Pixel-based layer segmentation of complex engineering drawings using convolutional neural networks

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

Pam Johnston
School of Computing Science
and Digital Media
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: p.johnston2@rgu.ac.uk

Bello Garkuwa
School of Computing Science
and Digital Media
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: b.garkuwa@rgu.ac.uk

***Abstract*—One of the key features of most document image digitisation systems is the capability of discerning between the main components of the printed representation at hand. In the case of engineering drawings, such as circuit diagrams, telephone exchanges or process diagrams, the three main shapes to be localised are the symbols, text and connectors. While most of the state of the art devotes to top-down recognition approaches which attempt to recognise these shapes based on their features and attributes, less work has been devoted to localising the actual pixels that constitute each shape, mostly because of the difficulty in obtaining a reliable source of training samples to classify each pixel individually. In this work, we present a convolutional neural network (CNN) capable of classifying each pixel, using a type of complex engineering drawings known as Piping and Instrumentation Diagram (P&ID) as a case study. To obtain the training patches, we have used a semi-automated heuristics-based tool which is capable of accurately detecting and producing the symbol, text and connector layers of a particular P&ID standard in a considerable amount of time (given the need of human interaction). Experimental validation shows that the CNN is capable of obtaining these three layers in a reduced time, with the pixel window size used to generate the training samples having a strong influence on the recognition rate achieved for the different shapes. Furthermore, we compare the average run time that both the heuristics-tool and the CNN need in order to produce the three layers for a single diagram, indicating future directions to increase accuracy for the CNN without compromising the speed.**

Keywords: Convolutional Neural Networks, Piping and Instrumentation Diagram, Pixel Classification, Engineering Drawing Digitisation.

I. Introduction

Digitisation of engineering drawings has been a persistent problem in different industries such as the electrical, chemical and the Oil & Gas sector. In the case of the latter, experts have expressed an urgent need to migrate legacy printed representations used in this industry towards a paperless environment, which can lead to improved data storage, data mining and security assessment. In particular, a class of engineering drawings known as a Piping and Instrumentation Diagram (P&ID) has been identified as a complex case of study which requires a combination of techniques to properly digitise and contextualise the obtained information. An extract from a P&ID is shown in Figure 1, but these drawings can be large and complex. They are also very common in industry so an efficient method of digitisation is important. Notice that this type of engineering drawing conveys a complex and entangled structure of symbols, text and connectors which are hard to understand even for the human eye. Efficiently untangling these symbols, text and connectors is a novel problem whose solution may lead to improvements in automated data extraction from P&ID drawings.

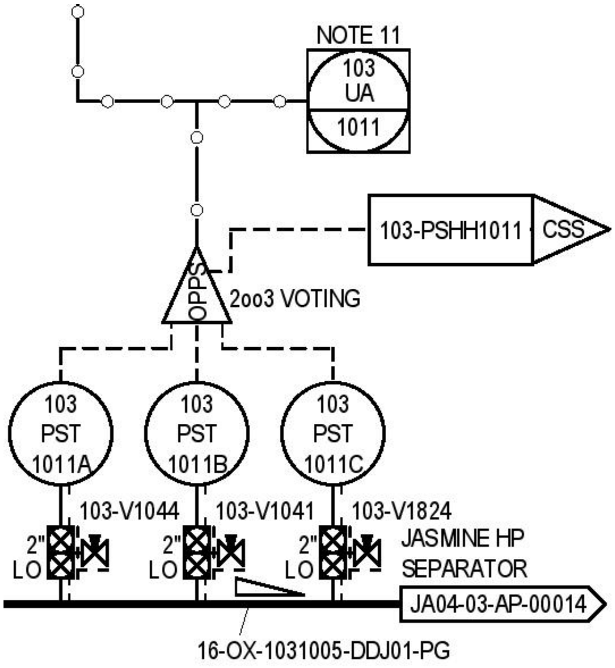

Fig. 1. Example of a P&ID.

