Semi-Automatic Digitization of Single-Line Diagrams for Accelerated Deployment of Algorithmic Control on Real-World Power Systems

Rajeev Datta[1], **Thuy-Linh Le**[1], **Elijah Cole**[1], **Lucien Werner**[1],
Adam Wierman[2], **Steven Low**[3]
[1]California Institute of Technology
[1]{rdatta,thuylinh,ecole,lwerner,slow,adamw}@caltech.edu

Abstract

With climate change remaining a monumental challenge for current and future generations, there has been continued interest in novel ways to mitigate carbon emissions. One of the more popular approaches is "green"-ifying electric grids, incorporating renewable energy sources. The inherent volatility of such grids motivates the need for a better understanding of the grid's hierarchy and the ability to simulate complex algorithms on these grids. For such analysis, accurate digital models are necessary. Unfortunately, traditional methods of annotating and, subsequently, digitizing physical single-line diagrams are tedious. Our project aims to combat this time and energy waste with an easy-to-use user interface for annotating and digitizing single-line diagrams using computer vision techniques.

1 Introduction

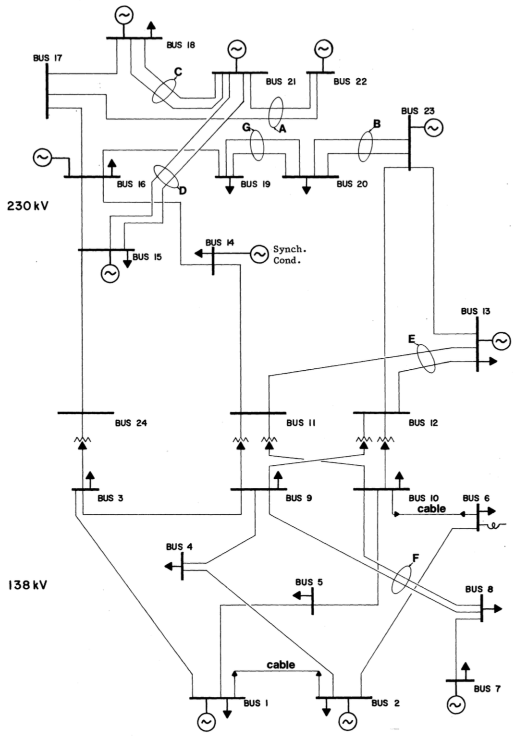
Climate change has loomed like an ever-ticking time bomb and caused widespread global repercussions. In light of climate change's catastrophic consequences, researchers have considered and implemented many approaches to mitigate or adapt to this inevitable crisis, including but not limited to reduction in energy use, innovations in renewable energy generation, and "green" -er grids.

As its name implies, "green" grids aim to steer away from traditional fossil fuel-driven energy generation and towards renewable energy generation (e.g. solar, wind, and hydro). Unfortunately, such grids have a glaring obstacle to widespread adaption: reliability. The highly variable power output of renewable sources, like photovoltaic (PV) systems, means meeting real-time demand becomes increasingly complex. Optimization problems, like the multi-interval economic dispatch problem, represent, to a certain degree, the potential complexity of such dynamics at scale.

Even in simplified convex formulations, the optimization problem requires some form of classical control algorithm to solve for dispatches in an efficient and effective manner. More realistic formulations, however, enforce safety constraints on lines and buses, requiring a detailed network representation of the grid for simulation. Unfortunately, formatting a real-world grid in such a way requires an already digitized, or otherwise accessible, network representation of the grid or reading through physical line diagrams to construct such a representation. The former only a select few major

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

Figure 1: Example of a line diagram [1]



utilities have built, while the latter is an arduous task wasting productivity of countless people in this field.

A possible solution to the current state of this field is an end-to-end pipeline that converts a single line diagram Portable Document Format (PDF) into a format readily available to a suite of relevant simulations. Luc and Thuy-Linh, with their ongoing work as a part of Professor Steven Low's DigitalTwin project (a project aiming to digitize the Caltech grid in an accessible manner), have developed an interface for visualizing digitized networks once annotated and constructed from single-line diagrams. This project worked to complete the workflow by minimizing the effort needed to extract the relevant information from single line diagrams.

