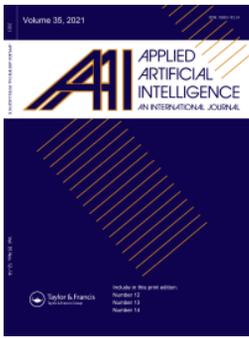


# Few-shot symbol detection in engineering drawings.

JAMIESON, L., ELYAN, E. and MORENO-GARCÍA, C.F.

2024

© 2024 The Author(s). Published with license by Taylor & Francis Group, LLC. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



# Applied Artificial Intelligence

## An International Journal

ISSN: (Print) (Online) Journal homepage: [www.tandfonline.com/journals/uaai20](http://www.tandfonline.com/journals/uaai20)

## Few-Shot Symbol Detection in Engineering Drawings

Laura Jamieson, Eyad Elyan & Carlos Francisco Moreno-García

To cite this article: Laura Jamieson, Eyad Elyan & Carlos Francisco Moreno-García (2024) Few-Shot Symbol Detection in Engineering Drawings, Applied Artificial Intelligence, 38:1, 2406712, DOI: [10.1080/08839514.2024.2406712](https://doi.org/10.1080/08839514.2024.2406712)

To link to this article: <https://doi.org/10.1080/08839514.2024.2406712>



© 2024 The Author(s). Published with license by Taylor & Francis Group, LLC.



[View supplementary material](#)



Published online: 29 Sep 2024.



[Submit your article to this journal](#)



[View related articles](#)



[View Crossmark data](#)

## Few-Shot Symbol Detection in Engineering Drawings

Laura Jamieson, Eyyad Elyan, and Carlos Francisco Moreno-García

School of Computing, Robert Gordon University, Aberdeen, Scotland, UK

### ABSTRACT

Recently, there has been significant interest in digitizing engineering drawings due to their complexity and practical benefits. Symbol digitization, a critical aspect in this field, is challenging as utilizing Deep Learning-based methods to recognize symbols of interest requires a large number of training instances for each class of symbols. Acquiring and annotating sufficient diagrams is difficult due to concerns about confidentiality and availability. The conventional manual annotation process is time-consuming, costly, and prone to human error. Additionally, obtaining an adequate number of samples for rare classes proves to be exceptionally challenging. This paper introduces a few-shot framework to address these challenges. Several experiments with fewer than ten, and sometimes just one, training instance per class using complex engineering drawings from industry sources were carried out. The results suggest that our method not only significantly improves symbol detection performance compared to other state-of-the-art methods but also decreases the necessary number of training instances.

### Introduction

Engineering drawings are widely used in numerous industries and they contain a large amount of critical information. They typically represent engineering equipment, its connections and relevant technical details. These documents are still commonly stored in an undigitised format, such as paper or PDF. Consequently, extracting data from them is very time-consuming, as it must normally be carried out manually by subject matter experts (Jakubik et al. 2022; Paliwal et al. 2021).

Recently there has been increased interest in using artificial intelligence to digitize these drawings (Bhanbhro et al. 2023). This involves creating various deep learning methods to extract all of the diagram components, which are the symbols, lines and text. One particular type of engineering drawing that has attracted the attention of both academia and industry is Piping and Instrumentation Diagrams (P&IDs) (Elyan, Jamieson, and Ali-Gombe 2020; Gao, Zhao, and Smidts 2020; Jamieson, Moreno-Garcia, and Elyan 2020; Mani

**CONTACT** Laura Jamieson,  [l.jamieson4@rgu.ac.uk](mailto:l.jamieson4@rgu.ac.uk)  School of Computing, Robert Gordon University, Garthdee Road, Aberdeen AB10 7QB, Scotland, UK

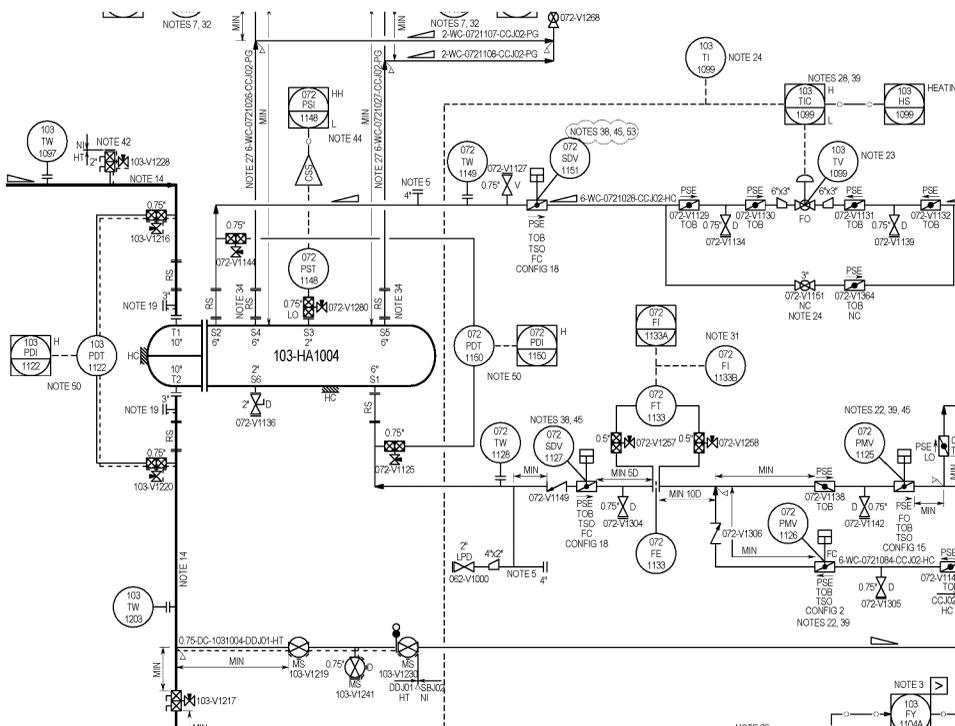
© 2024 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

et al. 2020). These are used in various domains including nuclear (Gao, Zhao, and Smidts 2020) and oil and gas (Elyan, Jamieson, and Ali-Gombe 2020; Jamieson, Moreno-Garcia, and Elyan 2020; Moreno-Garcia and Elyan 2019) P&IDs are often very complex and contain numerous symbols, connecting lines, and text, as is shown in Figure 1.

Symbol recognition is one of the main methods required for engineering drawing digitization. Very recently, researchers have created various deep learning approaches for this purpose (Elyan, Jamieson, and Ali-Gombe 2020; Jakubik et al. 2022; Mani et al. 2020). These were mainly based on object detectors, such as You Only Look Once (YOLO) series (Bochkovskiy, Wang, and Mark Liao 2020; Jocher et al. 2020; Jocher, Chaurasia, and Qiu 2023; Li et al. 2022; Redmon and Farhadi 2017, 2018; Redmon et al. 2016; Wang, Bochkovskiy, and Mark Liao 2022) and Faster Regions with Convolutional Neural Networks (R-CNN) (Ren et al. 2015).

