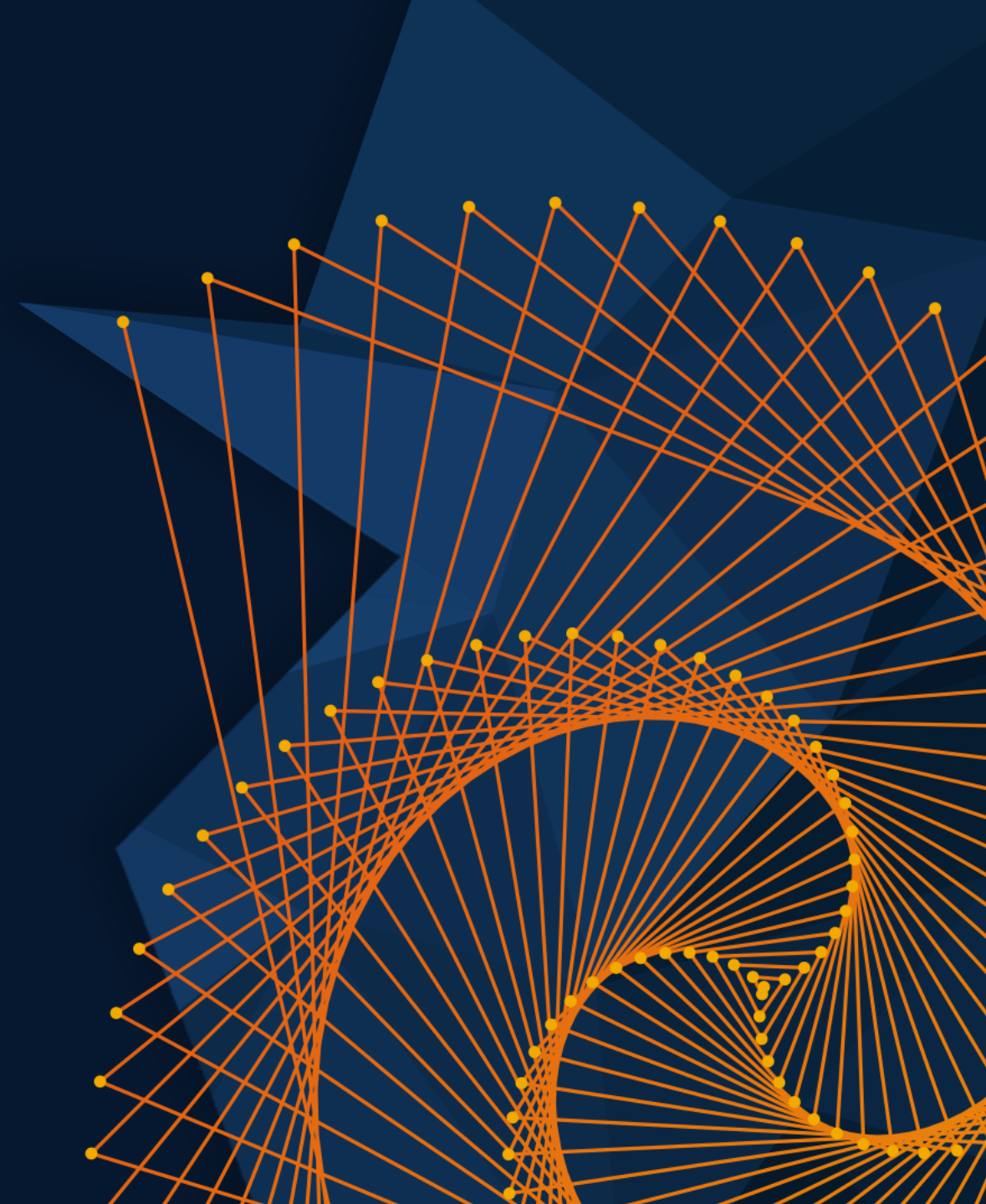


MATLAB EXPO

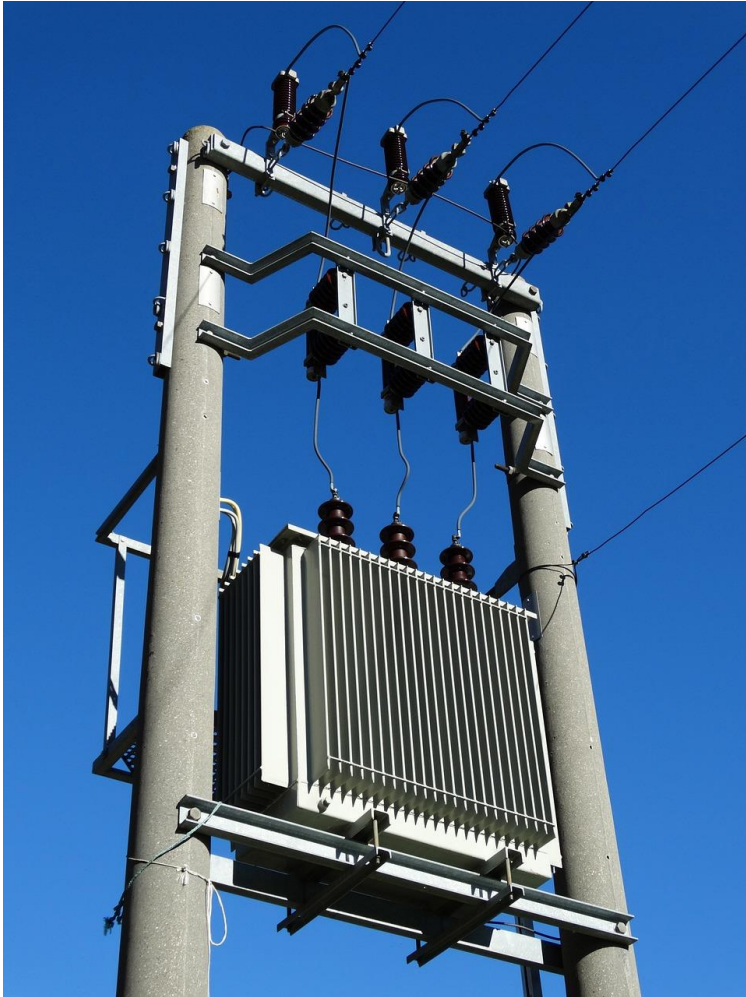
 UNITED KINGDOM

Automating Fault Detection Using Visual Inspection

Elre Oldewage, MathWorks



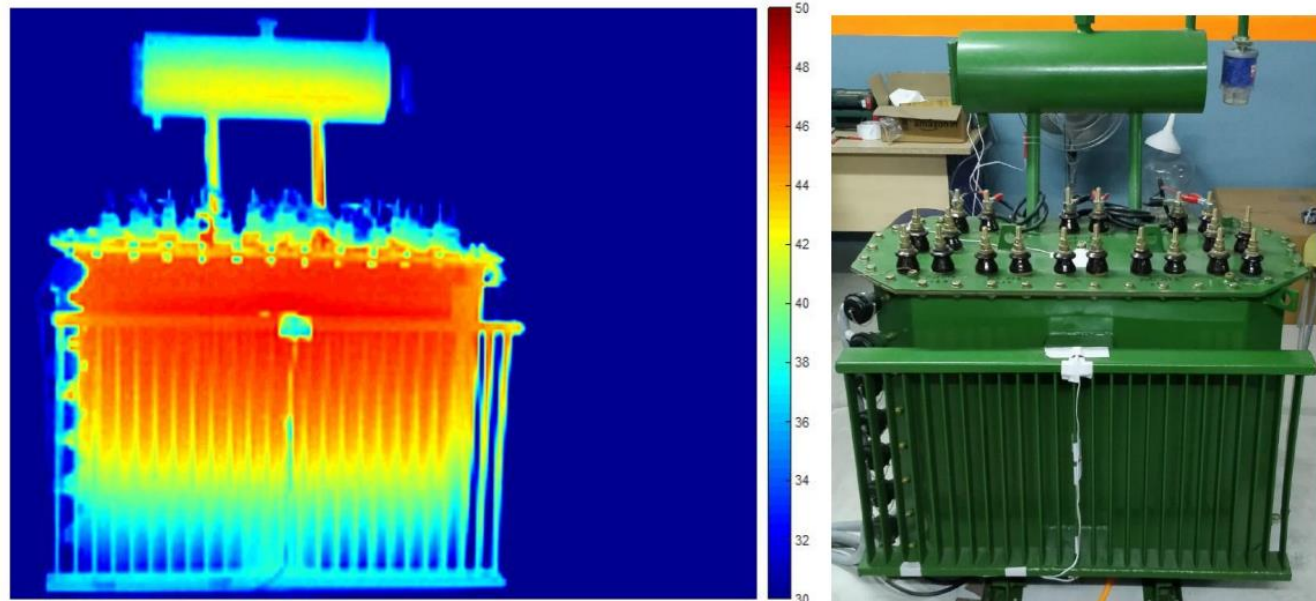
Siemens Develops Health Monitoring System for Distribution Transformers



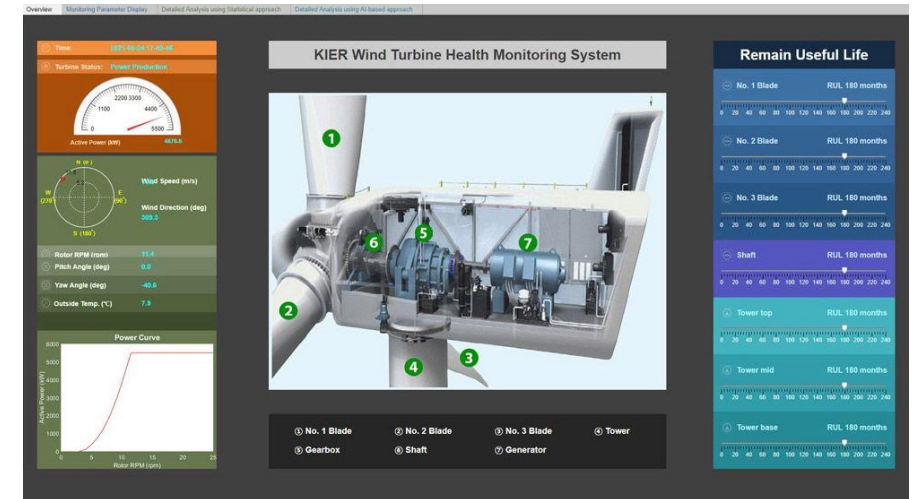
User Story

Results

- Retrofittable solution with non-invasive temperature sensors
- User-friendly commissioning
- Online learning for algorithm



Korea Institute of Energy Research Develops AI-Based Predictive Maintenance Models for Offshore Wind Power



Results

- Development time cut in half
- 90%+ prediction accuracy achieved
- Aggressive deadline met

Mondi Implements Statistics-Based Health Monitoring and Predictive Maintenance for Manufacturing Processes with Machine Learning



