

Towards fully automated processing and analysis of construction diagrams: AI-powered symbol detection.

JAMIESON, L., MORENO-GARCIA, C. and ELYAN, E.

2024

© The Author(s) 2024. The version of record of this article, first published in *International Journal on Document Analysis and Recognition*, is available online at Publisher's website:
<https://doi.org/10.1007/s10032-024-00492-9>



Towards fully automated processing and analysis of construction diagrams: AI-powered symbol detection

Laura Jamieson¹ · Carlos Francisco Moreno-Garcia¹ · Eyad Elyan¹

Received: 30 October 2023 / Revised: 26 June 2024 / Accepted: 15 July 2024
© The Author(s) 2024

Abstract

Construction drawings are frequently stored in undigitised formats and consequently, their analysis requires substantial manual effort. This is true for many crucial tasks, including material takeoff where the purpose is to obtain a list of the equipment and respective amounts required for a project. Engineering drawing digitisation has recently attracted increased attention, however construction drawings have received considerably less interest compared to other types. To address these issues, this paper presents a novel framework for the automatic processing of construction drawings. Extensive experiments were performed using two state-of-the-art deep learning models for object detection in challenging high-resolution drawings sourced from industry. The results show a significant reduction in the time required for drawing analysis. Promising performance was achieved for symbol detection across various classes, with a mean average precision of 79% for the YOLO-based method and 83% for the Faster R-CNN-based method. This framework enables the digital transformation of construction drawings, improving tasks such as material takeoff and many others.

Keywords Deep learning · Digitisation · Symbol detection · Engineering drawings · Convolutional neural networks · Artificial intelligence

1 Introduction

Construction drawings are essential documents as they show what will be built for a project. These drawings are still frequently stored in undigitised formats, and consequently, retrieving information from them must be carried out manually. This requires domain experts [1], and can be very time-consuming [2] and costly.

One of the most important processes in a construction project is material takeoff or quantity takeoff [1, 3]. The purpose of this task is to create a list of the required materials and quantities. The list is an essential document as it is used for cost estimation [3]. It is important that it is accurate, as any errors can impact the project budget and schedule [4, 5]. The

takeoff is traditionally carried out through manual drawing analysis. However, this can be time-consuming and prone to counting errors, particularly for large projects [2]. Furthermore, the results are dependent on individual interpretations [4].

Artificial Intelligence (AI) based methods can augment employees' capabilities by simplifying time-intensive and repetitive tasks [6]. The use of state-of-the-art digital technologies to transform traditional industry practices into autonomous systems has been referred to as Industry 4.0, or the fourth industrial revolution [7]. The number of publications that discussed AI applications in the construction industry has increased in recent years [8]. Within this, one of the main research topics was computer vision [8], where the applications mainly focussed on the monitoring of construction sites and structural health. However, the current adoption of AI-based applications in the building and construction industry is relatively low [7], and lags behind that in other industries [8, 9].

Across a range of sectors, there has recently been an increasing demand to create methods to digitise engineering diagrams [10, 11]. This involves extracting all diagram components, which are the symbols, text and lines. Although

✉ Laura Jamieson
l.jamieson4@rgu.ac.uk

Carlos Francisco Moreno-Garcia
c.moreno-garcia@rgu.ac.uk

Eyad Elyan
e.elyan@rgu.ac.uk

¹ School of Computing, Robert Gordon University, Garthdee Road, Aberdeen AB10 7QB, Scotland, UK

published research on this topic dates back to the 1980s [12, 13], as stated in a recent review by Moreno-Garcia et al. [14], digitising complex engineering diagrams is still considered challenging. For instance, deep learning methods were used for symbol digitisation in other types of engineering diagrams [15, 16]. This was considered a difficult task for multiple reasons, including the numerous symbols present in each diagram [15], relatively small symbol sizes [15, 16] and use of non-standard symbols [17, 18].

Although construction drawings are more complex compared to other types of engineering drawings, methods for their digitisation have received considerably less attention than those for other engineering drawing types, such as Piping and Instrumentation Diagrams (P&IDs) [10, 15–17, 19–21]. One reason that construction drawings are more complex is that they are typically composed of multiple drawing layers. These organise graphical elements by type [22], and can be shown overlapping each other. This means that symbols are typically shown on a highly complex background. Additionally, these drawings contain a significant amount of visually similar shapes. Furthermore, they are typically grayscale and thus, no colour information is available to help distinguish between components.

This paper presents a novel deep learning framework to process construction drawings automatically. It should be noted that engineering diagrams are generally unavailable in the public domain [11, 14] due to confidentiality reasons. Therefore, in this experiment, a dataset was obtained from an industry partner to ensure that the research is relevant to a real-world scenario. Multiple building systems are represented, including plumbing and Heating, Ventilation and Air Conditioning (HVAC). The drawings are very complex and contain various symbol classes, typically shown on a cluttered background, as shown in Fig. 1.

The main contributions of this paper are outlined as follows:

- A novel framework for the automatic processing of construction drawings is presented. This automatically detects symbols for the material takeoff, resulting in significant time-saving compared to manual drawing analysis.
- Extensive set of experiments has been carried out using a large dataset of challenging high-resolution drawings of different qualities. Various symbol classes were used, which have high levels of intra-class variability and inter-class similarity. This is believed to be the first example of these experiments using complex construction drawings from industry.
- Two state-of-the-art deep learning models were utilised for symbol detection in construction drawings. This allows for a comparative analysis of the performance and

speed between two object detection architecture types, one-stage and two-stage.

The rest of this paper is structured as follows. Section 2 critically discusses the related work in symbol digitisation in complex engineering drawings with a focus on construction drawings. Next, the proposed methods are discussed in Sect. 3. The experiments and results are presented in Sect. 4. Finally, the conclusion and future direction are provided in Sect. 5.

2 Related work

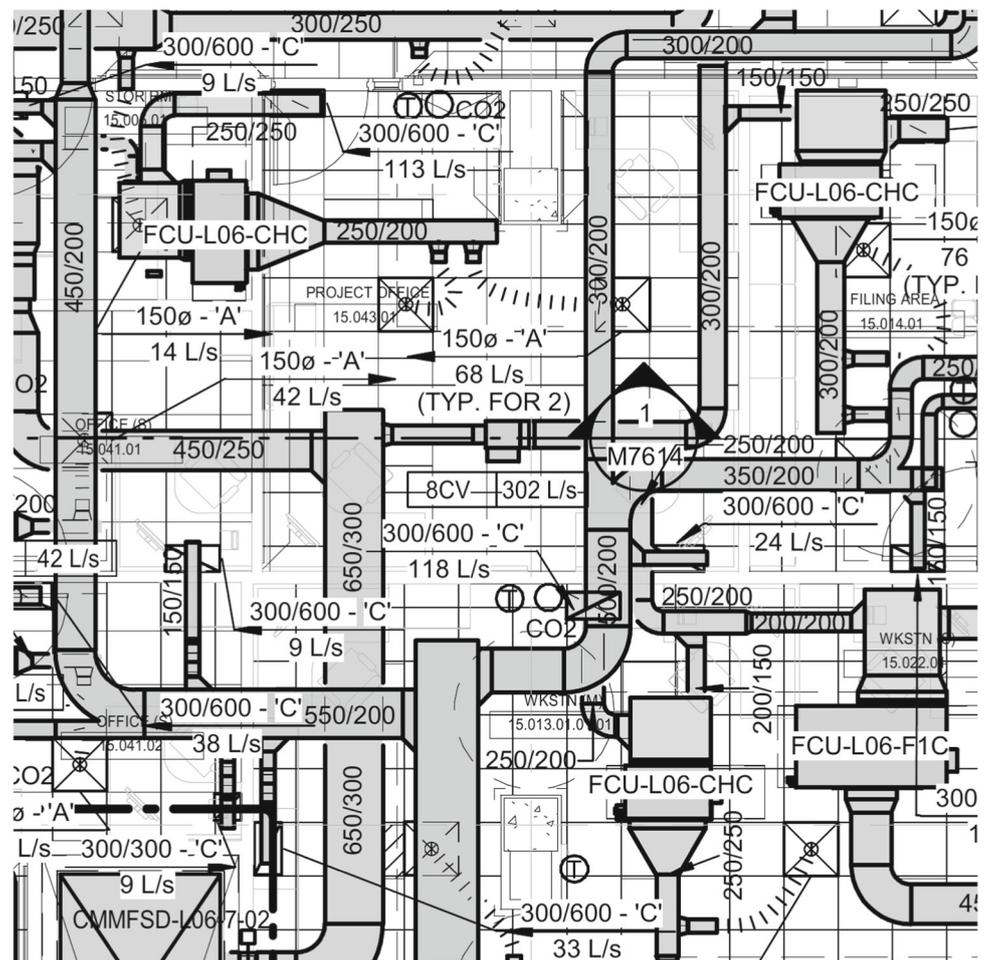
Methods to automate engineering diagram analysis have been presented since the 1980s [13, 23]. Most of these methods were based on traditional image processing approaches [14]. These rely on pre-established rules, which results in weak generalisation capability [10, 24]. Such approaches struggled to perform well across the wide range of variations present in engineering drawings, including image quality [25], object orientations [14], and overlapping objects [14].

