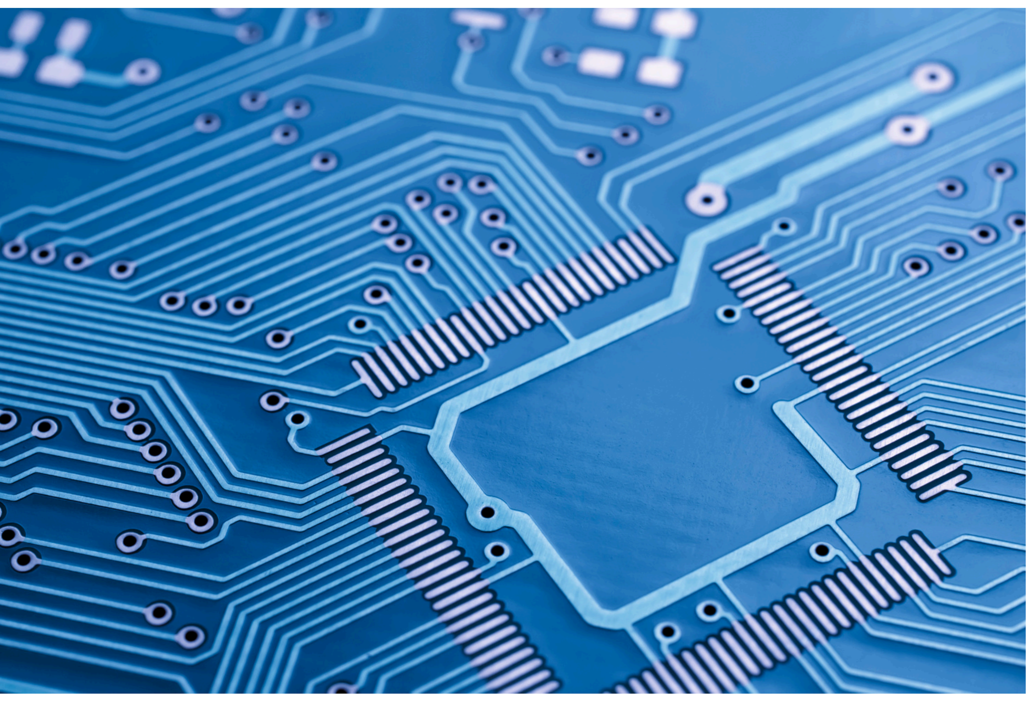
AUTHORS

Anika Mahesh, Navya Tiwari

Using object classification to find defects in PCB boards

To what extent can we predict the location and class of defects in PCB boards using the YOLOv8 model?

