real-time. Advanced models such as Fast R-CNN [26] and Faster R-CNN [27] improve network performance and speed by including the RPN in the neural network. In fast R-CNN, by employing a selective search algorithm as the RPN and implementing it into the neural network, it becomes possible to combine the different procedures into one end-to-end network. In Faster R-CNN, to reduce time consumption in the selective search algorithm, the algorithm was replaced by a combined neural network, which made the detection much faster. Many other state-of-the-art algorithms are being proposed as next-generation object identification strategies [28], [29]. However, these have focused on problems such as the improvement of model accuracies for colored images, improvement of detection speeds, and image segmentation (i.e., the process of partitioning the image into a set of pixels with multiple segments). However, engineering diagrams have a characteristic difference from colored images; they are an almost-binary component matrix with a specific size and shape for each symbol. This makes it difficult for a model to classify symbols using only limited information.

As an advanced application of object detection in EDs, we propose an R-CNN architecture containing clustering. For the RPN, we implement a sequential image-processing method that is modified for two types of target symbols: valves and instruments. By modifying the image-processing method for region detection, we propose candidate symbol regions using size-based detection. After the RPN, we get the symbols, but we also get the meaningless regions inevitably, such as truncated line, curve, noise, and so on, we call these ‘dummy’. The detected dummy images are decomposed by unsupervised learning methods, and negative classes are assigned to them for image classification. For images containing positive classes, which are our target symbols, the dataset is augmented with padding-block. Through a simple convolutional network, the multi-class classification model is trained and applied to new diagrams for the model test.

In this research, we propose a model based on an R-CNN architecture that features dummy image clustering. A sequential image-processing method is used for the RPN, instead of a selective search algorithm. After the RPN, the dataset is constructed by positive classes, and dummy clustering is applied to treat the unwelcomed detections as negative classes; thereby improving the classification performance of the convolutional network. The remainder of this paper is organized as follows. Section 2 provides the methodology for the extraction of target symbols from P&IDs. Based on our proposed method, the region proposal and symbol-recognition results are discussed in Section 3. Finally, we conclude the paper in Section 4.

2. Materials and Methods

Here, we propose our R-CNN framework for recognizing graphical symbols in P&IDs, as shown in Figure 1. There are two main types of graphical symbols targeted in this study: valves and
instruments. These symbols have characteristics such as size and shape, as shown in Figure 2. Using these characteristics, we construct an RPN by modifying several image-processing techniques. There are two types of proposed regions: symbol and dummy. For data annotation of the symbols representing positive samples, a P&ID symbol and legend sheet are referred to, which provide the standard set of shapes and symbols for documenting the diagram.

![Figure 1. A framework summary for symbol recognition of the piping and instrument diagram (P&ID).](image)

![Figure 2. Graphical characteristics of the target symbols.](image)

For the dummy images representing negative samples, two unsupervised learning algorithms: k-means clustering and deep adaptive clustering (DAC), are used to analyze their hidden patterns and assign classes. After the annotation is completed, data augmentation is applied to generate additional information about the symbols. A convolutional neural network (CNN) is used to classify the symbols in this research owing to its superiority in local feature extraction [30]. After training the network, we apply it to another diagram in the same project and verify the results.

### 2.1. Data Sets

To implement the proposed framework and validate its performance, we use 10 pages of P&IDs from a real project. The resolution of the diagrams is 300 dpi in A3 size; thus, they contain approximately $4000 \times 3000$ cubic pixels. Of the 10 pages, we take seven and apply the region proposal...
method to construct our dataset of the proposed regions, which contains both positive and negative samples. After augmentation by padding (100 × 100 pixels), this is fed into a simple CNN.

We construct and compare three models based on the type of data that they use—positive samples only (P), positive with negative samples through k-means clustering (PN_Kmeans), and positive with negative samples through deep adaptive clustering (PN_DAC). To investigate and test our models, they are coded using Python with a Tensorflow backend. We also maintain the same computational conditions using NVIDIA TITAN V with 8 GB GDDR5.

2.2. Region Proposal

Region proposal is the process to extract candidate regions of symbols. Instead of apply the selective search algorithm, we build a customized process to extract the candidate regions in EDs as given in Figure 3.

