

ELYAN, E., JAMIESON, L. and ALI-GOMBE, A. 2020. Deep learning for symbols detection and classification in engineering drawings. *Neural networks* [online], 129, pages 91-102. Available from: <https://doi.org/10.1016/j.neunet.2020.05.025>.

# Deep learning for symbols detection and classification in engineering drawings.

ELYAN, E., JAMIESON, L. and ALI-GOMBE, A.

2020

# Deep Learning for Symbols Detection and Classification in Engineering Drawings

Eyad Elyan\*, Laura Jamieson, Adamu Ali-Gombe

*School of Computing Science and Digital Media, Robert Gordon University, UK*

---

## Abstract

Engineering drawings are commonly used in different industries such as Oil and Gas, construction, and other types of engineering. Digitising these drawings is becoming increasingly important. This is mainly due to the need to improve business practices such as inventory, assets management, risk analysis, and other types of applications. However, processing and analysing these drawings is a challenging task. A typical diagram often contains a large number of different types of symbols belonging to various classes and with very little variation among them. Another key challenge is the class-imbalance problem, where some types of symbols largely dominate the data while others are hardly represented in the dataset. In this paper, we propose methods to handle these two challenges. First, we propose an advanced bounding-box detection method for localising and recognising symbols in engineering diagrams. Our method is end-to-end with no user interaction. Thorough experiments on a large collection of diagrams from an industrial partner proved that our methods accurately recognise more than 94% of the symbols. Secondly, we present a method based on Deep Generative Adversarial Neural Network for handling class-imbalance. The proposed GAN model proved to be capable of learning from a small number of training examples. Experiment results showed that the proposed method greatly improved the classification of symbols in engineering drawings.

*Keywords:*

---

\*Corresponding author  
*Email address:* [e.elyan@rgu.ac.uk](mailto:e.elyan@rgu.ac.uk) (Eyad Elyan)

## 1. Introduction

Large volumes of un-digitised and paper-based documents are still very common across different domains. Amongst this legacy, engineering drawings are known to be one of the most complex types of documents to process and analyse. They are widely used in different industries such as construction and city  
5 planning (i.e. Floor Plan diagrams [1]), Oil and Gas (i.e. P&IDs [2]), Mechanical Engineering [3], AutoCAD Drawing Exchange Format (DXF) [4] and others. Interpreting these drawings requires highly skilled people, and in some cases long hours of work.

10 In recent years, the digitisation of these drawings is becoming increasingly important. This is partly due to the urgent need to improve business practices such as inventory, assets management, risk analysis, safety checks and other types of applications, and also due to the recent advancements in the domain of machine vision and image understanding. Deep Learning (DL) [5], in particular,  
15 had significantly improved the performance by orders of magnitude in many domains such as Gaming and AI [6], Natural Language Processing [7], Health [8], and others. One particular domain that has benefited hugely from DL is machine vision [9]. Convolutional Neural Networks (CNNs) [10] have made significant progress in recent years in many image-related tasks [11]. It has  
20 been successfully applied to several fields such as hand-written recognition [12], image classification [13, 14], Face Recognition & Biometrics [15] and others. Before the CNNs, the improvements in image classification, segmentation, and object detection was marginal and incremental. However, the introduction of CNNs revolutionised this field. For example, Deep Face [16], a face recognition  
25 system that was first proposed by FaceBook in 2014 achieved an accuracy of 97.35%, beating the state-of-the-art then, by 27%.

Core image processing tasks such as shape and object detection, recognition,

and tracking have become much less challenging even under different conditions and in much less controlled environments. Faster Region-based CNN (R-CNN) [17], Single Shot Detectors (SSD) [18], Region-based Fully Convolutional Networks (R-FCN) [19] and You Only Look Once (YOLO) [20] are all relatively recent methods that showed superior performance in the field of object detection, tracking, and classification. These methods and their extensions have significantly advanced this area of research and solved some of the most challenging and inherent vision problems such as occlusions, light conditions, orientation, and others, which were considered major challenges, even for a specific vision task in a more controlled environment [21].

Significant advancement has also been made in the area of Generative Models and was successfully applied in many applications. Among these, Generative Adversarial Networks (GAN) proved to be one of the most established and commonly used methods in generating content. GANs were initially introduced by Ian Goodfellow in 2014 [22]. In the Methods section, we will discuss our GAN-based method to handle the class imbalance problem. This is another challenging problem that is common across many domains [23] including engineering drawings, where one or more class of symbols in the diagrams are either underrepresented or overrepresented in the dataset [24].

Despite this massive progress in the field of image processing and analysis, very little progress has been made in the area of digitising complex engineering drawings, and extracting information from these diagrams is still considered a challenging problem [25]. To date, a major problem of most of the existing solutions is that they still follow a traditional image-processing approach, which requires extensive features extraction and engineering and carefully designed heuristics [26]. These are often very domain-dependent, sensitive to noise and data distribution, and mostly dedicated to solving part of the problem (i.e. detecting symbols, separating graphics from text, and so on). As can be seen in Figure 1, not only such an approach difficult to generalise across different scenarios, but also the performance of any machine learning algorithm will hugely depend on the quality and accuracy of the extracted features.

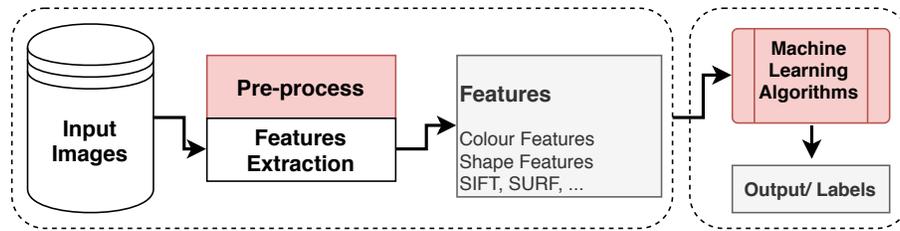


Figure 1: Traditional frameworks for analysing images/ documents

In this paper, we propose an end-to-end framework for processing and analysing  
 60 complex engineering drawings. We argue that the core task of such a frame-  
 work is the accurate localisation and recognition of symbols in the drawing that  
 constitute a major part of it and simplifies subsequent tasks (i.e. line and text  
 detection). We show how one of the main inherent problems in classifying en-  
 gineering symbols, namely class-imbalance can be addressed using Generative  
 65 Adversarial Neural Networks. Figure 2 provides a schematic diagram of the  
 work presented in this paper. The main contributions of this work are outlined  
 as follows:

- We propose a novel pipeline for processing and analysing complex engi-  
 70 neering drawings. At the core of this pipeline is the accurate detection  
 and recognition of symbols.
- We show that an advanced-bounding-box detection method performs very  
 accurately on challenging engineering diagrams. To the best of our knowl-  
 edge, Deep Learning models (e.g. YOLO [27], RCNN [9] ) were never used  
 in such domain at a large scale of symbols with minimal difference. This is  
 75 mainly due to the complexity of the problem, and the very little variation  
 and noise within symbols of engineering drawings.
- Methods to handle the class-imbalance within engineering drawings are  
 presented and thoroughly evaluated. We present a fine-grained method to  
 train GAN models to generate engineering symbols of different overlapping  
 80 classes.

- Thorough evaluation using large collection of P&ID diagrams provided by an industry partner in the Oil and Gas sector.

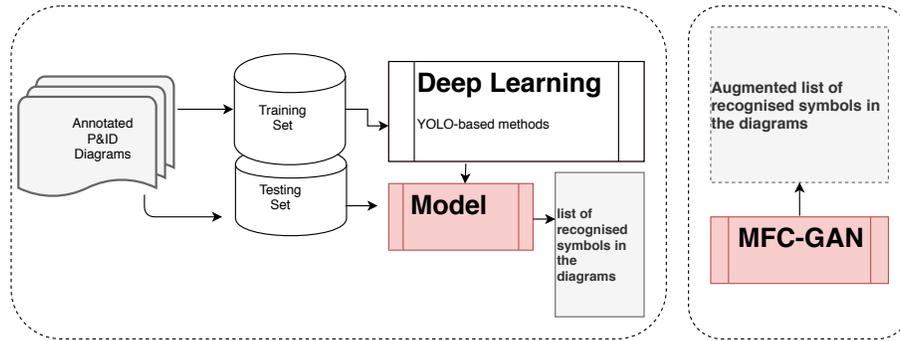


Figure 2: Schematic diagram of the framework for processing and analysing engineering documents

The rest of this paper is organised as follows: Section 2 presents an overview and critical discussion of relevant work. In section 3, we present our methods,  
 85 dataset and pre-processing steps carried out. In section 4 we present our experiments and discuss results. Finally, conclusions and future work are outlined in section 5.

