Computer Vision for Advanced Circuit Fabrication

•••

Isaac Rodriguez (Sonoma State University)
Connor Emery (Bryant University)
Case Kirk (Ozark Integrated Circuits)
Walt Niblock (Ozark Integrated Circuits)

National AI Campus
Spring 2023 Showcase

Team Members

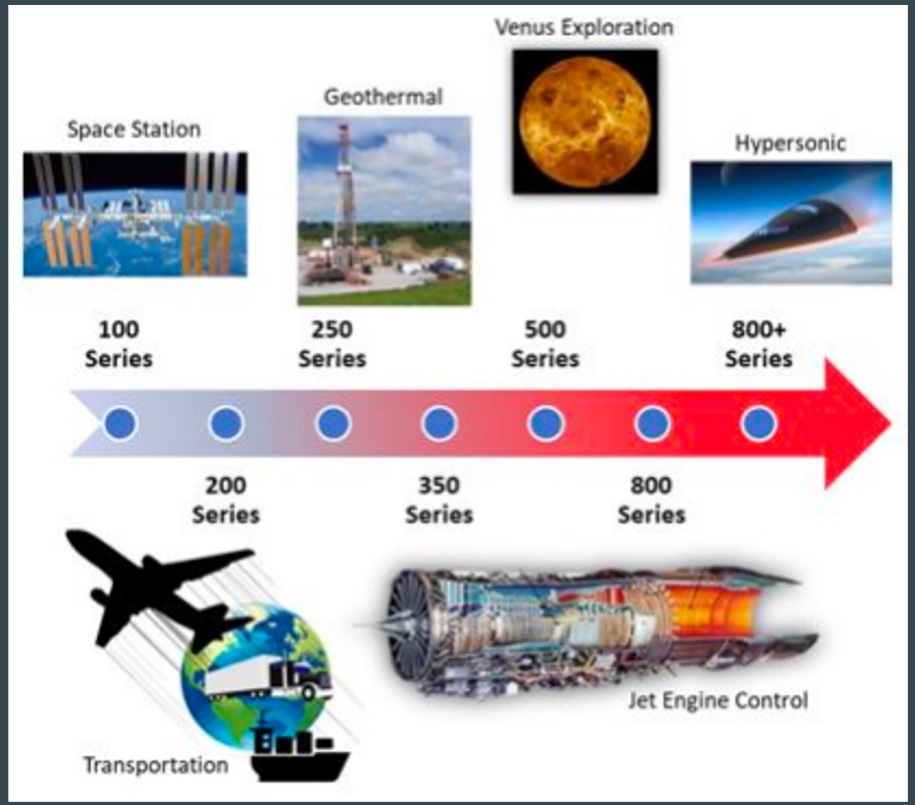
Coach Team:

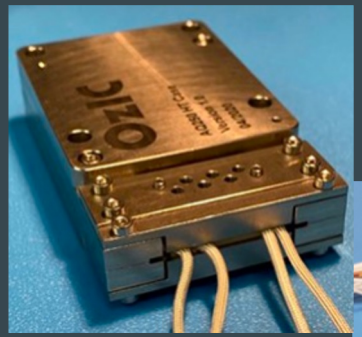
Case Kirk (Ozark Integrated Circuits)
Walt Niblock (Ozark Integrated Circuits)
Jennifer Fowler (Arkansas State University, AR NSF EPSCoR)
Drew Fleming (Arkansas State University)

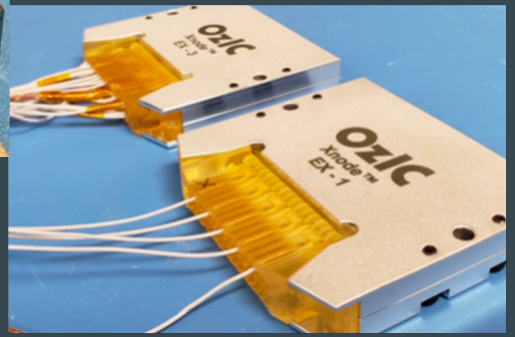
Participants:

Isaac Rodriguez (Sonoma State University, CA)
Connor Emery (Bryant University, RI)

Ozark Integrated Circuits (OzIC)

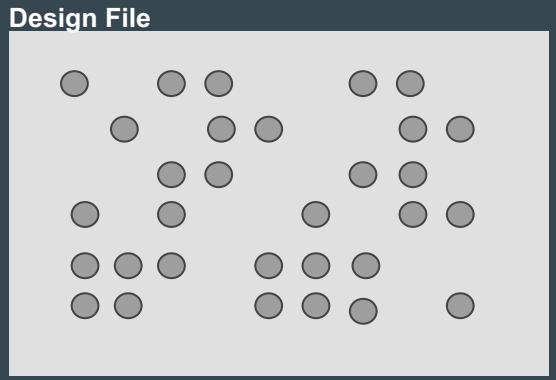


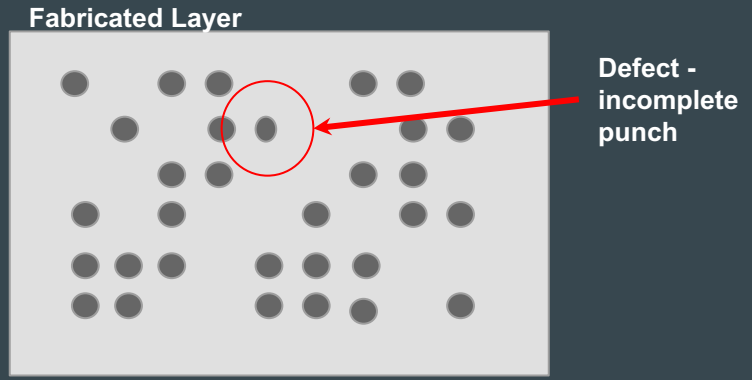




Project Background & Data Source

- Quality control is manually performed, with someone carefully examining scanned images of each layer of the substrate ensuring each punch went through cleanly with no blockages.
- OzIC already uses Python in a variety of processes and was interested in exploring automation.
- Dataset included scanned images of layers used in production modules (design templates, actual punched substrate, and some failed layer examples).
    - Actual images are subject to export control, and all team members signed non-disclosure agreements and worked with the data securely.





Project Challenges

Initial project challenges:

1. Comparing punches on the design template to the actual fabricated layer making sure the quantity of punches match.

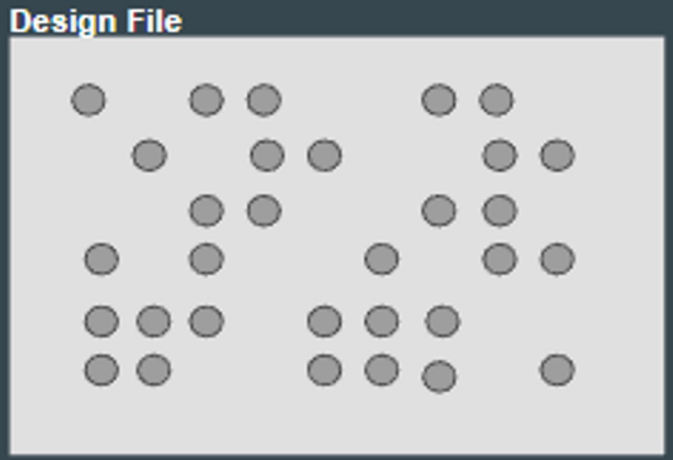
1. Examining each punch (thousands per layer) to make sure the punch was complete and clean.