More recently, Convolutional Neural Network (CNN) based deep learning models have significantly improved on traditional computer vision methods, including object detection, segmentation and classification [26]. Significant improvement has been seen since 2012 when the AlexNet [27] CNN-based classification model was shown to outperform previous methods by a large margin. Since then, method improvements have been facilitated by algorithm developments, together with an increase in data and computing power.

Despite this progress, deep learning methods for engineering diagram digitisation were only proposed very recently. Symbol digitisation methods were mostly based on object detection models, which predict the class and bounding box of target objects in an image. Most research focussed on one type of object detector, such as You Only Look Once (YOLO) [28] based approaches [11, 15, 24, 29–32] or Faster Region-based Convolutional Neural Network (Faster R-CNN) [33] based approaches [2, 6, 34, 35, 55]. Other approaches were based on Fully Convolutional Network (FCN) [37] segmentation models [38, 39] or graph-based methods [40–42].

Deep learning methods generally showed improvements compared to traditional approaches, although it is clear from the published literature that there are remaining challenges [43]. For instance, due to the lack of real world engineering drawing datasets in the public domain, much of the existing research has been carried out using small datasets or simplified drawings [43]. Furthermore, deep learning models typically require a large annotated dataset which is very time-consuming to obtain for these drawings [44]. Another challenge is that of imbalanced datasets, which results in the

Fig. 1 Section of an ‘HVAC’ drawing. This is challenging to digitise for multiple reasons, including the dense representation of equipment, overlapping components, and complex background



class imbalance problem [45]. Additional challenges that still require further research are evaluation of drawing digitisation methods and contextualisation [43].

The literature on symbol digitisation in engineering drawings covers a range of drawing types, with a particular focus on P&IDs [10, 15–17, 19–21]. For instance, Elyan et al. [15] created a YOLO-based method to detect symbols in P&IDs. They reported high performance overall with an accuracy of 95%, although the results varied across the symbol classes. Meanwhile, Gao et al. [16] presented a Faster R-CNN-based symbol detection method. On a dataset of publicly available nuclear power plant drawings, they reported mAP values of 92% and above for three separate groups of symbols. In another example on P&IDs, Mani et al. [10] created a CNN-based classification method for fixed size drawing patches. They obtained promising results for two symbol classes, however this method may be computationally slow when scaled up for a larger number of classes.

There is also published research on symbol digitisation in architectural floor plans [6, 46]. For instance, Rezvanifar et al. [46] presented a YOLO-based method for symbol detection. They evaluated the method on a private dataset as well as the

public Systems Evaluation SYnthetic Documents (SESYD) dataset. On the latter, they showed that their method outperformed traditional symbol spotting approaches. In the same domain, Jakubik et al. [6] presented a human-in-the-loop approach for the detection and classification of symbols. They used a Faster R-CNN-based symbol detection method that was trained using a synthetic dataset created using a data augmentation approach.

However, there was a lack of research on construction drawings, as there are only a few recent works that presented deep learning methods for generating a list of materials from construction drawings [2, 5]. Joy and Mounsef [2] presented a Faster R-CNN [33] based method to automate material takeoff from electrical engineering plans. They used a dataset of five drawings. Training data was generated using symbols extracted from the legend and image processing-based data augmentation. Whilst the method did not require extensive manual annotation, it relied on a legend being available. Prior to testing, the background and text strings were removed. This may be particularly important here, as these components were not included in the testing data.

Chowdhury and Moon [5] presented a Mask R-CNN [47] based method to automatically generate the bill of materials (BoM), which is a list of the required item quantities and costs, from 2D images of concrete formwork. Mask R-CNN is an object segmentation model, which predicts pixel-level object masks rather than bounding boxes. They created 206 drawings from 3D models, which were relatively clean with few components. On the validation data, an mAP of 98% was reported. The method showed promising results on the test drawings, however detailed metrics were not presented. On an actual construction shop drawing, the increased complexity meant that preprocessing was required to remove unnecessary elements. The solution relied on the manual selection of relevant items within a cost database to produce the BoM.

In a related area, published work discussed automated quantity takeoff from Building Information Modelling (BIM) models [1, 48]. BIM has played the leading role in digitising the construction industry [8], and it concerns the creation of a 3D model to manage building data. The drawback of BIM-based takeoff approaches is that considerable time and resources are needed to create the BIM. Furthermore, errors in the BIM impact the accuracy of the quantity takeoff [48].

The literature shows that although deep learning has significantly improved computer vision methods, there is a lack of progress in construction drawing digitisation methods. Deep learning methods for engineering diagram digitisation were proposed only very recently, and these were primarily focussed on other engineering diagram types [10, 15–17, 19–21]. Moreover, there was a lack of research showing how different deep learning object detection models performed on complex real-world construction drawings.

3 Methods

This section discusses the methods used in the proposed symbol detection framework for complex construction drawings. This includes a discussion of the real-world dataset used for evaluation purposes.

3.1 Dataset

3.1.1 Overview

A dataset of 198 PDF construction drawings was obtained from an industrial partner. It contains three unequally represented types, as there are 92 ‘plumbing’, 103 ‘HVAC’ and 3 ‘other’ drawings. To prepare the dataset for the experiment, the PDFs were converted to high-resolution 14, 042 × 9, 934 pixel PNG images at 300 dpi, as shown in Fig. 2.

The diagrams contain numerous symbol classes, 13 of which were selected for the experiment. These were chosen

as they are required in the takeoff, and are shown in multiple building systems. The ‘Detail Legend’ and ‘Direction of Flow’ symbols were also included, as detecting them can help to determine links between diagrams or flow direction [44].

The symbols of interest are represented by various shapes, as shown in Fig. 3. It should be noted that these examples were cropped from the legend, and are thus displayed on a white background, unlike typically seen within a diagram. The symbols are challenging for a model to detect for several reasons. Each symbol is only represented by a few lines or shapes and thus there are only a few features available for a model to learn from. They are commonly represented in different orientations, with different shading and often overlap other shapes. Intra-class variability in the graphical notations was also seen, as shown in Fig. 4. Furthermore, there is high inter-class similarity, for instance, the shape that represents a Gate Valve is also part of the Automatic Control Valve (ACV) and the Valve and Capped (V&C) Provision.

3.1.2 Data annotation

The diagrams were manually annotated to create a symbol dataset, which can be a very time-consuming and demanding task [29, 44, 49]. The process includes drawing bounding boxes closely around each target symbol. For the purpose of the experiment, the diagrams were manually annotated using Sloth¹. This is an open source tool which allows for object annotation. It should be noted that the annotations are exported to one output file per diagram. These record the labelled symbol information, including the class and bounding box coordinates. In total, the symbol dataset consists of 6231 symbols from 13 classes.

