JAMIESON, L, MORENO-GARCIA, C.F. and ELYAN, E. 2020. Deep learning for text detection and recognition in complex engineering diagrams. In *Proceedings of the 2020 Institute of Electrical and Electronics Engineers (IEEE) International joint conference on neural networks (IEEE IJCNN 2020), part of the 2020 IEEE World congress on computational intelligence (IEEE WCCI 2020) and co-located with the 2020 IEEE congress on evolutionary computation (IEEE CEC 2020) and the 2020 IEEE International fuzzy systems conference (FUZZ-IEEE 2020), 19-24 July 2020, [virtual conference].* Piscataway: IEEE [online], article ID 9207127. Available from: https://doi.org/10.1109/IJCNN48605.2020.9207127

# Deep learning for text detection and recognition in complex engineering diagrams.

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

2020

# Deep Learning for Text Detection and Recognition in Complex Engineering Diagrams

Laura Jamieson
School of Computing Science and Digital Media
Robert Gordon University
Aberdeen, AB10 7GJ, UK
Email: l.jamieson4@rgu.ac.uk

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

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

***Abstract*—Engineering drawings such as Piping and Instrumentation Diagrams contain a vast amount of text data which is essential to identify shapes, pipeline activities, tags, amongst others. These diagrams are often stored in undigitised format, such as paper copy, meaning the information contained within the diagrams is not readily accessible to inspect and use for further data analytics. In this paper, we make use of the benefits of recent deep learning advances by selecting models for both text detection and text recognition, and apply them to the digitisation of text from within real world complex engineering diagrams. Evaluation of such deep learning methods on a dataset of Piping and Instrumentation Diagrams from the Oil & Gas industry showed promising results on detecting and recognising text, without the need for pre-processing steps in complex engineering diagrams.**

## I. Introduction

It is common across many industry sectors for engineering diagrams (EDs) to be stored in an undigitised file format or as a paper copy. Digitisation of these documents is of importance to allow improved use of this vast amount of data. EDs can be very complex and contain text annotations in addition to a number of different components including vessels, symbols and connecting lines. In digitising these documents, the detection and recognition of the text elements also known as Optical Character Recognition (OCR) is a key part of the document digitisation. The ability to accurately read text in images is important for many applications. Images with low resolution, noisy and complex images with overlapping elements all present challenges for text detection and recognition, therefore methods are still in need of improvement.

OCR systems are typically comprised of 1) image acquisition, 2) pre-processing, 3) segmentation, 4) feature extraction, 5) classification and 6) postprocessing. Prior to the use of deep learning in text detection and recognition, features, predominantly low level or mid level, were extracted, requiring many pre and post processing steps. Colour, texture and edge features were often used for text localisation. Approaches commonly used connected component analysis or sliding windows [1] for that matter. In particular, a family of approaches known as *Text/Graphics Separation* (TGS) methods [2] were used for drawings such as general purpose EDs [3], circuit diagrams, maps [4] and musical scores, with moderate success.

More recently, deep learning has greatly advanced computer vision, one area being the topic of general object detection. Whilst text has specific properties in comparison to generic object detection, the detection of text can be viewed as a subset of object detection, and generic object detection models can be trained on text images as in [5], where authors adjusted the popular YOLO v3 model [6] in order to detect text. Whilst text detection may be viewed as a specific type of object detection, the specific properties of text mean that tailored models can be developed. A robust text detection method should be designed with features that distinguish text from the background. In deep learning text detection methods, the distinguishing features are learned from the training data, and thus it is also key to take this aspect into consideration. For the case of EDs, sometimes it is common that background is not a discerning factor from the text (as these most commonly have a clear, distinguishable white background); however, there are other complications when attempting to train a system that recognises text, mostly related to false positive elements such as symbols, connectors, amongst others.

Detection and recognition of text has specific challenges in comparison to general object detection, and thus it is relevant to create specific methods tailored for this task. One property of text unlike generic object detection is, the whole text object does not need to be visible for the object to be recognised as a piece of text. Text detection and recognition are more sensitive to blurring than general object detection. Deep Learning detection methods specifically developed for detection of text include Efficient and Accurate Scene Text (EAST) Detector [7], Connectionist Text Proposal Network (CTPN) [8], TextBoxes [9], TextBoxes++ [10], Fastext [11].

