

ENEE439D Final Presentation: PCB Anomaly

Tejas Guha, Mani Pabba, Hassaan Mastoor

Group Members



Tejas Guha



Hassaan Mastoor



Mani Pabba

The Problem

- Need to identify + classify defects in PCB
- Very labor intensive and error prone to do manually
- Issues
 - Non-uniform defects in Aerospace data
 - Almost only soldering defects
 - Lighting inconsistencies



Figure 1: Defect in data from Aerospace corporation



Past Work

- Potential Approaches
 - Direct Segmentation + Classification
 - Template-based approach / Autoencoder
- Past projects used computer generated PCB images

Our Data

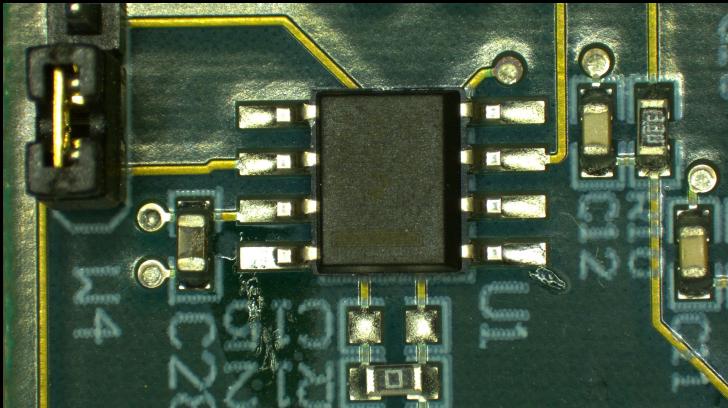


Figure 2: PCB from Aerospace corporation

Last Group's Data

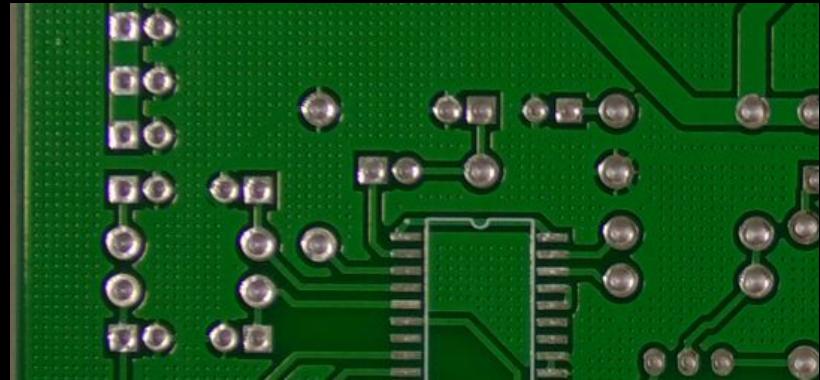


Figure 3: PCB From Kaggle Dataset (Peking University)

Approach Explored:
YOLO + Style Transfer

Model Performance

- Trained on Kaggle PCB Dataset - box loss of .1 and obj loss of .02 on validation

