

Reducing Human Effort in Engineering Drawing Validation

December 2019

Abstract

Oil & Gas facilities are extremely huge and have complex industrial structures that are documented using thousands of printed sheets. During the last years, it has been a tendency to migrate these paper sheets towards a digital environment, with the final end of regenerating the original computer-aided design (CAD) projects which are useful to visualise and analyse these facilities through diverse computer applications. Usually, this was done manually by re-sketching each page using CAD applications. Nevertheless, some applications have appeared which generate the CAD document automatically given the paper sheets. In this last case, the final document is always verified by an engineer due to the need of being a zero-error process. Since the need of an engineer is absolutely accepted, we present a new method to reduce the required engineer working time. This is done by highlighting the digitised components in the CAD document that the automatic method could have incorrectly identified. Thus, the engineer is required only to look at these components. The experimental section shows our method achieves a reduction of approximately 40% of the human effort keeping a zero-error process.

Keywords: Piping and Instrumentation Diagram (P&ID), automatic validation, digitisation, contextualisation, human validation.

1 Introduction

Piping and Instrumentation Diagrams (P&IDs) are commonly used for representing the structure and functionality of Oil & Gas facilities such as oil rigs and plants. Currently, these complex engineering drawings are generated by means of computer-aided design (CAD) tools but in the past, they were manually drawn on paper or by means of tools which are incompatible with modern software. Since these facilities are huge and composed of thousands of electric,

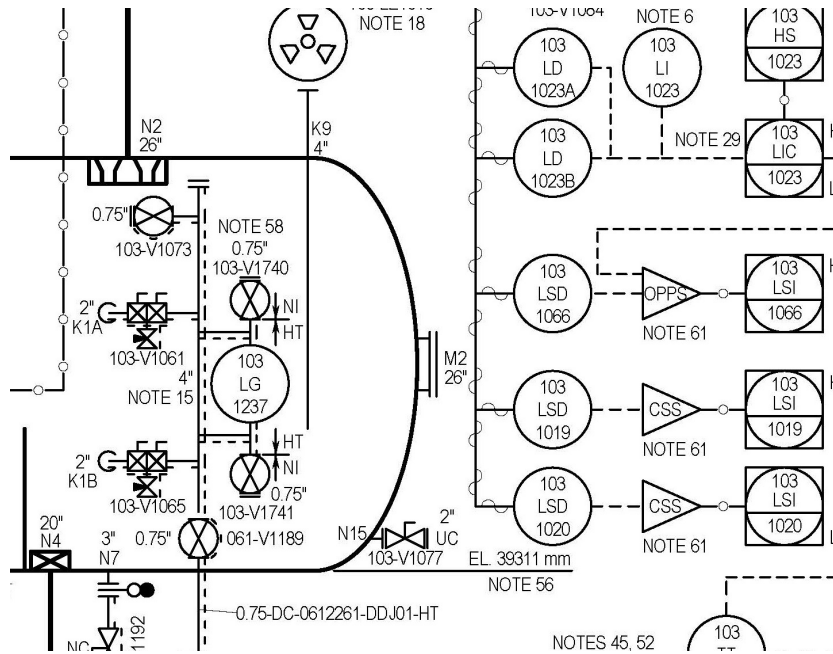


Figure 1: An example of a piece of a P&ID.

electronic or mechanical components connected by a vast network of pipelines, printed handbooks composed of thousands of pages are required to depict them. Figure 1 shows a portion of one of these sheets and illustrates the complexity of a P&ID.

In the last years, there has been an increased effort to join forces between the research community and industrial partners ¹ to collaborate in the development of methods for the migration of printed engineering drawings towards a digital environment. This has been tested in the past by numerous authors for different types of printed assets, such as mechanical drawings [32], electrical drawings [34], telephone manholes [1], sensor-equipment diagrams [20], P&IDs [15],[30],[22],[24],[25],[17], and even on maps [7] and musical scores [6].