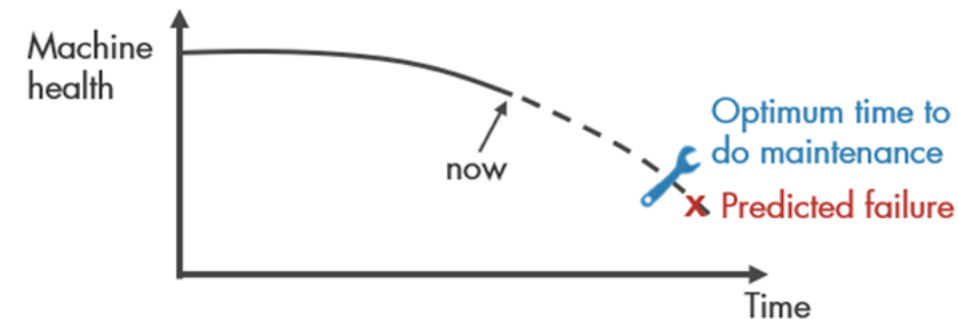
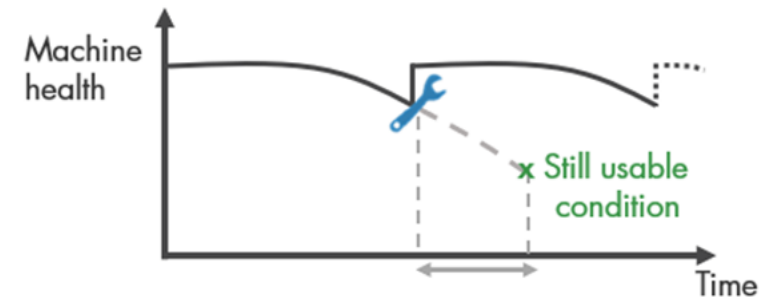
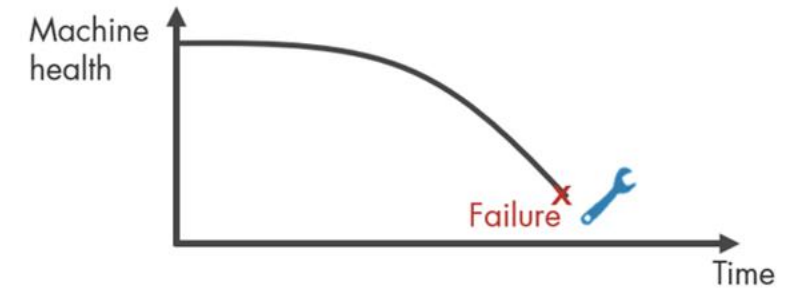
User Story

Results

- More than 50,000 euros saved per year
- Prototype completed in six months
- Production software run 24/7

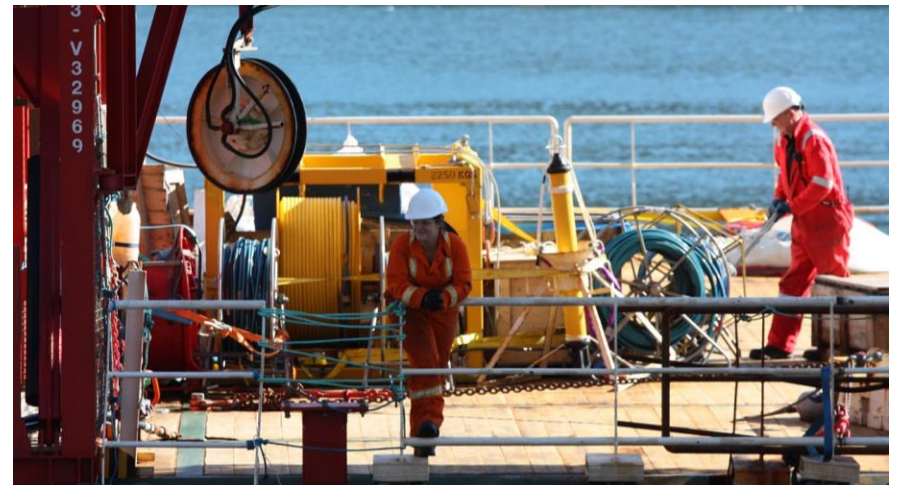
How best to do maintenance?

- **Reactive** – Do maintenance once there's a problem
 - Problem: unexpected failures can be expensive and potentially dangerous
- **Scheduled** – Do maintenance at a regular rate
 - Problem: unnecessary maintenance can be wasteful; may not eliminate all failures
- **Predictive** – Forecast when problems will arise
 - Problem: difficult to make accurate forecasts for complex equipment



Why perform predictive maintenance?

- Prevent loss or damage of expensive equipment
- Failures can be dangerous
- Maintenance also costly and possibly dangerous
- Reduced downtime
- Improved operating efficiency



What is visual inspection?