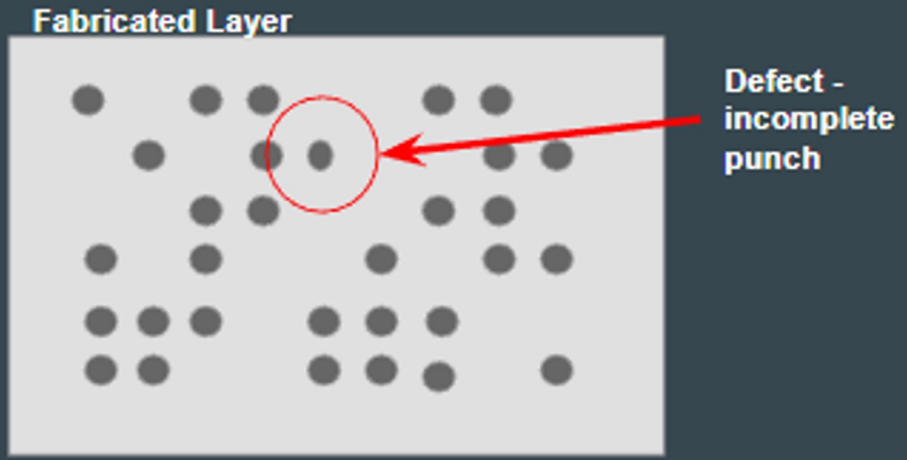
1. Creating data for a convolutional neural network by segmenting isolated pictures of vias into separate images.

1. Making sure that all defects were caught in the detection phase

Problem Statement

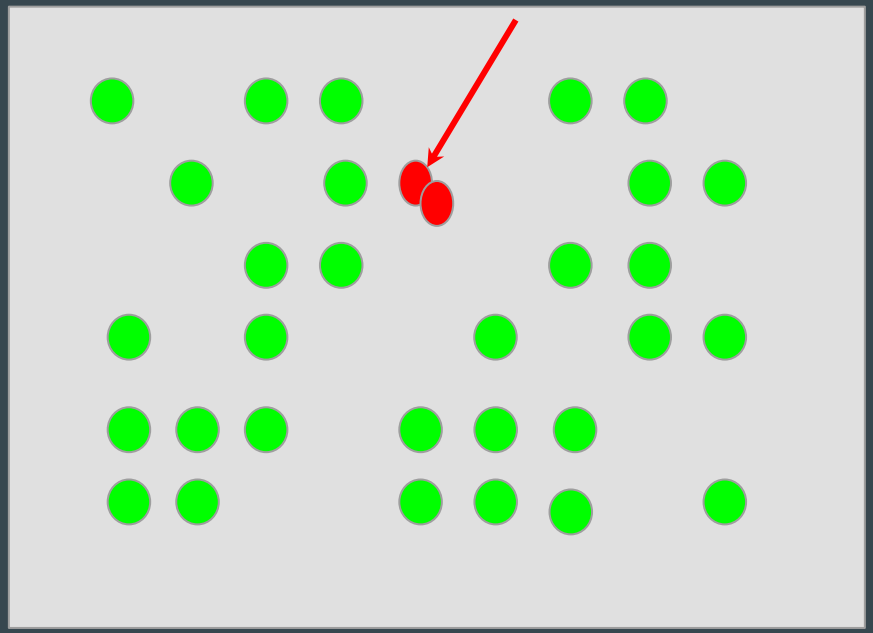
Yield verification of manufactured products in real-time through the use of python and machine learning.



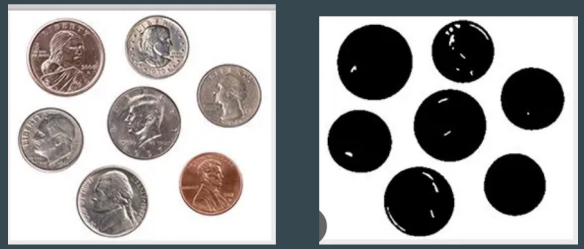


Punch Counter

- Counts all punches excluding defects



Binary Thresholding



Minimum Pooling

Erosion Filter

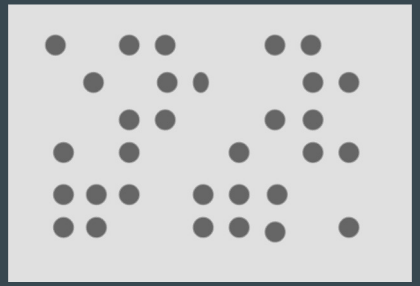
Visualizations

Defects:

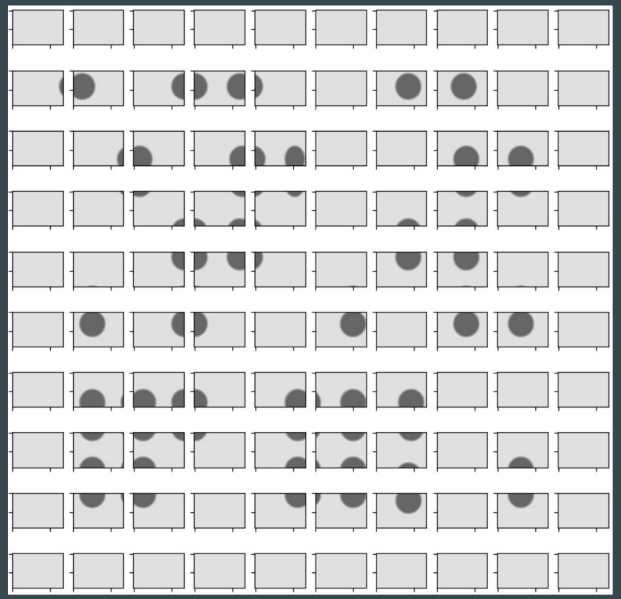
Clean Punches:

Object Detection

- Trying to capture vias



Original Image



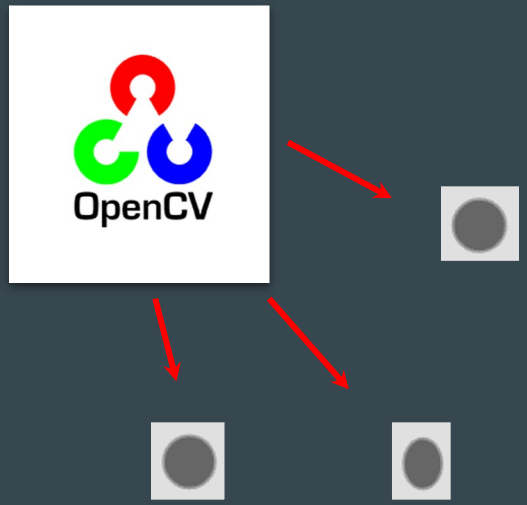


Image segmentation

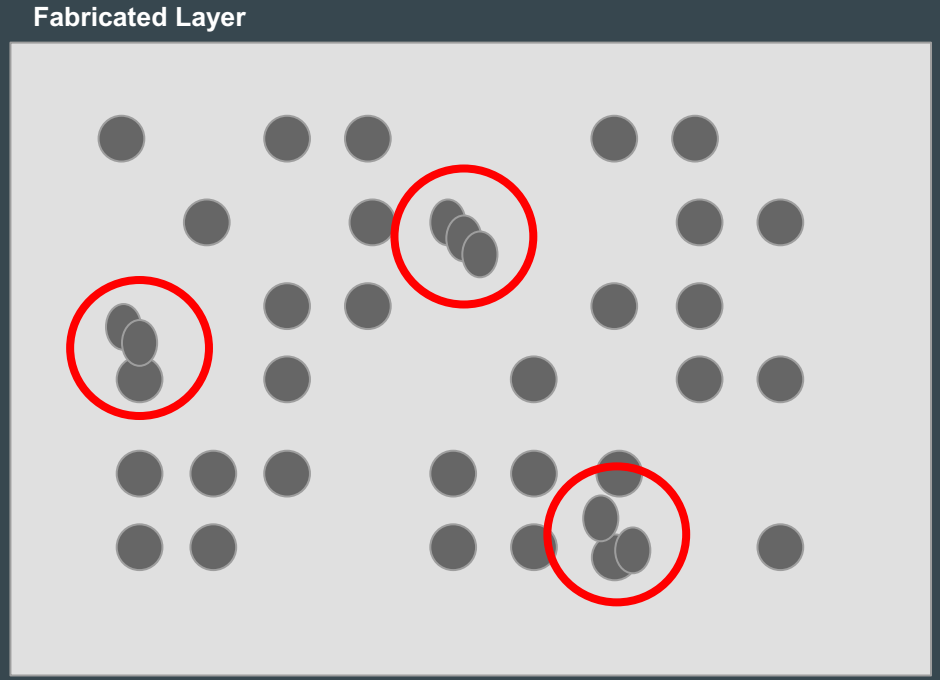
- Created a python script in order to take the original image, convert that image to grayscale and find contours using cv2 library
- These new images were then wrote to a new file so they could be analyzed further in a defect turker