The contributions provided in this paper are as follows:

- Discuss and evaluate a deep learning-based text detection method for its applicability to EDs, specifically the Efficient and Accurate Scene Text (EAST) Detector.
- Discuss and evaluate a text recognition method that uses deep learning in the form of Long Short Term Memory (LSTM) networks, evaluating its performance on the regions selected by the detection method.
- Test both the text detection and recognition methods on a real world dataset of complex EDs, specifically Piping

and Engineering Diagrams (P&IDs), provided through industry collaboration with the Oil & Gas industry, to demonstrate how the deep learning-based methods can improve the analysis of text in this scenario.

The rest of this paper is organised in the following sections: Section II describes related work in text detection in complex documents. Section III describes the dataset used and the methods used in our experiments. Section IV describes the experiments and discusses the results. The conclusion and suggestions for future work are presented in Section V.

## II. Related Work

Text detection and recognition methods can operate at two levels: character level and word level [12], [13]. Character level methods rely mostly on heuristic-based segmentation techniques that are able to distinguish text from other shapes based on innate characteristics of the letters and/or numbers such as size, stroke and geometry. This works particularly well in high resolution EDs and similar printed drawings, since it is common that the text font remains constant throughout the document. After the identification of letters occurs, these methods rely on techniques to constitute strings (mostly through character proximity and alignment) and to classify each character individually to interpret such strings. In contrast, word level recognition is primarily suited to situations where there is a restricted number of possible words in a document, allowing lexicons to be used in conjunction with character recognition outputs. Such is the case of P&IDs, which contain a set of standardised codes that have a fixed structure and thus, are easier to detect and recognise by using approaches of this nature. For instance, a lexicon of words to narrow down the number of possibilities for word recognition was a strategy used to improve the OCR accuracy in [14].

As stated in Section I, TGS methods were initially a very popular option for text detection in the document image analysis community some years ago, given their simplicity and robustness [15]. One of the cornerstones of this area was the work presented by Fletcher and Kasturi in 1988, where authors proposed the use of connected component analysis (CCA) to select text characters based on a predetermined size and width-to-height ratio. Afterwards, the resulting text layer was converted into its Hough transform to analyse the linearity of the characters and deduce the strings conformed. This approach resulted very favourable for simple EDs, however it was incapable of dealing with test overlapping shapes and with short strings (i.e. less than 3 characters of length). A number of reviews and upgrades were done to this working pipeline, mostly at the string conformation stage, such as Lu [16] who used a *brushing* morphological operation to join strings, Tombre et al. [2] who were able to discard dashed connectors from the text layer and applied proximity analysis for the string generation, and Tan and Ng [17] who by using a *pyramid* approach were able to scaled down the text layer until being able to find the optimal string conformation. Most recently in [18] authors, presented a comparison of different TGS methods to reduce the overall identification of shapes and connectors in P&IDs. It is worth to note that for the text recognition stage, most state-of-the-art methods rely on OCR for the text interpretation, nonetheless there is work in literature where character classification is preferred as it is more suited for EDs. A study by Das et al. [19], involved identifying areas of text in architecture, engineering and construction documents through traditional methods, however the study did not attempt to read the text instead focussing on classifying text as either machine printed or handwritten.

One essential drawback of TGS methods is their general inability to deal with text overlapping other shapes of the ED. Although some work has been presented in this matter [20], this usually relies on a series of heuristics that are not always applicable and thus have various rates of success depending on the overall quality of the ED. Moreover, it has been noted by authors such as Ye and Doermann [13] that general object detection methods would not perform well for text detection in a more general setting, based on a comparison of average images of three object types namely faces, pedestrians and text. In their experimental setting, the average images were composed of the mean of $2'000$ aligned samples of each object type and whilst the face and pedestrian image retained common features, the average text image resembled noise.

In methods more related to the domain of EDs, and specifically P&IDs, Sinha et al. [21] presented work on extracting text information from scanned raster versions of P&IDs. The proposed method however focussed only on text within tables, and used initial steps including contour detection to detect tables in the diagram. The method was tailored specifically for P&ID dataset used, with the tables detected having to match one of three specified formats containing specific keywords. To extract the table information, version 3.05 of Tesseract OCR and Python RegEx string matching were used. The method correctly identified $87.2\%$ of the tables present, however inconsistencies in the information extraction occurred, potentially due to some text touching table borders and logos appearing as text.