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

Automated Defect Detection

Computer Vision

Optical Inspection

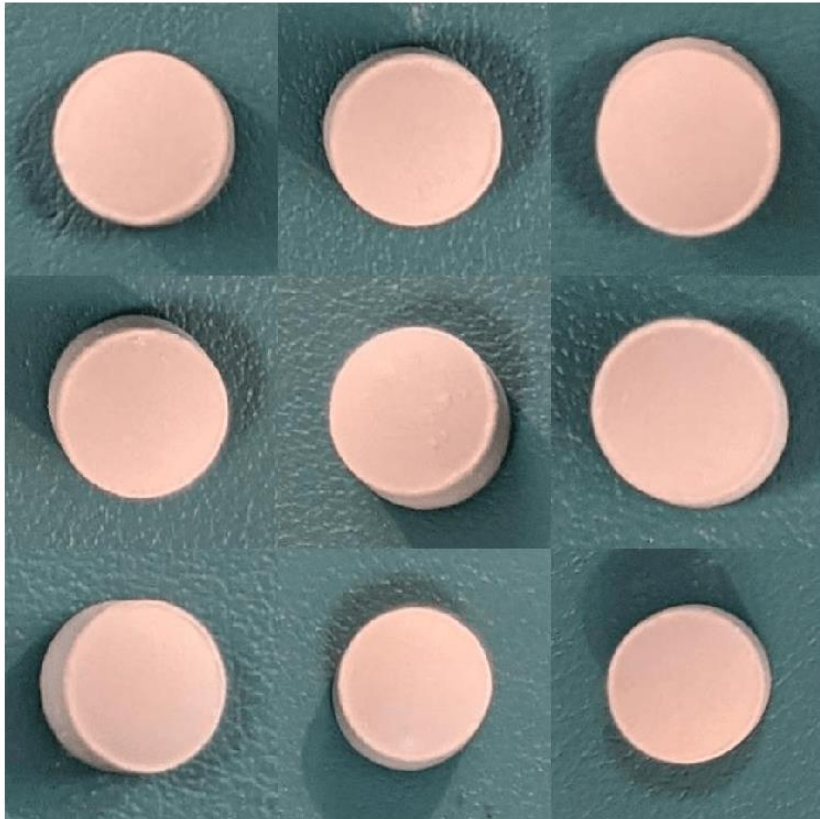
Automated Inspection

Image Processing

Machine Learning

Classical image processing

Detecting defective pills for quality control



Normal

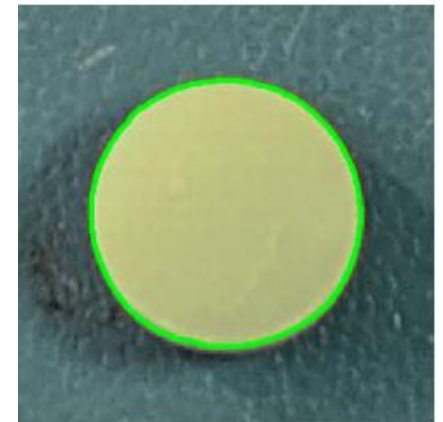
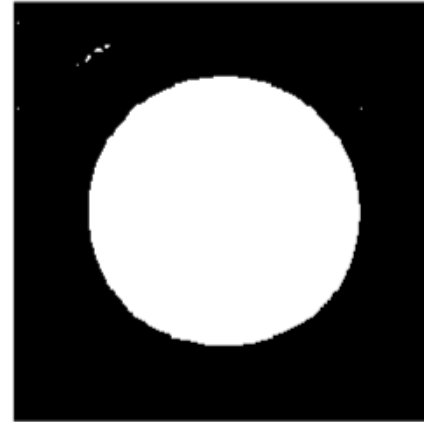
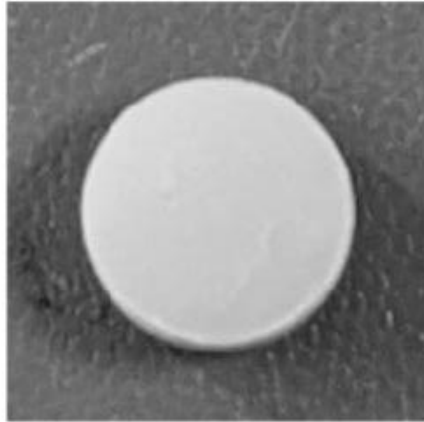
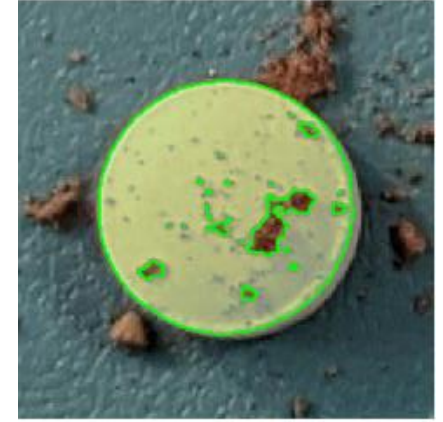
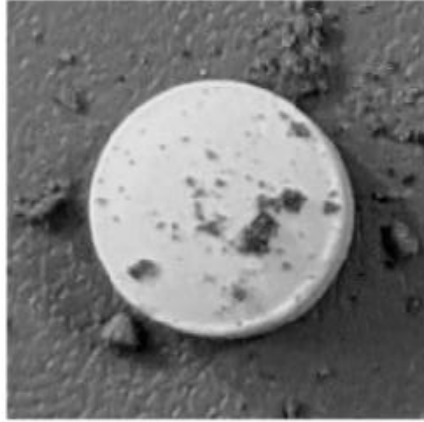
Dirty

Chipped

[Documentation example](#)