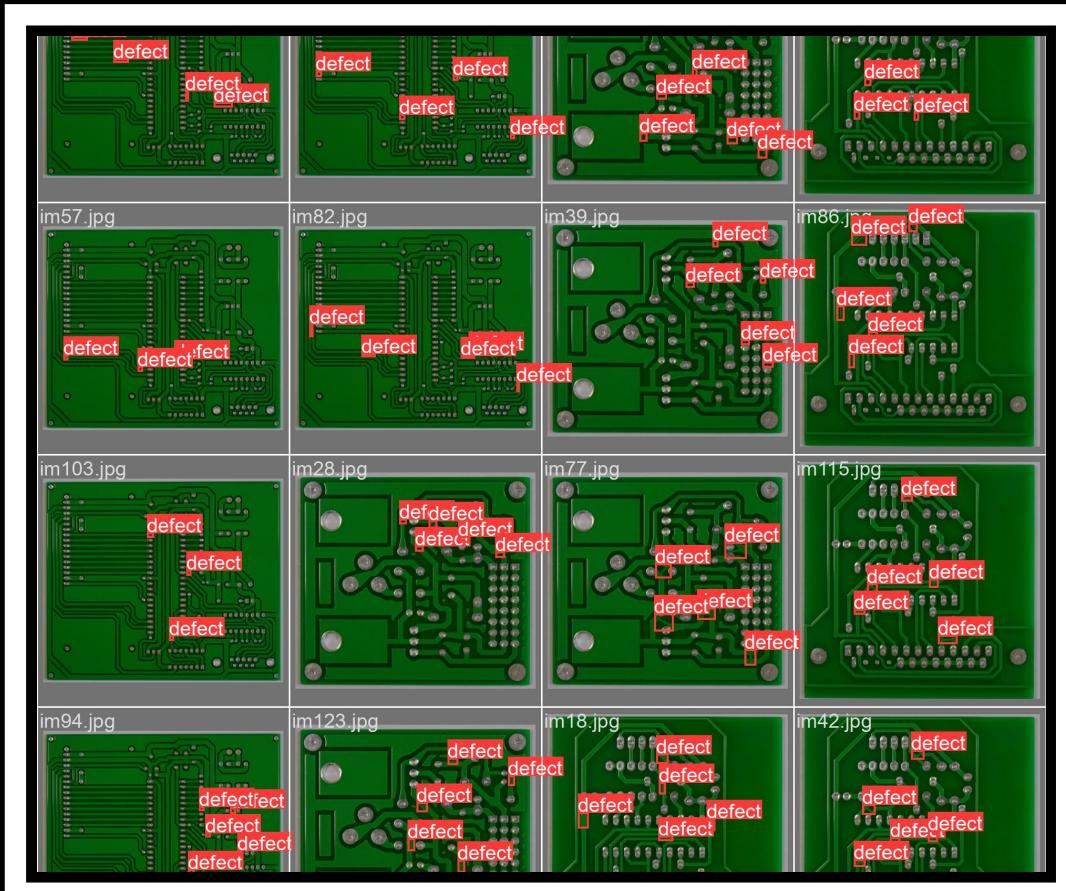


Figure 4. YOLO Model Performance

Style Transfer to Kaggle

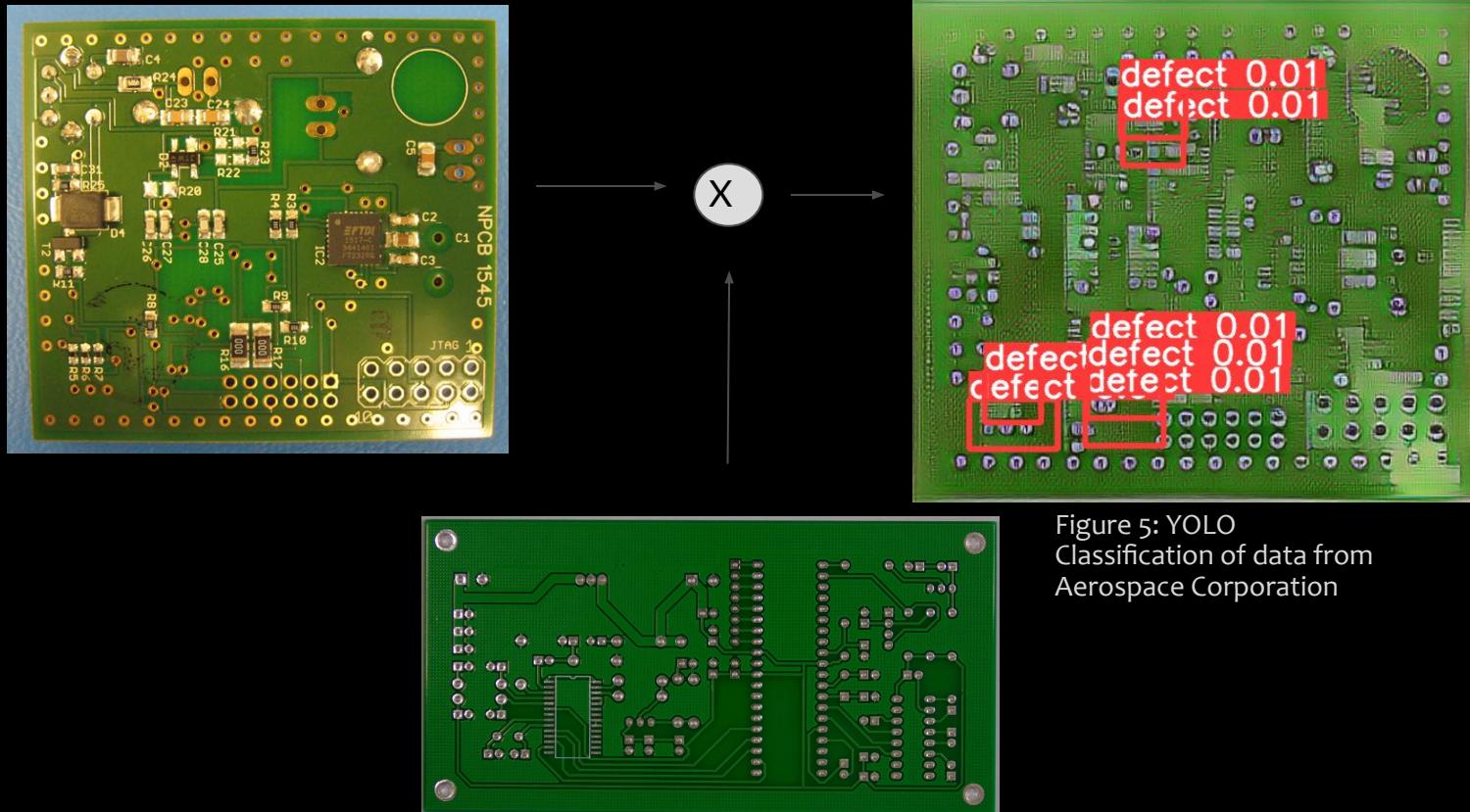


Figure 5: YOLO
Classification of data from
Aerospace Corporation

Current Approach:
Denoising Convolutional
Skip Autoencoder

Image Preprocessing

- RGB Image -> RB Image (Set green channel to zero)
- RB Image -> HSV Image
- Split HSV Image to H, S, and V images
- Multi-Otsu Thresholding on V image with 3 classes
- Produces a binarized image

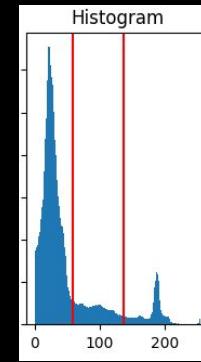
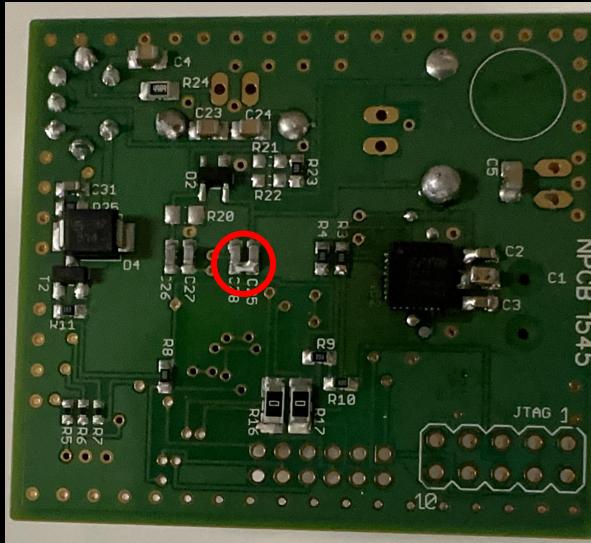
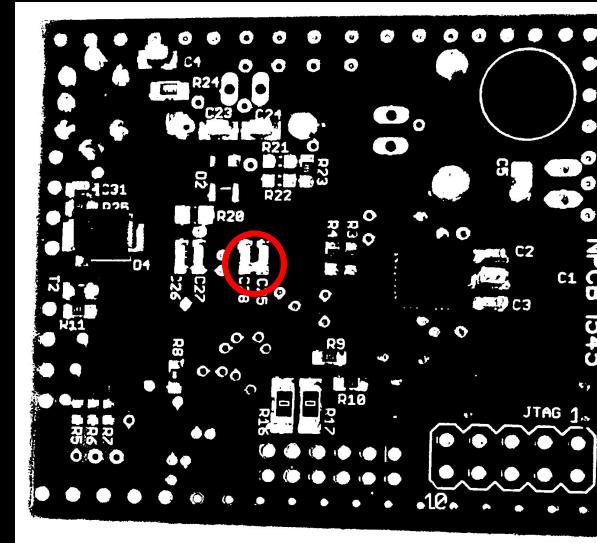