Research into the digitisation and contextualisation of

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

P&IDs is sparse in comparison to other image processing advancements. For instance, Furuta et al. [1] and Ishii et al. [2] developed systems aimed at the recognition of symbols and connecting lines in handwritten and CAD-based P&IDs. However, these papers were published more than 30 years ago, and thus the techniques that they present may seem outdated. Nonetheless, it is possible to see from this work how P&IDs have derived into the representations that we currently aim to digitise. In the mid-90's, Howie et al. [3] presented a comprehensive technical report on a system to detect and classify symbols based on a repository of symbols represented by using graphs. This approach has also been used by other authors such as Jiang et al. [4] with the aim of finding the median symbol from a collection of samples. More recent work in P&ID data extraction has been presented by Tan et al. [5], where authors proposed using local binary patterns and a sliding window technique to identify candidate symbols and verify them against a repository, thus classifying the obtained samples. In the same domain, Moreno-Garcia et al. [6] presented a study on how the implementation of text-graphics separation [7], [8] can enhance symbol recognition in P&IDs. Separating text from graphics allows for the possibility of leveraging advanced, existing techniques specific to each domain. Although techniques for text recognition are already well developed and widely applied, P&ID symbol recognition is still a relatively new field. The interested reader is referred to the work presented by Moreno-Garcia et al. [9], which encompasses a comprehensive literature review on former and recent trends for engineering drawing digitisation and contextualisation, with particular focus on the case on P&IDs. Using the techniques in [6] as a stepping stone, Elyan et al. [10] collected a symbol repository and performed classification tests considering the imbalance on the dataset [11], [12] by means of class decomposition [13], [14].

In parallel with the developments on digitisation and contextualisation of engineering drawings, there has been a notable increase in the use of deep learning techniques in document image analysis. In particular, the concept of the convolutional neural network (CNN), has been used to solve a wide range of image recognition problems [9]. In the specific case of CNN application to P&IDs, the literature is sparse. One of the few examples is a system [15] which uses a CNN architecture to perform symbol recognition for a fixed collection of symbols with a particular pattern. Although recent, this work pre-dates many of the modern advances in deep convolutional neural networks and uses a small dataset by today's standards. In addition to this example of CNN use applied to P&IDs, there is the aforementioned work presented by Elyan et al. [10], where a collection of symbols in P&IDs were extracted and classified using Support Vector Machine, Random Forests and a CNN.

In this paper, we propose the implementation of a CNN, which uses pixel patches as training data, instead of particular features or symbol images. This way, our network is capable of classifying all pixels in the engineering drawing as symbol, text or connector pixels, thus splitting the input image into its three main layers. To generate the layers to obtain the training pixel patches, we have used a heuristics-based tool [6], [10] which has been implemented for a particular P&ID standard. This tool is further detailed later in this paper. A diagram of the proposed framework is shown in Figure 2.

The rest of this paper is organised as follows: Section II discusses previous methods which have presented image processing solutions for a range of different document image domains which have been based on pixels as the main feature. Section III presents the methodology and the CNN architecture used, while section IV presents the dataset and the experimental framework. Finally, Section V is reserved for conclusions and future work.

II. Related Work

The idea of implementing pixel-based classification for the localisation of elements in printed documents is not new. In fact, it can be traced back to 1978 [16], where it was defined that the image segmentation problem could be refactored as a pixel classification one. Therefore, authors presented a method where the features to be used are the gray pixels, while the edge value is equal to the magnitude of an approximation to the gray level gradient at each point. This method was applied on Forward-Looking Infra Red (FLIR) images, where thermography is used to identify objects in the dark. The purpose of this method was to perform image thresholding [17].

In more recent work, it is possible to find pixel-based segmentation applications, mostly outside the domain of engineering drawings. Cote et al. [18] presented a model for classifying pixels in business document images. The authors classified pixels into four layers of interest (namely text, image, graphics and background) using low-dimensional feature descriptors based on textural properties, aided by a Support Vector Machine (SVM) classifier. Authors proposed to represent each pixel as a feature vector based upon the image response to a filter bank. To obtain the training data and the ground truth, they used a commercial software called Aletheia 1.5, which allowed them to select regions of more than a thousand business document pages and store such selections in XML format.

Other efforts have been done to produce better region of interest segmentation based on CNN architectures in the domain of photographic images. Pinehiro et al. [19] presented a pixel-level CNN capable of segmenting objects of interest for training purposes using weakly annotated images based on Multiple Instance Learning. This approach relies on an aggregation layer which works with the features computed by the CNN. As a result, this CNN is capable not only of classifying each image, but also of detecting which pixels comprise the annotated object.

