



Automated Visual Inspection with MATLAB

Paul Huxel, PhD

Senior Application Engineer, MathWorks

What is Automated Visual Inspection?

*“ Automated optical inspection is the **image-based** or **visual inspection** of manufacturing parts where a camera scans the device under test for both **failures** and **quality defects**”*

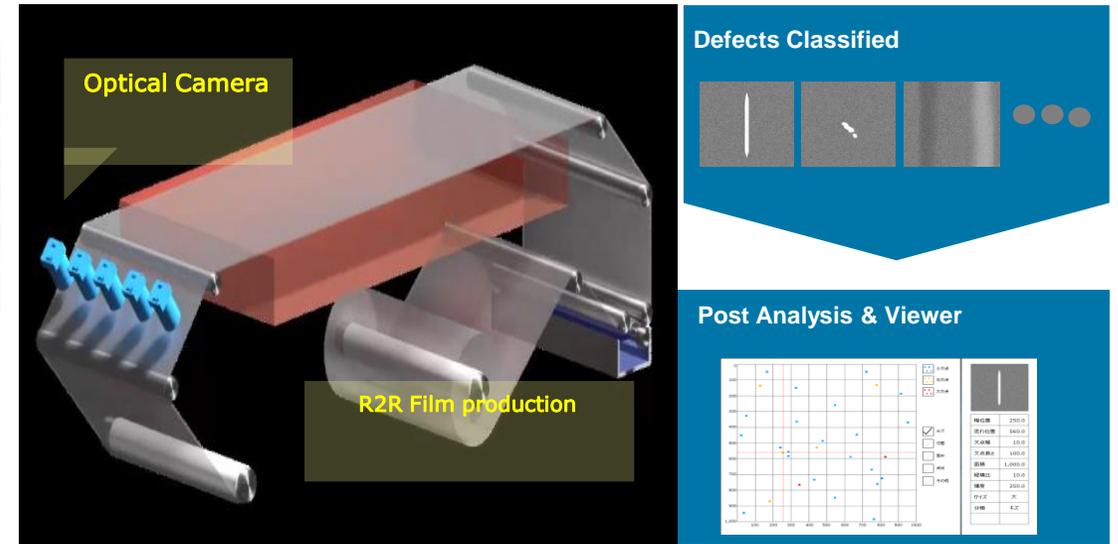
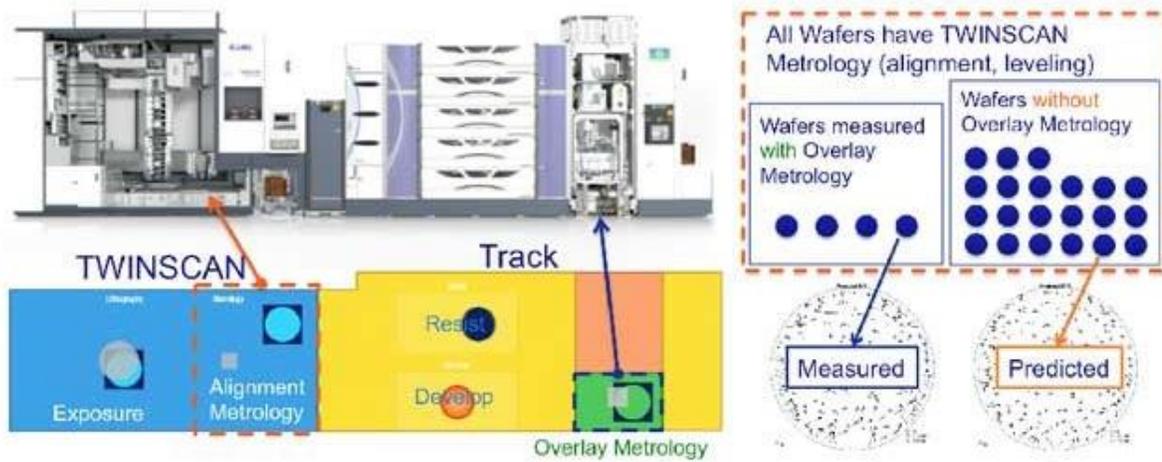
Automated Defect Detection

Machine Vision

Optical Inspection

Automated Inspection

User Stories



Metrology Technology for Semiconductor Manufacturing

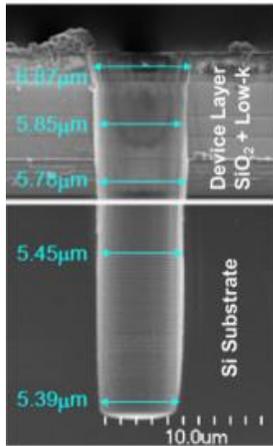
ASML

Film Production Inspection

Dexerials

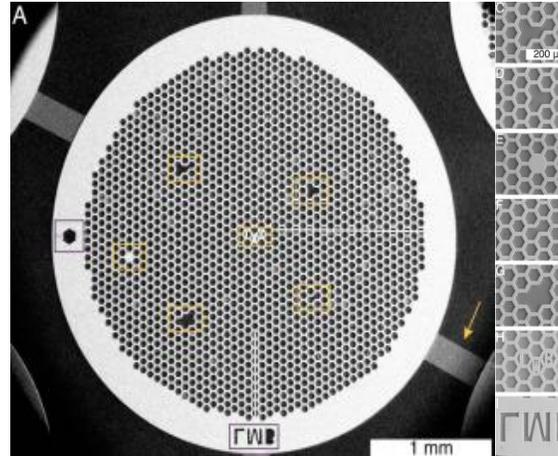
Visual Inspection Use Cases

- Consumer Electronics and Wafer Inspection



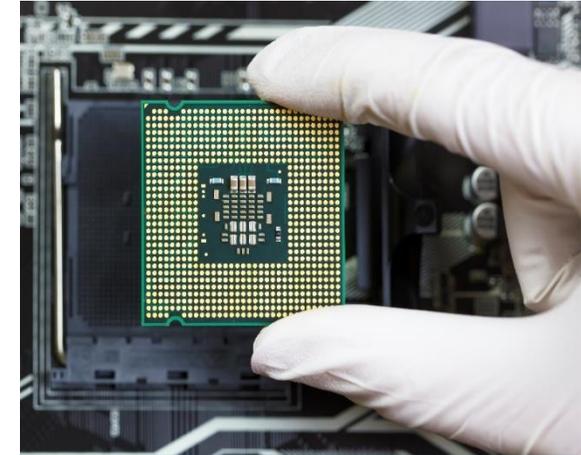
Metrology

[Image Credits: Review of Bumpless Build Cube \(BBCube\) Using Wafer-on-Wafer \(WOW\) and Chip-on-Wafer \(COW\) for Tera-Scale Three-Dimensional Integration \(3DI\)](#)

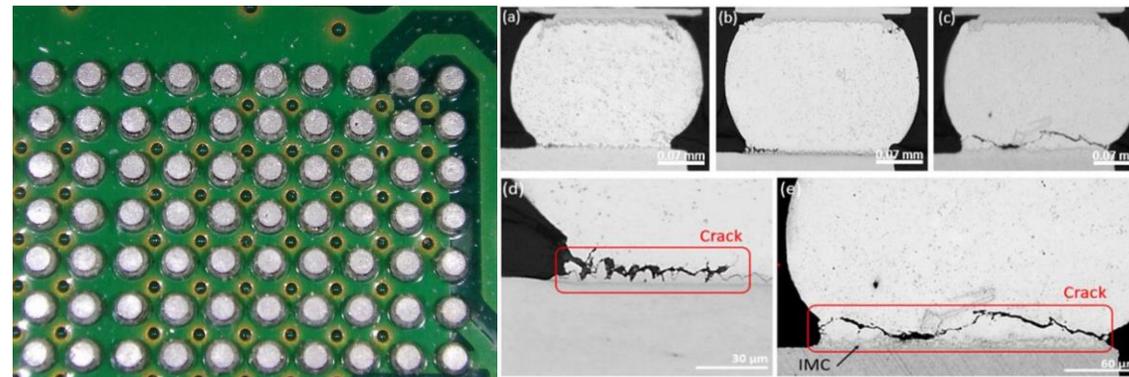


Fiducial Localization

[Image Credits: Integrated wafer-scale manufacturing of electron cryomicroscopy specimen supports](#)



Quality Control



Anomaly Detection

[Image Credits: Thermal cycling effect on the crack formation of solder joint in ball grid array package](#)

Typical Visual Inspection System

Inspection Cameras

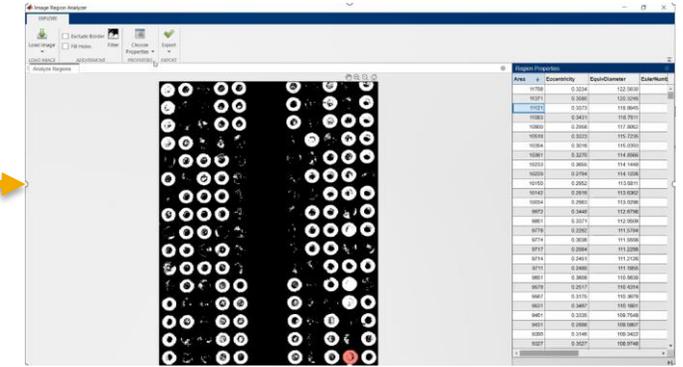
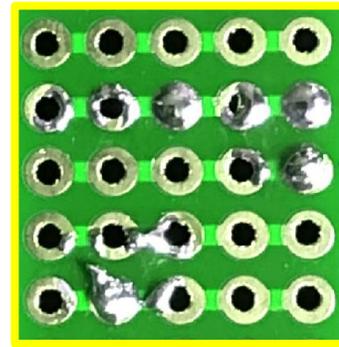
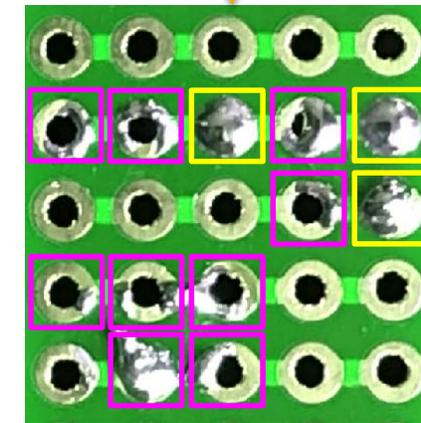
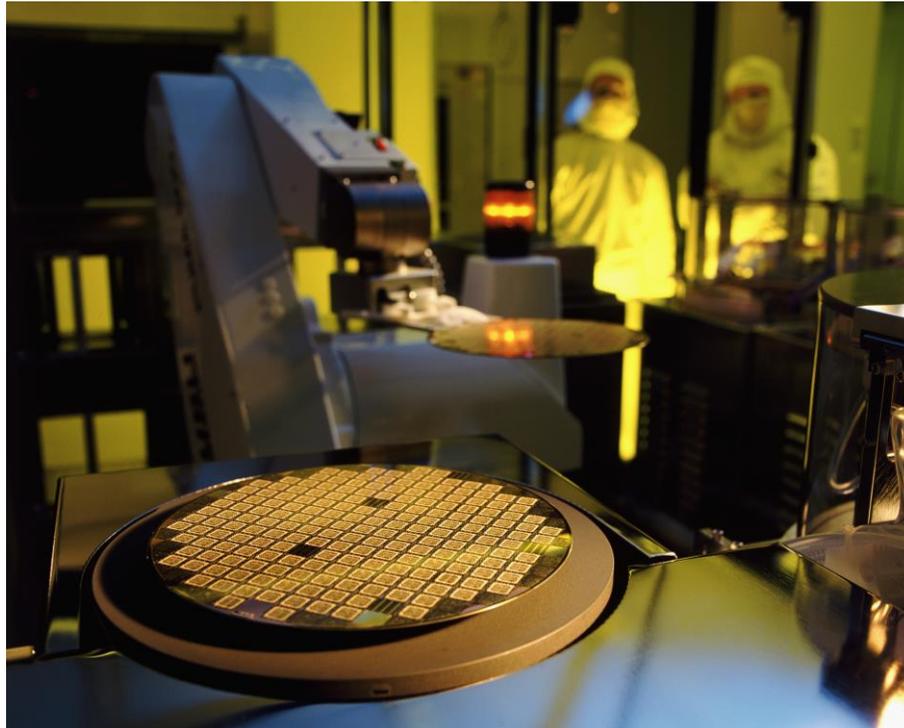
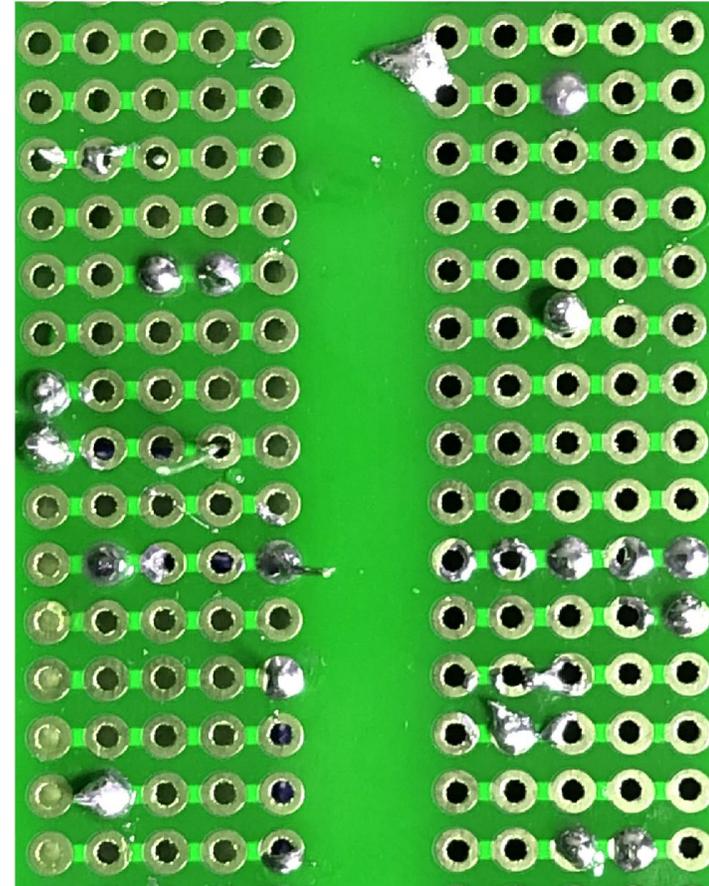
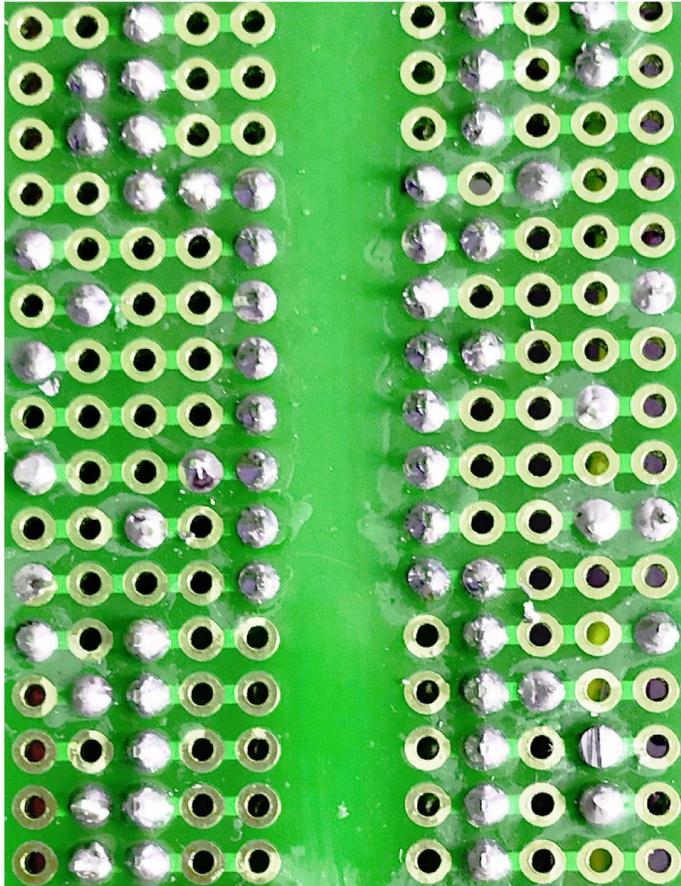


Image Analysis



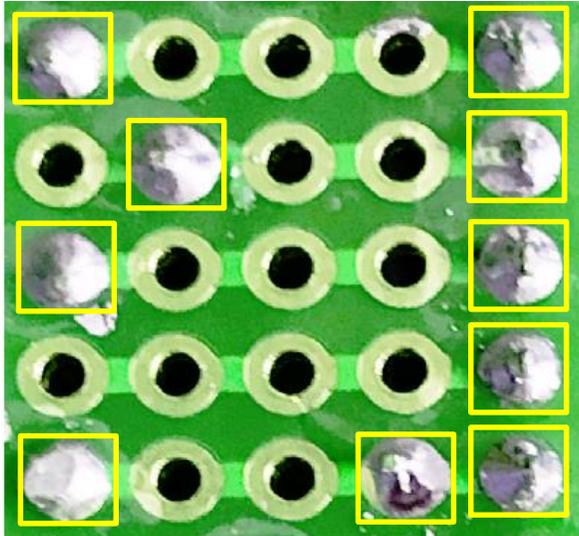
Defective Parts

Can you find the defective solder joint?