Object detectors typically require a large labeled training dataset, however obtaining sufficient annotated symbols is challenging and can be impossible (Antonelli et al. 2022). Firstly, due to confidentiality reasons there is a lack of publicly available technical drawing datasets (Schlagenhauf, Netzer, and Hillinger 2023). Secondly, the annotation task is very costly and time-



**Figure 1.** Small section of a P&ID. These are challenging to digitise as they contain numerous symbols of various classes, orientations and sizes.

consuming, potentially taking weeks for a typical dataset (Jakubik et al. 2022). The process involves manually drawing a bounding box closely around each target symbol and then assigning the relevant class label. Given the specialist nature of the drawings, the task must be completed by subject matter experts. Thirdly, it may be infeasible to obtain sufficient instances of the rarer classes. This can result in the class imbalance problem (Buda, Maki, and Mazurowski 2018; Johnson and Khoshgoftaar 2019) which is when a model trained on an imbalanced dataset is biased toward the majority classes.

Few-Shot Learning (FSL) is the task of learning from a limited number of training samples with supervised information (Liu et al. 2023). The problem of Few-Shot Object Detection (FSOD) has received less attention from researchers compared to Few-Shot Classification (FSC) (Fan et al. 2021; Wang et al. 2020). FSOD is considered more challenging than FSC, due to the additional requirement for object localization (Antonelli et al. 2022) and because multiple instances may be present per image.

In this paper, a few-shot approach for symbol digitization in engineering drawings is presented. The key contributions of this paper can be described as follows:

- A symbol detection approach for complex engineering drawings is presented, which improves performance for novel classes with limited training data.
- We introduce one of the first examples of few-shot methods used to detect symbols in real-world complex engineering drawings. Extensive experiments were completed to validate the approach, and the results clearly show improved performance for classes with few annotations.
- The paper opens a new direction toward the use of few-shot methods for symbol detection in engineering drawings. This is extremely beneficial for rare symbols and allows for additional classes to be incorporated into a symbol detection model without extensive annotation.

The rest of this paper is structured as follows: [Section 2](#) presents a critical discussion of the related work. In [Section 3](#), the proposed methods are introduced. The experiments and results are presented in [Section 4](#). The conclusion and future research direction are then presented in [Section 5](#).

## Related Work

Due to the large amount of critical data trapped in undigitised engineering drawings, there is considerable demand to automate their digitization (Paliwal, Sharma, and Vig 2021). However, the task is considered a challenging problem, with researchers creating various methods to process these drawings over the past four decades Groen, Sanderson, and Schlag (1985); Moreno-

García, Elyan, and Jayne (2019); Okazaki et al. (1988). Initial methods were based on traditional machine learning approaches, which demanded hand-crafted features as input (LeCun et al. 1998). Although these methods proved to be successful in specific use cases, their reliance on pre-established rules meant that they did not generalize well across the variations seen in engineering drawings, such as morphological changes and noise (Yu et al. 2019; Zhao, Deng, and Lai 2020). In recent years, deep learning-based methods have significantly improved computer vision methods for tasks such as object detection (LeCun, Bengio, and Hinton 2015). These methods outperform traditional approaches as they automatically learn features from pixel data and have improved generalization ability.

Over the last few years, various deep learning methods for symbol digitization have been proposed (Elyan, Jamieson, and Ali-Gombe 2020; Faltin, Gann, and König 2023; Faltin, Schönfelder, and König 2022; Gupta, Wei, and Czerniawski 2024; Jakubik et al. 2022; Zhao, Deng, and Lai 2020, 2021). Most were based on object detection models, which predict the location and class of objects within an image. For instance, Elyan, Jamieson, and Ali-Gombe (2020) presented a You Only Look Once (YOLO) based (Redmon and Farhadi 2018) approach for the detection of symbols in P&IDs. A symbol dataset was obtained through time-consuming manual annotation of 172 high-resolution industry P&IDs. The method performed well overall with an accuracy of 95%. However, the results were inconsistent across the symbols, with lower performance on the rare classes. Meanwhile, Jakubik et al. (2022) presented a Faster R-CNN based method for symbol detection in floor plans. To avoid extensive data labeling, they generated training data using a data augmentation technique. Another possibility here is to generate synthetic data (Faltin, Schönfelder, and König 2022). However, to obtain optimum accuracy, it is typically better to source the training data from the same distribution as the test data. In another example, Gupta, Wei, and Czerniawski (2024) presented a symbol detection method in which all symbols were detected as one class using a YOLO-based model. A Siamese Network was then used to differentiate between classes.

The need for sample-efficient symbol detection methods (Mani et al. 2020) could be addressed using FSOD. In FSOD, the object classes are split into two non-overlapping sets known as the base classes and the novel classes. Base classes have a large number of labeled samples, while the novel classes have only a few. FSOD methods aim to transfer generic object knowledge from the common heavy-tailed objects to the novel long-tailed object categories (Liu et al. 2023).

FSOD is an active research area, with the majority of the methods being published in the last four years (Köhler, Eisenbach, and Gross 2023). The models are typically based on object detection architectures, the most common being Faster R-CNN. The methods can be categorized as fine-tuning

based, such as the approach introduced by Wang et al. (2020), or meta learning based, such as the method created by Kang et al. (2019). In this paper, the methods are based on the fine-tuning approach due to the reported improved performance (Hou et al. 2023).

Wang et al. (2020) introduced the frustratingly simple few-shot object detection method, and showed that good FSOD performance could be achieved using a Two-stage Fine-tuning Approach (TFA). In the first stage the model was trained using base classes and in the second stage, it was fine-tuned using all classes. Here the box classifier and box regressor were fine-tuned, and the other model components were frozen. The authors showed that their approach outperformed various methods including the meta-learning approach Few Shot Object Detection via Feature Reweighting (Kang et al. 2019).

FSOD methods based on TFA have been proposed. For instance, Kaul, Xie, and Zisserman (2022) showed that fine-tuning the Region Proposal Network (RPN) using 30 shots of novel classes substantially increased the average recall compared to that using the base RPN. To further improve the performance, the Regions of Interest (ROI) module was fine-tuned. They incorporated semi-supervised learning to obtain additional samples of novel classes and reduce the class imbalance, however this relies on additional novel class data being available. Meanwhile, Fan et al. (2021) found that the RPN in TFA was not class-agnostic and was instead biased toward the base classes. This suggests that allowing the RPN and ROI to learn from novel class data in the fine-tuning phase may improve the performance.

The most commonly used FSOD benchmarks are the splits introduced by Kang et al. (2019) on the PASCAL Visual Object Classes (VOC) (Everingham et al. 2010) and Common Objects in Context (COCO) (Lin et al. 2015) datasets. Although these datasets were widely used in FSOD, these datasets do not represent realistic rare categories and further research on more realistic datasets is needed (Köhler, Eisenbach, and Gross 2023).

The literature shows that most engineering symbol digitization methods were based on object detection models that typically require a large labeled training dataset. However, this can be infeasible to acquire due to data unavailability, rare symbols and the costly annotation process. It was also seen that few-shot object detection is a relatively new research field in which methods are designed to learn from limited data, however they have not yet been explored for engineering symbol digitization.

