Project Proposal

ABSTRACT

One of the most common forms of representing engineering schematics are Piping and Instrumentation Diagrams (P&IDs), which describe the process flow along with interconnected process equipment and instrumentation. These P&IDs are generated manually, using specific software, thus requiring significant time and effort to generate. Moreover, the diagrams themselves do not provide easy reference to the different components of the schematics which can be used to analyze correctness of the system designed. In this study, we built a system which takes input from the user in the form of a user drawn P&ID, extracts gesture-based features from the drawing, and uses the features to classify P&ID symbols. Subsequently, the system derives association rules between symbols i.e., which component is connected to which component and finally, the system outputs a standardized drawing identical in form to the original hand-drawn drawing, but with standardized element symbols and lines. This way, a user could draft P&IDs by drawing them on a touch interface instead of drafting on CAD software and generate relational tables. This would significantly reduce the time and effort required to design P&IDs and verify their correctness as diagrams can be drafted and relationships between different process equipment can be verified on the get go. Moreover, this system would serve as an appropriate testing tool for new personnel who are required to have process flow knowledge as they can be tested to draw segments of P&IDs and the hierarchy table generated against their drawing can be compared to baseline hierarchy table to determine the correctness of their drawn schematic and hence, their process flow knowledge. However, gauging the effectiveness of this application is beyond the scope of this study. The system was evaluated using two methods; precision and recall of recognized symbols and lines from sketches and accuracy, precision and recall of association of process equipment with process lines and other process equipment. The precision and recall for symbol classification was found to be 0.988 both, whereas for association of process equipment, accuracy of 93.3%, a precision of 93.33% and a recall of 100%.

INTRODUCTION

Piping and Instrumentation Diagrams (P&IDs) are the gold standard of describing process flow information in any manufacturing operation involving in line flow of process chemicals for e.g. petrochemicals, oil refineries, cement and fossil fuel driven power plants. During the design of new plants or modification of existing plants, extensive effort goes into review of changes to P&IDs, with numerous hand drawn drafts used in conjunction with separate equipment hierarchy tables to ascertain the correctness of design. After this stage, the finalized draft is sent for drafting on CAD, requiring specialized expertise and extra man hours, which are a valuable in typically constricted project timelines. A sample P&ID is shown in Figure 1.