To enhance the use of CNNs to train recognition methods for hyper-spectral images, Li et al. [20] presented the concept of pixel-pair features. The idea is to pair training samples using as criterion the identification of the change of label. By designing the CNN to take this into account, plus the implementation

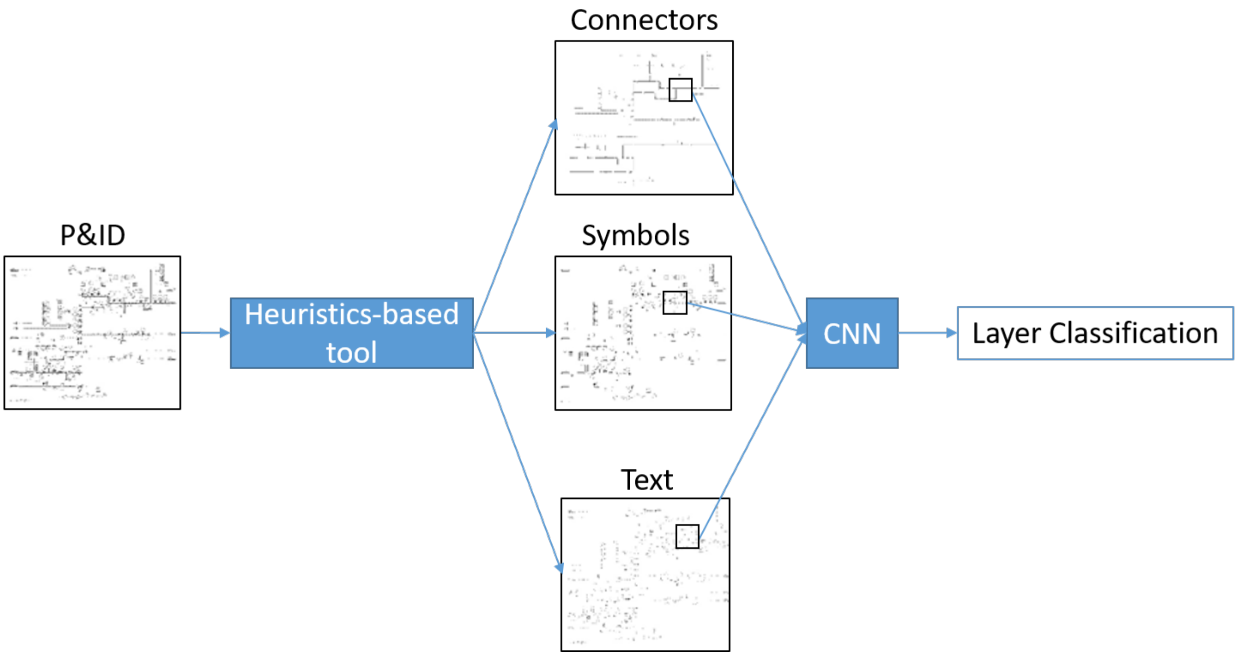

Fig. 2. Schematic illustration of the proposed method.

of a voting strategy for the joint classification, accuracy can be increased when attempting to classify neighbouring pixels in heterogeneous regions.

Most recently, Calvo-Zaragoza et al. [21] presented a deep learning methodology to classify pixels in medieval musical score images with poor quality. For the case at hand, the purpose of the method was to find the background, text, score and staff pixels, using manually-labelled pixel patches of different sizes to train CNNs with variable parameters, such as convolutional blocks and number of layers. Experimental validation showed that different training patch sizes yielded different accuracy results, achieving more success with rectangular rather than squared patches. This is mostly due to staff lines (horizontal lines) being recognised better as the network learns from such shape. In the case of engineering drawings, the same phenomenon could occur with connectors, although connecting lines are both horizontal and vertical in this case. Our experiments are designed to examine this.