![Figure 3. Procedure followed by the proposed region proposal network (RPN).](image)

There are several points of incongruence in the selective search algorithm. For the proposal of candidate regions, the selective search algorithm uses the traits of an image, such as its color and boundary. However, this is not appropriate for the detection of an object in a binary image, such as an engineering diagram. To investigate the most-suitable algorithm for an engineering diagram, we implement sequential image processing to create proposal regions; this detects the potential target symbols by their characteristics. The sizes and shapes of plant symbols are specified in the diagram; therefore, it is possible to modify the image processing technique for each type of target symbol—valves or instruments—as they have a set size and aspect ratio in copies with identical resolutions. Thus, using copies of the input image, the characteristics of each target symbol can be reflected and sought for in each step. To modify the progress, we divide it into four parts: (1) image binarization, (2) non-target removal, (3) morphological transformation, and (4) CC analysis, as shown in Figure 3.

In image binarization, the adaptive threshold method [31] is used to reduce the noise present in the input images and convert them into a binary representation. Comparing a pixel against the average of those surrounding it preserves hard contrast lines and discards soft gradient changes.

In the non-target removal step, we remove the obstacles for the detection of each symbol. In the case of valve detection, other symbols such as lines, instruments, and pipe fittings are considered obstacles to clear detection. This process is employed as an intermediate stage to remove the obstacles and reduce the number of meaningless detections in the region-proposal step. For line removal, dilation kernels are used in the horizontal and vertical directions, with the structures being (1 × p)
and \((q \times 1)\), respectively. The kernel parameters \(p\) and \(q\) are adjusted by considering the size of the symbol. In this study, we use a length similar to the shortest side of the symbol. For non-target symbol removal, a CC analysis algorithm [11] and Hough circle algorithm [32] are used to find the contours of the non-targeted objects such as instruments and pipe fittings.

In the morphological transformation step, “Closing,” which is derived from the basic operations of erosion and dilation, is commonly used to enhance object outlines and small cover-up holes in the image [33]. Through this closing method, the floating objects retained from previous steps protect those background regions that have a similar shape to their kernel, while deleting all other background pixels [34].

Finally, to propose regions for the candidates of the target symbol, CC analysis is used with the constraints of symbol size and aspect ratio. The algorithm analyzes the topological structure of binary images. At the level of individual pixels, it considers 4-(8-) neighboring regions for the connected cases. We assume each symbol size and aspect ratio as prior knowledge in the detection. Figure 2 presents the schematic procedure of our region-proposal method. The image processing for region proposal plays a role in reducing the total number of proposed regions, and it adjusts the detection of undesirable objects called dummies.

After the region proposal step, we construct a dataset for the classification network as given in Figure 4. There are two types of proposed regions: symbol and dummy. Symbols are our positive samples; they are the gate valves, check valves, sensors, etc. Dummy entities have unpredictable shapes and sizes, are not within our interest, and make it difficult to classify symbols through the machine learning algorithm. Therefore, they are considered to be negative samples. For the positive dataset, we use the P&ID symbol and legend sheets, which provide a standard set of symbol shapes and legends for documenting diagrams and assign the symbol images for each class, such as gates, balls, globes, checks, etc. From the proposed regions, the positive samples are manually classified into 10 classes according to their shape and function.

![Figure 4. Data annotation: positive/negative samples.](image)

2.3. Dummy Clustering

Aside from the positive samples, numerous images remain from the proposed regions, which are called dummies. They consist of curved lines, arrow shapes, revision clouds, or cut entities, as shown in Figure 3. It is difficult for the region proposal method to control the detection of dummy entities because the diagrams are quasi-binary representations and consist of many entangled lines and entities. In this research, we assign the negative classes of the detected dummies to improve the classification performance and consider the applicability of the procedure to real projects.