## 2. Related Work

In this section, we discuss relevant literature. First, we discuss literature  
 90 related to the processing and analysis of engineering drawings. This will be followed by a brief introduction to Generative Adversarial Neural Networks and how it can be applied to handle the class imbalance problem.

### 2.1. Engineering Drawings

An engineering drawing is a 2D image that contains different types of shapes,  
 95 symbols, lines, and text. These drawings are commonly used in different domains and provide a rich representation of complex engineering workflows or situations. The digitisation of these drawings was subject to extensive research from the machine vision research community over the past four decades

[28, 29, 30, 31]. In recent years, and due to the significant progress in machine  
100 vision research, computer power and also due to the availability of large vol-  
umes of un-digitised data, the demand to have a fully automated framework for  
digitising these drawings is becoming increasingly important.

Examples of work that aimed at extracting information from engineering  
documents include analysis of musical notes [32], mechanical drawings [33], op-  
105 tical character recognition (OCR) [34, 35, 36], and extracting information from  
P&ID drawings [2, 37, 38]. It can be argued that most of the existing litera-  
ture followed a traditional image processing approach [39], which requires some  
form of feature extraction from the image [28], features representations [30], and  
classification to determine the class of objects (i.e. symbols, digits, ...) [31].

110 The key limitation of traditional machine vision methods is that they re-  
quire extensive features engineering, depend heavily on the quality of extracted  
features, and often won't generalise well to other unseen examples. A recent  
extensive review showed that most of the existing literature focused on solving  
part of the problem rather than providing a fully automated framework for digi-  
115 tising an engineering diagram [26]. Examples include methods for recognising  
symbols and lines in a drawing [40], detecting and separating text from symbols  
and other graphics elements in diagram [38], classifying symbols in engineering  
drawings [2] and so on. This is partly due to the complexity of the problem (i.e.  
localising every single element in the document), and also due to the limitations  
120 of the traditional image processing and analysis methods and the inherent vision  
problems such as the sensitivity to noise, quality of the image, the orientation  
of shapes and so on. Consider for example the work in [2], the authors used  
a set of heuristic rules to localise symbols in the drawings, a Random Forest  
[41] was then used to classify the symbols achieving an average accuracy higher  
125 than 95%. Similar work was presented in [38], where a set of heuristics were  
also used to detect and separate text from graphics elements. However, such an  
approach is very dependent on the data distribution, and a slight variation in  
the diagrams or in symbols representation might require adapting the existing  
heuristics rules or creating new ones.

130 In a closely related area, Rebelo et al [42] presented a study on optical music recognition and classification methods for musical symbols. They suggested that adjoining staff lines, presence of symbols in close proximity to music notes, broken symbols, overlapping symbols and areas with high symbol density all contributed to the complexity of optical music recognition. Four classification  
135 methods namely a multi-layer perceptron neural network model, Hidden Markov Model, K-nearest neighbour and Support Vector Machine (SVM) were evaluated on datasets of both synthetic and handwritten music scores. The highest performance was obtained with an SVM model, however all approaches implemented detection then removal of music staff lines and segmented the symbols  
140 prior to symbol classification.

Khan et al [43], used video image analysis as part of a flight deck warning system, which combined automated dial reading of flight instruments with domain knowledge. Experiments on a flight simulator and real flight aimed to obtain the position of a white needle on the flight instrument using three image  
145 processing approaches: background subtraction, pattern matching and a convolution based approach. Results showed that the convolution method obtained the highest accuracy, highest true positive rate and highest true negative rate.

In recent years, DL-based methods were explored and successfully applied to some tasks that are similar to engineering drawings analysis. Ziran et al.  
150 proposed a method, based on Single Shot Detectors (SSD) [18], to detect and recognise furniture objects, doors, and windows in floor plan diagrams [44]. The results were encouraging. However, the datasets used were simple with a limited number of furniture objects in each drawing (12). The performance also dropped under the imbalanced class distribution of objects in the images.

155 Faster R-CNN was used in [45] for the detection and recognition of handwritten characters. Although the work focused mainly on specific elements of the documents (mathematical expressions and flowcharts), promising results over other traditional methods were achieved.

Detection and recognition of musical notes in documents have also benefited  
160 from adopting Deep Learning-based methods [44, 46]. R-CNN, R-FCN, and

SDD were applied successfully to detect and recognise handwritten music notes [46]. Results showed an improvement in symbols detection over other traditional structured machine vision methods.

A framework for extracting information from P&ID drawings was presented very recently in [47]. The authors used a two-step approach. First, Deep Learning methods were used to localise symbols and text, and then heuristic-based methods were employed to detect other elements of the drawing (i.e. Euclidean metrics for associating tags and symbols with pipelines, probabilistic Hough transform to detect pipelines, etc.) The methods for localising symbols were based on a fully connected convolutional neural network. A dataset of four sheets consisting of 672 flow diagrams was used. Results were an improvement over other traditional methods. However, accuracy wasn't consistent across all components. Class accuracy ranged from 100% for some components to 64.0% for others (i.e. symbols of a certain class). Moreover, only a limited number of symbols were used in this study (10 different classes of symbols) and the P&ID sheets seem to be of a very good quality which is not often the case in the real world.

To summarise, existing literature shows a clear gap between the current state of machine vision and image understanding -*due to the rapid development in this field*- and the slow and incremental progress in a very important application domain across many industries.

## 2.2. GAN Models

Generative Adversarial Networks (GAN) were initially introduced by Ian Goodfellow in 2014 [22]. These are considered as generative models that are capable of producing new content. GANs are made of two contesting models (i.e. Neural Networks, CNNs, etc), the Generator (G), and the Discriminator (D). The discriminator is a classifier that receives input from the training set (authentic content), and from the generator (fake input). During the training process, the discriminator will learn how to distinguish between authentic and fake input samples. On the other hand, the generator is trained to generate sam-

ples that capture the underlying characteristics of the original data (replicating original content). Figure 3 depicts the GAN model.

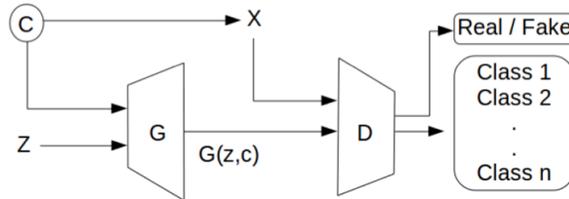


Figure 3: Generative Adversarial Neural Networks

Adversarial training of both models  $G$ , and  $D$  is carried out using value function as can be seen in Equation 1.

$$\min_D \max_G V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

195 Where  $p_{data}(x)$  is the probability distribution over the real data,  $x$  is a sample from the real training data,  $p_z$  is the probability distribution over the noise vector  $z$ , and  $G(z)$  is the output from the generator function  $G$  (or generated images). GANs are state-of-the-art in terms of the quality of the image generated.

200 GANs have been successfully applied to different problems including image generation [48, 49], segmentation and speech synthesis. In recent years they were also successfully applied to handle class-imbalance problems [50, 51]. The class imbalance is common across different domains including health, security, and banking [23]. The problem happens when one or more class is either underrepresented or overrepresented in the dataset. In such scenarios, a typical supervised learning algorithm tends to be biased towards the majority class  
 205 when dealing with imbalanced datasets [24].

Supervised GANs provide an extension to the original GAN framework by introducing conditional probabilities in the value function. This allows more  
 210 control over the generated samples and introduces diversity which is needed for augmenting synthetic input data for class-imbalanced datasets. Typical

examples include vanilla GAN [22], CGAN [52] and AC-GAN [53]. Although the literature shows that these models can be hugely affected by class-imbalance especially in extreme cases [54].