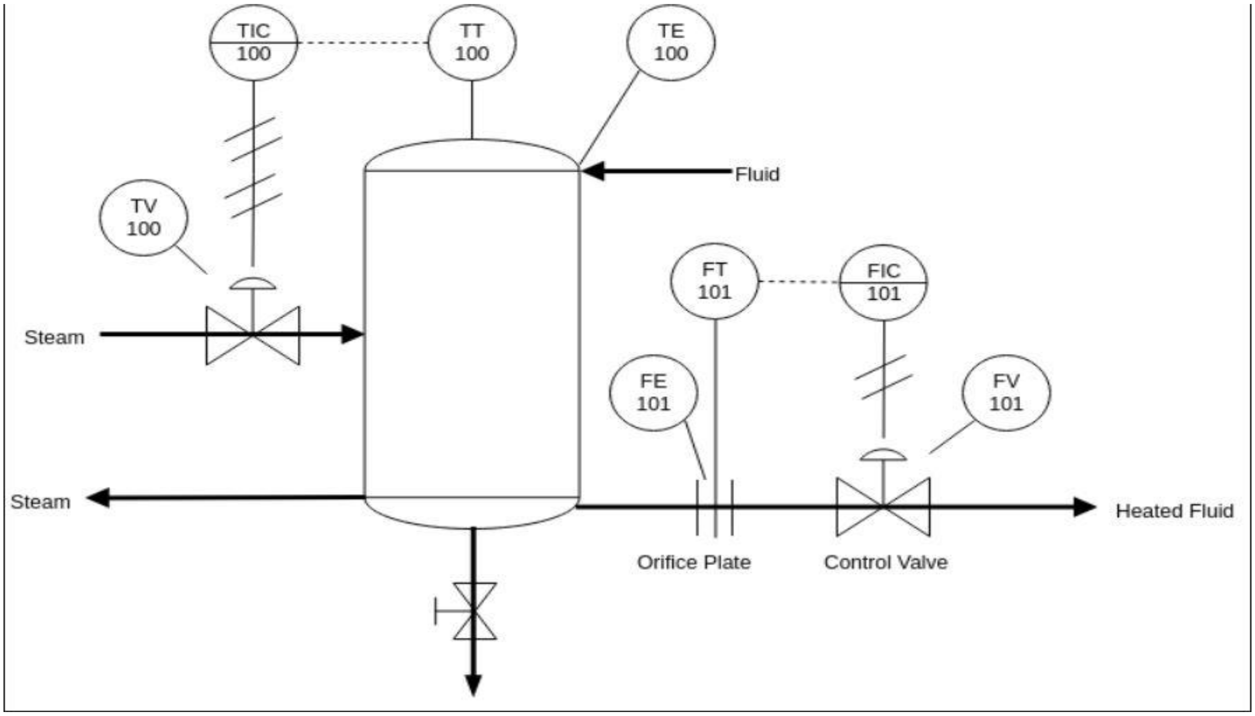


Figure 1

The procedure described above has 3 steps: (1) Sketching of P&IDs, (2) Generation of equipment relational tables and (3) Drafting of finalized draft on CAD. A solution that would translate hand-drawn diagrams into digitized diagrams and generate relational tables on the basis of these drawings would simplify the procedure as (1) it would allow more natural interaction than traditional mouse and palette tool (Hse H 1999), (2) It would eliminate the need to enter the same information twice and (3) Would eliminate the need to create relational tables describing equipment relationship with lines and other equipment. With significant strides already made utilizing various deep learning approaches to generate relational tables (Rohit Rahul 2019), the focus of our study is not exploration of ways to generate relational tables.

The solution, in the simplest of terms, aims to get a sketched P&ID and convert it into a digital drawing like Figure 1. The challenges associated with generating such a data are numerous and varying in their degree of complexity. The study explores both gesture-based and vision-based approaches for different application areas in generating a solution. Recognition of any hand-drawn diagram for its translation to digital files with standardized symbols and lines requires the detection and recognition of symbols, determination of relationship between interconnected lines and equipment and differentiating between process and electronic connection lines. These areas are the focus of our study as addressing these three basic challenges in recognition of a hand-drawn P&ID will streamline further development on the topic and will provide reasonable ground to generate an effective solution.

RELATED WORK

Significant strides have been made in the area of recognizing engineering drawings to hasten the process of digitization of engineering drawings. Various studies have been conducted delving into different challenges while recognizing P&ID symbols and lines. (Eun-seop Yu 2019) Describes the recognition of P&ID symbols in image format using deep learning model. Although a welcome development, the study only focuses on a set number of symbols and since it involves deep learning models, requires hefty amounts of image data to train. Moreover, heuristics are ignored in generation of a suitable model in this study which leaves room to be explored. (Rohit Rahul 2019) Does a better job at dividing the issue at hand into basic tasks and includes heuristics. However, this approach falls short of differentiating between continuous and dashed lines. With a 65% accuracy of pipeline detection, the approach presents a solution that has a significant room for improvement. (Eyad Elyan 2018) Provides a novel approach to differentiate between different sets of symbols in a P&ID by applying k-means clustering and then passing the preprocessed data with class differentiation into a convolutional neural network. However, this approach again relies on printed P&IDs with standardized drawing, hence having low variation in entropy for similar symbols, which can cause it to fall short in recognition of hand-drawn P&IDs. (Elena Rica 2020) Explores the area of reducing human intervention to validate P&IDs and proposes the identification of erroneous components in a P&ID. The study can offer valuable insights into the areas of a P&ID that are prone to error-some classification, therefore cannot be validated. However, it offers little beyond that and is of secondary importance in our study of recognition of hand-drawn P&IDs. (Sung-O Kang 2019) Follows the path of recognition and generating equipment association rules. Although our study can be linked with generation of association rules to expand on its application, the focus is solely on recognition of hand-drawn P&IDs, therefore the second part of this approach is not of utility to us. The recognition approach provides valuable insights into text recognition and text association with lines and equipment. However, the text recognition is based on standardized sizes and shapes of English alphabets and numerals, leaving room for incorporation of handwritten text recognition in engineering drawings.