Figure 6. Otsu Thresholding Histogram

Original Image

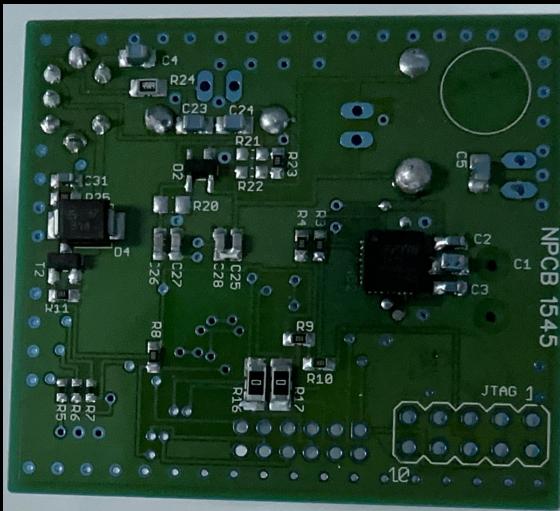


Processed Image

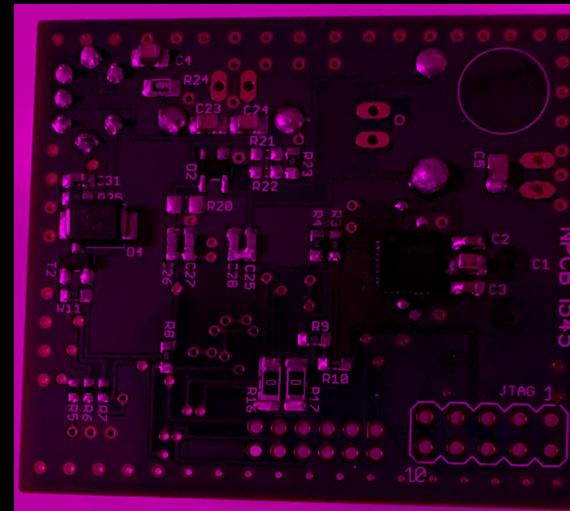


RGB Image -> RB Image

RGB Image

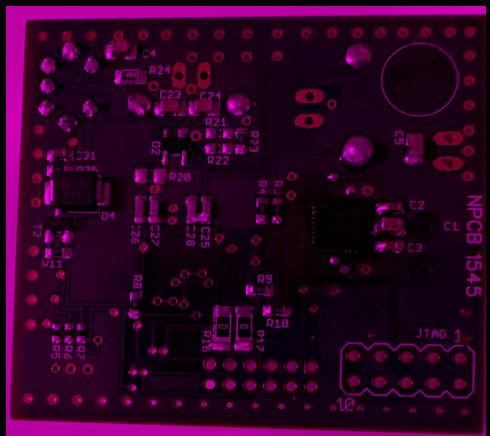


RB Image

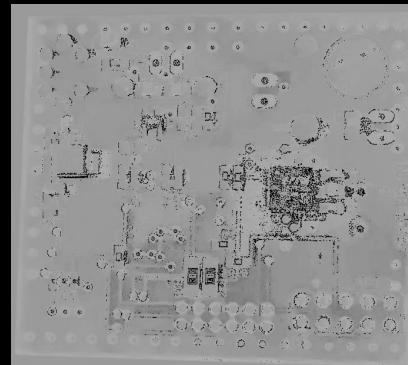


RB Image -> HSV Image

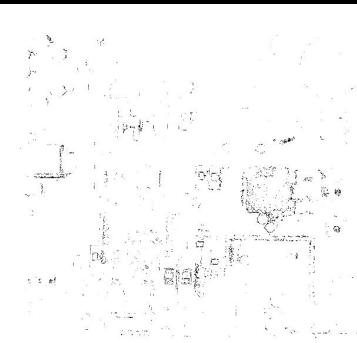
RB Image



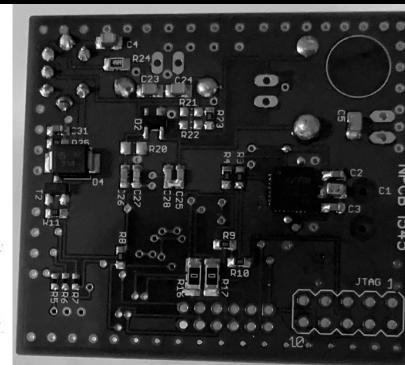
Hue Channel



Saturation Channel

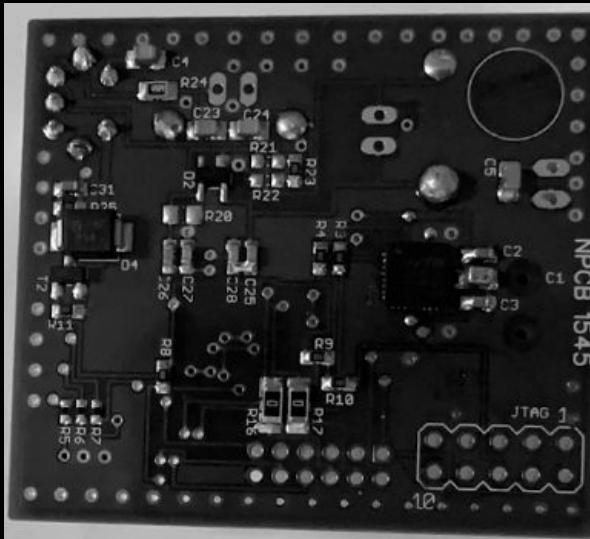


Value Channel