Dummies are obstacles for the classification model seeking to identify target symbols from a pool of proposed regions. P&ID is a type of grayscale image that is composed of only one channel, therefore, there is an arbitrary limitation to classifying the proposed regions when only using the positive samples from the target data. Furthermore, in the region-proposal network, the patterns of the detected dummies are unpredictable because it is difficult to erase all non-target entities during image processing. These patterns, such as shape, size, and detection frequency, are uncertain in every diagram, and this makes it difficult for the model to identify the target from the pool of detection.
images. Therefore, we assign negative samples to the classification models with unsupervised learning algorithms. To decompose the pool of dummy images and assign the class as a negative sample, we apply two unsupervised learning algorithms—k-means clustering and deep adaptive clustering (DAC). K-means clustering is a basic unsupervised learning algorithm. It is an iterative method to locating k-centroids in the dataset [35]; it locates them by optimizing the position of each centroid, based on the L2 norm in the feature space, as shown in Equation (1):

\[
\text{argmin}_{c} \sum_{i=1}^{K} \sum_{j \in C_i} \|x_j - c_i\|^2
\]

(1)

\[
X = C_1 \cup C_2 \cdots \cup C_K, C_i \cap C_j = \phi
\]

(2)

The quantity x represents a pool of unlabeled data, and \(c_i\) is a centroid of the i-th cluster, \(C_i\).

DAC is also applied to decompose the hidden patterns of the dummies with an advanced method [36]. It is one of the state-of-the-art algorithms for the image-clustering problem that uses a convolutional architecture and cosine distance to measure the similarity of pairwise images with adaptive parameters. It delivers superior performance in image clustering owing to its adaptive-learning algorithm. The network solves the image-clustering problem as a binary pairwise-classification problem. The flowchart is presented in Figure 5.

![Flowchart of the deep adaptive clustering (DAC) algorithm](image)

**Figure 5.** Flowchart of the deep adaptive clustering (DAC) algorithm [36].

Initially, unlabeled images are input to the convolutional network to generate a provisional latent vector for the image. Using the latent features, cosine similarities between pairwise images \(x_i\) and \(x_j\) are calculated; then, a confusion matrix is constructed for every batch. The network objective function is defined by:

\[
\text{Min}_\theta E(\theta) = \sum_{i,j} L(r_{ij}, l_i \cdot l_j)
\]

(3)

\[
\text{s.t.} \forall i, \|l_i\|^2 = 1, \text{and} \ l_{ih} \geq 0, h = 1, \cdots, k
\]

(4)

where \(r_{ij}\) is the unknown binary variable; if the pair of input images are in the same cluster, then \(r_{ij} = 1\), otherwise, \(r_{ij} = 0\). \(\|\cdot\|^2\) represents the L2 norm of a vector, and \(l_{ih}\) represents the h-th element of the label feature of the k-dimensional latent vector \(l_i\). As the cosine similarity of the input
image pair can be formulated by \( l_i \cdot l_j \), the objective function of DAC is expressed by the loss between \( r_{ij} \) and \( l_i \cdot l_j \). The expression for \( L(r_{ij}, l_i \cdot l_j) \) is formulated as follows:

\[
L(r_{ij}, l_i \cdot l_j) = -r_{ij} \log(l_i \cdot l_j) - (1 - r_{ij})\log(1 - l_i \cdot l_j) \tag{5}
\]

However, the unknown variable \( r_{ij} \) is prior information. Thus, an adaptive parameter \( \lambda \) is applied for the stepwise threshold value; we use \( \mu(\lambda) \) and \( l(\lambda) \) as the values for selecting similar (or dissimilar) image pairs.

\[
r_{ij} = \begin{cases} 
1, & \text{if } l_i \cdot l_j \geq \mu(\lambda) \\
0, & \text{if } l_i \cdot l_j \leq l(\lambda), i, j = 1, \ldots, n \\
\text{None, otherwise,} & 
\end{cases} \tag{6}
\]

In the clustering process, the value of \( \lambda \) starts at a specific value and gradually increases. Besides this, the relationships \( \mu(\lambda) \propto -\lambda \), \( l(\lambda) \propto \lambda \), and \( l(\lambda) \leq \mu(\lambda) \) are set in the algorithm. After finishing a batch process, the parameter \( \lambda \) is also updated by the gradient descent algorithm.

\[
\text{Min}_\lambda E(\lambda) = \mu(\lambda) - l(\lambda) \tag{7}
\]

\[
\lambda := \lambda - \eta \frac{\partial E(\lambda)}{\partial \lambda} \tag{8}
\]

Here, \( \eta \) represents the learning rate of \( \lambda \). Using this adaptive modification of the parameter \( \lambda \), the algorithm performs a stepwise selection between the pair images with increasing \( \lambda \). The performance of the DAC is detailed for various datasets in [36]; it delivers the best performance in a binary image clustering problem, such as MNIST when compared against other clustering methods. Therefore, as an advanced method to decompose dummy images, DAC is applied in this research, and the results are compared with those of the k-means clustering.

