MATLAB EXPO

산업용 어플리케이션을 위한 딥러닝 기반 머신비전 솔루션

송완빈, MathWorks

MathWorks®

What is Automated Optical 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**"*

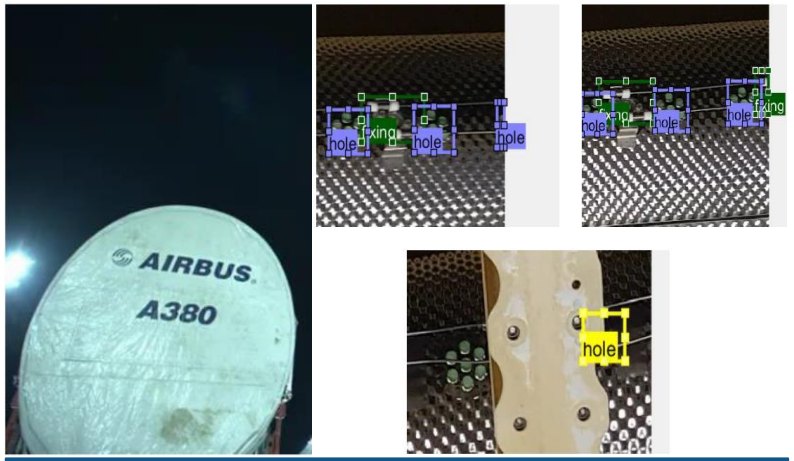
Automated Defect Detection

Machine Vision Visual Inspection

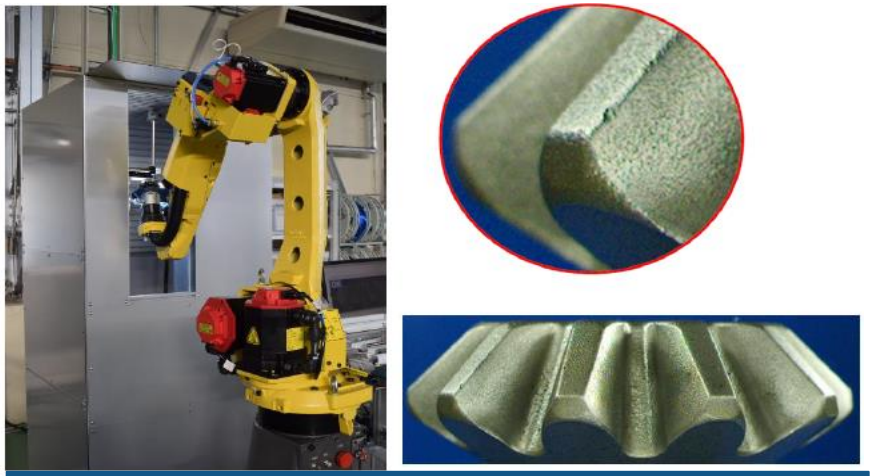
Automated Inspection

Customer References

AIRBUS



Automatic Defect Detection

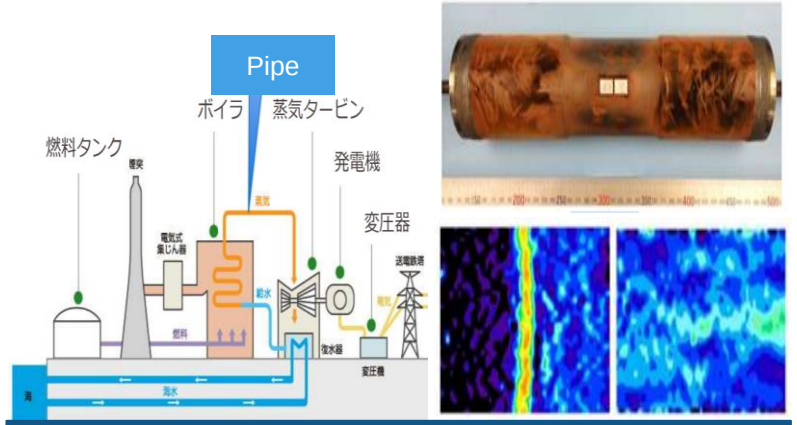


Visual Inspection of Automotive Parts

MUSASHI

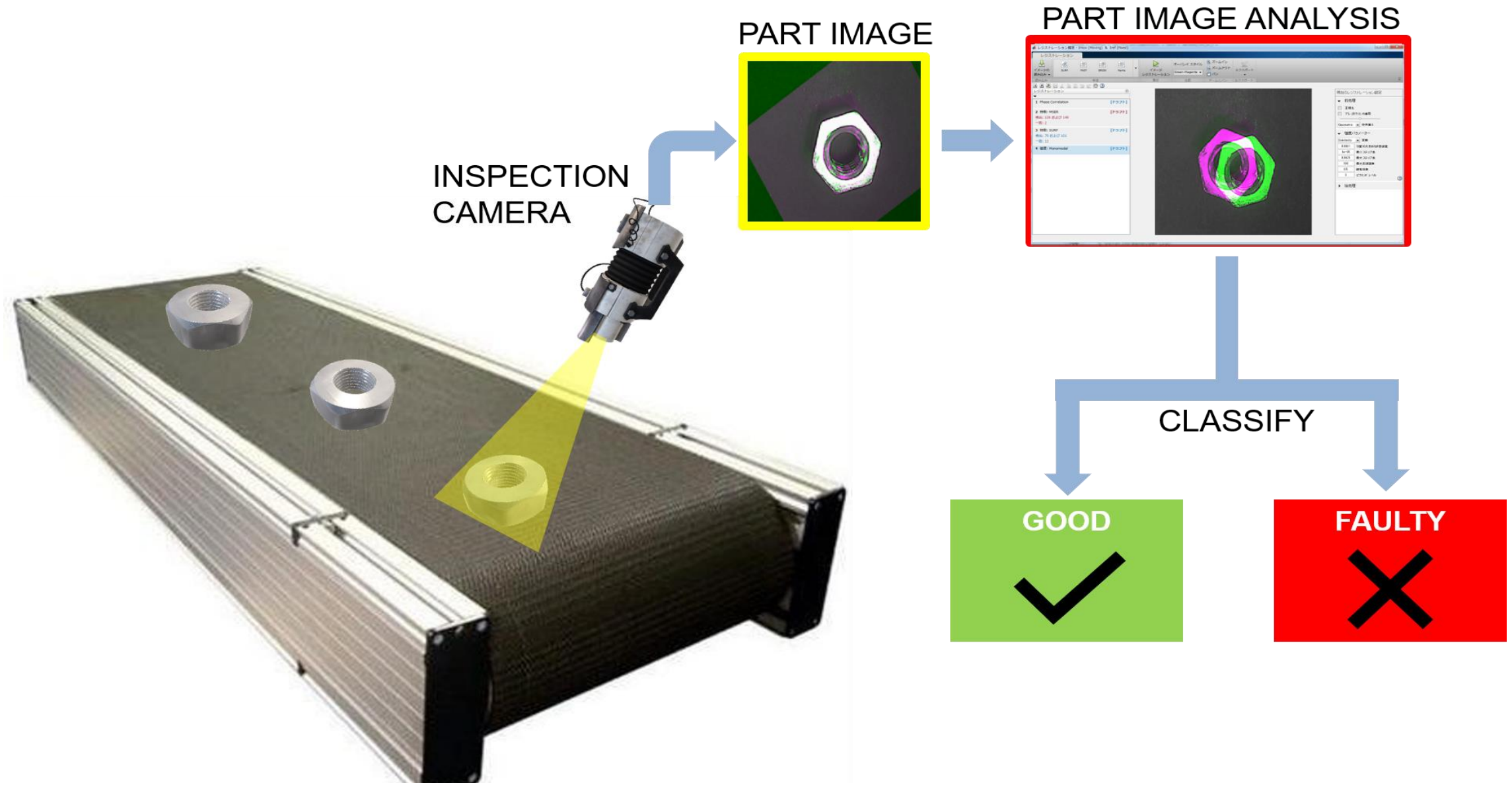
KRRI

Defect Detection in Railway Components

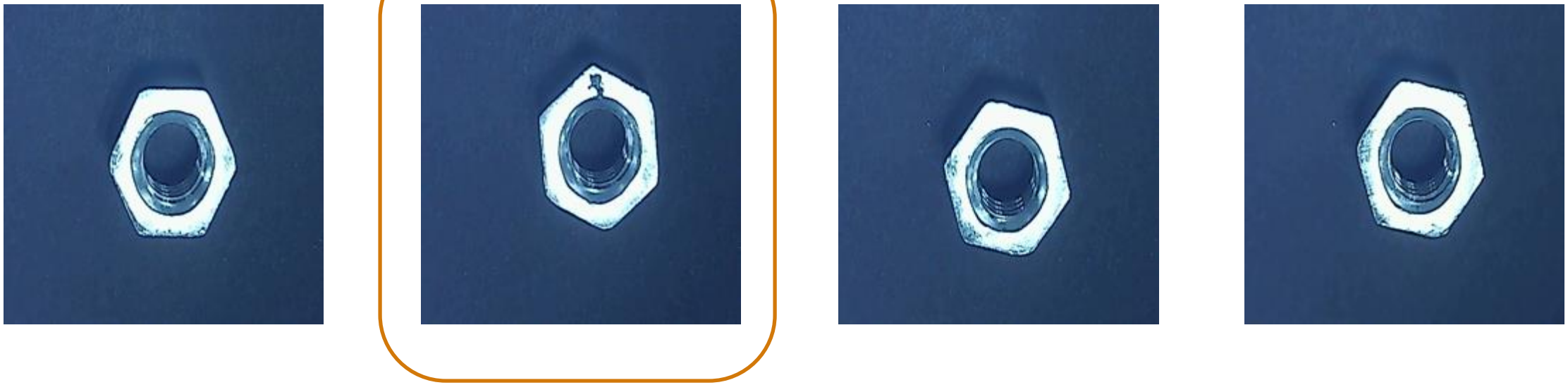


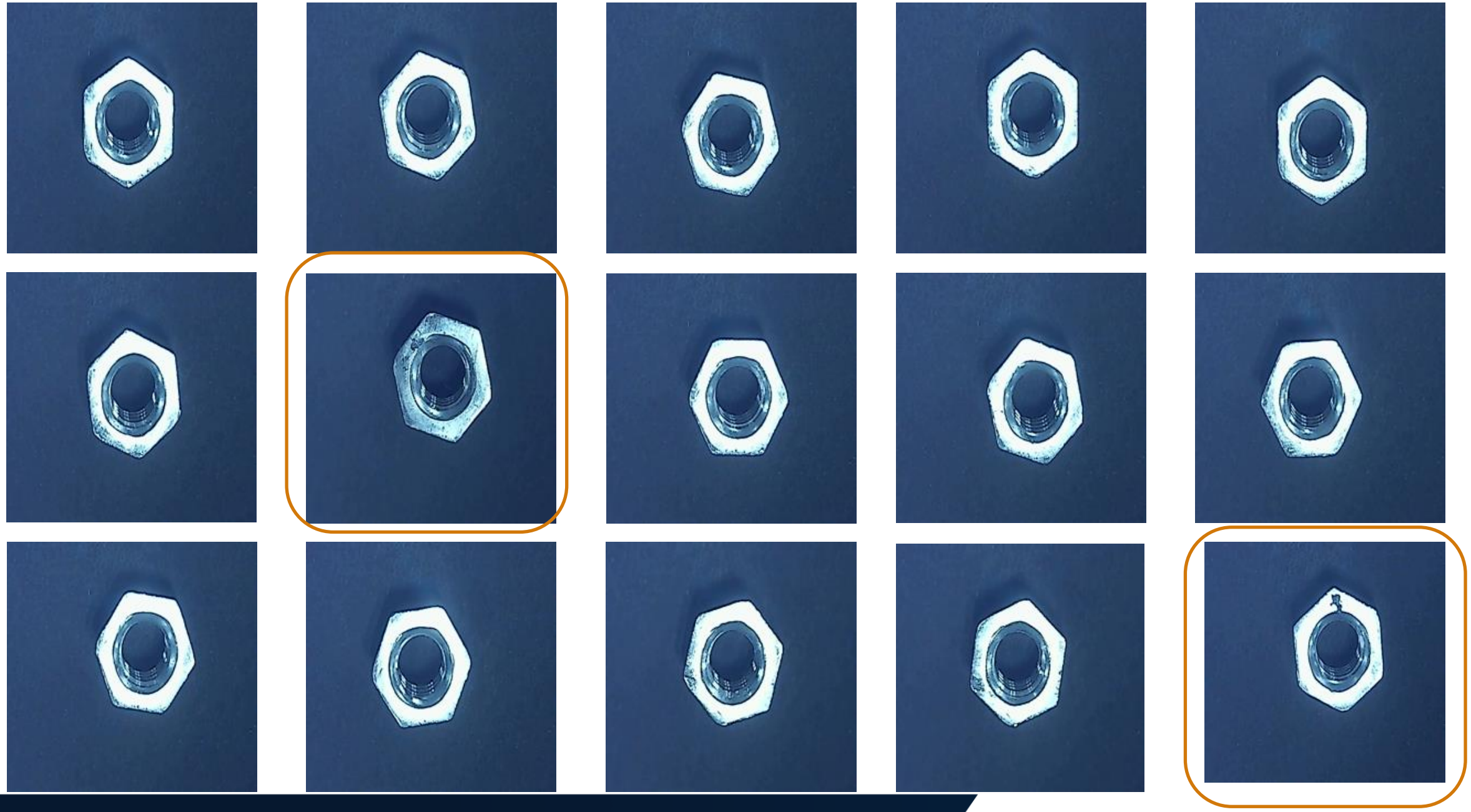
Assess Pipe Weld Damage at Power Plants

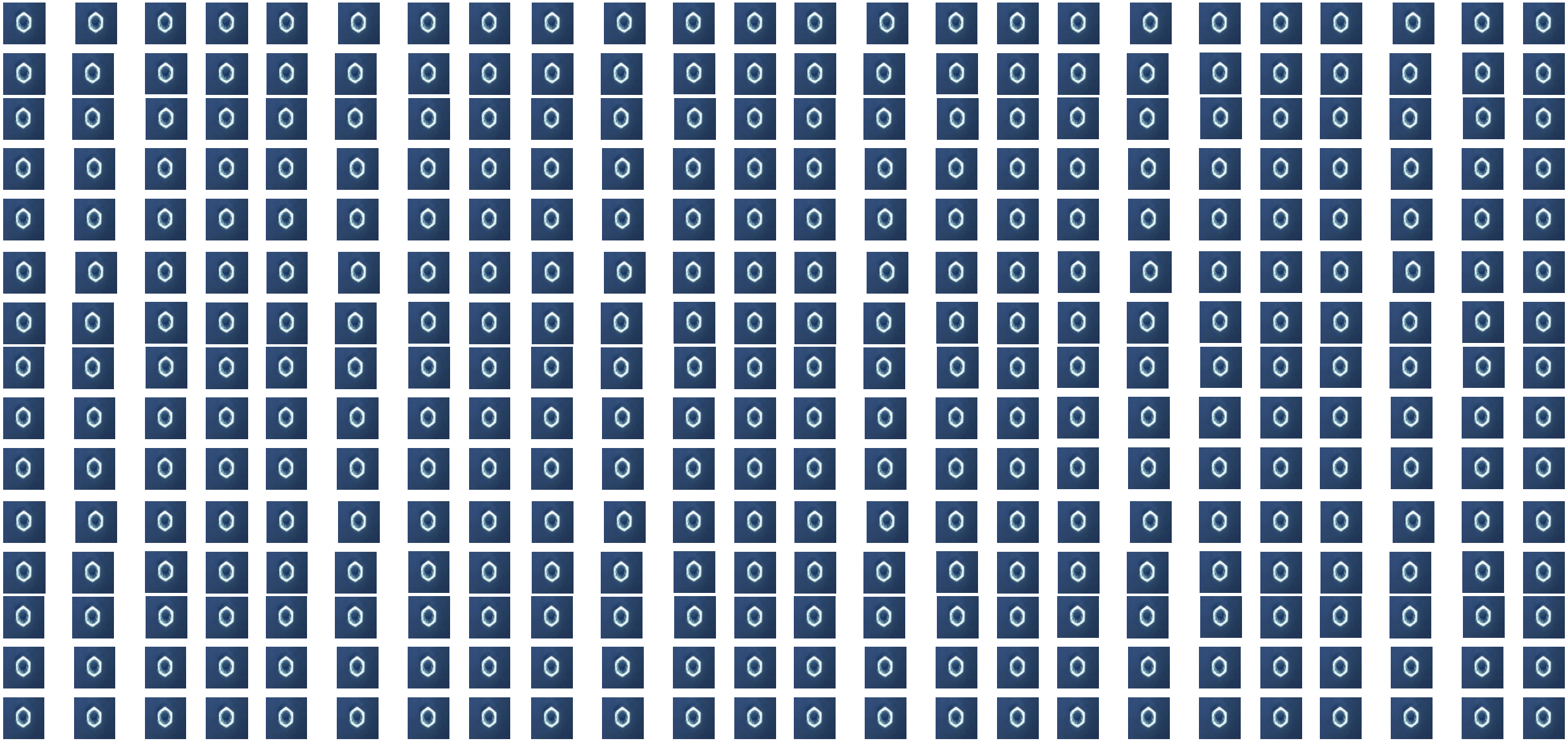
Kansai Electric Power
power with heart

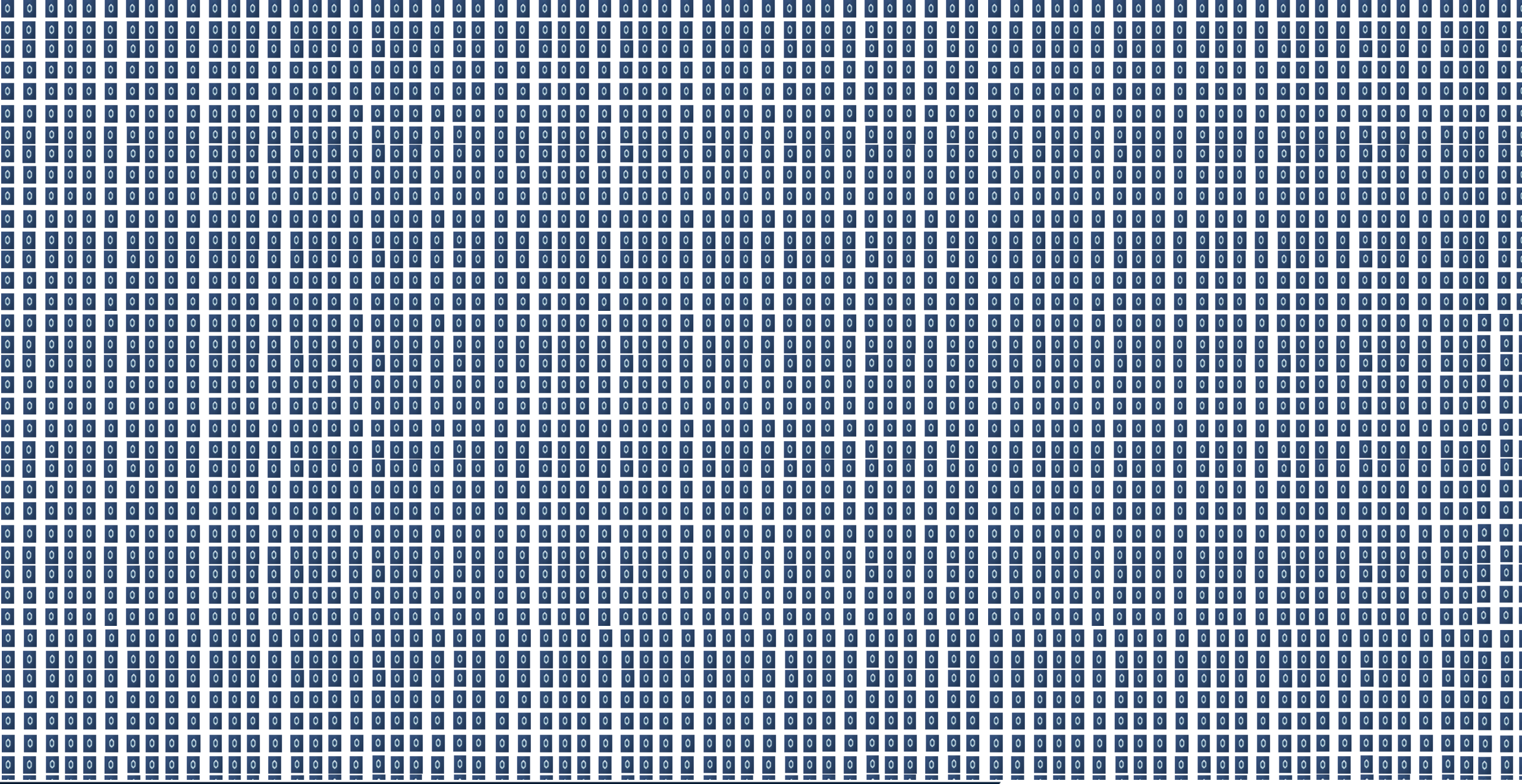

PART IMAGE
PART IMAGE ANALYSIS
INSPECTION
CAMERA
CLASSIFY
GOOD
FAULTY

Can you find the defective hex nut?