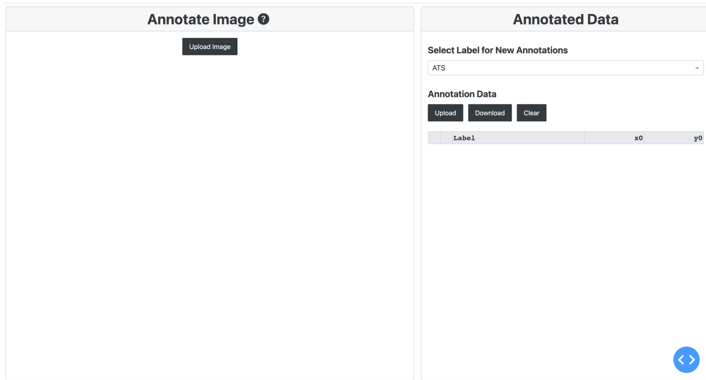
The project took the first step towards such digitization and tackled the annotation of relevant atomic symbols in diagrams. Our findings showed the feasibility of a semi-automatic process for annotation of different symbols in diagrams and highlighted a few shortcomings for future investigation.

2 Methodology

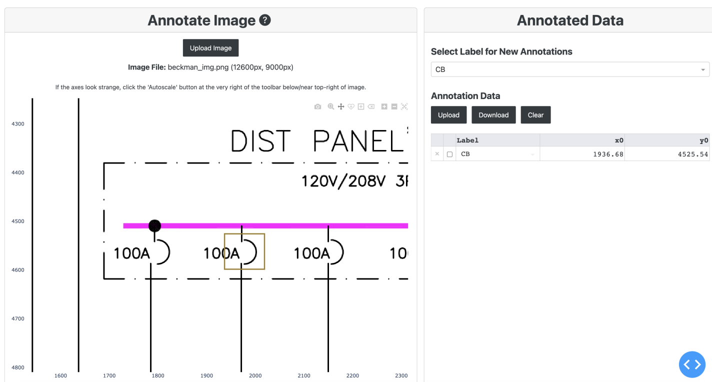
To simplify the process of annotating single line diagrams, we have developed a pipeline consisting of an intuitive web interface, a dataset of annotated single line diagrams, and a semi-automatic predictive model for symbol annotation.

This section will introduce these components and outline the high level implementation details.

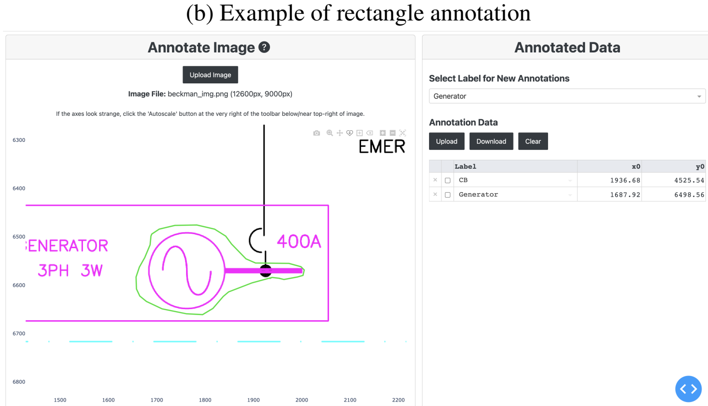
2.1 Easy-to-Use Web Interface



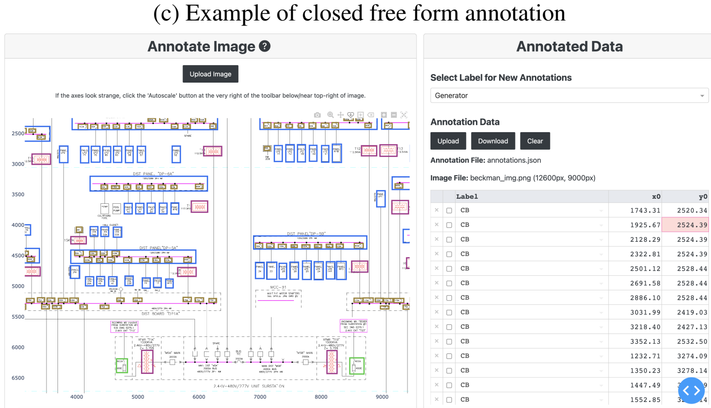
(a) Home page



(b) Example of rectangle annotation



(c) Example of closed free form annotation



(d) Example of an annotated diagram

Figure 2: Easy-to-use web interface for annotating single-line diagrams.

Thuy-Linh developed this web interface to create a one-stop shop for annotating symbols on single line diagrams. The only required input to the interface is a single line diagram PDF in portable network graphics (PNG) format. Users upload suitable diagrams on the left-hand side of the interface utilizing the "upload image" button and navigating to the desired image file (Figure 2a).