3.2 Data exploration and pre-processing

Different equipment items are shown with various frequencies within construction diagrams, therefore the symbol dataset is highly imbalanced, as shown in Fig. 5. Class imbalance is a major problem in both machine and deep learning [50–52] and is when algorithms trained on an imbalanced dataset are biased towards the majority class. It was observed that the Ball Valve symbol is significantly overrepresented, as it constitutes 35.3% of the dataset. In contrast, the four least represented classes each constitute less than 1%.

The problem of small object detection was also seen in this experiment. This is considered a problem due to reasons such as limited context information and indistinguishable features [53, 54]. For example, on the COCO dataset [54], the Average Precision (AP) of YOLOv7 for small objects was lower, 35.2%, compared to that for medium objects, 56.0%, and

¹ <https://sloth.readthedocs.io/en/latest/>.

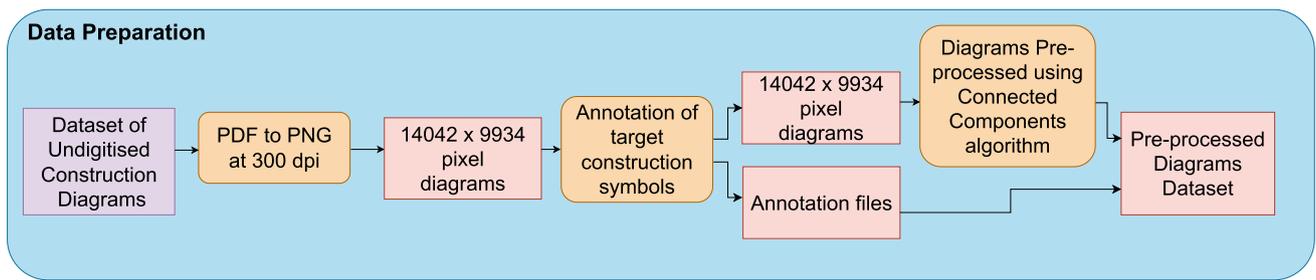


Fig. 2 Data preparation steps. The dataset of undigitised construction diagrams was pre-processed by firstly converting the PDF to PNG. Next, the image files were annotated with the target symbol classes. Finally, the drawing border was removed using a Connected Components algorithm

Fig. 3 Symbol legend

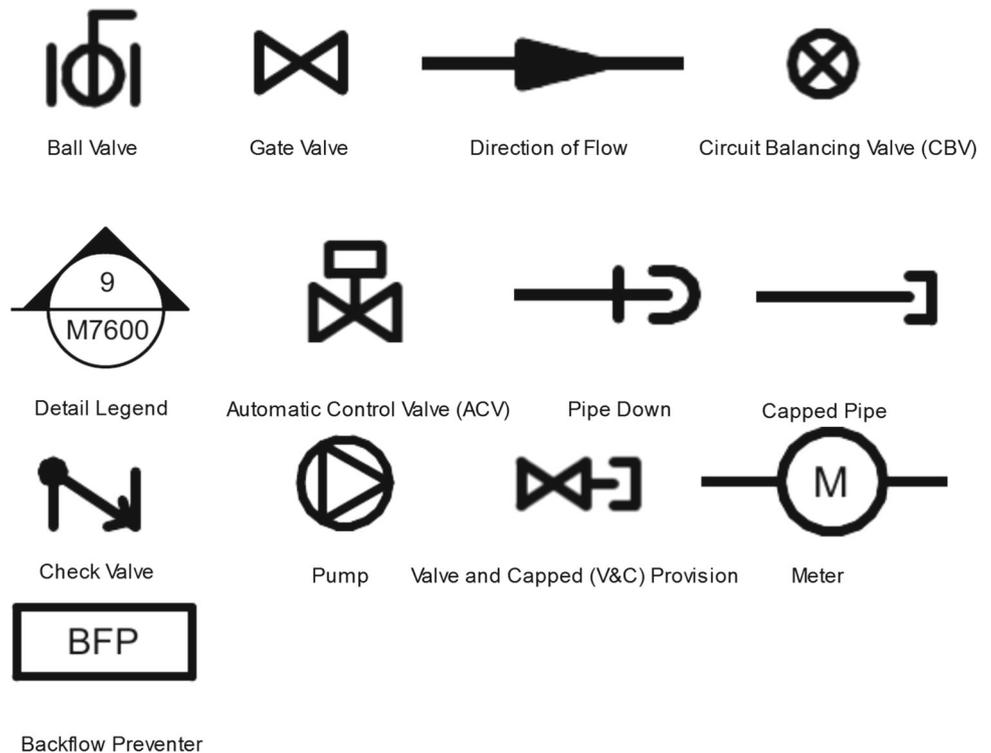
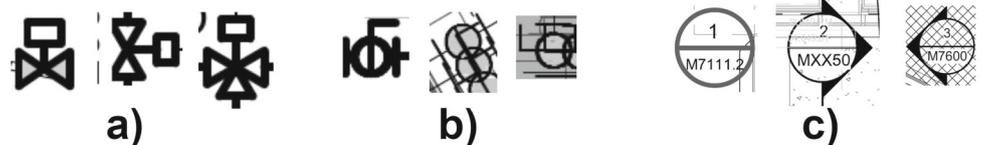


Fig. 4 Examples of intra-class variability. The symbols in each group represent the same class, which are **a** ACVs, **b** Ball valves and **c** Detail legends



large objects, 66.7% [55]. In this paper the COCO definition of object size was used. This classifies objects as small if their area was less than 32×32 pixels, medium if between 32×32 pixels and 96×96 pixels, and large if more than 96×96 pixels [56]. Most of the symbols here are small or medium sized using this criteria.

The diagrams were pre-processed to reduce false positives, as shown in Fig. 2. This was done by removing the diagram border, which contains text and no target symbols. A Connected Component (CC) algorithm was used to locate the largest CC of white pixels, which was considered to be

the background of the main diagram area. In this calculation, the pixels were defined as connected to each other if they had four-way connectivity. An image mask was then applied to replace the pixels outwith the bounding box of the largest CC with white pixels.

3.3 Symbol detection

Two state-of-the-art object detection models were used in the experiment. These are YOLOv7 [55] and Faster R-CNN [33]. YOLOv7 [55] is a variant from the YOLO series [28,

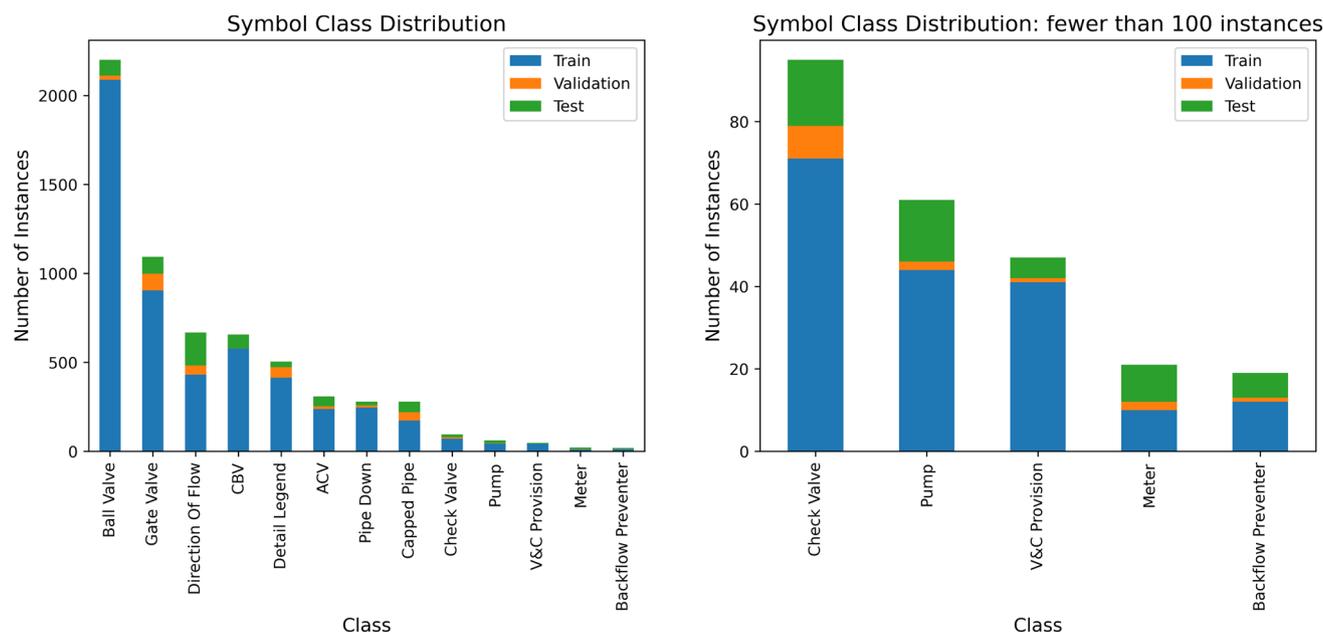


Fig. 5 The left image shows the class distribution across the whole symbol dataset. The right image shows the distribution amongst those classes with fewer than 100 instances in more detail

