ECE 4424 Final Project Report

John Fuerlinger
ECE Machine Learning Student
Virginia Tech
Blacksburg, VA 24060
john98@vt.edu

Michelle Fike
ECE Machine Learning Student
Virginia Tech
Blacksburg, VA 24060
michfike@vt.edu

Abstract

Our goal for this project was to perform a comprehensive review of papers in an area pertaining to our ECE major. We chose to immerse ourselves in papers related to using machine learning for detecting PCB defects. After identifying common problems different researchers faced, we decided to focus on the most common issues. Some of the issues we found in our research include the lack of publicly available datasets for defective PCBs as well as the tedious acquisition process for new datasets. Another interesting issue we identified in our research was the small size of the PCB defects and their tendency to be overlooked by classical machine learning models.

1 Introduction

PCBs are the building blocks of modern electronics; their failure can range from an inconvenience to a large expense. For many years, visual inspection was the only way to validate PCB's. This means of verification is labor intensive and not reliable.

Automated Optical Inspection (AOI) is now considered the most reliable means of inspection. Early AOI relied on referential-methods. In 2011, Chauhan and Bhardwaj introduced the idea of using a pixel-by-pixel comparison to detect defects on PCBs [1]. In this method, two images of PCB's are used. One of these images is the "reference" image, which is used to compare and contrast with other images to identify defects. Both images are also thresholded and converted to binary images before being XOR'd together in a pixel-by-pixel fashion. The result from this operation is essentially the "difference" between the two images, and because we inherently trust the reference image, we can assume any difference we find is some sort of PCB defect. This method had its limitations. There were many defects that were not revealed in this method and it required an identically oriented and sized image for comparison.

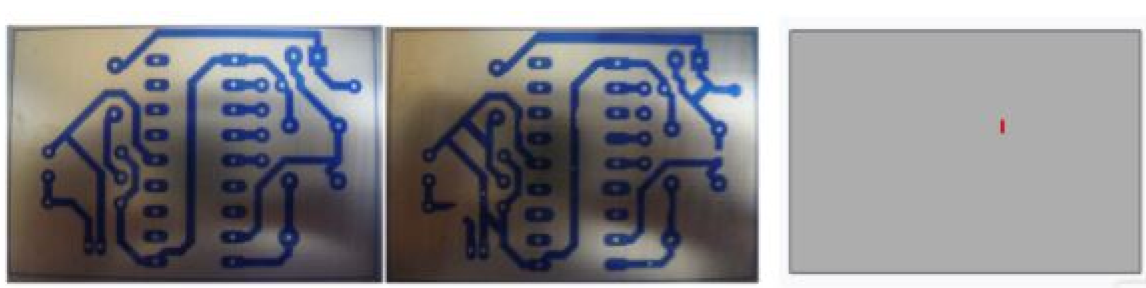

Figure 1. A template PCB image and a defective PCB image[2]. And the resulting XOR of the two images.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

As machine learning has progressed, its reliance on a large number of images has been a hurdle for PCB inspection. A major difficulty in gaining PCB data is rooted in the limited number of non-defective PCB images available. Compounding the issue is the even smaller number of defective PCB images [3]. As machine learning techniques have progressed, it is evident that the techniques are not a universal fit. Some techniques work better for certain problems. For the problem of detecting PCB defects, some classical machine learning approaches were not an adequate solution due to the relatively small size of the defects compared to the image.

Image detection makes up a large portion of machine learning's problem space, so we chose papers that delved into solutions that could be applied to areas beyond PCBs.

2 Resolving the Issue of a Low Quantity of Images

In [4], the method Huang and Wei use for image procurement was a major element in their strategy for PCB defect detection. Due to the lack of public images of defective PCB's, they needed to compile their own dataset. In creating this dataset, Huang and Wei took pictures of non-defective PCB's using a 16-megapixel industrial camera equipped with two frosted LED rings to reduce uneven illumination and specular reflection of the board. Then, they cropped each image and edited them in Adobe Photoshop to create artificial defects and superimpose them on the images. Lastly, they decided to convert the images to purely black and white to simplify the image processing techniques required.

In Figure 2, normal solders and defective solders are shown. It is evident from these images how Huang and Wei may have created these artificial defects to propagate their dataset.

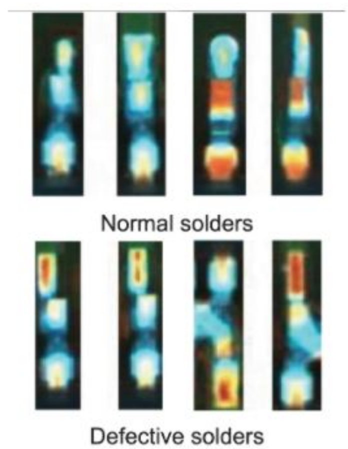

Figure 2. Solder images on PCBs [3].