Given that this process is extremely complex due to the quality of the scanned papers and the amount and variability of the involved electric, electronic and mechanic components, the possibility of symbol miss-identification during the digitisation or that some properties have not been correctly associated to certain components becomes high [2], [11]. Thus, it is expected that this process is not perfect and therefore, most systems enable human interaction to validate the symbol identification, connection and property association. Therefore, the final CAD document is always verified by an engineer due to the need of being a zero-

¹<https://cfmgcomputing.blogspot.com/2018/06/rgu-and-dnv-gl-join-forces-to-create.html>

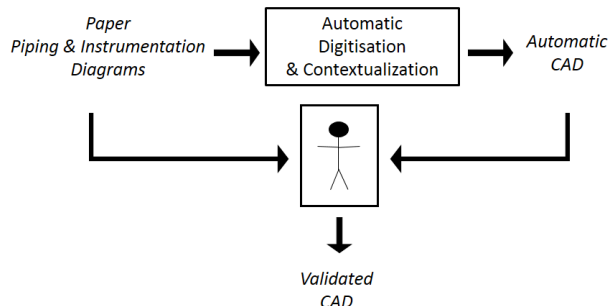


Figure 2: The process of deducing the CAD document, in which a human is involved to validate the data.

error process. Figure 2 shows a general flow diagram of the classical approach to extract a CAD document given a sheet of P&ID. The automatic module is composed of two steps: digitisation and contextualisation of the P&ID.

This paper proposes a method, summarised in Figure 3, to reduce the amount of effort the engineer needs to do to validate the CAD model keeping the zero-error aim. The main idea is that the engineer does not need to look at the whole diagram but only at the highlighted components, which are the ones that have a chance of having incorrectly identified by the automatic method.

The experiments have been carried out with industrial data and the method could be put into practise in other industrial cases. Nevertheless, to achieve the maximum performance, the training stage would have to be applied in each case.

The paper is structured as follows. Section 2 presents the current state of the art methodology used to transform P&IDs into digital assets. Section 3 examines the necessary concepts and presents the methodology aimed at aiding on the human validation of the digitised asset. Section 4 presents the experiments carried out to validate the proposed model. Finally, Section 5 is reserved for conclusions and future work.

2 Generating a CAD model given a P&ID

Generating a CAD model given a printed sheet of a P&ID is a migration process that implies two main steps: digitisation and contextualisation. In the first one, the main shapes of the P&ID are detected and localised. Basically, there are three types of figures in these drawings: symbols representing equipment and instrumentation, lines providing the connectivity between equipment and text

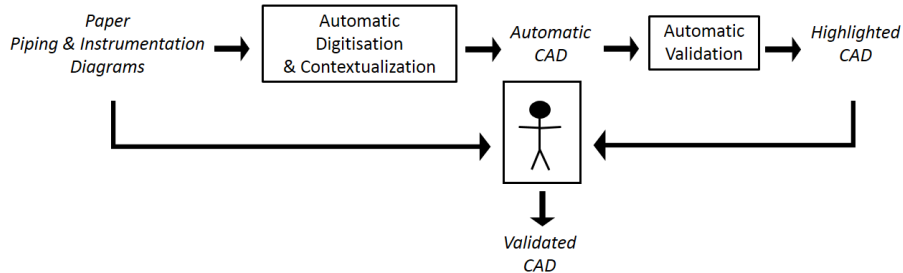


Figure 3: Our model for automatic detection of possible incorrectly identified components and final human validation.

describing the characteristics of equipment and connectors. Figure 1 shows a piece of a P&ID that contains these elements.

As mentioned in Section 1, numerous systems have been proposed in the literature for the digitisation of P&IDs. Most of these systems rely in heuristic-based computer vision methods to recognise the shapes [16], [30], [2], [22], [17], although most recent literature has preferred the use of deep learning based technologies to localise and extract the figures from the engineering drawing [13], [24]. Regardless of the approach, the outcome of a digitisation system is usually a list of components along with some characteristics (such as tag id, thickness or location within the drawing or the facility) and a list of pipelines (i.e. connectors between components). Notice that in the case of P&IDs, it may be the case that pipelines also have some associated properties, such as material, thickness, composition, etc. These properties are usually indicated by text which lies adjacent to the line or that is connected to the pipeline through a lead line.

Although most of the times digitisation is done as a standalone process where shapes are recognised either by heuristic or deep learning based methodologies, some previous work has been benefited from understanding *a priori* the standards used to produce these drawings to obtain all shapes with improved accuracy. For instance, Moreno-Garcia [22] presented a comparative study of text/graphics separation algorithms [31] applied on P&IDs, where parametrical symbols (i.e. the ones described with circles and lines) were found first by means of state-of-the-art methods such as Hough circles and transform [3]. Based on the previous knowledge, the text within the shapes is located and characterised easily. Thus, some text characters can be detected along with their properties (i.e. width, height, pixel density, stroke, amongst others). This allows an easier detection of the remaining text throughout the drawing.