55, 57–62]. It is a one-stage model, that predicts objects' locations and classes using a single CNN. It is known for its fast performance, for instance YOLOv7 had improved speed and detection accuracy on the COCO dataset [54] compared with other object detectors in the range 5 FPS to 160 FPS [55]. Additionally, YOLO also performs well across different types of diagrams [15]. Faster R-CNN [33] is known to be accurate, with state-of-the-art performance on the PASCAL Visual Object Classes (VOC) benchmarks [63]. Whilst Faster R-CNN [33] improved on the speed of earlier related models, Fast R-CNN [33] and R-CNN [64], its separate region proposal stage results in slower speeds compared to one-stage models.

The construction diagrams are significantly larger compared with the typical image input size for deep learning object detection models. For example, the diagrams are $14,042 \times 9,934$ pixels whereas the YOLOv7 input size is 640×640 pixels [55]. Using the whole diagrams as training images would require considerable computing resources and therefore, a patch-based approach was used. This involves splitting high-resolution images into smaller patches [15, 65, 66]. In this experiment, the patch size was set at 640×640 pixels. Note that the diagrams cannot be split exactly by the patch size, and the patches cropped at the edges of each diagram overlap each other. Only the symbols that appeared completely within a patch were used for training purposes. Note that the drawings were annotated prior to being split into patches, therefore these symbols were determined automatically.

Due to the limited size of the dataset, transfer learning was used. This technique improves a learner by transferring information from one domain to another [67]. Both models were pre-trained on a large-scale object detection dataset, 2017 Microsoft Common Objects in Context (COCO) [54]. All model layers were fine-tuned during training.

4 Experiment and results

4.1 Experiment setup

The experiment can be divided into two phases. The first is the Deep Learning Model Training on Construction Symbols and second is the Method Evaluation, as shown in Fig. 6. The input to the first phase is the pre-processed diagram dataset that was the output from the Data Preparation phase shown in Fig. 2.

The pre-processed dataset of 198 diagrams was split into training, validation and test sets. These contained 168, 15 and 15 diagrams respectively. Each subset contained all three diagram types and instances of each symbol class. As the classes were unevenly distributed across the diagrams, there was a different proportion of each class across the subsets, as seen in Fig. 5.

Following the patch-based approach described above, the 168 training diagrams were split into 59,136 patches. Of these, 1633 contained at least one annotated symbol. Patches not including symbols of interest were also included in the training data, in equal ratio to labelled patches. This may

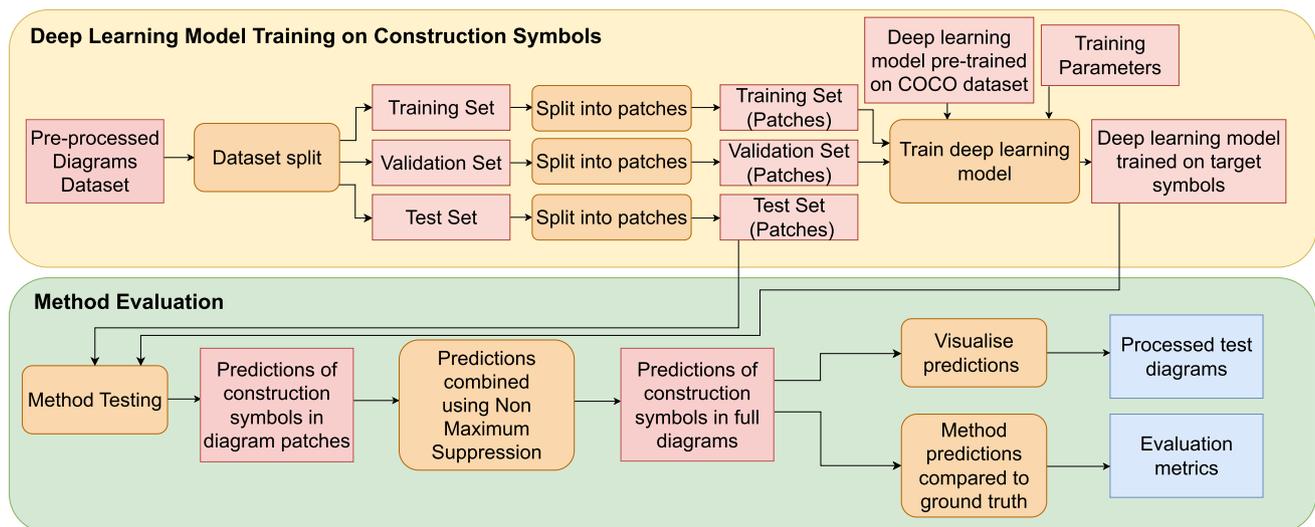


Fig. 6 Experiment steps. In deep learning model training on construction symbols, the pre-processed diagram dataset was split into training, validation and test sets prior to being split into patches. This is followed by model training. In Method Evaluation, the method was tested and the

predictions on individual patches were combined using Non-Maximum Suppression. Next the predictions were visualised to create the processed test diagrams and the predictions were compared to the ground truth

help to reduce false positives occurring due to similar shapes in the diagrams. To select the more cluttered patches, these were randomly sampled from those which contained over 15% black pixels. The 15 validation diagrams were split into 5280 patches, of which 94 were annotated. Again, patches without symbols of interest were included in equal amounts to the labelled patches.

YOLOv7 was trained using a batch size of eight as initial experiments showed improved results compared with larger batch sizes. The momentum was set to 0.937. To help prevent overfitting, mosaic data augmentation, which mixes four images [59] was used on all the training images. The idea is to show extra symbol contexts to the model, and it also reduces the requirement for a large mini-batch size [59]. The probability of a left-right flip was set at 0.5, and an up/down flip was set at 0.0. The image translation factor was set at ± 0.2 .

The Faster R-CNN batch size was set at four due to memory requirements. The momentum was set at 0.9, and the probability of a horizontal flip was set at 0.5. Following the original baseline model, a ResNet-50 backbone was used [33]. Note that as the aim was to compare the methods based on the two models, the data augmentations used in each approach were kept as is standard in each implementation.

Each model was trained for 100 epochs which took 2.92 hours for the YOLO-based method, and 40.86 hours for the Faster R-CNN-based method. Note that the official implementations were used ². [68]. The experiments were carried

out using an NVIDIA Quadro RTX5000 16GB GPU with 256GB RAM.