The detailed architecture of DAC is summarized as follows. After the input images are padded by 100 \( \times \) 100, we use six convolutional layers with a \((3 \times 3)\) kernel size, \((1 \times 1)\) stride, ReLU (Rectified Linear Unit) activation, and padding of the same structure in this network. The number of filters in each layer are 64, 64, 64, 128, 128, and 128, respectively. A max-pooling operation is applied with a \((2 \times 2)\) kernel and \((2 \times 2)\) stride. In fully connected layers, hidden units contain 128 and 64 nodes with ReLU activation. In all the layers, batch normalization is used to prevent the outputs of the hidden nodes from fluctuating. For adaptive learning, we set the selection-control equations according to Equation (9) and (10):

\[
u(\lambda) = 0.95 - \lambda \tag{9}
\]

\[
l(\lambda) = 0.455 + 0.1 \cdot \lambda \tag{10}
\]

There are 451 instances of dummy images from the seven pages of P&IDs. Using these two algorithms, the hidden patterns of the dummy pool are identified; then, we automatically assign them into \( k \) classes as negative samples. In this research, the value of \( k \) is fixed at 13. The optimum value of \( k \) is also an issue in clustering problems; however, we only focus on the effects of assigning a negative class for classification networks.

Table 1 presents the results of the data structure with positive and negative classes. Based on the clustering results, 13 negative classes are constructed, along with 10 positive classes. After annotating the data, which contains a total of 23 classes for the proposed regions, data augmentation [37] is applied to enhance the information in each class. To augment data, we apply two methods, central movement, and rotation. Central movement means that the extracted images from the RPN are
padded by 100 × 100, before entering the network. We implement it moves 1 × 1.3 × 3 pixels around the center of the image to catch a located symbol a little sideways for the same class. The rotation is also applied because some valves exist in a rotated form in the diagrams, so, in this study, only 45, 90, 135, 180, 225, 270, and 315 rotation angles were implemented to catch the rotated symbol.

Table 1. Data annotation and augmentation.

| Annotation Type | Classes | Instances (after augmentation) |
|-----------------|---------|------------------------------|
| Positive        | 10      | 1,213                        | 29,620 |
| Negative        | 13      | 451                          | 4,610  |
| Total           | 23      | 1,664                        | 34,230 |

2.4. Convolutional Network

Several machine-learning methods can be applied to the image-classification problem, including the support vector machine, random forest, and neural network-based models; however, we implement a simple convolutional neural network as our classification model to extract local information of the image data by convolutional and max-pooling filters [37].

The detailed model structure is presented in Figure 6. We construct three convolution layers and two fully connected layers in the network. The number of convolution filters in each layer is 64, 128, and 256, respectively. A kernel size of (3 × 3), a stride of (1 × 1), and a max-pooling layer with (2 × 2) filters are used for local feature extraction. Fully connected layers consist of 256 and 23 units, and the ReLU activation function is used throughout our model, except for the end unit, wherein a softmax function is used. For generalization of the model, the dropout method [38] is applied, which is set at 0.7. The purpose of the dropout is to prevent overfitting problems in the neural network-based model by applying a zero forward-direction propagation value stochastically to every layer.

Figure 6. R-CNN scheme for symbol recognition of P&ID.