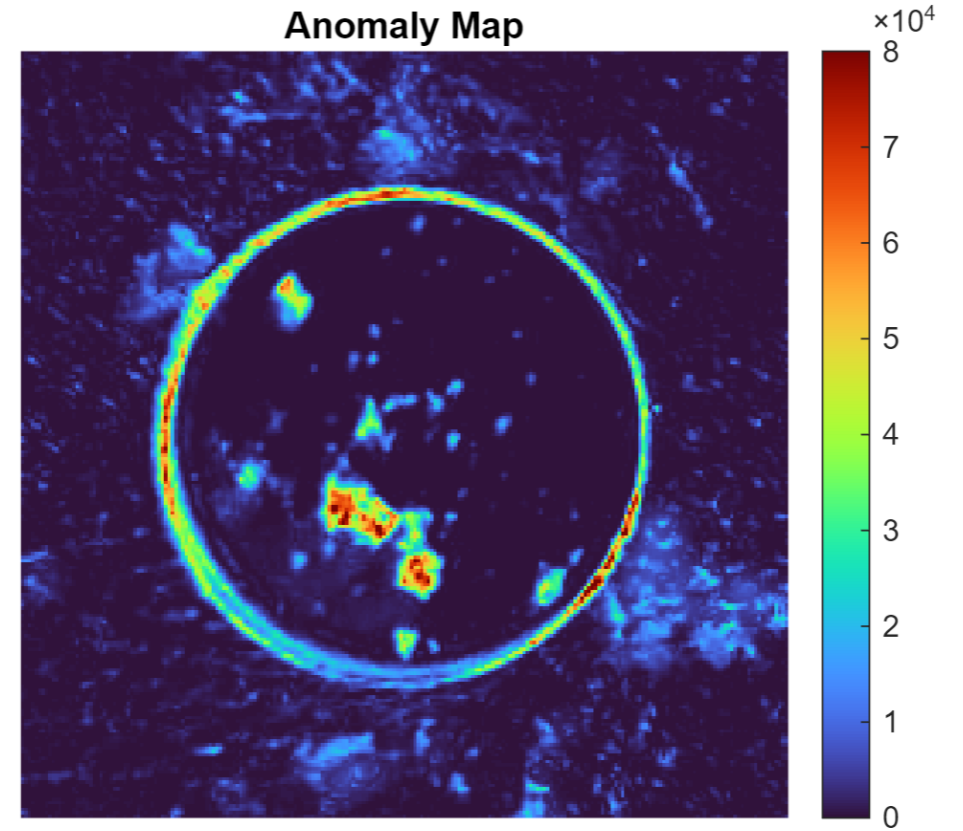
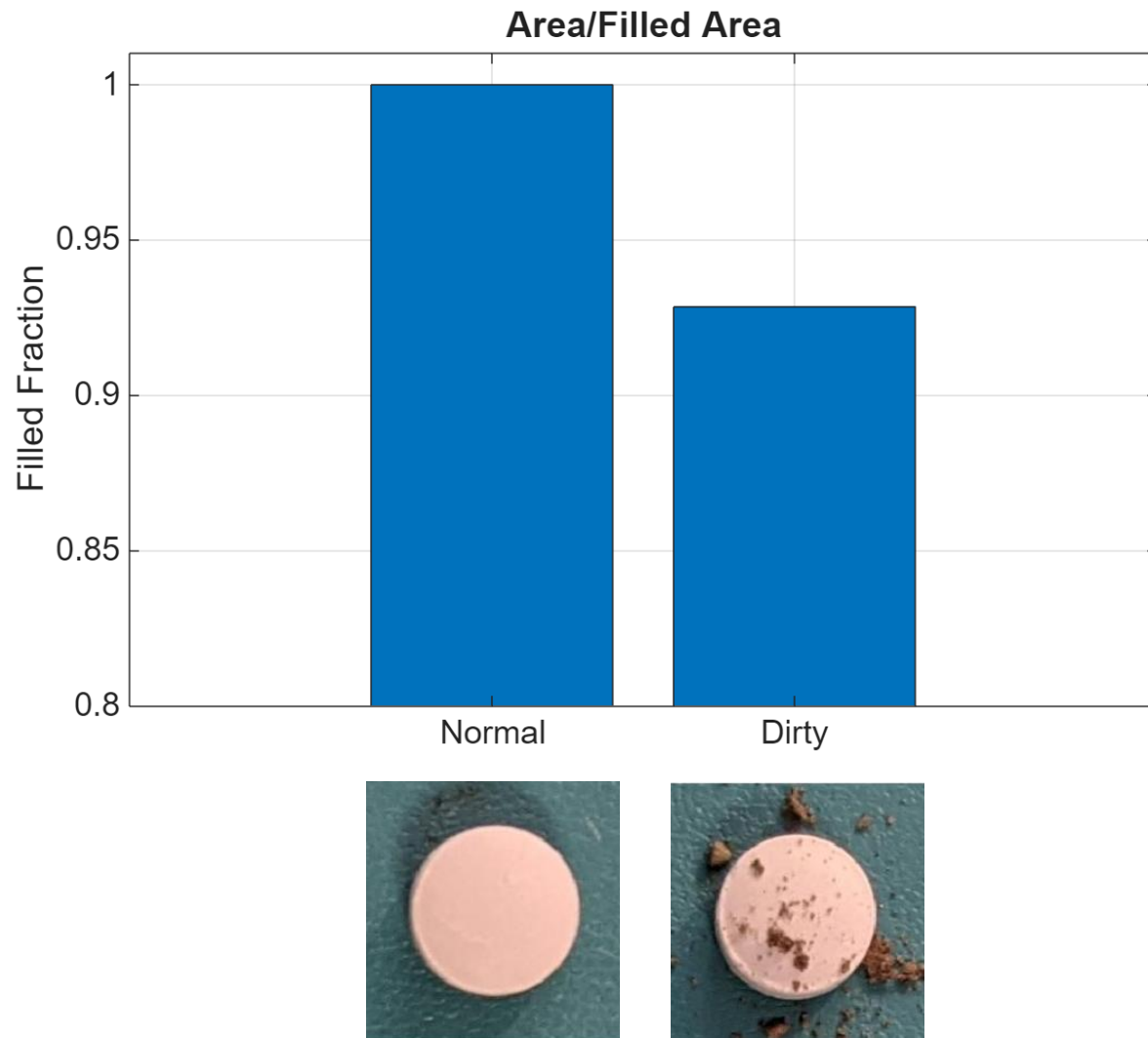
Classical image processing