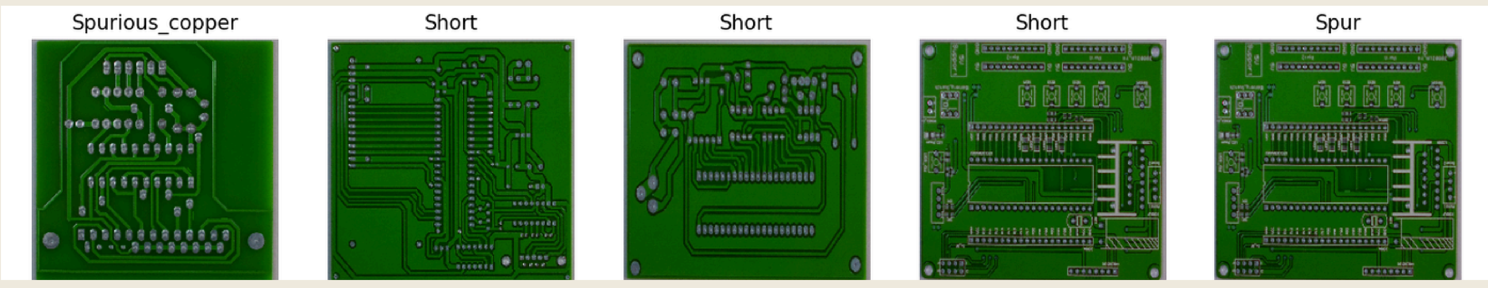


ABSTRACT

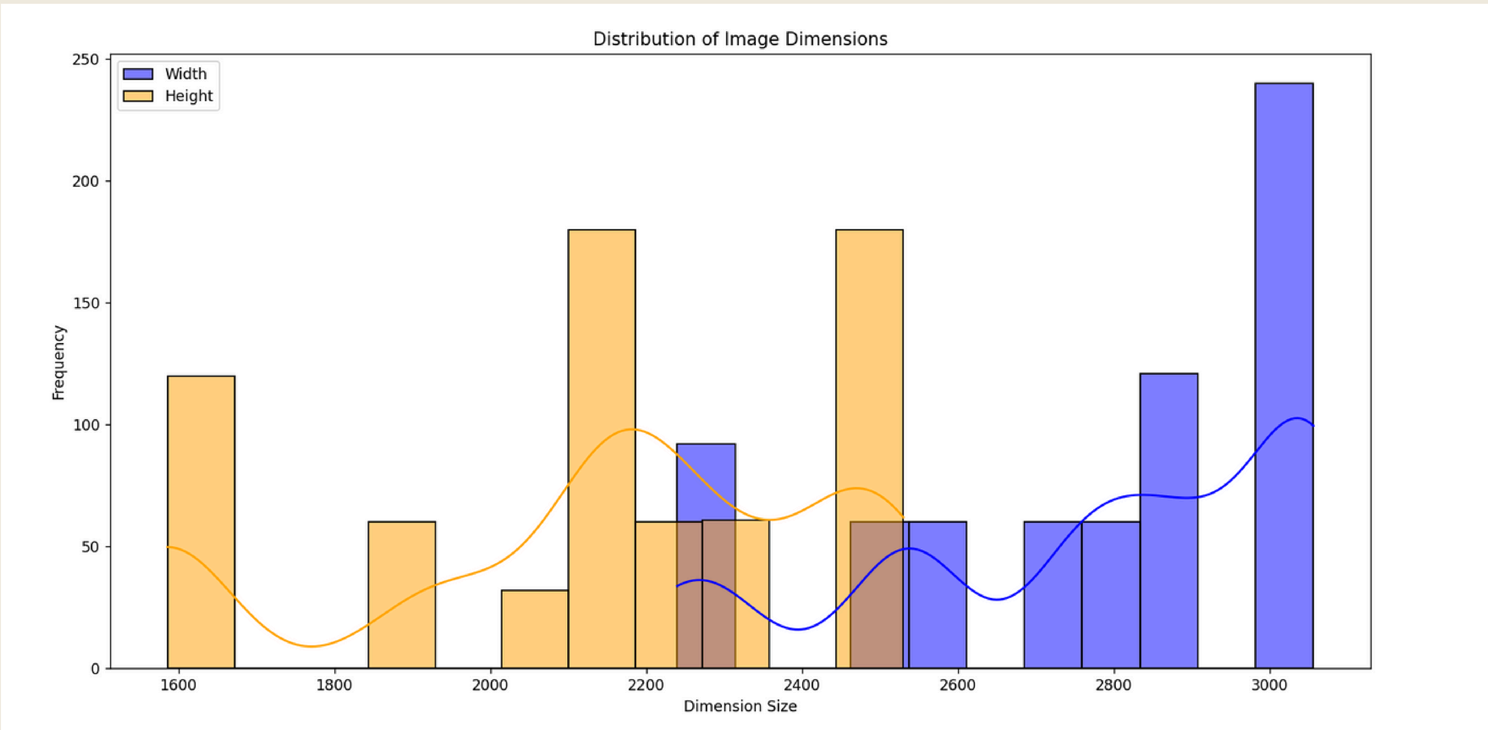
Motivated by our experience in working with PCB fabrication errors on project teams, we decided to work on a project on identifying and locating these errors on PCB boards. We decided to train an YOLOv8 object detection algorithm to locate and identify errors on a dataset containing 6 types of PCB errors and 693 images. Our goal is to establish a robust framework capable of identifying six classes of defects effectively.