Although not directly linked with recognition of engineering drawings, recognition of hand-drawn sketches and their classification into text and symbols in other application areas can prove valuable in recognition of various symbols in P&ID. (P. T. Tracy Hammond 2010) Presents with a unique approach to recognize hand-drawn shapes and text. The approach in this paper uses a variety of methods to recognize various characteristics occurring in a military course of action diagrams and builds a text recognition scheme in conjunction with shapes, recognition of shapes composed of multiple shapes, recognition of lines by breaking them down into line segments and differentiation between dashed and continuous lines, all useful while recognizing P&ID symbols and lines. The approach in this paper uses a variety of methods to recognize various characteristics occurring in a military course of action diagrams and builds a text recognition scheme in conjunction with shapes, recognition of shapes composed of multiple shapes, recognition of lines by breaking them down into line segments and differentiation between dashed and continuous lines, all useful while recognizing P&ID symbols and lines. Moreover, (R. D. Tracy Hammond 2005) provides a baseline language to recognize and beautify hand-drawn shapes and lines with contextual information. The in-built functionalities in the system can be incorporated for P&ID recognition but the system does not have a set functionality to detect and beautify hand-drawn P&IDs.

METHODOLOGY

P&IDs are specialized drawings consisting of different symbols to denote different components, along with different line notations used to communicate different connections. The basic components in a P&ID are control valves, manual valves, field instrumentation, process lines and electrical lines denoting an electrical connection between a field instrument and control valve. The symbols with their description are presented in Figure *2*.