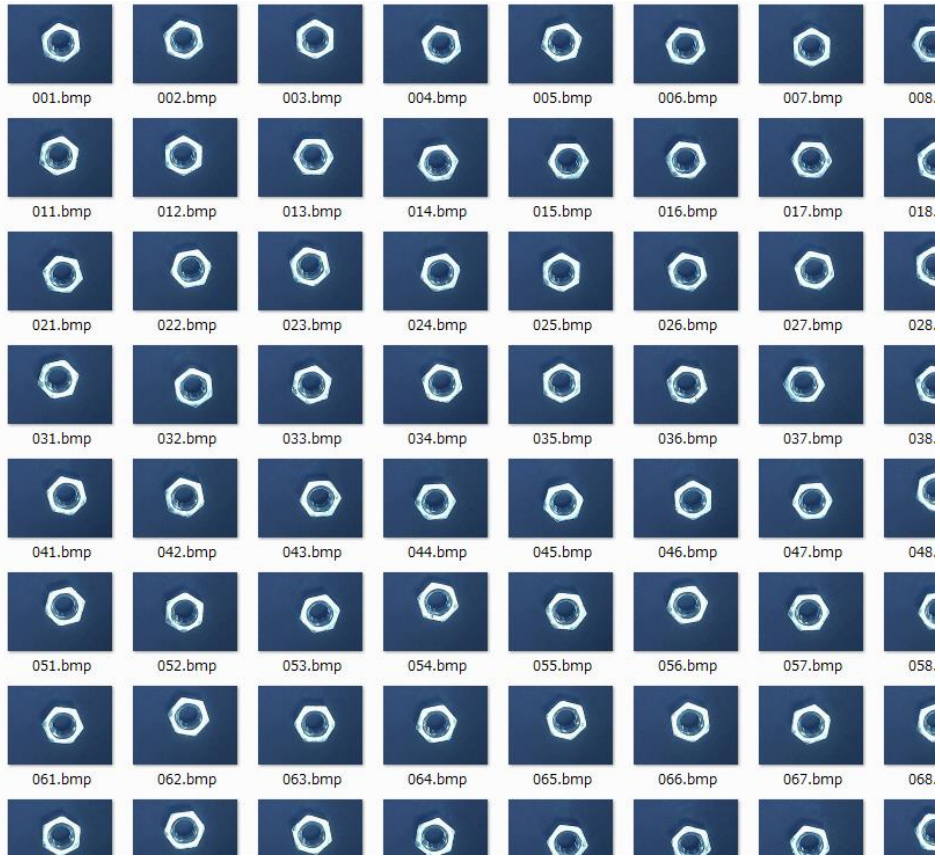




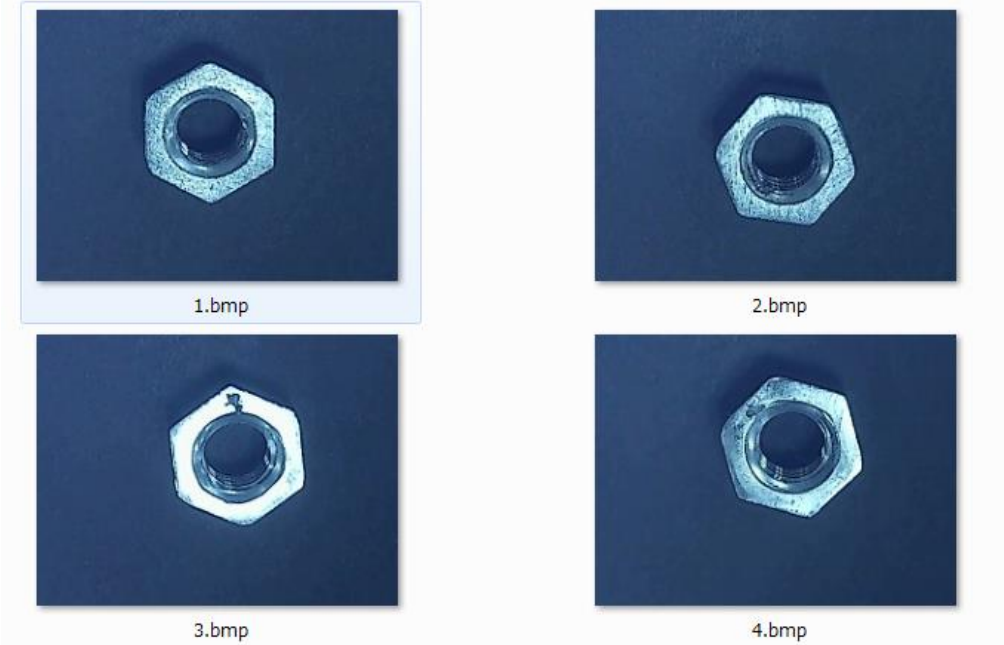


Finding Defective Hex Nuts

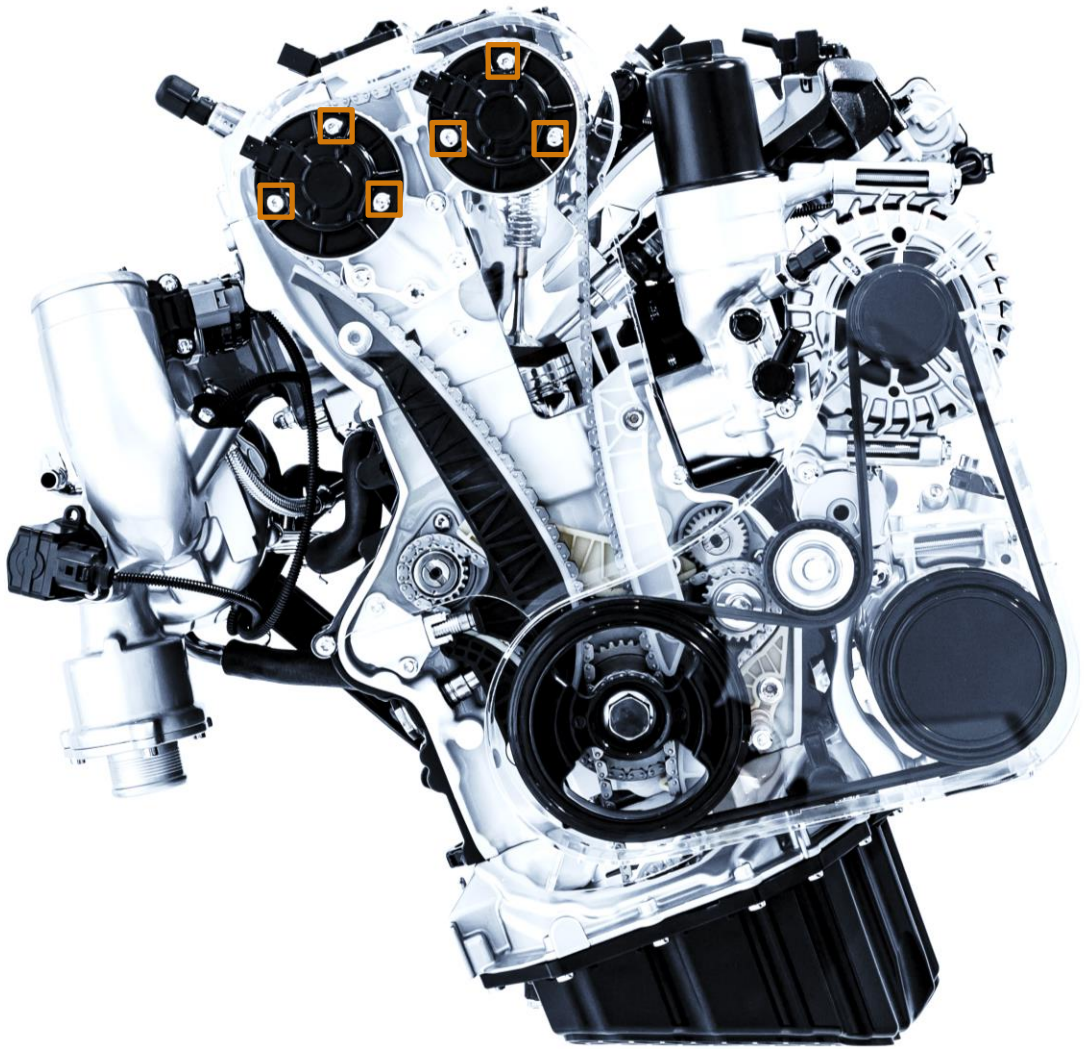
Good



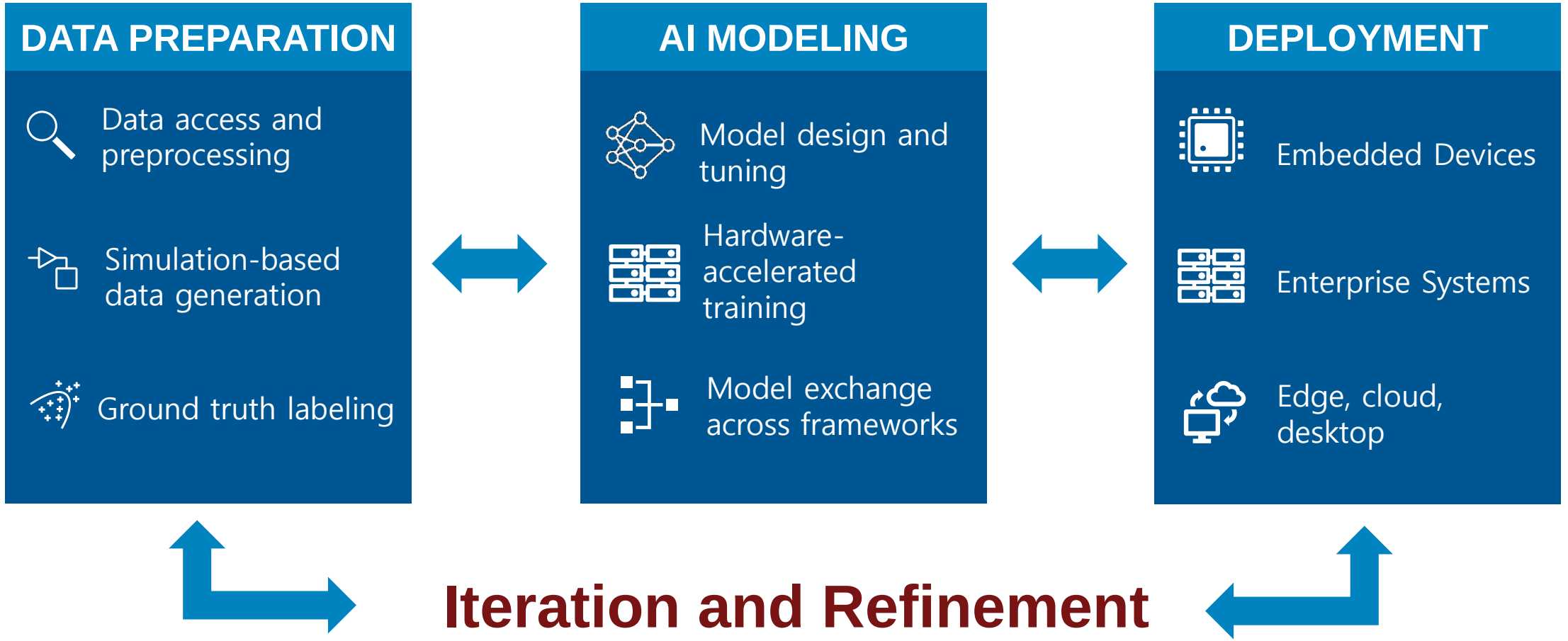
Defective



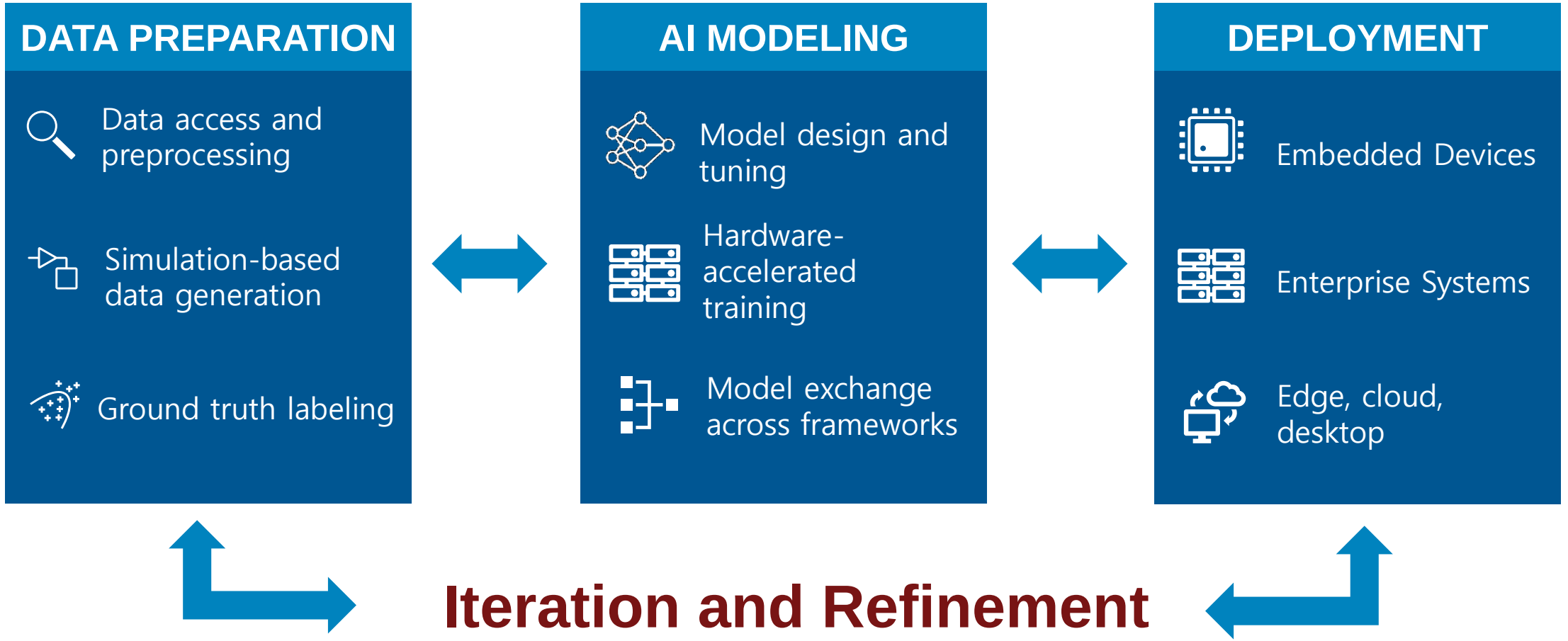
Detecting Parts



Defect Detection Workflow



Defect Detection Workflow



Data Access and Preprocessing – Common Challenges

- How do I access large data that might not fit in memory?
- 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

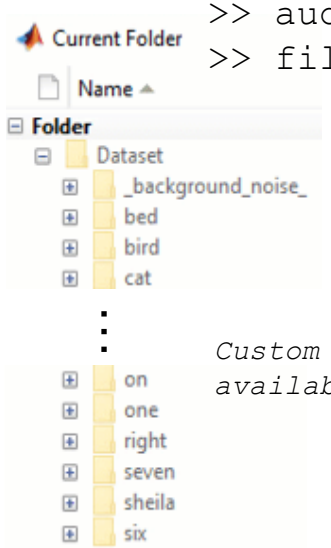
How do I access large data that might not fit in memory?

How do I load and access large amounts of data?