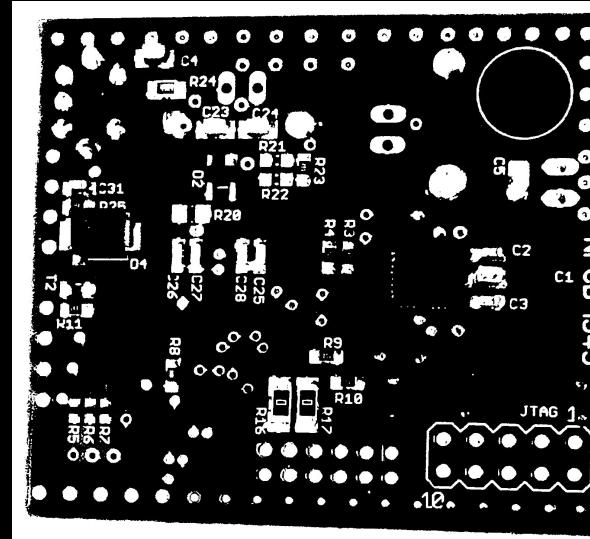


V Channel Image -> Otsu's Multi-Threshold Algorithm

Value Channel Image



Processed Image



Approach

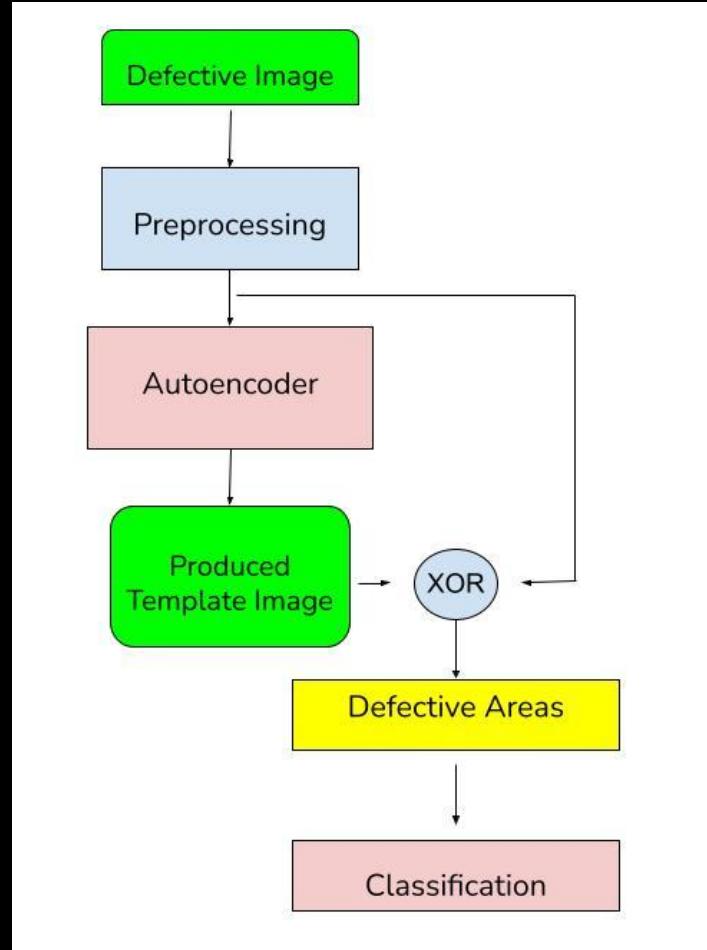


Figure 7. Overall Design Flow

Autoencoder Architecture

Encoder

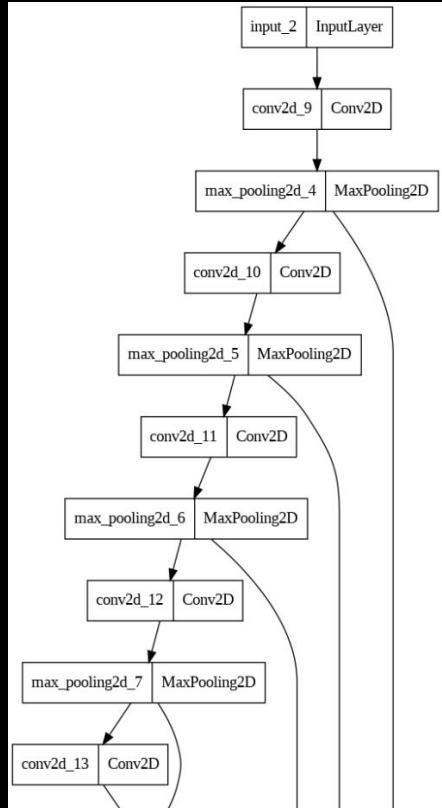


Figure 8. Encoder Flow Chart

Decoder

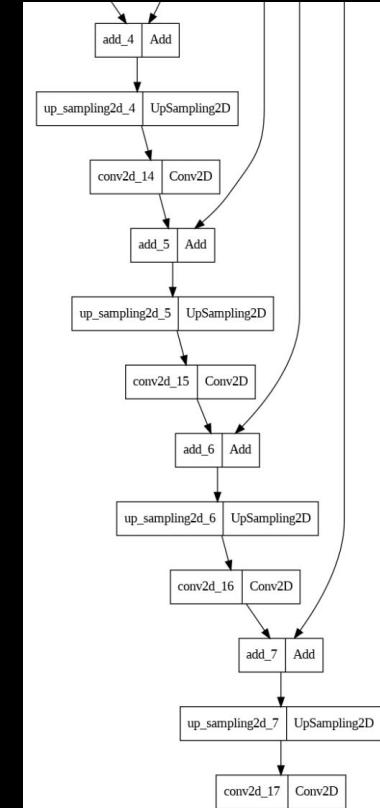


Figure 9. Decoder Flow Chart

Training

- Program for defect annotation of PCBs
- Performs alignment between images
- Just click on a defect