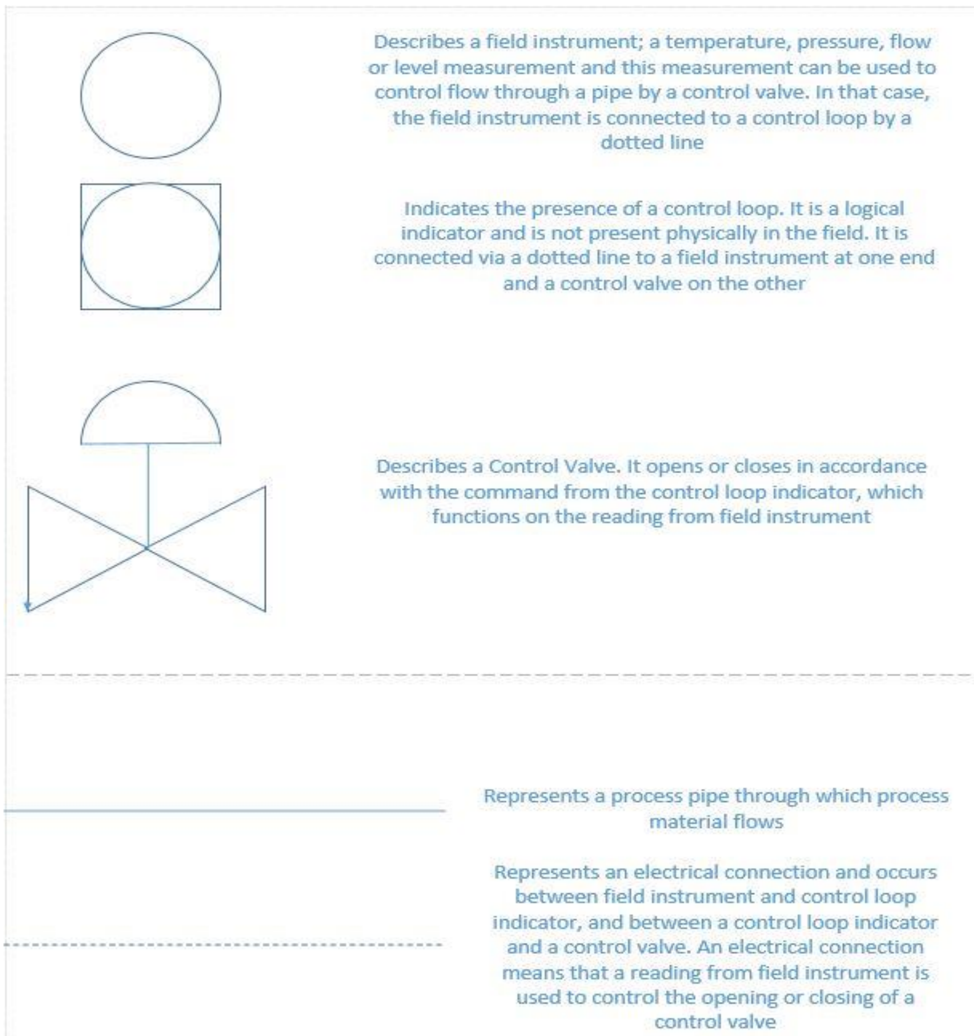


Figure 2

A control loop on the P&ID is denoted by a field instrument, mounted on a line and a control valve mounted on the same or even a different line, with both the instrument and control valve electrically connected to a control loop indicator. This would be displayed as a circle mounted on a line, connected to a square-circle via dashed line, which is then connected to a control valve via dashed line. A sample control loop is displayed in Figure 3.

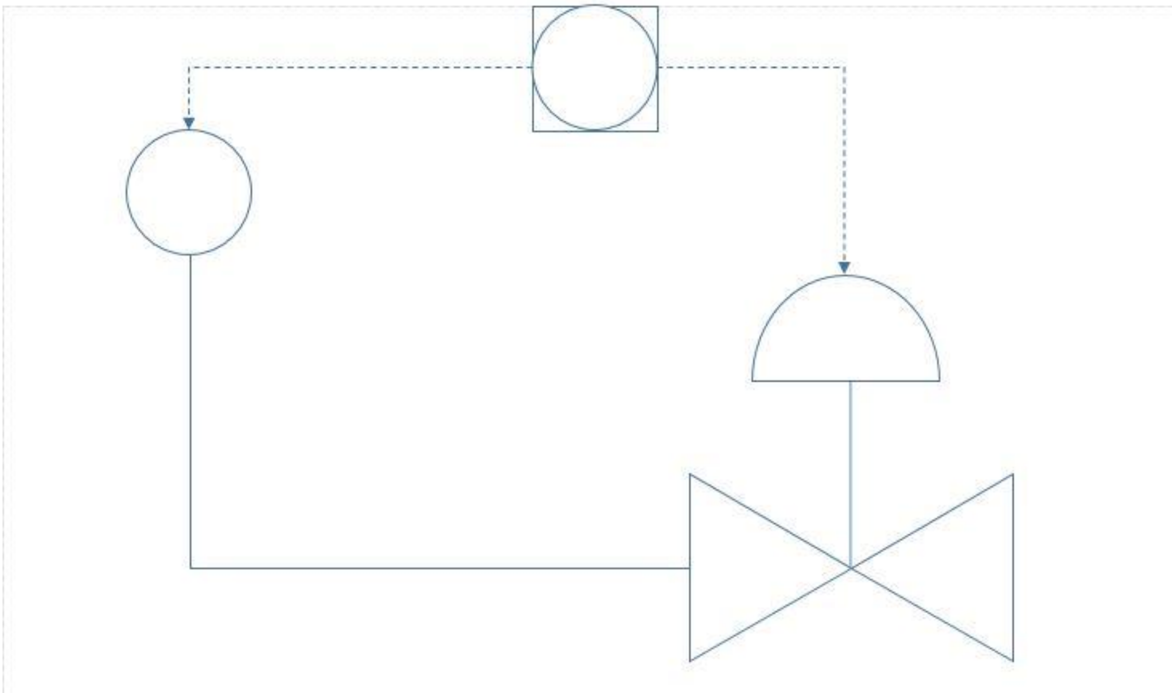


Figure 3

To extract information from a hand-drawn P&ID, the study required resolution of various challenges; differentiation between symbols, association of equipment with process lines without any pen overlap and differentiation between continuous and dotted lines. These challenges form the basis of our research questions that were addressed in this study.