Two potential pitfalls in Huang and Wei's strategy for image procurement are that they artificially created the PCB defects, as well as converted the images to binary images. By reducing the image quality, they have potentially lost vital information in the images. This may have compounding effects when paired with images of truly defective PCB's, where the exact visual impact of each defect is unknown and potentially variable.

Augmenting the dataset helped the teams to avoid the potential trap of overfitting. It was also employed by Ding, Dai, Li and Liu [5] who added Gaussian noise, changed the lighting and rotated, shifted, and flipped images. They also noted the hazard of manipulating images, which could result in an unintentional negation of the defect itself.

3 Resolving the Lack of Defective Images

The above referential method which compares a PCB to a template is labor intensive, requires clean image sets, and is subject to false-defect detections from reflections and shadows. Researchers have proven Convolutional Neural Networks (CNNs) capable of detecting defects using less images.

Reference images are also not required in the CNN approach, thus reducing the labor on several fronts.

In their paper, Zhang, Shi, Li, Zhang and Liu [6] further reduced the need for images by using the transfer learning method. This model is the use of a pre-trained neural network. In this study, the pre-trained network was created using ImageNet. Often used with language processing and computer vision, this method expedites the process. CNNs work by the lower-level convolution layers extracting general information and higher-level convolution layers extracting the more intricate features of the image. These computationally efficient layers act as a filter. The fully connected layers of a CNN involve more computation, but do the work of capturing information to aid in classification.

The authors used VGG-16, which is a pre-trained network trained in the ImageNet Large Scale Visual Recognition Competition. It uses five convolution layer stacks and small filters with a pixel stride of 1. This model also used a max-pooling layer with a 2x2 window size. Using the pre-trained network saved the authors several days of training and limited their workload to fine-tuning the network. To use this network, the images needed to be resized to 224 x 224. The convolution layers were frozen and the authors used them to extract the features. They fine-tuned the network by replacing the fully connected layers with a dense layer which trained its weights using back propagation.

The authors determined that their methods were able to detect the PCB flaws without the issue of shadows and reflections causing false-detections. CNNs have markedly reduced the number of images needed and the labor needed to obtain those images. The authors also deemed their method to have the potential to be implemented in more varied settings than their predecessors. The use of CNNs in detecting faults continues to be researched [7].

4 Resolving the Relative Size Issue

In looking for defects, the size of the defect on a PCB relative to the PCB image adds a dimension to the challenge. Because the defect is relatively small, classical CNN models are prone to overlooking especially small defects. TDD-net [5] addresses this. The authors begin with the Faster R-CNN. Faster R-CNN is known for its success in finding small features, making it a good starting point for detecting the tiny defects on a PCB. Faster R-CNN uses a large amount of computation power, but, for now that seems unavoidable when searching for relatively small images. One of the paper's novel approaches is using K-means clustering to set more appropriate anchor weights for the training data. The standard K-means being:

$$d(box, centroid) = 1 - IntersectinOfUnion(box, centroid)$$

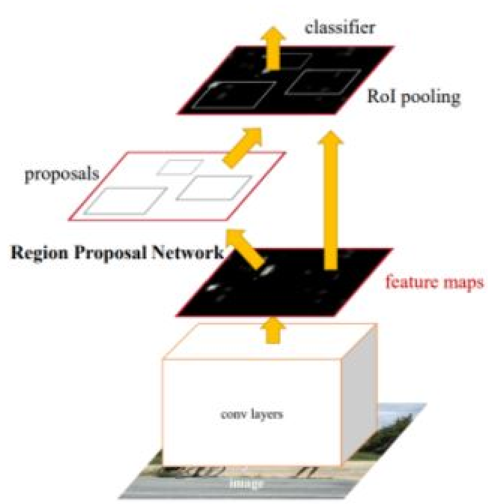

Figure 3. An overview of Faster-RCNN [8].

Faster R-CNN uses a Region Proposal Network (RPN) which generates object bounds and scores at each position; as the sliding window makes its way over the image, these proposals are generated and presented to the network as areas of interest.[8] Figure 3 helps us to visualize this process.

Each convolution layer produces a different view based on the resolution of that layer. High-resolution layers offer the fine details of the image while low-resolution layers offer more information about the general semantics. Figure 4a illustrates this. Since the defects are small, the high-resolution layer is important. But, as the image works its way through the network, the details can get lost. The authors' second novel implementation is how TDD-net concatenates these layers by upsampling so that details of the image can be considered at the same time as the semantics of the image, as seen in Figure 4b.