215 Recent work appeared in [51] introduced a new extension of the GAN models. The authors trained the GAN models at a fine-grained level by updating the discriminator objective to not only distinguish between fake and real instances but also to classify the fake instances into different classes (i.e. Fake 1, Fake 2, etc). Extensive experiments using four different datasets showed superior results over other GAN models. Generated samples proved to be of good quality  
220 and were successfully used to augment the dataset and improve the detection rate of minority class instances.

### 3. Methods

Most engineering drawings contain a set of symbols, connectivity information (lines) and some form of annotation (text). However, no public dataset  
225 is available for evaluation purposes. In Section 3.1 we introduce our approach for end-to-end symbols recognition from complex engineering drawings. The following subsection will discuss in detail the dataset used for experiments. This will include data exploration and pre-processing. Finally, Section 3.4 provides  
230 details of our proposed method to handle class-imbalance in these drawings.

#### 3.1. Symbols Recognition

For locating and recognising symbols in the P&IDs, we propose to use YOLO [55] method. This allows us to represent the problem as a set of bounding box coordinates and class probabilities. The method is based on dividing  
235 the entire image into  $S \times S$  grid, where each cell predicts  $B$  bounding boxes and confidence scores for those boxes [55]. The confidence scores are used to decide if a cell contains a symbol or not. These are represented as a five-dimensional vector  $(x, y, w, h, \text{ and confidence})$ . Here,  $(x,y)$  represents the center of the bounding box, while the width and height are predicted relative to the whole

240 image. The prediction from a grid is presented as  $S \times S \times (B * 5 + C)$ . Where  $S$  is the size of the grid,  $B$  is the bounding box and  $C$  is the class probabilities (i.e. probability of the symbol being gate valve, sensor, etc...). Figure 4 depicts this setting.

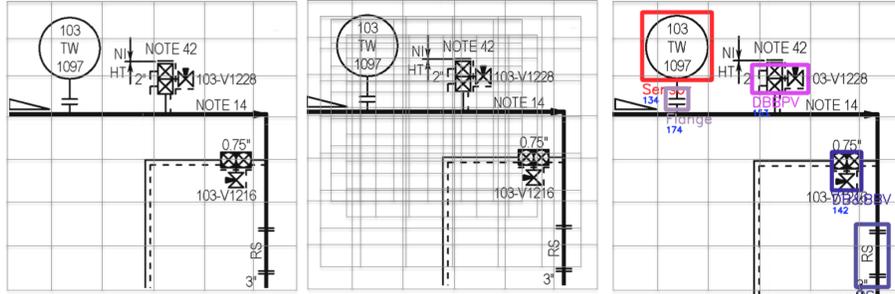


Figure 4: The method divides the P&ID Diagram into a grid, following the YOLO model [55], and predicts the class probabilities of the bounding boxes. The figure shows the symbols sensor, flange, DBBPV, DB&BBV and RS

The YOLO model was chosen for two main reasons. First, it is a simple  
 245 framework, which allows simultaneous predictions of multiple bounding boxes and class probabilities using a single convolutional neural network. Second, compared to other models, YOLO is considered extremely fast. For testing P&IDs that may contain on average 180 instances of various engineering symbols, this is very important in a practical context.

250 *3.2. Dataset - P&ID Diagrams*

For experiments in this paper, we chose to work with Piping and Instrumentation Diagrams (P&IDs) Figure 5. A collection of 172 P&ID sheets were obtained from an Oil and Gas industrial partner for evaluation purposes. These diagrams contain different types of symbols, lines, and text (Figure 5).

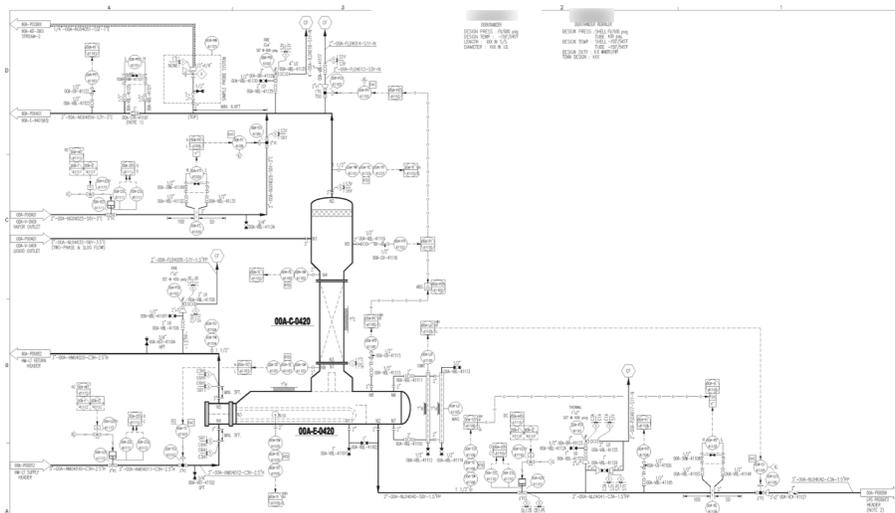


Figure 5: Part of a P&ID Diagram

255 Additionally, the P&IDs are of different qualities, which makes the dataset  
suitable for evaluation purposes. The P&ID diagrams can be defined as schematic  
diagrams representing the different components of the process and the connectiv-  
ity information. It is a representation of equipment (often depicted as symbols)  
and process flow (depicted as different types of lines) [2].

260 Such diagrams are available across many industries in the form of paper or  
scanned documents. Interpreting and analysing these documents requires expert  
knowledge, and is often time-consuming [26, 56]. Moreover, a misinterpretation  
of such documents can be very costly. For example, if a pipe needs to be replaced  
in an Oil and Gas installation, then an engineer needs to check the corresponding  
265 P&ID diagram, identify the valves that must be closed before carrying on the  
task to ensure safety. In other words, accurate interpretation of these drawings  
is paramount.

### 3.3. Data Exploration & Pre-processing

The original P&IDs sheets are large images,  $7500 \times 5250$  pixels. To speed  
270 up the training process we divided the sheet into  $6 \times 4$  grid, resulting in 24 sub-

images (patches) with relatively much smaller sizes compared to the original sheets ( $1250 \times 1300$ ).

Training a Deep Learning model requires fully annotated images/ diagrams. To do so, we have used the Sloth tool <sup>1</sup> to annotate the collection of P&ID diagrams. In total 29 different symbols were annotated in the whole dataset (Figure 7). The annotation process is simple and involves customising the sloth tool to record the corresponding symbols names (class) and its location in the diagram.

The resulting annotation of data is captured in a file representing the 29 symbols. Data recorded included the x,y coordinates of the center of the bounding boxes, width and height of the bounding box enclosing symbols. In total, 13,327 symbols belonging to the 29 different classes were annotated. The dataset is hugely imbalanced as can be seen in Figure 6.

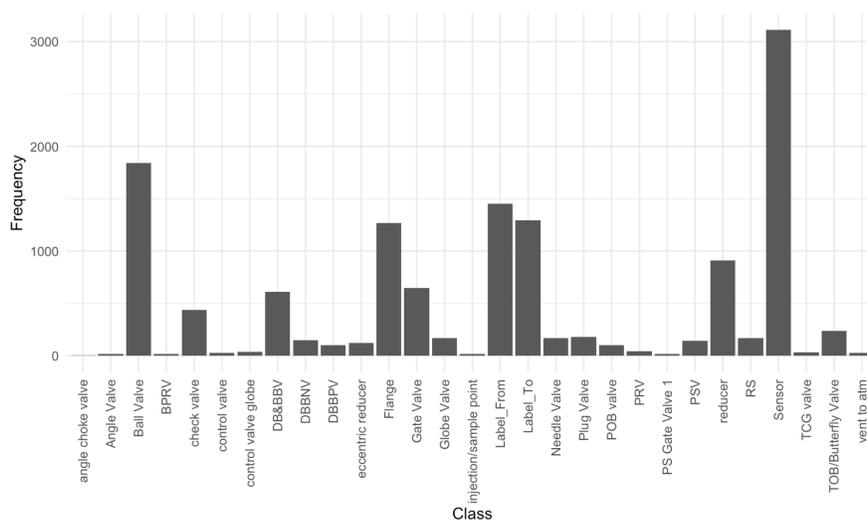


Figure 6: Class distribution of symbols in the whole dataset

Only 25 symbols of these were used in the experiments. These are shown in Figure 7. Five symbols that were extremely under-represented in the whole

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

dataset (i.e. only one or two instances of these symbols appear in the training and testing sets) were excluded from the first experiment.