2.5. Evaluation Metric

Considering the target symbols in the diagrams and the requirements of practical applications, we suggest two metrics for validating the proposed framework—symbol recognition rate (SR) and dummy detection rate (DR), using a constant confidence threshold of 0.7.
SR is the number of correctly recognized symbols divided by the number of symbols in the diagrams. It describes to what extent the model correctly detects the target symbols in the diagrams. DR is calculated by dividing the number of dummy images confused with symbols by the number of model predictions. It also describes the capacity of our model to distinguish dummies from symbols. If the model is well-trained in object detection for P&IDs, the value of SR will be large, whereas the value of DR will be small. In this study, the models are validated and compared with each other using these two metrics.

\[
SR (%) = \frac{\text{The Number of Correct Recognitions}}{\text{The Number of Symbols in the diagram}} \times 100 \quad (11)
\]

\[
DR (%) = \frac{\text{The Number of Dummies Confused with Symbols}}{\text{The Number of Model Predictions}} \times 100 \quad (12)
\]

3. Results

3.1. Region Proposal Results

Figure 7 summarizes the region proposal results. The target symbols-valves and instruments were well-detected in these results. We implemented a customized procedure for each target symbol and integrated the proposed regions into one diagram. All the targets in the diagram were detected using image processing. For each target, the image processing was set to modify the overlapping contours in the detected regions.

Since the proposed regions were detected by size constraints in the contour method—that is the CC analysis—there were unwelcomed images in the resulting diagram. These had a similar size to the target and represented sliced lines, the edges of instruments, entangled lines, etc. To reduce the number of dummy detections, the size constraints were used to customize the image processing for each target by adopting the target size as prior knowledge. The main purpose of the region proposal was to identify the candidate regions where the target symbol might exist; therefore, a noteworthy advantage of the process is that we are not required to focus on making the number of candidate symbols as small as possible; they must be detected conservatively and passed into the convolutional network for target identification.

[Figure 7. Sample results of the region proposal network (RPN).]

From these proposed regions, we obtained a pool of images containing symbols and dummies. In the P model, only the symbol data are constructed as the dataset for the classification model. On the other hand, in the PN models, both symbols and dummies are incorporated into the model. To assign classes to the dummies, a series of detected dummies was decomposed through the clustering
algorithms. Consequently, the effects of negative classes on symbol recognition in engineering diagrams were analyzed; these are described in the following section.

3.2. Effects of Negative Classes

First, we only investigated the positive samples to test the performance of the model. The model recognized symbols in the test diagrams but could not distinguish dummy images from the proposed region. This demonstrates that the model, which is only trained with positive samples, can distinguish only symbols. We could observe that both PN models-k-means and DAC were superior to the P model in terms of target-symbol classification from the proposed regions. In Figure 8a, the symbols were well recognized by the P model, but dummies were also detected in the results. This means that the model, which was trained only on positive data, had a weakness in identifying negative samples as false. In contrast, PN models such as Figure 8 (b) exhibited strong discriminative performance between symbols and dummies. The dummies confused with the check valves and gate valves were filtered out by the PN models. These results indicated that the assignment of a negative class for classification gives the model the ability to effectively identify symbols from the pool of binary component images.

We verified this statement in Table 2. In terms of SR—the extent to which the model could recognize the symbols in the diagram—all the models delivered good performance of over 96%. The PN_DAC model outperformed the other two, with 98.08%. This suggests that in the PN models, there was enhanced ability to classify targets through the assignment of a negative class.

DR demonstrated the remarkable ability of both PN models. In the P model, 42.3% of the dummy images from the test diagrams were confused with our target symbols. This means that the P model was incapable of identifying what images represented a genuine symbol. Due to the characteristics of EDs, i.e., their binary representation, it was difficult for the model to recognize them. In the latent space of the convolutional network, the latent features of both the dummies and the symbols were confused with each other under the P model because it does not possess any information concerning negative images. Hence, we conclude that negative classes are required for object-detection algorithms in EDs.

![Figure 8](image_url)

**Figure 8.** Sample results from the (a) P_model, (b) PN_Kmeans model.
Compared to the P model, the models that consider negative samples achieve a significant reduction in the dummy detection rate. Binary images contain only a limited amount of information. Although most images in real-world applications consist of three channels—red, green, and blue—engineering diagrams consist of one channel—grayscale. They feature only one channel, which consists of quasi-binary components; hence, there is limited information available in the image, such as local features or pixel intensity. In this respect, we can say that for engineering diagrams to effectively recognize plant symbols and discard the detected dummies from the proposed regions, a dataset containing positive and negative classes is required. Consideration of the negative results yields the additional model information through which candidates can be assessed effectively.