Once the P&ID has been digitised, a *netlist* which contains the position of each

element within the drawing is produced. Nonetheless, a *netlist* is not directly usable for an expert, as it only contains the count of the shapes found, and it does not reflect the topology of the schema. Some online tools have been presented to do quick visualisations of the extracted *netlists*, such as *NetVis* or *Netlist2CAD*², however none of these tools offer a true connectivity comprehension of the extracted information. Still, these come handy for a quick human inspection of the extracted data and to manipulate/correct some shapes.

In the second step to generate the CAD model, the shapes in the netlist are contextualised. Contextualisation means making sense out of the digitised data. Thus, the CAD model topology describes the connectivity and the relation between the components. The output can be a standardised file [16], [2] or a graph [25]. These files are naturally imported by CAD applications. For a more in-depth analysis of the challenges of digitising and contextualising these drawings, the reader is referred to the reviews presented in [23], [21].

Regarding actual contextualisation, the use of graph-based or tree-based structures to represent and store P&IDs has become widespread in recent years [15, 24, 25, 17], however no attempts have been done at analysing the topology of the diagram or exploiting the topological information contained on the key drawings (i.e. the sheet containing the legend of each symbol and the most common configurations, an example can be found in³) to improve or automatically correct the digitisation of P&IDs.

The methods presented until now perform the digitisation and contextualisation in a sequence of two steps. Thus, the topology of the P&ID is deduced from the digitisation but the identity of the components is never reviewed given the deduced topology of the P&ID. Our method reviews the identity of the components by comparing the deduced topology to the most probable one. Then, it presents to the human the ones that differ from the initial option. Being the most probable identity is deduced through a machine learning process.

3 Automatic P&ID validation

The difference between the classical models (Figure 2) and our model (Figure 3) is the incorporation of the Automatic Validation module. The aim of this module is to deduce the identity of the components in the *Automatic CAD* and highlight the components that must be reviewed by an human expert, reducing in this way, the number of components that should be reviewed when any CAD document is generated by the Automatic Digitisation module. In Figure 4, we show a scheme of this proposal. In the next two sections, we detail the two main modules of this scheme: the Graph representation and Data embedding module and the Machine learning module.

²<http://cfmgcomputing.blogspot.com/p/software-demos.html>

³<https://www.edrawsoft.com/pid-legend.php>

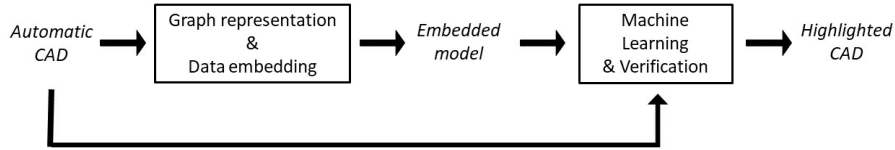


Figure 4: Automatic P&ID validation module incorporated in Figure 3.

3.1 Graph representation and data embedding

The automatic validation method that we present is based on defining P&IDs as attributed graphs. Attributed graphs have been of crucial importance in pattern recognition throughout more than four decades, [4, 26, 27], since they have been used to model several kinds of problems. Interesting reviews of techniques and applications are [8, 33, 18, 12]. If elements in pattern recognition are modelled through attributed graphs, error-tolerant graph-matching algorithms are needed that aim to compute a mapping between nodes of two attributed graphs that minimises some kind of objective function [29, 28].

In our graph, nodes represent components and edges represent pipelines that connect these components. Moreover, nodes have only one attribute, which is the component identity (valve, compressor,...) and edges are unattributed and undirected. In an attributed graph, a star is defined as a local structure composed of a node, its connected edges and also the nodes that these edges connect. Note these other nodes are usually called neighbour nodes of the central node. Our goal is to deduce the identity of each component (node in the graph) given its set of pipelines connected to it and their components that connect these pipelines (edges connected to it and neighbour nodes). For this reason we use the star, since by definition, this sub-structure contains this information.

