AUTHOR(S): ELYAN, E., MORENO GARCIA, C. and JAYNE, C.

TITLE: Symbols classification in engineering drawings.

YEAR: 2018

Publisher citation: ELYAN, E., MORENO GARCIA, C. and JAYNE, C. 2018. Symbols classification in engineering drawings. Presented at the International joint conference on neural networks 2018 (IJCNN), 8-13 July 2018, Rio de Janeiro, Brazil.

OpenAIR citation: ELYAN, E., MORENO GARCIA, C. and JAYNE, C. 2018. Symbols classification in engineering drawings. Presented at the International joint conference on neural networks 2018 (IJCNN), 8-13 July 2018, Rio de Janeiro, Brazil. Held on OpenAIR [online]. Available from: https://openair.rgu.ac.uk

Symbols Classification in Engineering Drawings

Eyad Elyan
School of Computing Science
and Digital Media
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: e.elyan@rgu.ac.uk

Carlos Moreno Garcia
School of Computing Science
and Digital Media
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: c.moreno-gracia@rgu.ac.uk

Chrisina Jayne
School of Engineering,
Mathematics and Computing
Oxford Brookes University
Oxford,OX3 0BP , UK
Email: cjayne@brookes.ac.uk

***Abstract*—Technical drawings are commonly used across different industries such as Oil and Gas, construction, mechanical and other types of engineering. In recent years, the digitization of these drawings is becoming increasingly important. In this paper, we present a semi-automatic and heuristic-based approach to detect and localise symbols within these drawings. This includes generating a labeled dataset from real world engineering drawings and investigating the classification performance of three different state-of the art supervised machine learning algorithms. In order to improve the classification accuracy the dataset was pre-processed using unsupervised learning algorithms to identify hidden patterns within classes. Testing and evaluating the proposed methods on a dataset of symbols representing one standard of drawings, namely Process and Instrumentation (P&ID) showed very competitive results.**

I. Introduction

Engineering drawings are commonly used across different domains such as Oil and Gas, mechanical engineering [1], logical circuits representation [2] and others. Attempts aiming at digitising these drawings can be traced back to the late 80's [3], and the 90s, [4] [5], [6], [7].

In recent years, the digitisation of these drawings is becoming increasingly important and attracting more attention from the research communities [8], [9], [10], [11]. This is partly due to the legacy and rich source of information that these drawings can provide, and also due to the advances in hardware and underlying machine learning and vision methods.

Process and Instrumentations (P&ID) diagrams such as the ones shown in Figure 1 represents one class of such drawings. These can be defined as schematic diagrams representing the different components of the process and the connectivity information. Digitising these drawings also received large attention from a commercial standpoint[1, 2, 3] given the wide range of applications that can be developed from a digital output, such as security assesment, graphic simulations or data analytics.

More than thirty years ago, Furuta et al. [12] and Ishii et al. [13] proposed methods towards implementing a fully automated P&ID digitisation framework. These approaches have now become obsolete given the incompatibility with current software and hardware requirements. Around ten years later, Howie et al. [14] presented a semi-automatic method in which symbols of interest were localised using the template of the symbols as input. Most recently, Gellaboina et al. [9] presented a symbol recognition method which applied an iterative learning strategy based on the recurrent training of a neural network (NN) using the Hopfield model. This method was designed to find the most common symbols in the drawing, which were characterised by having a prototype pattern.

Broadly speaking, processing and analysing these drawings is very much similar to any typical image-processing task, where the aim is to find the object/s of interest, and then classify these objects. However, the digitisation of engineering drawing proved to be more challenging. For example, it is estimated that on average a single P&ID drawing contains around 100 different types of shapes [11]. These could be symbols of a specific types (i.e. valves, compressors, etc), text, annotations, and others. Another challenging problem with these types of drawings is the presence of large amount of connecting lines. These represent both physical and logical relations between symbols and are often depicted using lines of different styles and thickness. For example, dotted lines, dashed lines, and lines with and without arrows (Figure 1).

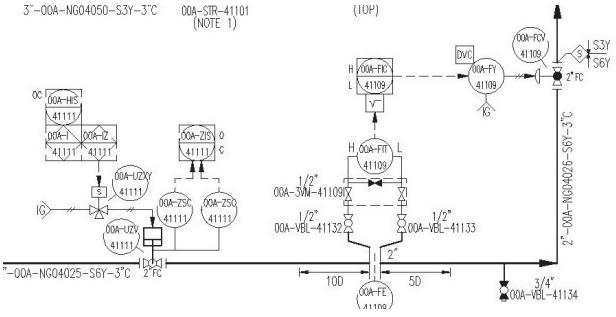

Fig. 1. Class distribution in the dataset

Classifying objects of interest is another potential challenge in the digitisation process. This is due to the within-class and cross-class similarity. Figure 2 shows a subset of the standard symbols that may appear in any P&ID drawing, while the highlighted areas shows symbols that have been extracted from a typical drawings. It can be seen that these symbols exist in

[1]http://www.pidpartscount.com
[2]http://www.radialsg.com/viewport
[3]https://www.rolloos.com/en/solutions/analytics-documents/viewport

any drawings in different orientation, and may be occluded by text, or other symbols, which adds more complexity to the classification task. Another challenge with the classification of these symbols is the lack of a benchmark and publicly available dataset. This makes it difficult to compare results and performance of algorithms. Finally, not only that here is no publicly available dataset, but in fact there is no one single data repository that is structured and labelled, which could be used for evaluating and testing different machine learning algorithms.