### Table 2. Results of the models.

| Model Type          | SR (%) | DR (%) |
|---------------------|--------|--------|
| P Model             | 96.97  | 42.31  |
| PN_Kmeans Model     | 97.88  | 1.35   |
| PN_DAC Model        | 98.08  | 0.39   |

#### 3.3. Effect of Clustering Methods

In Table 3, it is shown in a confusion matrix to depict the performance of PN_DAC model.

### Table 3. Confusion matrix: PN_DAC model.

| Actual  | 3way_Vlv | Ball_Vlv | Gate_Vlv | Butterfly_Vlv | Check_Vlv | Relief_Vlv | Globe_Vlv | Utility | Sensor | PLC | Dummy | Total |
|---------|----------|----------|----------|---------------|-----------|------------|-----------|---------|--------|-----|-------|-------|
| 3way_Vlv| 4        | 0        | 1        | 0             | 0         | 0          | 0         | 0       | 0      | 0   | 0     | 5     |
| Ball_Vlv| 0        | 5        | 0        | 0             | 0         | 0          | 0         | 0       | 0      | 0   | 0     | 5     |
| Gate_Vlv| 0        | 0        | 214      | 0             | 0         | 0          | 0         | 0       | 0      | 0   | 0     | 219   |
| Butterfly_Vlv| 0       | 0        | 0        | 9             | 0         | 0          | 0         | 0       | 0      | 0   | 0     | 9     |
| Check_Vlv| 0        | 0        | 0        | 11            | 0         | 0          | 0         | 0       | 0      | 2   | 13    | 15    |
| Relief_Vlv| 0        | 0        | 0        | 3             | 0         | 0          | 0         | 0       | 0      | 1   | 4     | 5     |
| Globe_Vlv| 0        | 0        | 0        | 0             | 28        | 0          | 0         | 0       | 2      | 1   | 2     | 30    |
| Utility  | 0        | 0        | 0        | 0             | 0         | 18         | 0         | 0       | 2      | 2   | 20    | 22    |
| Sensor   | 0        | 0        | 0        | 0             | 0         | 0          | 0         | 190     | 0      | 0   | 190   | 190   |
| PLC      | 0        | 0        | 0        | 0             | 0         | 0          | 21        | 0       | 0      | 0   | 0     | 291   |
| Dummy    | 0        | 0        | 5        | 1             | 2         | 1          | 2         | 0       | 2      | 291| 306   |       |

The PN_DAC model exhibits the optimum performance in SR and DR, with 98.08% and 0.39%, respectively. Though both PN models—k-means and DAC—had a low dummy detection rate, the PN_DAC model recorded a lower score in dummy detection than the PN_Kmeans model by approximately 1%. This resulted from the differences between the image clustering methods. The k-means clustering is an iterative algorithm based upon Euclidian distance, which represents a simple quantitative distance between entities in feature space. It does not consider the direction of the feature. Consequently, the algorithm is too weak to construct with high-dimensional data such as that contained within image representations. In contrast, the PN_DAC model obtains the latent features of the data by performing efficient feature extraction using a convolutional network. As shown in Table 3, most of the confusion is created among these symbol classes, except for the three-way valves, ball valves, and sensor symbols. We also calculate F1 scores for each symbol in PN_DAC model, as given in Table 4. By solving the pairwise binary classification problem with adaptive parameters, the model delivered good performance that could be interpreted as a good analysis of the hidden patterns in the regions. DAC has configured the negative class to make it easier to distinguish between the classes, thereby increasing its performance by entering well-defined data into the model.
### Table 4. F1 score for each class (PN_DAC).

| Class          | F1 Score |
|----------------|----------|
| 3way_Vlv       | 0.89     |
| Ball_Vlv       | 1.00     |
| Gate_Vlv       | 0.97     |
| Butterfly_Vlv  | 0.95     |
| Check_Vlv      | 0.85     |
| Relief_Vlv     | 0.75     |
| Globe_Vlv      | 0.93     |
| Utility        | 0.90     |
| Sensor         | 1.00     |
| PLC            | 0.91     |

### 4. Conclusions

In this study, an R-CNN for engineering diagrams was proposed, taking negative classes into account. For an RPN, sequential image processing was modified for each target—valve and instruments. To annotate the negative class for the dummy images, two unsupervised learning algorithms—k-means and DAC—were applied to decompose the hidden patterns of the dummies, and assign negative classes. A simple convolutional network was used as the classification model because of its superior characteristics in terms of local information extraction from the images.