Graphs have some limitations when they are applied to machine learning due to their intrinsic relational representation. This is because some trivial mathematical operations used in the traditional numeric machine learning representations have not an equivalence in the graph domain. Given an arbitrary set of graphs, a possible way to address this problem is to define an embedding function from the graph domain to a vector space [14]. Broadly speaking, an embedding function converts an attributed graph into a vector. However, defining such embedding function is extremely challenging, when the constraints on time efficiency and preserving the underlying structural information is concerned.

Explicit graph embedding is based on defining a function that, given a graph, generates a point in an Euclidean space. These embedding functions are divided into four classes: 1) Graph probing [19] that measures the frequency of specific substructures. 2) Spectral graph theory [5], which analyses the structural properties of graphs in terms of eigenvectors and eigenvalues. 3) Dissimilarity

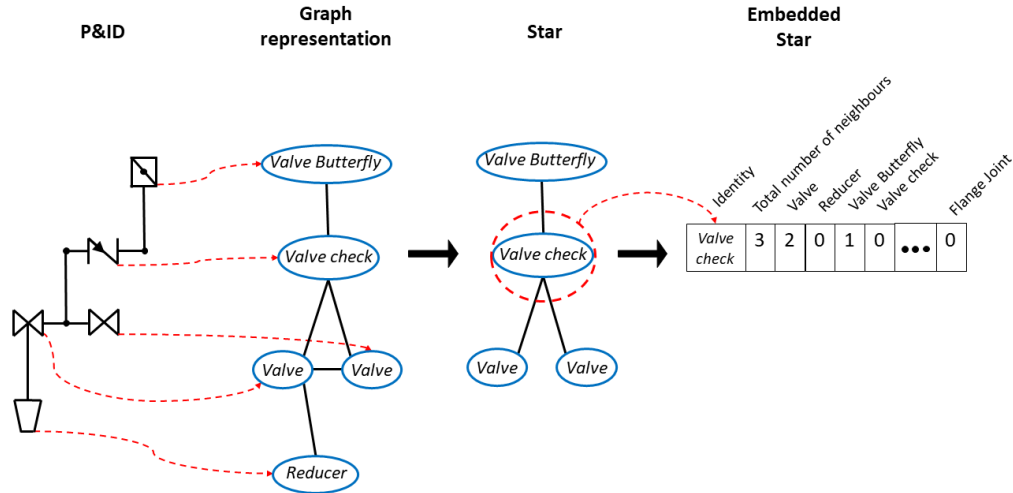


Figure 5: An example of embedding a *Valve check* star into a vector.

measurements [10], in which the function depends on its distance to a selected set of graphs. 4) Geometric deep learning [9], in which the embedding uses deep neural networks.

Since we want to use classical machine learning techniques to deduce component identities, we embed stars into vectors based on a graph probing method [19]. We selected this method since the structural and semantic information contained in a star is limited and the computational cost of embedding a star is only linear with respect to the number of outgoing edges per node. Thus, each star is embedded in a Euclidean space R^{n+2} , where n is the number of different component identities. The embedding of the i^{th} node in the graph (or the i^{th} component in the P&ID) is defined as a vector $E_i = (c_i, d_i, f_i^1, \dots, f_i^n)$ where c_i is the identity of the central node of the star; d_i is the number of edges in the star (or the number of connected pipelines to the central component); and f_i^p is the number of external nodes of the star that have the p identity, with $p = 1, \dots, n$ (or the number of components that have the p identity and that are connected by a pipeline to the i^{th} component). A sample of a star embedding is presented in Figure 5. The output of this module is the embedded model composed of all embedded stars in the graph representation of the CAD model.

3.2 Machine learning and verification

The Machine learning and verification module performs the following tasks:

- First, each component in the P&ID represented by an embedded vector is introduced into the machine learning algorithm that returns the predicted component identity.
- Second, the identities of the components returned by the machine learning algorithm are contrasted with identities obtained from the digitised and contextualized *netlist* of the *Automatic CAD*. Note these components have not been verified by the engineer. Thus, this task discerns whether the deduced identities by our machine learning algorithm are the same or they are different from the CAD model.
- Third, it detects the components of the *Automatic CAD* whose identities are different from the identities obtained by the machine learning algorithm. These detected components are highlighted to be validated by the human expert.