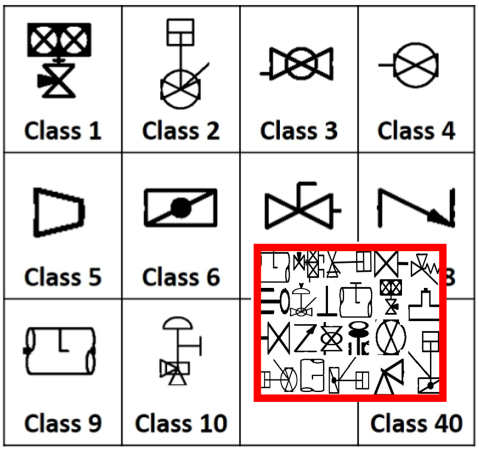

Fig. 2. Class distribution in the dataset

It can be argued that despite the massive advancement in machine vision and machine learning algorithms, real applications that are required by important industries such as the Oil and Gas for example, haven't yet benefited from these advances. This is in particular the case with Engineering and P&ID drawings, where large volumes of such data exists, however, it is not utilised at all despite the urgent need for having methods and techniques to transform such unstructured volumes of data into knowledge. In this paper, we are proposing a semi-automatic method for detecting and localising symbols within engineering drawings, and we evaluate three state-of the art supervised machine learning algorithms against a real dataset of symbols that have been extracted from a collection of P&ID drawings. The main contributions of the paper can be outlined as follows:

- A heuristic-based approach to localise and detect symbols in P&ID drawings, and use these symbols to create a structured and labelled repository of symbols that can be use for classification purposes
- Apply state-of the art machine learning methods to address an overlooked and important industrial problem aiming at classifying symbols in engineering drawings
- Transform the dataset into a decomposed set by means of $kmeans$ clustering to identify genuine subclasses within the class symbols and significantly improve the models performance
- Establish the importance of class decomposition in classification of symbols in engineering drawings by carrying out extensive experiments using an experimental framework for validating results on a dataset of 1187 symbols extracted from P&ID drawings

The rest of this paper is organised as follows: Section 2 gives the necessary background and discusses related literature. Section 3, discusses the dataset used and the proposed methods, while section 4 presents the experimental framework for validating results. Section 5 concludes the paper and discusses possible future direction.

II. Related Work

Engineering drawings are very common across several industries, such as Oil and Gas, constructions, planning and others. These drawings can be defined as a schematic representation, which depicts the flow or constitution of a circuit, device, process or facility. Some examples of these drawings include logical gate circuits, mechanical or architectural drawings, P&ID drawings and others. There is an increasing demand in different industries for developing digitisation frameworks for processing and analysing these diagrams. Having such framework will provide a unique opportunity for relevant industries to make use of large volumes of diagrams in informing their decision-making process and future practices.

Digitising and analysing engineering drawings require applying a set of image processing techniques through a sequence of steps including pre-processing, symbol detection and localisation, and classification. In other words, several common image pre-processing and analysis steps can be borrowed from other domains and applied to the digitisation of engineering drawings such as analysis of musical notes [15], processing and conversion of paper-based mechanical drawings into CAD files [16], optical character recognition (OCR) [17], [18], [19], and others.

Due to the complexity of these drawings, having a fully automated framework for reading, processing and analysing such drawings is still far from being reality. In some types of these diagrams, for example P&IDs, part of the digitisation also requires intelligence inference (i.e. relations between symbols and pipelines within the drawings [11]). This adds more complexity to the digitisation process, and often requires manual intervention. That said, there are many review papers in the literature that addresss specific component/s of the digitisation of these drawings, such as symbols detection [20], [21], symbols representation [22], and symbols classification [23], [8].

Deep Convolutional Neural Networks (CNN) have achieved tremoundus progress in the machine vision domain, where orders of magnitued of improvement in classification of objects in images were recorded [24]. It has been sucessfully applied across several domains such as document recognition [25], image classification [26], [24], and other machine vision related problems.