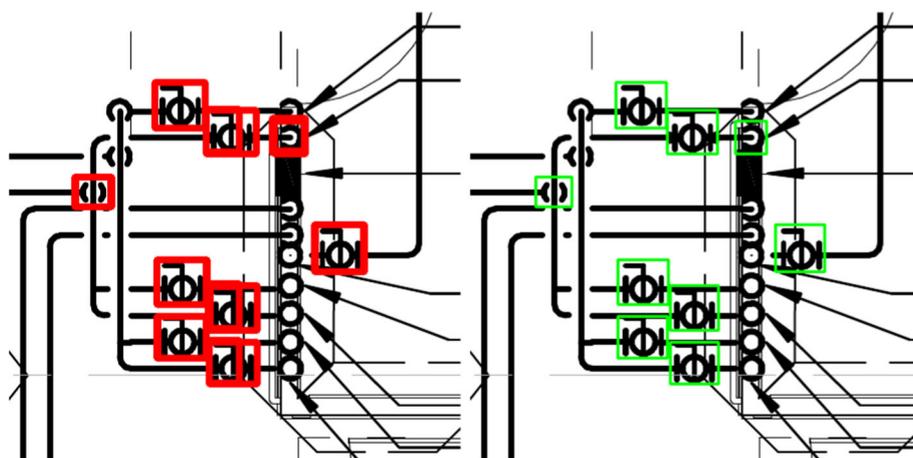
The methods were evaluated using a test set, which contains 15 drawings split into 19,995 patches, as seen in Fig. 6. Here an overlapping patches strategy was used to ensure all symbols fully appeared within a patch. This means that overlapping predictions can occur. Non-Maximum Suppression (NMS) was used to handle this, as shown in Fig. 7. It should be noted that the overlap threshold was set at 0.3, and the confidence threshold was set at 0.005. It is worth pointing out that the testing took 0.09 hours using the YOLO-based method, and 2.72 hours using the Faster R-CNN-based method. Note that this was for the whole test set of 15 drawings. This is significantly less than the time required for manual drawing analysis, which can take hours of work per drawing and requires subject matter specialists.

4.2 Evaluation metrics

The methods were evaluated using multiple metrics, including Precision, Recall, and F1-score. The Precision is the ratio of True Positives to predicted positives (Eq. 1). The Recall is the ratio of True Positives to the actual positives (Eq. 2). The F1-score is the harmonic mean of Precision and Recall (Eq. 3). The Intersection Over Union (IOU) defines the overlap between the predicted and the ground truth bounding boxes (Eq. 4). For a True Positive, the IOU threshold was set at 0.5 in accordance with the PASCAL evaluation metric [69], and the model confidence threshold was set at 0.25. The latter setting should ensure an appropriate trade-off between

² <https://github.com/WongKinYiu/yolov7>.

Fig. 7 Non-Maximum Suppression was used to handle the overlapping predictions. Initial predicted bounding boxes are shown in red (left image). The results following Non-Maximum Suppression are shown in green (right image) (color figure online)



obtaining true positives and reducing false positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

$$\text{F1 score} = \frac{1}{\frac{1}{2} \left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}} \right)} \quad (3)$$

$$\text{Intersection Over Union} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (4)$$

The method was also evaluated using the mean Average Precision (mAP) at IOU threshold of 0.5 (mAP@0.5). The AP of each class, the area under the Precision-Recall curve, was determined using the all-point interpolation method as in PASCAL VOC 2010 [63]. In addition, the AP@[0.5 : 0.05 : 0.95], AP_{small}, AP_{medium} and AP_{large} were reported [54]. An open-source toolkit for object detection metrics created by Padilla et al. [56] was used to perform this calculation.

4.3 Results

The results were initially evaluated by visual inspection with help from domain experts in order to understand the model performance. As shown in Fig. 8, this was facilitated by drawing bounding boxes around the ground truth in red, YOLO-based method correct predictions in orange and by the Faster R-CNN-based method in purple. The incorrect predictions by the YOLO-based method are shown in dark blue and those by the Faster R-CNN-based method in light blue. For in-depth analysis, the model confidence and the IOU were also shown. This suggested that various symbol classes were detected well, even with multiple overlapping components. It was also observed that where a correct prediction was recorded by both methods, the difference in predicted bounding box locations was small and most visible on the larger symbols, refer to patches a and b in Fig. 8.

The results on the whole dataset were assessed using several metrics, as shown in Table 1. The mAP@0.5 of the YOLO-based method was 79%, whilst that of the Faster R-CNN-based method was 83%. Out of the 665 symbols, 637 were correctly detected by the YOLO-based method and 636 by the Faster R-CNN-based method, equivalent to an accuracy of 95.8% and 95.6%, respectively. In terms of the AP@[0.5 : 0.05 : 0.95], both methods performed equally with a score of 0.50. The results were also evaluated according to symbol size. This shows that both methods perform better the larger the symbol size is. This is likely due to more information being present in the larger symbols compared to the smaller ones. Both methods performed equally on the medium sized symbols, as shown in Table 1. It is also evident that the YOLO-based method performs slightly better on the small symbols than the Faster R-CNN-based method, with a value of 0.27 compared to 0.19 obtained for AP_{small}. Similarly, the YOLO-based method performs slightly better on the large symbols than the Faster R-CNN-based method, with the values of AP_{large} being 0.68 and 0.64 respectively. These results suggest that although the YOLO-based method outperforms the Faster R-CNN-based method on certain metrics, both methods have performed well on this challenging dataset.

The precision, recall and F1-score were calculated for each class, as can be seen in Table 2. These results show that both methods performed well for the detection of various symbol classes. Although class imbalance can strongly effect performance, other factors also influenced these results. For instance, the highest F1-score was not obtained for the majority class, the Ball Valve. The recall was high indicating that most instances were detected correctly. This included those in different orientations, as shown in patches a, c and e in Fig. 8. However, the precision was lower, which may be due to several reasons, including model bias due to class imbalance. Furthermore, similar shapes were very common, refer to the incorrect predictions of Ball Valves shown in patches f, h