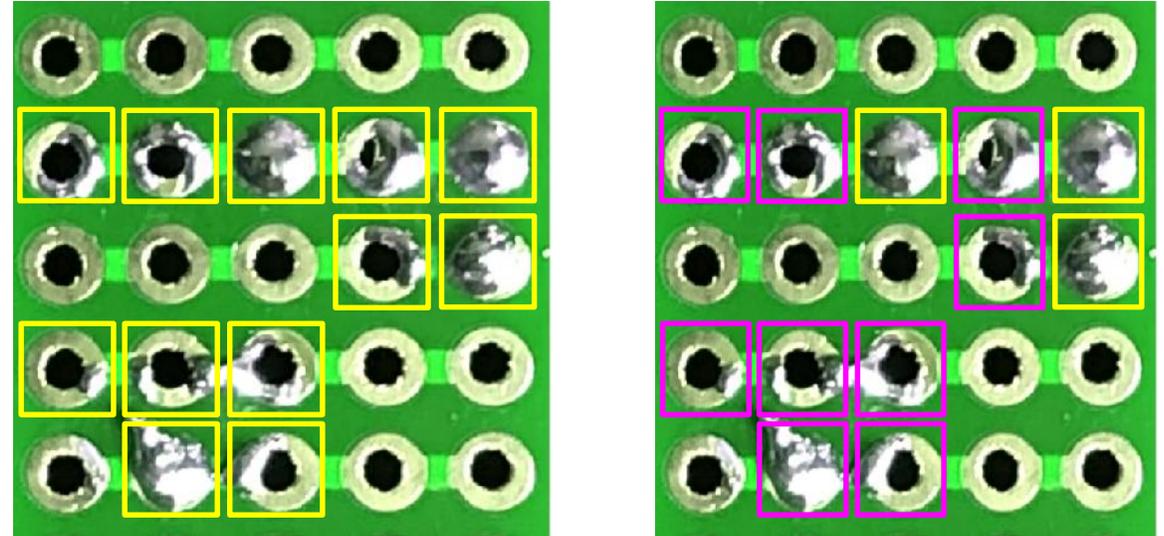


Defect Classification

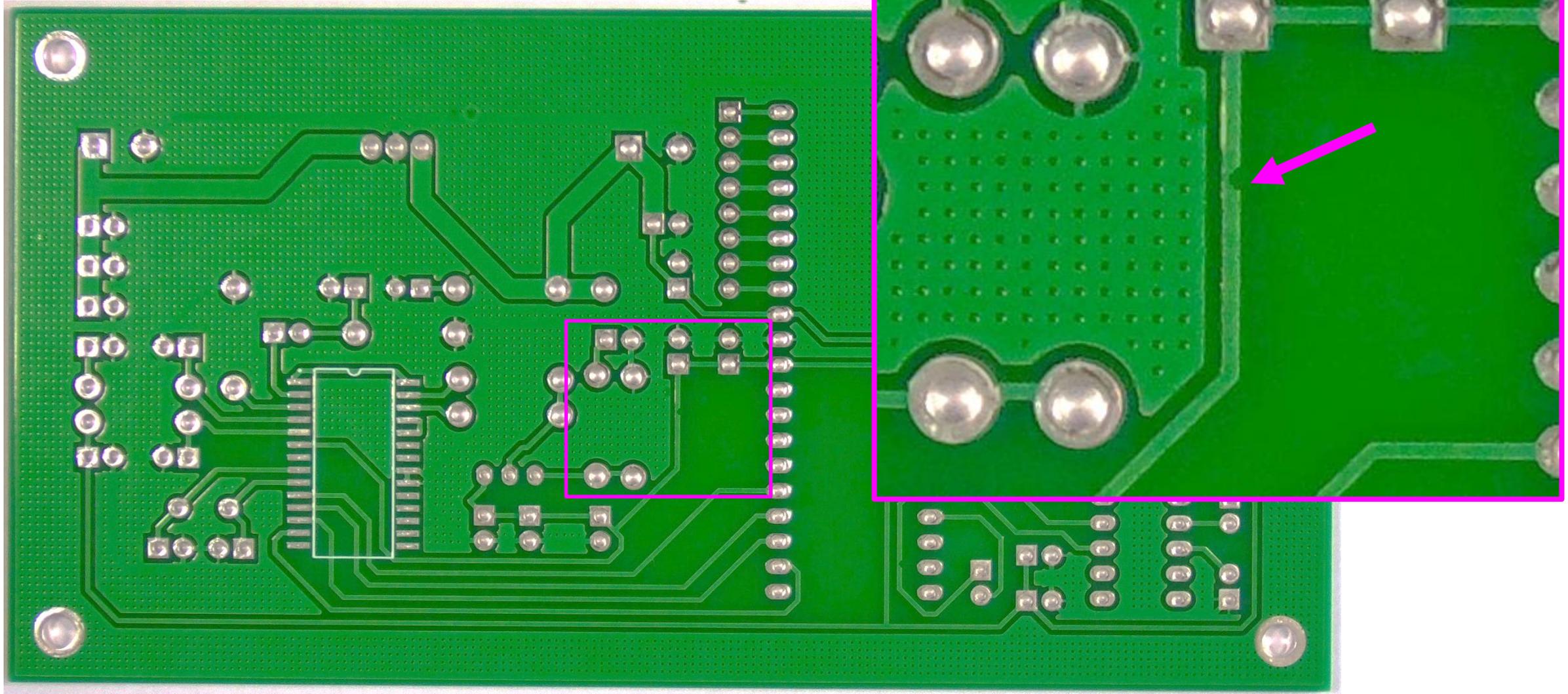
Good



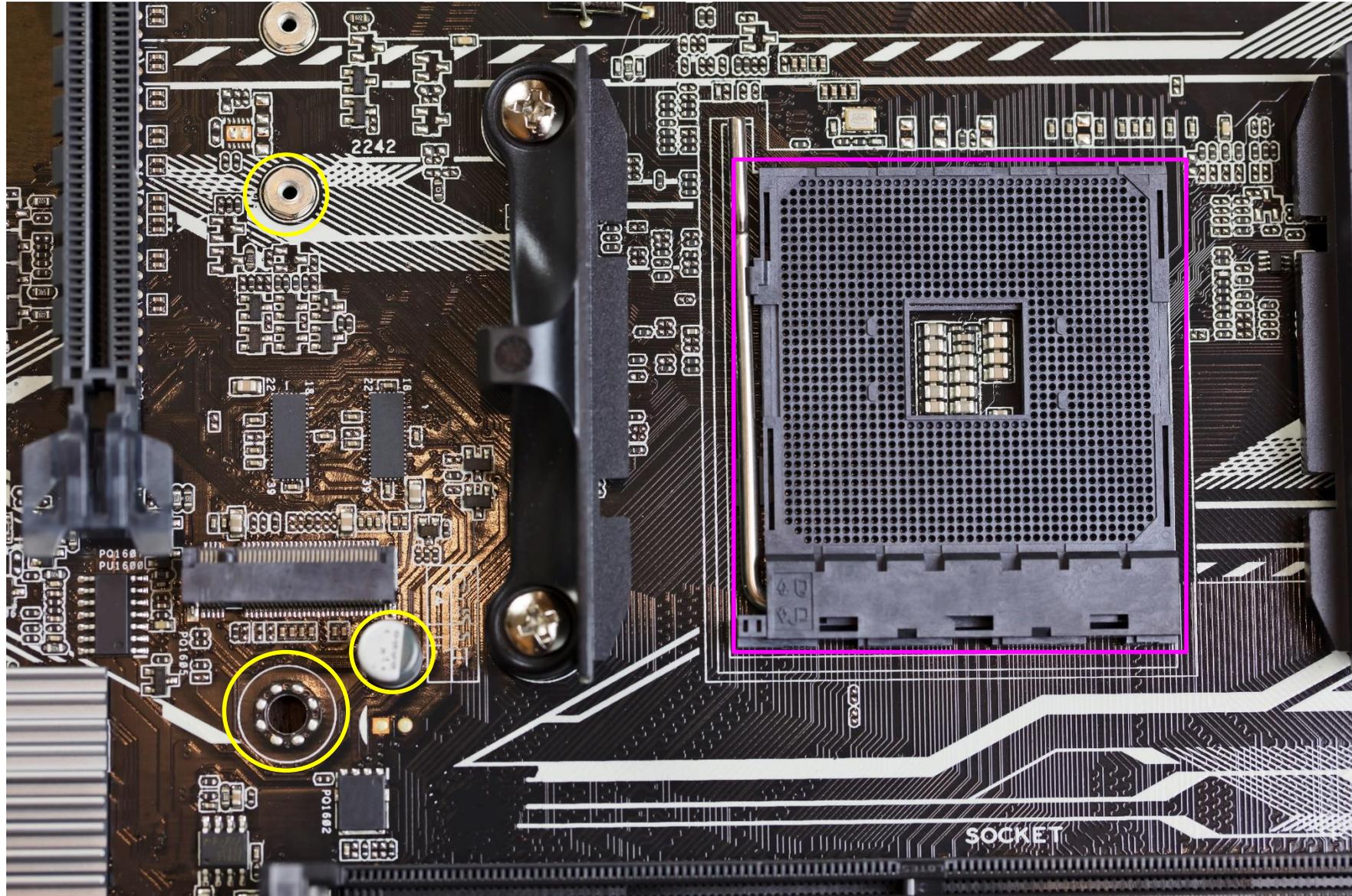
Defective



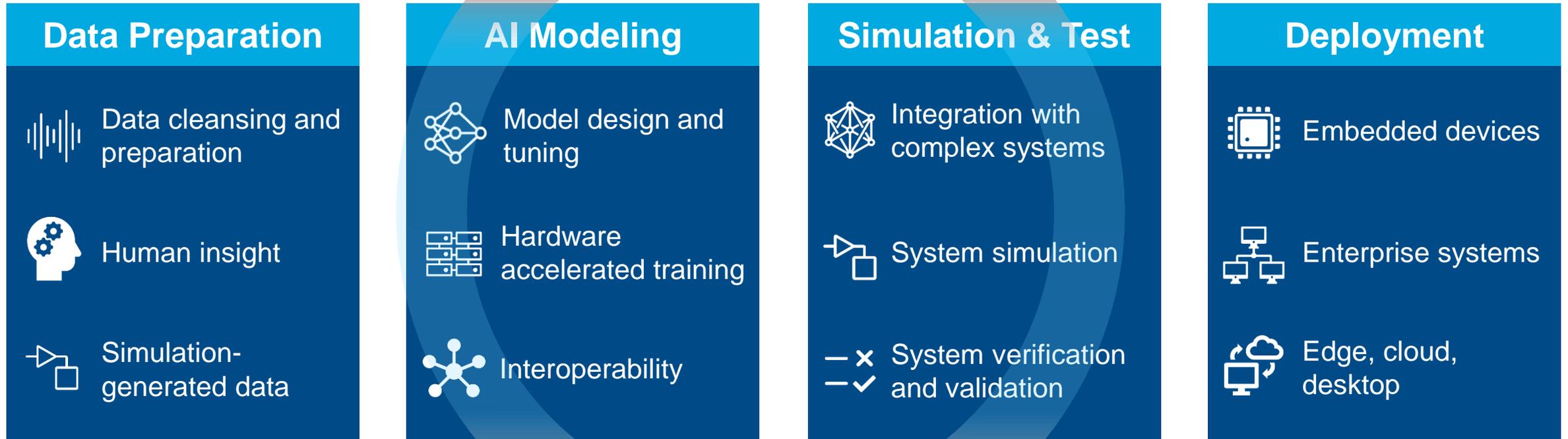
Anomaly Detection



Counting, Presence and Localization of Objects



Defect Detection Workflow



Defect Detection Workflow

Data Preparation

 Data cleansing and preparation

 Human insight

 Simulation-generated data

AI Modeling

 Model design and tuning

 Hardware accelerated training

 Interoperability

Simulation & Test

 Integration with complex systems

 System simulation

 System verification and validation

Deployment

 Embedded devices

 Enterprise systems

 Edge, cloud, desktop

Data Access and Preprocessing – Common Challenges

How do I preprocess data and get the right features?

How do I label my data faster?

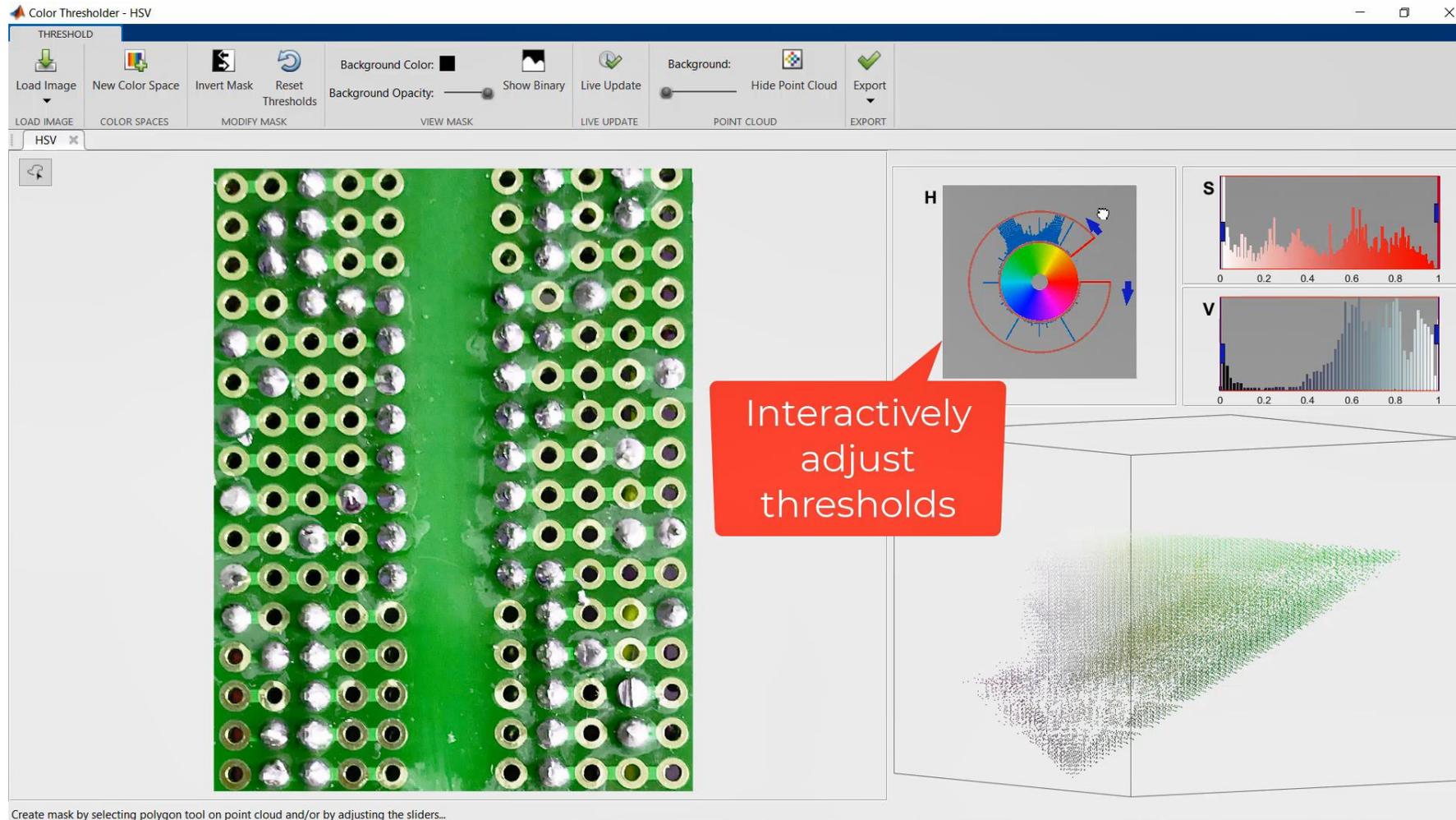
What if I have an imbalanced dataset or don't have enough data?

Data Access and Preprocessing – Common Challenges



How do I preprocess data
and get the right features?