The learning set is composed of a CAD model, validated by a human expert, which has been embedded using our graph representation and data embedding module (Section 3.1). This CAD model must include a representative number of components per identity in order to assure the proper learning of the data. Usually, the larger the learning set is, the better the prediction given by the machine learning algorithm.

3.3 Theoretical analysis of the validation method

The simplest metric to evaluate the quality of an identification method is by the percentage of correctly identified elements (called True Prediction) and the percentage of incorrectly identified elements (called False Prediction). To do so, it is necessary to have *a priori* knowledge of what is the correct identity of a set of elements. This information is usually called the ground truth set. Thus, it is assumed that any method correctly identifies an element if the predicted identity is exactly the same as the ground truth and makes an incorrect identification if the predicted identity is different from the ground truth.

In our case, the ground truth set is given by a human expert, who has thoroughly checked the paper P&ID. Nevertheless, what makes our method stand out is that we not only have the predicted identity (represented in *Highlighted CAD*) and the ground-truth identity, but also the component identity deduced by the Automatic Digitisation and Contextualisation module (represented in the *Automatic CAD*). Figure 3 best illustrated the process that generates these three CAD documents.

Table 1 shows the different cases that appears in our method. The true predictions and false predictions generated by our machine learning algorithm are split in two scenarios (the two columns in the table) considering whether the prediction of our method is the same as or different from the prediction of the *Automatic CAD*. This is because, in the first case, the human is not asked to

validate the components’ identity but in the second column one does. In that case, components are highlighted in the Highlighted CAD document.

From this model, we obtain the following four cases:

- Case *A*: Both methods make a correct prediction. The human would not have to validate the components’ identity and does not validate it.
- Case *B*: Both methods fail and deduce the same incorrect identity. The human would have to validate the components’ identity but does not validate it.
- Case *C*: Our method makes a correct identity prediction but the Automatic method does not properly predict it. There is no need of validation, although the human does it. In this case, the fact that the automatic method fails at predicting the identity makes the human effort to increase. Nevertheless, we cannot influence on the number of elements in *C* since it depends on the *Automatic CAD*, which is the input data of our machine learning algorithm.
- Case *D*: Our method fails at predicting the identity and it is different from the Automatic method. Validation is required to correct the prediction, and the human executes this action.

Considering the special feature of our method, which deals with three different types of data, we propose two metrics to validate our method instead of the classical *True Prediction* or *False Prediction*. These are:

- **Human effort**: The percentage of components that the human has to validate since the automatic method and our method deduce different identities. It is composed of the sum of cells *C* and *D*. Clearly, we wish this metric to be the lowest as possible to reduce the economical and temporal impact of checking the incorrectly identified elements by the automatic method, although we cannot influence on *C* as commented above.
- **Validation error**: The percentage of components that the human does not validate and would have to be validated. This value is shown in cell *B*. If we want the method to be error free, this value has to be strictly zero.

Highlighted CAD		
	= Automatic CAD	≠ Automatic CAD
True Prediction	A	C
False Prediction	B	D
	↓	↓
	Human does not validate	Human does validate

Table 1: True and false predictions obtained by our method.

	Sheet 1	Sheet 2	Sheet 3	Sheet 4	Total
Number of components	125	107	177	115	524
Number of identities	22	10	24	18	38

Table 2: Number of components and identities in the four sheets that compose the learning set.

Identity	Number of repetitions
Valve Ball	108
Reducer	77
Continuity Label	68
Flange Joint	41
Arrowhead	35
DB&BPV	23
DB&BBV	19

Table 3: Most common identities of components in the ground truth set.

4 Experiments and discussions

The experiments have been carried out with industrial data. A batch of P&IDs was obtained from an existing collaboration with an industrial partner by means of a project funded by Scottish Innovation Centres. Nevertheless, all data have been anonymised to fulfil the company requirements.

4.1 Experimental setup

The experimental validation was carried out as follows. First, we used four P&ID sheets from the same standard. Figure 1 shows a piece of one of these sheets. From each sheet, we used the *Automatic CAD*, in which components and pipelines have been automatically deduced by the digitisation and contextualisation process defined in [23], and also the *Validated CAD*, in which a specialist on these type of sheets has checked and corrected the resulting model. (see Figure 3). The test set is composed of the four *Automatic CAD* models and the learning set is composed of the four *Validated CAD* models. Table 2 shows the number of components and different component identities in each P&ID sheet in the learning set. The total number of components and identities appear in the last column. Note that not all the component identities appear in the four sheets.