Users can then select one of the common atomic symbols from the drop down menu and start annotating the image using the interface's built-in graphical tools. The website supports two types of annotation shapes: rectangle (Figure 2b) and closed free form shapes (Figure 2c). To aid in the annotation process, the website also includes zooming in and out, paning, autoscaling, erasing active shape, and saving the current annotated image to PNG format. A finished example for a set of common symbols is found in Figure 2d.

The website stores annotations in a json file containing the points of interest for the various annotation shapes (e.g. top left and bottom right corners for rectangular annotations) and their respective labels.

Users can upload and download json files of the correct format to the interface, allowing practitioners to save progress. Users also have the ability to start over using the clear button on the right hand side of the interface.

As for implementation, the website was developed using the python packages Dash, Plotly, and Flask. The frontend/UI of the application used Bootstrap, allowing for mobile-friendly formatting.

2.2 Dataset

The dataset consists of ten single line diagram images taken from a larger dataset of 19 single line diagrams. All diagrams were collected as a part of the DigitalTwin project and represent different buildings within California Institute of Technology's campus (Figure 3). Each raw diagram image was constructed by converting a corresponding PDF into a 200 dots per inch (dpi) PNG image.

The above procedure provided the raw PNG diagram images for annotation. To create the corresponding annotated "ground truth" images, we had to determine the symbols to annotate. We chose the following atomic elements due to their frequency and/or relevancy to the grid's overall functionality: circuit breakers, fused switches, contactors, multi-position switches, generators, panels, and transformers. We then use the web interface outlined in the previous section to annotate all ten images for the seven symbols of interest.

2.3 Semi-Automatic Algorithm

To aid symbol annotation, we developed a semi-automatic pipeline capable of prediction generation and evaluation. The algorithm requires a user to annotate a single instance of a symbol in question using the rectangle drawing tool. Given the example image of the symbol, the algorithm uses matched filtering to check the image for similar symbols.

Generating potential matches requires three inputs: a single line diagram image, a template image representing a bounding box around the symbol of interest, and a confidence threshold for how well a candidate match must resemble the template. Using the given inputs, an initial set of possible matches are generated using OpenCV's template matching algorithm under the normalized correlation coefficient. Formally, the normalized correlation coefficient is specified as

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y') * I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 * \sum_{x',y'} I'(x+x',y+y')^2}}$$

where

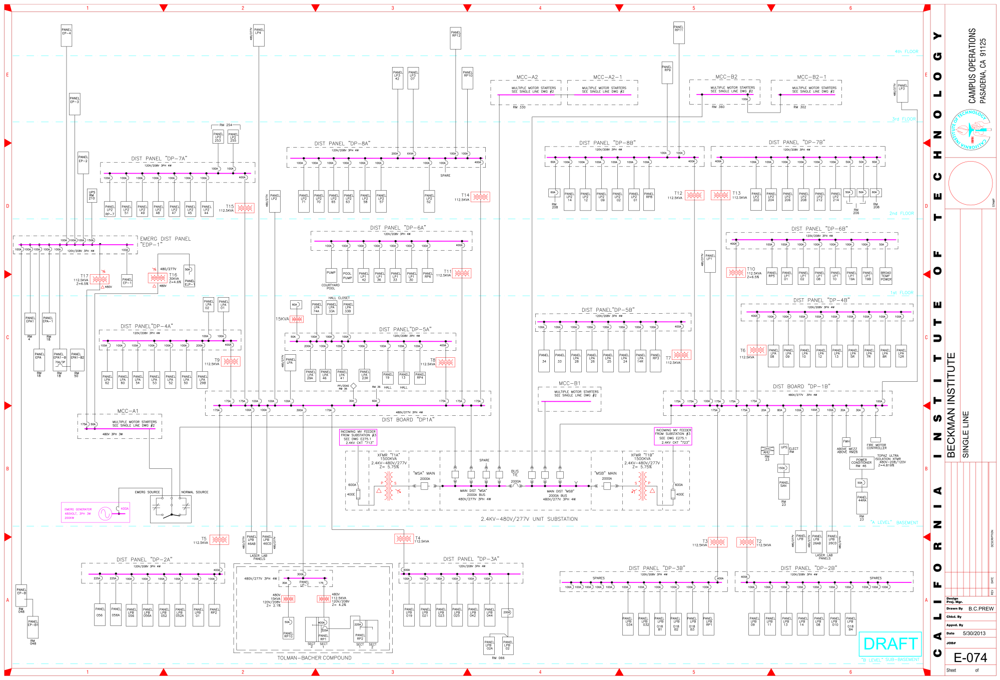


Figure 3: Example of a raw diagram image in PNG format. This image was the single-line diagram of the Beckman institute on California Institute of Technology's campus.

$$T'(x', y') = T(x', y') - \frac{1}{w * h} * \sum_{x'', y''} T(x'', y''),$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{w * h} * \sum_{x'', y''} I(x + x'', y + y''),$$

and T, I, w, h are the template image, original image, width of template, and height of template respectively.