In [22], a study on detecting characters in EDs was presented that used a convolutional object detector based on Overfeat [23], Faster R-CNN RPN [24] and Feature Pyramid Networks [25]. The detector took a single image as input without preprocessing and output class confidence scores and bounding box predictions. The system was tested on a dataset of 150 EDs and results only showed passable accuracy with some misclassifications and false negatives. Moreover, Eman et al. [26] presented work aimed at improving OCR accuracy in complex cursive scripts, using conditional GANs to transform cursive text into straight scripts, where characters are not joined, before LSTM based OCR was carried out. Results, evaluated on character level error rate with the Levenshtein distance, showed improvement with the recognition of handwritten and italicised cursive scripts.

Traditional methods for text detection were compared with deep learning methods on text in floor plan images in [14]. The analysis compared four methods: 1) EAST, 2) Connectionist Text Proposal Network (CTPN), 3) a standard image pro-

cessing approach using Maximally Stable Extremal Regions (MSER) and Stroke Width Transform (SWT) and Tesseract to discard areas of non-readable text, and 4) a combined approach with all of the first three methods. For the CTPN method, additional sub images along the border were used as CTPN struggled with identifying text close to the image borders [14]. The combined method compared results from all three other methods against each other to produce an output based on voting. Post processing was carried out on all methods to merge specific text boxes into one text item. The text was firstly classified based on rules, then room descriptions were compared with a dictionary of valid words and replaced with the closest word based on edit distance and word frequency.The proposed methods were evaluated on datasets of varying quality. Performance with the CTPN method was shown to be significantly reduced by the noise and low resolution images. On the low quality images, the EAST method had the highest recall and F1-score, whilst the combined method had the highest precision. None of the proposed methods were able to detect vertical or curved text items and the accuracy of the recognised text wasn't analysed in detail, however it was noted that Tesseract did not give correct predictions on the low resolution images.

All of the aforementioned methods work with a varying degree of success for complex EDs such as P&IDs, as this type of printed drawings present multiple challenges, such as a dense and entangled structure between shapes, a complex hierarchical relation between elements, overlapping of text with other shapes and the similarity between symbols and text, amongst others.

## III. Methods

### A. Dataset

Engineering diagrams used in industry are not widely available in the public domain primarily due to data confidentiality reasons. To evaluate the methods on real world data we have, through collaboration with an industry partner, obtained a dataset of P&IDs. The dataset we have chosen to use will allow the selected deep learning methods to be evaluated on real world complex engineering diagrams; the P&IDs in the dataset are from the Oil and Gas industry however P&IDs are also used in many other industry sectors to convey and store information about process equipment and its operation.

The dataset comprises 172 complex P&IDs, which contain components of symbols, connector lines and text annotation, as shown in Figure 1. Due to data confidentiality reasons only a section of the diagram is shown.