There were three types of datasets used for the classification problem—positive (P model), positive with negative through k-means (PN-Kmeans model), and positive with negative through DAC (PN-DAC model). Compared to the P model, both k-means and DAC had relatively low dummy detection rates of 1.35% and 0.39%, respectively, because the negative class from the unsupervised algorithm improved the model’s ability to distinguish dummies from the symbols in the diagrams. Moreover, DAC was a superior algorithm for decomposing binary representations, as the PN-DAC model had superior performance, in which the symbol recognition rate (SR) and the dummy detection rate (DR) were 98.08% and 0.39%, respectively.

From these results, we can verify that the proposed model meets the applicability and practicality criteria for P&ID object detection algorithms. The algorithm’s negative sample detection reduces due to dummies, which makes its application to real projects difficult. This object detection algorithm is expected to contribute to the automatic digitalization of engineering diagrams. Regarding further work, state-of-the-art algorithms for object detection, such as Faster R-CNN, You Only Look Once (YOLO) v3, and Single Shot Multi-Box Detector (SSD), could be modified to suit engineering diagrams. For real-world applications, a tiny-object detector also would be useful as a plant symbol recognition model.

**Author Contributions:** “Conceptualization, D.Y.Y.; methodology, D.Y.Y.; software, D.Y.Y. and S.K.S.; validation, D.Y.Y. and S.K.S.; formal analysis, D.Y.Y. and S.K.S.; investigation, D.Y.Y.; resources, D.Y.Y.; data curation, D.Y.Y.; writing—original draft preparation, D.Y.Y. and S.K.S.; writing—review and editing, D.Y.Y. and S.K.S.; visualization, D.Y.Y.; supervision, C.J.L. and U.Z.; project administration, C.J.L. and U.Z.; funding acquisition, C.J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Chung-Ang University Research Grants in 2018 and Seoul R&BD Program (20191471).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.
References

1. Howie, C.; Kunz, J.; Binford, T.; Chen, T.; Law, K. Computer interpretation of process and instrumentation drawings. *Adv. Eng. Softw.* **1998**, *29*, 563–570, doi:10.1016/s0965-9978(98)00022-2.

2. Moreno-Garcia, C.F.; Elyan, E.; Jayne, C. New trends on digitisation of complex engineering drawings. *Neural Comput. Appl.* **2018**, *31*, 1695–1712, doi:10.1007/s00521-018-3583-1.

3. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man, Cybern.* **1979**, *9*, 62–66, doi:10.1109/tsmc.1979.4310076.

4. Sauvola, J.; Pietikäinen, M. Adaptive document image binarization. *Pattern Recognit.* **2000**, *33*, 225–236, doi:10.1016/s0031-3203(99)00055-2.

5. Kittler J.; Fu K.S.; Pau L.F. *Pattern Recognition Theory and Application*; D. Reidel Publishing Company: Dordrecht, Holland, 1981; Volume 81, pp. 292-305.

6. Ballard, D. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **1981**, *13*, 111–122, doi:10.1016/0031-3203(81)90009-1.

7. Kiryati, N.; Eldar, Y.; Bruckstein, A. A probabilistic Hough transform. *Pattern Recognit.* **1991**, *24*, 303–316, doi:10.1016/0031-3203(91)90073-e.

8. Matas, J.; Galambos, C.; Kittler, J. Robust detection of lines using the progressive probabilistic hough transform. *Comput. Vis. Image Process.* **1985**, *30*, 32–46, doi:10.1016/0734-189x(85)90136-7.

9. Fletcher, L.; Kasturi, R. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 910–918, doi:10.1109/34.9112.

10. Cote, M.; Albu, A.B. Texture sparseness for pixel classification of business document images. *Int. J. Doc. Anal. Recognit.* (IJDAR) **2014**, *17*, 257–273, doi:10.1007/s10107-014-0217-8.