Datastores

Loads image/signal data into memory as and when needed

```
>> imageDatastore
>> audioDatastore
>> fileDatastore
```

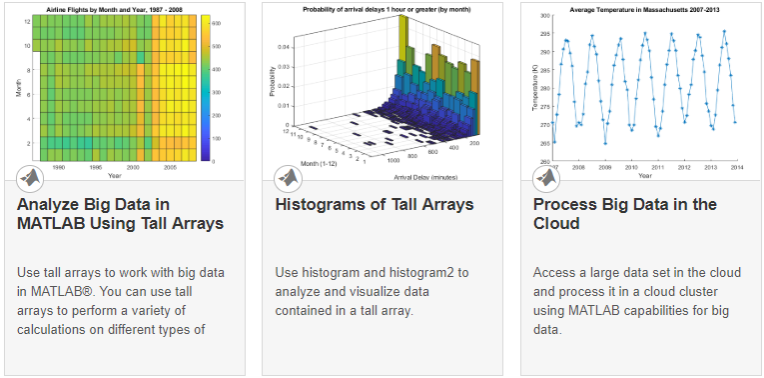


Custom Datastores also available

Tall Arrays

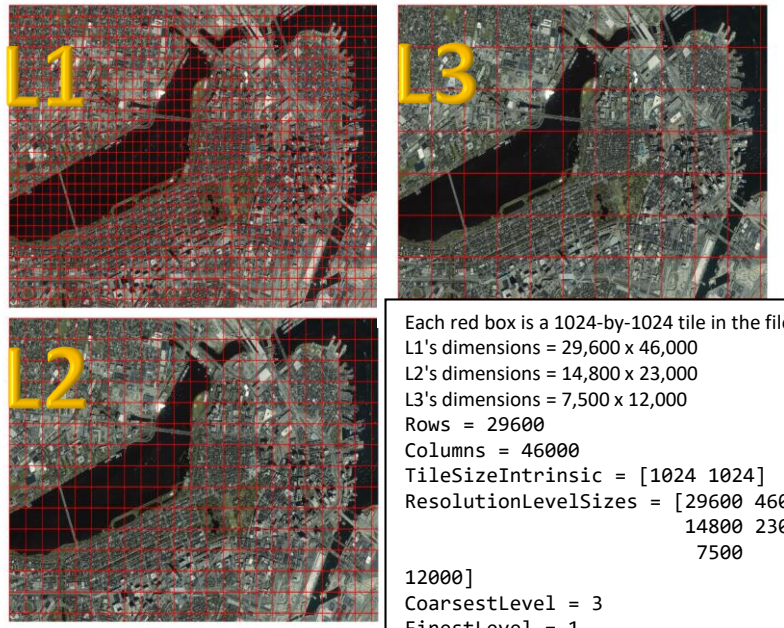
Work with out-of-memory numeric data

- Train deep neural networks for numeric arrays



BigImage

Work with very large, tiled and multi-resolution images



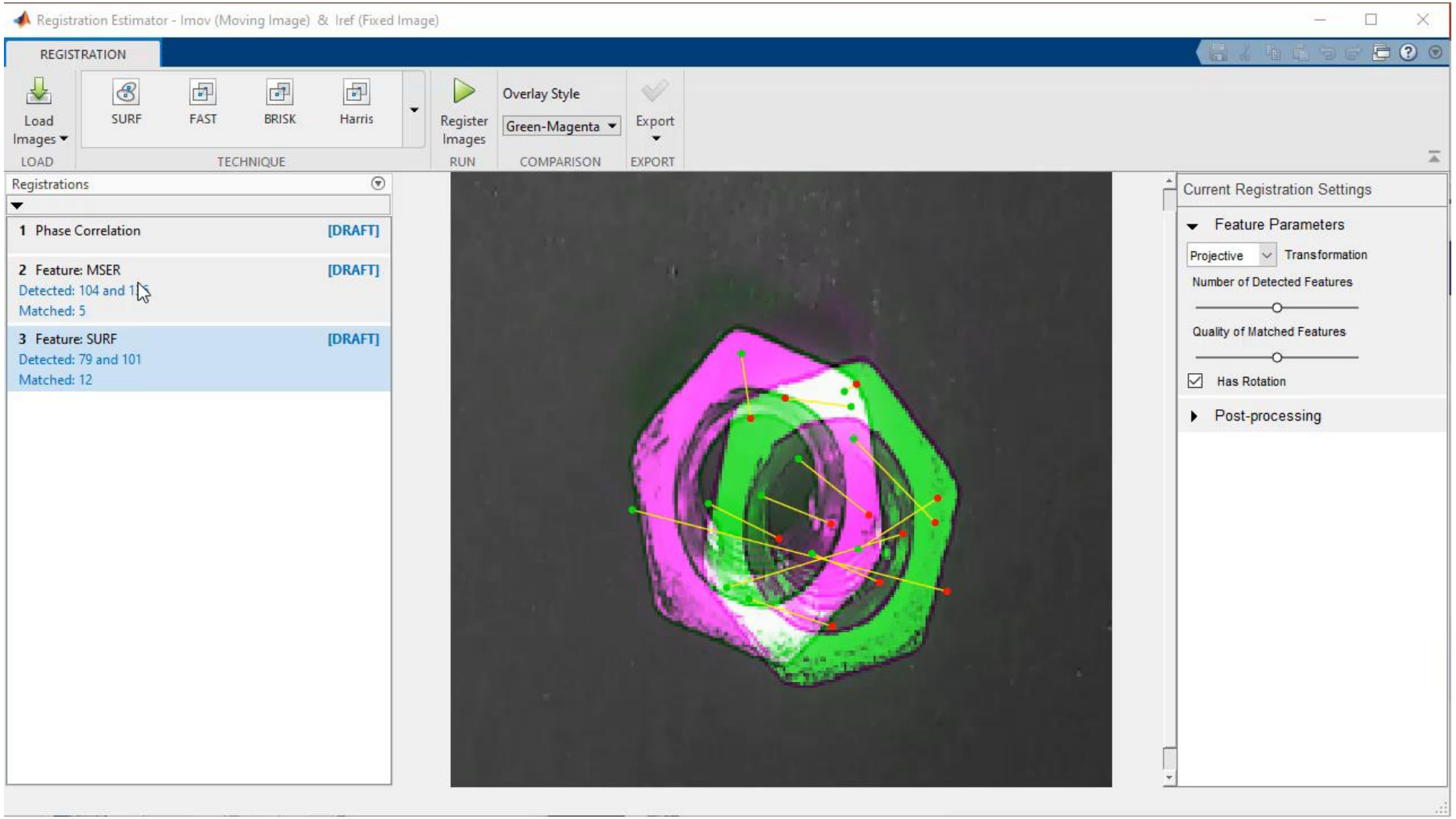
Each red box is a 1024-by-1024 tile in the file.
L1's dimensions = 29,600 x 46,000
L2's dimensions = 14,800 x 23,000
L3's dimensions = 7,500 x 12,000

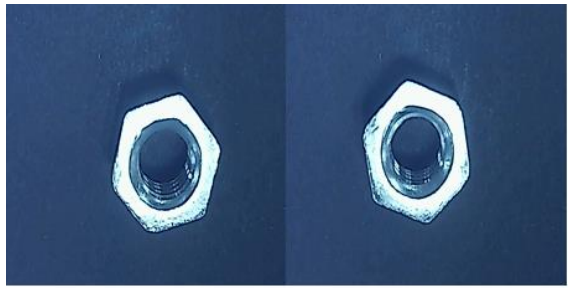
```
Rows = 29600
Columns = 46000
TileSizeIntrinsic = [1024 1024]
ResolutionLevelSizes = [29600 46000
                        14800 23000
                         7500
12000]
CoarsestLevel = 3
FinestLevel = 1
PixelSpacings = [1 1; 2 2; 3.947
3.833]
```

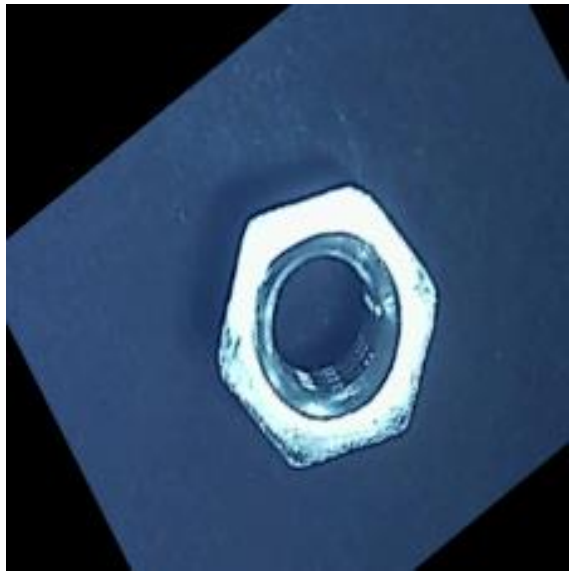
Data Access and Preprocessing – Common Challenges

How do I preprocess data and get the right features?

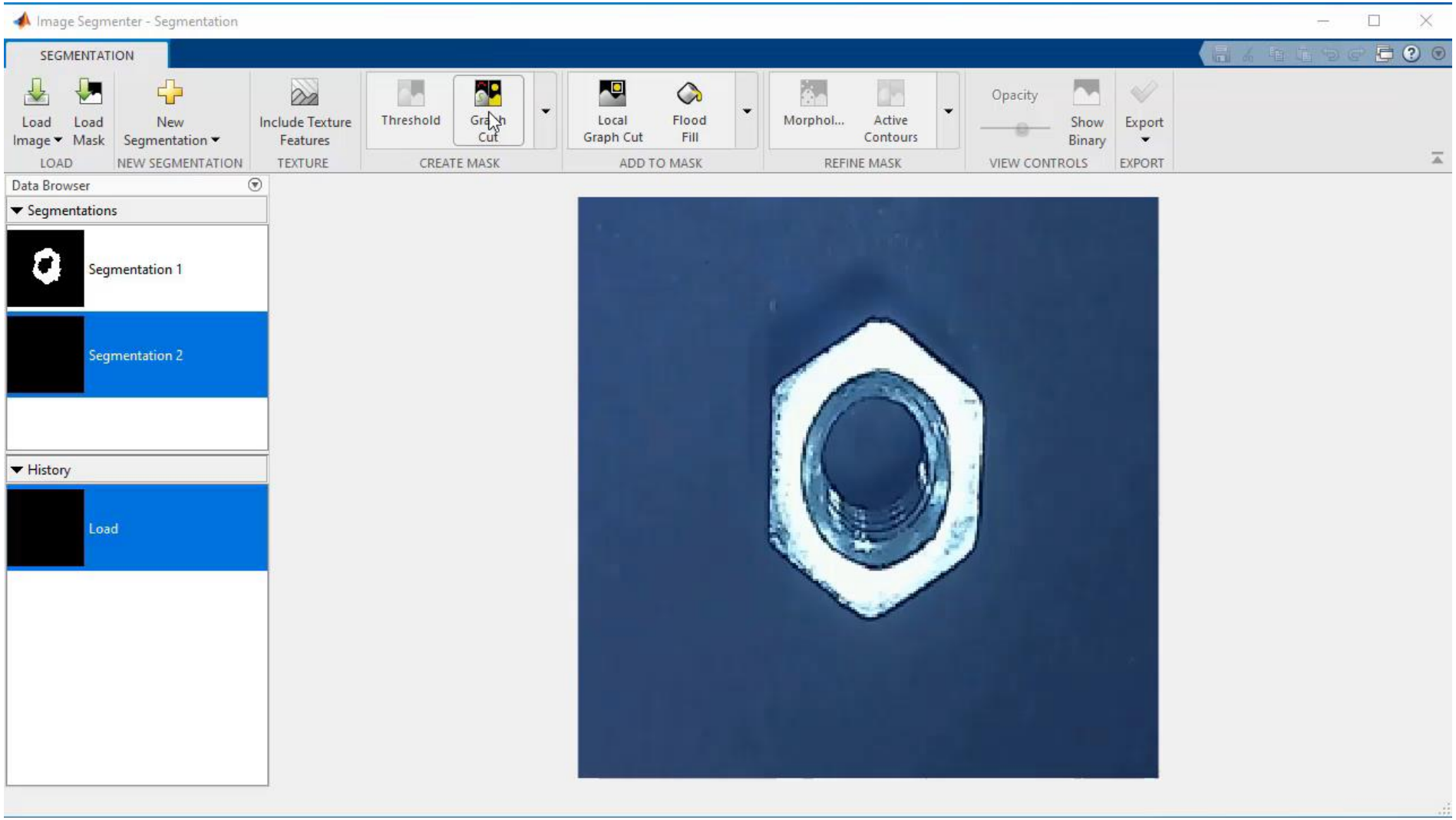
Pre-processing Data – Registration Estimator App

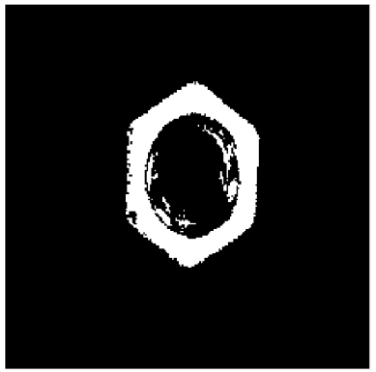




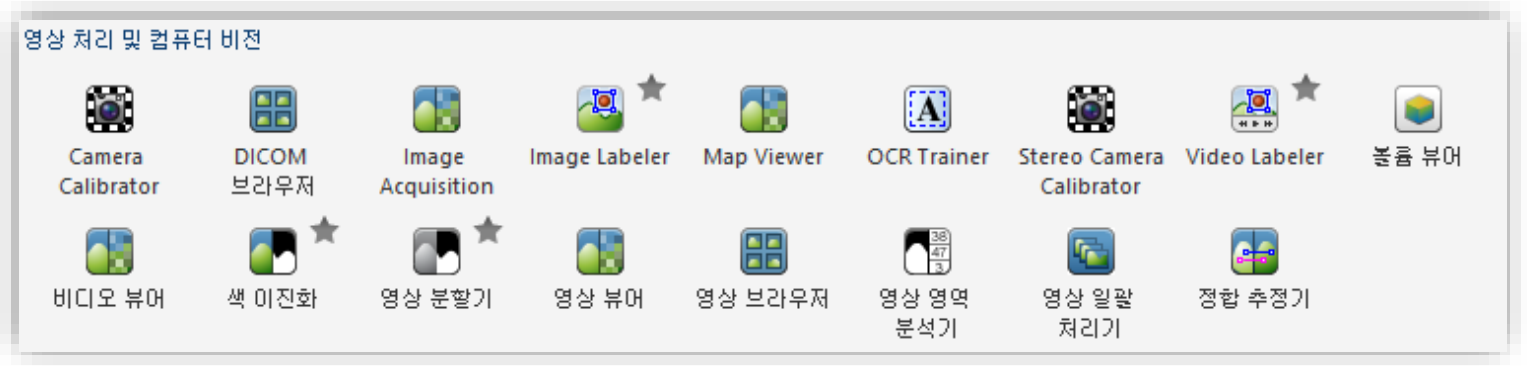


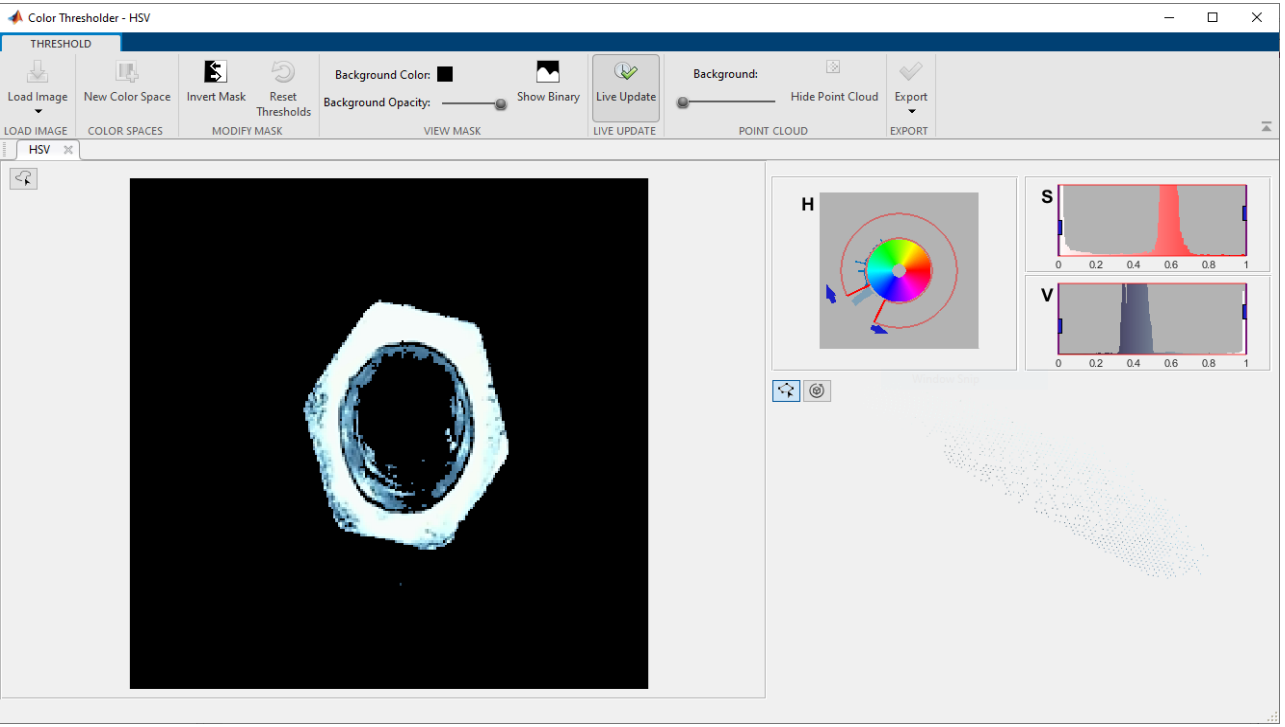
Pre-processing Data – Image Segmenter App