Issues: Alignment not 100%

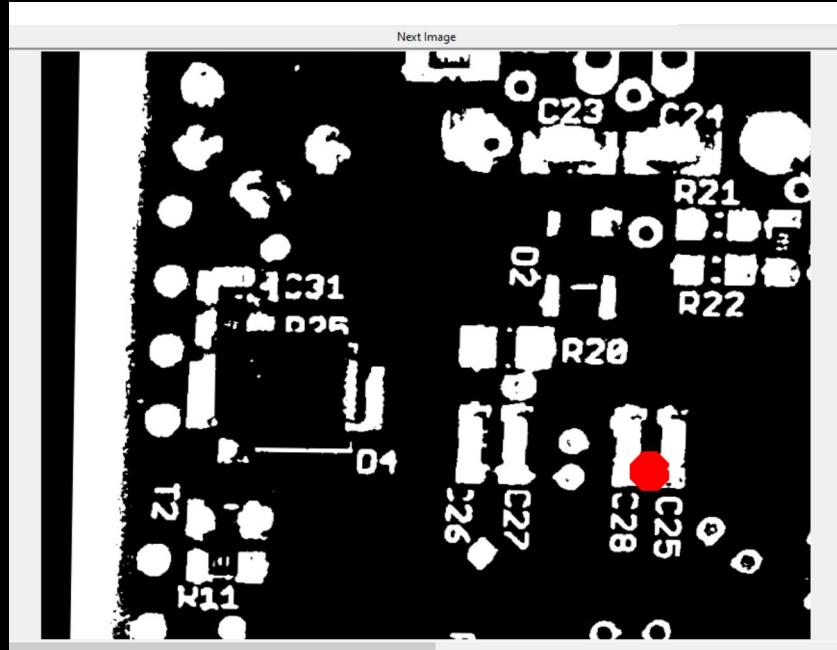


Figure 10. Screenshot of Annotation Program



Training

Dataset:

- 1550 Aerospace PCBs (50x repeated)
- 2000 DeepPCB

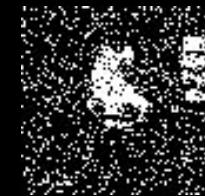
Training steps:

- Adding Salt + Pepper noise to a defective image and reconstructing nondefective
- 200 epochs with MSE Loss