III. Methods

A. Heuristics-based Tool for Layer Generation

To split the original image into the three layers of interest, we have used a semi-automated heuristics-based tool which has been previously presented in [6], [10]. This tool is capable of locating the elemental shapes of a P&ID (i.e. continuity labels, sensors, text, dashed connectors, pipeline connectors and equipment symbols) by sequentially locating and segmenting these shapes. Therefore, we have created three layers by combining the elements as follows:

- Symbols (composed by continuity labels, sensors and equipment symbols)
- Connectors (composed by dashed and pipeline connectors)
- Text

Figure 3 shows the symbol (top-left), connector (top-right) and text (centre) layers for a portion of a P&ID. Notice that since we have also considered dashed connectors in the connector layer; these more closely resemble text characters and thus, are harder to detect. The heuristics-based tool can only identify objects that it has been explicitly programmed to detect. The process with the heuristics tool is semi-automated, with additional human annotation used to produce an accurate ground truth. The classifier that we train using this labelled data will be capable of classifying any pixel within a P&ID image as *symbol*, *connector* or *text*, including objects that it has not necessarily been trained for, without any human intervention.

B. Convolutional Neural Network Framework

Once the heuristics-based tool has produced the three layers for a given image, it is possible to obtain samples for the CNN by applying a sliding window on each layer, centring such window on a pixel of interest. As a result, the input for the CNN is a rectangular patch. Figure 4 shows samples for a symbol, text and connector patch respectively. Notice that since this is a pair-wise classification scenario, the pixel patches can also be obtained from the original image by implementing a sliding window approach switch centres in a specific pixel type and acquires the pixel patch. We have decided to use the former rather than the latter as we already have an algorithm which produces the three layers and thus, images are cleaner in order to generate the patches for each shape/class.

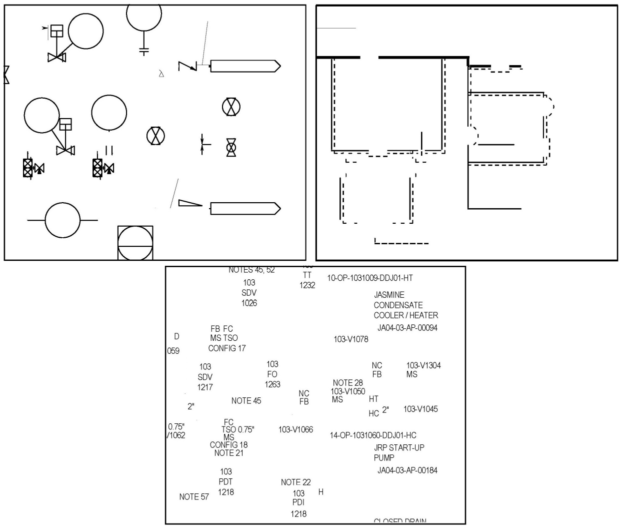

Fig. 3. Symbols (top-left), connectors (top-right) and text (bottom) layers produced from a P&ID using the heuristics-based tool.

The size of the patch shall be determined in accordance to the size of the drawing. Given that these images are approximately of 5000×7000 pixels, at least a 25×25 window is required, following the guideline proposed in [21].

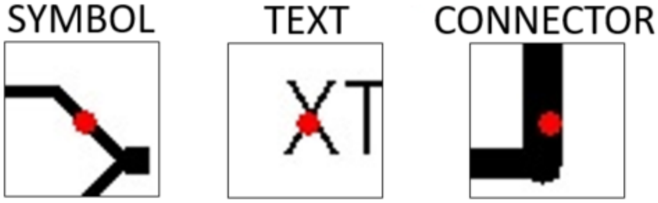

Fig. 4. Samples of patches from the symbol, text and connector layer respectively. The central pixel of interest is marked in red.

The network configuration is as follows: A CNN with a depth of 3 layers, 1 convolution per layer, 32 filters and kernel size of 3×3 pixels has been set up as a starting point. Each layer consists of stacked convolutions and a Rectified Linear Unit (ReLU) activation function, followed by a fixed 2×2 down sampling (pooling) function. The final layer is fully connected, leading to a softmax classifier returning the probability for the pixel patch to belong to a given class. For each of the convolution layers, The size of the 32 filters has been fixed to 3×3 kernels for the first layer, followed by two layers with 64 filters and the same fixed size and a ReLU activation function. The network was compiled using a RMSProp gradient descent optimiser, with a categorical cross-entropy loss function.

IV. Experimental Validation

The experimental validation is divided into four subsections. First, the P&IDs used and the parameters to obtain the training data are briefly described. Then, the metrics to be considered in this study are defined. Finally, the experimental study on how different window sizes used to train the CNN is presented, when attempting to segment pixels from P&IDs of the same standard as the drawings from where the training samples where obtained. Finally, we discuss the average run time that takes for the heuristics-based tool and the CNN to produce the three layers for a single page. A direct accuracy comparison between the heuristics tool and CNN method is not pertinent because our heuristics-based tool achieves 100% accuracy due to the fact that it allows human interaction.