Table 3 shows the seventh most common component identities with their number of repetitions that appear in the four sheets of the learning set.

The Graph representation & Data embedding module (Figure 4 and Section 3.1) returns a vector of 40 elements. The first two positions in the vector are

the component identity and the number of pipelines connected to the component. The other 38 positions are the number of appearances of the 38 possible identities (last column in Table 2).

The machine learning module (Figure 4 and Section 3.2) has been implemented by a neural network composed of three layers (input, hidden and output layer). The number of neurons in the input, hidden and output layers are 40, 39 and 38, respectively. It is a feed-forward network, with sigmoid hidden and softmax output neurons. The neural network architecture and training process has been carried out through the "*Neural Network Matlab tool*". We have selected the scaled conjugate gradient back propagation algorithm to train the neural network.

The aim of this paper is not to deeply analyse the applicability of several machine learning methods but to present a method to reduce the human effort for validating the CAD model. For this reason, we have selected the well known neural network architecture and an easy to be used tool, which has been proven to return good enough classification results in several applications.

4.2 Experimental results

Table 4 shows the true and false predictions per each sheet in the test set. Considering the sum of the metrics of the four sheets, we observe, there are 199 components that do not need to be validated by the human expert because both predictions are equal and both are correct. There are six cases in which both predictions do not match, and our method is correct but the automatic method does not. On the other hand, there are 319 cases in which the two predictions are different and our method predicts an incorrect identity. In these cases, some of them have been correctly predicted by the automatic method, but others do not. The human expert has to check the correct identity of $6 + 319$ components. We observe that there are not cases in which both methods deduced the same incorrect identity and, consequently, all errors are detected and we achieve the aim of zero-error process.

Table 5 shows the two metrics presented in Section 3: *Human effort* and *Validation error*. It turns out that, thanks to our method, in the worst of the cases (Sheet 4), the human expert needs to check the 69.57% of the components and in the best case (Sheet 2), just the 38.32%. The average human effort is the 60.66% of the effort without applying our method, this means that there is a human effort reduction of the 39.34%, which is the main aim of our method. Also, it is important to note that in all the sheets, the zero-error process is achieved because all the incorrectly identified components have been detected (Validation error = 0).

We realise that there is a human effort reduction keeping an error-free process and therefore, our aim has been achieved. Nevertheless, there is a gap to improve these results. We believe the reduction of the human effort could be achieved

		Highlighted CAD	
		= Automatic CAD	≠ Automatic CAD
Sheet 1	True Prediction	42	2
	False Prediction	0	81
Sheet 2	True Prediction	66	1
	False Prediction	0	40
Sheet 3	True Prediction	56	3
	False Prediction	0	118
Sheet 4	True Prediction	35	0
	False Prediction	0	80
Total	True Prediction	199	6
	False Prediction	0	319

Table 4: Evaluation of the proposed method.

Sheet	Human effort	Validation error
1	66.40 %	0 %
2	38.32 %	0 %
3	68.36 %	0 %
4	69.57 %	0 %
Average	60.66 %	0 %

Table 5: Human effort and Validation error of the four sheets and their average.

by increasing the size of the learning set. The learning set is composed of 524 components, which have 38 different identities (Table 2). Note there are 108 *ValveBall* components (Table 3), which is one fifth of the learning set. Moreover, there are 31 component identities only represented by 153 components. This means we have a tiny and highly unbalanced learning set that hinders learning the data.

5 Conclusions and future work

In this paper, we present a method to reduce the human effort while validating CAD documents that have been automatically generated from a class of complex engineering drawings called P&IDs. The experimental validation shows that we achieve an average reduction of approximately the 40% of the human effort, while keeping an error-free process.

Our method is based on detecting the identity of the components through learning their topology. To do so, we have represented the P&IDs by attributed graphs and we have embedded the local structure of components into vectors. Given each vector, a neural network has been used to predict the identity of the component represented by this vector. Only the ones that our predicted

identity is different from the automatically deduced identity are presented to the human expert.