Figure 11. Autoencoder Training Procedure

Trained
Image With
Noise



Autoencoder
Reconstruction



Differenced
Image



Usage of DeepPCB Dataset to Supplement Learning

Figure 12. DeepPCB Reconstruction



Original



Difference



Reconstructed

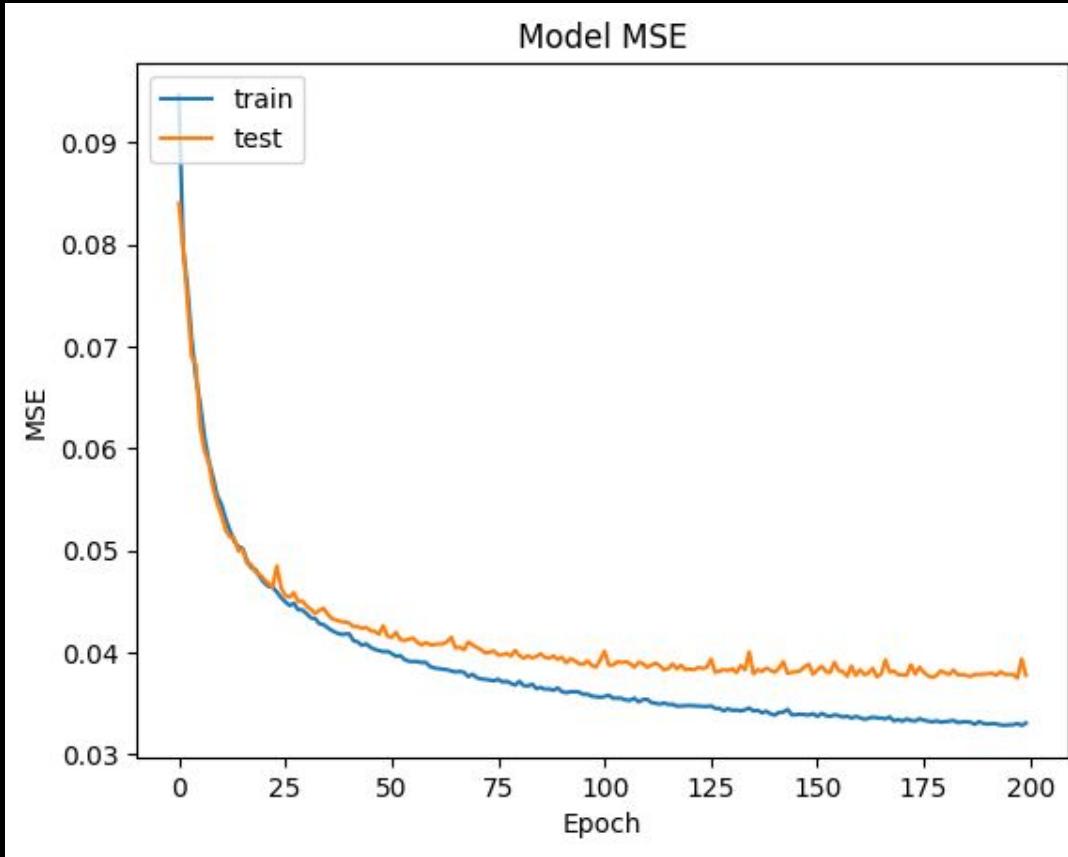


Figure 13. Autoencoder Loss MSE

MSE Loss of Autoencoder

Quick Overview of the Process

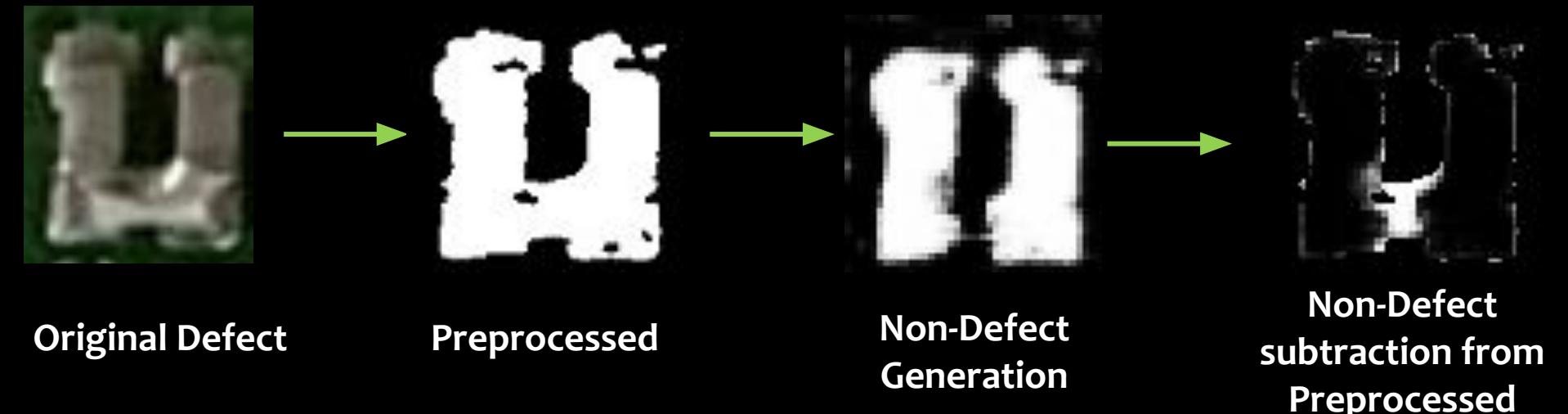


Figure 13. 80X80 Defect Reconstruction

Demonstration

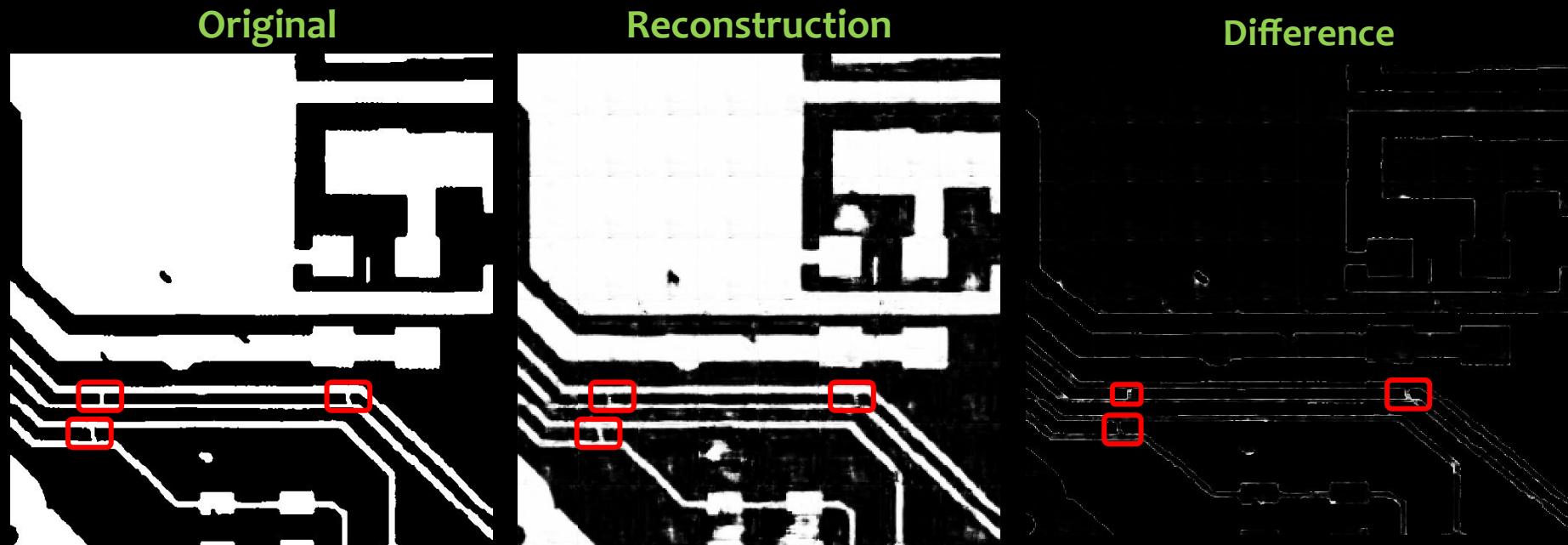
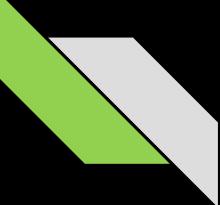