A. Data used

We have used eight drawings from a P&ID standard similar to the one shown in Figure 1. Due to confidentiality reasons, these drawings cannot be shared in full in the public domain, although some work has been presented related on symbol classification [10], where the reader can be referred to observe the quality and characteristics of the shapes. These drawings have been processed using a heuristics-based tool [6] to segment the drawing into the three layers of interest. As a result, a total of 24 layer drawings have been obtained.

To obtain the training samples, a sliding window method has been implemented to iterate over the whole image, with a fixed stride of 10 pixels. The window verifies that the centre pixel is of interest, and then produces a $width \times height$ patch for training. The dimensions of the sliding window are matched to the individual classifier under test, and are always odd so that there is always a defined centre pixel.

B. Metrics

Given that the CNN has to classify an input pixel into one of three different classes, and that a large amount of samples can be obtained from each image (a single P&ID can have approximately $4'000'000$ shape pixels), we have used precision, recall, and the $F_1 score$ to determine the accuracy performance for each class. These metrics are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TOT} \tag{2}$$

$$F1_{score} = 2 \times \frac{P \times R}{P + R} \tag{3}$$

where TP represents the true positives (i.e. correctly classified pixel for a specific shape), FP represents the false positives (i.e. incorrectly classified pixel for said shape) and TOT is the total number of pixels belonging to the class. In addition, we have included the average run time to train a model and to classify all pixels on an image. This is an important consideration given the large number of legacy paper P&ID images there are in existence.

Experiments have been carried out using a PC with an Intel(R) Core CPU @ 2.70 GHz processor, 16 GB RAM and Windows 10 as operating system. The code was implemented

TABLE I

AVERAGE PRECISION (P), RECALL (R), $F_1 score$ (F_1) FOR DIFFERENT WINDOW SIZES. THE BEST VALUES FOR EACH COLUMN ARE HIGHLIGHTED IN BOLD.

	Runtime		Symbols				Text				Connectors			
Window Size	Train	Test	Samples	P	R	F_1	Samples	P	R	F_1	Samples	P	R	F_1
25x25	**548.25**	**3320.06**	15394	0.68	**0.95**	0.79	12576	**0.97**	0.96	**0.96**	18039	0.86	0.36	0.50
25x51	1529.54	4637.73	**15785**	**0.73**	0.90	**0.80**	12809	0.91	**0.97**	0.94	18005	0.87	**0.46**	**0.59**
51x25	1582.36	4633.84	15556	0.71	0.82	0.76	13084	0.84	**0.97**	0.90	18882	0.84	0.45	0.58
51x51	1927.72	5752.99	15776	0.68	0.91	0.77	**13287**	0.87	**0.97**	0.92	18787	0.93	0.36	0.51
51x75	6062.34	13828.59	15612	0.70	0.88	0.78	13241	0.84	**0.97**	0.90	18002	0.93	0.41	0.56
75x51	6309.22	13789.33	15430	0.68	0.75	0.71	12736	0.71	0.95	0.80	**19998**	0.93	0.40	0.55
75x75	6295.82	16192.98	15179	0.67	0.87	0.75	12763	0.83	0.94	0.88	19188	**0.94**	0.37	0.53
101x101	11849.01	17143.04	15457	0.66	0.85	0.74	12055	0.76	0.91	0.82	18500	0.93	0.34	0.49

using Python 3.6, with a Keras framework and TensorFlow as back-end.

C. Implementation

Table I shows the average runtime (train and test), samples obtained per class, precision, recall and $F_1 score$ for a two-fold cross validation of the dataset for different window sizes. In terms of runtime, it is clearly noticeable that a 25×25 window size offers better training and testing results in comparison to the other configurations. Since the stride remains constant, all configurations worked with similar number of pixel patch samples, where we can notice that the majority of them correspond to the *connector* class ($\sim 18k$ samples). Still, the other two classes (*text* and *symbol*) work with a comparable number of samples, with $\sim 15k$ and $\sim 13k$ samples for symbols and text respectively.