Pre-processing Data – Color Thresholder App



Pre-processing Data – Region Analyzer App

Image Region Analyzer

EXPLORE

Load Image Exclude Border Fill Holes Filter Choose Properties Export

LOAD IMAGE ADD/REMOVE PROPERTIES EXPORT

Analyze Regions

Region Properties

Area	Eccentricity	EquivDiameter	EulerNumb
11798	0.3234	122.5630	
11371	0.3580	120.3246	
11121	0.3373	118.9945	
11083	0.3431	118.7911	
10900	0.2958	117.8062	
10518	0.3223	115.7235	
10394	0.3016	115.0393	
10361	0.3270	114.8566	
10233	0.3655	114.1449	
10229	0.2794	114.1226	
10150	0.2952	113.6811	
10142	0.2916	113.6362	
10034	0.2963	113.0296	
9972	0.3448	112.6798	
9861	0.3371	112.0509	
9778	0.2262	111.5784	
9774	0.3036	111.5556	
9717	0.2984	111.2298	
9714	0.2451	111.2126	
9711	0.2480	111.1955	
9601	0.3608	110.5639	
9578	0.2517	110.4314	
9567	0.3175	110.3679	
9531	0.3487	110.1601	
9461	0.3335	109.7548	
9431	0.2898	109.5807	
9390	0.3146	109.3422	
9327	0.3527	108.9748	

Visualize regions by property

Pre-processing Data – Registration Estimator App

Registration Estimator - 01_missing_hole_01.jpg (Moving Image) & 01_missing_hole_01.jpg (Fixed Image)

REGISTRATION

Load Images | SURF | FAST | BRISK | Harris | Register Images | Overlay Style: Green-Magenta | Export

ID	Technique	Quality	Time	Features Matched	Fea
1	Phase Correlation	DRAFT			
2	MSER	DRAFT		1965	110
3	SURF	DRAFT		2870	469

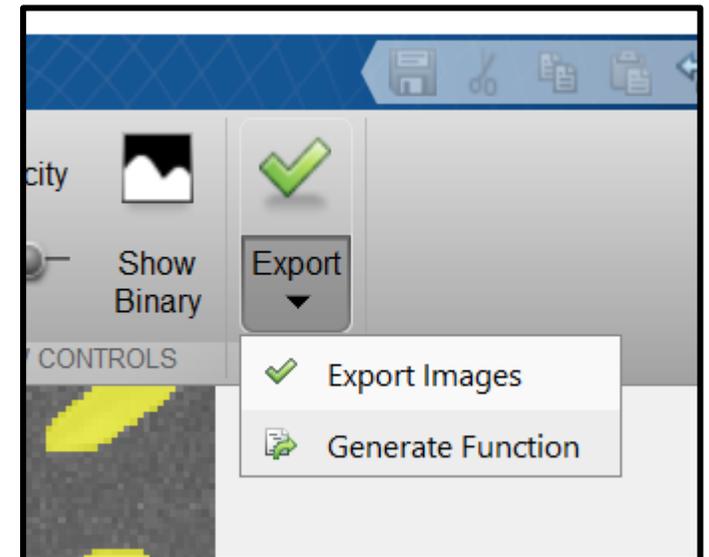
Image Overlay

Parameters

- Feature Parameters
 - Projective Transformation
 - Number of Detected Features
 - Quality of Matched Features
 - Has Rotation
 - Post-processing

Try different feature extraction techniques

Apps Accelerate Workflow



Other Filtering and Enhancement Functions

Help Center Search Help

CONTENTS

- « Documentation Home
- « Functions
- « Image Processing Toolbox

Category

- Image Filtering and Enhancement** 146
 - Image Filtering 40
 - Contrast Adjustment 18
 - ROI-Based Processing 37
 - Morphological Operations 31
 - Deblurring 8
 - Neighborhood and Block Processing 7
 - Image Arithmetic 8
- Image Segmentation and Analysis 108
- Deep Learning for Image Processing 33
- 3-D Volumetric Image Processing 75

Extended Capability

- C/C++ Code Generation 54
- GPU Code Generation 28
- Automatic Parallel Support 1
- GPU Arrays 26

Documentation Examples **Functions** Apps Videos Answers

<code>ordfilt2</code>	2-D order-statistic filtering
<code>stdfilt</code>	Local standard deviation of image
<code>rangefilt</code>	Local range of image
<code>entropyfilt</code>	Local entropy of grayscale image
<code>imboxfilt</code>	2-D box filtering of images
<code>imboxfilt3</code>	3-D box filtering of 3-D images
<code>fibermetric</code>	Enhance elongated or tubular structures in image using Frangi vesselness filter
<code>maxhessiannorm</code>	Maximum of Frobenius norm of Hessian of matrix
<code>padarray</code>	Pad array

Edge-Preserving Filtering

<code>imbilatfilt</code>	Bilateral filtering of images with Gaussian kernels
<code>imdiffuseest</code>	Estimate parameters for anisotropic diffusion filtering
<code>imdiffusefilt</code>	Anisotropic diffusion filtering of images
<code>imguidedfilter</code>	Guided filtering of images
<code>imnlmfilt</code>	Non-local means filtering of image
<code>burstinterpolant</code>	Create high-resolution image from set of low-resolution burst mode images

Texture Filtering

<code>gabor</code>	Create Gabor filter or Gabor filter bank
<code>imgaborfilt</code>	Apply Gabor filter or filter bank to 2-D image

Filtering By Property Characteristics

<code>bwareafilt</code>	Extract objects from binary image by size
<code>bwpropfilt</code>	Extract objects from binary image using properties

Integral Image Domain Filtering

<code>integralImage</code>	Calculate 2-D integral image
<code>integralImage3</code>	Calculate 3-D integral image
<code>integralBoxFilter</code>	2-D box filtering of integral images
<code>integralBoxFilter3</code>	3-D box filtering of 3-D integral images

*imgaborfilt(A, wavelength, orientatio
n)*

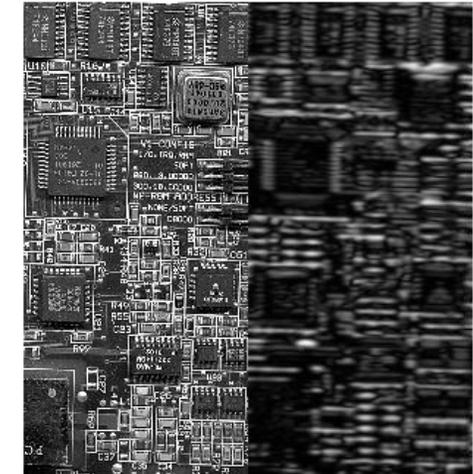


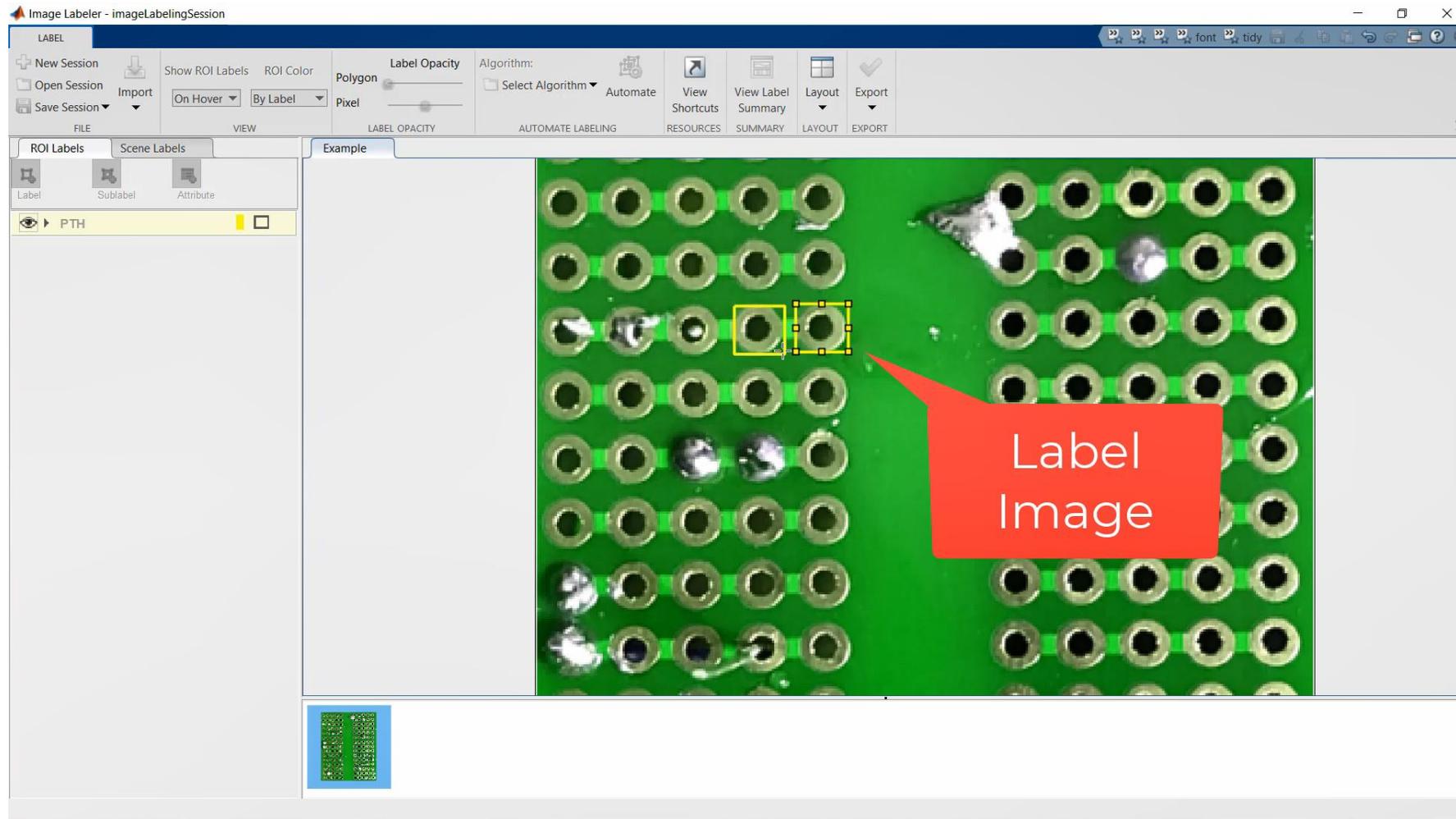
Image Filtering and Enhancement Functions

Data Access and Preprocessing – Common Challenges



How do I label my data faster?

How do I label my data?

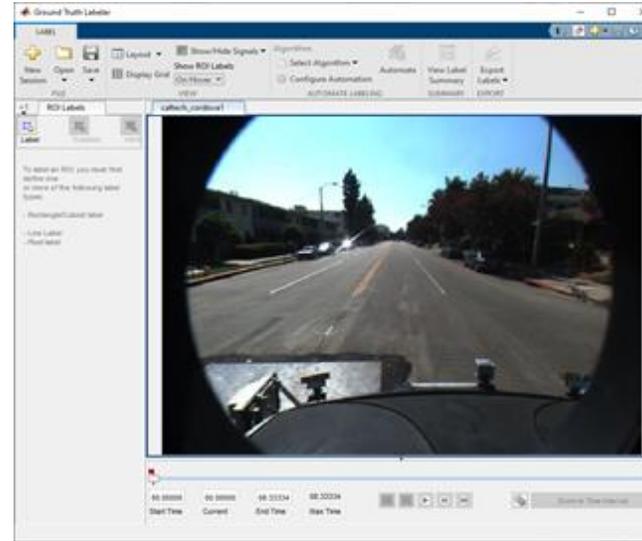


How do I label my data?