The matches are then filtered down according to the image dimensions, removing any nonsensical candidates at the edges of the image. With remaining candidates, we apply an optimized version of non-max suppression (NMS) based on Malisiewicz's method that starts with highest confidence bounding boxes and removes any overlapping matches. The output is a set of distinct bounding boxes in the desired json format, readily available for visualization.

The prediction evaluation and optimization pipeline also requires three inputs: a single line diagram image, a template image of the symbol of interest, and a json file of the diagram's ground truth. Once again, the template and diagram image are passed into the template matching algorithm and the resulting heat map is normalized. All possible bounding boxes are kept except for those outside the bounds of the image. Then, to save computation time, the bottom 90% of the matches according to the confidence score in the heatmap are removed. The remaining matches are filtered further via the previously mentioned NMS algorithm. The top 1000 distinct bounding boxes are passed into a final evaluation step where a precision-recall curve is constructed and an optimal confidence threshold is determined according to F_1 scores. The final output of the pipeline is the corresponding distinct bounding boxes satisfying the optimal threshold in json format and the Precision-Recall curve data.

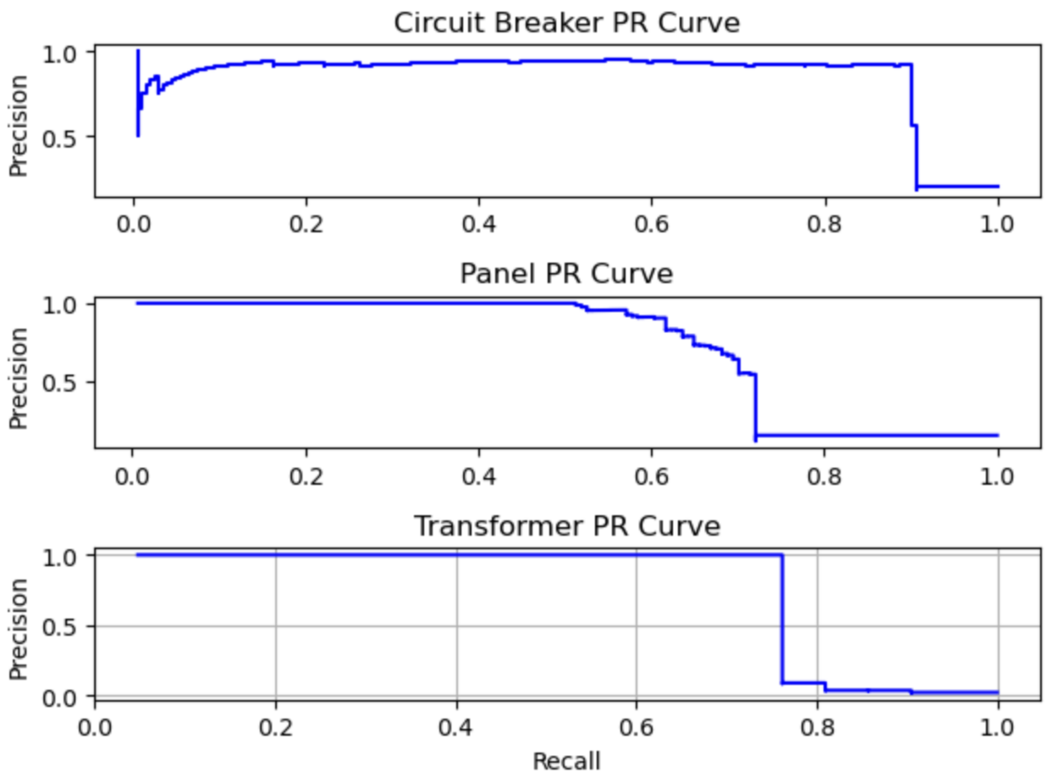


Figure 4: Precision-recall (PR) curves outputted by evaluation pipeline for, going top to bottom, circuit breakers, panels, and transformers on the Beckman Institute single-line diagram.