Despite its power and success in recent years, the straightforward application of CNNs for the digitization of engineering drawings is still a challenging task. This mainly

due to the complexity of the problem and also due to the lack of sufficient annotated examples or publicly available datasets. Another reason, is that there are no clear guidelines on how to interpret these drawings, for example in the case of P&IDs where inference is also required as part of the digitisation process. Despite these difficulties, there are some methods where CNNs have been applied to specific task of the engineering drawings digitisation process. For instance, Fu et. al [27] presented a CNN based method to recognize symbols in engineering drawings produced by computer-aided systems or hand sketches and convert them into CAD designs. The method proposed requires large amount of training data to achieve acceptable level of accuracy.

It can be argued that despite the recent significant advances in image processing, and in particular in Deep Neural Networks, automatic analysis and processing of these engineering drawings is still far from being complete.

III. Methods

A. Symbols Detection

A set of heuristic-based methods have been developed and applied sequentially to localise symbols within a collection of P&ID drawings provided by an industrial partner. By applying these heuristics sequentially, it was possible to extract the main components of the engineering drawings. These include circles which constitute the most frequent existing symbols within P&ID drawings and often referred to as sensors, text, lines, and then the rest of all other symbols as will be discussed later.

Following the work presented in [11] and as part of the pre-processing stage, a thresholding method was first applied to reduce the noise. Areas of interest of the drawing were then identified interactively to discard boundaries, text and annotation outside the border of the drawing. This was followed by applying a method based on blurring the image and circle detection through the Hough transform [28] to identify all symbols of a class known as sensors. It is worth pointing out that each detected sensor (circle) contains text within it. Based on the average sizes of the text within the detected circles, two values were set empirically as H_{av}, W_{av} to represent height and width of a text character in the drawing respectively. Using these empirical values and in order to separate text from the engineering drawing, a text/graphics segmentation method based on [29] was implemented. The CC generation algorithm described in [30] which involves grouping together black pixels which are *eight-connected* to one another was used to select all contours all contours with an approximate area of $A_{av} = H_{av} \times W_{av}$. Then, each of these contours were defined as either:

- Noise if the area enclosed within these contours falls below a certain threshold value. For the set of P&ID drawings we used, this value was found empirically to equal $A_{av}/4.0$
- Small elongated component such as text characters (i.e. I, l,$-$,$/$, etc...) and dashed segments (which often constitute lines) were defined as in Equation 1, where t is a threshold value

$$\frac{max(H_{av}, W_{av})}{min(H_{av}, W_{av})} \leq t \tag{1}$$

- Otherwise, the contour would be considered as text.

All candidate text characters were then grouped into strings following the work presented in [31], and strings were expanded by a factor of $1.5 \times W_{av}$ to account for any false positive noises or small elongated components which were true text characters. It is worth pointing out that with such approach, we are not taking into account any potential character/text overlapping. This is simply because we haven't encountered such problem within the collection of drawings we used for the experiment. However, the method can easily be expanded to take such problem into consideration with methods such as [32]. Once the above heuristics are applied, all the elements of the drawings such as text, circles, dashed segments, etc... are extracted. This results in an image with a set of lines and symbols as can be seen in Figure 3.

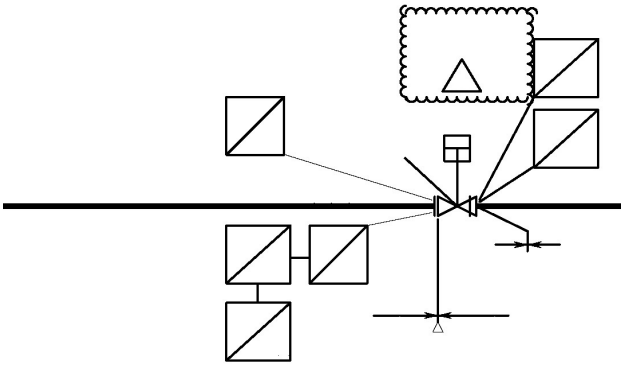

Fig. 3. Processed P&ID drawing

This image is used as an input for the following state, where the line detection algorithm in [33] is applied to identify line segments which length and thickness exceeds certain threshold values. These lines constitutes the portion of the P&ID often referred to as the pipework, which is the section where all symbols of interest are attached. As a result, all remaining contours which lie between the pipework were isolated and stored in the symbol repository. Figure 4 shows a portion of the resulting P&ID drawing where elements such as the ones below are detected:

- Sensors (blue circle).
- Text (green).
- Pipe lines (brown).
- Symbols of interest (red)

It is important to note that the labelling of these symbols have been done manually with experts from the industry due to the lack of a labelled dataset. Also, it is worth pointing out that the approach presented above applies only to one standard of P&ID drawings, hence, such approach may require

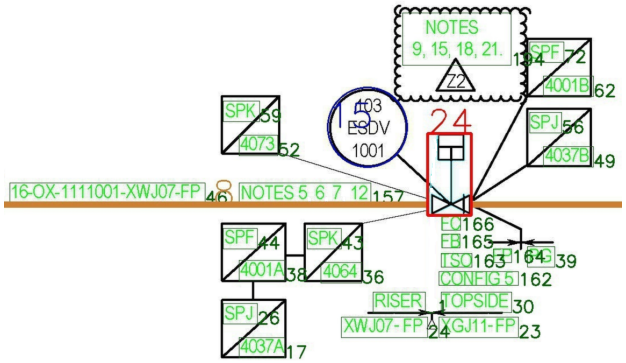

Fig. 4. Elements detected in a P&ID drawing

customisation, or extension to account for other standards of P&ID and for other types of engineering drawings.