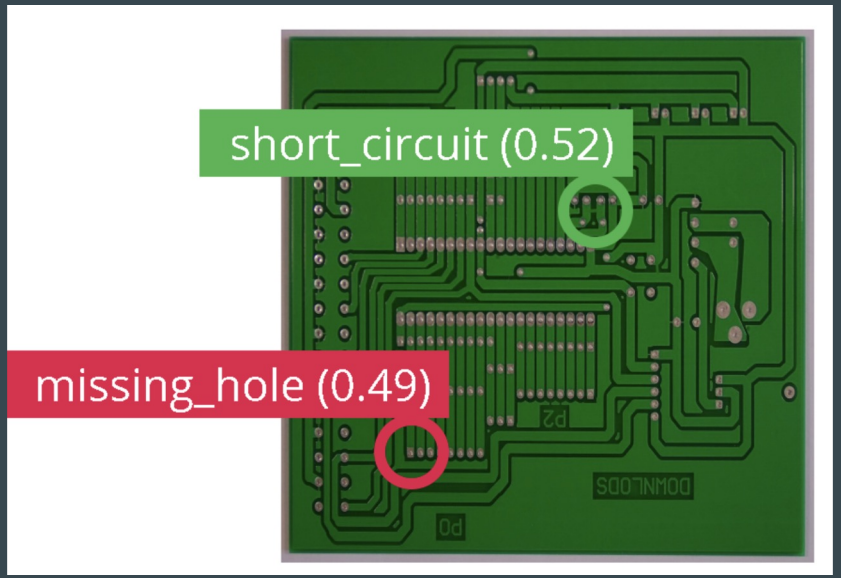
Challenges

- Due to the nature of the images, many times images would come out grainy and it was hard to tell exactly where the borders were. By creating the initial threshold and backing off of it slightly, this should be able to fix this issue for the future

Anomaly Detection & Analysis

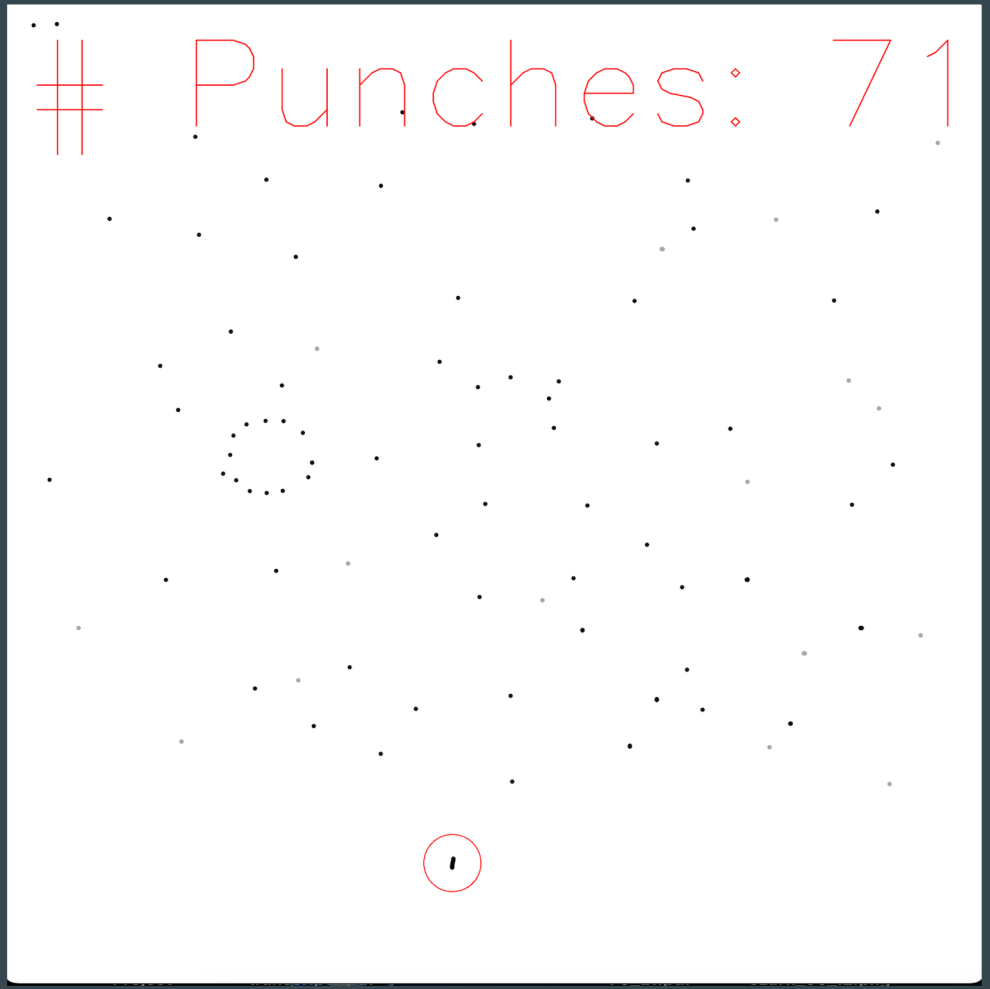
- Highlights areas that may be defective