Figure 14. Entire PCB Reconstruction



Challenges

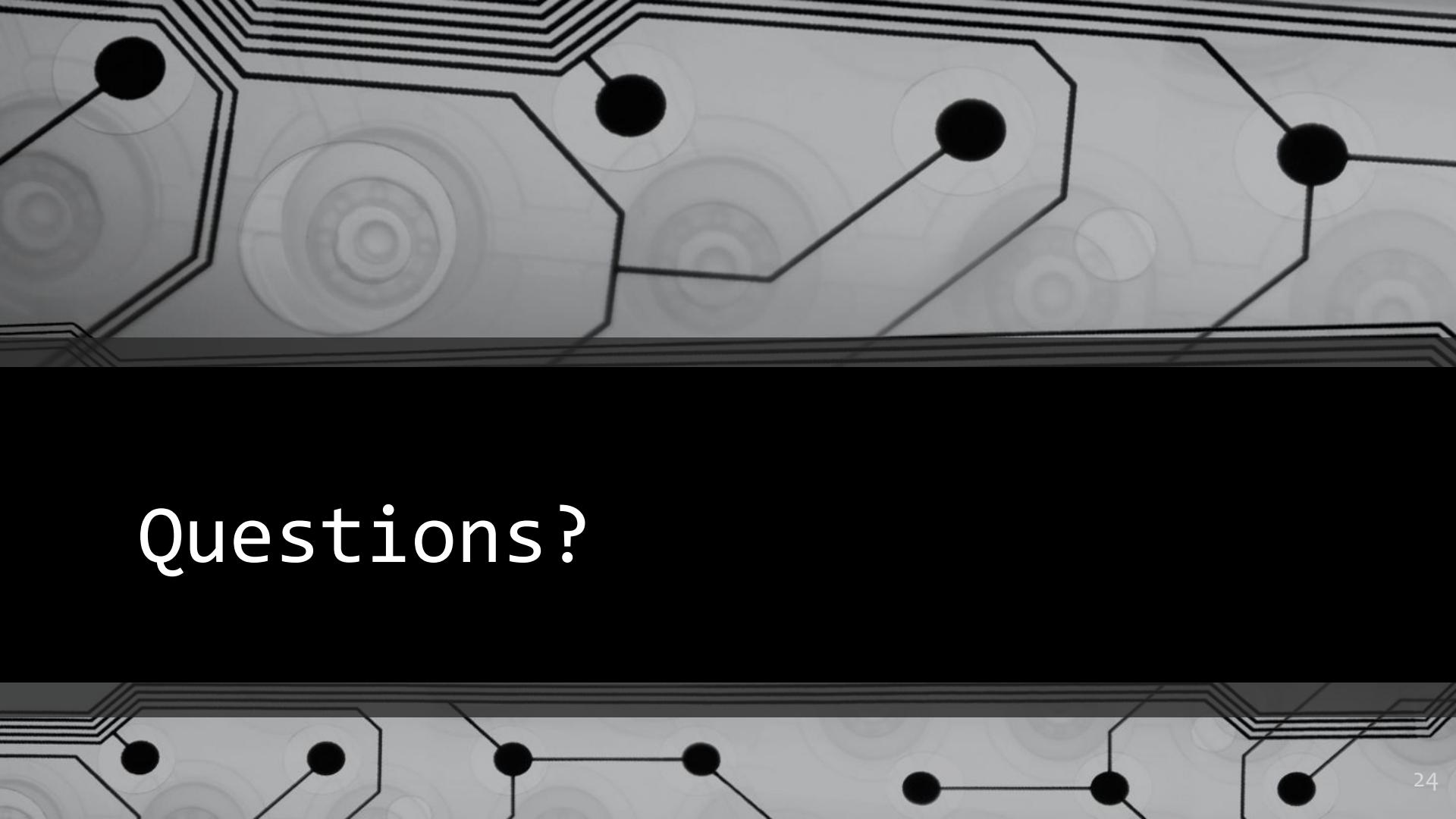
- Image pre-processing
- Size of data set
- Lack of nondefective/defective image pairs
- Image alignment + image capture



Recommendations

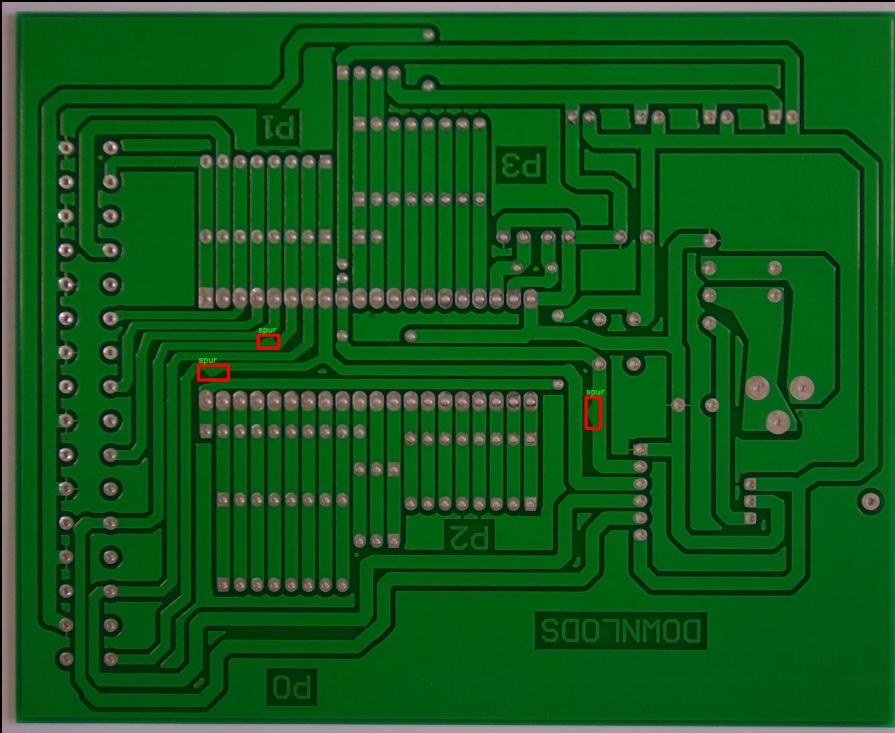
- Segment PCBs by components
- Augment training images
- Larger image sizes
- Images from different perspectives for solder-related defects

Demo



Questions?

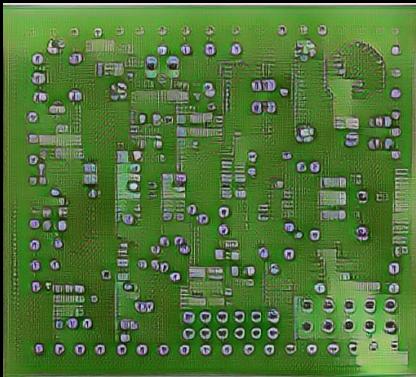
Kaggle Dataset Visualizations



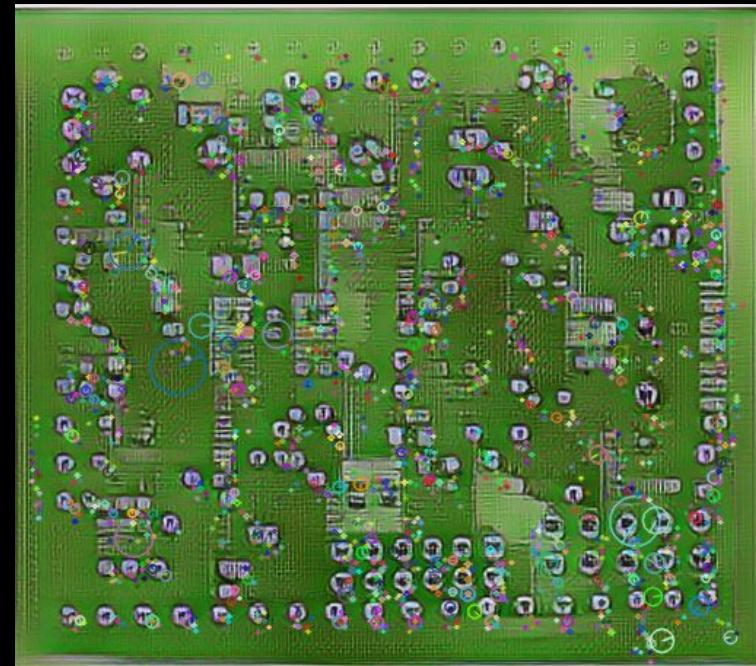
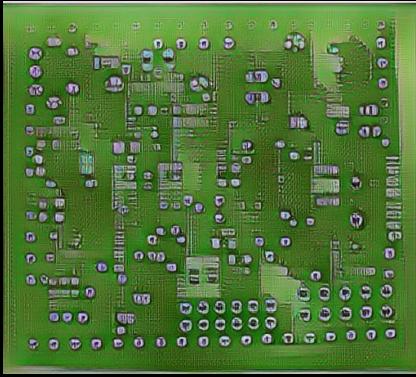
Spurs Highlighted in sample image