B. Dataset

Using the method presented above, a collection of P&ID drawings have been processed and analysed. This resulted in a collection of symbols that represent different types of equipments within the drawings. These have been scaled to a standard size of 100×100 pixels as can be seen in Equation 2, where n is the total number of symbols, and x_i^j represents the j^{th} pixel value of the i^{th} symbol or instance in the dataset.

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^{10000} \\ x_2^1 & x_2^2 & \dots & x_2^{10000} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & \vdots & \dots & x_n^{10000} \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ .. \\ .. \\ y_m \end{bmatrix} \quad (2)$$

The dataset contains 1187 symbols and shapes distributed over 37 different classes. These include valves, connectors and types of components.

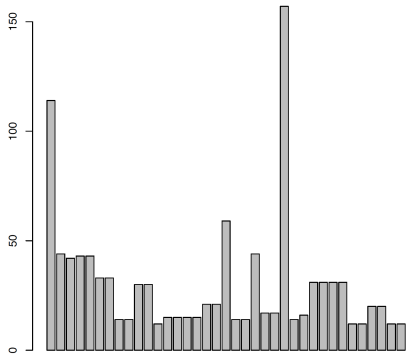

Fig. 5. Class distribution in the dataset

The distribution of these symbols is shown in Figure 5. The average number of instances per class is 30.21, while the standard deviation is 28.6. It can be noted that the data is not hugely imbalanced, however, some classes are way beyond the average with 114, and 157 instance/ class.

C. Data Preprocessing

Before applying classification models, and aiming at improving prediction accuracy, we applied class decomposition to the dataset of symbols as a preprocessing step. Class decomposition is the process breaking down labelled datasets to a larger number of subclasses by means of applying clustering to the instances that belong to one class at a time. As such, the decomposition can be applied to one or more class/s in the dataset [34], [35] by applying unsupervised learning algorithms (i.e. $kmeans$). In other words, for a set of instances $X_i = x_1, x_2, ..., x_n \in Y_1$ belonging to a dataset A, where Y_1 is the class label, then by decomposing X into a set of subclasses we can obtain a new set of class labels $Y_{11}, Y_{12}, ..., Y_{1K}$. This approach can be tracked back to 2003 [36], and was presented then to mitigate the issue of low variance classification methods.

The motivation behind adopting such approach is that genuine subclasses can be detected and as such improving the classification accuracy. In [35], Random Forests over class decomposed medical diagnosis data sets has been adopted. In this work, the authors performed an exhaustive search over a set of iterations to find the best k values for each class and then decomposed the classes accordingly. A heuristic was used to discard minority classes from the decomposition process. Experiments showed that by decomposing the datasets into subclasses favorable results can be achieved. The impvement of the resulted was attributed to the diversified search space resulting from the decomposition process. In [34], an evolutionary-based method namely Genetic Algorithm was used to optimise a set of parameters including the best k values, and again an improved classification accuracy was achieved when the proposed method was tested on 22 different life science and medical datasets.

In this paper class decomposition is adopted to boost classification accuracy. The motivation behind adopting this approach is that by applying class decomposition to the symbols dataset will help identify any hidden pattern within the class symbols, diversify the search space and potentially improve classification accuracy. .Unlike the work presented in [36], [35] and more recently in [34], class decomposition in this paper is achieved by computing the k values based on the average number of instances /class as shown in Equation 3 .

$$k_i = \left\lfloor \frac{c_i}{A_{vg}} + 1 \right\rfloor \quad (3)$$

where c_i is the total number of instances of a specific class, and A_{vg} is the mean of the class distribution in the dataset. This ensures that only classes that exceeds the average class distribution will be subject to decomposition (clustering). Also according to this formula, the maximum value k can take is dependant on the maximum number of instances per class. By adopting this approach, we end up having a dataset of 57 different class and with different class distribution as can be seen in Figure 6. It is also worth pointing out that the mean

of the class distribution is now reduced to 19.61 with standard deviation equals to 7.80.