01. Data Investigation and Augmentation

Our dataset contains 6 classes of PCB errors

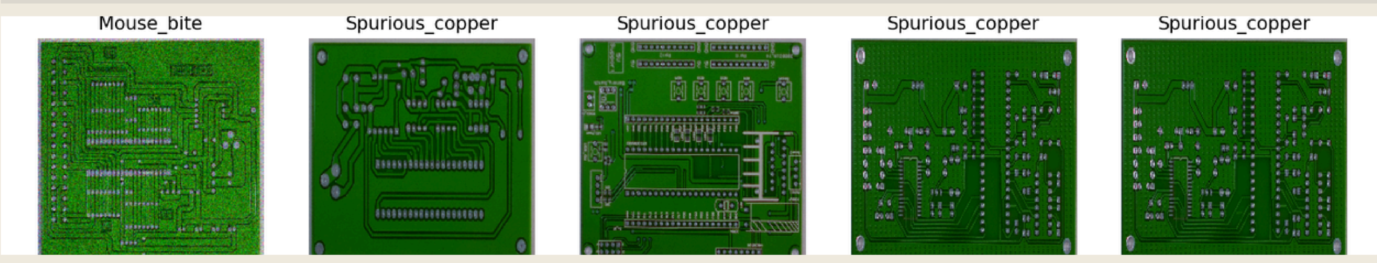


The following is a visualization of how our data is spread

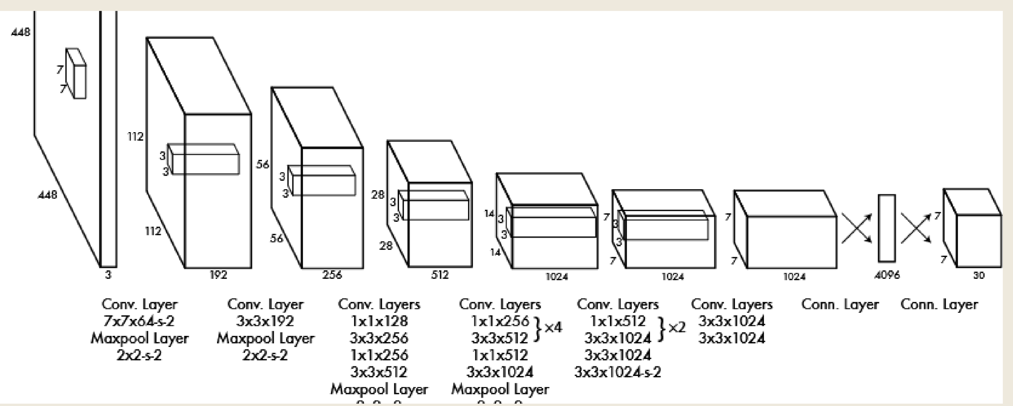


02. Data Augmentation

To improve variability of dataset, we added random Gaussian noise, gamma correction, orientation and brightness. To standardize the dataset algorithm resized images to have a resolution of 448 by 448