As a future work, we want to analyse the impact of different configurations of the neural network or to analyse if a deep neural network could be used. In this last case, we need to drastically increase the size of the learning set. Moreover, we also want to move our system from the laboratory to the industry, thus being in use in the digitisation process of P&ID sheets. This method could be applied to other kind of industries in which the relational information between the components is available on the P&IDs. We believe our method could drastically reduce the human effort and therefore the economical and temporal cost of this essential task.

Acknowledgements

This research is supported by Spanish projects TIN2016-77836-C2-1-R and DPI2016-78957-R, by the Data Lab and the Oil & Gas Scottish Innovation Centres, and by Det Norske Veritas Germanischer Lloyd (DNV GL).

References

- [1] Juan F. Arias et al. “Interpretation of Telephone System Manhole Drawings”. In: *Pattern Recognition Letters* 16.4 (1995), pp. 365–368.
- [2] Esteban Arroyo et al. “Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams”. In: *Computers and Chemical Engineering* 92 (2016), pp. 112–132. ISSN: 00981354. DOI: 10.1016/j.compchemeng.2016.04.040. URL: <http://dx.doi.org/10.1016/j.compchemeng.2016.04.040>.
- [3] D. H. Ballard. “Generalizing the Hough transform to detect arbitrary shapes”. In: *Pattern Recognition* 13.2 (1981), pp. 111–122. ISSN: 00313203. DOI: 10.1016/0031-3203(81)90009-1.
- [4] H Bunke and G Allermann. “Inexact Graph Matching for Structural Pattern Recognition”. In: *Pattern Recogn. Lett.* 1.4 (May 1983), pp. 245–253. ISSN: 0167-8655. DOI: 10.1016/0167-8655(83)90033-8. URL: [http://dx.doi.org/10.1016/0167-8655\(83\)90033-8](http://dx.doi.org/10.1016/0167-8655(83)90033-8).
- [5] Terry Caelli and Serhiy Kosinov. “An Eigenspace Projection Clustering Method for Inexact Graph Matching”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 26.4 (2004), pp. 515–519.
- [6] Jorge Calvo-Zaragoza et al. “Deep Neural Networks for Document Processing of Music Score Images”. In: *Applied Sciences* 8.5 (2018), p. 654. ISSN: 2076-3417. DOI: 10.3390/app8050654. URL: <http://www.mdpi.com/2076-3417/8/5/654>.

- [7] Ruini Cao and Chew Lim Tan. “Text/Graphics Separation in Maps”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. 2002, pp. 167–177. ISBN: 0769527701.
- [8] Donatello Conte et al. “Thirty years of graph matching in pattern recognition”. In: *International journal of pattern recognition and artificial intelligence* 18.03 (2004), pp. 265–298.
- [9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *NIPS*. 2016, pp. 3837–3845.
- [10] Robert P. W. Duin and Elzbieta Pekalska. “The Dissimilarity Representation for Structural Pattern Recognition”. In: *CIARP*. Vol. 7042. Lecture Notes in Computer Science. Springer, 2011, pp. 1–24.
- [11] Eyad Elyan, Carlos Francisco Moreno-García, and Chrisina Jayne. “Symbols classification in engineering drawings”. In: *International Joint Conference on Neural Networks (IJCNN)*. 2018. DOI: 10.1109/IJCNN.2018.8489087.
- [12] Pasquale Foggia, Gennaro Percannella, and Mario Vento. “Graph matching and learning in pattern recognition in the last 10 years”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 28.01 (2014), p. 1450001.
- [13] Mahesh Kumar Gellaboina and Vijendran 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. ISBN: 9780769535203. DOI: 10.1109/ICAPR.2009.45.
- [14] Jaume Gibert, Ernest Valveny, and Horst Bunke. “Graph embedding in vector spaces by node attribute statistics”. In: *Pattern Recognition* 45.9 (2012), pp. 3072–3083.
- [15] C. Howie et al. “Computer Interpretation of Process and Instrumentation Drawings”. In: *Advances in Engineering Software* 29.7-9 (1998), pp. 563–570. ISSN: 09659978. DOI: 10.1016/S0965-9978(98)00022-2.
- [16] C. Howie et al. *Computer Interpretation of Process and Instrumentation Drawings (Progress Report)*. Tech. rep. 1995, p. 18.
- [17] Sung-O Kang, Eul-Bun Lee, and Hum-Kyung Baek. “A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (P&ID)”. In: *Energies* 12.2593 (2019), pp. 1–26. DOI: 10.3390/en12132593.
- [18] Lorenzo Livi and Antonello Rizzi. “The graph matching problem”. In: *Pattern Analysis and Applications* 16.3 (2013), pp. 253–283.
- [19] Muhammad Muzzamil Luqman et al. “Fuzzy Multilevel Graph Embedding”. In: *Pattern Recogn.* 46.2 (Feb. 2013), pp. 551–565. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2012.07.029. URL: <http://dx.doi.org/10.1016/j.patcog.2012.07.029>.
- [20] Carlos Francisco Moreno-García. “Digital interpretation of sensor-equipment diagrams”. In: *Proceedings of the SICSA Workshop on Reasoning, Learning and Explainability (ReaLX 2018)*. 2018.