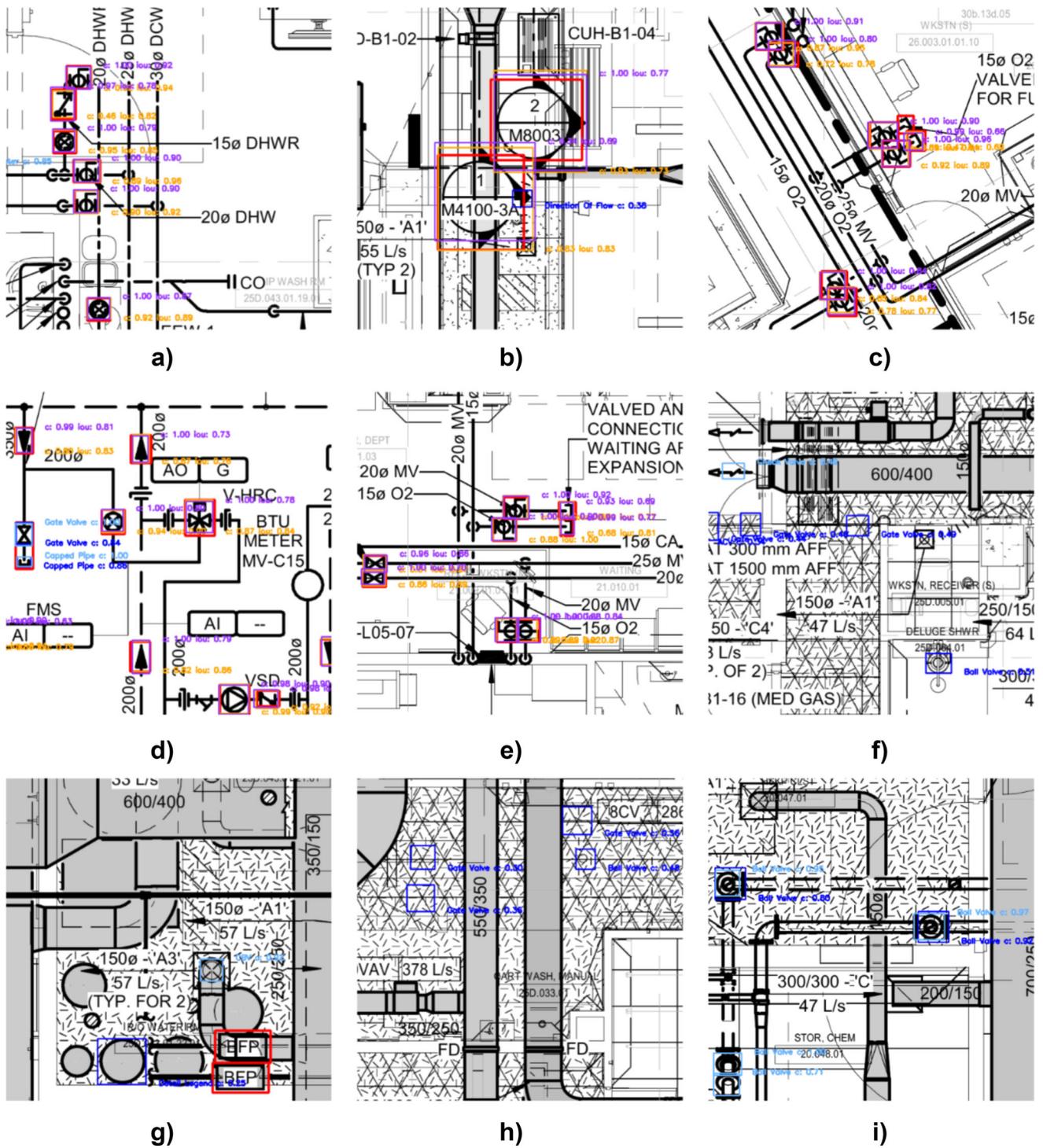


Fig. 8 Examples of test patches. To facilitate visual inspection, bounding boxes were shown around the ground truth in red, YOLO-based method correct predictions in orange and by the Faster R-CNN-based method in purple. The incorrect predictions by the YOLO-based method

are shown in dark blue and those by the Faster R-CNN-based method are in light blue. The model confidence, *c*, and the IOU were also shown (color figure online)

Table 1 Method performance on the test set. The highest performing score for each metric is highlighted in bold

Method	Accuracy	mAP@0.5	AP@[0.5:0.05:0.95]	APsmall	APmedium	APlarge
YOLO-based method	95.8	0.79	0.50	0.27	0.50	0.68
Faster R-CNN-based method	95.6	0.83	0.50	0.19	0.50	0.64

Bold values represent the highest values obtained for each metric

Table 2 Method performance on the test set per class. The highest recall, precision and F1-score for each symbol is highlighted in bold

Class	Total No.	Train No.	Test No.	YOLO			Faster R-CNN		
				Recall	Precision	F1-score	Recall	Precision	F1-score
Ball valve	2201	2088	89	0.99	0.69	0.81	0.97	0.70	0.81
Gate valve	1093	904	95	1.00	0.39	0.56	0.99	0.90	0.94
Direction of flow	668	430	186	0.99	0.94	0.96	0.98	0.89	0.93
CBV	656	577	78	1.00	0.77	0.87	1.00	0.85	0.92
Detail legend	504	414	31	0.97	0.38	0.55	1.00	0.69	0.82
ACV	308	237	55	1.00	0.87	0.93	1.00	0.98	0.99
Pipe down	279	246	21	0.90	0.59	0.71	0.90	0.58	0.71
Capped pipe	279	173	59	0.88	0.78	0.83	0.83	0.72	0.77
Check valve	95	71	16	0.88	0.67	0.76	1.00	0.80	0.89
Pump	61	44	15	1.00	1.00	1.00	1.00	0.58	0.73
V&C provision	47	41	5	0.00	0.00	0.00	0.00	0.00	0.00
Meter	21	10	9	0.78	0.64	0.70	1.00	0.82	0.90
Backflow preventer	19	12	6	0.00	0.00	0.00	0.33	0.22	0.26

Bold values represent the highest values obtained for each metric for each class

and i in Fig. 8. The highest performance by the YOLO-based method was for the Pump and by the Faster R-CNN-based method was for the ACV, even given that these were the sixth and tenth most represented classes respectively. This indicates that the results are impacted by other factors as well as class representation, such as similar shapes in the drawings.

The results also show that the lowest performance was obtained for classes with very few instances. For example, an F1-score of 0.00 was recorded by both methods for one class, the V&C Provision, which had only 47 instances. Although the Meter had fewer instances, 21, the performance was higher, likely due to the relatively consistent appearance of this symbol.

It can also be seen in Table 2 that there were higher levels of recall compared to precision. This was due to false positives, of which there were 320 by the YOLO-based method and 154 by the Faster R-CNN-based method. Only a few of these were as a result of the inter-class similarity. There was one prediction of a Meter as a Detail Legend by the YOLO-based method, and one prediction of a Gate Valve as a Check Valve by the Faster R-CNN-based method. The other misclassifications between target classes were that all V&C Provisions were predicted as two separate symbols, the Gate Valve and Capped Pipe. An example of this is shown in patch d in Fig. 8. This can be explained as the shapes that

constitute the V&C Provision are essentially a combination of these two symbols, see Fig. 3.

The majority of the false positives were due to similar shapes in the background of the drawing. The highest number of false positives for any symbol, 150, was for the Gate Valve by the YOLO-based method. This was often due to similar triangular shapes used to shade parts of the diagram, as shown in patches f and h in Fig. 8. In contrast, the Faster R-CNN method performed better here and only predicted 9 false positives for the Gate Valve. Another noticeable difference between the methods was in the number of false positives recorded for the Detail Legend symbol, for which the YOLO-based method predicted 47 whereas the Faster-RCNN-based method predicted less at 14. Both methods predicted a similar number of false positives for the most represented symbol in the dataset, the Ball Valve, with 39 false positives by the YOLO-based method and 36 for the Faster R-CNN-based method. Both of the methods predicted that similar shaped components in the diagram were Ball Valves, see patch i of Fig. 8. It should also be pointed out that there were no false positives from similar shapes in the diagram border area, as this section of the drawing was removed in the drawing pre-processing, refer to Fig. 2. Overall, these results show that both methods have high discriminative power between the target classes, and that most incorrect predictions result from the similar shapes that are used throughout the drawing.

The YOLO-based method required less time for testing compared to the Faster R-CNN-based method. However, both methods substantially reduced the time required to process a drawing compared to manual analysis. For instance, using one test drawing as an example, the YOLO-based method took 0.54 minutes, whereas the Faster R-CNN-based method took 18.17. Note that this time would be complemented by that required for manual review to correct any errors in the model output, which in this case took an additional 1.20 minutes. Contrast this with the much longer time needed for manual analysis of the whole drawing, which took a subject matter expert 1.34 hours. These time savings would be substantial, especially in projects with a large dataset of complex drawings.

5 Conclusion and future direction

In this paper, we present a deep learning framework for the automatic processing of construction drawings. This enables symbol digitisation and can therefore automate tasks such as material takeoff. Two state-of-the-art object detection models, YOLO and Faster R-CNN, were utilised. An extensive set of experiments was carried out using a large dataset of challenging high-resolution drawings sourced from an industry partner.

The results show significant time-saving compared with manual drawing analysis. Although the highest accuracy was obtained with the YOLO-based method, both methods were shown to obtain high performance, for both recall and precision, for a range of symbols. This was obtained even with the challenges posed by the dataset, such as relatively small symbol size, different orientations and the presence of multiple overlapping objects. One limitation was that the performance was inconsistent across the classes, due to factors including the class imbalance, similar shapes and intra-class variations such as size and orientation.

This work could be extended by improving the symbol detection methods. For instance, further experiments could assess the impact of various model backbones on the performance. Another interesting idea that we aim to investigate is how segmentation models such as Mask R-CNN [47] or FCN [37] would perform in this scenario. As segmentation models predict a pixel-level outline of an object instead of a bounding box, it may alleviate some of the errors that occur as a result of drawing objects that are located in close proximity to, or overlap the target object. Additionally, the challenge of class imbalance could be addressed. One possible direction is to create synthetic image patches to balance the dataset, using generative deep learning models such as Generative Adversarial Networks (GAN).

Future directions of this work also include automatically processing the entire construction drawing. This involves

methods to digitise all components including the text and lines. Extending this framework will enable the digital transformation of the whole drawing. This will allow for the extraction of vast amounts of valuable data, and additionally will substantially reduce the manual effort required to analyse construction drawings for a wide range of tasks.

Acknowledgements The construction drawings were provided by TaksoAI.

Funding Not applicable.

Data availability The engineering drawings used were not made publicly available due to confidentiality concerns related to the industrial partner.