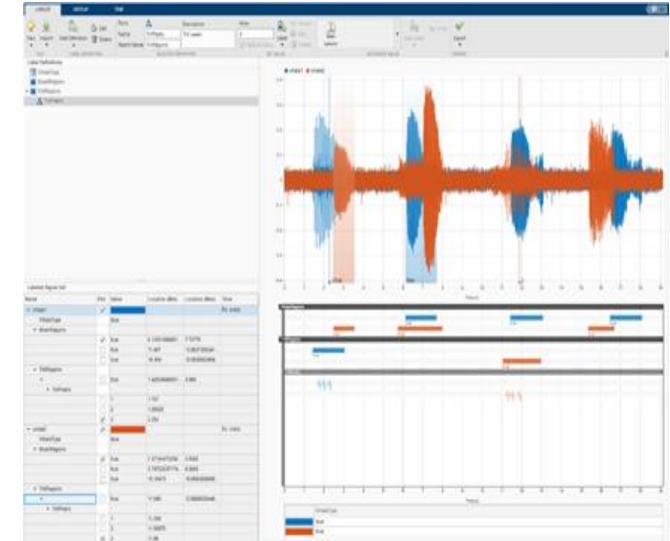
imageLabeler



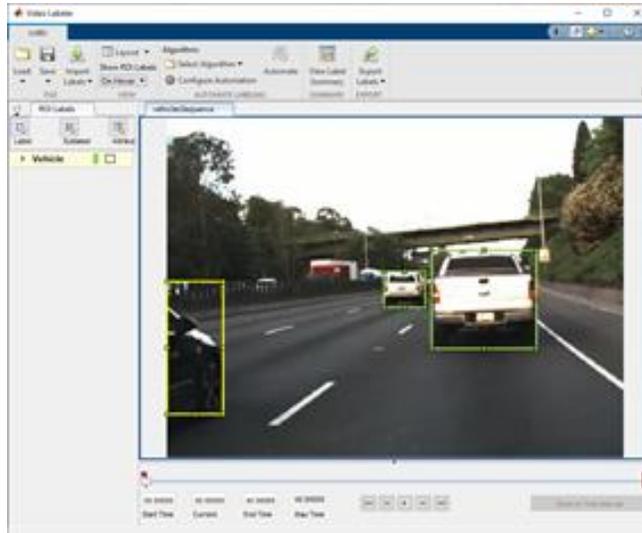
groundTruthLabeler



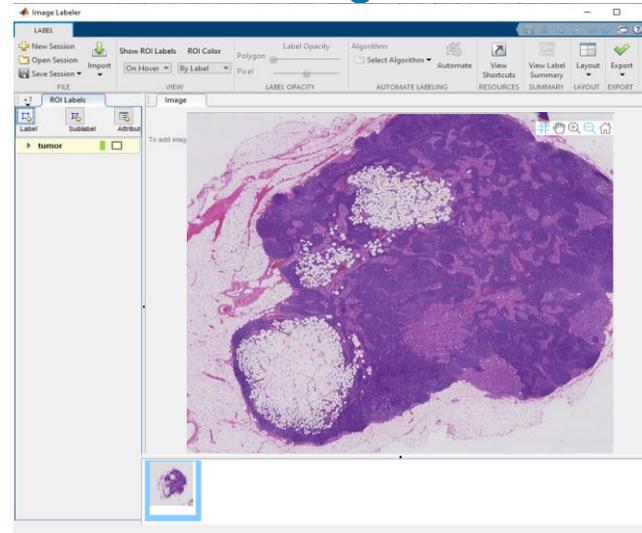
signalLabeler



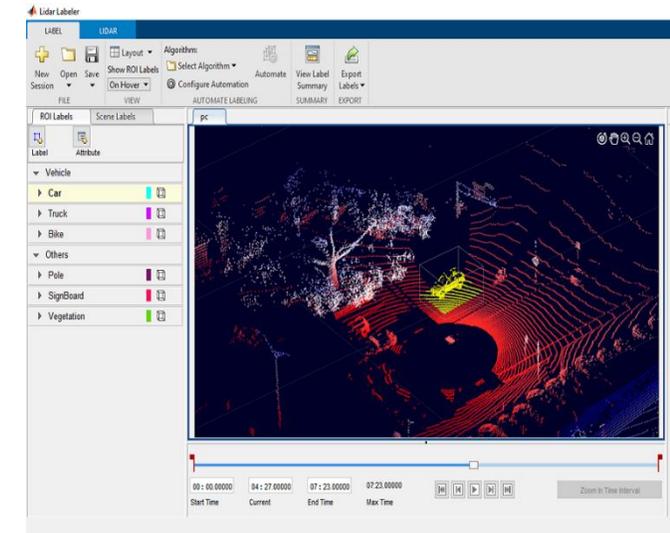
videoLabeler



blockedImage Labeler



lidarLabeler

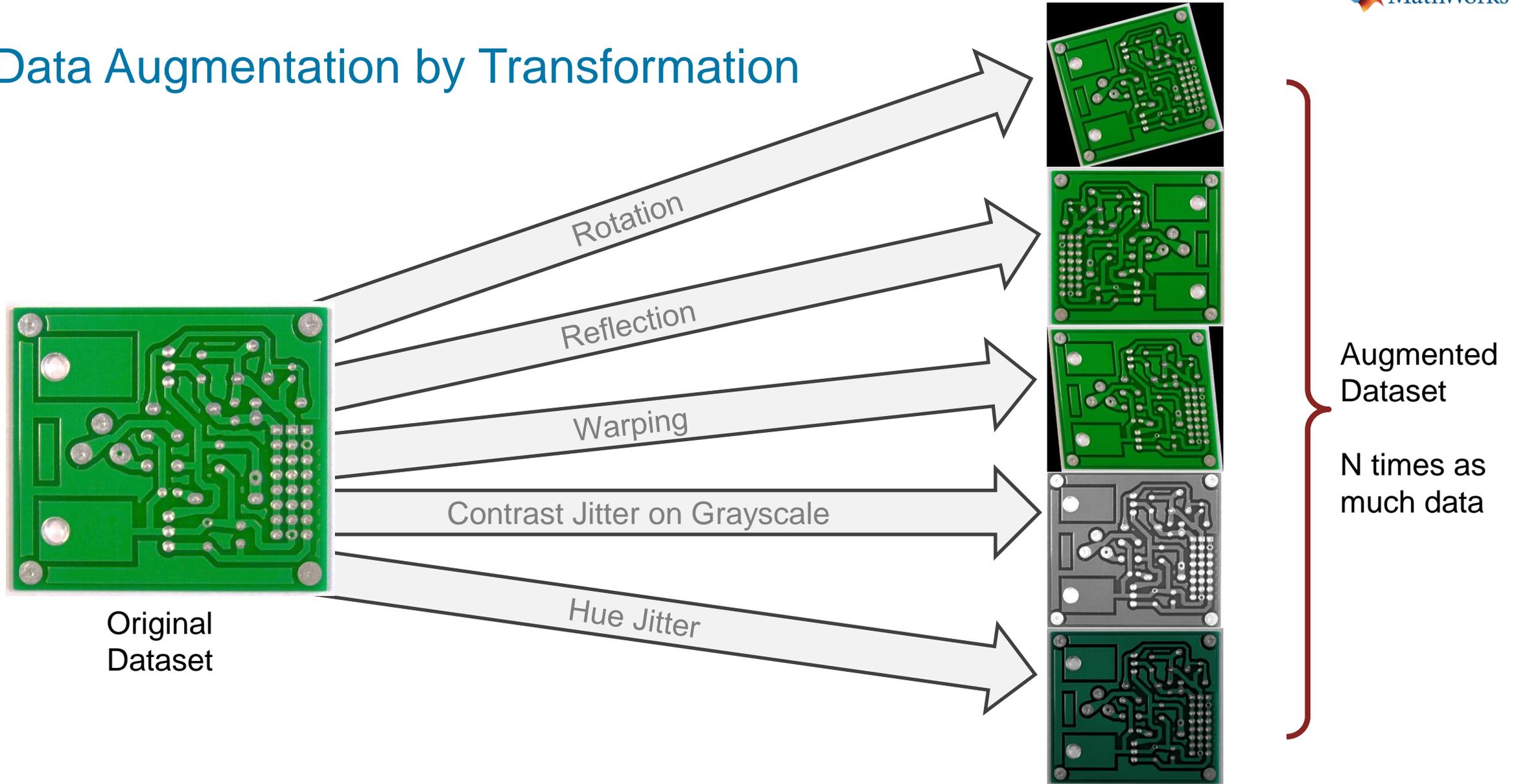


Data Access and Preprocessing – Common Challenges



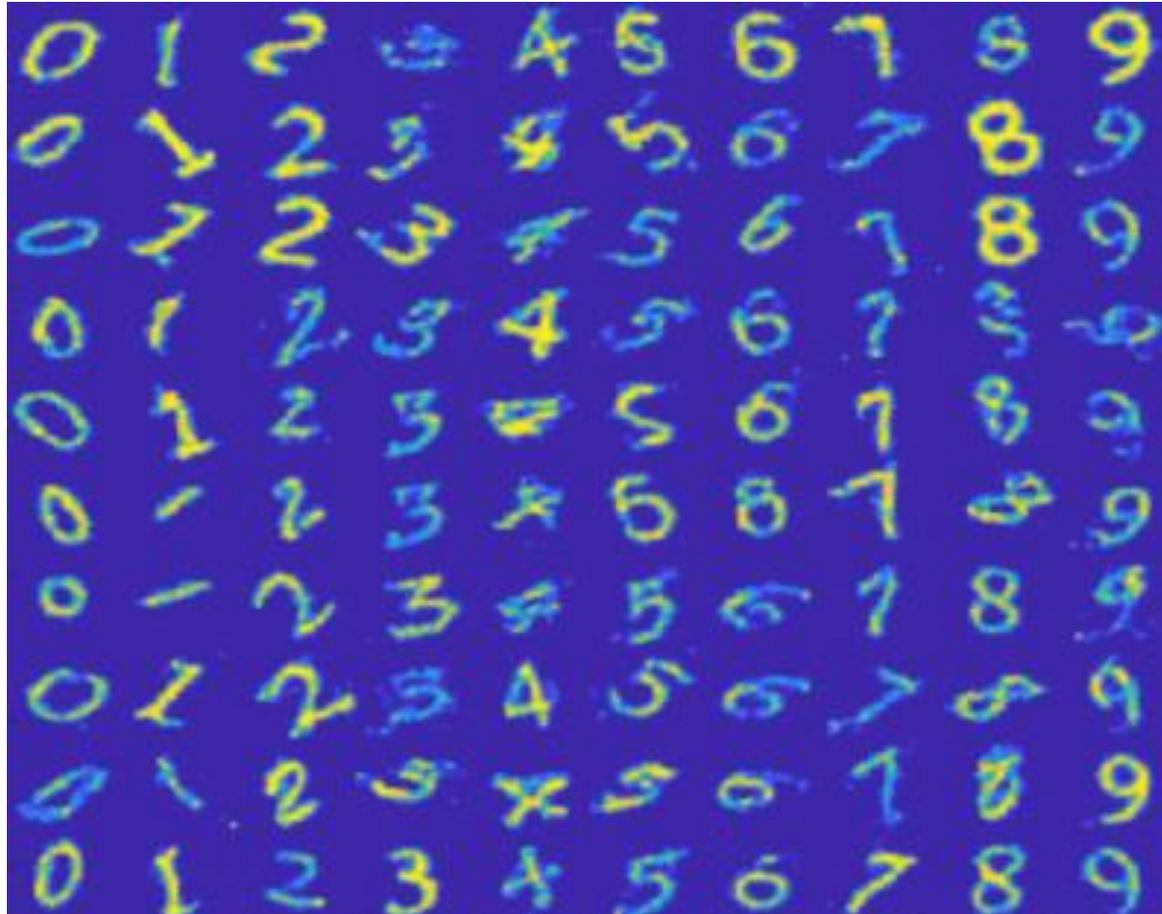
What if I have an imbalanced dataset or don't have enough data?

Data Augmentation by Transformation



Data augmentation allows building more complex and more robust models

Data Augmentation using Generative Adversarial Networks (GANs)



Images of digits generated from noise.

[File Exchange: Conditional GAN \(Generative Adversarial Network\) with MNIST](#)

Defect Detection Workflow

Data Preparation

 Data cleansing and preparation

 Human insight

 Simulation-generated data

AI Modeling

 Model design and tuning

 Hardware accelerated training

 Interoperability

Simulation & Test

 Integration with complex systems

 System simulation

 System verification and validation

Deployment

 Embedded devices

 Enterprise systems

 Edge, cloud, desktop

Types of Defect Detection

Center 	Donut 	Edge-Loc
Edge-Ring 	Loc 	Near-full
Random 	Scratch 	none

Defect Classification

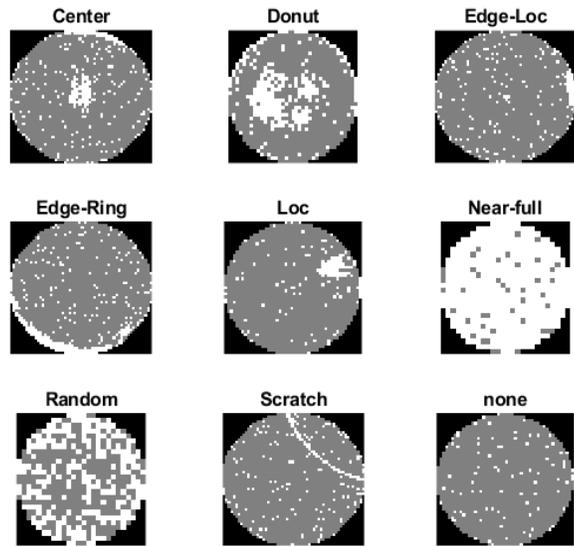
Original Image 	Anomaly Score Heatmap
--------------------	---------------------------

Anomaly Detection

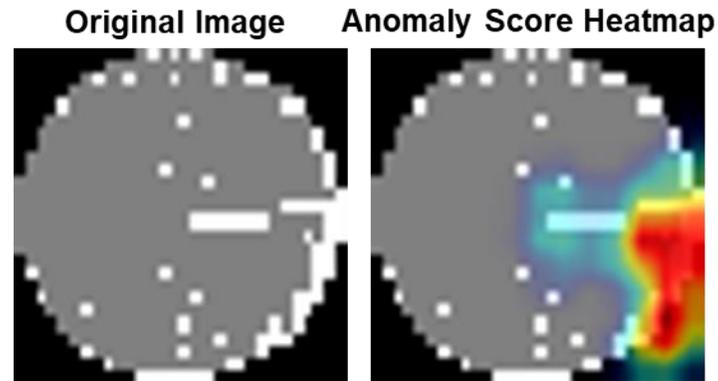
(a) PCB with mouse bite.

Counting, Presence and Localization of Objects

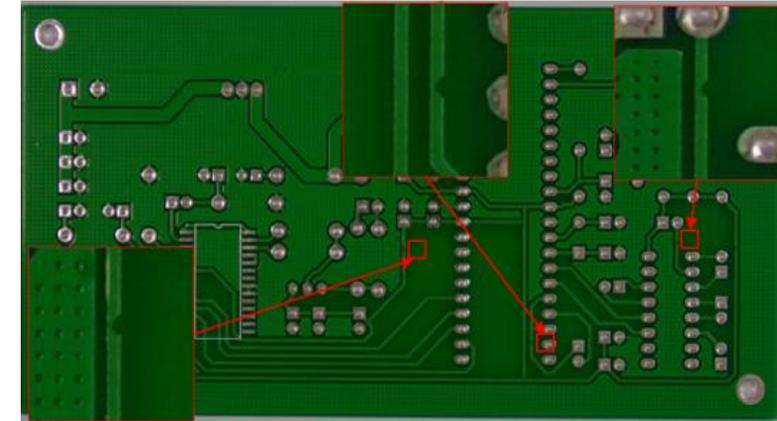
Defect Classification



Defect Classification

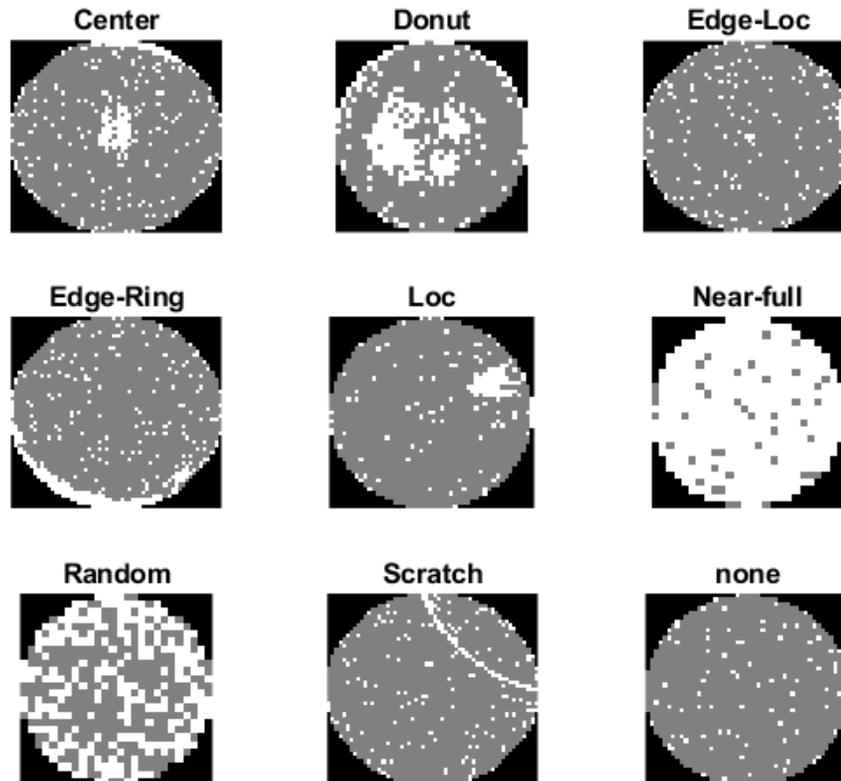


Anomaly Detection



Counting, Presence and Localization of Objects

WM-811K Wafer Defect Map Data



```
summary(waferData.FailureType)
```

Center	4294
Donut	555
Edge-Loc	5189
Edge-Ring	9680
Loc	3593
Near-full	149
Random	866
Scratch	1193
none	147431

Live Editor - C:\Content - nonsync\Examples\R2022a\images_deeplearning\ClassifyWaferMapDefectsUsingDeepLearningExample\ClassifyWaferMapDefectsUsingDeepLearningExample.mlx

LIVE EDITOR INSERT VIEW

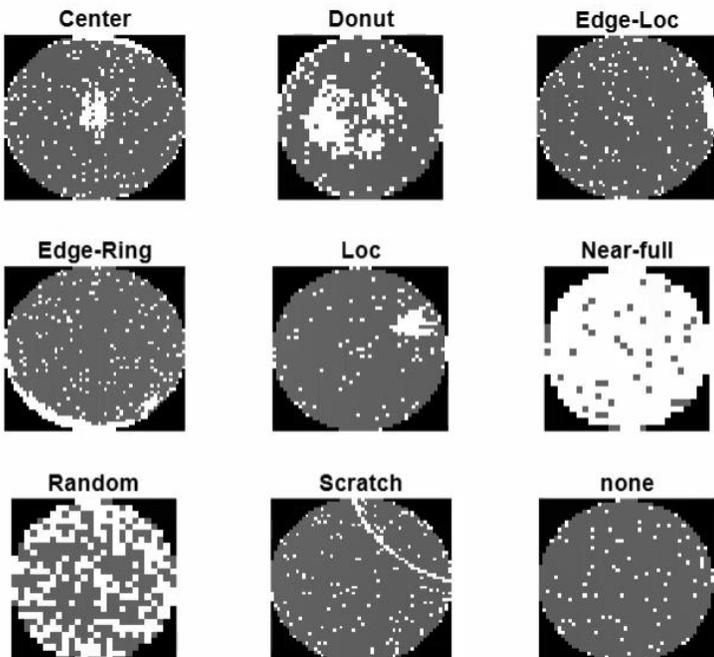
New Open Save Print Export Compare Go To Find Bookmark Text B I U M Code Control Task Refactor Run Section Run and Advance Run to End Run Step Stop

```
waferData.FailureType = categorical(waferData.FailureType,defectClasses);
```

20

Display a sample image from each input image class using the `displaySampleWaferMaps` helper function. This function is attached to the example as a supporting file.

```
displaySampleWaferMaps(waferData)
```



Center Donut Edge-Loc

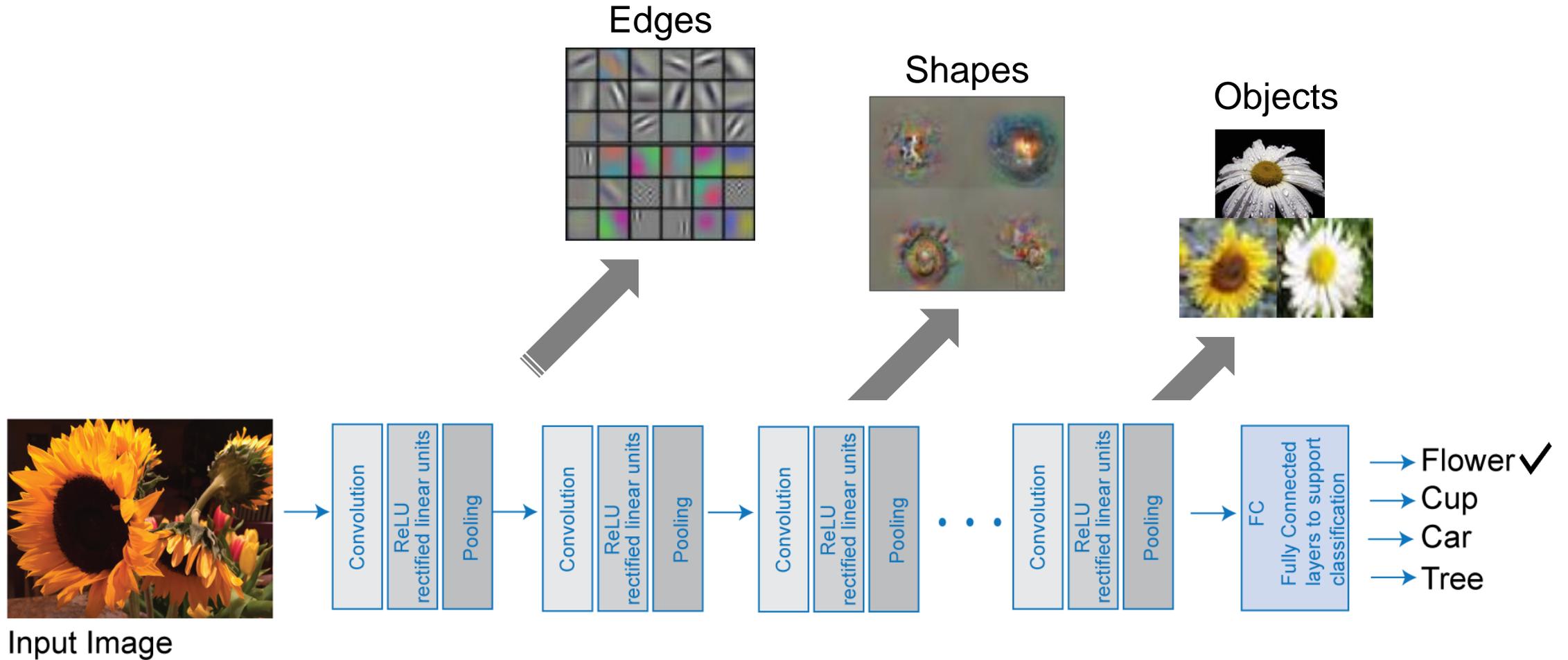
Edge-Ring Loc Near-full

Random Scratch none

Explore dataset

Zoom: 125% UTF-8 LF script Ln 20 Col 34

Convolutional Neural Networks (CNN)



Deep Network Designer

DESIGNER

+ New Duplicate Cut Copy Paste Fit to View Zoom In Zoom Out Auto Arrange Analyze Export

NETWORK BUILD NAVIGATE LAYOUT ANALYSIS EXPORT

Layer Library

Filter layers...