In terms of accuracy, a window size of 25×25 shows a high recall on symbols, the best $F_1 score$ on text, but poorer results for connectors compared to other window sizes. The rectangular patch of 25×51 pixels delivers not only the highest number of symbol pixel patch samples, but also the best precision and $F_1 score$ for this class, as well as the best recall for text and the best recall and $F_1 score$ for connectors. It has been noted in advance that these P&IDs contain more vertical than horizontal lines, which may be the case of the improved performance with respect to the inverse 51×25 window size. In the remaining window sizes, we only notice a superior performance in terms of precision for connectors in the case of the 75×75 window size. It can be said that the larger window sizes lead to an increase in precision, possibly because they eliminate some of the false positives. Theoretically, any straight line can be potentially mistaken for a connector unless some other feature precludes it as such. Enlarging the window size increases the chances that the window will contain some feature that distinguishes a connector from another class. Finally, the 101×101 window size experiment confirms the observations of [21], where it was shown that large window sizes yield poor results. This may be attributed to the fact that while small window sizes are likely to contain a single class of object in the majority of pixels, larger window sizes may contain substantial proportions of more than one class of object, thus confusing the classification.

To have a visual confirmation of these results, Figure 5 shows examples of the segmentation of the P&ID of Figure 1 for the following window sizes: A) 25×25, B) 25×51, C) 51×25 and D) 75×75, where symbols are indicated in blue, text in green and connectors in red. Firstly, it can be confirmed that thin connectors, such as the left-most connector in the image, are largely misclassified as a symbol for the 25×25 window size case. The same is true for parts of the horizontal dashed connectors. While it looks like the horizontal dashed connectors in (A) have been misclassified as symbols, a closer inspection reveals that the *ends* of the thin dashes are, in fact, correctly classified. The larger window sizes correctly classify more of the dashes because they are more likely to encompass the end of the dash. In contrast, using the 25×25 window size it is possible to correctly classify all of the thick connector (i.e. horizontal line that connects the bottom-left symbol with the bottom-right arrow) with more precision than the other cases. This can be partially attributed to the fact that this window size is not able to capture enough information for the thin connectors and horizontal dashes, but is also unable to capture confounding information to decrease the accuracy on the thick connector.

By visually comparing the 25×51 (B) and the 51×25 (C) windows, it can be appreciated many more vertical connector pixels are obtained in comparison to the second one, which performs very similar to the 25×25 case. Nonetheless, the segmentation of horizontal line connectors is comparable in both cases.

Finally, in the case of the 75×75 (D) window size case, the accuracy on detecting the vertical line connector on the left is very high, however as the window size becomes bigger, the likelihood of segmenting portions of the symbols as text increases. This can be noticed in the three symbols in the centre, where it can be seen that parts of the circles have been labelled as text. This effect can be viewed to a lesser extent in (B) and (C) where the shape of the misclassified pixels along the symbol edge bears some resemblance to the window shape.

D. Run time comparison

As stated before, the heuristics-based tool that we currently use[1] to digitise the P&IDs and generate the three layers is a semi-automatic software which allows human interaction after each stage. This means that the tool detects the elements sequentially and then shows the provisional result to the user.