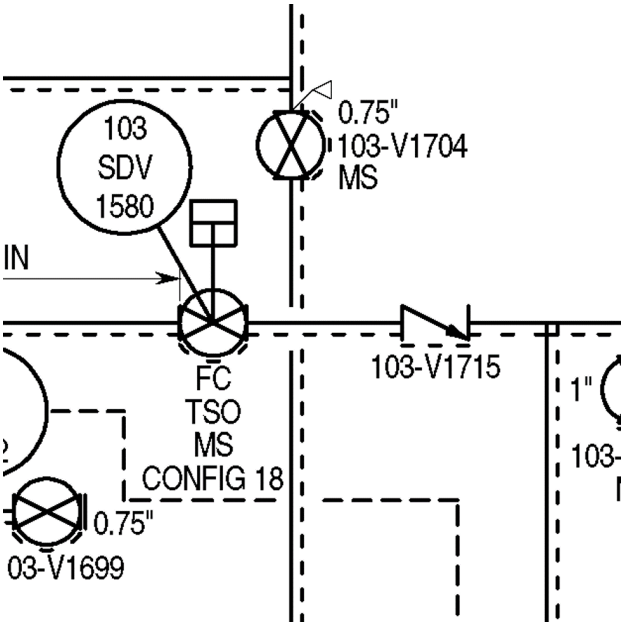


Fig. 1. Section of P&ID showing Symbols, Connectors and Text Elements

Text in the P&IDs dataset analysed, can be split into two types according to its purpose and location. Text located in the main diagram is used to annotate the graphical elements, including equipment tags whereas the second type of text is located within in the diagram template and is used to provide additional details including drawing number, revision history, and further related details about the equipment shown in the diagram. Analysis of a subset of P&IDs showed that in the main diagram section, there are approximately 415 text instances per P&ID.

Text within the P&IDs is used to annotate the equipment in the diagram and includes text that ranges from short text strings including two character annotations, through to line numbers and equipment tags, to longer full sentences showing operating information for equipment in the diagram. Text is located vertically in addition to horizontally, one such situation where vertical annotations are used occurs where an associated line number is aligned next to a vertical pipeline. There are also several text strings printed diagonally to align with equipment. Text is located throughout the diagram, with some text in close proximity to other components and there are text annotations situated within symbols and vessels. In P&IDs parts of some non text components are similar in appearance to text characters and certain elements such as dashes occur in both text and non text components. Dashes are present in a large amount of text strings such as equipment tags, whilst dashed lines are used as a form of connection line between two pieces of equipment and located adjacent to pipelines to indicate a property of the line.

Therefore the images in the dataset chosen, contain several challenges related to both text detection and text recognition and are suitable for evaluation of deep learning methods applied to digitise text from within complex engineering diagrams.

### B. Text Detection

A deep learning method, the Efficient and Accurate Scene Text (EAST) detector [7], was chosen to detect text instances in the P&IDs. The EAST detector is reported to outperform other state of the art methods when evaluated on F-score on the text detection tasks COCO-Text [27], ICDAR 2015 Challenge Text Localization Task [28] and the MSRA-TD500 [29] dataset.

At the time of EAST proposal, existing methods were commonly designed with several stages, [30], [31]. The EAST detector does not contain any intermediate steps like candidate proposal, instead it produces text predictions directly from a single neural network. Output results from the neural network are then filtered using a very similar process to Non Maximum Suppression (NMS) where the geometries are averaged as opposed to being selected.

More specifically, EAST uses a Fully Convolutional Network (FCN) which is trained to predict word or text line instances from full images. The network was based on the general design of DenseBox [32]. The FCN architecture used in EAST can be split into a feature extractor stem, feature merging branch and an output layer. The purpose of the feature merging branch is to improve detection of both small and large word regions by merging features from both lower and higher layers of the feature extractor. The output channels consist of a score map in the range [0,1] which represent the confidence of the geometry shapes, which are predicted in the other output channels. Output geometries are predicted as a rotated box or quadrangle.

The loss function used in training the model comprises the loss for the score map and the loss for the geometry predictions. Zhou et al. [7] selected a balanced cross-entropy loss for the score map. A scale invariant loss is selected for the geometry, to ensure accurate predictions for both large and small text regions. The two loss functions chosen for the geometries are IOU loss for rotated rectangle output predictions, and scale normalised smoothed L1 loss for the quadrangle geometries. Zhou et al. [7] trained the EAST network end-to-end using the ADAM optimiser [33].

### C. Text Recognition

A deep learning method, specifically Long Short Term Memory networks, was used for text recognition. LSTM, Long Short Term Memory, [34], network is a type of a RNN (Recurrent Neural Network) designed to retain information from long sequences.

### D. Pre-Processing and Post-Processing Data

*1) Selection of Input Image into EAST Detection Model:* In the dataset used, the ED had a template format with a table that was consistently located on the right hand side. The focus in this study was to interpret the main part of the diagram itself, therefore the text detection and text recognition will be applied only to text located in the main diagram and not the text information in the diagram template.

The method selected to discard the template was based on connected components and it was chosen as it is independent of template layout, as the method does not require heuristics based on the position of the template within the diagram. The method used to select the diagram area would also be applicable to other datasets, including those without a border line.

The largest white component by area represents the background of the main diagram itself, and thus the connected components algorithm method was used to determine the largest connected component in the P&ID by area in order to select the diagram area to be processed by the text detection model.