## Methods

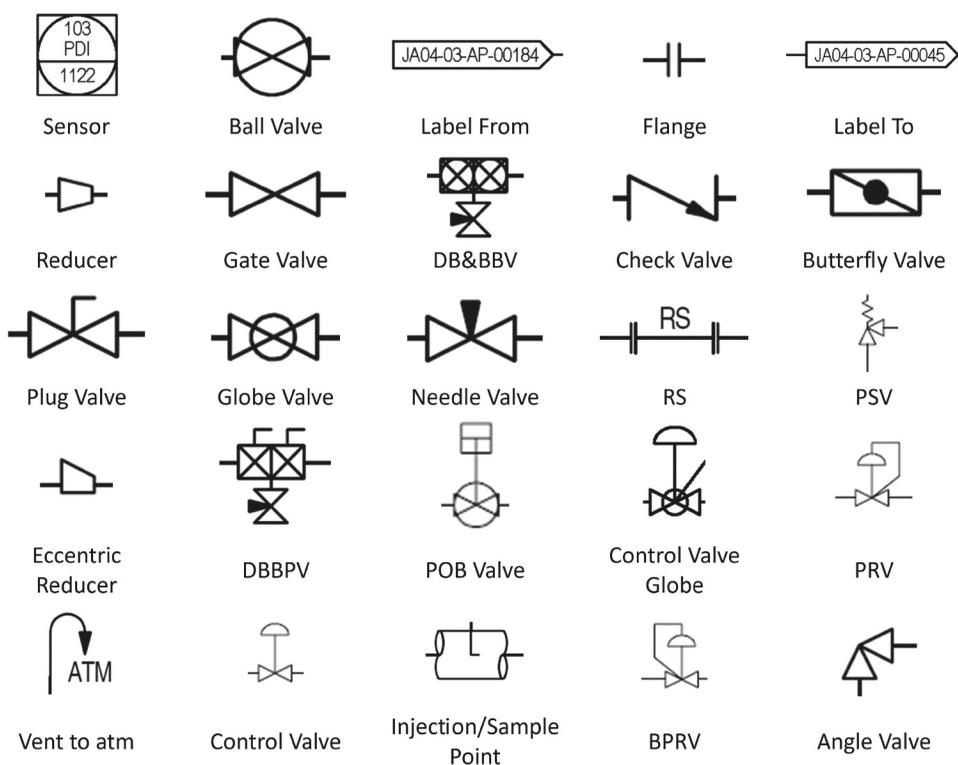
In this section, firstly the dataset of real-world engineering diagrams is introduced. Next, the methods used to pre-process this data are described. This is followed by a detailed description of the few-shot symbol detection method.

## Dataset

A dataset of 172 P&IDs was sourced from an industry partner in the Oil and Gas domain. The images are high-resolution PNG files with a size of 7428 x 5251 pixels. The diagrams represent various engineering equipment and their connections. For experiment purposes 25 symbol classes, such as valves and flow labels, were selected, as shown in [Figure 2](#).

The dataset is very challenging for object detection models. One reason for this is that the symbols are only represented by a few lines and therefore contain few features for a model to learn from. Additionally, there is usually high intra-class variability, high inter-class similarity, and the presence of similar shapes. Furthermore, unlike commonly used datasets such as PASCAL VOC, in which objects are mostly of the same orientation and are mainly located in the center of images ([Cheng et al. 2021](#)), engineering symbols are frequently in different orientations and can be located anywhere in an image.

The diagrams had been manually annotated to obtain a labeled symbol dataset. Various annotation software is available for this task, such as Sloth<sup>1</sup> and Computer Vision Annotation Tool (CVAT).<sup>2</sup> The task is known to be very time-consuming and costly for engineering diagrams ([Theisen et al. 2023](#)).



**Figure 2.** The P&ID symbol classes used in the experiment. These are challenging to detect as they are represented by only a few shapes, have high inter-class similarity and high intra-class variability.

## **Data Pre-Processing**

A series of image processing algorithms was used to remove the diagram border. This section contained various information, such as the drawing title and drawing revision details, however it contained no equipment symbols. First, the diagram was binarised and a Connected Components (CC) algorithm (Bolelli, Allegretti, and Grana 2022) was used to locate the largest CC of white pixels. This CC was considered to be the background of the main diagram area. The pixels were considered as connected if they had four-way connectivity. Each pixel outwith the bounding box of the largest CC was then replaced with a white pixel using an image mask.

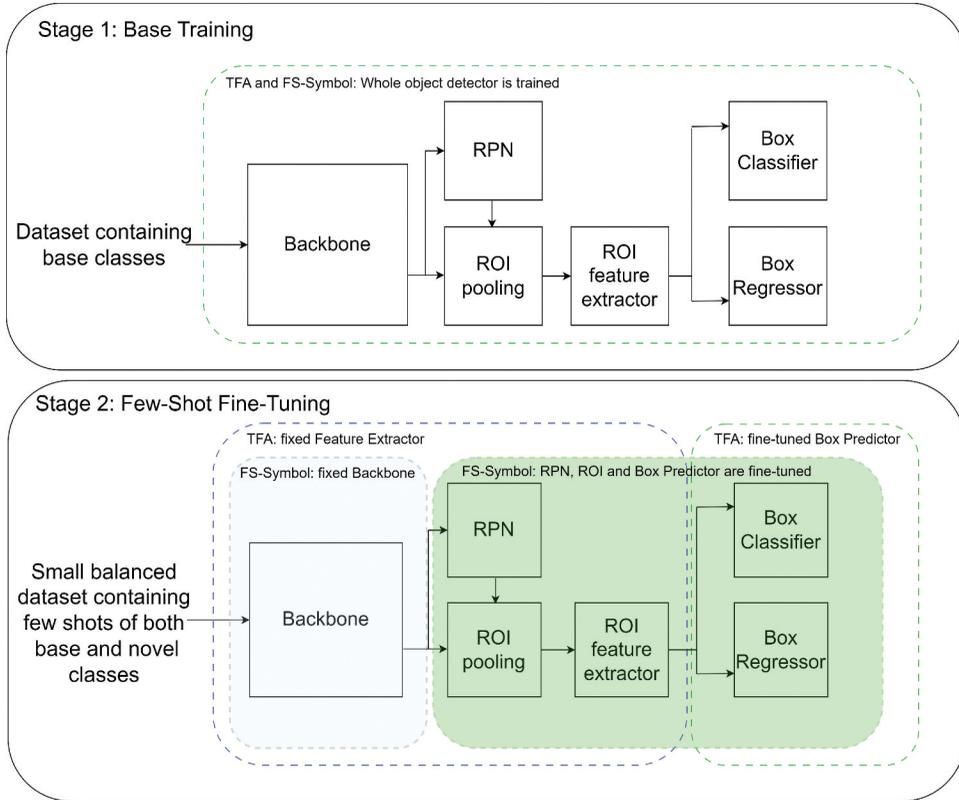
The diagrams are significantly larger than the typical input size for neural networks, and therefore a patch-based method (Elyan, Jamieson, and Ali-Gombe 2020; Ruzicka and Franchetti 2018) was used. Here, the patch size was 640 x 640 pixels. It should be noted that the patches overlapped each other at the diagram edges. Any annotation that overlapped multiple patches was not used in the training data.

## **Few-Shot Symbol Detection**

The main idea of the few-shot approach used is to separate learning of different model components. The method is based on TFA (Wang et al. 2020), which separates feature representation learning and box predictor learning. The model architecture was based on Faster R-CNN (Ren et al. 2015). The feature extractor components consist of a ResNet-101 (He et al. 2016) with Feature Pyramid Network (FPN) (Lin et al. 2016) backbone, RPN, ROI pooling layer and ROI feature extractor. The box predictor consists of a box classifier and box regressor, which predict the object categories and bounding box regression offsets respectively.

The model was trained in two stages, as shown in [Figure 3](#).