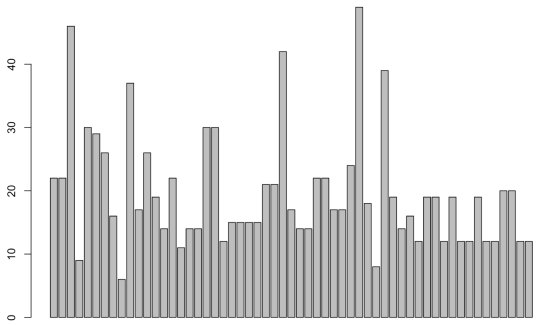

Fig. 6. Class distribution in the decomposed dataset

Upon decomposing the dataset, and for a any set of instances belonging to class label (Y_i), these instances are now assigned to different clusters within this class ($Y_{i1}, Y_{i2}, , Yik_i$) which constitute the new class labels for these instances. Notice that k value here could take any value that ranges from 1 which means apply no decomposition (i.e. clustering) to this class, all the way up to a $maxK$ which is bounded by Equation 3. It is worth pointing out that with such an arrangement, for any classifier $h(x)$ where x belongs to class y_i , $h(x) = y_{ij}$ is considered as a correct classification $\forall j \in y_i$ subclasses. This requires different approach to compute the classification accuracy.

TABLE I
CONFUSION MATRIX OF $h(x)$

	y_{11}	y_{12}	y_{13}	...	y_{i1}	y_{i2}
y_{11}	**1**	**3**	**4**	...	0	10
y_{12}	**5**	**4**	**10**	...	4	4
y_{13}	**17**	**3**	**20**	...	4	4
⋮	⋮	⋮	⋮	⋮	⋮	
y_{i1}	9	8	4	...	**5**	**7**
y_{i1}	1	3	4	...	**0**	**10**

Consider Table I, and assume that this is the resulting confusion matrix of the classifier $h(x)$, then classification accuracy (A_{cc}) can be computed as follows:

$$A_{cc}(h) = \frac{\sum_{i=0}^{nClasses} \sum_{j=0}^{k_i} h_c(i, j + [k_{i-1} * i])}{m} \quad (4)$$

Where m is the number of instances in the dataset, and $nClasses$ represents the number of discrete classes in the data set, while k_i represents the number of clusters applied to each class. In short, Equation 4 will result in summing all the **bold** elements of the confusion matrix in Table I and divide it by the m (total number of instances in the dataset).

D. Classification Models

Three different models have been used to classify symbols in the dataset. These have been also used to assess and evaluate the impact of decomposing classes on classification accuracies. The models are Random Forest (RF), Support Vector Machine (SVM), and Deep Convolutional Neural Networks (CNN).

Random Forests is an Ensemble classification that proved to be highly accurate prediction and classification technique. According to the winning solutions in *Kaggle*[4], the state-of-the-art ensemble methods are Random Forests [37] and Gradient Boosting trees [38]. It has also proved superiority experimentally in a relatively recent large experiment when compared with all widely adopted classifiers (179 classification model) using 121 different dataset from the UCI repository[5] [39].

Support Vector Machines (SVM) [40] is another supervised machine learning algorithm that boosts classification accuracy by projecting the data points to a higher dimensional space aiming at finding an optimal hyperplane that separates positive and negative classes. SVM has also proven its superiority over other classification methods. In [39] and when compared to other widely adopted learning algorithms, SVM with Gaussian kernel ranked second after Random Forests without statistically significant difference.

Deep Convolutional Neural Network was chosen in this paper for its recent success in particular in the machine vision domain [24]. CNN architecture for recognizing visual patterns was first proposed by Fukushima under the name Neocognitron [41]. Since then, Deep convolutional neural network algorithms have been successfully applied for document recognition [25], image classification [26], [24], and other vision related problems. The typical CNN architecture consists of an input layer, hidden layers made of convolutional, pooling and fully connected layers, and an output layer. The convolutional layers enable sharing of weights and detecting the same patterns in different parts of the image while the pooling layers merge similar features and create an invariance to small shifts and distortions [42]. The CNNs are easier to train as they have fewer parameters than fully connected networks with the same number of hidden units. CNNS are trained using back propagation algorithms similarly to fully connected neural networks [42].

E. Experimental Framework

A framework for implementing and evaluating the learning algorithms and establish the impact of class decomposition on classification accuracy was designed. Here, the dataset was decomposed using $kmeans$ algorithm where the k values for each class was computed based on Equation 3. As can be seen in Algorithm 1, for any given dataset A, first A is decomposed into a new dataset denoted by A_c, the learning algorithm ML is then applied on the original dataset A and then applied on the decomposed dataset A_c, results are then returned for comparison purposes.

[4]Kaggle: www.kaggle.com
[5]UCI repository:http://archive.ics.uci.edu/ml/

Algorithm 1 Compute Classification Accuracy

Data: Dataset, ML
Result: Accuracy
begin
  $A \longleftarrow Dataset$;
  $A_c \longleftarrow decomposeSet(A, K_{values})$;
  $model \longleftarrow Model(A, ML)$;
  $model_c \longleftarrow Model(A_c, ML)$;
  $r = Accuracy(model)$;
  $r_c = Accuracy(model_c)$;
  $return(r, r_c)$;
end

With this arrangement, the aim is to evaluate the classification of symbols in engineering drawings and also to asses and evaluate the impact of class decomposition on classification accuracy.