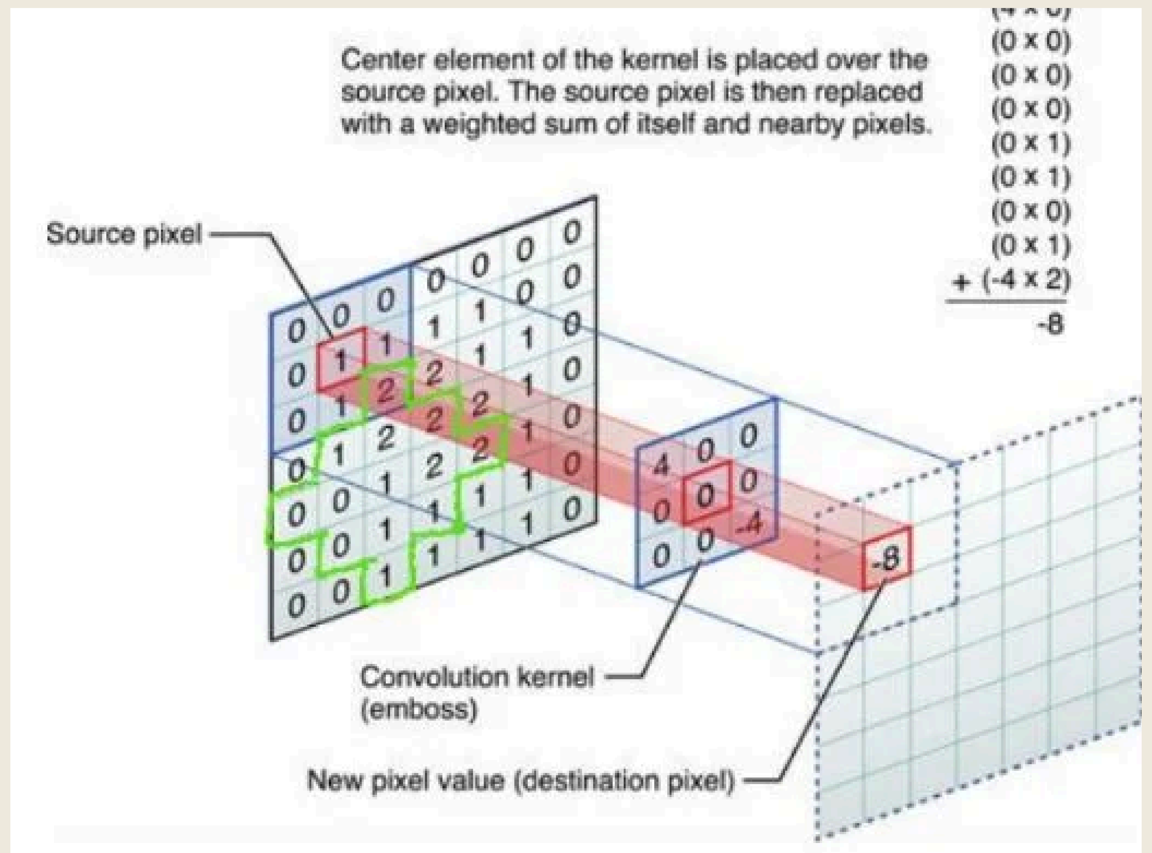
03. Object Detection with YOLOv8



- YOLOv8 is a neural network built to spatially separate bounding boxes and predict class probabilities in a single network quickly
- Resizes images to 448 to 448 and runs 1 convolutional network on image
- Architecture consists of 24 convolutional layers followed by 2 fully connected layers
- Network results in 7x7x30 tensor containing confidence predictions for each class and bounding box data such as center and dimensions.

03. Understanding Convolution

Convolutional layers can be understood as a matrix operation of a nxn matrix kernel on an image. Kernel iterates over image and kernel is multiplied with every possible nxn grid that can be made in the image. Every value in the kernel is a tunable weight calculated in the training process. A convolutional layer can contain multiple kernels which increases the depth of the output. Usually followed by a maxpooling layer that creates a smaller matrix based on the maximum value in a nxn region of the image.

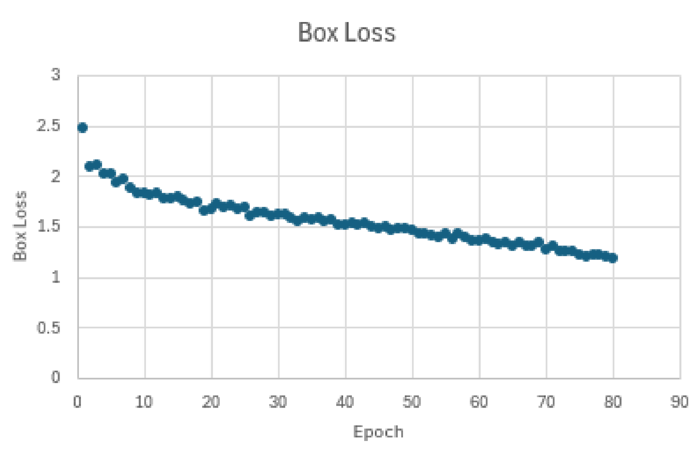


How kernel is applied to an image to extract features

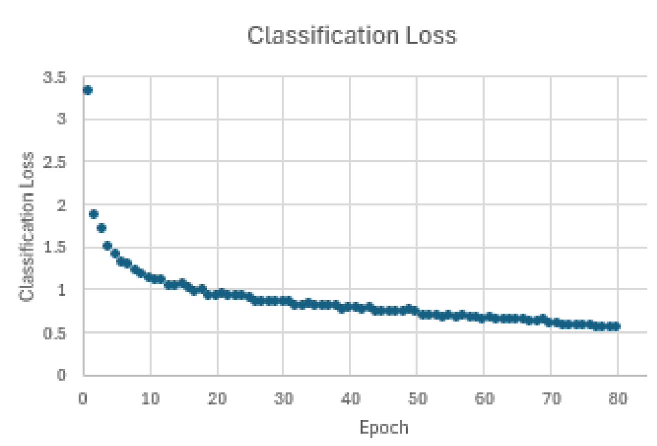
05. Model Training

We trained our model for 80 epochs to minimize the following losses:

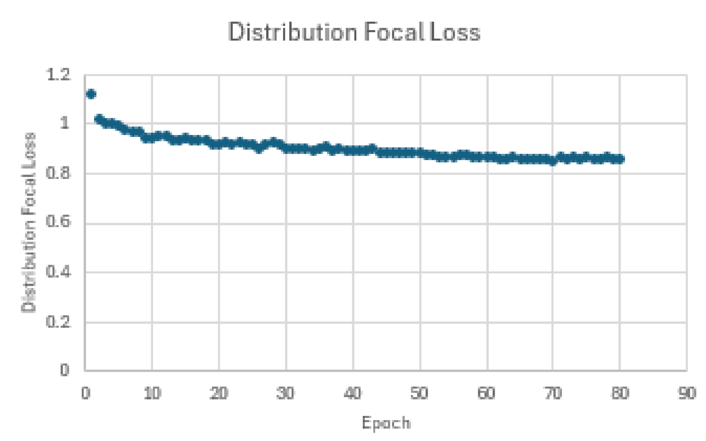
1. box loss : The loss related to bounding box regression. It measures how well the model predicts the position and size of bounding boxes for objects in an image.
2. classification loss : Measures how well the model classifies objects into the correct categories
3. distribution focal loss: used for refining bounding box regression with high precision. It ensures the predicted boxes align accurately with the ground truth.