The P&ID diagrams were large images, approximately 7500 x 5250 pixels in size. To process the whole diagram area in one step by the EAST model would need a high amount of computational requirements therefore to reduce processing requirement, the diagram is processed by the EAST detector in four patches. The image patches to be processed were obtained by dividing both the height and width of the selected diagram area in half.

*2) Post-Processing of Text Bounding Boxes:* Padding was applied to the detected text boxes to ensure that all of the text string was included in the bounding box. To make the method applicable to text regardless of font size, the amount of padding added was calculated as a percentage of the original detected text box size. The height of the text box was padded by 10% and the width was padded by 12% at the start and 24% at the end of the string.

A post processing step was then taken to merge nearby detected text boxes based on the proximity of the bounding boxes. Detected text boxes were split into horizontal text or vertical text based on the ratio of the width to height of the bounding box. The area of overlap between each pair of detected text boxes was calculated and if text boxes overlapped, they were combined into the smallest bounding box that would combine both original detected boxes. The resultant bounding boxes were then used as input for the text recognition step.

## IV. Experiment

### A. Setup

Experiments were run to evaluate the performance of the selected Deep Learning text detection and text recognition methods on complex EDs. To evaluate the chosen methods on real world data, experiments were performed on the dataset of 172 P&IDs from the Oil and Gas industry.

The performance of pre-trained models for both text detection and text recognition was investigated, therefore no specific training of the Deep Learning models was carried out with the P&ID dataset.

The text detection and text recognition methods were applied on the P&IDs by creating a framework to process the diagrams. Results were evaluated by displaying the results on the processed P&ID. Bounding boxes were shown on the detected text instances, with the output string from the text

recognition step shown adjacent to the detected bounding box. Additionally for evaluation purposes, output files listing the detected bounding box co-ordinates, dimensions and predicted text output, were also produced.

A pretrained EAST model [7], was used to locate text instances. The P&IDS were large images, approximately 7500 x 5250 pixels in size and therefore due to processing limitations, patch detection was used for the text detection step. Patches were obtained by splitting the area to be processed into four equal patches by dividing it in half across the height and width. Text strings located across more than one patch were therefore split into multiple sections when input to the detection model.

To perform text recognition, open source LSTM based Tesseract engine was utilised. Speed of processing the diagrams was an important factor in this study, therefore the smallest LSTM network from Tesseract was chosen as this had the fastest processing speed available, however this LSTM model was also associated with decreased accuracy levels compared to the larger LSTM network model available in Tesseract.

### *B. Results & Discussion*

The P&ID images produced from the experiments with results overlaid show that the EAST detection model and LSTM based text recognition method gives promising results when applied to detect and read text in complex engineering diagrams. Results from experiments on the real world P&ID dataset are discussed in further detail below.

In experiments, the EAST model was able to detect varying orientations of text, as stated in [7], with both horizontal and vertical text instances in the diagram being detected. The model was also able to detect text strings of varying lengths. Samples of text strings that were correctly detected and correctly recognised are shown in Figure 2.

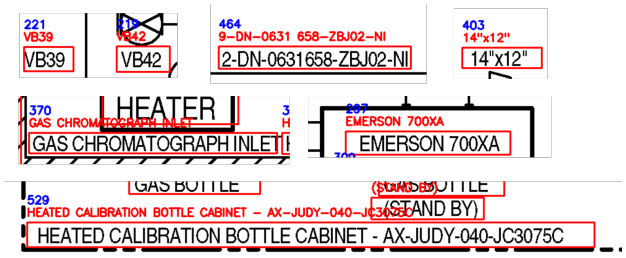


Fig. 2. Instances where text was correctly detected and recognised

To analyse the results in further detail, from the dataset of 172 P&IDs we have selected five representative P&IDs for which to present the text detection and recognition results. Table I shows the numbers of text instances present in each diagram, the number of detected instances, number of False Negative (FN) and False Positive (FP) detections. The number of text detections with associated text string from the recognition step is also listed.

Analysis of this subset of drawings shows that on average, there are 415 text instances, counted as one text string or multiple text strings that would be combined into one detection by the post-processing, present in each diagram. When images were passed to the EAST detection network, 90% of the text instances were successfully detected, without the need for any pre-processing of the image or training on the specific font from the P&IDs.