Detecting defective pills for quality control



Classical image processing

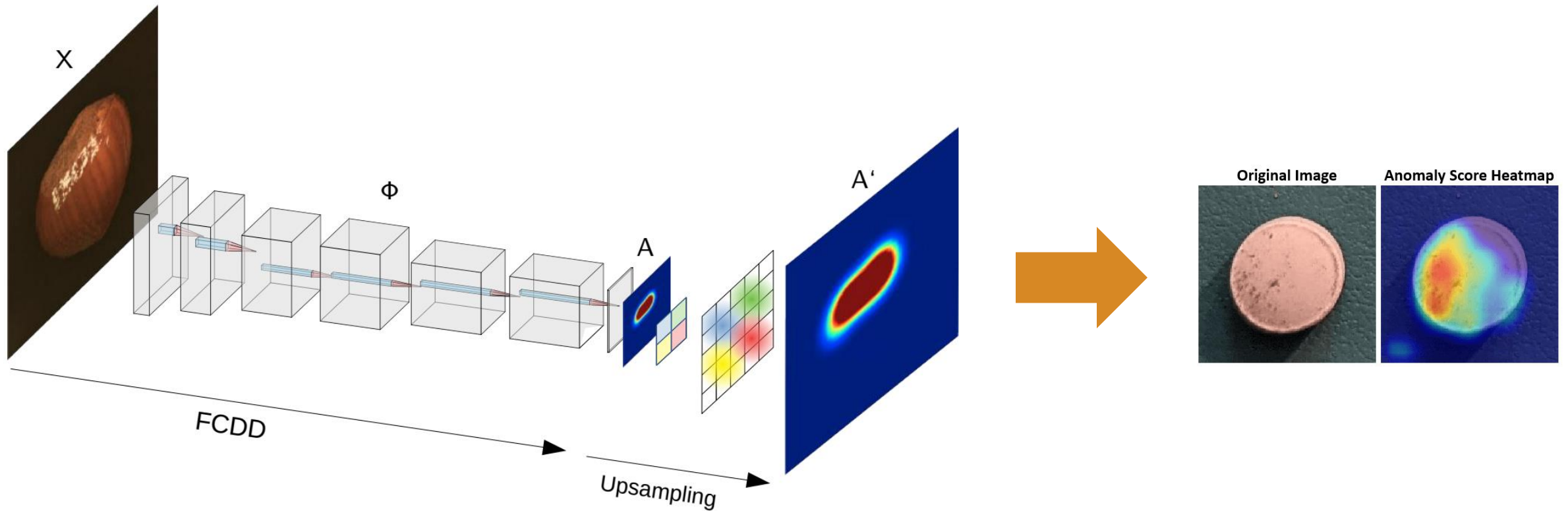
Detecting defective pills for quality control



```
normDiff = sum( (double(normal) - double(dirty)).^2 , 3 );
```

Deep learning

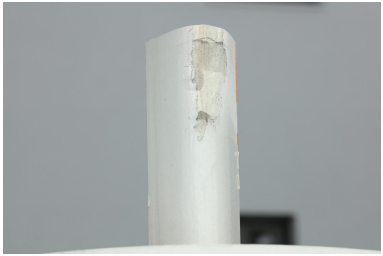
Detecting defective pills for quality control



Explainable Deep One-Class Classification, Liznerski et. al (2021)

<https://arxiv.org/pdf/2007.01760>

Challenge: Poor quality input data



Are conditions under which the data is gathered controlled?