[1] http://cfmgcomputing.blogspot.co.uk/p/circuits-dev-digitisation-tool.html

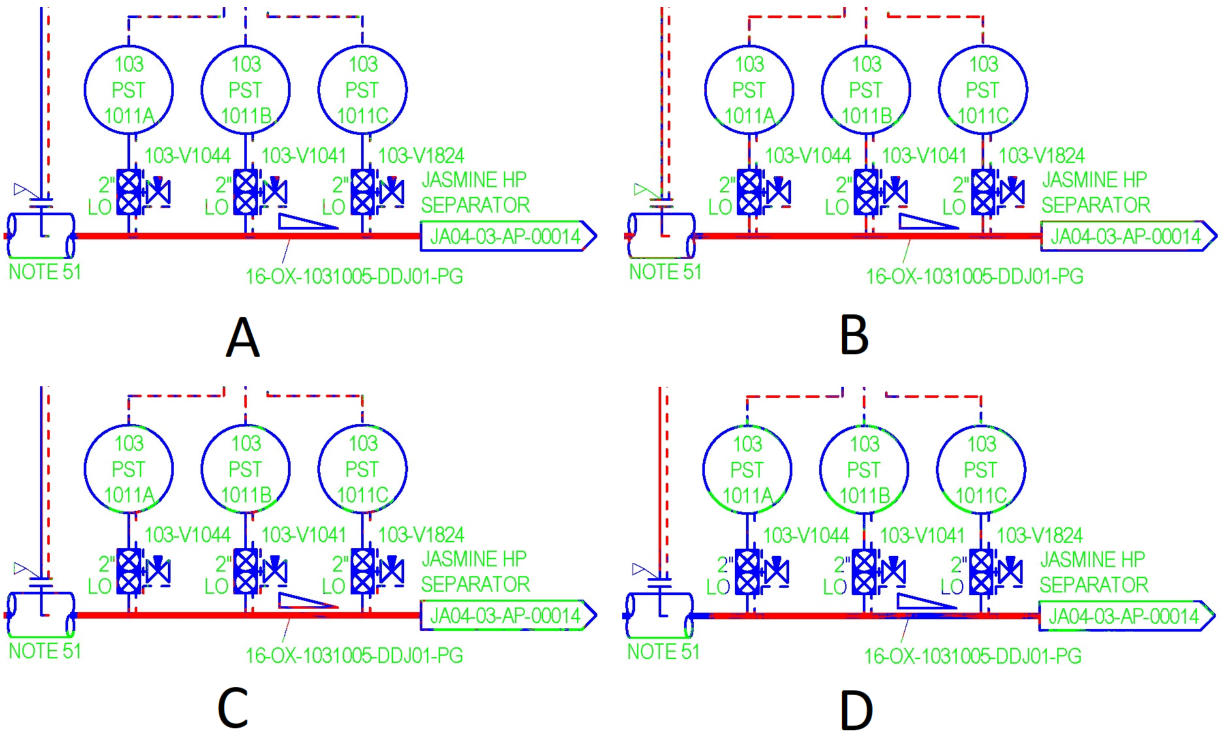

Fig. 5. Pixel segmentation (blue = symbol, green = text, red = connector) of the P&ID shown in Figure 1 for A) 25×25, B) 25×51, C) 51×25 and D) 75×75.

Then, the user is capable of manually re-assigning any shape to the correct layers, Notice that this process can be very tedious due to the fact that the human has to examine the drawing again and review until all shapes have been detected and classified correctly. Our tests with both programmers and human experts from the Oil & Gas Industry have revealed that to digitise a single page a user takes on average between 3 and 5 hours of work, depending on their expertise both on using the tool and in the domain.

In contrast, the aim of presenting a pixel-based CNN approach in this paper is to show that by learning from a few samples digitised with the heuristics-based tool, it is possible to further automate the task of producing the three layers and then applying other recognition methods (i.e. line detection to the connector layer, optical character recognition to the text layer and any image classifier to the symbol layer) to correctly classify each shape. After experimenting with the new tool, we noticed that the time to digitise a single page can be reduced to 30 min to 1 hour, depending on the expertise. This is vast reduction of time and poses an interesting reduction of human effort, which as evidenced in [22], is essential for the industry.

V. Conclusion

This paper presents a first step towards defining a methodology for pixel-based segmentation of symbols, text and connectors in a class of complex engineering drawings known as P&IDs. The framework to implement this methodology is composed of two steps. The first is based on the generation of the layers containing the elements of the three classes through a heuristics-based algorithm presented in [6] designed for a particular P&ID standard. The second is a state-of-the-art CNN architecture trained using pixel patches which are obtained from the layers produced. Experimental validation for different window sizes shows that a size of 25×25 pixels yields good results in terms of runtime (training and test), while obtaining good results in terms of class precision, recall and $F_1 score$. In addition, the use of rectangular window sizes, such as 25×51 or 51×25, and of larger squared patches, such as 75×75, may increase the accuracy to segment connector pixels, with the drawback of increasing the false positive text classification.

There are numerous ways in which the performance of the system can be improved. To start, it is acknowledged that the CNN architecture used was an out-of-the-box option and few parametrisation was employed. The influence of regular and hyperparameter tuning in this and any CNN architecture has to be thoroughly studied, such as, the convolutional layers, batch size and epochs, amongst others. Moreover, it is intended to use different strides to see if a reduced or increased number of samples has an effect on these results.