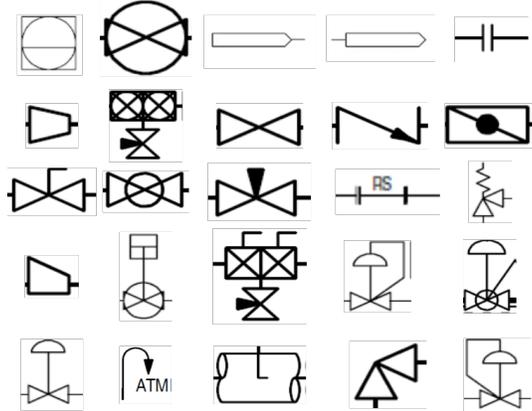


Figure 7: Symbols used in the training and testing sets

### 3.4. MFC-GANs

To handle the class-imbalance in the dataset of engineering symbols (at the  
 290 classification level), we are proposing to use a method similar to the MFC-GAN  
 model presented in [51]. This model is chosen due to the very minor and in  
 some cases subtle difference between different classes of symbols. MFC-GAN  
 model allows us to train the discriminator to classify not only real symbols  
 but also fake symbols, which provides more fine-grained discrimination between  
 295 instances.

For this work, the discriminator network is designed to have four convolution  
 layers with strides of two and batch normalization is used between layers. All  
 convolution layers are activated using Leaky ReLu with alpha set to 0.2, and  
 Sigmoid function is used in the final layer as the activation function.

300 The discriminator layers are shared with a classifier model that outputs  $2 \times N$   
 soft-max. Where  $N$  is the number of classes. We also designed the generator to  
 have one linear layer and five transpose convolution layers with strides of two  
 in each layer. Batch normalization was also used between adjacent layers and  
 all layers were activated using Leaky ReLu apart from the final layer which is

305 sigmoid activated.

Similar to most GAN models the generator’s input is a noise vector of size 100 and combined with symbol label encoding (see [51] for details). This label encoding is used to control the class-specific generation, which is essential for our experiment.

310 The generator output is a  $64 \times 64$  greyscale symbols image. For our experiments (following sections) we used a batch size of 100 and a learning rate of 0.001 which was experimentally chosen. Spectral normalisation was used in both the generator and the discriminator. The proposed model will be trained using Equations 2 and 3, 4.

$$\mathcal{L}_s = \mathbb{E}[\log P(S = real|X_{real})] + \mathbb{E}[\log P(S = fake|X_{fake})] \quad (2)$$

$$\mathcal{L}_{cd} = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C' = c'|X_{fake})] \quad (3)$$

$$\mathcal{L}_{cg} = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C = c|X_{fake})] \quad (4)$$

315 Where  $\mathcal{L}_s$  is used to estimate the sampling loss, which represents the probability of the sample being real or fake.  $\mathcal{L}_{cd}$  and  $\mathcal{L}_{cg}$  are used to estimate the classification losses over the generator and the discriminator.  $X_{real}$  represents the training data and  $X_{fake}$  is the set of generated images.

#### 4. Experiment & Results

320 Two experiments were carried out. The first experiment was designed to evaluate an end-to-end solution for recognising symbols in engineering drawings. We are assuming here, that locating and recognising these symbols will simplify subsequent tasks in a framework for analysing the whole drawings (i.e. detecting text, pipelines, etc...). This is simply because the majority of these types of  
325 drawings are made of symbols. The second experiment is separate and is focused on handling the class-imbalance problem using GAN-based methods.

#### 4.1. Symbols Recognition

In our dataset, the P&ID sheets were approximately 7500 x 5250 pixels in size. To use such image size in training data is computationally expensive and  
330 therefore each P&ID was split into 24 patches by dividing the original P&ID width by 6 and the height by 4. This gave a patch size of approximately 1250 x 1300 pixels. The annotation data for each patch was obtained using the annotations for the whole P&ID as discussed in the previous section.

For the training phase, we excluded symbols that overlapped multiple patches.  
335 After extensive experiments, it was decided to use the 3<sup>rd</sup> version of the YOLO framework which proved to be improving the detection rate of small objects compared to the first and second YOLO models [55], [27]. It is worth pointing out that the sizes of the various engineering symbols in our dataset are relatively small compared to the image size.

340 The YOLO architecture was customised for the purpose of this experiment. First the number of classes in each of the three YOLO layers was set to 25, and the number of filters was changed accordingly and was set to  $3 \times (Class_{no} + 5)$ , where the  $Class_{no}$  denotes the number of classes in the dataset.

The dataset was split approximately 90%:10% into training (155 P&IDs) and  
345 test (16 P&IDs) sets. A pre-trained Network was used and retrained using our dataset and all layers were fine-tuned. Darknet implementation of the YOLO was used in this experiment <sup>2</sup>. During the training process the network input size of 416 x 416 was adjusted after every ten batches; adjusting the input size during training was reported to improve object detection across different object  
350 scales [27]. The network was trained with a learning rate of 0.001 and training is stopped when the model was trained on 10,000 batches, (batch size of 64).

At testing time, the model input was adjusted from  $416 \times 416$  to  $2400 \times 2400$ . In this way, we were able to test on the original P&ID images and simplify symbol detection across a whole P&ID diagram in one step as opposed to combin-  
355 ing detections from the P&ID patches. For evaluation, symbols were compared

---

<sup>2</sup><https://github.com/AlexeyAB/darknet>, A. A.B., Darknet,(2019)

against the ground truth and the Intersection over Union *IOU* was set experimentally to 0.5. A simple front end was developed using Python Libraries and OpenCV<sup>3</sup> for visualisation and manual error analysis purposes.

#### 4.1.1. Results

360 The training accuracy achieved was  $\sim 96\%$ . On the testing set, 1352 symbols out of 1424 were correctly located and recognised with a testing accuracy equal to  $\sim 94.9\%$ . A heatmap of the confusion matrix for the testing set is presented in Figure 8. It can be seen and as expected that majority class instances were accurately detected and recognised. In other words, symbols with enough  
 365 examples in the training set were accurately recognised.

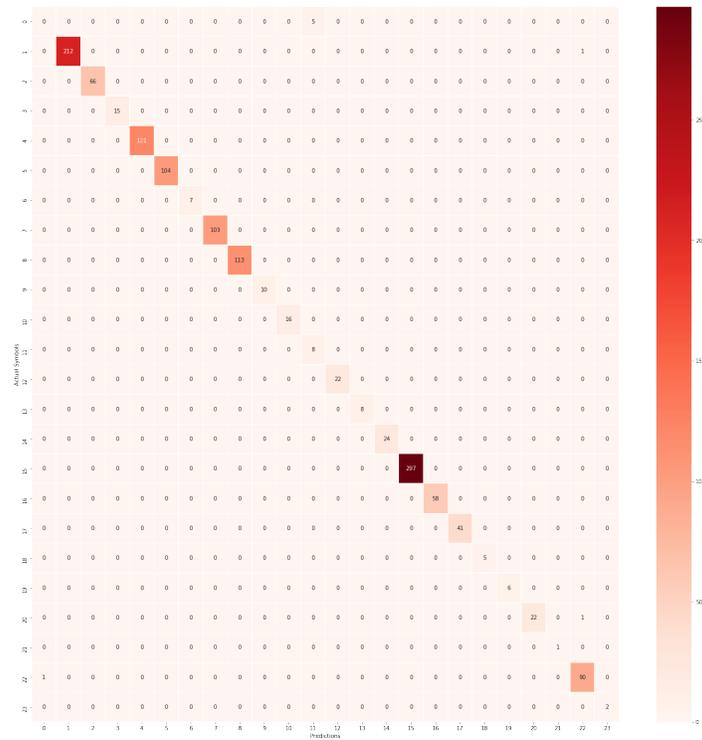


Figure 8: Heatmap of the Confusion matrix of the 25 symbols predictions (Testing Set)

<sup>3</sup><https://opencv.org/>

A typical output from the proposed methods where different symbols are highlighted in different colours is shown Figure 9 . Recognised symbols here were numbered and the predicted labels were recorded for further comparison against the ground truth. These symbols include inlets/outlets that are denoted by *label\_to*, *label\_from*, sensors, ball valve, reducers, gate valves, globe valves, and others.