Results

- We were able to preprocess and correctly analyze many of the boards
- Ultimately unable to train a model yet to detect whether or not there was a defect in a given image
- Accomplished all the base work for a future model

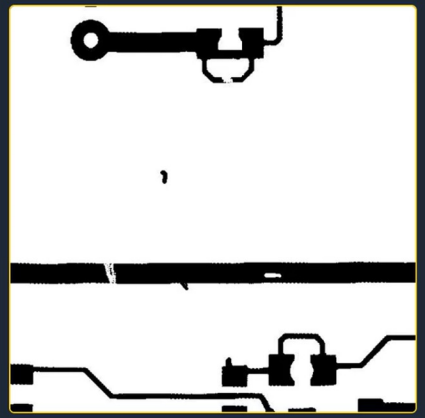
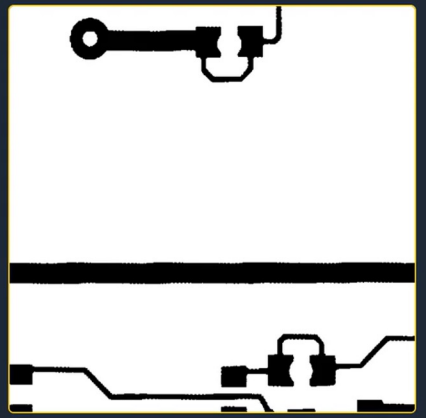


Suggestions for Future Work

- Expand program functionality to inspect printed via-fill and trace yield.
  - Via-Fill inspection will use the same image segmentation as the punched files with different preprocessing and analysis steps to highlight issues.
- In order to create a working algorithm, we first have to create an application that can take in the images as input and allow someone to mark what kind of defect they are
  - This is the first step in taking our preprocessing work and making into a machine learning algorithm

FETCH BOARDS | RUN PIPELINE

PCB DEFECT CLASSIFIER

RESULTS	TEST	TEMPLATE
		

SPUR		$1320
SPURIOUS		$2180
SHORT		$1985

Acknowledgements

Thank you to the National AI campus spring cohort team and leadership for a great semester and to the OzIC team for the constant information and mentorship that they provided throughout the semester!

References

Vimeo defect turker image:
https://vimeo.com/388296477?embedded=true&source=video_title&owner=104920680

https://opencv.org/opencv-python-is-now-an-official-opencv-project/

https://pyimagesearch.com/2021/04/28/opencv-thresholding-cv2-threshold/

https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9

https://www.geeksforgeeks.org/erosion-dilation-images-using-opencv-python/

https://docs.edgeimpulse.com/experts/prototype-and-concept-projects/identifying-pcb-defects