Data Collection

The first step in the study was to design an interface on which drawings could be made and the position and chronological information of the stroke can be captured. For this, we designed a web application with Javascript frontend to record stroke coordinates with an option to download an image of the stroke for vision applications and the other option to download the (x,y,time) coordinates for gesture-based feature analysis. The interface used can support multi-stroke drawings and only stops recording features when the download button is pressed. The interface is hosted via Github page along with stylus support, making it portable on mobile devices.

Data preprocessing

The CSV raw data file consists of sampled points under short intervals, each row containing the positional and chronological information obtained from the sketch interface. The input CSV file is then being read and parsed into a numpy array. The strokes are segmented by filtering and grouping the strokes by their difference in time interval. Through observation, a threshold of 500ms is selected. The stroke segments are separated using the starting indices obtained. For normalization, the first and last three points are removed from each stroke. Using the raw stroke segments, Rubine feature 1-13 are collected and parsed into a dataframe. A typical P&ID input graph ranged from 10-20 strokes.

The next step was to implement a solution to test our research questions. The remaining solution is exclusively implemented in python and numpy and pandas are used throughout for data manipulation. The first research question is *How effective are gesture-based features in recognition of different P&ID symbols.* Using the Rubine Features (Rubine 1991), we calculated the density metrics of strokes as described by Long (A. Chris Long 2000) to classify between line and non-line segments. Note that line segments include both continuous and dashed line segments.

For classification of non-line segments, we implement a Random Forest Classifier (Tin Kam Ho 1995) in Python using Scikit-Learn. Our train-test data set contained a total of 80 hand-drawn symbols across 4 classes: Circle, Square, Valve shape and Semi-circle. The model was trained with a train-test split of 75-25. Using pickle, the Random Forest Classifier model was saved to be used in our main program. The model was then used in the main program to classify non-line segments into respective classes.

After the non-line segments are classified into respective classes, we find if there is a control loop indicator in the diagram. To do that, we need to find if there is a circle within a square. We solve this problem by Euclidean distances between square and circle segments. If the minimum and maximum x and y coordinates of two segments differ by less than a threshold, we can conclude that the 2 shapes describe a control loop indicator.

The second research question is *How effective are distance and context information in association of equipment with process lines?* Once we have classified the strokes to an extent, association is to be found. Association describes which segment is connected to which segment and defines the relationships between P&ID lines and equipment or between equipment. As an example, consider Figure 4.

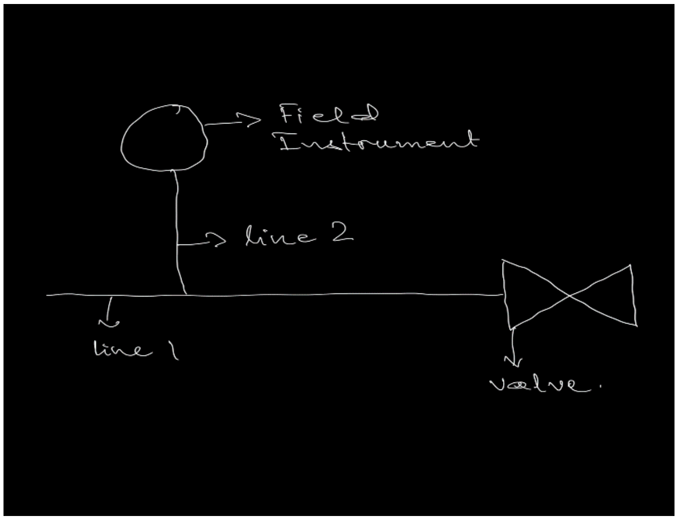


Figure 4

The association rules of the drawing in figure 4 are presented in Table 1

Stroke	Associated Strokes
Line 1	Line 2, Valve
Line 2	Line 1, Field Instrument
Field Instrument	Line 2
Valve	Line 1

Table 1

The intuition behind generating association rules is that strokes connected to one another will have either 0 or a minimal distance below a certain threshold between them. Utilizing this intuition, we utilize single link Euclidean distance (K. K. Mohbey 2013) (Mohammed J. Zaki 2020) which calculates the minimum distance between two strokes according to the formula below:

$$\delta(C_i, C_j) = \min\{\delta(x,y) \mid x \in C_i, y \in C_j\}$$

The resultant association rules are stored in a python dictionary, with each stroke as a key and its connected strokes represented by values against the key.