TABLE I
ANALYSIS OF TEXT DETECTION AND RECOGNITION ON SELECTED P&IDS

| Diagram No. | Text Instances | Detected | FN | FP | Recognised |
|---|---|---|---|---|---|
| 1 | 426 | 388 | 54 | 16 | 337 |
| 2 | 492 | 463 | 42 | 13 | 384 |
| 3 | 545 | 506 | 61 | 22 | 439 |
| 4 | 407 | 385 | 37 | 15 | 333 |
| 5 | 201 | 194 | 16 | 9 | 167 |

False negative detections, where non-text elements of the diagram were detected as text and the detected results contained no text characters, were observed to occur on average in only 4% of output detections based on the sample set analysed. This situation occurred in particular with dashed lines detected as text, whilst other elements also resemble text characters, refer to Figure 3.

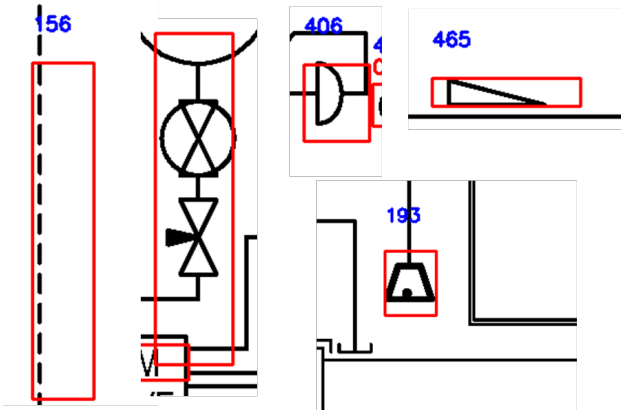


Fig. 3. Instances of P&ID Elements Misdetected as Text

There were also instances of text that were not detected by the chosen method. Results show that on average, there were 42 text instances undetected in each P&ID, approximately 11% of text instances.

*1) Text Detection:* Incorrect bounding boxes round the text strings were observed to occur in three scenarios, 1) partially detected text string, where one or more characters is determined not to be text by the deep learning model, 2) non-text elements determined to be text data and 3) non-text components included in the text area bounding box as a result of the post processing steps of merging text boxes and padding. Furthermore it is possible for combinations of these scenarios to occur in one output text box.

There are many technical annotations used in the engineering diagrams, including line numbers that often start with a single character followed by a dash, in many instances the start of this text string was missed from the detected bounding box.

It was observed that when text was located in close proximity to other components, the non text components could be included in the bounding box, likely due to the padding applied in the post-processing.

One of the images in the dataset consisted of a table containing line numbers, rather than a diagram. Text was detected in every cell of the table. The only text strings not detected were short strings of two letters. Additionally some of the text was joined into blocks and detected, and in several instances the string was partially detected.

*2) Text Recognition:* Results of the detection step feed directly to the recognition step, therefore obtaining a good output from the detection step, allows a cleaner image of the text string to be passed to the text recognition step.

Whilst the EAST model was able to detect text in the vertical direction, instances where the vertical text appeared to have been read in the wrong direction were observed, refer to Figure 4.

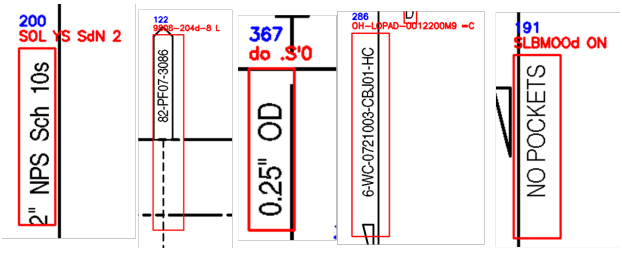


Fig. 4. Incorrect Recognition of Vertical Text, Output of Text Recognition in Red

## V. Conclusion

Deep learning methods have brought advancements in the area of image text detection and recognition. However the benefits of these Deep Learning models have still to be applied to the problem of digitising text information from complex EDs.