The first is base training, in which the entire object detector was trained on the base classes. The second stage is few-shot fine-tuning. In TFA the last layers of the model, the box classifier and regressor, were fine-tuned while the feature extractor was fixed. Note that the weights of the novel classifier were randomly initialized prior to the fine tuning, following the settings recommended for the PASCAL VOC dataset (Wang et al. 2020). In Few-Shot-Symbol (FS-Symbol), TFA was altered with the aim to improve the performance on the novel classes. To do this, the RPN and ROI were unfrozen in the second stage, as can be seen in [Figure 3](#). In the second step, a small balanced support dataset consisting of  $K$  shots of both the base and novel classes was used for training. In both stages, the model was trained using the multi-task loss function as in (Wang et al. 2020) and originally introduced in (Ren et al. 2015), as shown in (1).



**Figure 3.** The Two-stage Fine-tuning Approach (TFA) (Wang et al. 2020) and few-shot-symbol (FS-Symbol) methods. In the first stage, the whole object detector is trained on the data-abundant base classes. In the second stage of TFA, the feature extractor is fixed and the box predictor is fine-tuned on a small balanced dataset containing few shots of base and novel classes. In the second stage of FS-Symbol, the backbone is frozen and all other model components are fine-tuned.

$$\mathcal{L} = \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{loc} \quad (1)$$

Here  $\mathcal{L}_{rpn}$  is the RPN loss, which is the object proposal loss that determines the foreground from background and refines the anchors.  $\mathcal{L}_{cls}$  is the cross-entropy loss for the box classifier and  $\mathcal{L}_{loc}$  is the smoothed  $L_1$  loss for the box regressor.

In the box classifier, in place of fully connected classification layers, a cosine similarity classifier based on the instance-level distance measurement was used. This was reported to help to reduce the intra-class variance and improve the detection of novel classes, with less decrease in the detection accuracy of base classes (Wang et al. 2020). The classifier outputs scaled similarity scores where the similarity score,  $s_{i,j}$  between the  $i$ -th object proposal of the input  $x$ , and  $w_j$ , the weight vector of class  $j$ , is defined as shown in (2). Here,  $\mathcal{F}(x)_i$  is the input feature of the  $i$ -th object, and  $\alpha$  is a scaling parameter set to 20 following the settings implemented by (Wang et al. 2020).

$$s_{i,j} = \frac{\alpha \mathcal{F}(x)_i^T w_j}{\|\mathcal{F}(x)_i\| \|w_j\|} \quad (2)$$

## Experiment and Results

Six few-shot methods were used. The experiment setup is discussed in detail here. The various evaluation metrics used are also introduced. This section also includes the presentation of the results and detailed analysis.

### Experiment Setup

The dataset of 172 P&IDs was split into training and test sets, which contained 155 and 16 diagrams respectively. Note that one diagram was mislabeled and therefore not used.

The patch-based approach detailed above was then used to split the 155 training diagrams into 16, 488 patches. Only those patches labeled with one or more symbols, 3, 984 patches, were included in the training dataset. Using the same method, the test diagrams were split into 5, 888 patches. Here, a patch overlap of 320 pixels was used to ensure that all symbols were fully contained within a patch.

Following the FSOD setting, the symbol classes,  $C$ , were split into base classes  $C_{base}$  and novel classes  $C_{novel}$  such that  $C_{base} \cap C_{novel} = \emptyset$ . The 7 least represented symbols were chosen as the novel classes and the remaining 18 symbols as the base classes. This results in a base to novel class ratio of 2.6:1, which is similar to the 3:1 ratio used in the common FSOD benchmarks (Kang et al. 2019; Wang et al. 2020). Note that following the setup in TFA, if a patch contained both base and novel classes, then the novel class annotations were not utilized in base training.

The model was trained using a batch size of 8 instead of the default 16. The linear scaling rule (Goyal et al. 2017), which states that the learning rate should be multiplied by  $k$  when the minibatch size is multiplied by  $k$ , was used to set the learning rate to 0.01 in base training and 0.0005 in fine-tuning. The model was trained for 17 epochs in base training and 3200 epochs in fine-tuning, following the settings in TFA (Wang et al. 2020). There were 200 warmup iterations. The novel class weights for the box prediction networks were randomly initialized prior to fine-tuning. Multiscale training was used to improve model performance on symbols of different sizes. Here the patch size,  $x$ , was selected such that  $x \in \{480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800\}$ . The probability of horizontal and vertical flip was set to zero, to ensure that the high amount of text-containing symbols remain realistic.

The number of shots,  $K$ , of novel classes was set to  $K = 1, 2, 3, 5, 9$ . Note that this is similar to the setting in the PASCAL VOC benchmark, however

here the maximum  $K$  value was set at nine as there were insufficient instances of each class in the diagram dataset to use ten shots. To ensure fair comparison between all the model architectures, the  $K$  shots were randomly selected and fixed across the experiments.

Six methods were evaluated, as shown in [Table 1](#). The first, TFA ( $R$ -101), was the baseline few-shot method TFA with a ResNet-101 backbone. In the second method, TFA ( $R$ -50), a smaller network, ResNet-50, was used as the backbone. Both networks were pre-trained using ImageNet (Deng et al. 2009), which is a large-scale dataset designed for image classification.

In the third method, Balanced, an undersampled training set comprising of a few shots for all 25 classes was used. By this definition, all classes are considered novel and there are no base classes. Here the balanced dataset of  $K$  shots was used to train the whole model, similar to the base training step shown in [Figure 3](#). No fine-tuning phase was used. The balanced models were trained for longer in the first step, specifically for 100 epochs, by which point the models had converged.

Three other methods were evaluated to determine the impact of unfreezing specific model components in the fine-tuning stage, as shown in [Table 1](#). In the first of these methods, Few-Shot (FT all), the entire model was unfrozen and fine-tuned. This allows the model’s feature extraction and classification parameters to be updated based on information from both the base and novel classes. Next, in Few-Shot (FT ROI + box), the backbone and RPN were frozen whilst the ROI and box predictor components were fine-tuned. Here, the model backbone is frozen in order to retain a wider range of features, that were learned during training on the more diverse dataset of base classes. The RPN is also frozen to determine if the additional information learned from the dataset of novel classes can be used to generate a more beneficial set of region proposals within the drawing. Lastly, in the proposed method, FS-Symbol, the model backbone was frozen and all other components were fine-tuned. The reasoning behind this is to preserve the model backbone that learned from the more diverse set of base classes, whilst the RPN, ROI and box predictor are then fine-tuned with the aim to provide more accurate predictions on both the base and novel classes.

**Table 1.** Method training settings in the fine-tuning phase.

Method	Fine-tuned components			
	Backbone	RPN	ROI	Box Predictor
TFA ( $R$ -101)				✓
TFA ( $R$ -50)				✓
Balanced	-	-	-	-
Few-Shot (FT all)	✓	✓	✓	✓
Few-Shot (FT ROI + box)			✓	✓
FS-Symbol		✓	✓	✓

The inference was carried out on individual test patches, and the results were combined. Non-Maximum Suppression was used to handle the overlapping predictions that occurred as a result of the patches strategy.

The methods were based on the official TFA implementation.<sup>3</sup> All experiments were carried out using a NVIDIA Quadro RTX5000 16GB GPU with 256GB RAM.