Still requiring the differentiation of continuous and dashed line system, the third research question is *How effective are vision-based techniques in differentiating between continuous lines and dotted lines?*

Our implementation utilizes a set of algorithms from the Python opencv2 module to provide a vision-based approach to the problem. cv.Canny is used for image data preprocessing and to edge detection. Hough Line Transform (cv.HoughLinesP) detects points that form a line.

We observe that the Hough Line Transform often yields multiple lines on a single stroke. Therefore, we developed a filter to remove duplicate lines detected. We first sort the xy coordinates of the endpoints. Lines that have endpoint euclidean distance under a certain threshold are merged by averaging the coordinates of endpoints. From observation, a threshold value of 50 pixels is selected. The resulting line is highlighted and displayed to the user.

Besides utilizing vision-based techniques in differentiating between hand-drawn continuous and dotted lines, gesture-based features also provide valuable intuition to solve the issue of differentiation. The basic difference between continuous and dashed lines is that dashed lines are segmented and have shorter segments describing them. Moreover, in P&IDs, dashed lines occur only between a field instrument and a control loop indicator, and a control loop indicator and a control valve. We inculcate these rules and length thresholds to overcome this challenge.

Graph Amalgamation

The Schmedraw library is utilized for symbol drawing, with a predefined set of custom P&ID symbols. With the association rules, a network structure of symbols can be constructed, with each node containing information about their type and neighbors. The drawing module starts from an arbitrary node and iterates the symbols in a depth-first search manner. Upon each iteration, the symbol drawing is appended to the current endpoint, and the algorithm updates the endpoint accordingly. To maintain a cyclic structure, the last symbol draws wires back to the starting position. The final figure is saved and downloaded for the user. A sample output of the drawing module is shown below.

Results

Since the first research question deals with a classification problem, we will use evaluation metrics suitable for a classification (Hossin 2015) i.e., precision, recall and F1 score. We first started by training a Random Forest Classifier over hand-drawn non-line objects described in the methodology section and the confusion matrix of the classifier over test data is shown in Table 2.

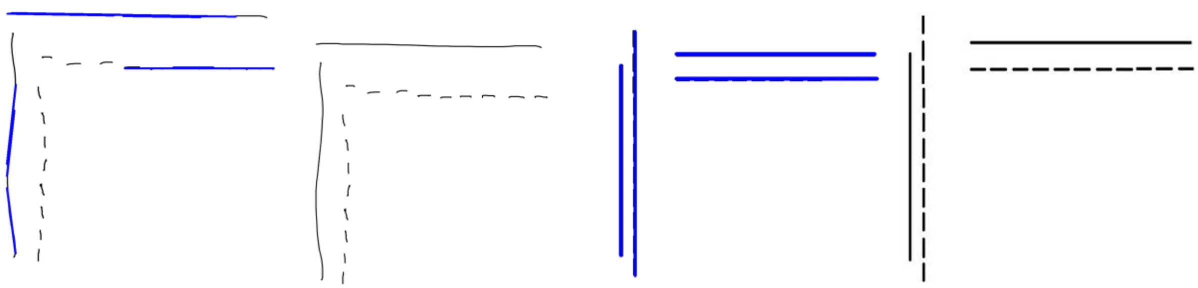
	Circle	Semi-Circle	Square	Valve
Circle	7	0	0	0
Semi-Circle	0	4	0	0
Square	0	0	5	0
Valve	0	0	0	5

Table 2

The subsequent precision recall of the Classifier are calculated as 1 and the F1 score is also 1.

For association of segments with their connected segments, our evaluation methodology is to treat every segment as a class and the subsequent rightly connected segments as true positives. Any segment which is not connected to another segment but shows as connected is classified as a false positive. A segment which is connected but system classifies it as unconnected is a false negative. Using these extrapolations onto association rules evaluation gave an accuracy of 87%, a precision of 92.9% and a recall of 92.9%, which yields an F-1 score of 92.9%.