State of the art Deep Learning methods for text detection and text recognition were used for this study. Evaluation of these methods was carried out on a dataset of 172 complex EDs from the Oil and Gas industry, specifically P&IDs. Experiments show promising results for both text detection and text recognition, without any traditional preprocessing of the P&IDs or pretraining the Deep Learning models on the specific font used in the diagrams.

Future work will look at increasing the accuracy of the deep learning methods including by training the models on the specific font and testing with the more accurate, although slower, LSTM model in Tesseract. Focus will also be on training the models to more accurately detect and recognise text which is located in close proximity to non-text elements, to improve the overall results in digitising text from complex EDs.

## Acknowledgment

The authors would like to thank the Data Lab Innovation Centre in Scotland, the Oil and Gas Innovation Centre (OGIC) and DNV·GL for supporting this work. The authors would also like to thank DeepMiner for supporting the first author.

## References

[1] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *CoRR*, vol. abs/1811.04256, 2018. [Online]. Available: http://arxiv.org/abs/1811.04256

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

[3] D. Dori and Y. Velkovitch, "Segmentation and Recognition of Dimensioning Text from Engineering Drawings," *Computer Vision and Image Understanding*, vol. 69, no. 2, pp. 196–201, 1998. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314297905853

[4] R. Cao and C. L. Tan, "Text/Graphics Separation in Maps," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2002, pp. 167–177.

[5] H. Wang and Z. Zhang, "Text detection algorithm based on improved yolov3," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, July 2019, pp. 147–150.

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[7] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector," *CoRR*, vol. abs/1704.03155, 2017. [Online]. Available: http://arxiv.org/abs/1704.03155

[8] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," *CoRR*, vol. abs/1609.03605, 2016. [Online]. Available: http://arxiv.org/abs/1609.03605

[9] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," *CoRR*, vol. abs/1611.06779, 2016. [Online]. Available: http://arxiv.org/abs/1611.06779

[10] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *CoRR*, vol. abs/1801.02765, 2018. [Online]. Available: http://arxiv.org/abs/1801.02765

[11] M. Busta, L. Neumann, and J. Matas, "Fastext: Efficient unconstrained scene text detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1206–1214.

[12] 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.

[13] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, July 2015.

[14] J. Ravagli, Z. Ziran, and S. Marinai, "Text recognition and classification in floor plan images," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 1, Sep. 2019, pp. 1–6.

[15] C. F. Moreno-García, E. Elyan, and C. Jayne, "New trends on digitisation of complex engineering drawings," *Neural Computing and Applications*, vol. 31, no. 6, pp. 1695–1712, 2019.

[16] T. Lu, C. L. Tai, F. Su, and S. Cai, "A new recognition model for electronic architectural drawings," *Computer-Aided Design*, vol. 37, no. 10, pp. 1053–1069, 2005.

[17] C. L. 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

[18] 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.

[19] S. Das, P. Banerjee, B. Seraogi, H. Majumder, S. Mukkamala, R. Roy, and B. B. Chaudhuri, "Hand-written and machine-printed text classification in architecture, engineering construction documents," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug 2018, pp. 546–551.

[20] R. Cao and C. L. Tan, "Separation of Touching Text from Graphics," in *Graphics Recognition Methods and Applications (GREC)*, 2001, pp. 5–8.

[21] A. Sinha, J. Bayer, and S. S. Bukhari, "Table localization and field value extraction in piping and instrumentation diagram images," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 1, Sep. 2019, pp. 26–31.

[22] A. Brock, T. Lim, J. Ritchie, and N. Weston, "Convnet-based optical recognition for engineering drawings," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 01, 2017.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2013. [Online]. Available: http://arxiv.org/abs/1312.6229

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 91–99. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969239.2969250

[25] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[26] E. Eman, S. S. Bukhari, M. Jenckel, and A. Dengel, "Cursive script textline image transformation for improving ocr accuracy," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 5, Sep. 2019, pp. 59–64.

[27] A. Veit, T. Matera, L. Neumann, J. E. S. Matas, and S. J. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," *ArXiv*, vol. abs/1601.07140, 2016.

[28] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1156–1160.

[29] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1083–1090.

[30] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, jan 2016.

[31] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4159–4167.

[32] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "Densebox: Unifying landmark localization with end to end object detection," 2015.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735