IV. Experiment

A. Setup

Extensive experiments were carried out to establish the validity and stability of the proposed method using repeated hold-out approach. The dataset was split into training and testing sets where 80% of the data was used for training and the remaining 20% for testing. The dataset was decomposed by means of $kmeans$ clustering as discussed in the previous section. RF, SVM and CNN denotes the application of the classification models discussed in the previous section, while RF_c, SVM_c and CNN_c denotes the application of the classification models (RF, SVM, and CNN) on the preprocessed dataset (decomposed dataset).

The parameters settings for each learning model were kept the same in both experiments. Number of trees for Random Forest (RF) was chosen to be equal to 500, while the $mtry$ was set to equal the square root of the total number of features (this is the default settings in Random Forest). Same settings were used in RF_c. Support Vector machines with Gaussian kernels were used in both experiments SVM and SVM_c.

The CNN architecture in this experiment consists of the input layer [100x100] of the raw pixel values of the image; conv layer of 32 (5x5) filters; *relu* layer which applies an element wise activation function the $max(0, x)$; max pooling layer (2,2); fully-connected layer of 300 hidden units and the output layer of 37 units with softmax activation function. The CNN_c has similar architecture except the fully connected layer consists 500 hidden units and the output layer 57 units. The parameters of the networks were established experimentally using (3x3), (5x5) and (7x7) filters, and 200, 300, 400, 500 hidden units. Dropout was used in the max pooling layer and in the fully connected layer with rates 0.25 and 0.5 respectively. Dropout [43] is a regularization method that sets to zero the activations of the hidden units stochastically and it is used to address the over-fitting problem. It is also considered as a form of model averaging of training a large collection of networks with extensive weight sharing.

B. Results & Discussion

Table II shows the results of the 10 runs of the experiment across the three classification models (SVM, RF and CNN). It can be seen that overall, SVM and RF models performed better when they were applied on the preprocessed (decomposed) dataset.

TABLE II
Results of 10 runs on the dataset with and without Clustering

Run	CNN	CNN_c	RF	RF_c	SVM	SVM_c
1	93.75	92.41	96.17	98.56	95.07	96.41
2	94.20	99.55	95.69	97.61	95.96	96.41
3	98.21	96.88	97.13	97.13	93.72	94.62
4	98.21	95.98	95.69	97.61	96.41	96.86
5	95.09	95.54	95.69	98.56	93.72	95.96
6	94.64	93.30	96.17	97.13	95.52	96.41
7	95.98	96.88	95.22	96.65	95.52	95.52
8	97.32	94.64	97.61	98.09	95.96	98.21
9	93.75	96.43	95.69	98.09	96.86	96.41
10	97.32	94.20	96.17	97.13	94.17	95.96

Figure 7 shows the median results of the 10 runs in Table II. This clearly indicates the benefits of preprocessing and decomposing the dataset as described in the previous section where significant improvement in the classification accuracy has been achieved when using Random Forest or Support Vector Machines with Gaussian kernels. It can also be noted from Figure 7 that although the median of CNN is slightly higher than CNN_c, CNN overall performs better on the original dataset (on average) than when applied on the decomposed dataset. This could be attributed to the limited number of instances in the dataset, compared to the requirements of CNNs in terms of large collection of instances.

Summary statistics shown in Table III shows that on average the classification accuracy benefits from decomposing the dataset, apart from the case when applying CNN method. It is also worth pointing out that the reported standard deviation of the methods indicate stability of the proposed method.

TABLE III
Summary of the results

Stat	RF	RF_c	SVM	SVM_c	CNN	CNN_c
Avg	96.12	97.66	95.29	96.28	95.84	95.58
SD	0.73	1.10	0.92	0.02	1.79	2.05

To assess the statistical significance of the results, $t-test$ was carried out to compare the performance of each model's performance on the two datasets (original one and the decomposed one) using the results presented in Table II with with 95% confidence interval. In other words comparing RF against RF_c, CNN against CNN_c and so on. It was found that the performance of RF is significantly better when applied on the decomposed dataset with a $p-value$ of 0.0005339, and similar results were obtained for SVM with $p-value$ equals to 0.007869 which indicates evident improvement of the mode. Although CNN performed better than CNN_c on average, however the results are not statistically significant with $p-value$ equals to 0.757, and this could be attributed to the size of the symbols dataset.