For differentiating between continuous and dashed-line segments, we evaluate the performance of the vision-based classification module based on two sets of data. The first dataset contains standardized lines collected from drawing utility, while the second dataset contains hand-drawn images collected from the sketching interface. Both dataset contains different images of vertical and horizontal lines at different position and orientation.

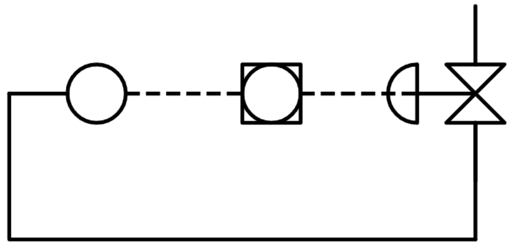


The Canny+Hough Line transforms yield a high accuracy on detecting the standardized strokes, while lacking performance on sketch data. We believe this is because the Hough Line transform algorithm is unable to detect lines that are not perfectly straight. It is also missing the order information from strokes, making the algorithm unable to detect strokes that are positionally close to each other. Dedicated preprocessing and fine-tuning methods are needed to achieve a higher accuracy.

Tackling the issue by gesture-based features proved much easier as connection between lines is very easy to detect due to time feature and distance of stroke. We achieved an accuracy of 100% in this regard.

Graph amalgamation results

To test on the robustness of the module, we sketched by hand various different figures as input, and with a stylus drawn diagram with minimal noise, got the following output for a control loop:



DISCUSSION

The results of object detection based on gesture-based features were encouraging and had high success rates. In the initial stages of the study, while not eliminating the starting and ending points of a single stroke, the system contained significant noise which reduced the performance of the system so the extreme points were eliminated so the true essence of every stroke is captured. The methodology implemented did not have any errors while distinguishing between line and non-line segments and classified non-line objects with a high accuracy rate. This reveals that the approach we used is a good starting step in the direction of recognizing hand-drawn P&IDs. In case of single stroke, the system correctly identifies objects drawn in different styles, for example a valve drawn from the top or bottom. However, the methodology for classification and differentiation is tight-jacketed and allows for single stroke objects, which may serve as an impediment in dealing with different ways of drawing the same shape with multiple strokes.

Similarly, the generation of association rules also generated positive results. This was possible after defining some rules while dealing with association, which significantly improved the performance of the system. While at first, complete link hierarchical clustering was used to generate association rules via clustering, the error was significant in that approach as the distances between segments are calculated between the centroid of the segments. Before inculcating P&ID specific rules, complete link clustering was approached to eliminate false positives, but thresholding was extremely varied for the system and hence a particular threshold value could not be set to eliminate errors, resorting towards explicit rule definition. The performance of the system can be improved by inculcating more rules relating to the P&IDs as the scope of the application is broadened. However, the threshold defined in the system is static and may cause faulty classification when working with different systems with different screen resolutions.

Differentiating between continuous and dashed lines via vision methods proved to be a challenging task. The first approach which was eventually discarded was to detect continuous and detected lines through Shannon entropy, but this approach also ended up detecting non-line objects which was not the requirement for answering our research question. Moving on to the approach of CannyP edge detection and HoughLinesP point detection that form a line, the system performed well on the straight line, whether dotted or continuous but failed to properly detect hand-drawn lines which was the object of our study. The reason for this can be understood by the structure of a hand-drawn line, which is never straight and includes waves. The vision-based system infers each straight segment between waves as a single segment and thus, detects a single line as an amalgamation of multiple segments and in case of a line break, does not even process on further line. This results in:

1. Incomplete recognition of line

2. Broken recognition of line

The aforementioned impacts defeat the purpose of what we are trying to achieve through line recognition and differentiation, thus this approach is not suitable for our study. However, gesture-based features, owing to an amalgamation of tracking time which tells when a line is drawn and the coordinates which are spaced closely, along with the multiple rubine angle features which provide additional context about a line's orientation.