3 Experiments

Using the evaluation pipeline outlined before, we compared the performance of the matched filtering algorithm on three different common symbols for the Beckman Institute single-line diagram (Figure 4). From the curves, we notice, for all of the symbols, the precision and recall is steady for a large portion of the confidence threshold's available range, and then suddenly the precision drops as the confidence threshold becomes sufficiently large. We conclude that this behavior most likely implies, at high enough confidence thresholds, most of the correct bounding boxes stop being detected, proving the importance of precisely selecting thresholds to the predictive annotation problem.

To better understand the shortcomings of our current pipeline, we analyzed the different output json files and extracted high confidence (relative to the overall distribution of confidence scores from the matched filtering operation) false positives (Figure 5). From the example, we can notice our current algorithm has a difficult time differentiating parallel branches in the grid and a rectangular panel. We believe this behavior motivates the need for a more customized metric in the confidence scoring that incorporates heuristics like having all sides bounded with a black line.

The pipeline, however, captures accurately labels matches to a given template symbol. One such example of the algorithm's promising performance is on the detection of circuit breakers in the Beckman Institute diagram (Figure 6). From the annotated diagram, we can note the high degree of accuracy displayed, representing the viability of this approach to the annotation problem.

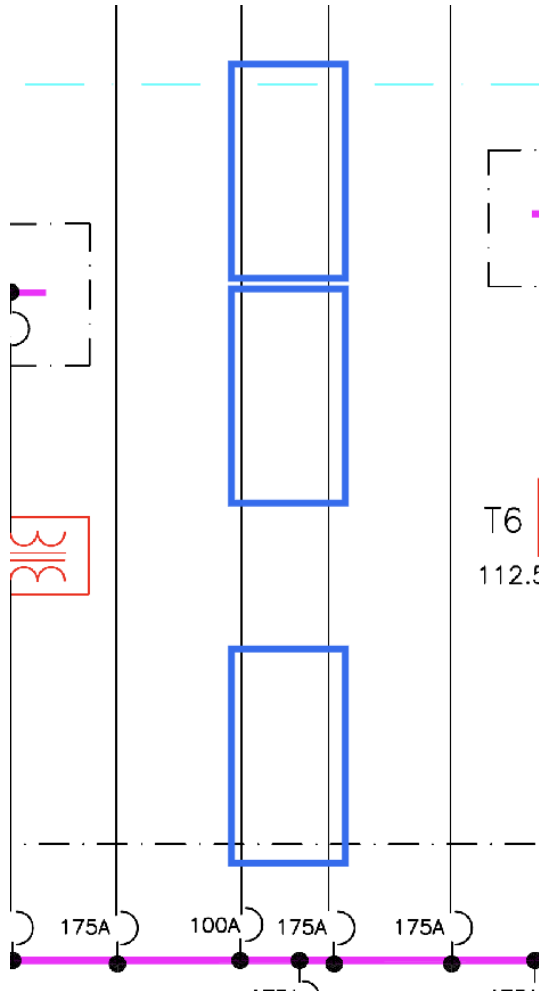


Figure 5: Examples of false positives in the annotation of panels in the Beckman Institute single-line diagram.

4 Discussion and Conclusion

Our experiments, shown in the previous section, demonstrate the performance of the classical computer vision technique, matched filtering, on a novel use case: annotating symbols in single-line diagrams. At high level, the performance of matched filtering provides a promising picture for future exploration (Figure 6). However, a closer look exposes certain shortcomings of the technique. For example, in the panel annotation case, we noticed the filtering is sensitive to close but not quite right matches to the template image (Figure 5). We also noticed shortcomings inherent with the traditional matched filtering process. Specifically, the algorithm is not scale nor rotation invariant which means rotations and different sizes of the symbol in question will not be found unless a corresponding template image is selected. Such annotation requires more human intervention in the predictive model, leaving room for further improvement.