- [21] Carlos Francisco Moreno-García and Eyad Elyan. “Digitisation of Assets from the Oil & Gas Industry: Challenges and Opportunities”. In: *International Conference on Document Analysis and Recognition (ICDAR)*. Workshop on Industrial Applications of Document Analysis and Recognition (WIADAR). 2019, pp. 16–19. ISBN: 9781728150543. DOI: 10.1109/ICDARW.2019.60122.
- [22] Carlos Francisco Moreno-García, Eyad Elyan, and Chrisina 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.
- [23] Carlos Francisco Moreno-García, Eyad Elyan, and Chrisina Jayne. “New trends on digitisation of complex engineering drawings”. In: *Neural Computing and Applications* 31.6 (2019), pp. 1695–1712. DOI: 10.1007/s00521-018-3583-1.
- [24] Rohit Rahul et al. “Automatic Information Extraction from Piping and Instrumentation Diagrams”. In: *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*. 2019, pp. 163–172. DOI: 10.5220/0007376401630172. arXiv: 1901.11383. URL: <http://arxiv.org/abs/1901.11383>.
- [25] Miia Rantala et al. “Applying graph matching techniques to enhance reuse of plant design information”. In: *Computers in Industry* 107 (2019), pp. 81–98. ISSN: 01663615. DOI: 10.1016/j.compind.2019.01.005. URL: <https://doi.org/10.1016/j.compind.2019.01.005>.
- [26] A Sanfeliu and King-Sun Fu. “A distance measure between attributed relational graphs for pattern recognition”. In: *Systems, Man and Cybernetics, IEEE Transactions on SMC-13* (June 1983). DOI: 10.1109/TSMC.1983.6313167.
- [27] Alberto Sanfeliu et al. “Graph-based representations and techniques for image processing and image analysis”. In: *Pattern Recognition* 35 (2002), pp. 639–650.
- [28] Francesc Serratosa and Xavier Cortés. “Graph edit distance: Moving from global to local structure to solve the graph-matching problem”. In: *Pattern Recognition Letters* 65 (2015), pp. 204–210.
- [29] Albert Solé-Ribalta, Francesc Serratosa, and Alberto Sanfeliu. “On the Graph Edit Distance Cost: Properties and Applications”. In: *IJPRAI* 26.5 (2012).
- [30] Wei Chian Tan, I. Ming Chen, and Hoon Kiang Tan. “Automated identification of components in raster piping and instrumentation diagram with minimal pre-processing”. In: *IEEE International Conference on Automation Science and Engineering* November (2016), pp. 1301–1306. ISSN: 21618089. DOI: 10.1109/COASE.2016.7743558.
- [31] Karl Tombre et al. “Text/Graphics Separation Revisited”. In: *Document Analysis Systems*. Vol. 2423. 2002, pp. 200–211.
- [32] Pascal Vaxiviere and Karl Tombre. “Cesstin: CAD Conversion of Mechanical Drawings”. In: *IEEE Computer Magazine* 25.7 (1992), pp. 46–54.

- [33] Mario Vento. “A long trip in the charming world of graphs for pattern recognition”. In: *Pattern Recognition* 48.2 (2015), pp. 291–301.
- [34] Yuhong Yu, Ashok Samal, and Sharad C. Seth. “A system for recognizing a large class of engineering drawings”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.8 (1997), pp. 868–890. ISSN: 01628828. DOI: 10.1109/34.608290.