### Evaluation Metrics

Object detection models are commonly evaluated using the mean Average Precision (mAP). This is the mean of the Average Precision (AP) across all classes (3). Here  $AP_i$  is the AP of the  $i$ -th class and  $C$  is the total number of classes. The AP values were calculated according to the all-point interpolation method (Everingham et al. 2010). In FSOD, separate metrics are used for the base classes AP (bAP) and novel classes AP (nAP). In this paper, mAP, bAP and nAP were all used. An open-source toolkit for object detection metrics created by Padilla et al. (2021) was used to perform the calculation.

$$mAP = \frac{1}{C} \sum_{i=1}^C AP_i \quad (3)$$

The methods were also evaluated on a per-class basis using the Recall. The Recall is the fraction of test instances that are True Positives (TP) (4). In all the experiments, the Intersection Over Union (IOU) threshold for a true positive was set at 0.5. IOU is the ratio of the intersection to the union of the bounding boxes of the prediction and the ground truth, as defined in (5).

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Intersection\ Over\ Union = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (5)$$

### Results and Discussion

The various few-shot methods were evaluated at each  $K$  value using the nAP, bAP and mAP, as presented in Table 2. The results show that the performance typically improves with the  $K$  value, however in certain cases increasing  $K$  results in a slight performance decrease. For instance, this is seen when comparing the nAP obtained by the TFA (R-101) method at  $K = 1$ , which was 10.0, and that at  $K = 2$ , which was 8.4. Also using the baseline method, an nAP of 27.6 at  $K = 3$  is obtained however the nAP reduces to 24.4 using  $K = 5$ . This finding is potentially related to how closely the few randomly selected training instances represent the test data.

**Table 2.** Few-shot detection performance on the test diagrams for novel symbols (nAP), base symbols (bAP) and all symbols (mAP). Highest performance at each shot is in bold.

Method/Shot	nAP					bAP					mAP				
	1	2	3	5	9	1	2	3	5	9	1	2	3	5	9
TFA ( <i>R</i> -101)	10.0	8.4	27.6	24.4	43.0	<b>97.3</b>	<b>97.8</b>	97.6	<b>98.3</b>	<b>98.2</b>	72.9	72.8	78.0	77.6	82.8
TFA ( <i>R</i> -50)	1.5	15.4	22.0	24.7	29.7	97.2	97.3	97.5	97.6	97.6	70.4	74.4	76.4	77.2	78.6
Balanced	38.8	39.3	67.7	68.9	82.8	29.6	37.3	43.6	47.0	56.6	32.2	37.9	50.3	53.2	64.0
Few-Shot (FT all)	38.6	33.5	62.9	69.3	71.3	48.1	57.0	58.7	59.6	64.4	45.4	50.4	59.9	62.3	66.3
Few-Shot (FT ROI + box)	45.0	<b>44.1</b>	54.2	61.7	65.2	96.1	96.2	<b>98.0</b>	96.9	98.1	<b>81.8</b>	<b>81.6</b>	<b>85.7</b>	<b>87.0</b>	<b>88.9</b>
FS-Symbol	<b>45.1</b>	42.8	<b>68.2</b>	<b>74.8</b>	<b>83.4</b>	78.8	85.9	87.1	87.9	89.9	69.4	73.8	81.8	84.2	88.0

Note that there is a low amount of variation in the training data for novel symbols, whereas there is a large amount of intra-class variance seen in the test symbols. Comparing the results of the first two methods shows that using ResNet-50 (*R*-50) instead of ResNet-101 (*R*-101) for the model backbone negatively impacted performance across all metrics in most cases. This suggests that the additional complexity of the *R*-101 network improves the model’s ability to capture the symbol features. An *R*-101 backbone was therefore used in all further experiments.

The results also show that FS-Symbol outperforms all other methods for nAP at most  $K$  values ( $K = 1, 3, 5, 9$ ). At  $K = 2$ , FS-Symbol performance is the second highest, whilst freezing the RPN in the second training phase, Few-Shot (FT ROI + box), resulted in the highest nAP by 1.3. Across all shots, there was an increase of between 35.1 and 50.4 in nAP using FS-Symbol compared to the baseline TFA (*R*-101) method. A statistical test, the independent t-test implemented in SciPy,<sup>4</sup> was carried out to determine if the difference in novel class performance using FS-Symbol compared to the baseline TFA (*R*-101) method was significant. A  $p$ -value of 0.0045 was obtained, which is less than the alpha value of 0.05 and therefore shows a statistically significant improvement. Although FS-Symbol outperformed Few-Shot (FT ROI + Box) at most  $K$  values, the difference here was not found to be significant, as the  $p$ -value was 0.363. These results suggest that allowing the RPN and ROI to learn from novel symbol data improves the region proposals, resulting in higher performance.

The highest base class performance was obtained using the baseline TFA (*R*-101) method for  $K = 1, 2, 5, 9$ . Fine-tuning the ROI in addition to the box predictor, Few-Shot (FT ROI + box), gave the highest performance at  $K = 3$  and resulted in a small decrease of up to 1.6 bAP at other  $K$  values. These results highlight that fine-tuning the model backbone, RPN and ROI can lead to a loss of information learned in the first training stage.

The highest mAP at all shots was recorded using the Few-Shot (FT ROI + box) method. There was a statistically significant improvement using this method compared to the baseline TFA (*R*-101) method, as the  $p$ -value

obtained using the t-test was 0.008. It was also observed that using the Balanced method harmed the bAP and thus the mAP, compared to using all available base class instances. Although there was no class imbalance, the performance was inconsistent across the various classes. This highlights that additional challenges exist for symbol detection, which includes high intra-class variation, high inter-class similarity and varying symbol orientation.

Another important metric to consider for engineering symbol detection is the recall. The per-class recall, or accuracy, obtained using various few-shot methods and a YOLO-based method (Elyan, Jamieson, and Ali-Gombe 2020) trained on the fully annotated dataset was compared, as shown in Table 3. Note that more base class training samples were used for the YOLO method (Elyan, Jamieson, and Ali-Gombe 2020) as the larger image patch size,  $1250 \times 1300$  pixels compared to  $640 \times 640$  pixels, meant that more symbols appeared completely within a patch.

The results in Table 3 clearly show that the highest recall for each novel class was recorded using the FS-Symbol method. For five of the seven novel classes, a recall of 1.00 was obtained using only nine training samples per class.

**Table 3.** Comparison of few-shot and object detection method recall on test diagrams. Few-shot 1 is few-shot (FT all) and few-shot 2 is few-shot (FT ROI + box). Few-shot results reported using  $K = 9$ . Highest performance for each class is noted with asterisks and is in bold.