5 Future Directions

Our future plans for this project can be divided into two time scales: near-term and long-term. For the near-term goals, we would like to polish the matched-filtering-based prediction pipeline and incorporate the initial pass directly into the web interface's backend. To accomplish this, we need to first construct a more robust confidence metric for proposed bounding boxes using known priors in

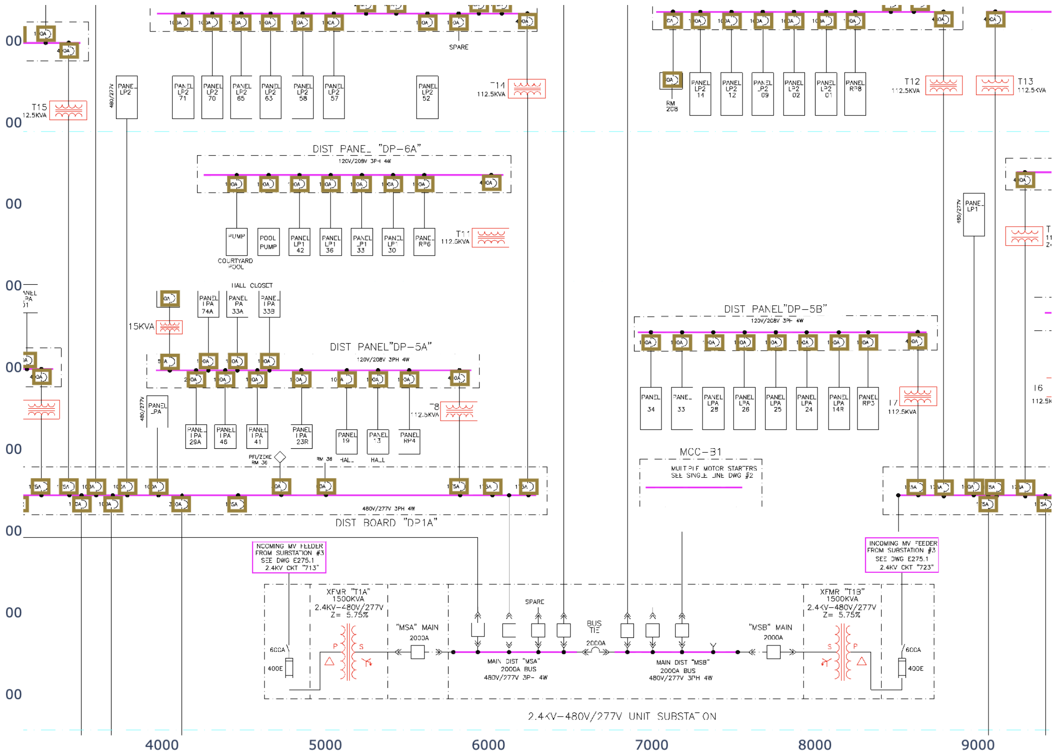


Figure 6: Example of the annotation of circuit breakers from our proposed pipeline on the Beckman Institute single-line diagram.

the structure of the problem (e.g. differentiating between two parallel lines and an actual rectangular panel in the diagram). Another potential room for improvement is improving the efficiency of the non-max suppression (NMS) algorithm. Although the current vectorized version of the traditional NMS algorithm is able to parse through the over one hundred of million potential bounding boxes in a typical single-line diagram image within the time scale of hours instead of days, it is still to inefficient to reliably allow practitioners to not remove low confidence boxes prior to using the algorithm, reducing the overall robustness of our proposed pipeline. We could also explore and analyze the instances of high confidence false positives and potential false negatives in other symbols and diagrams to ensure performance translates as expected across different templates and images.

Once we have developed a basic workflow based on matched filtering, we could study the benefits of replacing matched filtering with more modern object detection algorithms like Faster R-CNN [2] or YOLOv3 [3]. We could use our annotated dataset for training and evaluation. One particularly interesting machine learning question is whether generic symbol detection is possible (as opposed to detecting different types of symbols), in analogy work on class-agnostic object detection [4]. If annotating symbols seems to work reliably to the degree necessary for practitioners in the field, we could begin exploring using computer vision techniques to detect and segment connectivity in diagrams. We could even explore incorporating meta data in the web interface and corresponding text detection models time permitting.

Acknowledgments and Disclosure of Funding

We would like to acknowledge Lucien Werner and Elijah Cole for their feedback and input throughout the course of this project. Without their specialized expertise in the fields of digitizing electric grid diagrams and computer vision respectively, this project could not have happened. We also appreciate the advice and guidance of Professor Adam Wierman and the structure granted by the CS 145 course.

References

[1] Probability Methods Subcommittee. Ieee reliability test system. *IEEE Transactions on Power Apparatus and Systems*, PAS-98(6):2047–2054, 1979. doi: 10.1109/TPAS.1979.319398.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[3] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[4] Muhammad Maaz, Hanoona Rasheed, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Ming-Hsuan Yang. Class-agnostic object detection with multi-modal transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 512–531. Springer, 2022.