Challenge: Difficulty identifying anomalous inputs



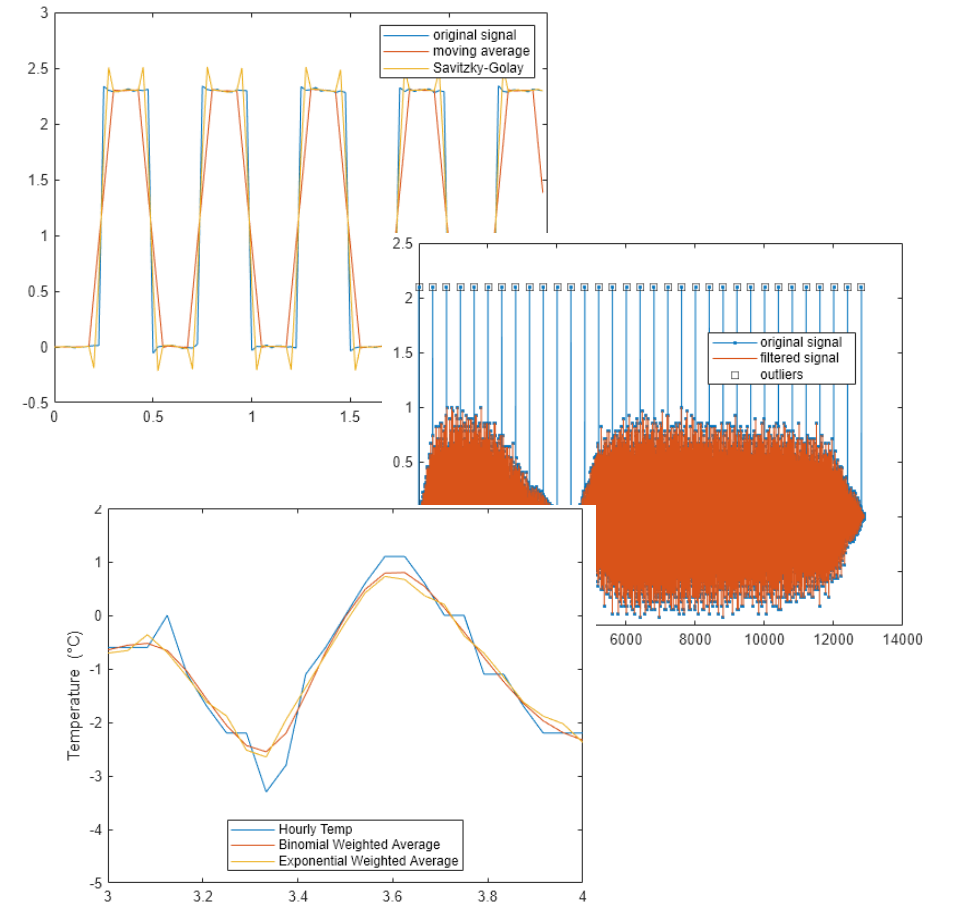
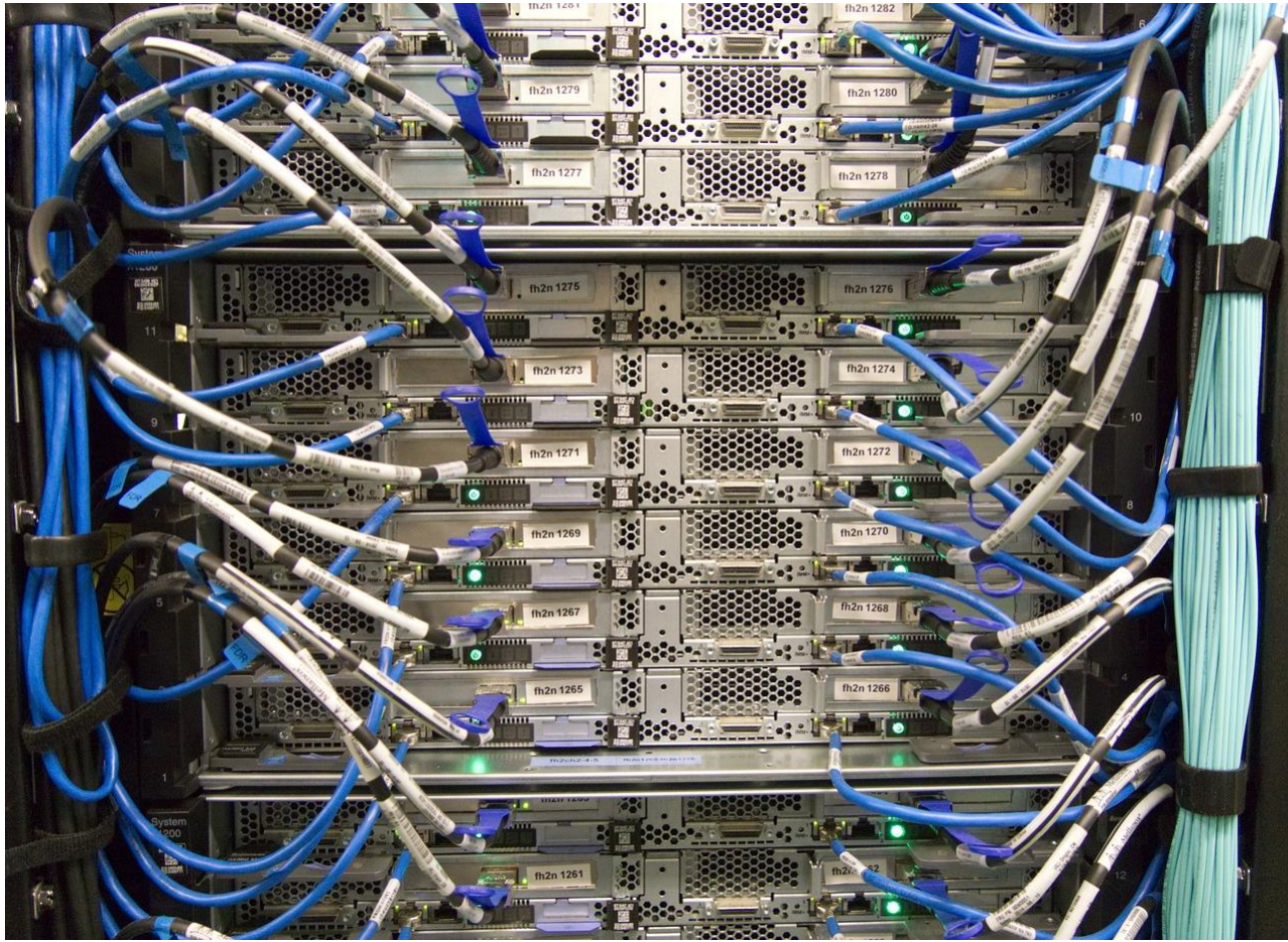
Concrete Crack Images for Classification