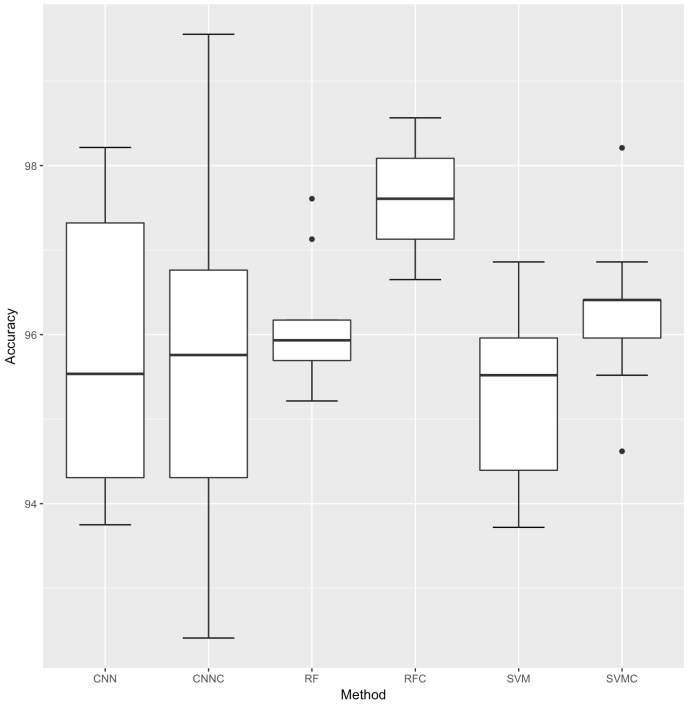

Fig. 7. Summary results of clustering datasets across the three classification models

The results clearly indicate that classification accuracy is hugely benefiting from decomposing the dataset of the symbols. As was shown, the performance of the two models (RF, SVM), improved when was applied on the decomposed dataset in comparison with performance on the original dataset. It is important to point out here that the purpose of this experiment wasn't to compare the performance of these models against each other. Therefore, while RF appear to be outperforming the other two models on this dataset, these results are not conclusive, taking into consideration that CNN's often require larger volumes of data to achieve good performance. This opens a platform for a future research direction where the aim would be to assess the impact of class decomposition on the performance of CNNs when enough data is available. To overcome the limited amount of available data, data augmentation [44] which proved to be improving the robustness and the training of the neural networks might be utilised.

V. Conclusion

Despite the recent advancements in the domain of machine vision, automatic processing and analysis of engineering drawings is still one of the challenging tasks. This is due to the lack of standard benchmark datasets and the inherent complexity of these drawings.

In this paper, we presented a semi-automatic and heuristic-based approach to localise symbols within these drawings. This method was then used to create a dataset of 1187 instances of these symbols. Three state-of the art machine learning methods (RF, SVM and CNN) were then applied and relatively accurate results were obtained. Classification accuracy was then boosted and significantly improved by applying class decomposition to identify hidden and genuine subclasses within the symbols classes.

Applying CNNs produced comparable results with SVM and RF, despite the limited size of the dataset. However, CNNs and unlike SVM and RF performed better on the original dataset. This can be attributed to the limited size of the dataset. Part of the future work will include deploying methods such as data augmentation, which proved to be improving the performance of CNNs and then investigate the impact of decomposing the dataset on the performance of CNNs.

Acknowledgment

The authors would like to thank the Data Lab Innovation Centre in Scotland and DNV GL ltd for supporting this work.

References

[1] P. Vaxiviere and K. Tombre, "Celesstin: CAD Conversion of Mechanical Drawings," *IEEE Computer Magazine*, vol. 25, no. 7, pp. 46–54, 1992.

[2] H. Bunke, *Automatic Interpretation of Lines and Text in Circuit Diagrams*. Dordrecht: Springer Netherlands, 1982, pp. 297–310. [Online]. Available: http://dx.doi.org/10.1007/978-94-009-7772-3_18

[3] A. Okazaki, T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto, "Automatic Circuit Diagram Reader With Loop-Structure-Based Symbol Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 331–341, 1988.

[4] R. Kasturi, S. T. Bow, J. El-Masri, W.and Shah, and J. R. Gattiker, "A system for interpretation of line drawings," *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 978–992, 1990.

[5] S. H. Kim, J. W. Suh, and J. H. Kim, "Recognition of Logic Diagrams by Identifying Loops and Rectilinear Polylines," in *ProceedIngs of the Second International Conference on Document Analysis and Recognition - ICDAR'93*, 1993, pp. 349–352.

[6] J. F. Arias, C. P. Lai, S. Chandran, R. Kasturi, and A. Chhabra, "Interpretation of Telephone System Manhole Drawings," *Pattern Recognition Letters*, vol. 16, no. 4, pp. 365–368, 1995.

[7] Y. Yu, A. Samal, and S. C. Seth, "A system for recognizing a large class of engineering drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 868–890, 1997.