Code availability The official public implementations of deep learning architectures were utilised in this paper.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Liu, H., Cheng, J.C., Gan, V.J., et al.: A knowledge model-based bim framework for automatic code-compliant quantity take-off. *Autom. Constr.* **133**, 104024 (2022). <https://doi.org/10.1016/j.autcon.2021.104024>. (<https://www.sciencedirect.com/science/article/pii/S0926580521004751>)
2. Joy, J., Mounsef, J.: Automation of material takeoff using computer vision. In: 2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pp 196–200, (2021) <https://doi.org/10.1109/IAICT52856.2021.9532514>
3. Monteiro, A., Poças Martins, J.: A survey on modeling guidelines for quantity takeoff-oriented bim-based design. *Autom. Constr.* **35**, 238–253 (2013). <https://doi.org/10.1016/j.autcon.2013.05.005>. (<https://www.sciencedirect.com/science/article/pii/S0926580513000721>)
4. Khosakitchalart, C., Yabuki, N., Fukuda, T.: Automated modification of compound elements for accurate bim-based quantity takeoff. *Autom. Constr.* **113**, 103142 (2020). <https://doi.org/10.1016/j.autcon.2020.103142>. (<https://www.sciencedirect.com/science/article/pii/S0926580519310751>)
5. Chowdhury, A.M., Moon, S.: Generating integrated bill of materials using mask r-cnn artificial intelligence model. *Autom.*

- Constr. **145**, 104644 (2023). <https://doi.org/10.1016/j.autcon.2022.104644>. (<https://www.sciencedirect.com/science/article/pii/S0926580522005143>)
6. Jakubik, J., Hemmer, P., Vössing, M., et al.: Designing a human-in-the-loop system for object detection in floor plans. Proc. AAAI Conf. Artif. Intell. **36**(11), 12524–12530 (2022). <https://doi.org/10.1609/aaai.v36i11.21522>. (<https://ojs.aaai.org/index.php/AAAI/article/view/21522>)
 7. Baduge, S.K., Thilakarathna, S., Perera, J.S., et al.: Artificial intelligence and smart vision for building and construction 4.0: machine and deep learning methods and applications. Autom. Constr. **141**, 104440 (2022). <https://doi.org/10.1016/j.autcon.2022.104440>
 8. Pan, Y., Zhang, L.: Roles of artificial intelligence in construction engineering and management: a critical review and future trends. Autom. Constr. **122**, 103517 (2021). <https://doi.org/10.1016/j.autcon.2020.103517>. (<https://www.sciencedirect.com/science/article/pii/S0926580520310979>)
 9. Abioye, S.O., Oyedele, L.O., Akanbi, L., et al.: Artificial intelligence in the construction industry: a review of present status, opportunities and future challenges. J. Build. Eng. **44**, 103299 (2021). <https://doi.org/10.1016/j.jobe.2021.103299>. (<https://www.sciencedirect.com/science/article/pii/S2352710221011578>)
 10. Mani, S., Haddad, M.A., Constantini, D., et al.: Automatic digitization of engineering diagrams using deep learning and graph search. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 673–679 (2020)
 11. Hantach, R., Lechuga, G., Calvez, P.: Key information recognition from piping and instrumentation diagrams: Where we are? In: Barney Smith, E.H., Pal, U. (eds.) Document Analysis and Recognition - ICDAR 2021 Workshops, pp. 504–508. Springer International Publishing, Cham (2021)
 12. Ablameyko, S., Uchida, S.: Recognition of engineering drawing entities: review of approaches. Int. J. Image Gr. **7**, 709–733 (2007). <https://doi.org/10.1142/S0219467807002878>
 13. Groen, F.C., Sanderson, A.C., Schlag, J.F.: Symbol recognition in electrical diagrams using probabilistic graph matching. Pattern Recogn. Lett. **3**(5), 343–350 (1985)
 14. Moreno-García, C.F., Elyan, E., Jayne, C.: New trends on digitisation of complex engineering drawings. Neural Comput. Appl. (2018). <https://doi.org/10.1007/s00521-018-3583-1>
 15. Elyan, E., Jamieson, L., Ali-Gombe, A.: Deep learning for symbols detection and classification in engineering drawings. Neural Netw. **129**, 91–102 (2020). <https://doi.org/10.1016/j.neunet.2020.05.025>. (<https://www.sciencedirect.com/science/article/pii/S0893608020301957>)
 16. Gao, W., Zhao, Y., Smids, C.: Component detection in piping and instrumentation diagrams of nuclear power plants based on neural networks. Prog. Nucl. Energy **128**, 103491 (2020). <https://doi.org/10.1016/j.pnucene.2020.103491>. (<https://www.sciencedirect.com/science/article/pii/S0149197020302419>)
 17. Kim, H., Lee, W., Kim, M., et al.: Deep-learning-based recognition of symbols and texts at an industrially applicable level from images of high-density piping and instrumentation diagrams. Expert Syst. Appl. **183**, 115337 (2021). <https://doi.org/10.1016/j.eswa.2021.115337>. (<https://www.sciencedirect.com/science/article/pii/S0957417421007661>)
 18. Ressel, A., Schmidt-Vollus, R.: Reverse engineering in process automation. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp 1–4, (2021). <https://doi.org/10.1109/ETFA45728.2021.9613602>
 19. Moon, Y., Lee, J., Mun, D., et al.: Deep learning-based method to recognize line objects and flow arrows from image-format piping and instrumentation diagrams for digitization. Appl. Sci. **11**(21), 10054 (2021)
 20. Kang, S.O., Lee, E.B., Baek, H.K.: A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams p&id. Energies **12**(13), 2593 (2019). <https://doi.org/10.3390/en12132593>. (<https://www.mdpi.com/1996-1073/12/13/2593>)
 21. Yu, E.S., Cha, J.M., Lee, T., et al.: Features recognition from piping and instrumentation diagrams in image format using a deep learning network. Energies **12**(23), 4425 (2019). <https://doi.org/10.3390/en12234425>
 22. Yin, M., Tang, L., Zhou, T., et al.: Automatic layer classification method-based elevation recognition in architectural drawings for reconstruction of 3d bim models. Autom. Constr. **113**, 103082 (2020). <https://doi.org/10.1016/j.autcon.2020.103082>. (<https://www.sciencedirect.com/science/article/pii/S0926580519303735>)
 23. Okazaki, A., Kondo, T., Mori, K., et al.: An automatic circuit diagram reader with loop-structure-based symbol recognition. IEEE Trans. Pattern Anal. Mach. Intell. **10**(3), 331–341 (1988). <https://doi.org/10.1109/34.3898>
 24. Zhao, Y., Deng, X., Lai, H.: A deep learning-based method to detect components from scanned structural drawings for reconstructing 3d models. Appl. Sci. **10**(6), 2066 (2020). <https://doi.org/10.3390/app10062066>. (<https://www.mdpi.com/2076-3417/10/6/2066>)
 25. Moreno-García, C.F., Elyan, E.: Digitisation of assets from the oil and gas industry: challenges and opportunities. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), pp 2–5, (2019). <https://doi.org/10.1109/ICDARW.2019.60122>
 26. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
 27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, et al (eds) Advances in Neural Information Processing Systems 25. Curran Associates, Inc., p 1097–1105, (2012). <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
 28. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788 (2016)
 29. Gupta, M., Wei, C., Czerniawski, T.: "Automated valve detection in piping and instrumentation (p&id) diagrams". In: "Proceedings of the 39th International Symposium on Automation and Robotics in Construction, ISARC 2022". "International Association for Automation and Robotics in Construction (IAARC)", "Proceedings of the International Symposium on Automation and Robotics in Construction", pp "630–637" ("2022")
 30. Nurminen, J.K., Rainio, K., Numminen, J.P., et al.: Object detection in design diagrams with machine learning. In: Burduk, R., Kurzynski, M., Wozniak, M. (eds.) Progress in Computer Recognition Systems, pp. 27–36. Springer International Publishing, Cham (2020)
 31. Toral, L., Moreno-García, C.F., Elyan, E., et al.: A deep learning digitisation framework to mark up corrosion circuits in piping and instrumentation diagrams. In: Barney Smith, E.H., Pal, U. (eds.) Document Analysis and Recognition - ICDAR 2021 Workshops, pp. 268–276. Springer International Publishing, Cham (2021)
 32. Haar, C., Kim, H., Koberg, L.: Ai-based engineering and production drawing information extraction. In: International Conference on Flexible Automation and Intelligent Manufacturing, Springer, pp 374–382 (2023)
 33. Ren, S., He, K., Girshick, R., et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems- Volume 1. MIT Press, Cambridge, NIPS'15, pp 91–99, (2015). <http://dl.acm.org/citation.cfm?id=2969239.2969250>
 34. Nguyen, T., Pham, L.V., Nguyen, C., et al.: Object detection and text recognition in large-scale technical drawings. In: Proceedings of the 10th International Conference on Pattern Recognition Appli-