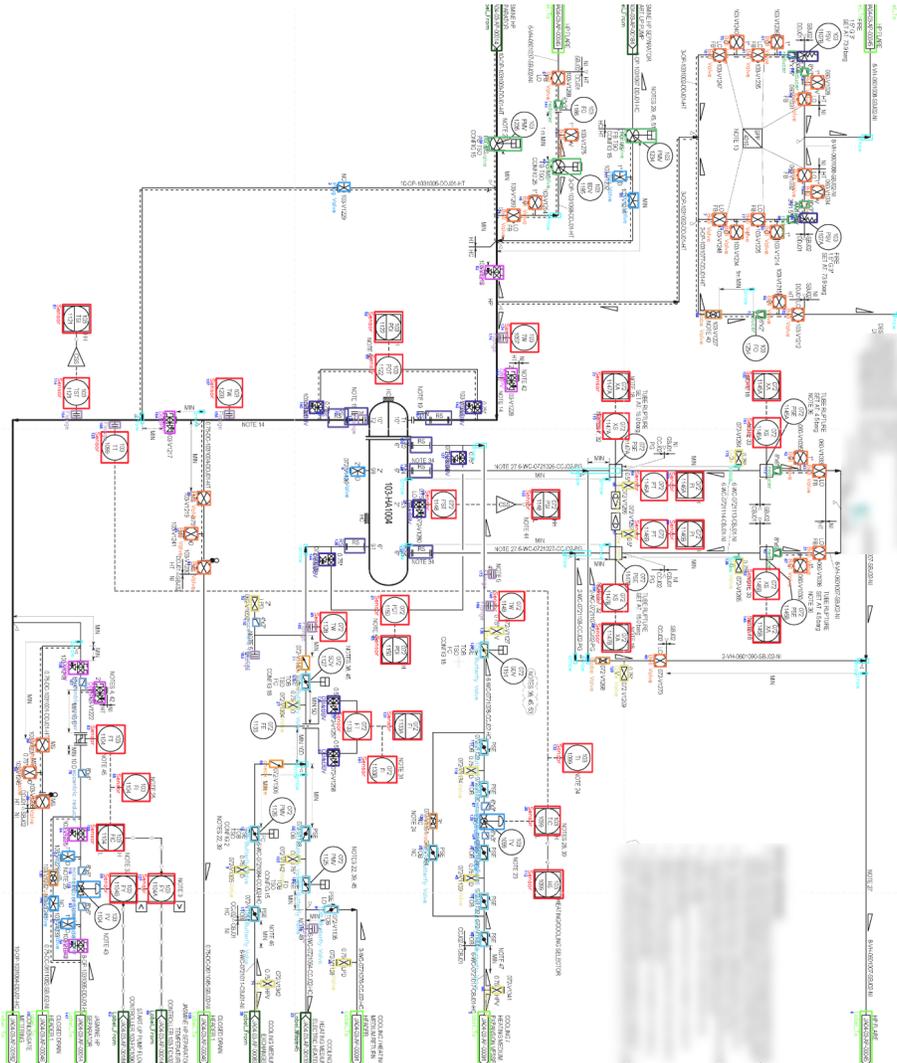


Figure 9: A P&ID diagram with various recognised symbols (Testing Set)

Table 1 provides more details about the number of instances of each symbol in the training and testing set named as *No of Training Symbols* and *No of Testing Symbols* respectively. Furthermore, it also shows the number of correctly recognised symbols in the testing set (*Correctly Recognised*) and the testing accuracy per class (*Class Accuracy*).

Table 1: Results of the proposed methods for symbols recognition (Testing Set)

Symbol	No of Training Symbols	No of Testing Symbols	Correctly Recognised (Testing)	Class Accuracy
Sensor	2810	302	297	98%
Ball Valve	1629	213	212	99%
Label_From	1347	103	103	100%
Label_To	1178	113	113	100%
Flange	1110	158	121	77%
reducer	821	91	90	99%
DB&BBV	542	67	66	98%
Gate Valve	535	110	104	94%
check valve	396	42	42	100%
TOB/Butterfly Valve	178	59	58	98%
Plug Valve	173	8	8	100%
Globe Valve	161	7	7	100%
Needle Valve	160	10	10	100 %
RS	143	26	24	92%
PSV	118	25	22	88%
eccentric reducer	98	23	22	96%
POB valve	84	16	16	100%
DBBPV	83	15	15	100%
PRV	32	8	8	100%
control valve globe	30	6	6	100%
control valve	22	5	5	100%
vent to atm	19	8	2	25%
injection/sample point	13	2	1	50%
Angle Valve	11	2	0	0.0%
BPRV	11	5	0	0.0%

Results show that the majority of instances were accurately detected and recognised (1352 of 1424). Figure 10 shows different symbols from various P&ID diagrams. Notice here that symbols are accurately detected and recognised regardless of its orientation. For example, reducers, gate valves, check valves, and others appear in different orientations (Figure 10). Similarly, sensors are accurately detected and recognised regardless of the text overlap with these instances. This clearly shows that unlike traditional methods the proposed

method is robust to these inherent vision challenges (at least in this context).

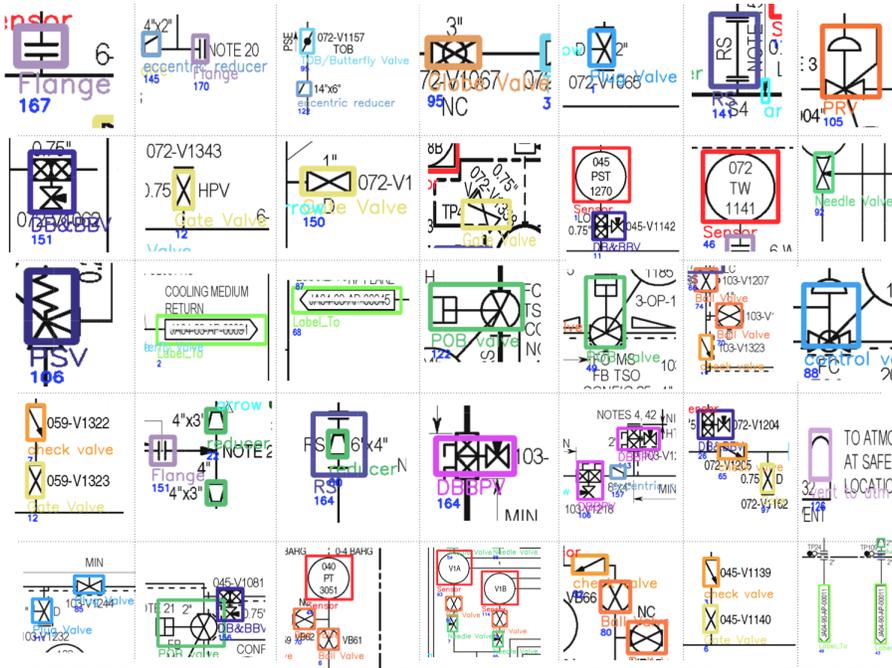


Figure 10: Examples of detected symbols

385 As can be seen in Table 2, in total 72 instances of the P&ID symbols were  
 either unrecognised at all (missed), or incorrectly classified as different symbols.  
 Of these, 8 instances of symbols were incorrectly classified (Table 2). Additionally,  
 64 symbols were completely missed. This can be largely attributed  
 to the nature of the drawings, wherein these cases symbols will have text and  
 390 annotation almost covering its entirety. This is evident if we look at the *IOU*  
 in Table 2 which is zero across all these missed symbols.