Preprocessing Data - Apps





Color Thresholder

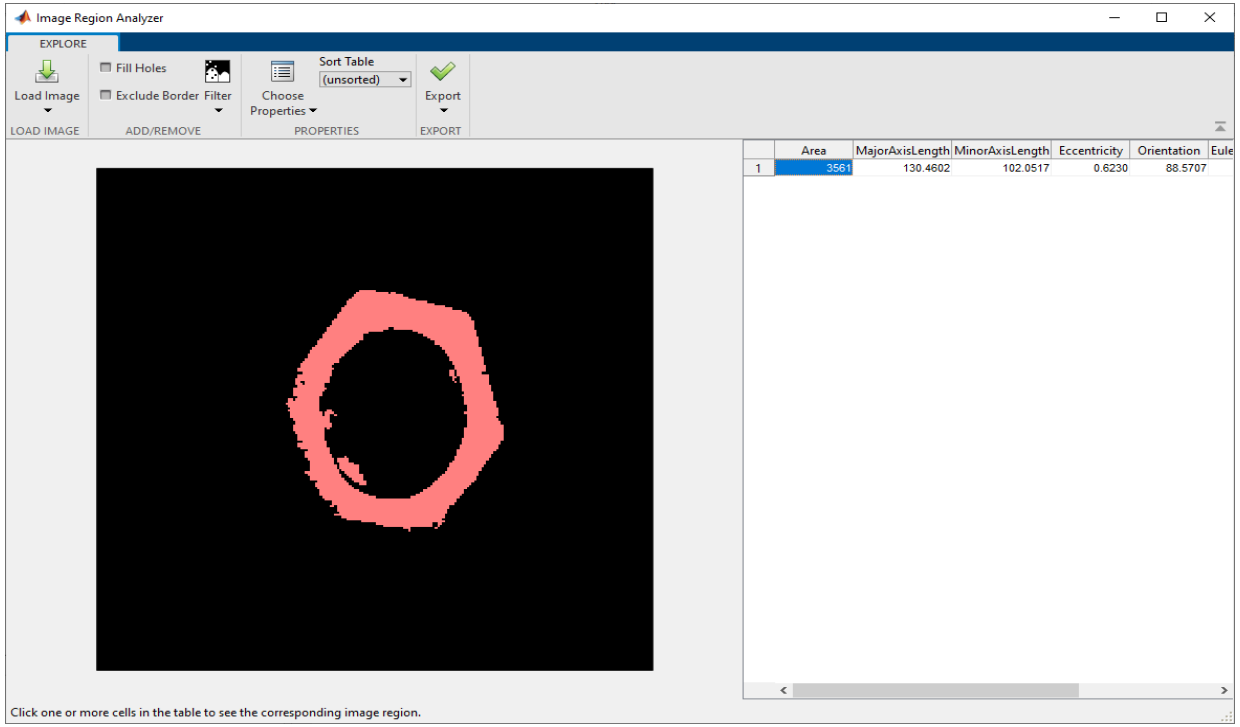
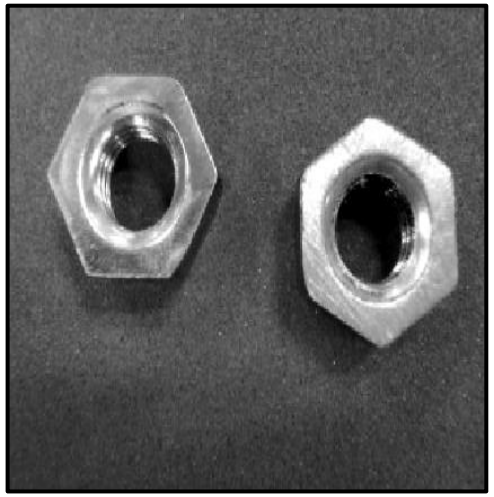


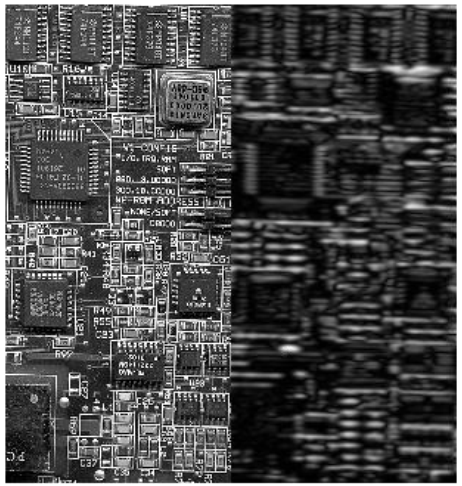
Image Region Analyzer

Pre-processing Data – Built-in Algorithms

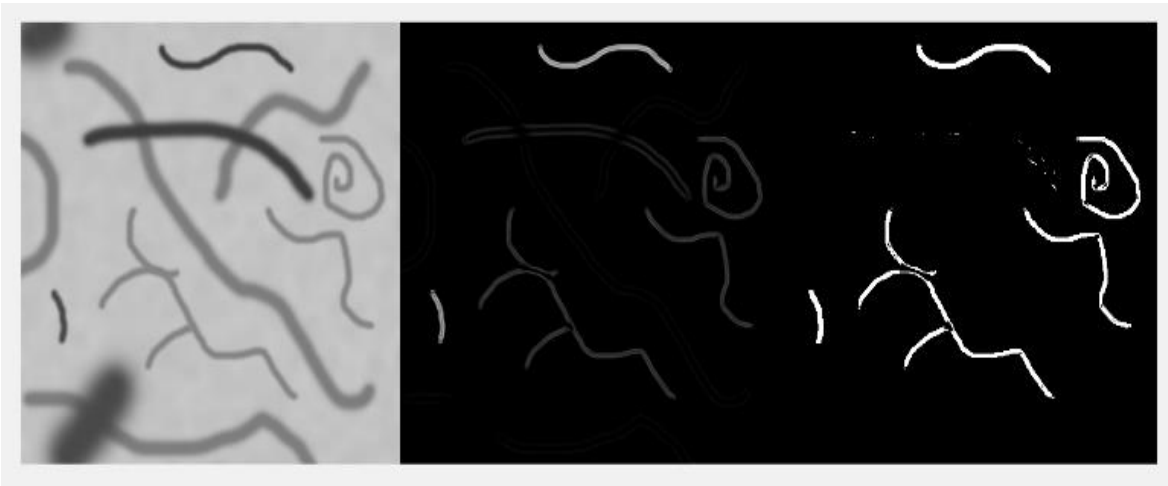
`imadjust`



`imgaborfilt`

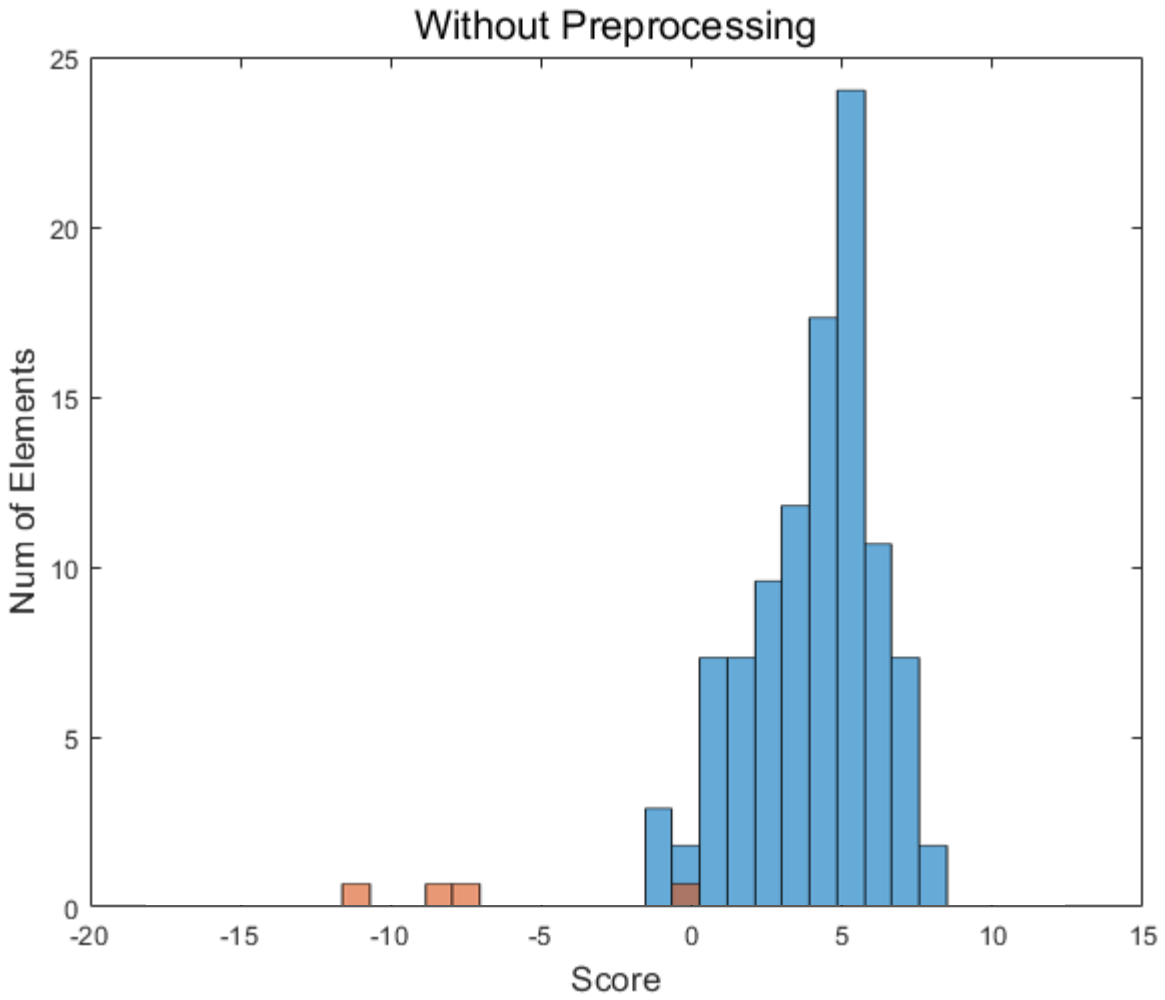


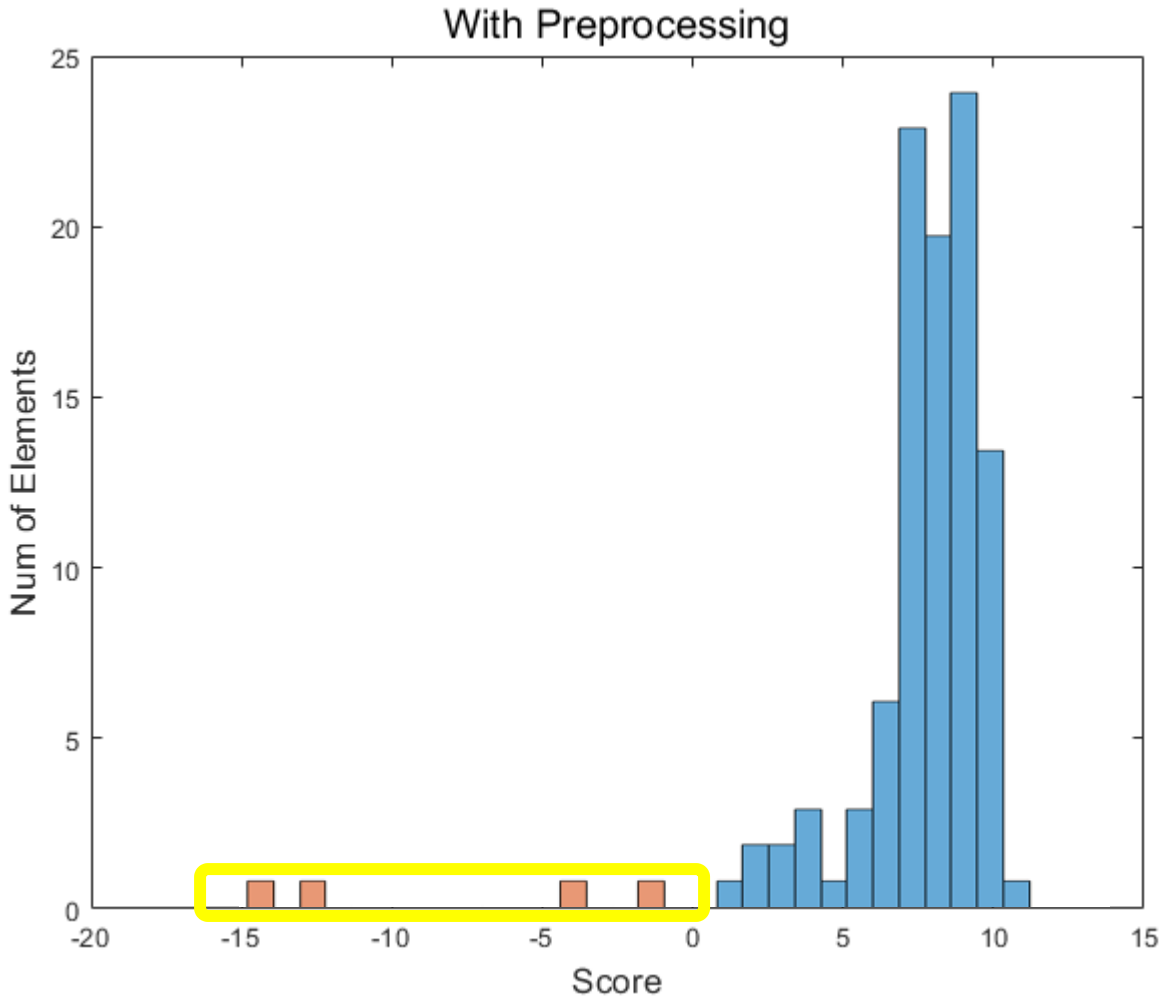
`fibermetric`



And Many More!

Defect detection using AlexNet: Results with preprocessing





Data Access and Preprocessing – Common Challenges

How do I label my data faster?

Data Preprocessing - Labeling

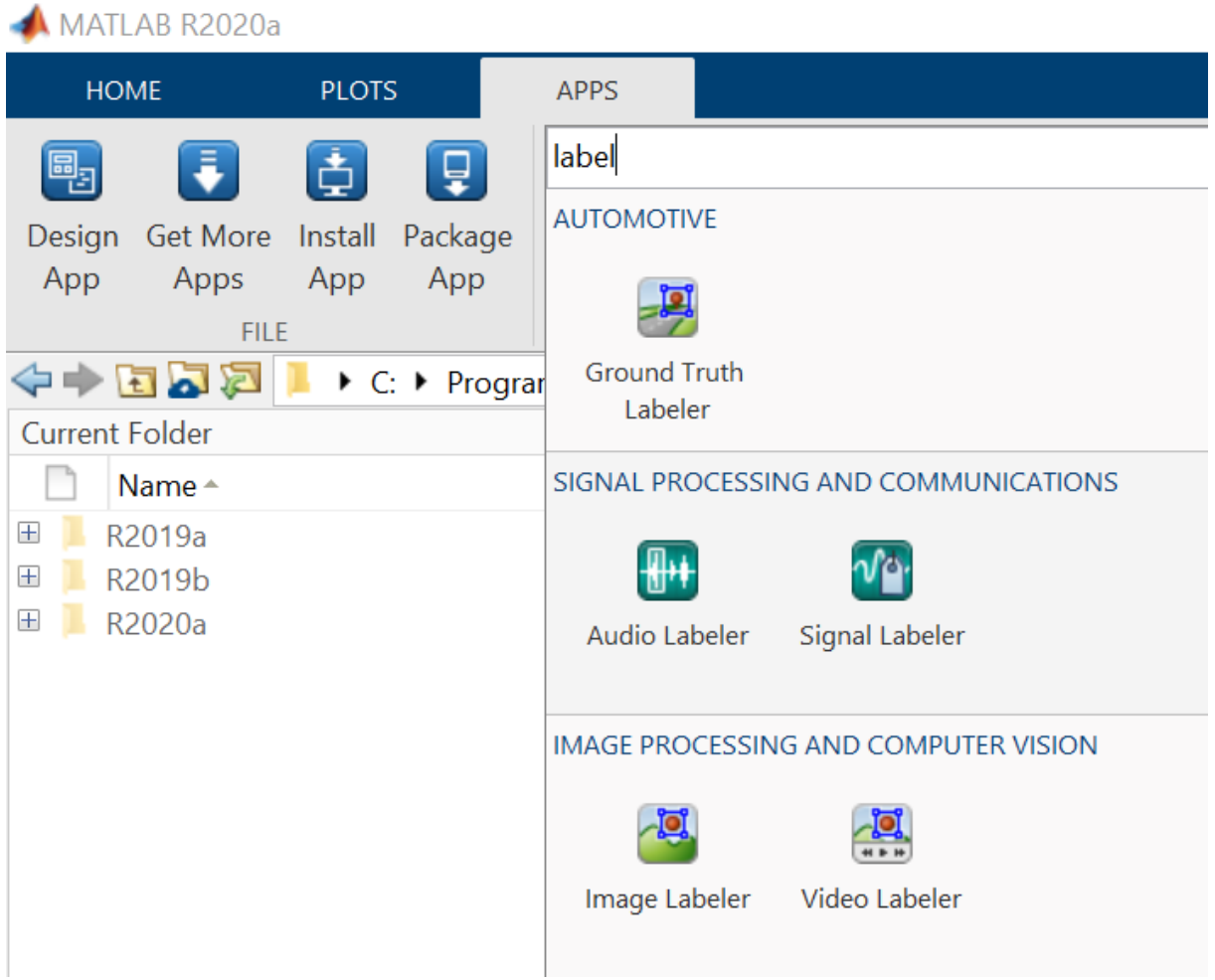
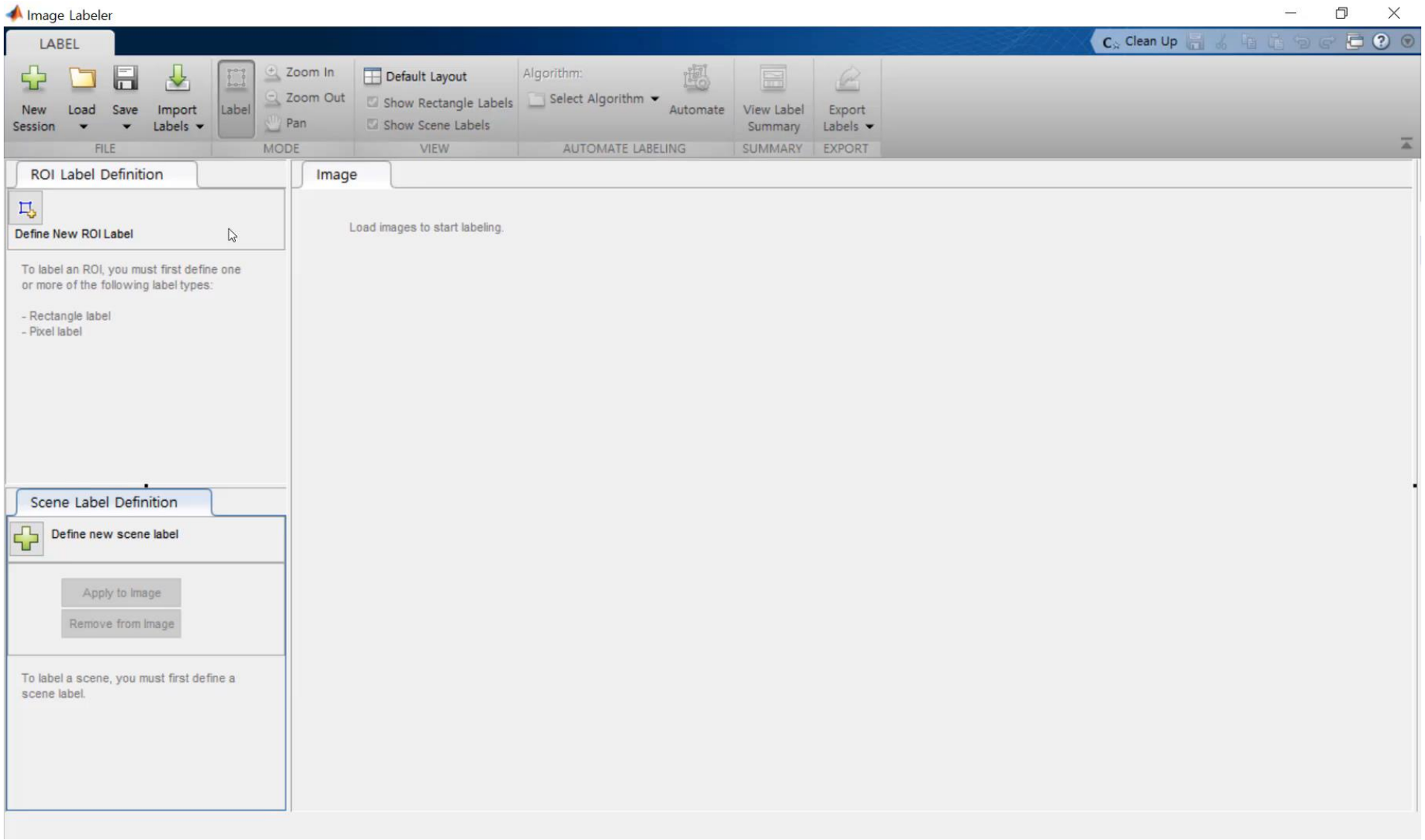