[8] S. V. Ablameyko and S. Uchida, "Recognition of engineering drawing entities: Review of approaches," *International Journal of Image and Graphics*, vol. 07, no. 04, pp. 709–733, 2007. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219467807002878

[9] M. K. Gellaboina and V. G. Venkoparao, "Graphic symbol recognition using auto associative neural network model," in *Proceedings of the 7th International Conference on Advances in Pattern Recognition, ICAPR 2009*, 2009, pp. 297–301.

[10] P. De, S. Mandal, and P. Bhowmick, "Identification of Annotations for Circuit Symbols in Electrical Diagrams of Document Images," *2014 Fifth International Conference on Signal and Image Processing*, pp. 297–302, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6754892/

[11] C. F. Moreno-García, E. Elyan, and 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.

[12] M. Furuta, N. Kase, and S. Emori, "Segmentation and recognition of symbols for handwritten piping and instrument diagram," 1984, pp. 626–629.

[13] M. Ishii, Y. Ito, M. Yamamoto, H. Harada, and M. Iwasaki, "An automatic recognition system for piping and instrument diagrams," *Systems and computers in Japan*, vol. 20, no. 3, pp. 32–46, 1989.

[14] C. Howie, J. Kunz, T. Binford, T. Chen, and K. H. Law, "Computer Interpretation of Process and Instrumentation Drawings," *Advances in Engineering Softwae*, vol. 29, no. 7-9, pp. 563–570, 1998.

[15] D. Blostein, "General Diagram-Recognition Methodologies," in *Proceedings of the 1st International Conference on Graphics Recognition (GREC'95)*, 1995, pp. 200–212.

[16] T. Kanungo, R. M. Haralick, and D. Dori, "Understanding Engineering Drawings: A Survey," in *Proceedings of the 1st International Conference on Graphics Recognition (GREC'95)*, 1995, pp. 119–130.

[17] C. R. Kulkarni and A. B. Barbadekar, "Text Detection and Recognition: A Review," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 6, pp. 179–185, 2017.

[18] Y. Lu, "Machine printed character segmentation - An overview," *Pattern Recognition*, vol. 28, no. 1, pp. 67–80, 1995.

[19] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

[20] A. K. Chhabra, "Graphics Recognition Algorithms and Systems," in *Proceedings of the 2nd International Conference on Graphics Recognition (GREC'97)*, 1997, pp. 244–252. [Online]. Available: http://www.springerlink.com/index/10.1007/3-540-64381-8

[21] L. P. Cordella and M. Vento, "Symbol recognition in documents: A collection of techniques?" *International Journal on Document Analysis and Recognition*, vol. 3, no. 2, pp. 73–88, 2000.

[22] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[23] J. Lladós, E. Valveny, G. Sánchez, and E. Martí, "Symbol recognition: Current advances and perspectives," in *International Workshop on Graphics Recognition*. Springer, 2001, pp. 104–128.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3065386

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[27] L. Fu and L. B. Kara, "From engineering diagrams to engineering models: Visual recognition and applications," *Computer-Aided Design*, vol. 43, no. 3, pp. 278 – 292, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010448510002447

[28] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[29] K. Tombre, S. Tabbone, B. Lamiroy, and P. Dosch, "Text/Graphics Separation Revisited," in *DAS*, vol. 2423, 2002, pp. 200–211.

[30] L. A. Fletcher and R. Kasturi, "Robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910–918, 1988.

[31] C. Tan and P. O. Ng, "Text extraction using pyramid," *Pattern Recognition*, vol. 31, no. 1, pp. 63–72, 1998. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0031320397000265

[32] R. Cao and C. L. Tan, "Text/graphics separation in maps," in *Selected Papers from the Fourth International Workshop on Graphics Recognition Algorithms and Applications*, ser. GREC '01. London, UK, UK: Springer-Verlag, 2002, pp. 167–177. [Online]. Available: http://dl.acm.org/citation.cfm?id=645439.652788

[33] Z. Lu, "Detection of text regions from digital engineering drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 431–439, 1998.

[34] E. Elyan and M. M. Gaber, "A genetic algorithm approach to optimising random forests applied to class engineered data," *Information Sciences*, vol. 384, no. Supplement C, pp. 220 – 234, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025516305783

[35] ——, "A fine-grained random forests using class decomposition: an application to medical diagnosis," *Neural Computing and Applications*, pp. 1–10.

[36] R. Vilalta, M.-K. Achari, and C. F. Eick, "Class decomposition via clustering: a new framework for low-variance classifiers," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 673–676.

[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[38] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[39] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014. [Online]. Available: http://jmlr.org/papers/v15/delgado14a.html

[40] Y. Zhang, *Support Vector Machine Classification Algorithm and Its Application*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 179–186. [Online]. Available: https://doi.org/10.1007/978-3-642-34041-3_27

[41] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119 – 130, 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0893608088900147

[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 EP –, 05 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[43] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors."

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.