Beyond these efforts, there are many more ways in which the segmentation may be re-assessed after the classification produced by the CNN. For instance, it could be possible to reclassify pixels whose neighbours have a different class. One proposed solution is using frameworks similar to [20], which enhances the accuracy of pixel classification in heterogeneous regions, or in cases where the majority of pixels are of a certain class within a specific contour. Another option is to use more robust features, and not feeding the CNN with the pixel patch directly. To that aim, it is possible to consider sparseness-based features such as the ones used by [18]. Our future work will focus on refining this method and enhancing it in order to fully automate accurate digitisation and contextualisation of P&ID images.

Acknowledgement

We would like to thank the Oil & Gas Innovation Centre (OGIC), The Data Lab (TDL) Scotland and Det Norske Veritas Germanischer Lloyd (DNV GL) for supporting this work. In addition, we would like to thank Dr. Eyad Elyan and Dr. Brian Bain for their contributions to this work.

References

[1] M. Furuta, N. Kase, and S. Emori, "Segmentation and recognition of symbols for handwritten piping and instrument diagram," in *Conference Proceedings of the 7th International Conference on Pattern Recognition (ICPR)*, 1984, pp. 626–629.

[2] 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.

[3] C. Howie, J. Kunz, T. Binford, T. Chen, and K. H. Law, "Computer interpretation of process and instrumentation drawings," *Advances in Engineering Software*, vol. 29, no. 7-9, pp. 563–570, 1998.

[4] X. Jiang, A. Munger, and H. Bunke, "Synthesis of representative graphical symbols by computing generalized median graph," in *Conference Proceedings of Graphics Recognition Methods and Applications (GREC)*, vol. 1941, 2000, pp. 183–192.

[5] W. C. Tan, I. M. Chen, and H. K. Tan, "Automated identification of components in raster piping and instrumentation diagram with minimal pre-processing," *Conference Proceedings of the IEEE International Conference on Automation Science and Engineering (ICASE)*, vol. November, pp. 1301–1306, 2016.

[6] 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.

[7] L. A. Fletcher and R. Kasturi, "A 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.

[8] K. Tombre, S. Tabbone, B. Lamiroy, and P. Dosch, "Text/Graphics Separation Revisited," in *Conference Proceedings of the IAPR International Workshop on Document Analysis Systems (DAS)*, vol. 2423, 2002, pp. 200–211.

[9] C. F. Moreno-García, E. Elyan, and C. Jayne, "New trends on digitisation of complex engineering drawings," *Neural Computing and Applications*, pp. 1–18, 2018.

[10] E. Elyan, C. F. Moreno-García, and C. Jayne, "Symbols classification in engineering drawings," in *Conference Proceedings of the International Joint Conference in Neural Networks (IJCNN)*, 2018.

[11] D. Ramyachitra and P. Manikandan, "Imbalanced dataset classification and solutions: A review," *International Journal of Computing and Business Research*, vol. 5, no. 4, pp. 2229–6166, 2014.

[12] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *International Journal of Advances in Soft Computing and its Applications*, 2015.

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

[14] E. Elyan and M. M. Gaber, "A fine-grained random forests using class decomposition: an application to medical diagnosis," *Neural Computing and Applications*, vol. 27, no. 8, pp. 2279–2288, 2016.

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

[16] D. P. Panda and A. Rosenfeld, "Image segmentation by pixel classification in (gray level, edge value) space," *IEEE Transactions on Computers*, vol. C-27, no. 9, pp. 875–879, 1978.

[17] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[18] M. Cote and A. Branzan Albu, "Texture sparseness for pixel classification of business document images," *International Journal on Document Analysis and Recognition*, vol. 17, no. 3, pp. 257–273, 2014.

[19] P. O. Pinheiro, R. Collobert, and E. De Lausanne, "From image-level to pixel-level labeling with convolutional networks," in *Conference Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[20] W. Li, G. Wu, S. Member, and F. Zhang, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2017.

[21] J. Calvo-Zaragoza, F. Castellanos, G. Vigliensoni, and I. Fujinaga, "Deep neural networks for document processing of music score images," *Applied Sciences*, vol. 8, no. 5, p. 654, 2018.

[22] E. Rica, C. F. Moreno-García, S. Álvarez, and F. Serratosa, "Reducing human effort in engineering drawing validation," *Computers in Industry*, vol. 117, 2020.