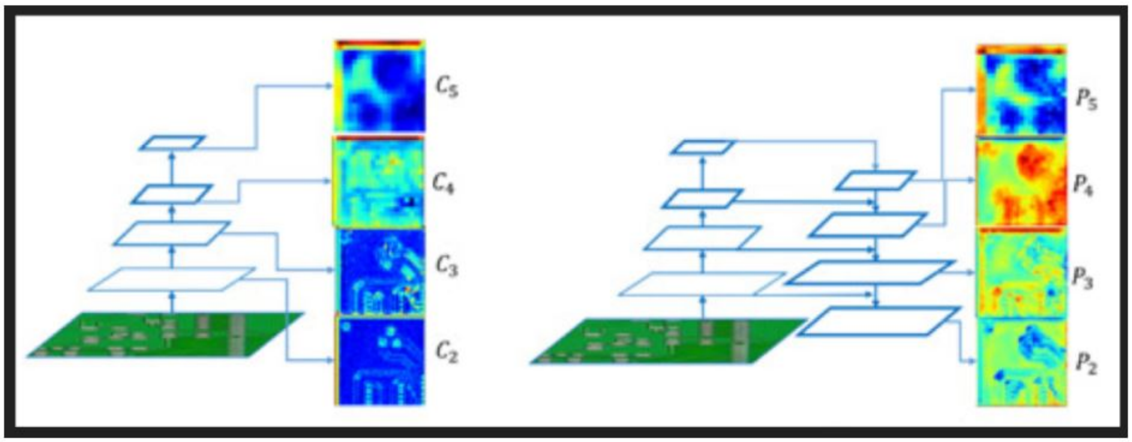

Figure 4a. Proposals for different convolution layers. Figure 4b. TDD-net concatenated proposals.[6]

P_k is calculated using:

$$P_k = \|k_0 + log_2(\frac{RegionOfInterestwidth \cdot RegionOfInterestheight}{224})\|$$

The authors found that use of appropriate anchors for small features resulted in greater success in locating defects. Also, their system for upsampling and concatenating feature layers allowed for better feature extraction.

5 Conclusion

In reviewing journal articles spanning a decade on machine learning and PCB defect detection, we recognize that this is an evolving journey, each step making its unique contribution. A solution that began as a simple XOR function has developed into a solution whose inner-workings is not fully understood. First, the ideas for augmenting images were successful and continue to be used. Next, the research using CNNs for PCB faults allowed for more agile uses of defect detection; research is ongoing. Finally, the TDD-net's use of appropriate anchor weights for small features resulted in greater success in locating the defects. The upsampling and concatenating of feature layers allowed for better feature extraction. The steps in PCB detection and machine learning that have made the largest strides are the steps that involve a collaborative effort, whether it be the sharing of images or sharing of pre-trained CNN networks. The community effort has revealed the altruistic vision and humble efforts of many scientists.

References

[1] Pal, Ajay & Chauhan, Singh & Bhardwaj, Sharat. (2011). Detection of Bare PCB Defects by Image Subtraction Method using Machine Vision. Proceedings of the World Congress on Engineering 2011, WCE 2011. 2.

[2] I. Ibrahim, S. Bakar, M. Mokji, J. Mukred, Z. Yusof, Z. Ibrahim, K. Khalil, M. Mohamad, “A printed circuit board inspection system with defect classification capability,” International Journal of Innovative Management, Information and Production, vol. 3, no. 1, pp. 82-87, 2012.

[3] I. Volkau, A. Mujeeb, D. Wenting, E. Marius and S. Alexei, "Detection Defect in Printed Circuit Boards using Unsupervised Feature Extraction Upon Transfer Learning," 2019 International Conference on Cyberworlds (CW), Kyoto, Japan, 2019, pp. 101-108, doi: 10.1109/CW.2019.00025.

[4] W. B. Huang and P. Wei, “A PCB dataset for defects detection and classification,” Comput. Vis. Pattern Recognit., vol. 14, no. 8, pp. 1–9, 2018.

[5] R. Ding, L. Dai, G. Li and H. Liu, "TDD-net: a tiny defect detection network for printed circuit boards," in CAAI Transactions on Intelligence Technology, vol. 4, no. 2, pp. 110-116, 6 2019, doi: 10.1049/trit.2019.0019.

[6] C. Zhang, W. Shi, X. Li, H. Zhang and H. Liu, "Improved bare PCB defect detection approach based on deep feature learning," in The Journal of Engineering, vol. 2018, no. 16, pp. 1415-1420, 11 2018, doi: 10.1049/joe.2018.8275.

[7] Hu, Bing, and Jianhui Wang. “Detection of Pcb Surface Defects with Improved Faster-Rcnn and Feature Pyramid Network.” Ieee Access 8 (2020): 108335–45.

[8] L. Hulstraet, A Beginners Guide to Object Detection, DataCamp, April 19, 2018. Accessed on December 2, 2020 [online]. Available: https://www.datacamp.com/community/tutorials/object- detection-guide.