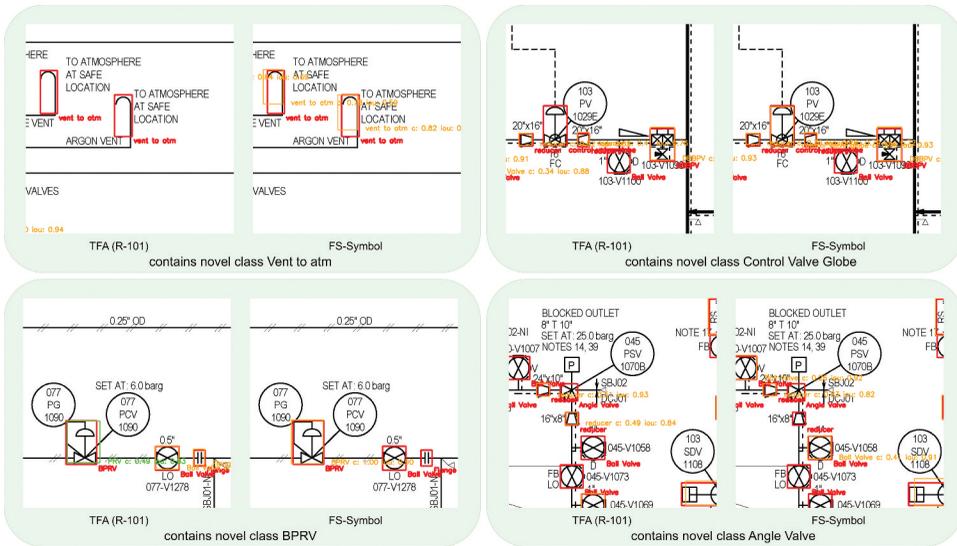
Class	Test No.	YOLO			Recall					
		Train No.	Base Train No.	Few-Shot YOLO	TFA (R-101)	TFA (R-50)	Balanced	Few-Shot 1	Few-Shot 2	FS-Symbol
Sensor	302	2810	1739	0.98	0.97	<b>*0.99*</b>	0.10	0.15	0.98	0.43
Ball Valve	213	1629	1346	<b>*0.99*</b>	0.46	0.40	0.18	0.21	0.44	0.39
Label From	103	1347	982	<b>*1.00*</b>	0.92	0.94	0.35	0.39	0.92	0.58
Label To	113	1178	828	<b>*1.00*</b>	<b>*1.00*</b>	0.99	0.35	0.50	0.99	0.80
Flange	158	1110	739	0.77	<b>*0.99*</b>	0.98	0.33	0.25	0.98	0.58
Reducer	91	821	505	0.99	<b>*1.00*</b>	<b>*1.00*</b>	0.68	0.70	<b>*1.00*</b>	0.77
DB&BBV	67	542	469	<b>*0.98*</b>	0.96	0.96	0.36	0.31	0.96	0.58
Gate Valve	110	535	429	0.94	<b>*1.00*</b>	<b>*1.00*</b>	0.51	0.61	<b>*1.00*</b>	0.91
Check Valve	42	396	335	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	0.40	0.38	<b>*1.00*</b>	0.74
TOB/Butterfly Valve	59	178	168	0.98	<b>*1.00*</b>	<b>*1.00*</b>	0.38	0.62	<b>*1.00*</b>	<b>*1.00*</b>
Plug Valve	8	173	154	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	0.80	0.81	<b>*1.00*</b>	<b>*1.00*</b>
Globe Valve	7	161	150	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	0.90	<b>*1.00*</b>	<b>*1.00*</b>
Needle Valve	10	160	133	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	0.71	0.86	<b>*1.00*</b>	<b>*1.00*</b>
RS	26	143	114	<b>*0.92*</b>	0.88	<b>*0.92*</b>	0.65	0.77	0.88	0.85
PSV	25	118	94	<b>*0.88*</b>	0.74	0.78	0.17	0.39	0.83	0.39
Eccentric Reducer	23	98	92	0.96	<b>*1.00*</b>	0.88	0.76	0.84	<b>*1.00*</b>	0.88
POB Valve	16	84	65	<b>*1.00*</b>	0.94	0.94	0.69	0.62	0.94	0.94
DBBPV	15	83	65	<b>*1.00*</b>	0.93	<b>*1.00*</b>	0.87	0.93	0.93	<b>*1.00*</b>
PRV	8	32	0	<b>1.00</b>	0.83	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>
Control Valve Globe	6	30	0	<b>*1.00*</b>	0.88	0.88	<b>*1.00*</b>	<b>*1.00*</b>	0.62	<b>*1.00*</b>
Control Valve	5	22	0	<b>*1.00*</b>	<b>*1.00*</b>	0.00	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>
Vent to atm	8	19	0	0.25	0.00	0.00	0.62	0.50	<b>*0.88*</b>	<b>*0.88*</b>
Injection/Sample Point	2	13	0	0.50	0.00	0.00	<b>*1.00*</b>	<b>*1.00*</b>	0.50	<b>*1.00*</b>
Angle Valve	2	11	0	0.00	<b>*0.50*</b>	<b>*0.50*</b>	<b>*0.50*</b>	0.00	<b>*0.50*</b>	<b>*0.50*</b>
BPRV	5	11	0	0.00	0.20	0.00	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>	<b>*1.00*</b>

Furthermore, the recall for the four least represented classes increased considerably compared to that obtained with the YOLO-based method. For instance, for the Vent to atm symbol, a recall of 0.88 was recorded using FS-Symbol compared to 0.25 using the YOLO-based method. Note that nine samples of this symbol were used to train the few-shot methods whereas nineteen were used to train YOLO. Also evident is the need to preserve features learned from the data-abundant base classes, which is shown by analyzing the performance of the Few-Shot (FT all) method. For instance, a recall of 0.00 was obtained for the novel class Angle Valve, compared to 0.50 with TFA (*R*-101). This suggests that fine-tuning the whole model results in a loss of information learned in the first training stage.

Further evidence for the validity of the FS-Symbol training approach can be seen by comparing the results obtained with those using TFA (*R*-101). For example, the latter method did not detect the classes Vent to atm and Injection/Sample Point, however recall values of 0.88 and 1.00 were recorded using FS-Symbol. These are the only two novel classes that are not valves and as such they are more visually distinct from the base classes compared to the other novel classes, refer to [Figure 2](#). This indicates that fine-tuning only the box predictor on novel class data may not be sufficient when there is a large difference in symbol appearance between the novel and base classes.

Base class performance was typically higher using the YOLO-based method compared to the FS-Symbol method. There were only five classes for which equal or higher recall was recorded using the latter method. Another finding is that for certain base classes, competitive recall was recorded using several of the few-shot methods compared to the YOLO-based method. For example, for the sensor symbol, a recall of 0.98 was recorded using the YOLO-based method, compared to 0.97, 0.99 and 0.98 using TFA (*R*-101), TFA (*R*-50) and Few Shot (FT ROI + box), respectively.

The performance of FS-Symbol compared to TFA (*R*-101) can also be observed in [Figure 4](#). In these processed test patches, the ground truth bounding boxes are shown in red, correct predictions in orange and incorrect predictions in green. These patches show the improved performance of FS-Symbol on the novel classes. For example, the vent to atm symbols shown were correctly predicted by FS-Symbol but not the TFA (*R*-101) method. Another example shows a BPRV symbol that was successfully detected using FS-Symbol, but predicted as a PRV by TFA (*R*-101). As these two classes contain the same shapes but with different orientations, see [Figure 2](#), this suggests that training the RPN and ROI on novel data has improved the method's discriminative ability between similar symbols. The test patches also indicate that base class performance was higher using TFA (*R*-101) compared to FS-Symbol. For example, they contain instances of reducer, ball valve and flange symbols detected by the first method but not the latter. Overall, these results suggest that the training approach implemented in FS-Symbol improves the model's ability to detect novel classes.



**Figure 4.** Small sections of test diagrams. In each image pair, the left image was processed using TFA (R-101), and the right image was processed using FS-Symbol. Both methods used  $K = 9$ . Ground truth bounding boxes are shown in red, correct predictions in orange and incorrect predictions in green. The confidence and IOU are also shown.

## Conclusion and Future Direction

In this paper, one of the first approaches to the problem of few-shot symbol detection in engineering diagrams is presented. The method can be used to detect rare classes using fewer than ten training samples. Furthermore, this approach allows new symbols to be incorporated into an existing object detector without extensive annotation. Thorough experiments on complex engineering diagrams sourced from industry were completed to demonstrate the validity of the proposed method.