- cations and Methods - Volume 1: ICPRAM,, INSTICC. SciTePress, pp 612–619, (2021). <https://doi.org/10.5220/0010314406120619>
35. Stinner, F., Wiecek, M., Baranski, M., et al.: Automatic digital twin data model generation of building energy systems from piping and instrumentation diagrams. 2108.13912 (2021)
 36. Sarkar, S., Pandey, P., Kar, S.: Automatic detection and classification of symbols in engineering drawings. (2022). <https://doi.org/10.48550/ARXIV.2204.13277>, <https://arxiv.org/abs/2204.13277>
 37. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3431–3440, (2015). <https://doi.org/10.1109/CVPR.2015.7298965>
 38. Rahul, R., Paliwal, S., Sharma, M., et al.: Automatic information extraction from piping and instrumentation diagrams. In: Marsico MD, di Baja GS, Fred ALN (eds) Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2019, Prague, Czech Republic, February 19–21, 2019. SciTePress, pp 163–172, (2019). <https://doi.org/10.5220/0007376401630172>
 39. Paliwal, S., Jain, A., Sharma, M., et al.: Digitize-pid: Automatic digitization of piping and instrumentation diagrams. In: Gupta M, Ramakrishnan G (eds) Trends and Applications in Knowledge Discovery and Data Mining - PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings, Lecture Notes in Computer Science, **12705**. Springer, pp 168–180, (2021). https://doi.org/10.1007/978-3-030-75015-2_17
 40. Wang, Y., Sun, Y., Liu, Z., et al.: Dynamic graph CNN for learning on point clouds. CoRR abs/1801.07829 (2018). [arxiv:1801.07829](https://arxiv.org/abs/1801.07829)
 41. Paliwal, S., Sharma, M., Vig, L.: Ocsr-pid: One-shot symbol recognition in piping sheets using path sampling and gen. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp 1–8, (2021). <https://doi.org/10.1109/IJCNN52387.2021.9534122>
 42. Renton, G., Balcilar, M., Héroux, P., et al.: Symbols detection and classification using graph neural networks. Pattern Recognit. Lett. **152**, 391–397 (2021). <https://doi.org/10.1016/j.patrec.2021.09.020>
 43. Jamieson, L., Francisco Moreno-García, C., Elyan, E.: A review of deep learning methods for digitisation of complex documents and engineering diagrams. Artif. Intell. Rev. **57**(6), 1–37 (2024)
 44. Theisen, M.F., Flores, K.N., Schulze Balhorn, L., et al.: Digitization of chemical process flow diagrams using deep convolutional neural networks. Digit. Chem. Eng. **6**, 100072 (2023). <https://doi.org/10.1016/j.dche.2022.100072>. (<https://www.sciencedirect.com/science/article/pii/S2772508122000631>)
 45. Elyan, E., Moreno-García, C.F., Johnston, P.: Symbols in engineering drawings (sied): An imbalanced dataset benchmarked by convolutional neural networks. In: Iliadis L, Angelov PP, Jayne C, et al (eds) Proceedings of the 21st EANN (Engineering Applications of Neural Networks) 2020 Conference. Springer International Publishing, Cham, pp 215–224 (2020)
 46. Rezvaniyar, A., Cote, M., Albu, A.B.: Symbol spotting on digital architectural floor plans using a deep learning-based framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2020)
 47. He, K., Gkioxari, G., Dollár, P., et al.: Mask r-cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 2980–2988 (2017)
 48. Khosakitchalart, C., Yabuki, N., Fukuda, T.: Improving the accuracy of bim-based quantity takeoff for compound elements. Autom. Constr. **106**, 102891 (2019). <https://doi.org/10.1016/j.autcon.2019.102891>. (<https://www.sciencedirect.com/science/article/pii/S0926580518311944>)
 49. Wu, X., Sahoo, D., Hoi, S.C.: Recent advances in deep learning for object detection. Neurocomputing **396**, 39–64 (2020). <https://doi.org/10.1016/j.neucom.2020.01.085>. (<https://www.sciencedirect.com/science/article/pii/S0925231220301430>)
 50. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. **106**, 249–259 (2018). <https://doi.org/10.1016/j.neunet.2018.07.011>. (<https://www.sciencedirect.com/science/article/pii/S0893608018302107>)
 51. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. J. Big Data **6**(1), 27 (2019)
 52. Elyan, E., Moreno-García, C.F., Jayne, C.: Cdsmote: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. Neural Comput. Appl. (2020). <https://doi.org/10.1007/s00521-020-05130-z>
 53. Liu, Y., Sun, P., Wergeles, N., et al.: A survey and performance evaluation of deep learning methods for small object detection. Expert Syst. Appl. **172**, 114602 (2021). <https://doi.org/10.1016/j.eswa.2021.114602>. (<https://www.sciencedirect.com/science/article/pii/S0957417421000439>)
 54. Lin, T.Y., Maire, M., Belongie, S., et al.: Microsoft coco: common objects in context. In: European Conference on Computer Vision, Springer, pp 740–755 (2014)
 55. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. (2022). <https://doi.org/10.48550/ARXIV.2207.02696>, <https://arxiv.org/abs/2207.02696>
 56. Padilla, R., Passos, W.L., Dias, T.L.B., et al.: A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics **10**(3), 279 (2021). <https://doi.org/10.3390/electronics10030279>. (<https://www.mdpi.com/2079-9292/10/3/279>)
 57. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6517–6525, (2017). <https://doi.org/10.1109/CVPR.2017.690>
 58. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. CoRR abs/1804.02767. (2018). [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
 59. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: optimal speed and accuracy of object detection. (2020). [arXiv preprint arXiv:2004.10934](https://arxiv.org/abs/2004.10934)
 60. Jocher, G., Nishimura, K., Mineeva, T., et al.: yolov5. Code repository (2020). <https://github.com/ultralytics/yolov5>
 61. Li, C., Li, L., Jiang, H., et al.: Yolov6: a single-stage object detection framework for industrial applications. (2022). <https://doi.org/10.48550/ARXIV.2209.02976>, <https://arxiv.org/abs/2209.02976>
 62. Jocher, G., Chaurasia, A., Qiu, J.: Yolo by ultralytics. (2023). <https://github.com/ultralytics/ultralytics>
 63. Everingham, M., Eslami, S.A., Van Gool, L., et al.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**, 98–136 (2015)
 64. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp 580–587, (2014). <https://doi.org/10.1109/CVPR.2014.81>
 65. Ruzicka, V., Franchetti, F.: Fast and accurate object detection in high resolution 4k and 8k video using gpus. In: 2018 IEEE High Performance Extreme Computing Conference (HPEC), IEEE, pp 1–7 (2018)
 66. Rezvaniyar, A., Cote, M., Albu, A.B.: Symbol spotting on digital architectural floor plans using a deep learning-based framework. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 2419–2428, (2020). <https://doi.org/10.1109/CVPRW50498.2020.00292>
 67. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. J. Big data **3**(1), 1–40 (2016)
 68. Wu, Y., Kirillov, A., Massa, F., et al.: Detectron2. (2019). <https://github.com/facebookresearch/detectron2>

69. Everingham, M., Van Gool, L., Williams, C.K., et al.: The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision* **88**(2), 303–338 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.