Table 2: Unrecognised and misclassified symbols in Engineering Drawings

Actual Class	No of Instances	Predicted Class	IOU
Ball Valve	1	reducer	0.81
BPRV	5	PRV	0.91
eccentric reducer	1	reducer	0.72
reducer	1	eccentric reducer	0.90
Angle Valve	2	-	0.00
Flange	37	-	0.00
Gate Valve	6	-	0.00
Sensor	5	-	0.00
TOB/Butterfly Valve	1	-	0.00
Vent to Atm	6	-	0.00
injection/sample point	1	-	0.00
PSV	3	-	0.00
RS	2	-	0.00
DB&BBV	1	-	0.00

Further visual analysis of the results presented in Table 2 showed that some symbols were incorrectly labeled. In particular, the instance of the symbol *Ball Valve*, although the model predicted the 'wrong' class symbol, visualising the results showed that the model actually predicted the right class for these symbols despite the wrong label. This is illustrated in Figure 11. Here, we use a simple front end to visualise the recognised symbols alongside an item number that we assign for each of them. This has greatly facilitated the analysis and visualisation of the results.

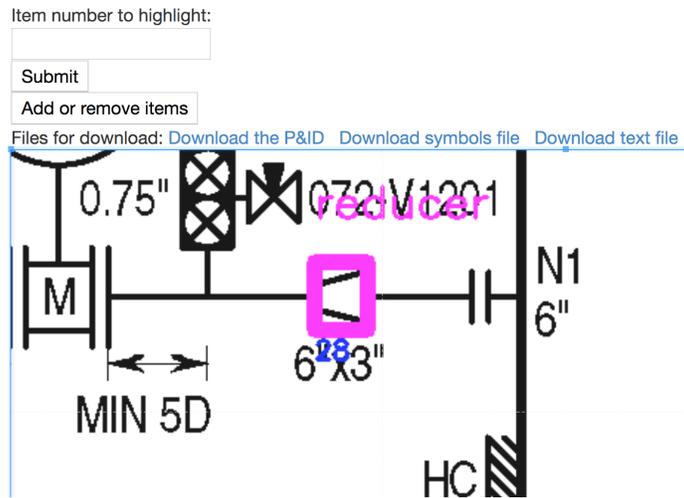


Figure 11: Incorrectly labeled symbol

400 Similarly, consider the symbol of class *BPRV* which was classified as *PRV*  
 in all five instances in the testing set. First, it is worth noting that the number  
 of training instances of this symbol is extremely low (11). Additionally, the  
 symbol is very much similar to the PRV class. However, it is anticipated that  
 more training examples of this symbol will certainly improve its detection rate,  
 405 as it is the case with most majority class symbols (i.e. Sensor, Ball Valve,  
 Reducer, Gate Valve, Check Valve, Globe Valve and so on). Figure 12 shows  
 samples of the BPRV symbols which were incorrectly classified alongside the  
 actual PRV symbol.

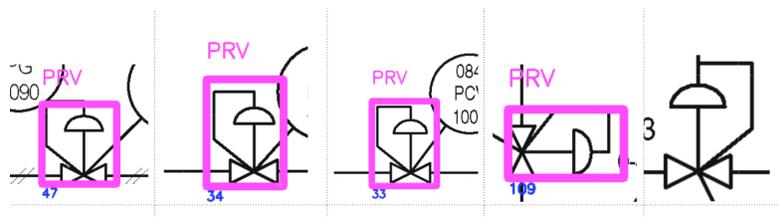


Figure 12: Incorrectly classified BPRV symbols (first four instances from left) as PRV symbol (fifth instance)

In summary, it can also be seen from the results presented in Table 1 that

410 the recognition rate of the symbols (*vent to atm*, *Angle Valve*, and *BPRV*) was quite low. This is mainly due to the limited numbers of training samples that represent these symbols. But overall, and excluding these three symbols, the average class accuracy of the remaining 22 symbols in the dataset is over 92% which is very encouraging results for such a challenging problem.

#### 415 4.2. MFC-GAN for Class-Imbalance

In this experiment we aim at evaluating a GAN-based model to handle the class-imbalance problem in the dataset. This is not a recognition task as in the first experiment but rather a classification problem. The experiment aims first at generating more symbols using MFC-GAN model. Then these synthesized samples will be used to augment the training set aiming at improving classification results.

The dataset used in this experiment is the almost the same one used in Experiment 1. All symbols were resized to  $64 \times 64$  grey-scaled images. The problem is formulated as a supervised learning task where the aim is to learn a function  $f(x)$  that maps an instance  $x_i$  of a particular engineering symbol to the corresponding class  $y_i$ . In this case,  $y_i \in Y$  where  $Y$  is a discrete set of classes representing the 29 symbols in the dataset. As discussed earlier the dataset is hugely imbalanced, and some of the instance that were dropped in Experiment 1 populates less than  $\sim 0.01\%$  such as angle choke valve.

430 The experiment was carried out in two stages, a GAN training stage and a classification stage. First, we trained MFC-GAN using all the samples in the dataset. The MFC-GAN model was conditioned to generate engineering symbols in extreme cases of class imbalance. To do so, we considered the least represented symbols in the whole dataset. These are Angle Choke Valve, Angle Valve, Injection Sample Point, Back Pressure Regulating Valve, PS Gate Valve, 435 Control Valve, Through Conduit Gate Valve, Control Valve Globe and Pressure Regulating Valve. These symbols have 2, 13, 15, 17, 17, 27, 31, 36 and 42 instances respectively in the training set. The model was trained only once on this dataset and the samples were generated after training was completed.

440 During training, the minority classes were resampled to encourage learning of minority instances structure.

The trained MFC-GAN model was then used to generate symbols of minority class instances (the least represented in the dataset, nine symbols). The original dataset was split 70% for training and 30% for testing set. Synthetic datasets  
445 were then added to the training set. For each minority class, 5000 more synthetic samples were added. This enabled us to rebalance the dataset by increasing the presence of the least represented symbols.

In order to evaluate the quality of the generated symbols, we build a classification model to compare performance before and after adding the generated  
450 symbols to the training set.

The classification model chosen is a CNN with 4 layers. The first three layers are convolution layers with 32, 64, 128 outputs. These layers have a kernel size of  $3 \times 3$ ,  $2 \times 2$  and max-pooling in-between them. The fourth layer is a fully connected layer with 256 units that feeds in to a 29-way SoftMax output  
455 representing the 29 symbol classes. The CNN was trained using SGD with a batch size of 64 and a learning rate of 0.001. Classification results were recorded using common metrics, namely true positive rate, balanced accuracy, G-mean and F1-Score.

#### 4.2.1. Results

460 Figure 13 compares the generated samples from MFC-GAN model with the original symbols from the diagram. We also report the symbols classification results in Table 3.

MFC-GAN generated far superior and more realistic samples. Visual inspection revealed distinct symbols features and the required categories were  
465 generated in each instance. Moreover, MFC-GAN high-quality samples had a positive effect on the performance of the classifier. For example, the G-Mean and sensitivity improved from 0 to 100% on angle choke valve as can be seen in Table 3 with just two instances of the classes. This result is consistent in seven of the nine minority classes. However, we observed that the model did

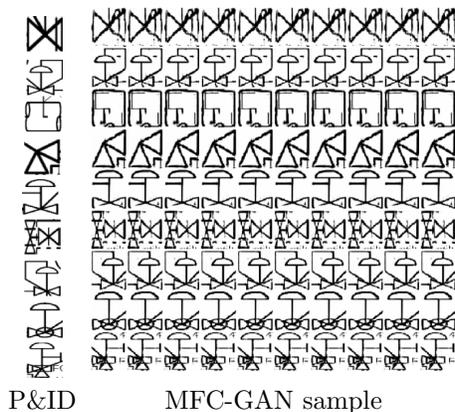


Figure 13: Comparing original P&ID samples with MFC-GAN generated samples.

not improve the baseline in the other two classes control valve and PRV classes. A closer look at Figure 13 revealed a high similarity between symbols. There is extreme similarity between angle valve (fifth symbol from the top) with control valve globe (eighth symbols from the top) and PRV (seventh symbol from the top) and BPRV (second symbol from the top). Although symbols were distinctly generated, the similarity of symbols dwindled the classification results in these classes. The low precision in BPRV and control valve globe classes from Table 3 further solidifies this observation.