- INPUT
 - imageInputLayer
 - image3dInputLayer
 - sequenceInputLayer
 - featureInputLayer
 - roiInputLayer
- CONVOLUTION AND FULLY CONNECTED
 - convolution1dLayer
 - convolution2dLayer
 - convolution3dLayer
 - groupedConvolution2dLayer
 - transposedConv1dLayer
 - transposedConv2dLayer
 - transposedConv3dLayer
 - fullyConnectedLayer
- SEQUENCE
 - lstmLayer
 - biLstmLayer

Designer Data Training

Visualize network architecture

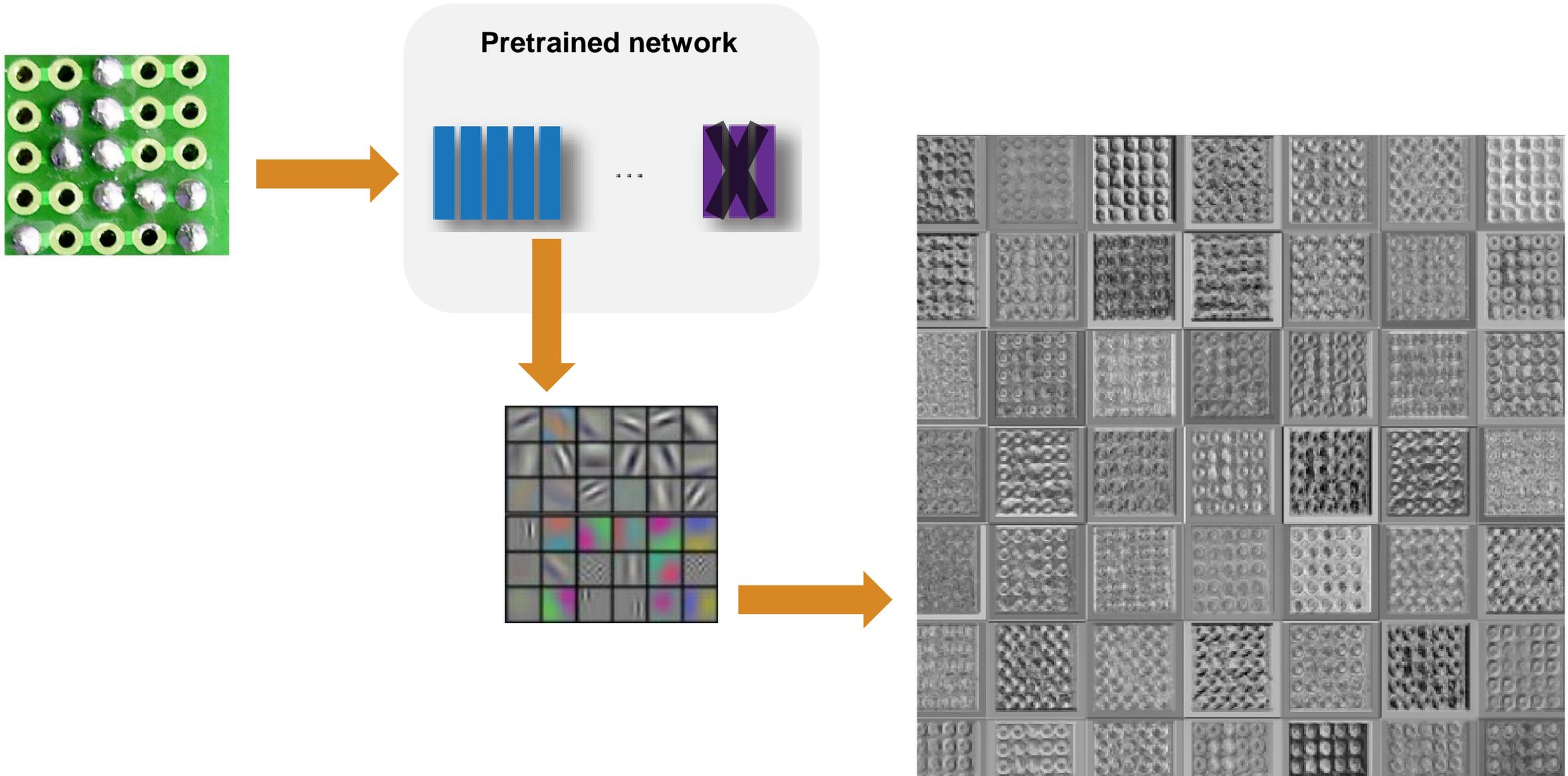
conv_1 convolution2dL...
 batchnorm_1 batchNormaliza...
 relu_1 reluLayer
 maxpool_1 maxPooling2dL...
 conv_2 convolution2dL...
 batchnorm_2 batchNormaliza...
 relu_2 reluLayer
 maxpool_2 maxPooling2dL...

Properties

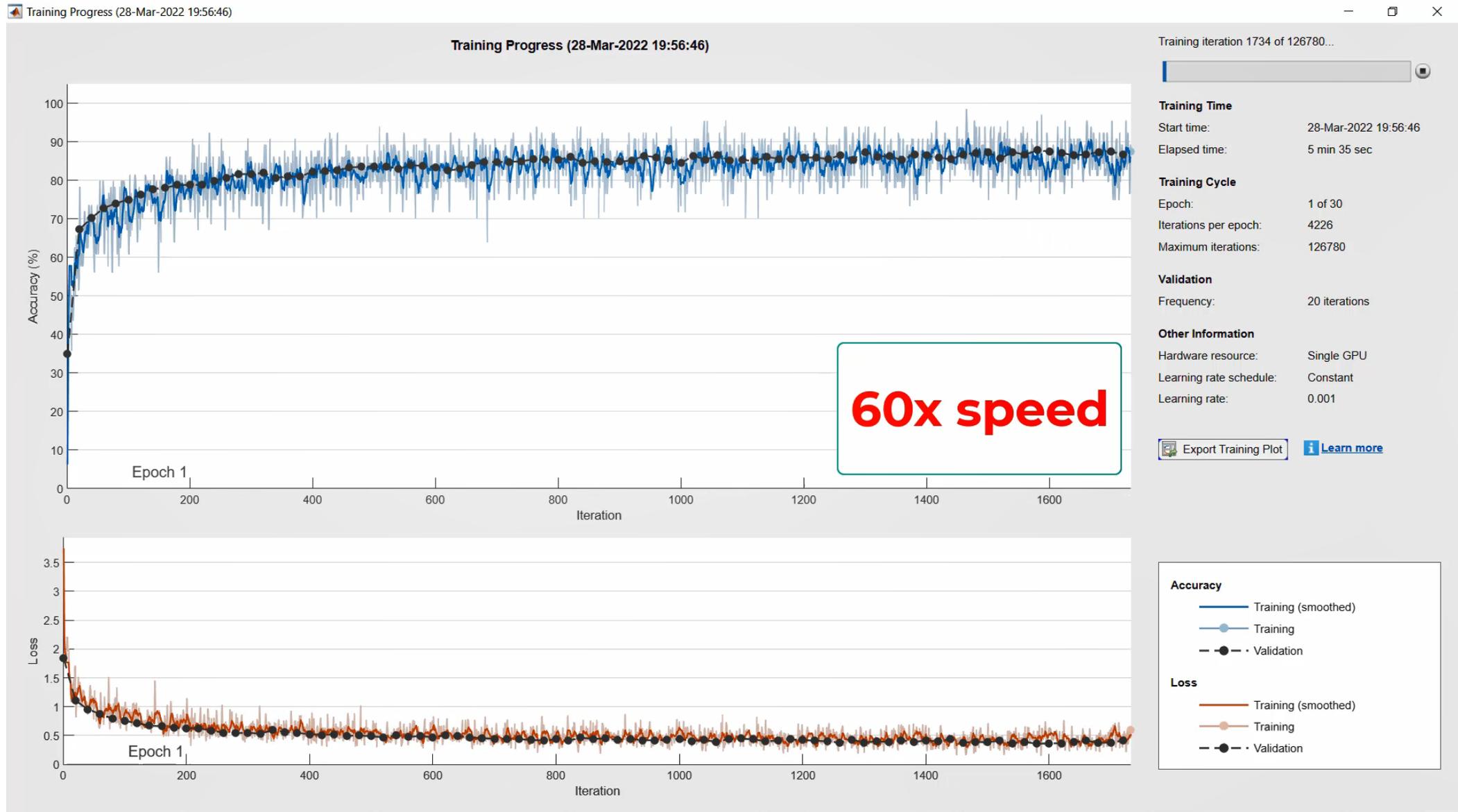
Input type	Image
Output type	Classification
Number of layers	20
Number of connections	19

Overview

Approach to find the defective units



Training the network with GPU



Evaluating Network Performance

Test Data Confusion Matrix

True Class \ Predicted Class	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-full	Random	Scratch	none
Center	91.7%	0.5%	0.1%	0.5%	0.8%		1.2%		5.3%
Donut	1.2%	82.5%	1.2%		11.4%		3.0%		0.6%
Edge-Loc	0.4%		79.5%	12.1%	1.2%		1.6%	0.1%	5.1%
Edge-Ring			0.2%	98.8%			0.1%		1.0%
Loc	2.7%	0.7%	12.9%	0.6%	66.4%		1.7%	1.0%	14.0%
Near-full	2.2%		2.2%			77.8%	17.8%		
Random	0.4%	0.8%	1.2%	1.2%	1.9%		94.6%		
Scratch	0.6%	0.3%	4.2%	2.0%	5.6%		0.3%	49.4%	37.7%
none	0.2%		1.5%	1.4%	0.2%		0.1%	0.0%	96.6%

Anomaly Detection

Center 	Donut 	Edge-Loc
Edge-Ring 	Loc 	Near-full
Random 	Scratch 	none

Defect Classification

Original Image 	Anomaly Score Heatmap
--------------------	---------------------------

Anomaly Detection

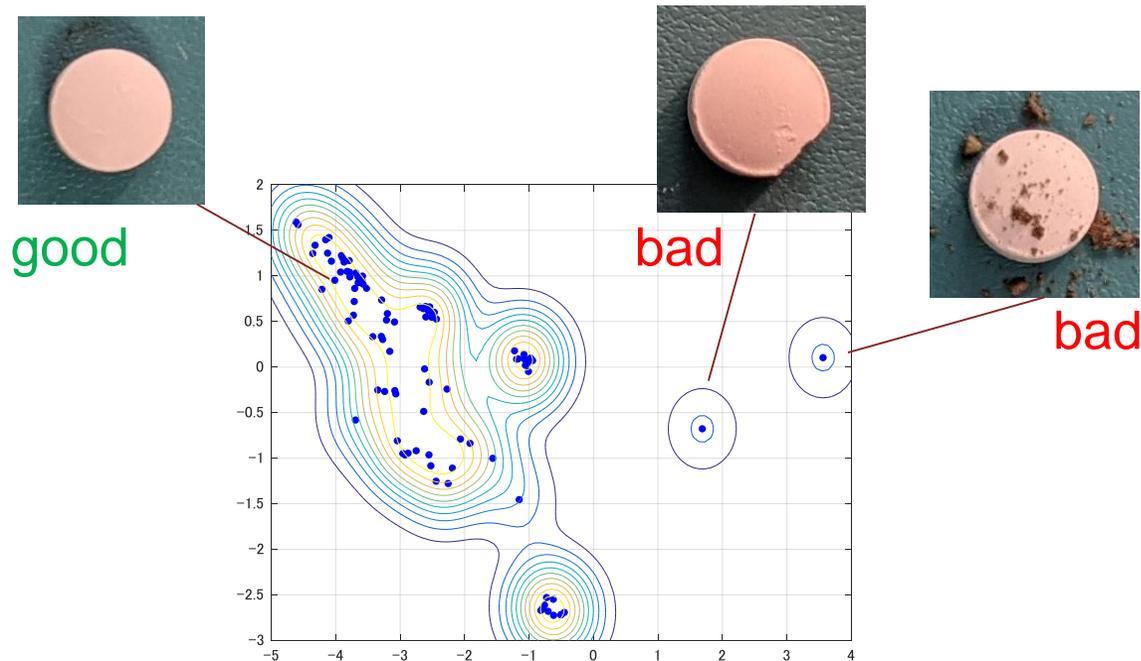
(a) PCB with mouse bite.

Counting, Presence and Localization of Objects

[Example: Detect Image Anomalies Using Explainable One-Class Classification Neural Network](#)

Anomaly detection for an imbalanced dataset

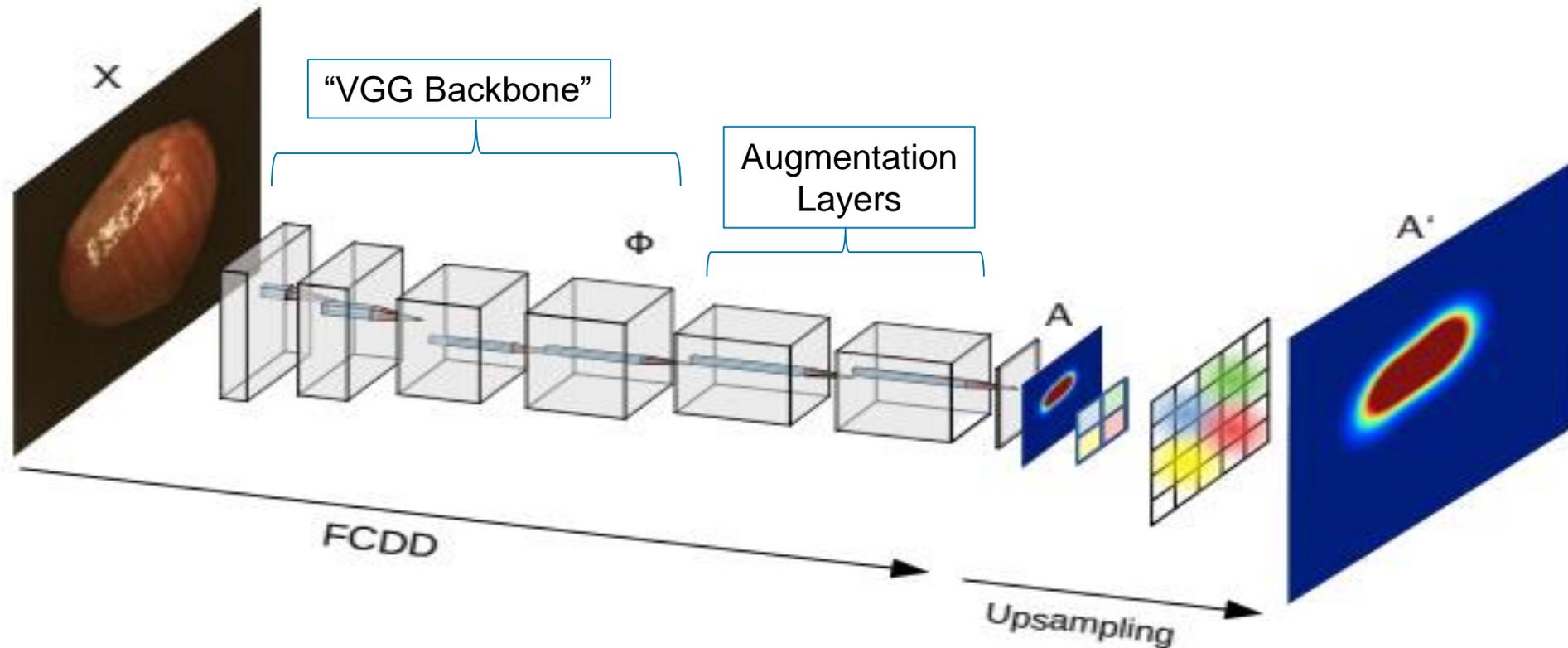
- Use **only good images** at training.
- Automatically sets **optimal anomaly threshold**
- Predict images as **good/bad** at inference time