Image Labeler

Image Labeler

Video Labeler

Big-Image Labeler

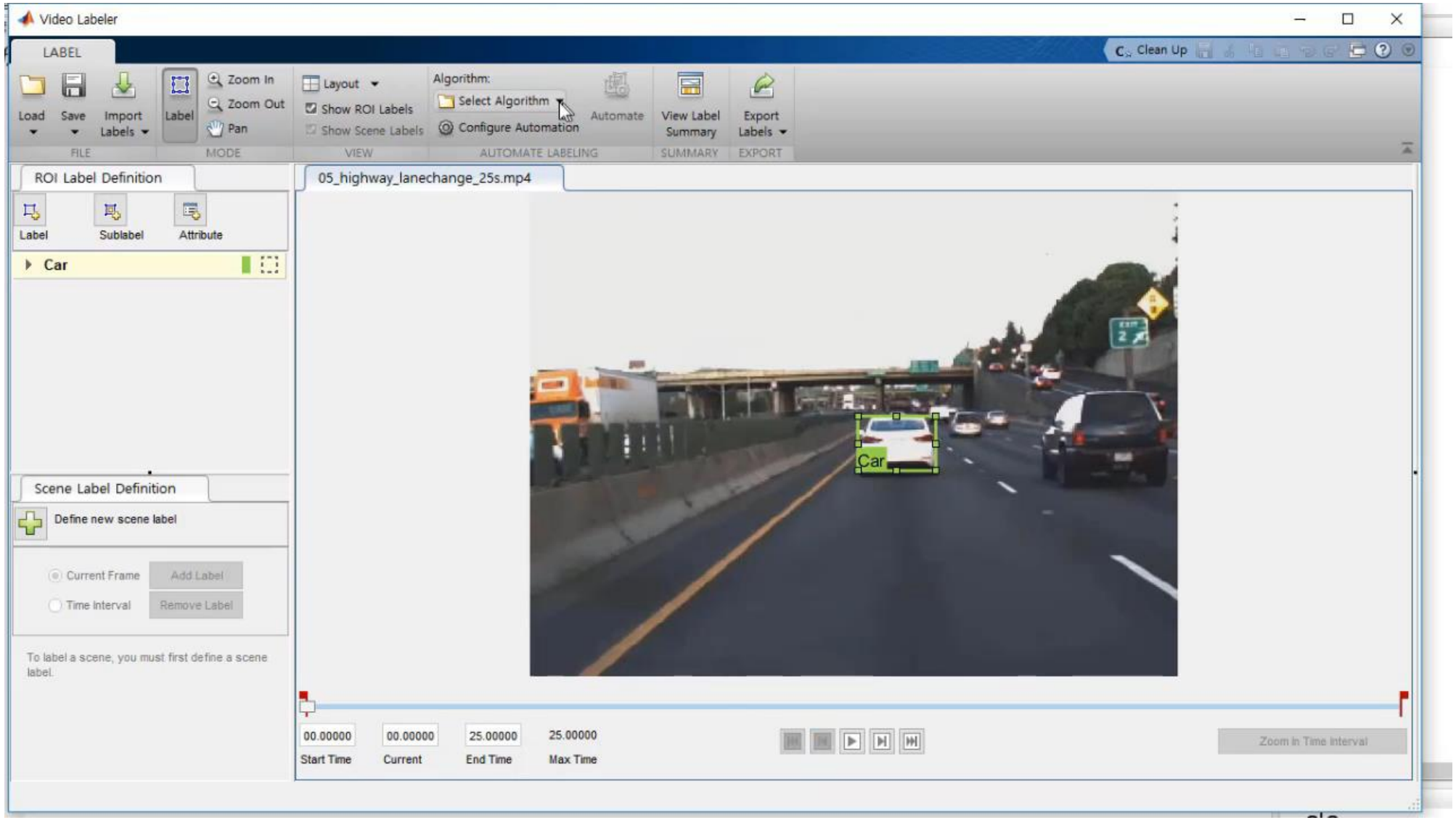


Video Labeler

Image Labeler

Video Labeler

Big-Image Labeler

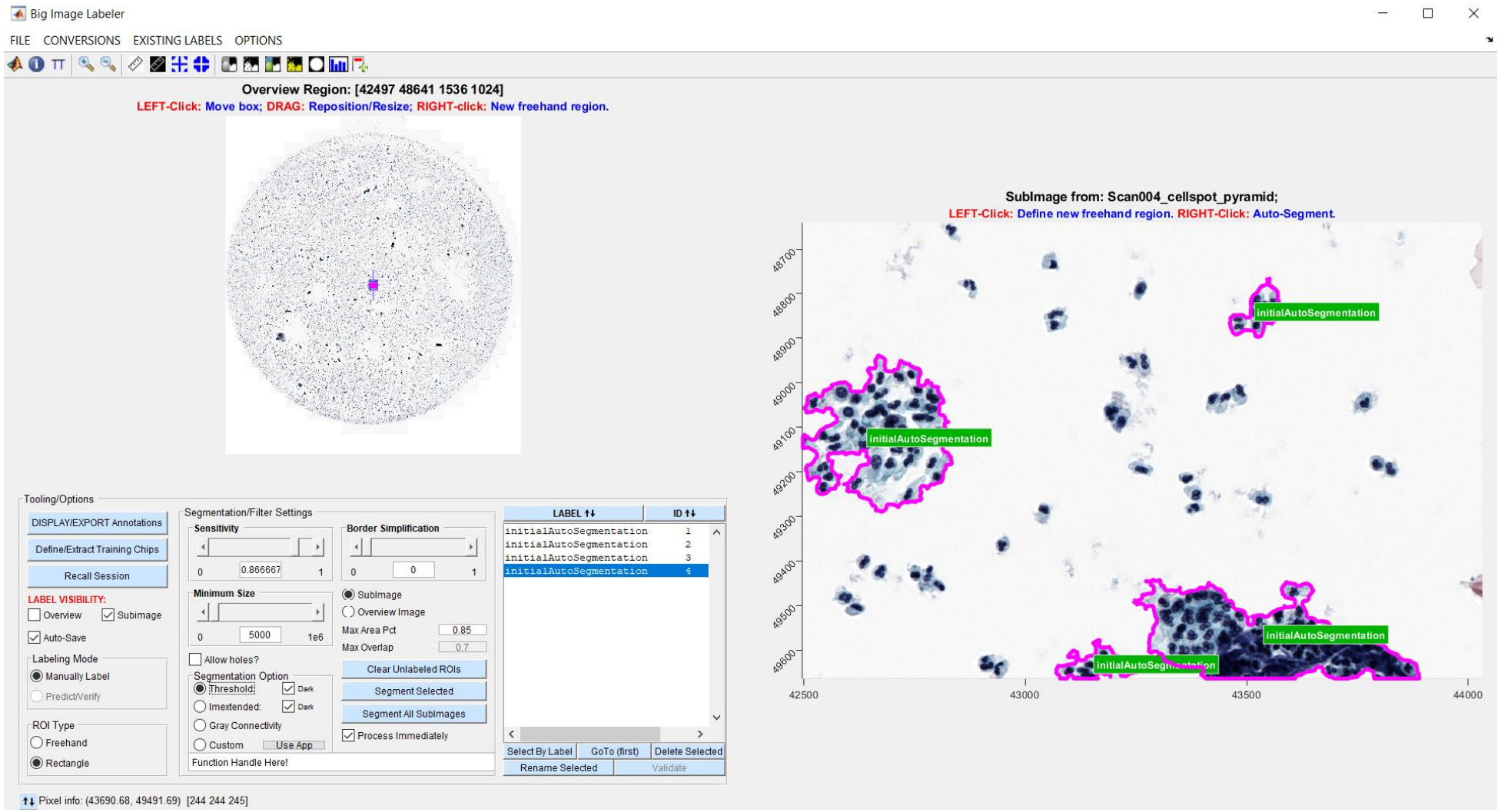


Big Image Labeler

Image Labeler

Video Labeler

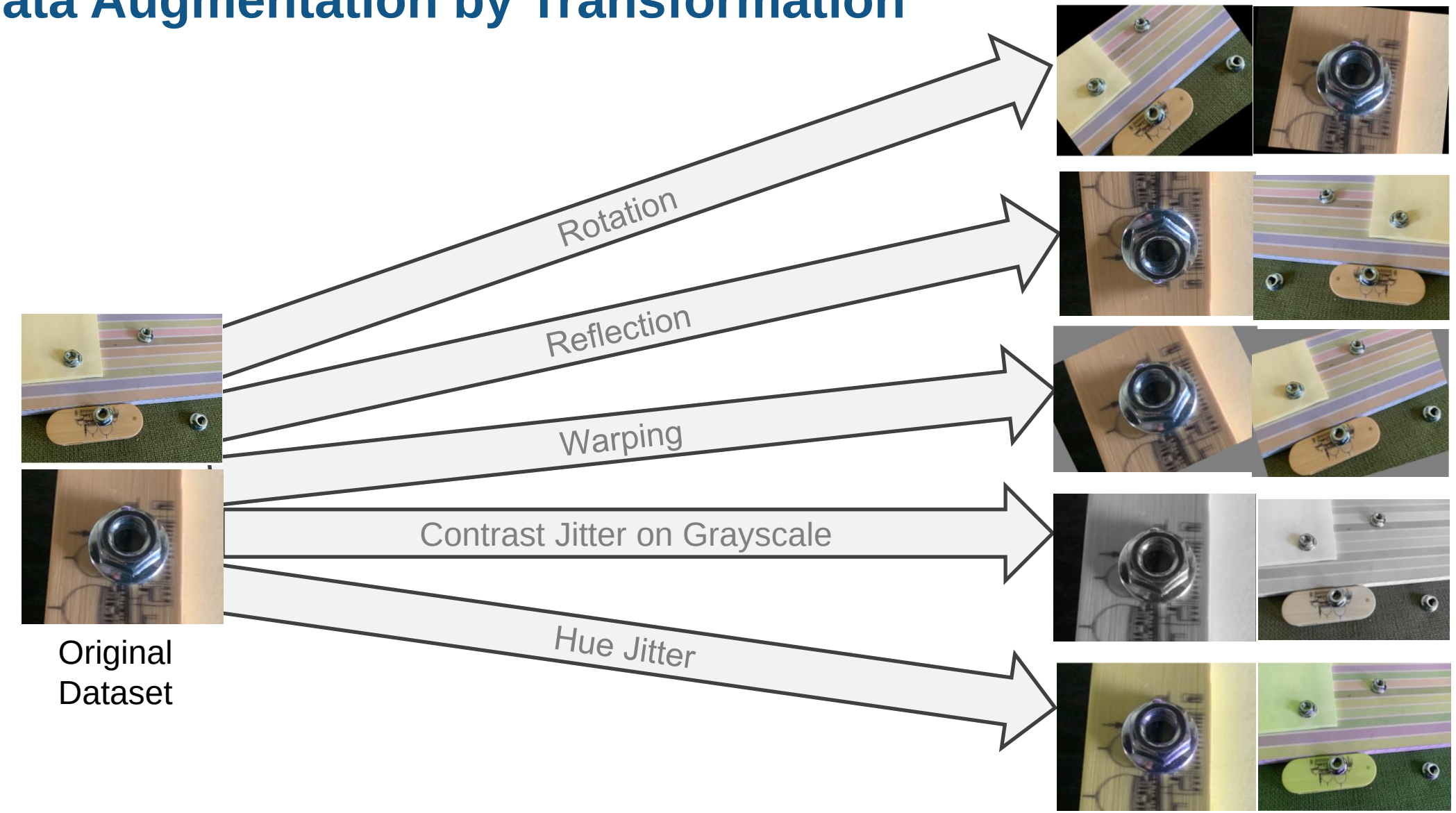
Big-Image Labeler



Data Access and Preprocessing – Common Challenges

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

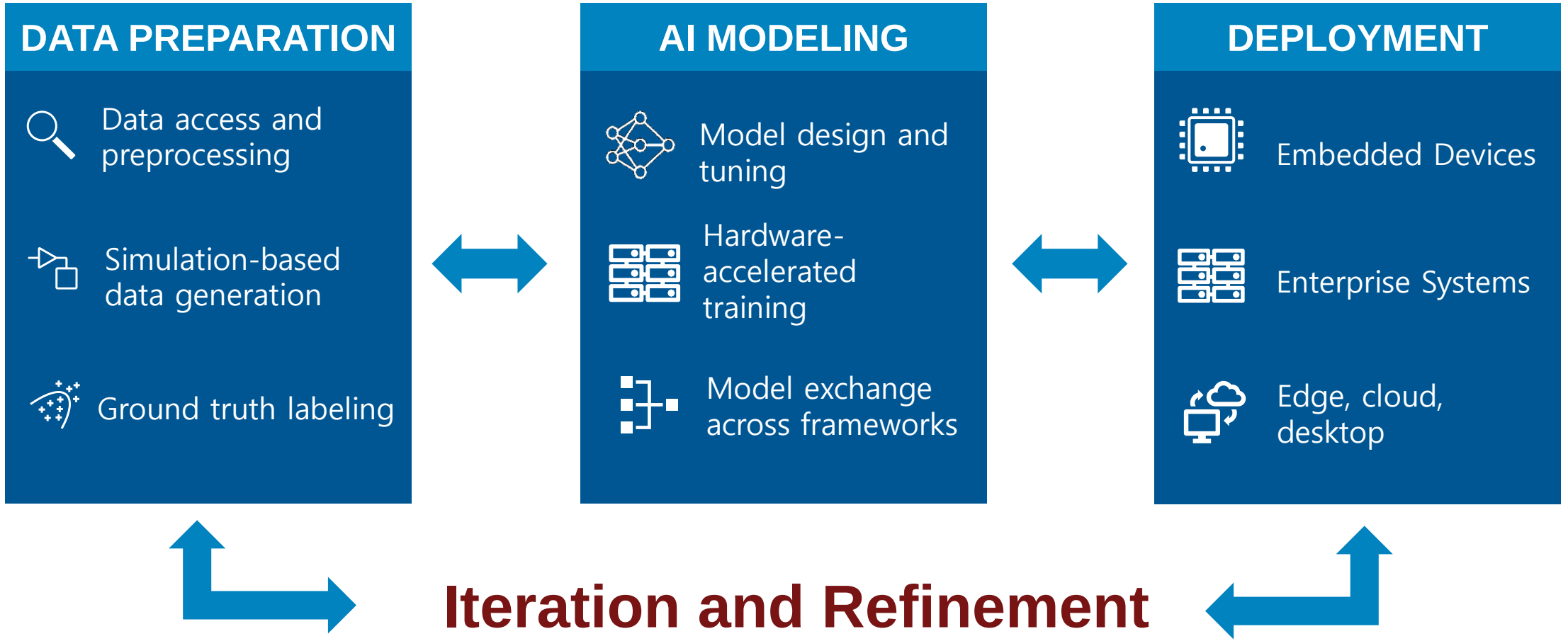
Data Augmentation by Transformation



Data Augmentation : Generative Adversarial Networks (GANs)