11. Mokhtarian, F.; Abbasi, S. Matching shapes with self-intersections: application to leaf classification. *IEEE Trans. Image Process.* **2004**, *13*, 653–661, doi:10.1109/tip.2004.826126.

12. Tuncer, T; Avci, E.; Çöteli, R. A new method for object detection from binary images. In Proceedings of the 2015 23rd Signal Processing and Communications Applications Conference (SIU), Malatya, Turkey, 16–19 May 2015; pp. 1725–1728.

13. Belongie, S.; Malik, J.; Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522, doi:10.1109/34.993558.

14. De, P.; Mandal, S.; Bhowmick, P. Recognition of electrical symbols in document images using morphology and geometric analysis. In Proceedings of the 2011 International Conference on Image Information Processing, Shimla, India, 3–5 November 2011; pp. 1–6; doi:10.1109/icip.2011.6108910.

15. Fu, L.; Kara, L.B. From engineering diagrams to engineering models: Visual recognition and applications. *Comput. Des.* **2011**, *43*, 278–292, doi:10.1016/j.cad.2010.12.011.

16. Gellaboina, M.K.; Venkparao, V.G. Graphic symbol recognition using auto associative neural network model. In Proceedings of the 2009 Seventh International Conference on Advances in Pattern Recognition, Kolkata, India, 4–6 February 2009; pp. 297–301.

17. Elyan, E.; Garcia, C.F.M.; Jane, C. Symbols classification in engineering drawings. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.

18. Rahul, R.; Paliwal, S.; Sharma, M.; Vig, L. Automatic information extraction from piping and instrumentation diagrams. In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, Prague, Czech Republic, 19–21 February 2019; pp. 163–172.

19. Yu, E.-S.; Cha, J.-M.; Lee, T.; Kim, J.; Mun, D. Features recognition from piping and instrumentation diagrams in image format using a deep learning network. *Energies* **2019**, *12*, 4425, doi:10.3390/en12234425.

20. Rica, E.; Moreno-Garcia, C.F.; Álvarez, S.; Serratos, F. Reducing human effort in engineering drawing validation. *Comput. Ind.* **2020**, *117*, 103198, doi:10.1016/j.compind.2020.103198.
24. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

25. Uijlings, J.R.R.; Van De Sande, K.E.A.; Gevers, T.; Smeulders, A.W.M. Selective search for object recognition. Int. J. Comput. Vis. 2013, 104, 154–171, doi:10.1007/s11263-013-0620-5.

26. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 1440–1448.

27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 36, 1137–1149.

28. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.

29. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. Available online: https://arxiv.org/abs/1804.02767 (accessed on 8 August 2019).

30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. Commun. ACM 2017, 60, 84–90, doi:10.1201/9781420010749.

31. Peuwnuan, K.; Woraratpanya, K.; Pasupa, K. Modified adaptive thresholding using integral image. In Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, Thailand, 13–15 July 2016; doi:10.1109/jcsse.2016.7748917.

32. Duda, R.O.; Hart, P.E. Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM 1972, 15, 11–15, doi:10.1145/361237.361242.

33. Gonzalez, R.C.; Woods, R.E.; Masters, B.R. Digital image processing, third edition. J. Biomed. Opt. 2009, 14, 029901, doi:10.1117/1.3115362.

34. Vermon, D. Machine Vision: Automated Visual Inspection and Robot Vision; New York: Prentice Hall 1994; Volume 30.

35. Vilalta, R.; Achari, M.-K.; Eick, C. Class decomposition via clustering: a new framework for low-variance classifiers. In Proceedings of the Third IEEE International Conference on Data Mining, Melbourne, FL, USA, 22–22 November 2003; pp. 673–676.

36. Chang, J.; Wang, L.; Meng, G.; Xiang, S.; Pan, C. Deep adaptive image clustering. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.

37. Wiradarma T.P. Comparison of Image Classification Models on Varying Dataset Sizes; Hasso Plattner Institute: Potsdam 2015.

38. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. 2012. pp. 1–18. Available online: http://arxiv.org/abs/1207.0580 (accessed on 12 August 2019).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).