Supported Anomaly Detection Method

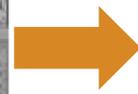
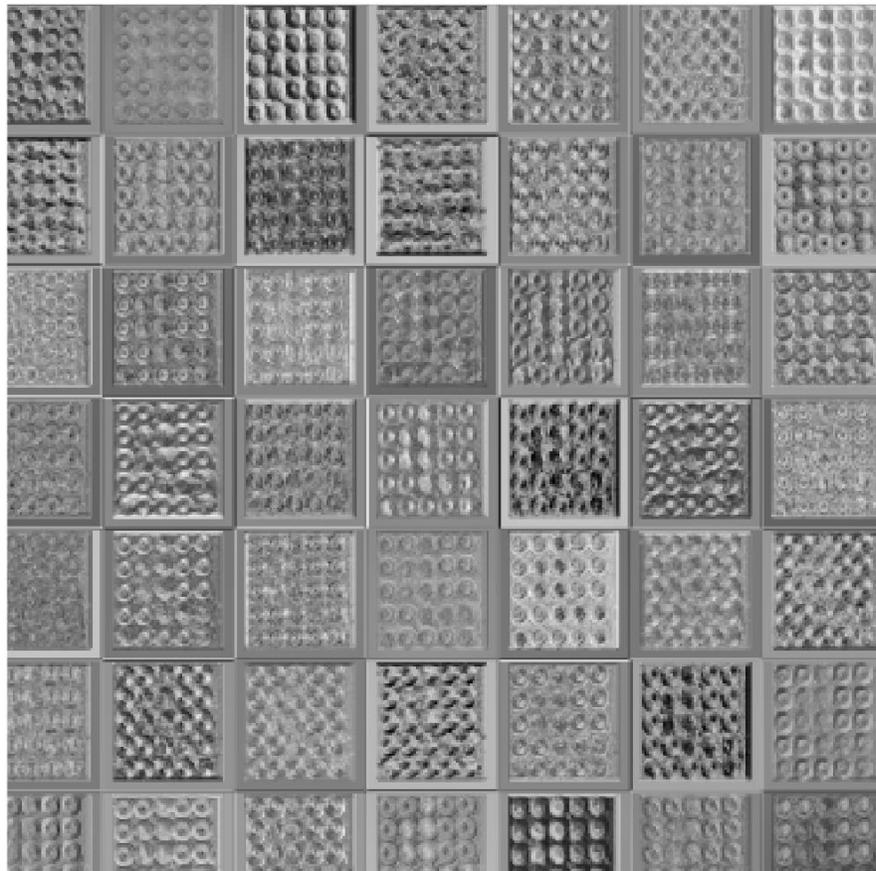
- FCDD
- FastFlow
- PatchCore
- PaDiM
- 1-class SVM

Anomaly Detection Using Fully Convolutional Data Description (FCDD)

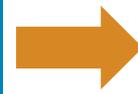


- Quick-to-train, and requires very few anomalous samples
- Auto-generates a heatmap

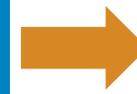
Approach to find the defective units



Feature
vector



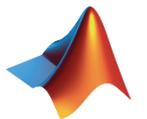
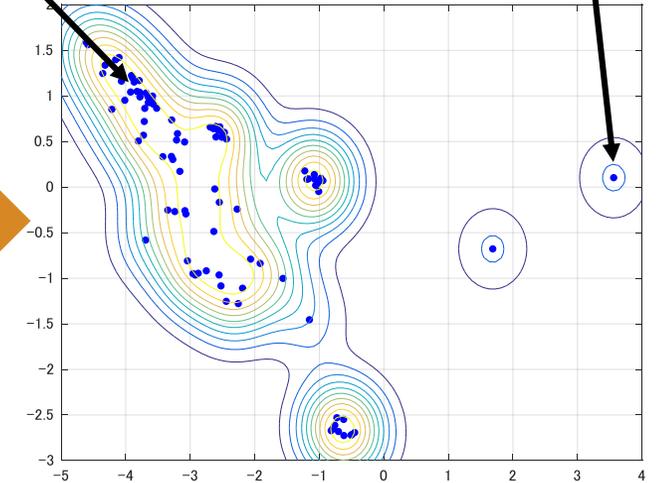
One-class
SVM



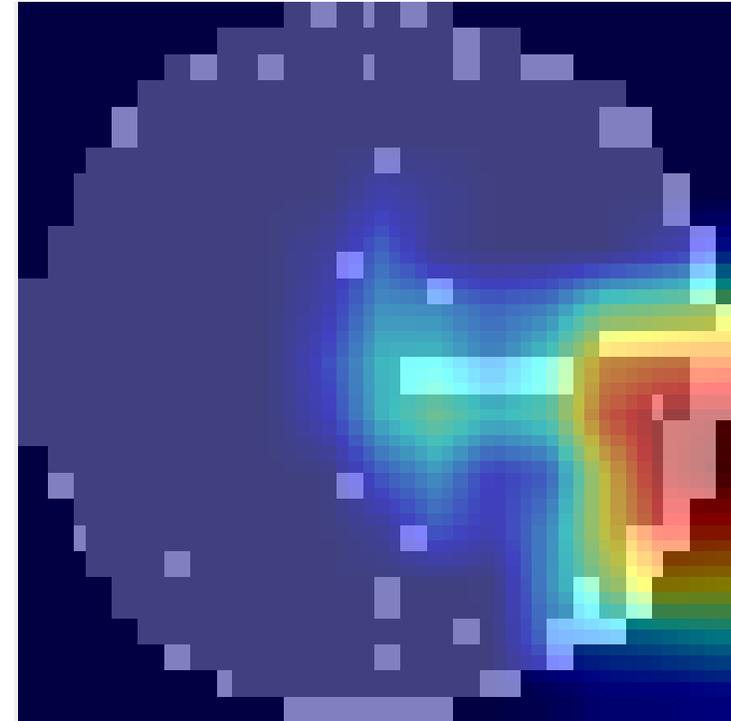
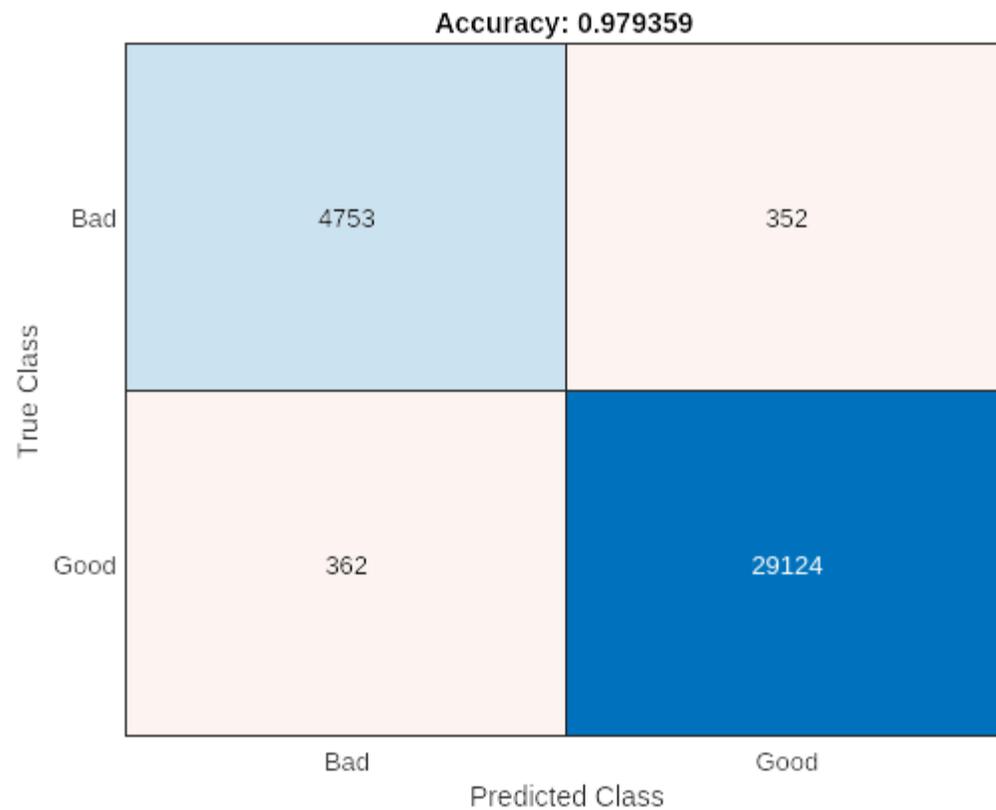
High score



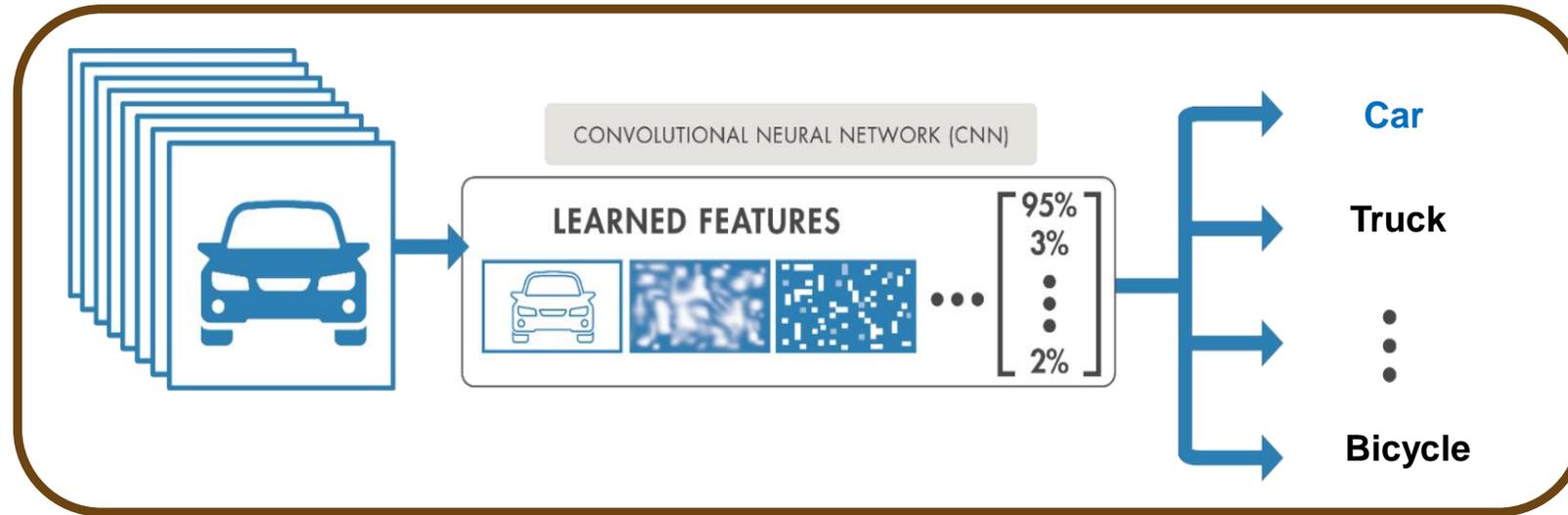
Low score



wm811k: Wafer Anomaly Detection Using Fully Convolutional Data Description (FCDD)



Challenges with Deep Learning Models

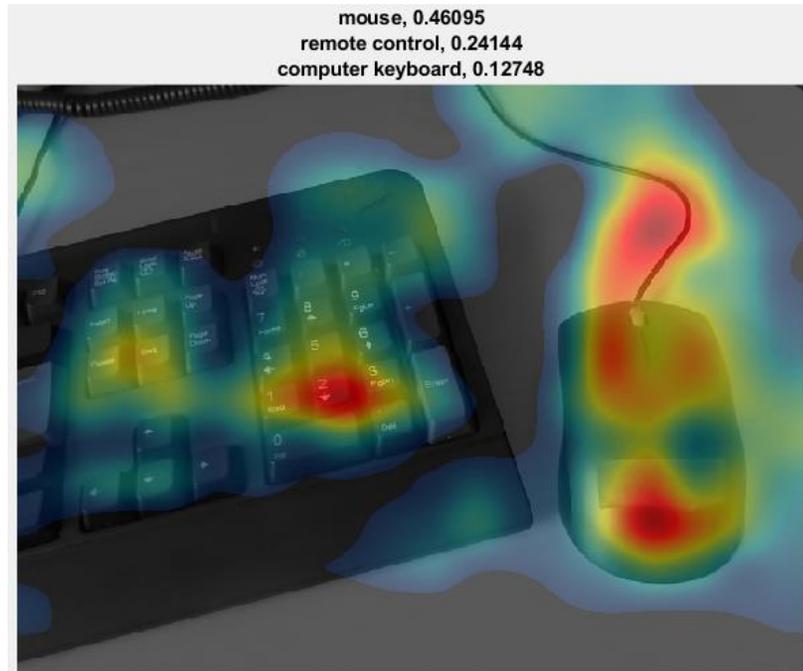


Anomaly map helps to understand why AI says it's a defect

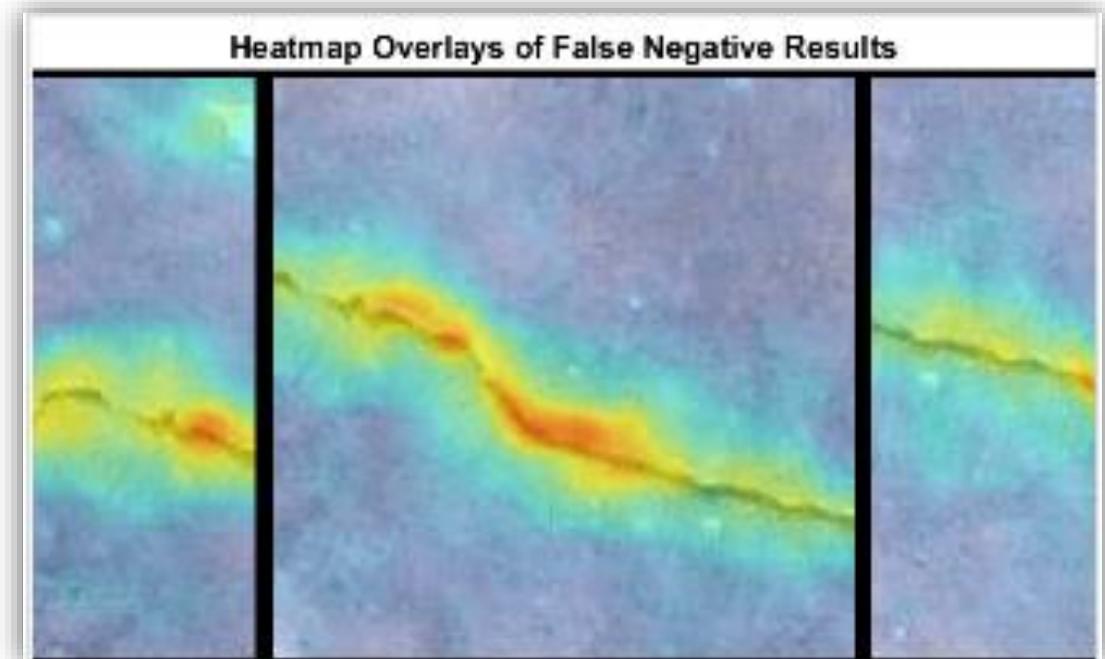


- [Class Activation Mapping \(CAM\)](#)
- [Grad-CAM](#)

Anomaly Map to Investigate Network Predictions

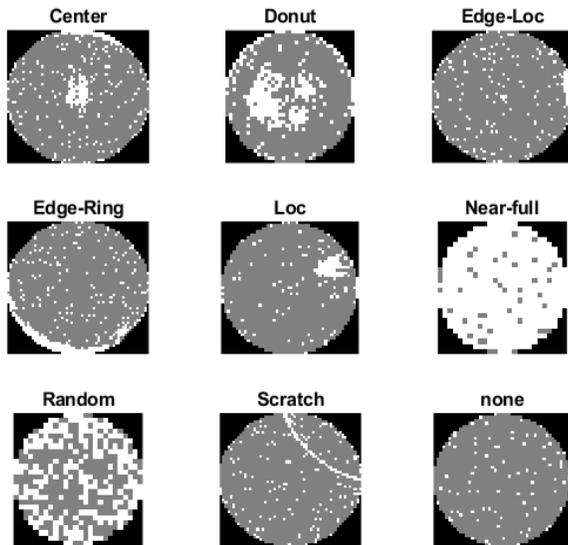


Classified as “keyboard” due to the presence of the mouse

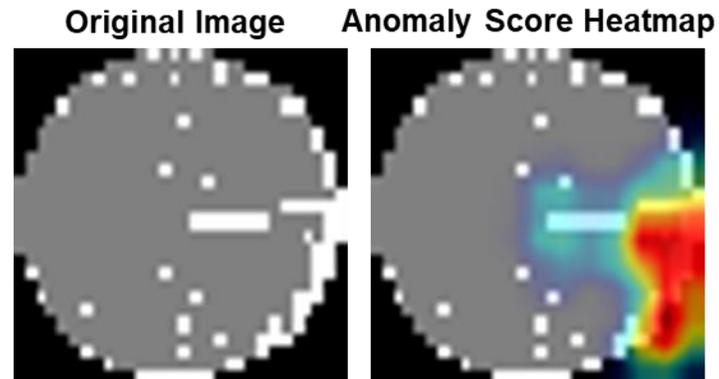


Cracks are highlighted in Anomaly detection

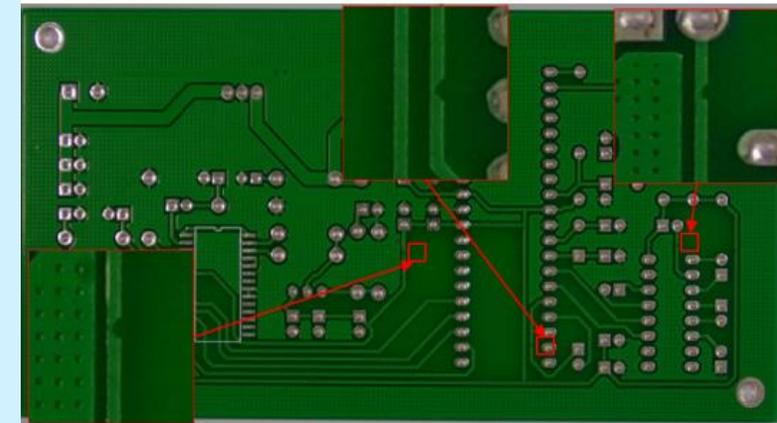
Counting, Presence and Localization of Objects



Defect Classification



Anomaly Detection



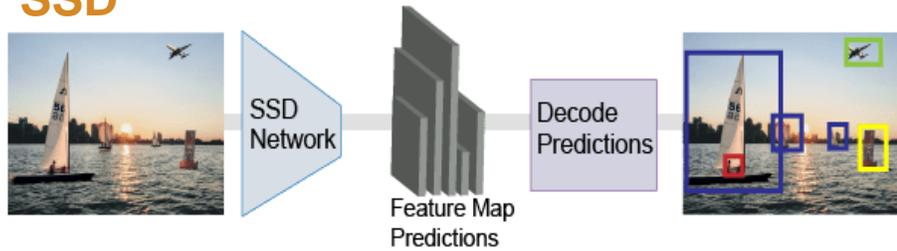
(a) PCB with mouse bite.