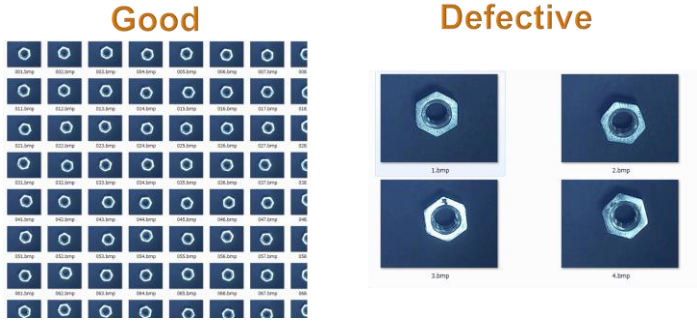
Generative Adversarial Networks

Defect Detection Workflow

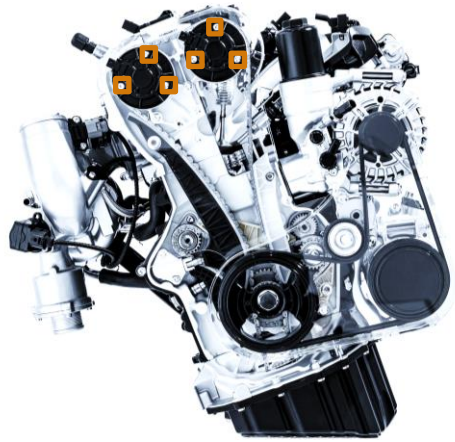


Deep Learning for Defect Detection

- Deep learning for Classification



- Deep Learning for Object Detection

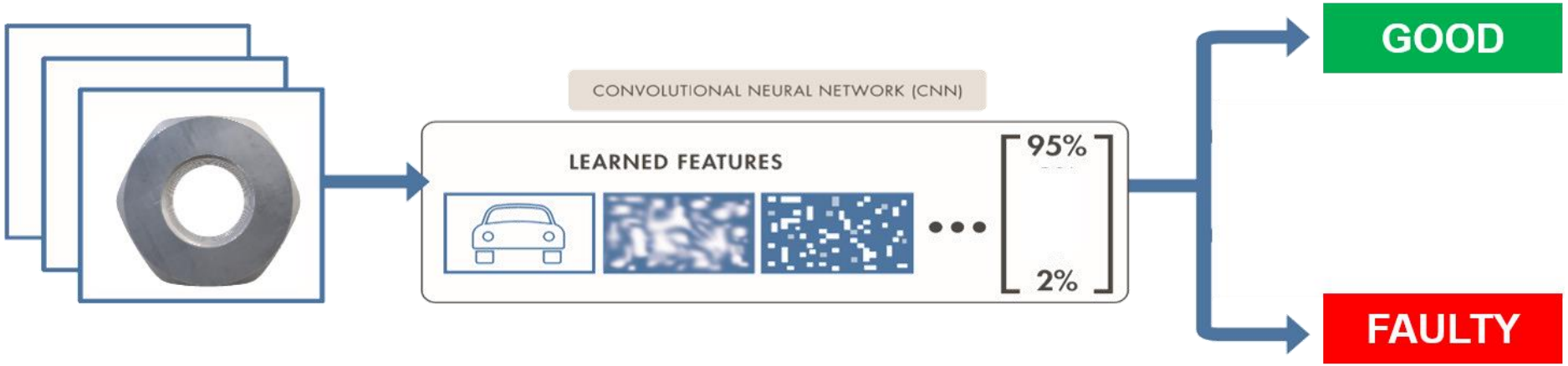


Deep Learning for Defect Detection – Multiple techniques

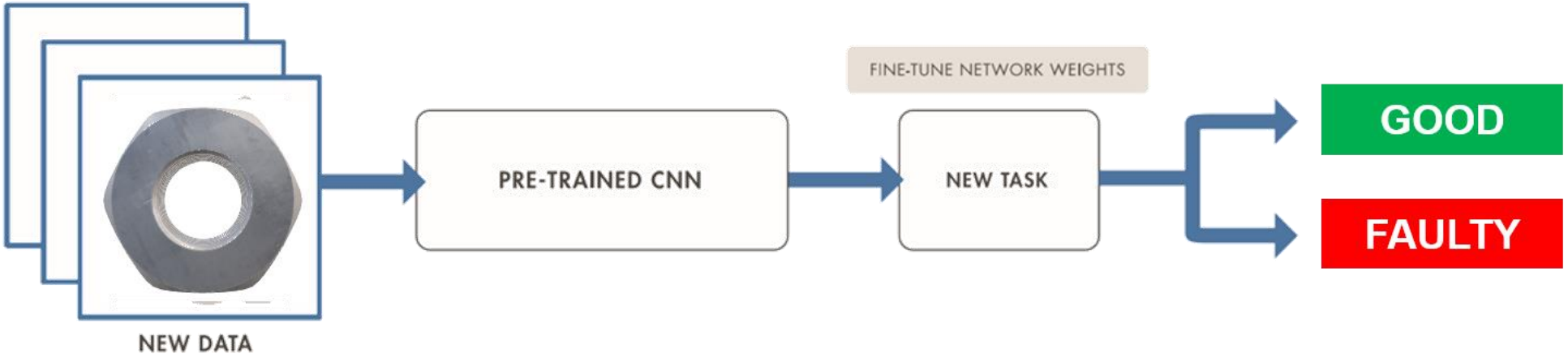
Deep learning for Classification

Two Approaches for Deep Learning

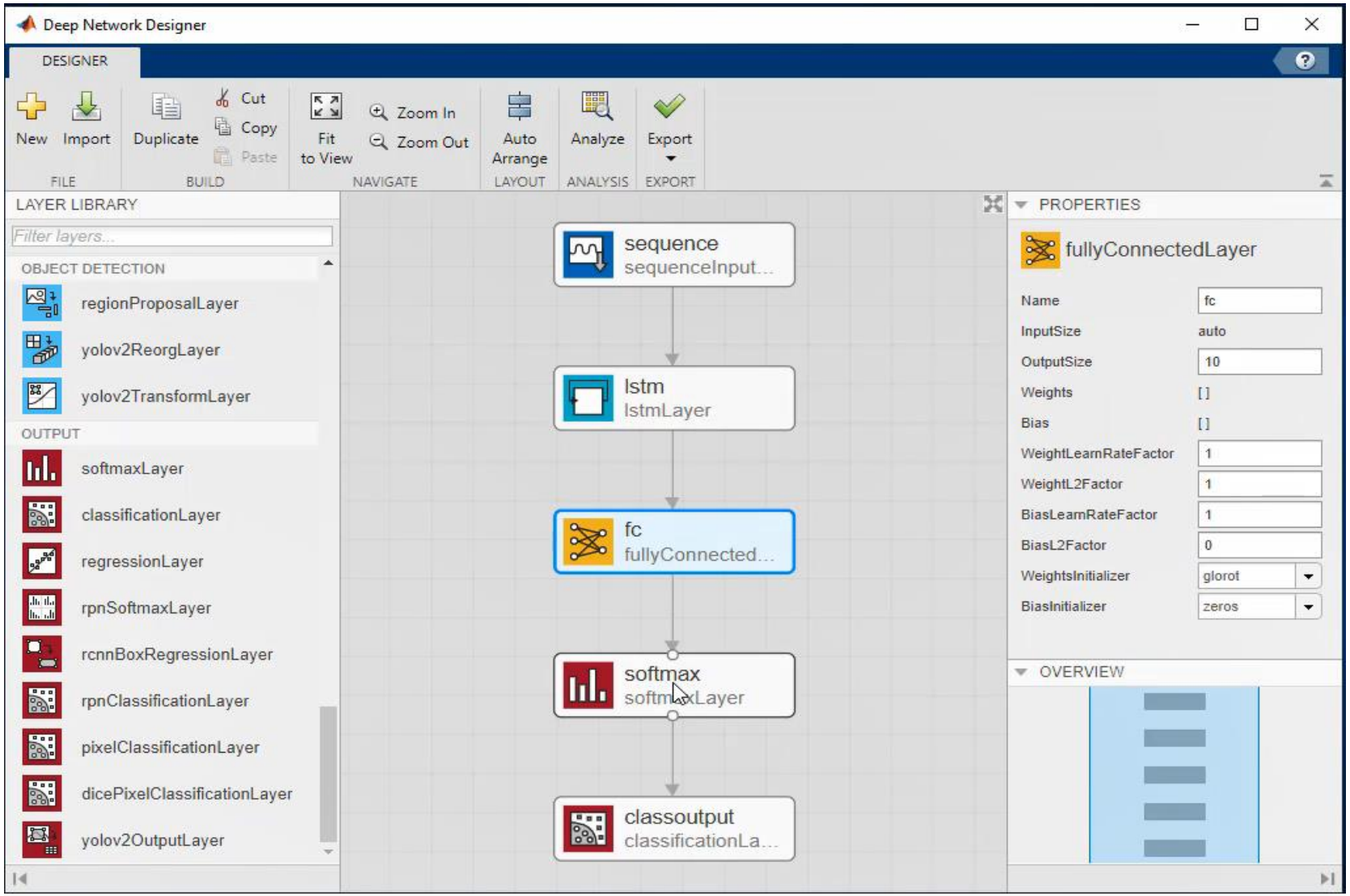
1. Train a deep neural network from scratch



2. Fine-tune a pre-trained model (transfer learning)

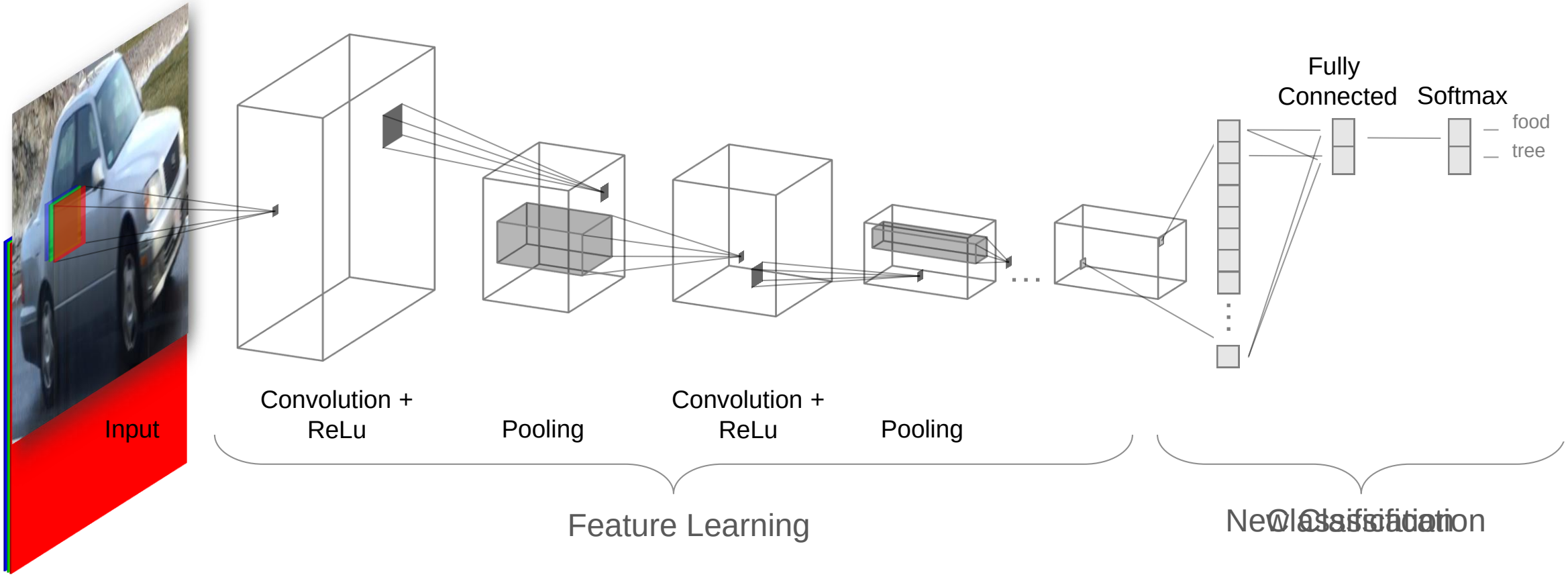


Train a Deep Neural Network from Scratch

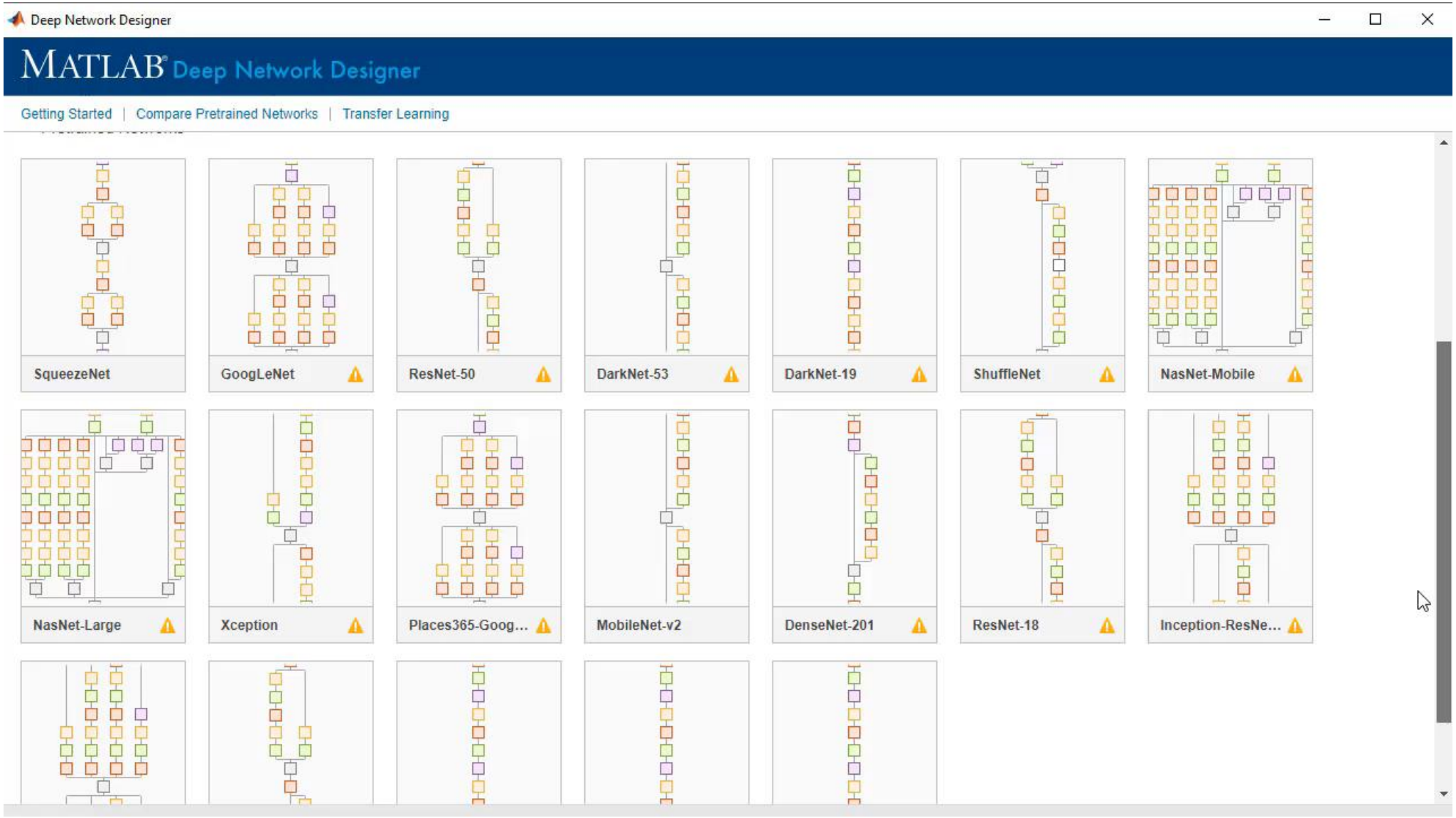


Two approaches for Deep learning

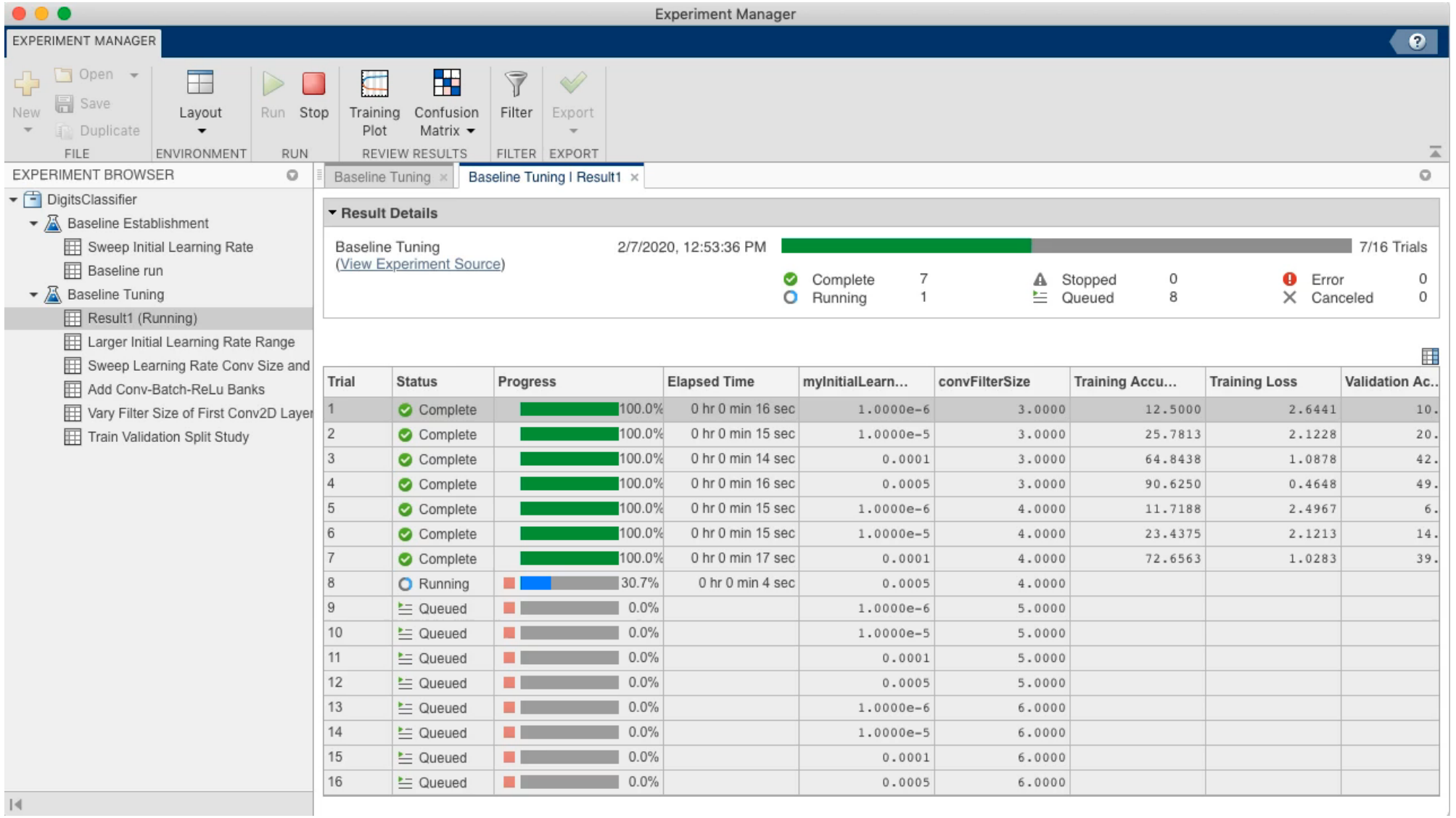
Approach 2. Fine-tune a pre-trained model (Transfer learning)



Fine-tune a Pre-trained Model (Transfer Learning)