Style Transfer Seems to Perform better in keypoint detection

Defect 1



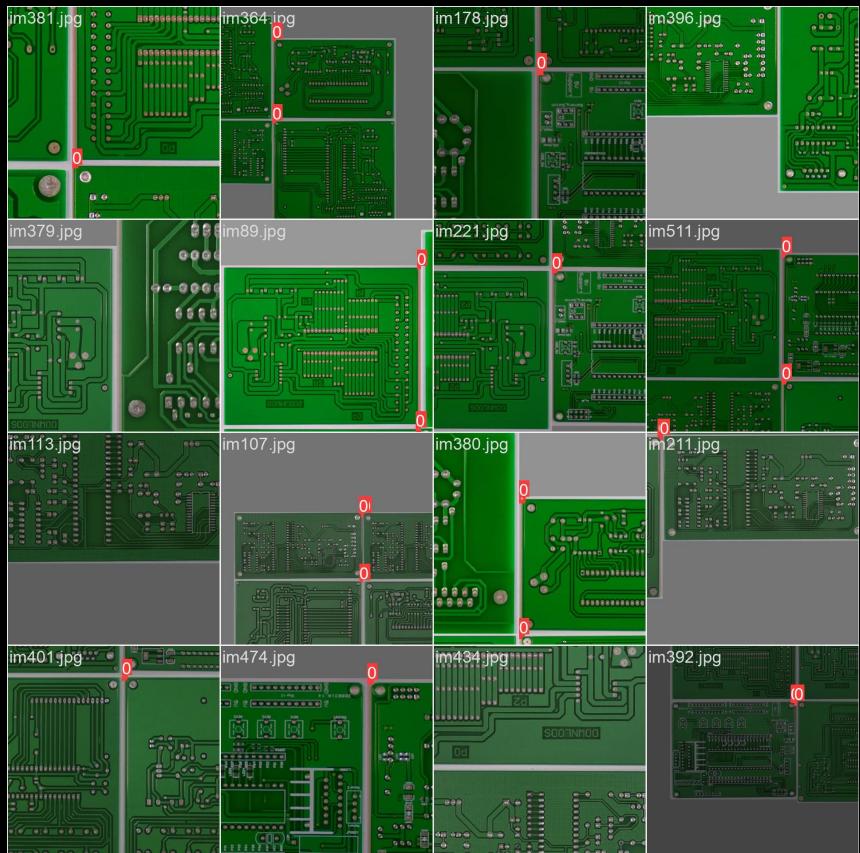
Defect 2



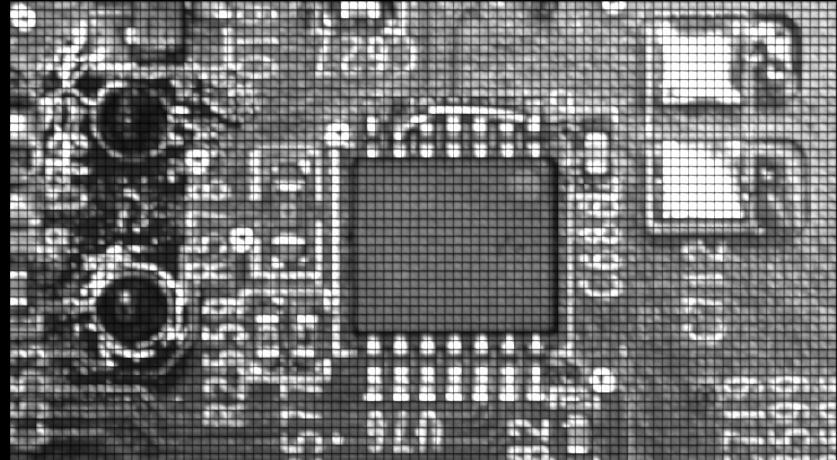
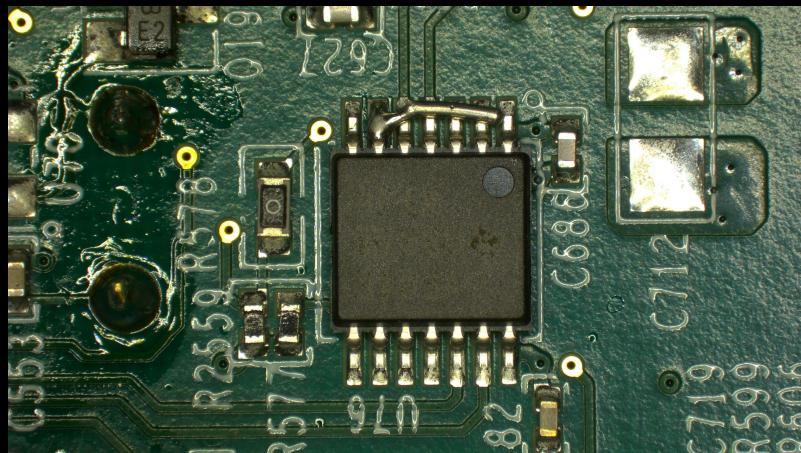
YOLO v5 Model Output

Identifying only corners in images

- Most likely a defect in the training images – currently investigation the issue



Denoising Autoencoders



Preprocessing