Counting, Presence and Localization of Objects

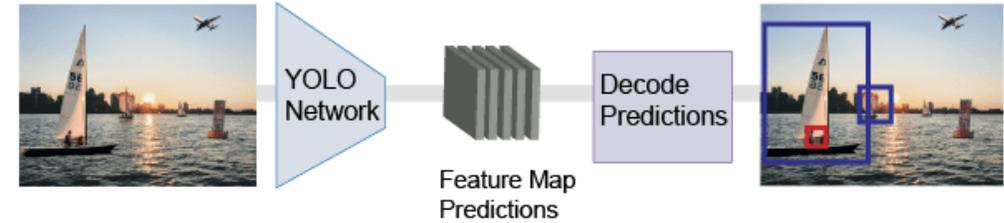
Object Detectors

Build, test, and deploy a deep learning solution that can detect objects in images and video

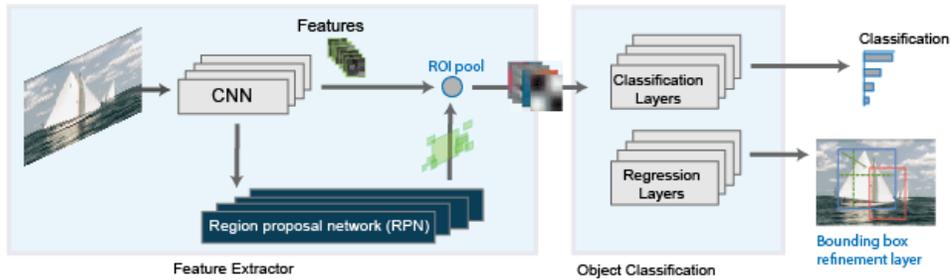
SSD



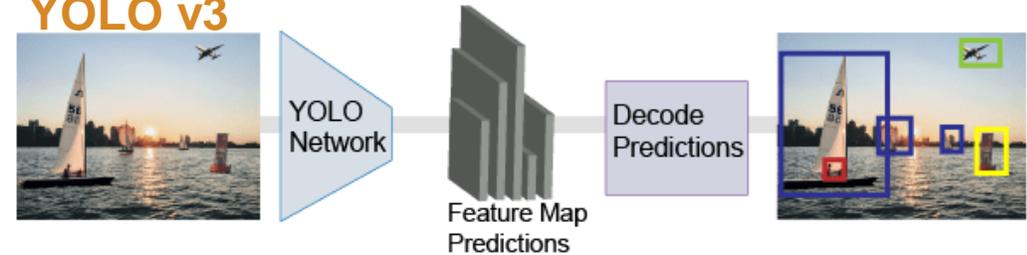
YOLO v2



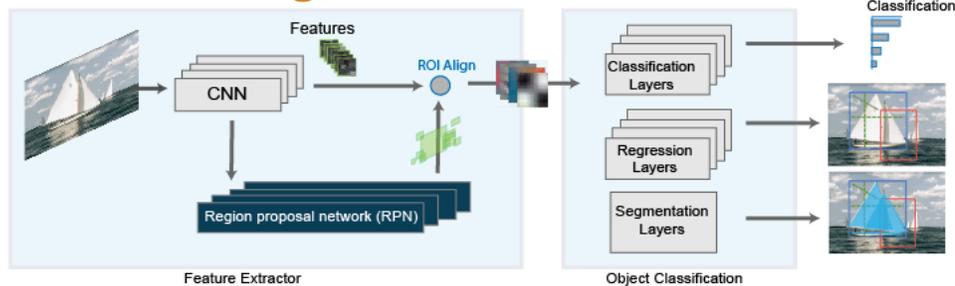
Faster R-CNN



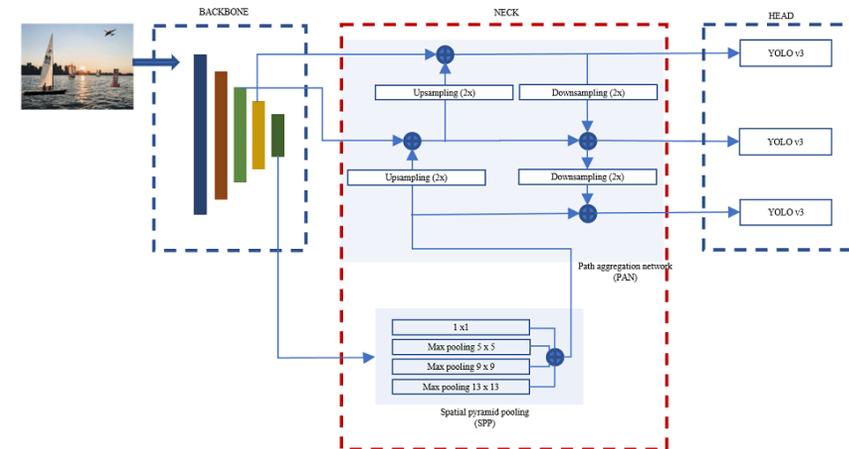
YOLO v3



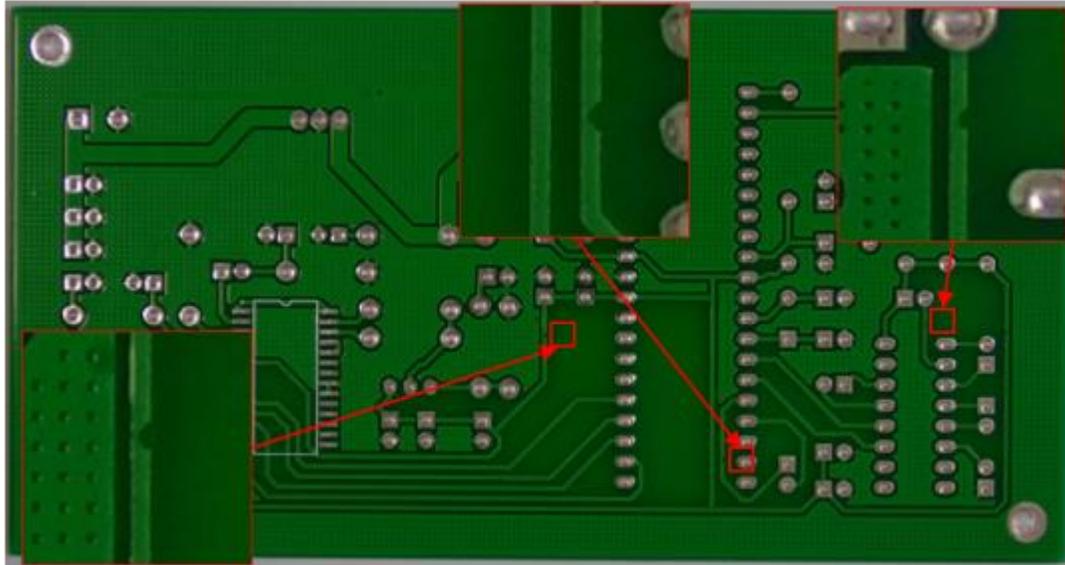
Instance Segmentation



YOLO v4

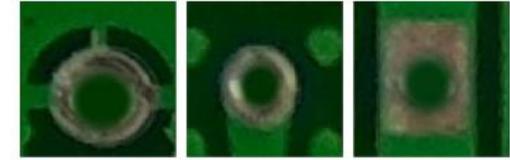


Counting, Presence and Localization of Objects

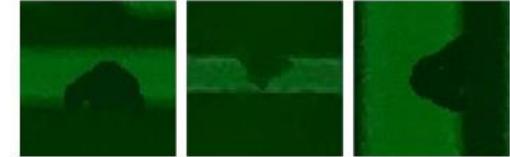


(a) PCB with mouse bite.

Missing hole



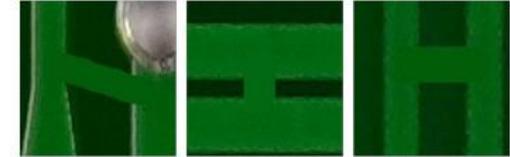
Mouse bite



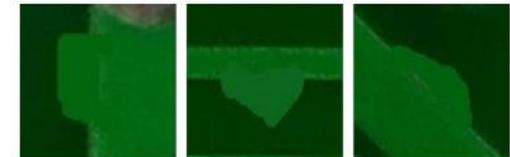
Open circuit



Short



Spur



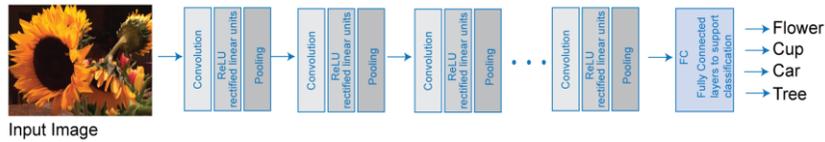
Spurious copper



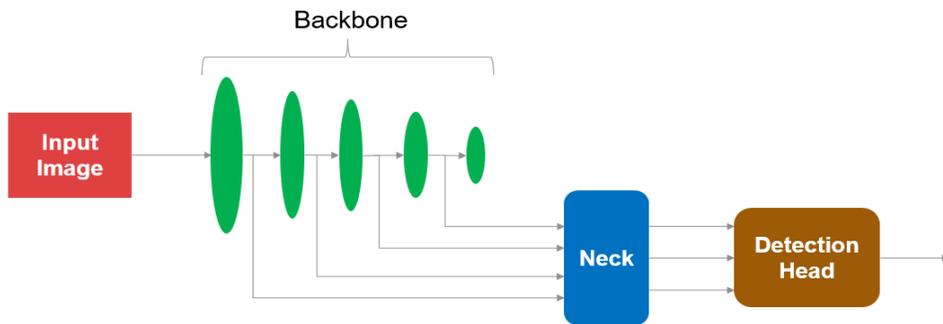
Fig. 11. An example of the training data for neural network. All the resolutions of the images are resized to 64×64 .

Which architecture to choose for Visual Inspection?

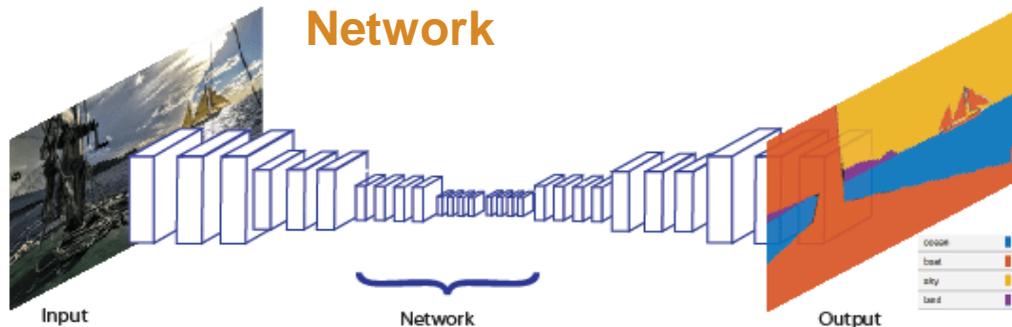
Convolutional Neural Networks



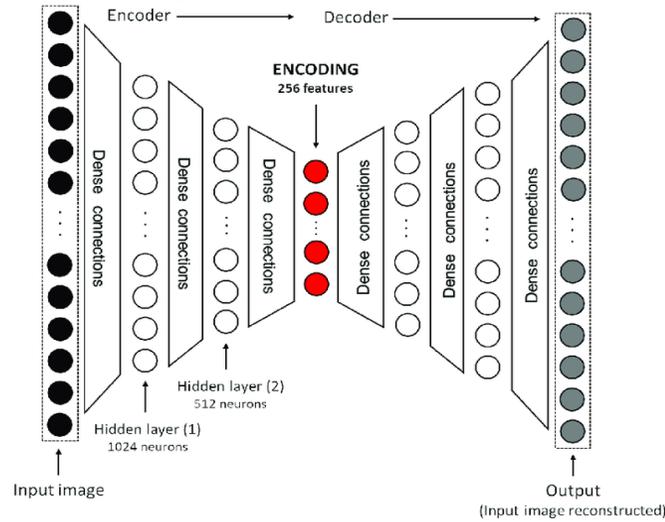
YOLO – Object Detector



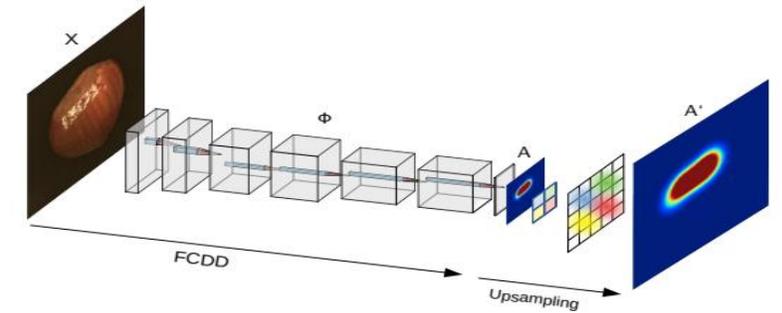
Semantic Segmentation Network



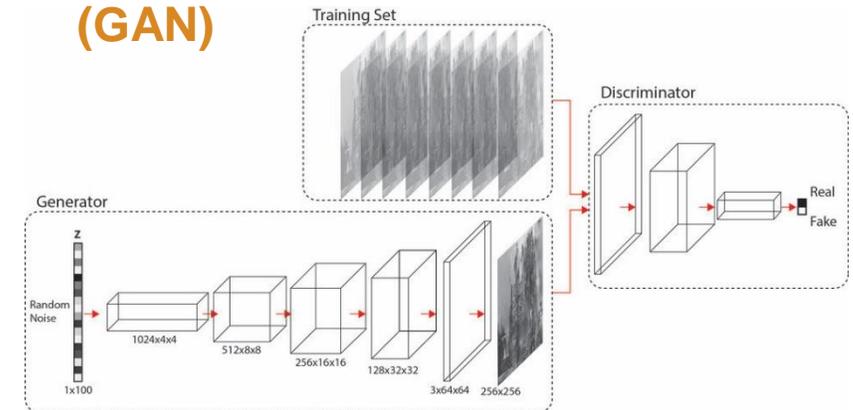
Deep Autoencoder



Fully Convolutional Data Description (FCDD) – Anomaly Detector



Generative Adversarial Network (GAN)



Defect Detection Workflow

Data Preparation

 Data cleansing and preparation

 Human insight

 Simulation-generated data

AI Modeling

 Model design and tuning

 Hardware accelerated training

 Interoperability

Simulation & Test

 Integration with complex systems

 System simulation

 System verification and validation

Deployment

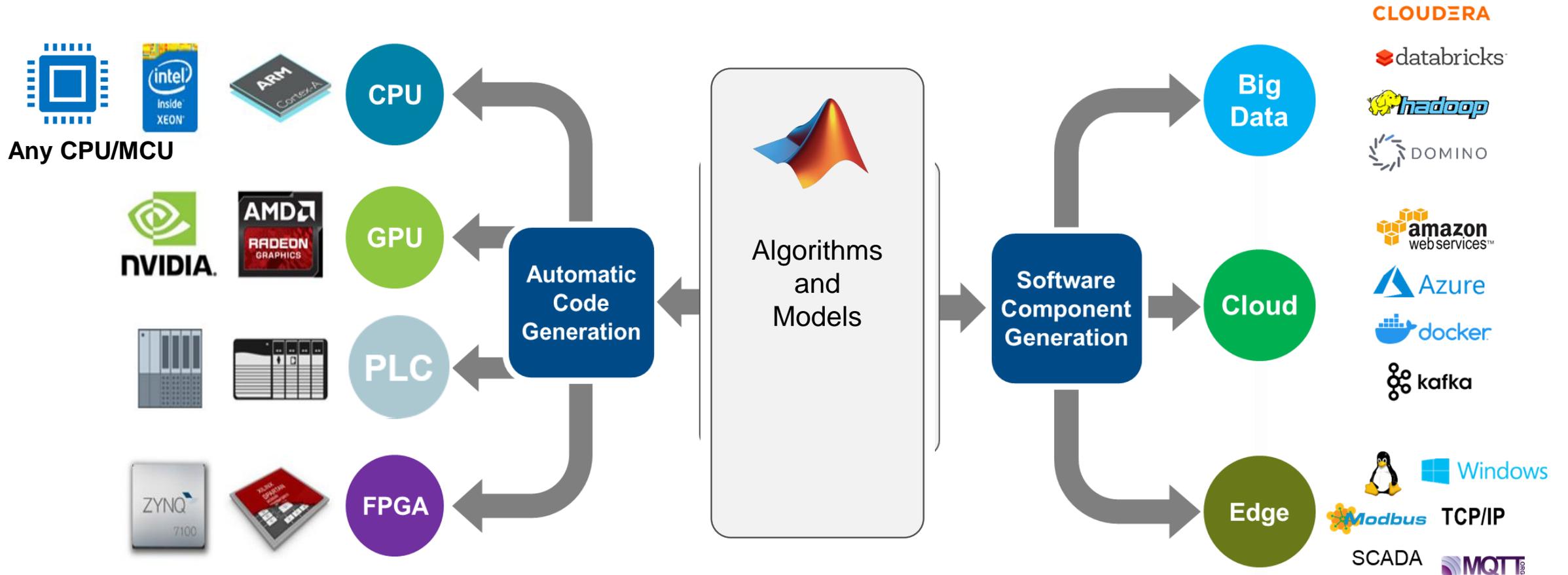
 Embedded devices

 Enterprise systems

 Edge, cloud, desktop

Deploy to Enterprise Infrastructure or Embedded Systems

AI models in MATLAB and Simulink can be deployed on enterprise systems or the cloud, or on embedded devices.