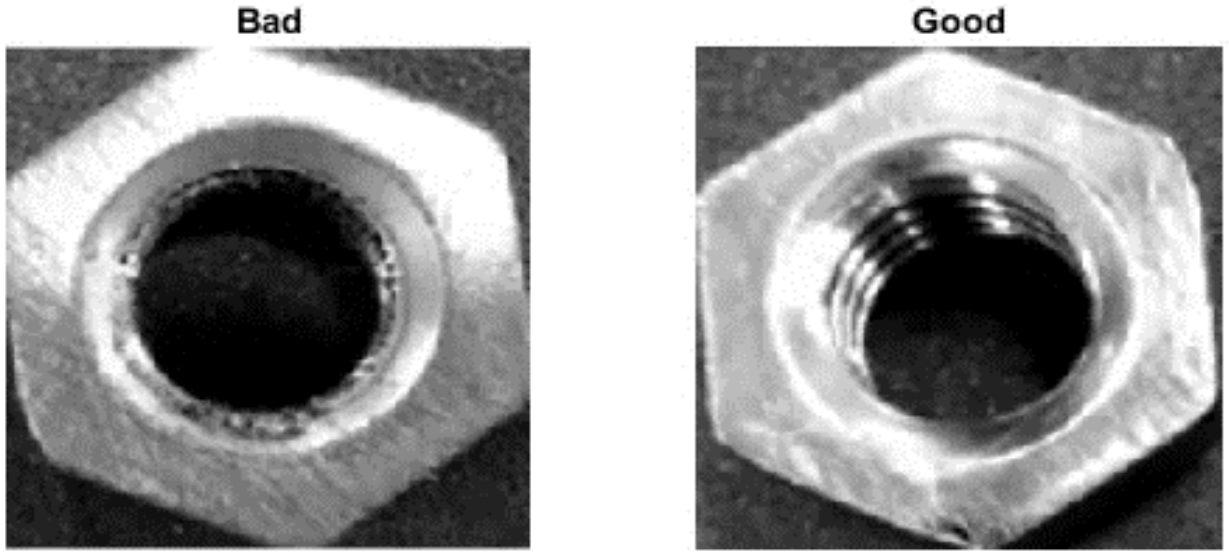


Experiment Manager



Trial	Status	Progress	Elapsed Time	myInitialLearn...	convFilterSize	Training Accu...	Training Loss	Validation Ac..
1	Complete	100.0%	0 hr 0 min 16 sec	1.0000e-6	3.0000	12.5000	2.6441	10.
2	Complete	100.0%	0 hr 0 min 15 sec	1.0000e-5	3.0000	25.7813	2.1228	20.
3	Complete	100.0%	0 hr 0 min 14 sec	0.0001	3.0000	64.8438	1.0878	42.
4	Complete	100.0%	0 hr 0 min 16 sec	0.0005	3.0000	90.6250	0.4648	49.
5	Complete	100.0%	0 hr 0 min 15 sec	1.0000e-6	4.0000	11.7188	2.4967	6.
6	Complete	100.0%	0 hr 0 min 15 sec	1.0000e-5	4.0000	23.4375	2.1213	14.
7	Complete	100.0%	0 hr 0 min 17 sec	0.0001	4.0000	72.6563	1.0283	39.
8	Running	30.7%	0 hr 0 min 4 sec	0.0005	4.0000			
9	Queued	0.0%		1.0000e-6	5.0000			
10	Queued	0.0%		1.0000e-5	5.0000			
11	Queued	0.0%		0.0001	5.0000			
12	Queued	0.0%		0.0005	5.0000			
13	Queued	0.0%		1.0000e-6	6.0000			
14	Queued	0.0%		1.0000e-5	6.0000			
15	Queued	0.0%		0.0001	6.0000			
16	Queued	0.0%		0.0005	6.0000			

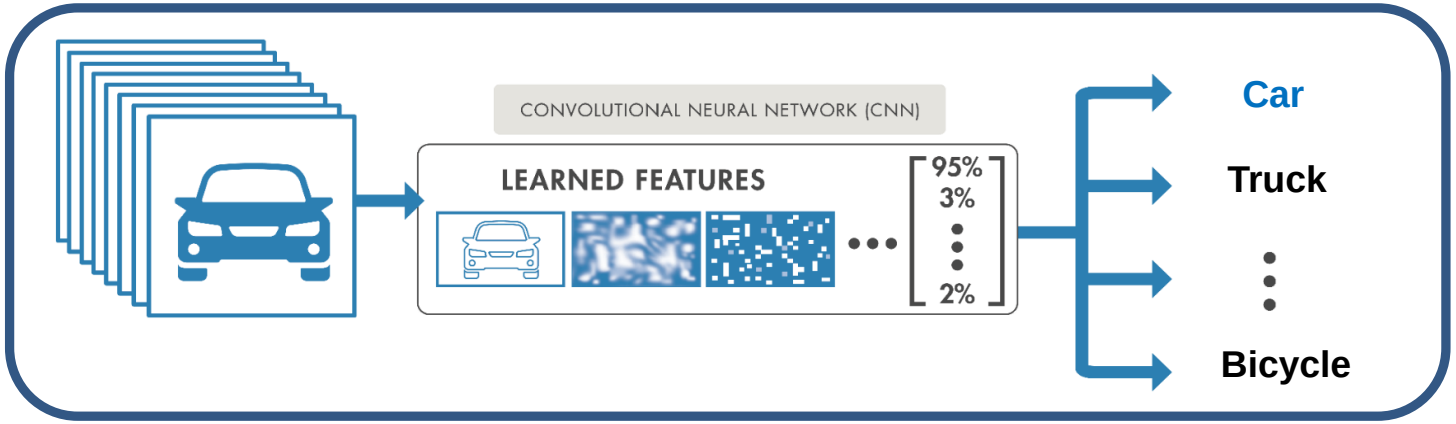
Classification with Trained MobileNetV2



Why the defect nuts are classified as ‘Bad’?

Challenges with Deep Learning Models

- Deep Learning model is a black box model



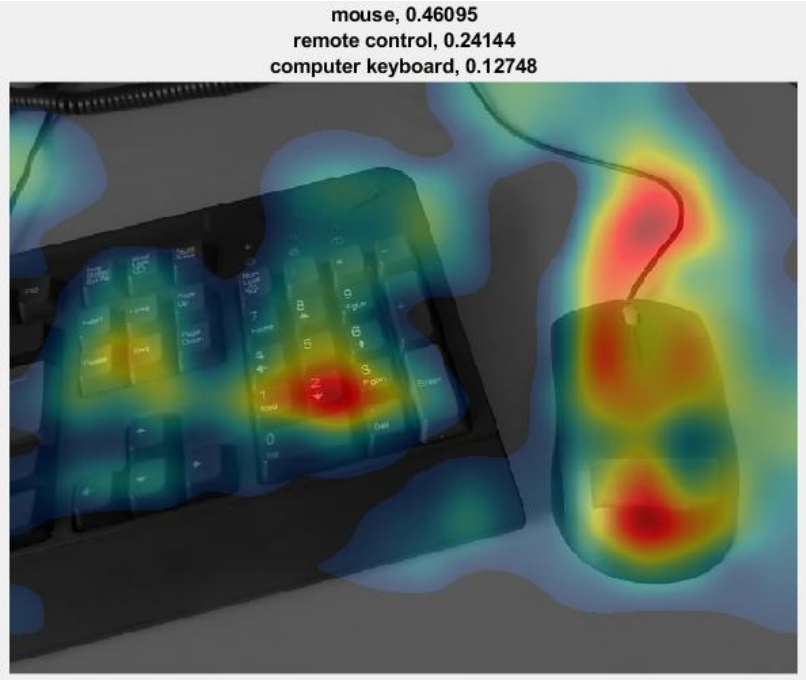
  - Is it possible to classify an unknown image correctly?
  - Why the model misclassify for certain images?

Explainable AI is required

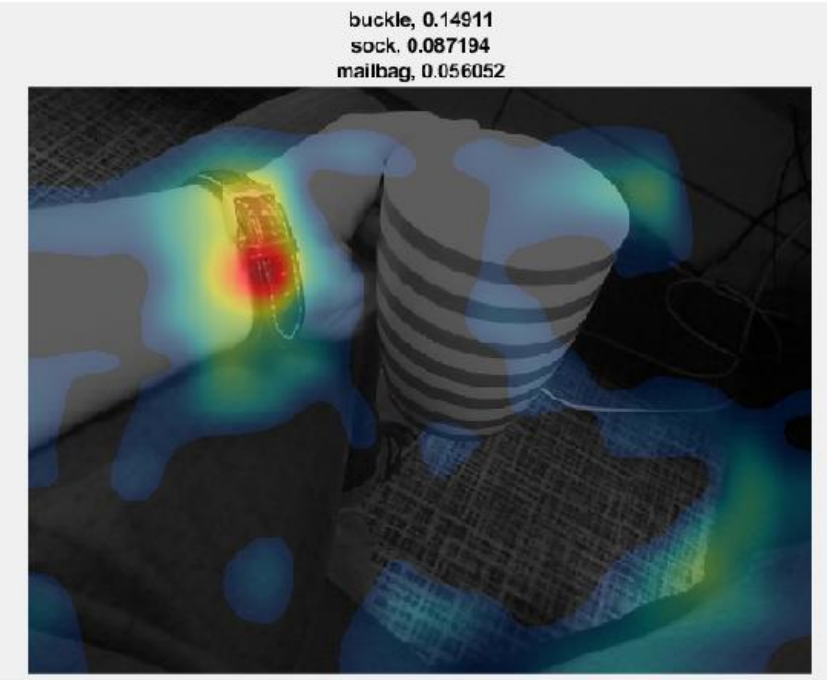
- Class Activation Mapping（CAM）
- Grad-CAM

Class Activation Mapping to Investigate Network Predictions

Attribution Reveals the Why Behind Deep Learning Decisions [Full code available]



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

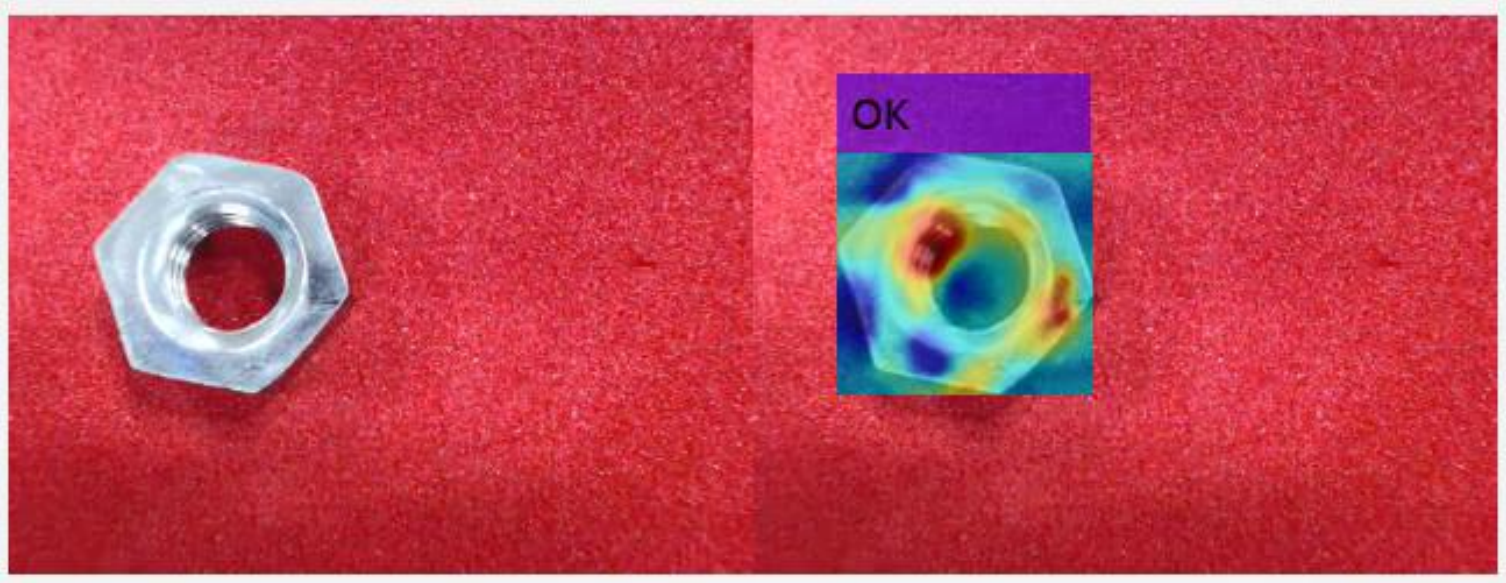


Incorrectly classified “coffee mug” as “buckle” due to the watch

Visualization of Features with CAM

Captured Image

Classification and CAM



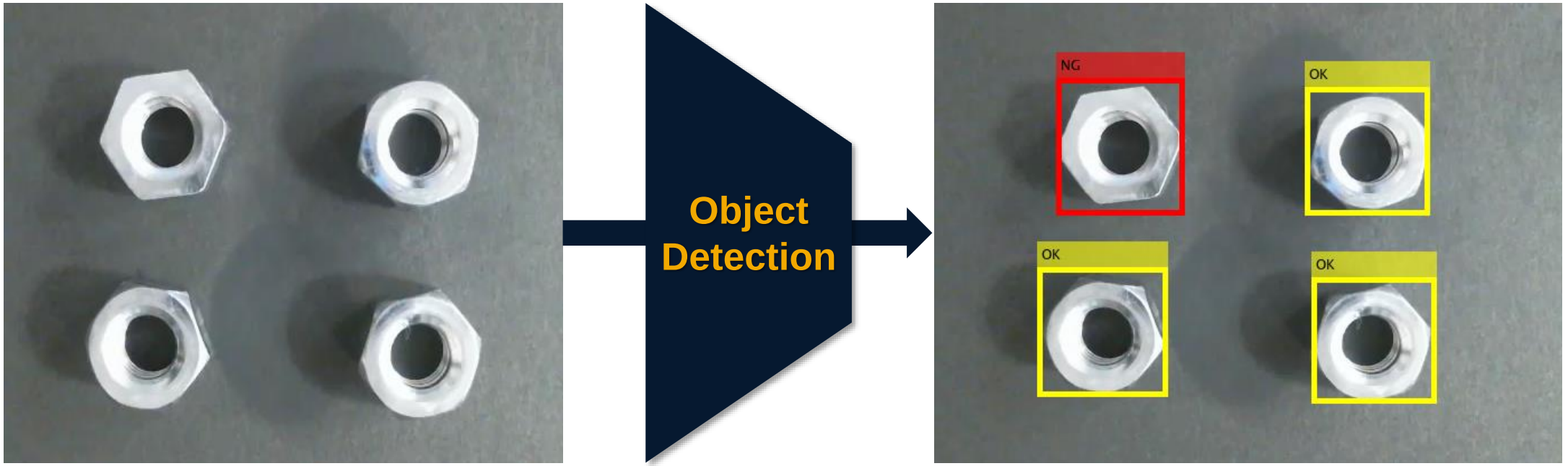
OK → Reacts to whole surface

Bad→ Reacts to the scratch

The network judges the unit as Bad by seeing the scratched area

Deep Learning for Object Detection

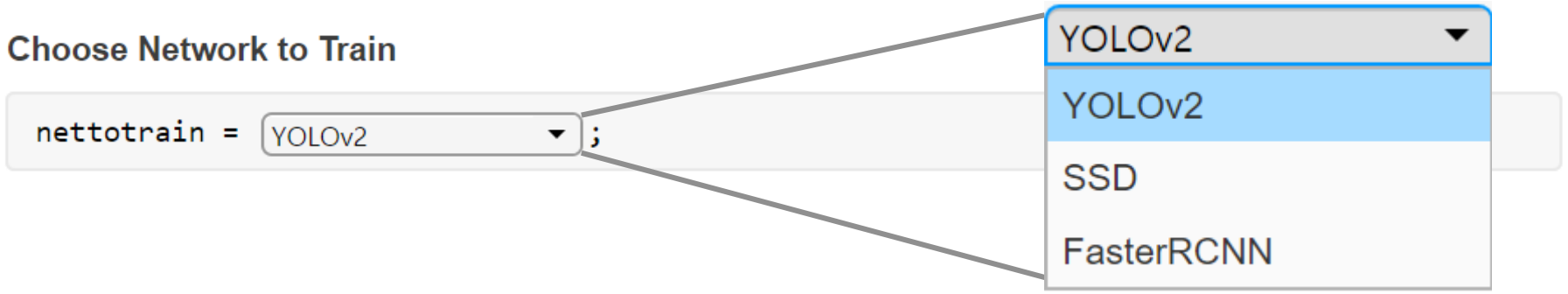
Object Detection in Image/Vision System



- **Object detection**
  - Computer technology related to computer vision and image processing that deals with **detecting instances of semantic objects** of a certain class (such as humans, buildings, or cars) in digital images and videos

Increase productivity using built-in functions for design and analysis

Use MATLAB high-level API for changing network type, backbone network for better performance easily.



AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

Practical Object Detection Examples using Deep Learning

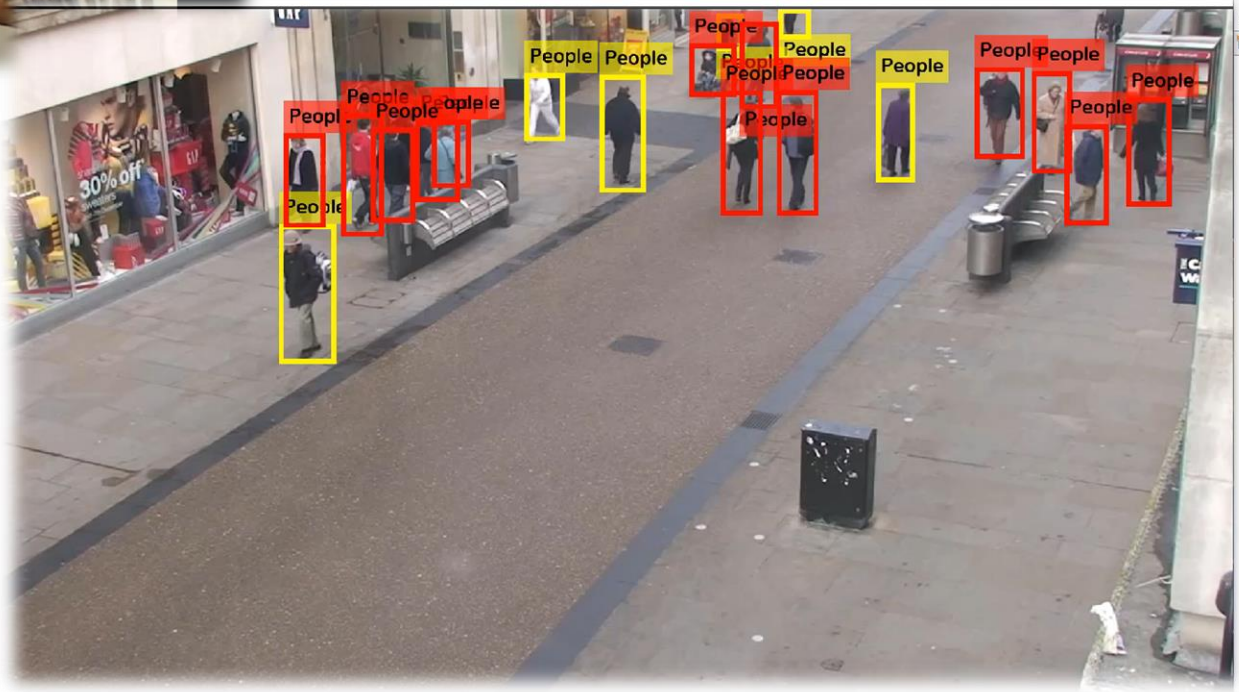
Repository Link

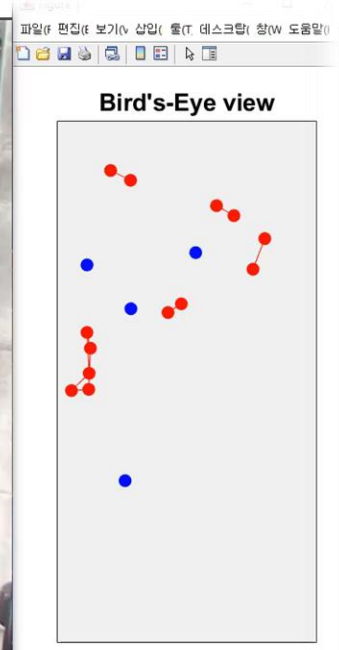
MATLAB Deep Learning

mathworks.github.io

https://www.mathworks.com/solutions/deep-le...