Box loss during training of 80 epochs. The graph goes from exponential to linear and has not flattened out.



Classification Loss training for 80 epochs

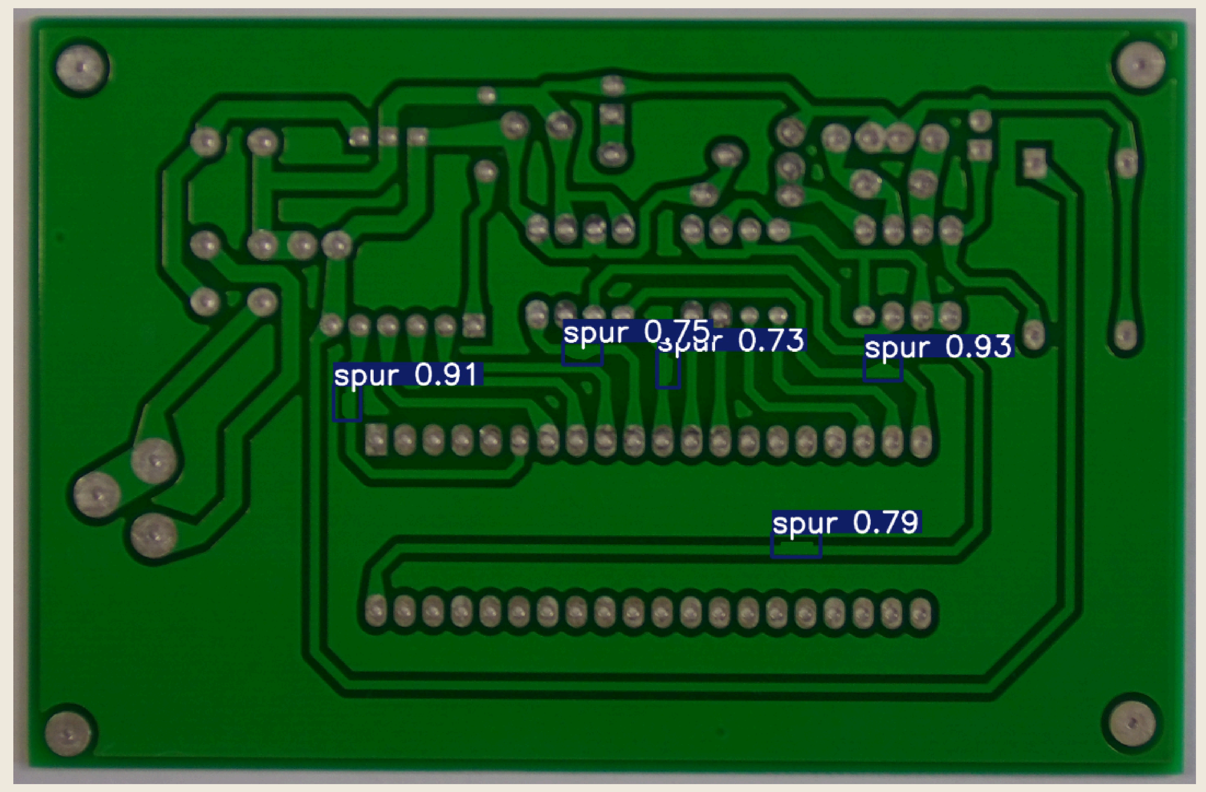


DFL Loss training for 80 epochs

06. Limitations

- Some classes are underrepresented in the data
- The dataset might not be large enough to train a robust model that generalizes to unseen data, causing overfitting
- Variations in image resolution, lighting, and angles may affect the quality of annotations and feature extraction.
- Bounding boxes and class labels may contain inaccuracies due to human error or inconsistent annotation practices.
- PCB defects may be subtle or difficult to distinguish visually, especially when defects are small or hidden in complex backgrounds.
- The model is computationally expensive

06. Results and Conclusion



Our model performed well in a number of images particularly in the spur and spurious copper category images. However, is not accurate enough to perform a correct reading on the vast majority of PCBs. This could be improved in the future by using a larger dataset to train our model and more real PCB images to train the model for more epochs. we were limited on time and computational power and could not run it on enough epochs.

06. Future work

If we have an opportunity to continue working on this project, we would love to build a website fronted for the model where one can upload images and get immediate feedback regarding the defects.