Table 3: CNN performance on symbols classification.

Metric	Model	angle choke valve	Angle Valve	BPRV	control valve	control valve globe	injectionsample point	PRV	PS Gate Valve	TCGvalve
Sensitivity	Baseline	0.00	0.50	0.60	0.88	1.00	0.80	1.00	1.00	0.89
	MFC-GAN	<b>1.00</b>	<b>1.00</b>	<b>0.80</b>	0.88	<b>1.00</b>	<b>0.88</b>	0.77	<b>1.00</b>	<b>0.91</b>
Specificity	Baseline	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	MFC-GAN	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
Precision	Baseline	0.00	1.00	1.00	1.00	0.85	1.00	0.72	1.00	1.00
	MFC-GAN	<b>1.00</b>	<b>1.00</b>	0.67	<b>1.00</b>	0.92	<b>1.00</b>	<b>0.91</b>	0.83	<b>1.00</b>
F1-score	Baseline	0.00	0.67	0.75	0.93	0.92	0.89	0.84	1.00	0.94
	MFC-GAN	<b>1.00</b>	<b>1.00</b>	0.73	<b>0.93</b>	<b>0.96</b>	<b>0.93</b>	0.83	0.91	<b>0.95</b>
Accuracy	Baseline	0.50	0.75	0.80	0.94	1.00	0.90	1.00	1.00	0.95
	MFC-GAN	<b>1.00</b>	<b>1.00</b>	<b>0.90</b>	0.94	<b>1.00</b>	<b>0.94</b>	0.89	<b>1.00</b>	<b>0.96</b>
G-Mean	Baseline	0.00	0.71	0.77	0.94	1.00	0.89	1.00	1.00	0.94
	MFC-GAN	<b>1.00</b>	<b>1.00</b>	<b>0.89</b>	0.94	<b>1.00</b>	<b>0.93</b>	0.88	<b>1.00</b>	<b>0.95</b>

MFC-GAN models proved in this experiment to be able to generate minority class instances that are extremely under-represented in the dataset. The

480 quality of these samples was evaluated subjectively by inspecting the resulting  
samples, and objectively by measuring a classifier performance before and after  
adding the generated samples to the training sets. Results show clearly that  
performance improved across several common evaluation metrics. However, it  
has to be said that MFC-GAN is only one method that can be used to han-  
485 dle the class imbalance problem. Other possible methods can also be explored  
and utilized. Class-imbalance is a very well researched problem, and there is a  
wide range of methods that ranges from simple data augmentation, sampling to  
more advanced methods such as GAN [24]. For an extensive review of different  
possible methods, the reader is referred to [57].

## 490 5. Conclusion & Future Direction

In this paper, we proposed an end-to-end framework for processing and  
analysing complex engineering drawings. Thorough experiments using a large  
collection of P&ID sheets from an industrial partner showed that our method  
accurately recognises more than 94% of the symbols in the drawings. Advanced-  
495 bounding-box detection methods proved in our experiments that they perform  
accurately in such challenging tasks by recognising symbols of 25 different  
classes, despite the very little differences between some of these symbols. Ad-  
ditionally, we proposed a GAN-based model to handle class-imbalance in the  
symbols dataset. Our experiments demonstrated that our method was capable  
500 of generating plausible engineering symbols and also proved to be improving  
classification accuracy when augmenting the training set with this synthesized  
data. Experiments results show that the proposed GAN model can learn from  
a smaller number of training examples.

A future direction of this work will focus on utilising Generative Adversarial  
505 Neural Networks to generating symbols in a diagram context. In other words,  
generate part of the engineering diagram, and not only the symbols. This will  
greatly help in saving efforts needed for manual data annotation. Additionally,  
future work will include building a unified framework based on the proposed

methods to allow full processing and analysis of engineering diagrams such as  
510 P&ID. We hypothesize that the work presented in this paper will greatly simplify  
subsequent tasks such as text localisation and line detection.

### Acknowledgement

This work was supported by Scotland Data Lab Innovation Centre, Oil and  
Gas Innovation Centre and DNV GL.

### 515 References

- [1] S. Ahmed, M. Liwicki, M. Weber, A. Dengel, Automatic room detection and room labeling from architectural floor plans, in: 2012 10th IAPR International Workshop on Document Analysis Systems, 2012, pp. 339–343. doi:10.1109/DAS.2012.22.
- 520 [2] E. Elyan, C. M. Garcia, C. Jayne, Symbols classification in engineering drawings, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489087.
- [3] P. Vaxiviere, K. Tombre, Celesstin: Cad conversion of mechanical drawings, Computer 25 (7) (1992) 46–54. doi:10.1109/2.144439.
- 525 [4] K. N. Goh, S. R. Mohd. Shukri, R. B. H. Manao, Automatic assessment for engineering drawing, in: H. B. Zaman, P. Robinson, P. Olivier, T. K. Shih, S. Velastin (Eds.), Advances in Visual Informatics, Springer International Publishing, Cham, 2013, pp. 497–507.
- [5] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016,  
530 <http://www.deeplearningbook.org>.
- [6] S. D. Holcomb, W. K. Porter, S. V. Ault, G. Mao, J. Wang, Overview on deepmind and its alphago zero ai, in: Proceedings of the 2018 International Conference on Big Data and Education, ICBDE '18, ACM, New York, NY,

USA, 2018, pp. 67–71. doi:10.1145/3206157.3206174.

535 URL <http://doi.acm.org/10.1145/3206157.3206174>

- [7] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1480–1489.  
540 doi:10.18653/v1/N16-1174.

URL <https://www.aclweb.org/anthology/N16-1174>

- [8] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare, *Nature Medicine* 25 (1) (2019) 24–29. doi:10.1038/s41591-018-0316-z.  
545

URL <https://doi.org/10.1038/s41591-018-0316-z>

- [9] R. Girshick, Fast r-cnn, in: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 1440–1448. doi:10.1109/ICCV.2015.169.  
550

URL <http://dx.doi.org/10.1109/ICCV.2015.169>

- [10] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, *Pattern Recognition* 77 (2018) 354 – 377.  
555 doi:<https://doi.org/10.1016/j.patcog.2017.10.013>.

URL <http://www.sciencedirect.com/science/article/pii/S0031320317304120>

- [11] A. Ali-Gombe, E. Elyan, C. Jayne, Fish classification in context of noisy images, in: G. Boracchi, L. Iliadis, C. Jayne, A. Likas (Eds.), *Engineering Applications of Neural Networks*, Springer International Publishing, Cham, 2017, pp. 216–226.  
560

- [12] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324. doi:10.1109/5.726791.
- 565
- [13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9. doi:10.1109/CVPR.2015.7298594.
- [14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90. doi:10.1145/3065386.
- 570
- [15] U. Park, A. K. Jain, Face matching and retrieval using soft biometrics, IEEE Transactions on Information Forensics and Security 5 (3) (2010) 406–415. doi:10.1109/TIFS.2010.2049842.
- 575
- [16] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708. doi:10.1109/CVPR.2014.220.
- [17] S. Ren, K. He, R. Girshick, J. Sun, 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, NIPS’15, MIT Press, Cambridge, MA, USA, 2015, pp. 91–99. URL <http://dl.acm.org/citation.cfm?id=2969239.2969250>
- 580
- [18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, A. C. Berg, SSD: single shot multibox detector, CoRR abs/1512.02325. arXiv:1512.02325. URL <http://arxiv.org/abs/1512.02325>
- 585
- [19] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully

- 590 convolutional networks, CoRR abs/1605.06409. arXiv:1605.06409.  
URL <http://arxiv.org/abs/1605.06409>
- [20] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
- 595 [21] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, Face recognition: A literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458. doi:10.1145/954339.954342.  
URL <http://doi.acm.org/10.1145/954339.954342>
- [22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley,  
600 S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
- [23] P. Vuttipittayamongkol, E. Elyan, A. Petrovski, C. Jayne, Overlap-based undersampling for improving imbalanced data classification, in: H. Yin, D. Camacho, P. Novais, A. J. Tallón-Ballesteros (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2018, Springer International Publishing, Cham, 2018, pp. 689–697.
- 605 [24] P. Vuttipittayamongkol, E. Elyan, Neighbourhood-based undersampling approach for handling imbalanced and overlapped data, Information Sciences 509 (2020) 47 – 70. doi:<https://doi.org/10.1016/j.ins.2019.08.062>.  
610 URL <http://www.sciencedirect.com/science/article/pii/S0020025519308114>
- [25] E. Arroyo, A. Fay, M. Chioua, M. Hoernicke, Integrating plant and process information as a basis for automated plant diagnosis tasks, in: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA),  
615 2014, pp. 1–8. doi:10.1109/ETFA.2014.7005098.