Various few-shot methods were evaluated and the results show that the highest performance on the novel classes was obtained using the proposed approach. Statistically significant improvement compared to the baseline few-shot method was also shown. The method was also compared against an object detection-based method trained on a dataset of fully annotated diagrams, and improved novel class performance was reported.

The research also showed limitations of few-shot methods for symbol digitization. The main drawback seen with the proposed method is a reduction in performance for the majority of the base classes. To obtain competitive performance on all classes, separate models could be used to predict base and novel classes. An alternative suggestion is to alter the model so it contains separate branches to predict the base and novel categories. The model backbone would be shared between the two branches and there would be separate RPN, ROI and box predictor modules. In this approach, the fine-tuning step would alter the novel

class branch, whilst the base class branch would remain fixed and thus performance for the base classes will not be reduced due to the fine-tuning.

The method required a relatively large number of labeled base class instances. To reduce the annotation effort, semi-supervised learning techniques could be used. This involves manually labeling a small subset of available base symbols to train a detector that can then be used to create pseudo labels and increase the training dataset size.

Another limitation of the method is that it relies on very few samples of the novel classes and thus the performance of the model is likely to be linked to how well the chosen training samples represent the test data. This may be particularly important for symbol classes with high intra-class variation. To increase the variance of the novel symbols, data augmentation techniques could be used, such as resizing, orientation, mosaic data augmentation and random cropping. This would also allow the results to be reported over multiple random runs.

An additional future work suggestion is to experiment with a wider variety of backbones, increasing the number of layers with the aim of improving the generalizability from the base classes to the novel classes. Another aim is to demonstrate the robustness of the approach on more diverse datasets. This includes applying the method to a range of engineering drawing types, including construction drawings, process flow diagrams and architectural drawings. It should be noted that this may be challenging due to the difficulty in acquiring and annotating engineering drawings.

Overall this paper opens up a new direction toward using few shot approaches for engineering symbol digitization, which is highly beneficial for rare symbols and also reduces the required annotation effort.

## Notes

1. <https://sloth.readthedocs.io/en/latest/>.
2. <https://github.com/open-cv/cvat>.
3. <https://github.com/ucbdrive/few-shot-object-detection>.
4. <https://scipy.org/>.

## Disclosure Statement

No potential conflict of interest was reported by the author(s).

## Data Availability Statement

The data that support the findings of this study are available from the corresponding author, LJ, upon reasonable request.

## References

- Antonelli, S., D. Avola, L. Cinque, D. Crisostomi, G. Luca Foresti, F. Galasso, M. Raoul Marini, A. Mecca, and D. Pannone. 2022. Few-shot object detection: A survey. *ACM Computing Surveys* 54 (11s):1–37. doi: [10.1145/3519022](https://doi.org/10.1145/3519022).
- Bhanbhro, H., Y. Kwang Hooi, W. Kusakunniran, and Z. H. Amur. 2023. A symbol recognition system for single-line diagrams developed using a deep-learning approach. *Applied Sciences* 13 (15):15. doi: [10.3390/app13158816](https://doi.org/10.3390/app13158816).
- Bochkovskiy, A., C.-Y. Wang, and H.-Y. Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv Preprint arXiv: 2004.10934*. doi: [10.48550/arXiv.2004.10934](https://doi.org/10.48550/arXiv.2004.10934).
- Bolelli, F., S. Allegretti, and C. Grana. 2022. One DAG to rule them all. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (7):3647–58. doi: [10.1109/TPAMI.2021.3055337](https://doi.org/10.1109/TPAMI.2021.3055337).
- Buda, M., A. Maki, and M. A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106:249–59. doi: [10.1016/j.neunet.2018.07.011](https://doi.org/10.1016/j.neunet.2018.07.011).
- Cheng, G., B. Yan, P. Shi, K. Li, X. Yao, L. Guo, and J. Han. 2021. Prototype-cnn for few-shot object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 60:1–10. doi: [10.1109/TGRS.2021.3078507](https://doi.org/10.1109/TGRS.2021.3078507).
- Deng, J., W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–55, June. doi: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- Elyan, E., L. Jamieson, and A. Ali-Gombe. 2020. Deep learning for symbols detection and classification in engineering drawings. *Neural Networks* 129:91–102. doi: [10.1016/j.neunet.2020.05.025](https://doi.org/10.1016/j.neunet.2020.05.025).
- Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88 (2):303–38. doi: [10.1007/s11263-009-0275-4](https://doi.org/10.1007/s11263-009-0275-4).
- Faltin, B., D. Gann, and M. König. 2023. *A comparative study of deep learning models for symbol detection in technical drawings*, 877–86. Firenze University Press. doi: [10.36253/979-12-215-0289-3.87](https://doi.org/10.36253/979-12-215-0289-3.87).
- Faltin, B., P. Schönfelder, and M. König. 2022. Inferring interconnections of construction drawings for bridges using deep learning-based methods. *ECPPM 2022 - eWork and eBusiness in Architecture, Engineering and Construction 2022* (9):343–50. doi: [10.1201/9781003354222-44](https://doi.org/10.1201/9781003354222-44).
- Fan, Z., Y. Ma, Z. Li, and J. Sun. 2021. Generalized few-shot object detection without forgetting. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4527–36. doi: [10.1109/CVPR46437.2021.00450](https://doi.org/10.1109/CVPR46437.2021.00450).
- Gao, W., Y. Zhao, and C. Smidts. 2020. Component detection in piping and instrumentation diagrams of nuclear power plants based on neural networks. *Progress in Nuclear Energy* 128:103491. doi: [10.1016/j.pnucene.2020.103491](https://doi.org/10.1016/j.pnucene.2020.103491).
- Goyal, P., P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv Preprint arXiv: 1706.02677*. doi: [10.48550/arXiv.1706.02677](https://doi.org/10.48550/arXiv.1706.02677).
- Groen, F. C., A. C. Sanderson, and J. F. Schlag. 1985. Symbol recognition in electrical diagrams using probabilistic graph matching. *Pattern recognition letters. Pattern Recognition Letters* 3 (5):343–50. doi: [10.1016/0167-8655\(85\)90066-2](https://doi.org/10.1016/0167-8655(85)90066-2).
- Gupta, M., C. Wei, and T. Czerniawski. 2024. Semi-supervised symbol detection for piping and instrumentation drawings. *Automation in Construction* 159:105260. doi: [10.1016/j.autcon.2023.105260](https://doi.org/10.1016/j.autcon.2023.105260).