FUTURE WORK

The scope of this application is vast. P&ID contains much more symbols than the ones mentioned in this study but these symbols are less frequent than the aforementioned symbols. Work can be expanded to include more symbols in the present system to increase the range of the system. Moreover, work could be expanded to include multiple strokes for a single object and then classifying that object as one. A fundamental part of any P&ID diagram is the text that describes the symbols i.e., whether a field instrument measures pressure, flow, level or temperature. A valuable direction from this study would be to incorporate text separation from symbols and lines and associating the text with the correct symbol and equipment which would drive the system closer to Industrial Acceptability. This study had no design of recognizing Tanks and Vessels, and the subsequent considerations while associating lines with Tanks and Vessels as the lines in such a situation are close by and different rules may have to be applied on components in vicinity of tanks and vessels, or an intelligent algorithm which adjusts threshold according to concentration of equipment in vicinity could also be introduced which would be a pertinent contribution to the subject matter. Although not directly related to P&IDs, the amalgamation of concepts in this study could also be used for digitizing or expanding the realm of sketch recognition to other engineering drawings for example Process Flow Diagrams, Instrumentation Loop Drawings and Mechanical Drawings for Process Industry.

CONCLUSION

In this paper, we presented a method for recognizing common P&ID symbols in a handwritten format and their association with other symbols and lines. We achieved an accuracy of 100% in classifying non-line objects, a precision and recall of 92.9% in generating association rules and a 100% accuracy in differentiating between continuous and dotted lines. Although scope is limited, this study is a step towards creating engineering drawings, an easier process, and the scope can be subsequently expanded to realize the full scope of a P&ID.

Bibliography

A. Chris Long, James A. Landy, Lawrence A. Rowe, Joseph Michiels. 2000. "Visual Similarity of pen gestures." *SIGCHI Conference on Human Factors in Computing Systems.* New York: ACM. 360-367.

Elena Rica, Carlos Francisco Moreno-Garcia, Susana Alvarez, Francesc Serratosa. 2020. "Reducing human effort in Engineering Drawing Validation." *Computers in Industry Volume 117.*

Eun-seop Yu, Jae-Min Cha, Taekyong Lee, Jinil Kim and Duhwan Mun. 2019. "Features Recognition from Piping and Instrumentation Diagrams in Image Format using a Deep Learning Network." *MDPI Energies.*

Eyad Elyan, Carlos Moreno Garcia, Chrisina Jane. 2018. "Symbols Classification in Engineering Drawing." *IEEE International Joint Conference on Neural Networks.*

Hossin, Muhammad and Md Nasir Suleman. 2015. "A review on evaluation metrics for data classification evaluations." *International Journal of data mining and knowledge management process 5* 1-5.

Hse H, Shilman M, Newton AR, Landay J. 1999. "Sketch-based user interfaces for collaborative object oriented modelling."

K. K. Mohbey, G. S. Thakur. 2013. "An Experimental survey on Single Linkage Clustering." *International Journal of Computer Applications* 6-11.

Mohammed J. Zaki, Wagner Meira. 2020. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms.* Cambridge University Press.

Rohit Rahul, Shubham Paliwal, Monika Sharma and Lovekesh Vig. 2019. *Automatic Information Extraction from Piping and Instrumentation.* New Delhi, India: TCS Research.

Rubine, Dean. 1991. "Specifying gestures by example." *ACM SIGGRAPH Computer Graphics, Volume 25, Issue 4* 329-337.

Sung-O Kang, Eul-Bum Lee, Hum-Kyung Baek. 2019. "A Digitization and Conversion Tool for Imaged Drawings to Intelligent Piping and." *MDPI Energies Vol 12.*

Tin Kam Ho. 1995. "Random Decision Forests." *3rd International Conference on Document Analysis and Recognition* . Montreal. 278-282.

Tracy Hammond, Paul Teele, Joshua Johnston, Aaron Wolin, Brandon Paulson, Drew Logsdon, Joshua Peschel. 2010. "A Sketch Recognition System for Recognizing Free-Hand Course of Action Diagrams." *Proceedings of the Twenty-Second Conference on Innovative Applications of Artificial Intelligence.* Atlanta, Georgia.

Tracy Hammond, Randall Davis. 2005. "LADDER, a sketching language for user interface developers." *Computers and Graphics.*