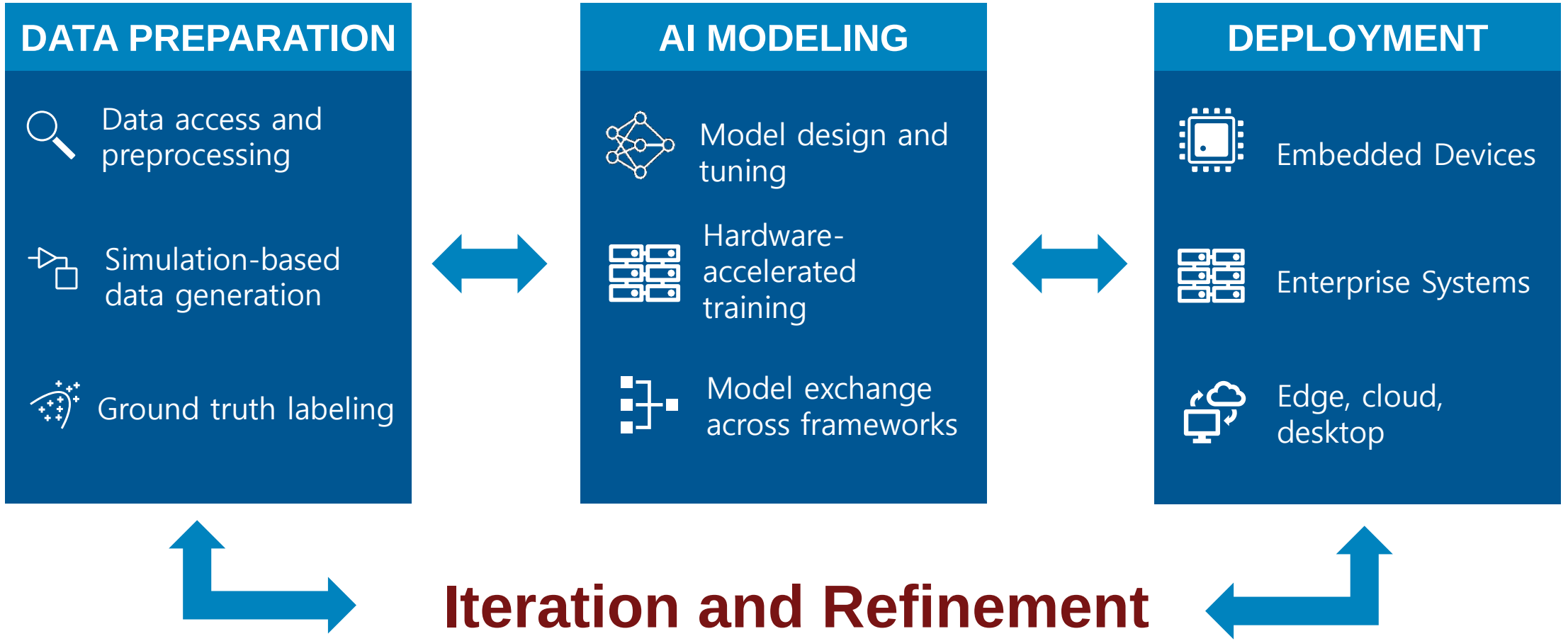
Repository Link



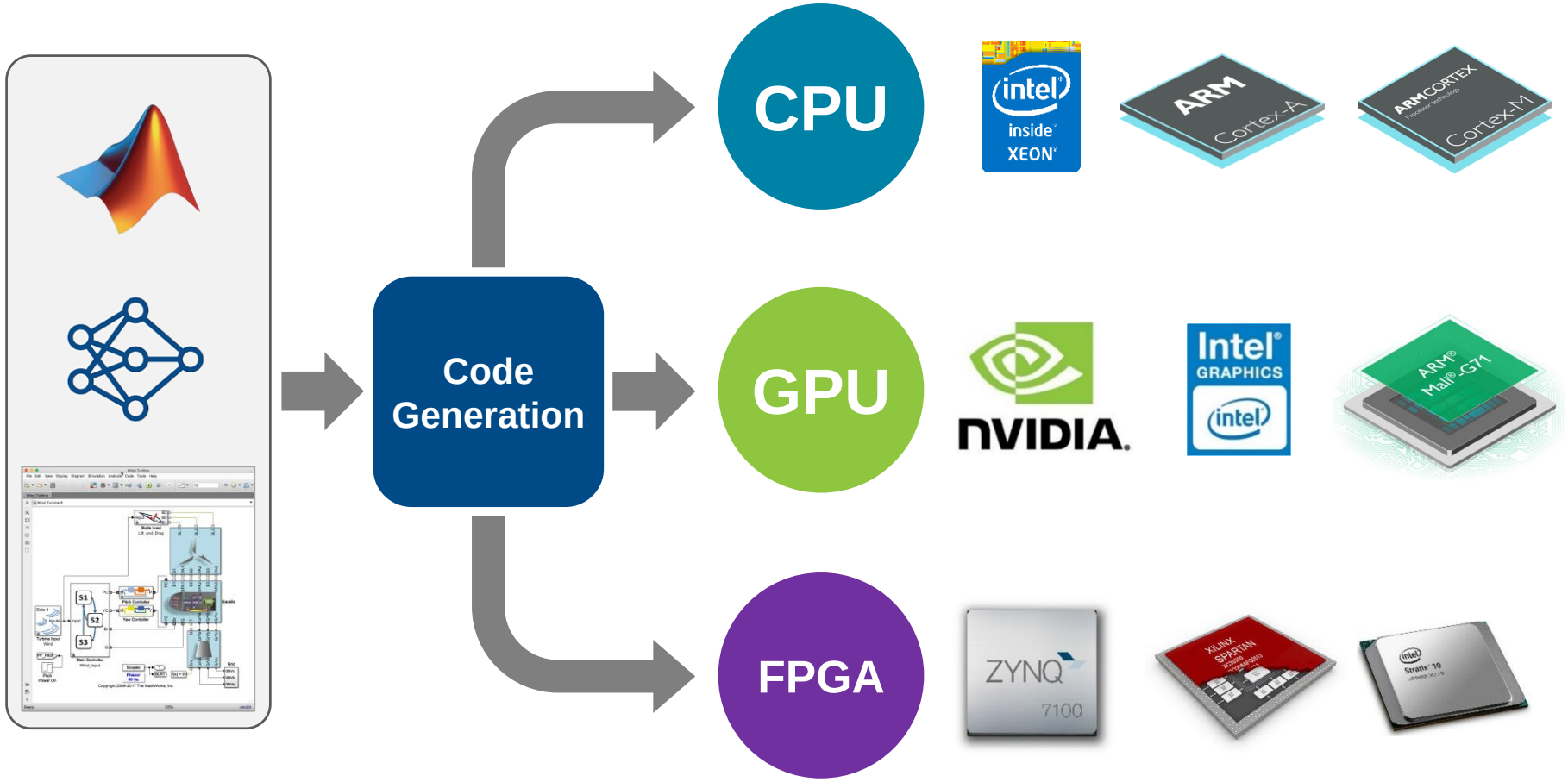


Repository Link

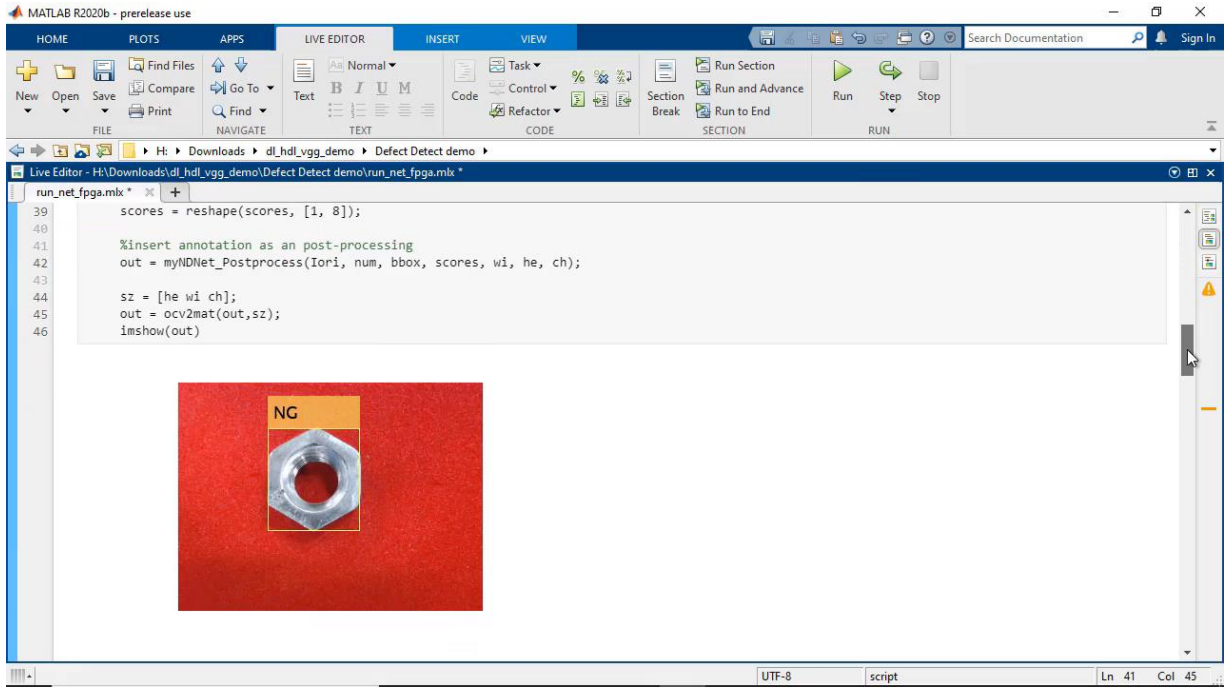
Defect Detection Workflow



Deploy to Any Processor with Best-in-class Performance



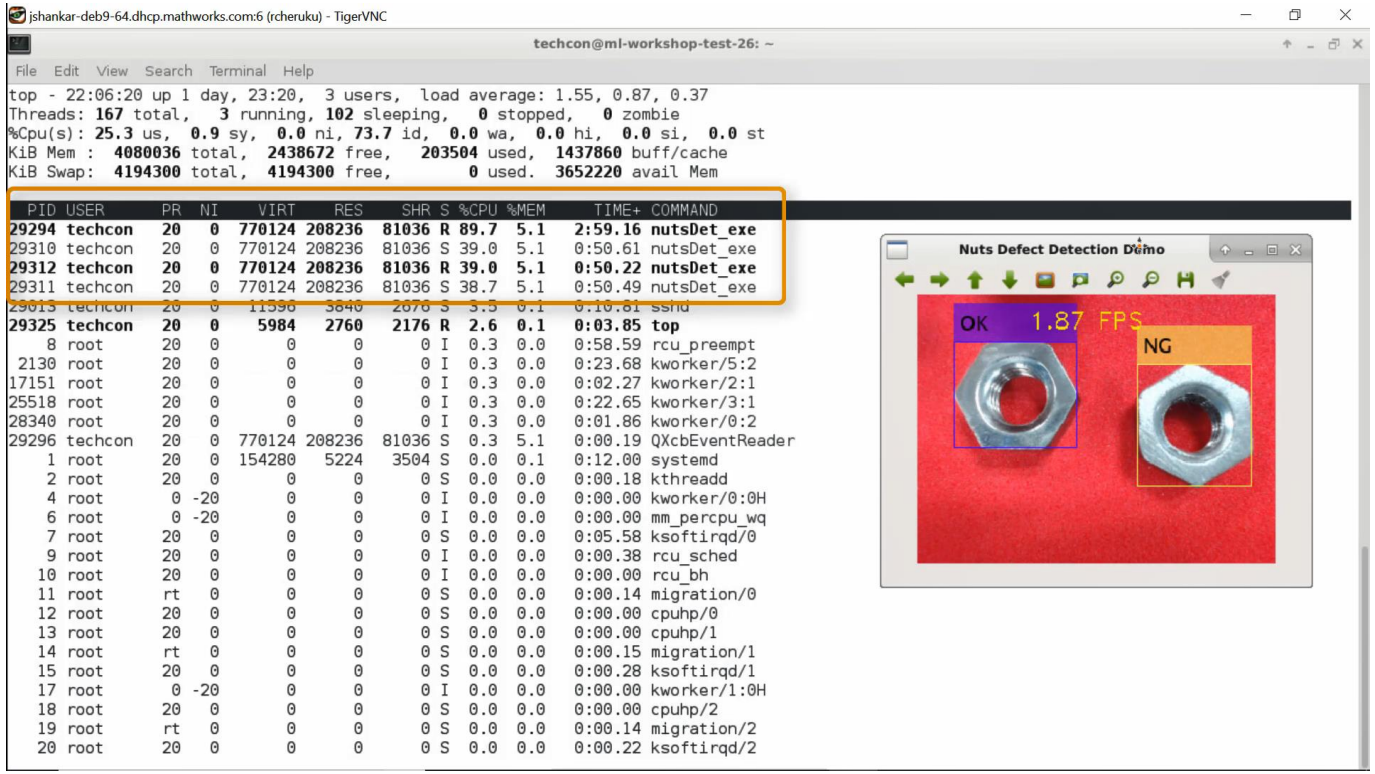
Deploy to Hardware



Deploy defect detection algorithms from MATLAB to ZCU102 board from Xilinx

Deploy defect detection algorithms from MATLAB to Jetson AGX Xavier

Deploy to Hardware

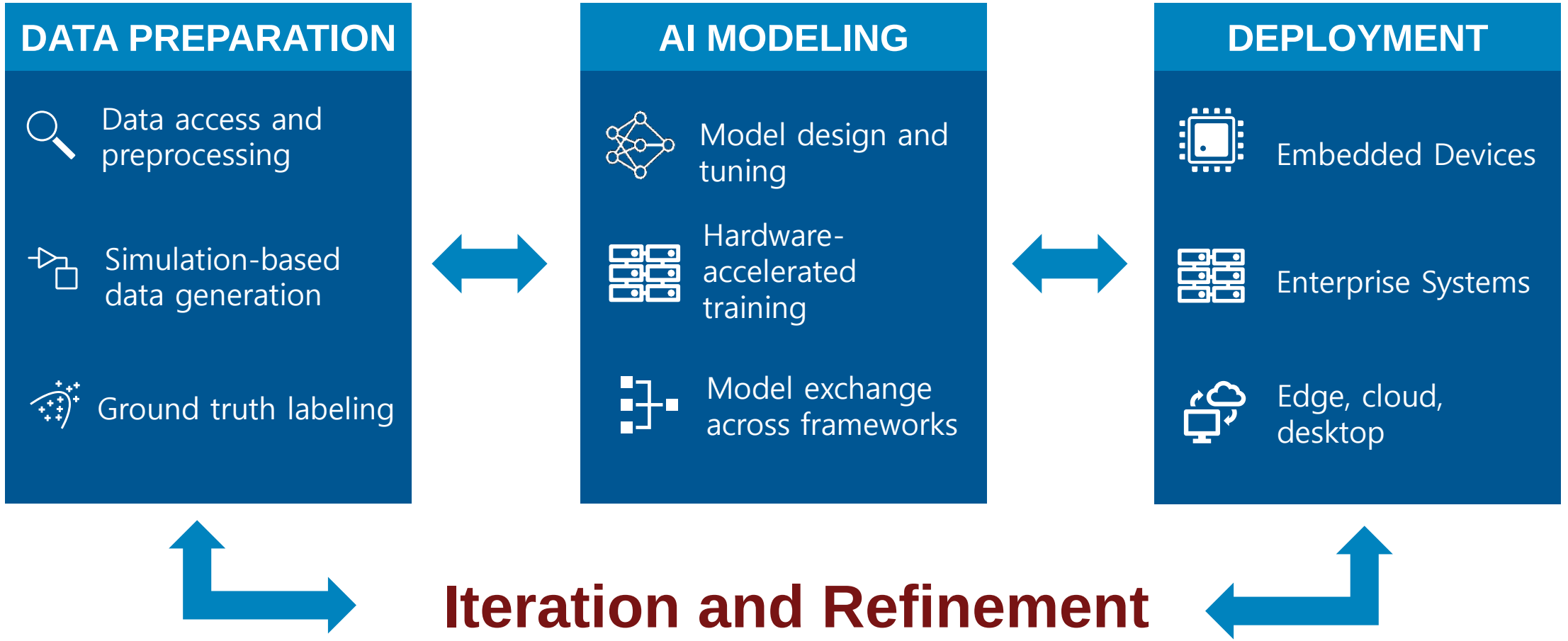


Defect detection deployed on ARM Cortex-A microprocessor

Additional Resources

- Deploying Deep Neural Networks to GPUs and CPUs Using MATLAB Coder and GPU Coder
- Using GPU Coder to Prototype and Deploy on NVIDIA Drive, Jetson
- Real-Time Object Detection with YOLO v2 Using GPU Coder
- Image Classification on ARM CPU: SqueezeNet on Raspberry Pi
- Deep Learning on an Intel Processor with MKL-DNN

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 defect detection from data access to deployment

Thank You