- He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–78, June. doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- Hou, S., W. Liu, A. Karim, Z. Jia, W. Jia, and Y. Zheng. 2023. Few-shot logo Detection.” *IET computer vision. IET Computer Vision* 17 (5):586–98. doi: [10.1049/cvi2.12205](https://doi.org/10.1049/cvi2.12205).
- Jakubik, J., P. Hemmer, M. Vössing, B. Blumenstiel, A. Bartos, and K. Mohr. 2022. Designing a human-in-the-loop system for object detection in floor plans. *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (11):12524–30. doi:[10.1609/aaai.v36i11.21522](https://doi.org/10.1609/aaai.v36i11.21522).
- Jamieson, L., C. F. Moreno-Garcia, and E. Elyan. 2020. Deep learning for text detection and recognition in complex engineering diagrams. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. doi: [10.1109/IJCNN48605.2020.9207127](https://doi.org/10.1109/IJCNN48605.2020.9207127).
- Jocher, G., A. Chaurasia, and J. Qiu. 2023. YOLO by Ultralytics. <https://github.com/ultralytics/ultralytics>.
- Jocher, G., K. Nishimura, T. Mineeva, and R. Vilariño. 2020. yolov5. <https://github.com/ultralytics/yolov5>.
- Johnson, J. M., and T. M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6 (1):27. doi: [10.1186/s40537-019-0192-5](https://doi.org/10.1186/s40537-019-0192-5).
- Kang, B., Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. 2019. Few-shot object detection via feature reweighting. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8420–29. doi: [10.1109/ICCV.2019.00851](https://doi.org/10.1109/ICCV.2019.00851).
- Kaul, P., W. Xie, and A. Zisserman. 2022. Label, verify, correct: A simple few shot object detection method. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14237–47, June. doi: [10.1109/CVPR52688.2022.01384](https://doi.org/10.1109/CVPR52688.2022.01384).
- Köhler, M., M. Eisenbach, and H.-M. Gross. 2023. Few-shot object detection: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems* 1–21. doi: [10.1109/TNNLS.2023.3265051](https://doi.org/10.1109/TNNLS.2023.3265051).
- LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521 (7553):436. doi: [10.1038/nature14539](https://doi.org/10.1038/nature14539).
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86 (11):2278–324. doi: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- Li, C., L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, and Y. Li. 2022. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv*. doi: [10.48550/ARXIV.2209.02976](https://doi.org/10.48550/ARXIV.2209.02976).
- Lin, T.-Y., P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. 2016. Feature pyramid networks for object detection. *CoRR Abs/1612.03144*. doi: [10.1109/CVPR.2017.106](https://doi.org/10.1109/CVPR.2017.106).
- Lin, T.-Y., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, C. L. Z. Deva Ramanan, and P. Dollár. 2015. Microsoft COCO: Common objects in context. *arXiv*. doi: [10.48550/arXiv.1405.0312](https://doi.org/10.48550/arXiv.1405.0312).
- Liu, T., L. Zhang, Y. Wang, J. Guan, Y. Fu, J. Zhao, and S. Zhou. 2023. Recent few-shot object detection algorithms: A survey with performance Comparison. *ACM Transactions on Intelligent Systems and Technology* 14 (4):1–36. doi: [10.1145/3593588](https://doi.org/10.1145/3593588).
- Mani, S., M. A. Haddad, D. Constantini, W. Douhard, Q. Li, and L. Poirier. 2020. Automatic digitization of engineering diagrams using deep learning and graph search. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 673–79. doi: [10.1109/CVPRW50498.2020.00096](https://doi.org/10.1109/CVPRW50498.2020.00096).
- Moreno-Garcia, C. F., and E. Elyan. 2019. Digitisation of assets from the oil and gas industry: Challenges and opportunities. *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)* 7, 2–5. doi: [10.1109/ICDARW.2019.60122](https://doi.org/10.1109/ICDARW.2019.60122).
- Moreno-García, C. F., E. Elyan, and C. Jayne. 2019. New trends on digitisation of complex engineering drawings. *Neural Computing & Applications* 31 (6):1695–712. doi: [10.1007/s00521-018-3583-1](https://doi.org/10.1007/s00521-018-3583-1).

- Okazaki, A., T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto. 1988. An automatic circuit diagram reader with loop-structure-based symbol recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 10 (3):331–41. doi: [10.1109/34.3898](https://doi.org/10.1109/34.3898).
- Padilla, R., W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva. 2021. A comparative analysis of object detection metrics with a Companion open-source toolkit. *Electronics* 10 (3):3. doi: [10.3390/electronics10030279](https://doi.org/10.3390/electronics10030279).
- Paliwal, S., J. Arushi, S. Monika, and V. Lovekesh. 2021. Digitize-pid: Automatic digitization of piping and instrumentation diagrams. *Trends and Applications in Knowledge Discovery and Data Mining - PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings, edited by Manish Gupta and Ganesh Ramakrishnan, Vol. 12705 of Lecture Notes in Computer Science*, 168–80, Springer. doi: [10.1007/978-3-030-75015-2\\_17](https://doi.org/10.1007/978-3-030-75015-2_17).
- Paliwal, S., M. Sharma, and L. Vig. 2021. OSSR-PID: One-shot symbol recognition in P & ID sheets using path sampling and GCN. *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. doi: [10.1109/IJCNN52387.2021.9534122](https://doi.org/10.1109/IJCNN52387.2021.9534122).
- Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–88. doi: [10.1109/CVPR.2016.91](https://doi.org/10.1109/CVPR.2016.91).
- Redmon, J., and A. Farhadi. 2017. YOLO9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–25, July. doi: [10.1109/CVPR.2017.690](https://doi.org/10.1109/CVPR.2017.690).
- Redmon, J., and A. Farhadi. 2018. YoloV3: An incremental improvement. *arXiv Preprint arXiv:180402767*. doi: [10.48550/arXiv.1804.02767](https://doi.org/10.48550/arXiv.1804.02767).
- Ren, S., K. He, R. Girshick, and J. Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15*, 91–99, Cambridge, MA, USA: MIT Press. doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031).
- Ruzicka, V., and F. Franchetti. 2018. Fast and accurate object detection in high resolution 4K and 8K video using GPUs. *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 1–7, IEEE. doi: [10.1109/HPEC.2018.8547574](https://doi.org/10.1109/HPEC.2018.8547574).
- Schlagenhauf, T., M. Netzer, and J. Hillinger. 2023. Text detection on technical drawings for the digitization of brown-field processes. *Procedia CIRP 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering*, 118, 372–77. doi: [10.1016/j.procir.2023.06.064](https://doi.org/10.1016/j.procir.2023.06.064).
- Theisen, M. F., K. N. Flores, L. S. Balhorn, and A. M. Schweidtmann. 2023. Digitization of chemical process flow diagrams using deep convolutional neural networks. *Digital Chemical Engineering* 6:100072. doi: [10.1016/j.dche.2022.100072](https://doi.org/10.1016/j.dche.2022.100072).
- Wang, C.-Y., A. Bochkovskiy, and H.-Y. Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. doi: [10.48550/ARXIV.2207.02696](https://doi.org/10.48550/ARXIV.2207.02696).
- Wang, X., T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu. 2020. Frustratingly simple few-shot object detection. *Proceedings of the 37th International Conference on Machine Learning. ICML'20. JMLR.org*. [10.48550/arXiv.2003.06957](https://arxiv.org/abs/2003.06957).
- Yu, E.-S., J.-M. Cha, T. Lee, J. Kim, and D. Mun. 2019. Features recognition from piping and instrumentation diagrams in image format using a deep learning network. *Energies* 12 (23):4425. doi: [10.3390/en12234425](https://doi.org/10.3390/en12234425).
- Zhao, Y., X. Deng, and H. Lai. 2020. A deep learning-based method to detect components from scanned structural drawings for reconstructing 3D models. *Applied Sciences* 10 (6):6. doi: [10.3390/app10062066](https://doi.org/10.3390/app10062066).
- Zhao, Y., X. Deng, and H. Lai. 2021. Reconstructing BIM from 2D structural drawings for existing buildings. *Automation in Construction* 128:103750. doi: [10.1016/j.autcon.2021.103750](https://doi.org/10.1016/j.autcon.2021.103750).