External Language Interfaces

Calling Libraries in Other Languages

C++ Libraries in MATLAB

Directly call C++ library functionality from MATLAB

C Libraries in MATLAB

Directly call C library functions from MATLAB

MEX File Functions

Call C/C++ or Fortran MEX file functions from MATLAB

Java Libraries in MATLAB

Access Java libraries from MATLAB

Calling Python from MATLAB

Directly call Python functionality from MATLAB

.NET Libraries in MATLAB

Access .NET libraries from MATLAB

COM Objects in MATLAB

Access COM components from MATLAB

Calling Web Services

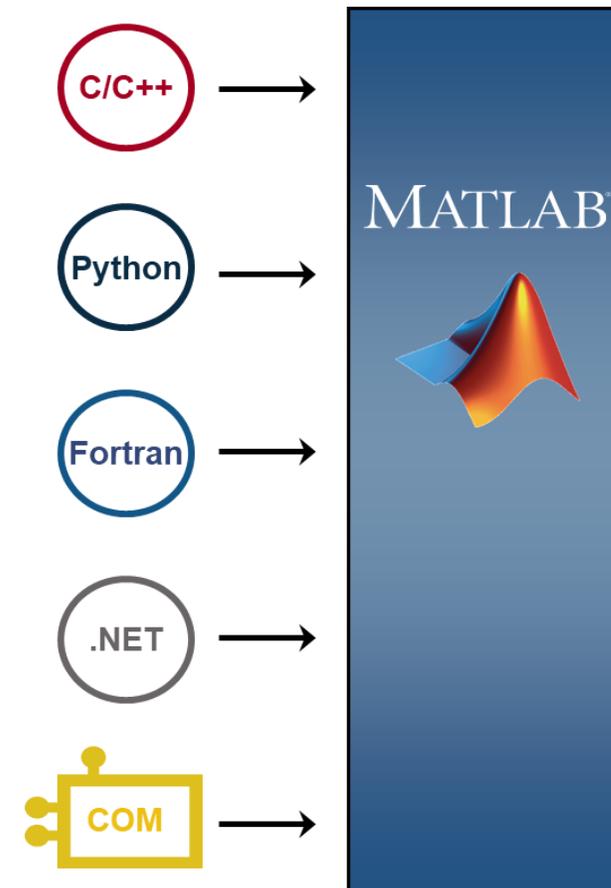
HTTP Interface

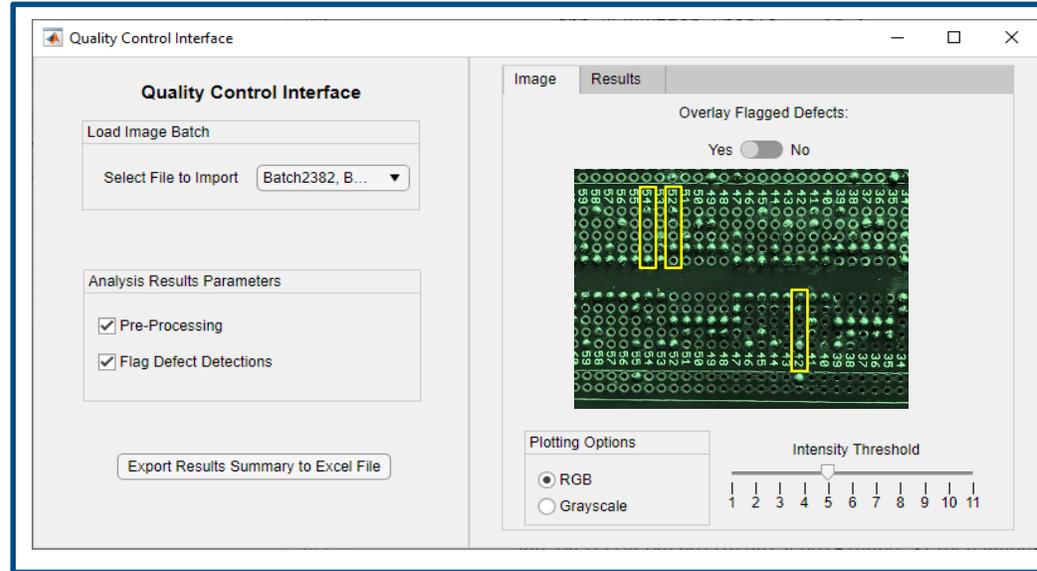
Communicate with Web service from MATLAB using HTTP (Hypertext Transfer Protocol)

WSDL (Web Services Description Language)

Communicate with Web service from MATLAB using WSDL (Web Services Description Language)

Calling MATLAB from Other Languages



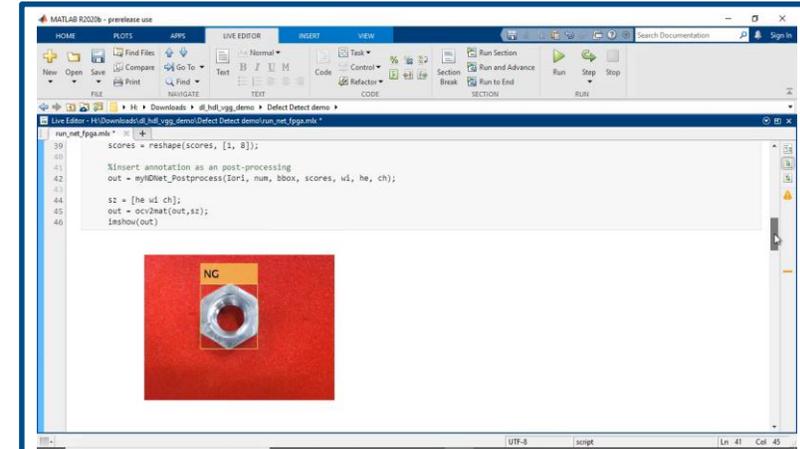
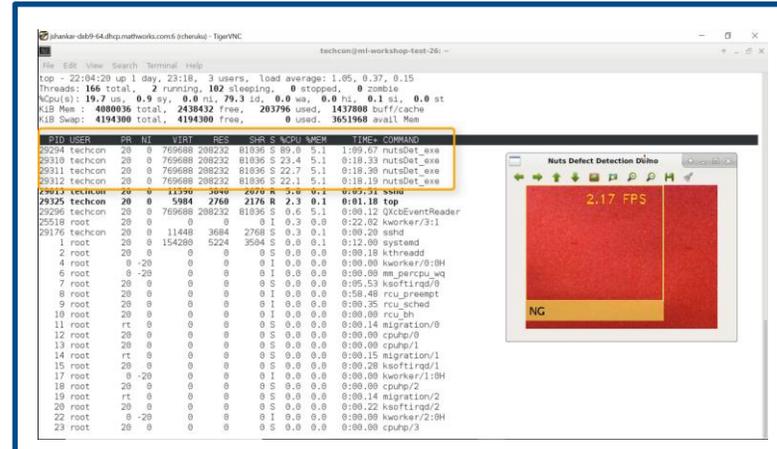


Deploy as
Desktop or
Web App

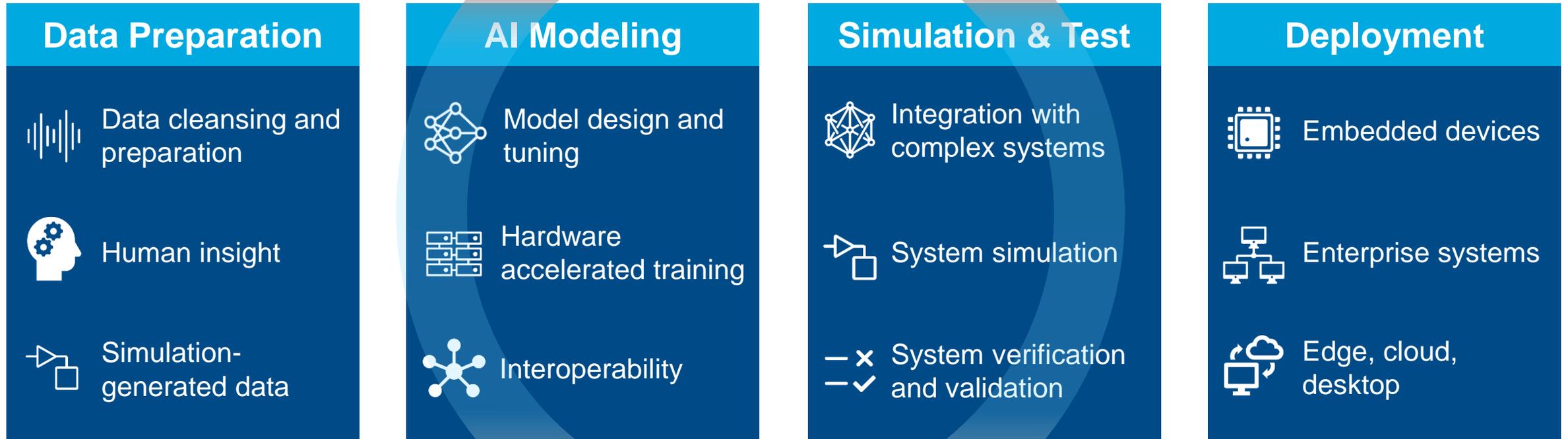
Deploy to
Jetson AGX
Xavier

Deploy to ARM
Cortex-A
microprocessor

Deploy to
ZCU102 board
from Xilinx



Defect Detection Workflow



Key Takeaways

- Interactive and easy to use apps help explore, iterate and automate workflows
- Flexibility and options to choose networks and optimizations based on data and requirements
- MATLAB provides an easy and extensible framework for visual inspection from data access to deployment

Image Processing Toolbox

Perform image processing, visualization, and analysis

[Watch video](#)

[Download a free trial](#)



Computer Vision Toolbox

Design and test computer vision, 3D vision, and video processing systems

[Watch video](#)

[Download a free trial](#)



Deep Learning Toolbox

Design, train, and analyze deep learning networks

[Watch video](#)

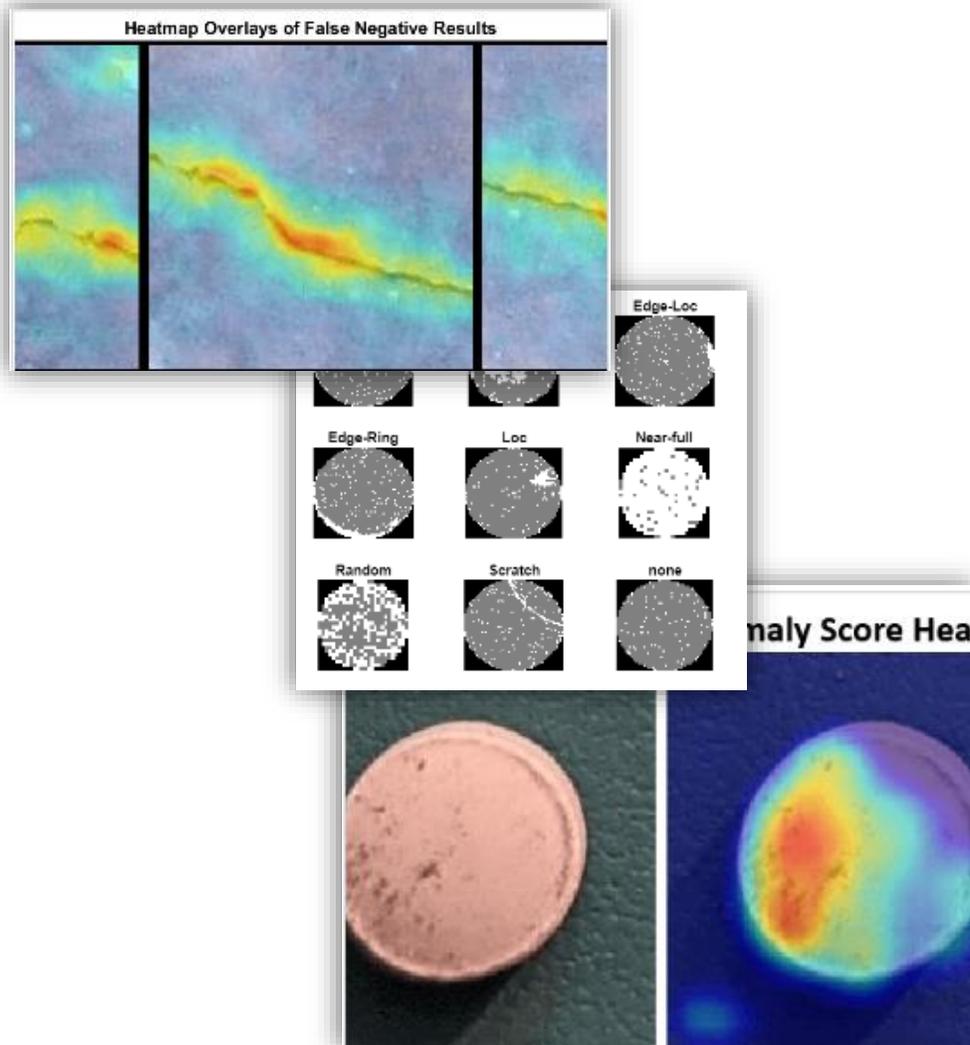
[Download a free trial](#)

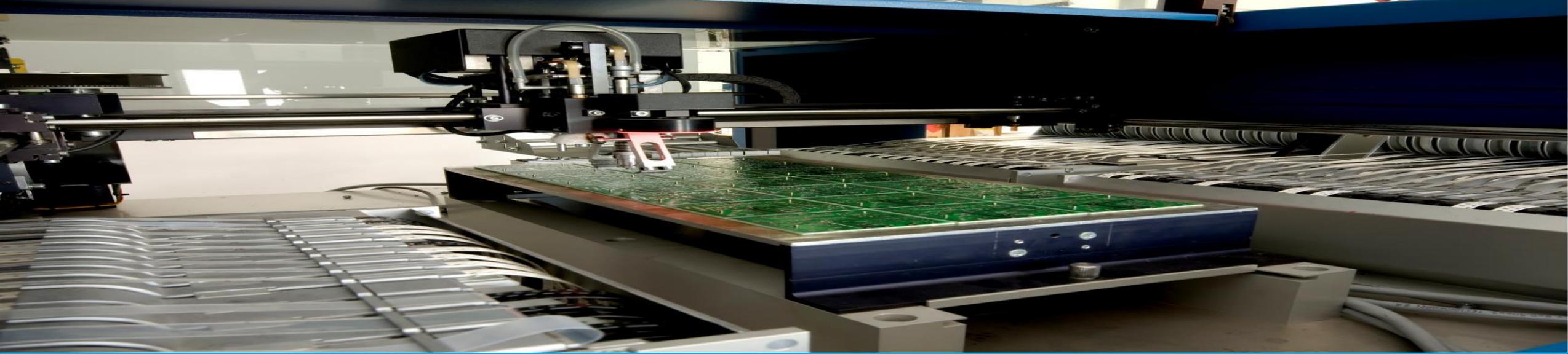


Visual Inspection Support package

[Computer Vision Toolbox Automated Visual Inspection Library](#)

- Anomaly detector
- Parameter optimization
- Visualization and evaluation tools
- Dedicated examples
 - [Detect Image Anomalies Using Pretrained ResNet-18 Feature Embeddings](#)
 - [Classify Defects on Wafer Maps Using Deep Learning](#)
 - [Detect Image Anomalies Using Explainable One-Class Classification Neural Network](#)





Questions?