- [26] C. F. Moreno-García, E. Elyan, C. Jayne, New trends on digitisation of complex engineering drawings, *Neural Computing and Applications* doi: 10.1007/s00521-018-3583-1.  
620 URL <https://doi.org/10.1007/s00521-018-3583-1>
- [27] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525. doi:10.1109/CVPR.2017.690.
- [28] A. K. Chhabra, Graphics Recognition Algorithms and Systems, in: Proceedings of the 2nd International Conference on Graphics Recognition (GREC'97 ), 1997, pp. 244–252. doi:10.1007/3-540-64381-8\_40.  
625
- [29] L. P. Cordella, M. Vento, Symbol recognition in documents: A collection of techniques?, *International Journal on Document Analysis and Recognition* 3 (2) (2000) 73–88. doi:10.1007/s100320000036.
- [30] D. Zhang, G. Lu, Review of shape representation and description techniques, *Pattern Recognition* 37 (1) (2004) 1–19. doi:10.1016/j.patcog.2003.07.008.  
630
- [31] S. V. Ablameyko, S. Uchida, Recognition of engineering drawing entities: Review of approaches, *International Journal of Image and Graphics* 07 (04) (2007) 709–733. arXiv:<http://www.worldscientific.com/doi/pdf/10.1142/S0219467807002878>, doi:10.1142/S0219467807002878.  
635
- [32] D. Blostein, General Diagram-Recognition Methodologies, in: Proceedings of the 1st International Conference on Graphics Recognition (GREC'95), 1995, pp. 200–212.
- [33] T. Kanungo, R. M. Haralick, D. Dori, Understanding Engineering Drawings: A Survey, in: Proceedings of the 1st International Conference on Graphics Recognition (GREC'95), 1995, pp. 119–130.  
640

- [34] C. R. Kulkarni, A. B. Barbadekar, Text Detection and Recognition: A Review, International Research Journal of Engineering and Technology (IR-JET) 4 (6) (2017) 179–185. 645
- [35] Y. Lu, Machine printed character segmentation - An overview, Pattern Recognition 28 (1) (1995) 67–80. doi:10.1016/0031-3203(94)00068-W.
- [36] S. Mori, C. Y. Suen, K. Yamamoto, Historical Review of OCR Research and Development, Proceedings of the IEEE 80 (7) (1992) 1029–1058. doi: 10.1109/5.156468. 650
- [37] C. Howie, J. Kunz, T. Binford, T. Chen, K. Law, Computer interpretation of process and instrumentation drawings, Advances in Engineering Software 29 (7) (1998) 563 – 570. doi:https://doi.org/10.1016/S0965-9978(98)00022-2. 655  
URL <http://www.sciencedirect.com/science/article/pii/S0965997898000222>
- [38] C. F. Moreno-García, E. Elyan, C. Jayne, Heuristics-Based Detection to Improve Text / Graphics Segmentation in Complex Engineering Drawings, in: Engineering Applications of Neural Networks, Vol. CCIS 744, 2017, pp. 87–98. 660
- [39] R. C. Gonzalez, R. E. Woods, Digital image processing, Prentice Hall, Upper Saddle River, N.J., 2008.  
URL <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>
- [40] L. Boatto, V. Consorti, M. Del Buono, V. Eramo, A. Esposito, F. Melcarne, M. Meucci, A. Morelli, M. Mosciatti, A. Spirito, M. Tucci, Detection and separation of symbols connected to graphics in line drawings, in: Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems, 1992, pp. 545–548. doi:10.1109/ICPR.1992.201837. 670

- [41] E. Elyan and M. M. Gaber, A genetic algorithm approach to optimising random forests applied to class engineered data, *Information Sciences* 384 (2017) 220 – 234. doi:<https://doi.org/10.1016/j.ins.2016.08.007>.  
URL <http://www.sciencedirect.com/science/article/pii/S0020025516305783>
- 675
- [42] A. Rebelo, G. Capela, J. S. Cardoso, Optical recognition of music symbols: A comparative study, *Int. J. Doc. Anal. Recognit.* 13 (1) (2010) 1931. doi:[10.1007/s10032-009-0100-1](https://doi.org/10.1007/s10032-009-0100-1).  
URL <https://doi.org/10.1007/s10032-009-0100-1>
- [43] W. Khan, D. Ansell, K. Kuru, M. Bilal, Flight guardian: Autonomous flight safety improvement by monitoring aircraft cockpit instruments, *Journal of Aerospace Information Systems* 15 (4) (2018) 203–214. arXiv:<https://doi.org/10.2514/1.I010570>, doi:[10.2514/1.I010570](https://doi.org/10.2514/1.I010570).  
URL <https://doi.org/10.2514/1.I010570>
- 680
- [44] A. Pacha, J. Haji, J. Calvo-Zaragoza, A baseline for general music object detection with deep learning, *Applied Sciences* 8 (9). doi:[10.3390/app8091488](https://doi.org/10.3390/app8091488).  
URL <http://www.mdpi.com/2076-3417/8/9/1488>
- 685
- [45] F. D. Julca-Aguilar, N. S. T. Hirata, Symbol detection in online handwritten graphics using faster R-CNN, *CoRR* abs/1712.04833. arXiv:[1712.04833](https://arxiv.org/abs/1712.04833).  
URL <http://arxiv.org/abs/1712.04833>
- 690
- [46] A. Pacha, K. Choi, B. Coasnon, Y. Ricquebourg, R. Zanibbi, H. Eidenberger, Handwritten music object detection: Open issues and baseline results, in: *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 163–168. doi:[10.1109/DAS.2018.51](https://doi.org/10.1109/DAS.2018.51).
- 695
- [47] R. Rahul, S. Paliwal, M. Sharma, L. Vig, Automatic information extraction from piping and instrumentation diagrams, *CoRR* abs/1901.11383. arXiv:

1901.11383.

700 URL <http://arxiv.org/abs/1901.11383>

- [48] A. Ali-Gombe, E. Elyan, Y. Savoye, C. Jayne, Few-shot classifier gan, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489387.
- [49] A. Ali-Gombe, E. Elyan, C. Jayne, Multiple fake classes gan for data aug-  
705 mentation in face image dataset, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8. doi:10.1109/IJCNN.2019.8851953.
- [50] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, arXiv preprint arXiv:1711.04340.
- 710 [51] A. Ali-Gombe, E. Elyan, Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network, *Neurocomputing* 361 (2019) 212–221.
- [52] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.
- 715 [53] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, *International conference on machine learning*, page 2642–2651 70 (AUG 2017) 2642–2651.
- [54] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, C. Malossi, Bagan: Data augmentation with balancing gan, arXiv preprint arXiv:1803.09655.
- 720 [55] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.
- 725 [56] E. Arroyo, X. L. Hoang, A. Fay, Automatic detection and recognition of structural and connectivity objects in svg-coded engineering documents, in:

2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA), 2015, pp. 1–8. doi:10.1109/ETFA.2015.7301510.

- [57] Learning from class-imbalanced data: Review of methods and applications, *Expert Systems with Applications* 73 (2017) 220 – 239.