DOI:10.17632/5y9wdsg2zt.2

Challenge: Automation



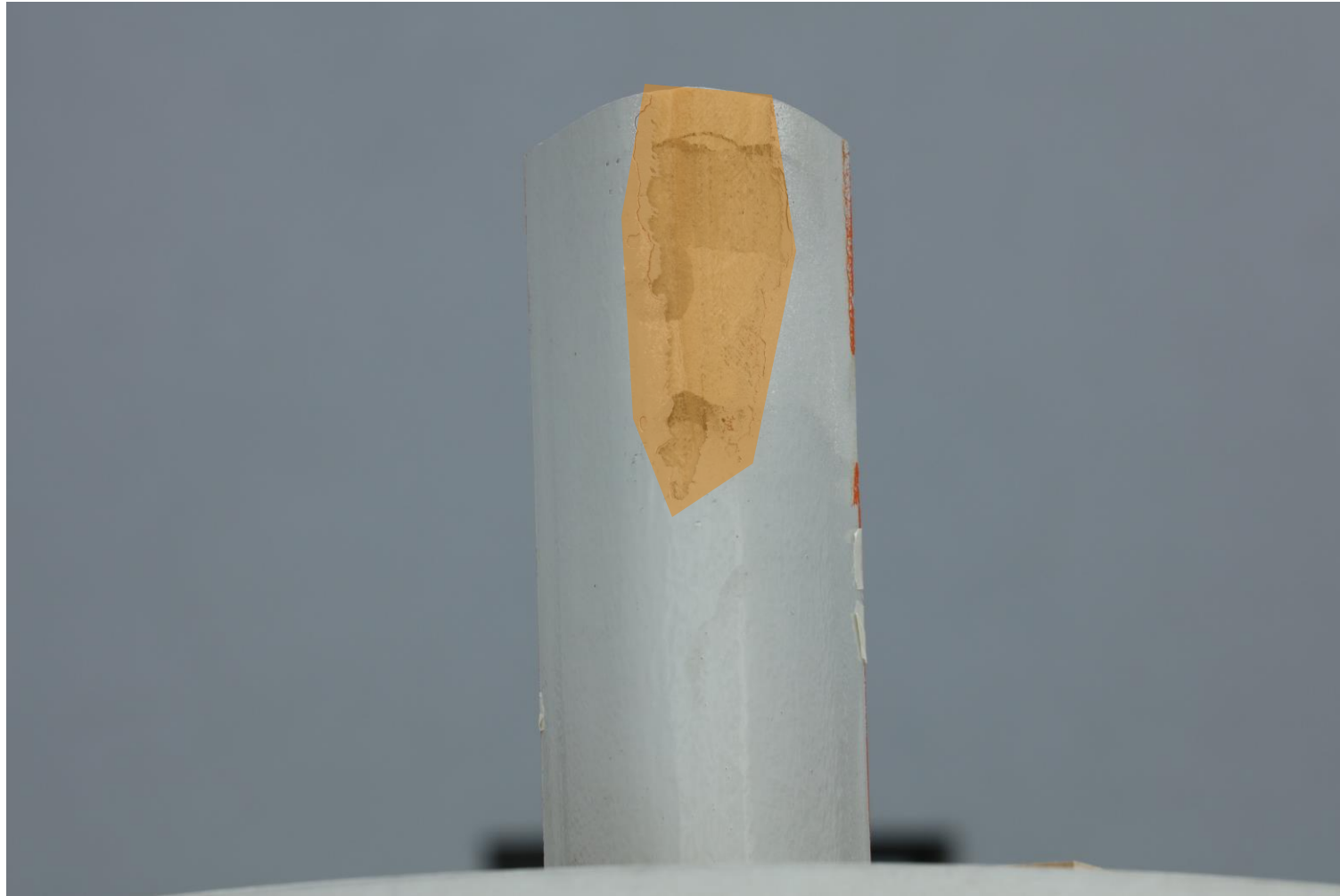
Challenge: Scale



Case study – Visual inspection for wind turbines



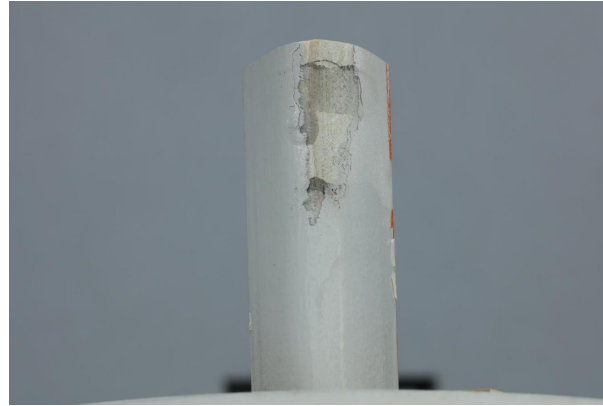
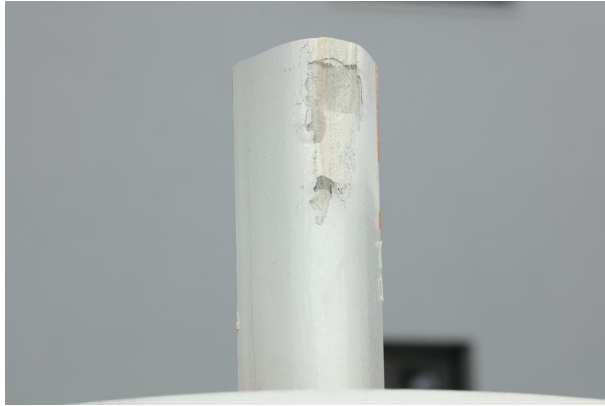
Can we identify damaged areas on wind turbine blades?



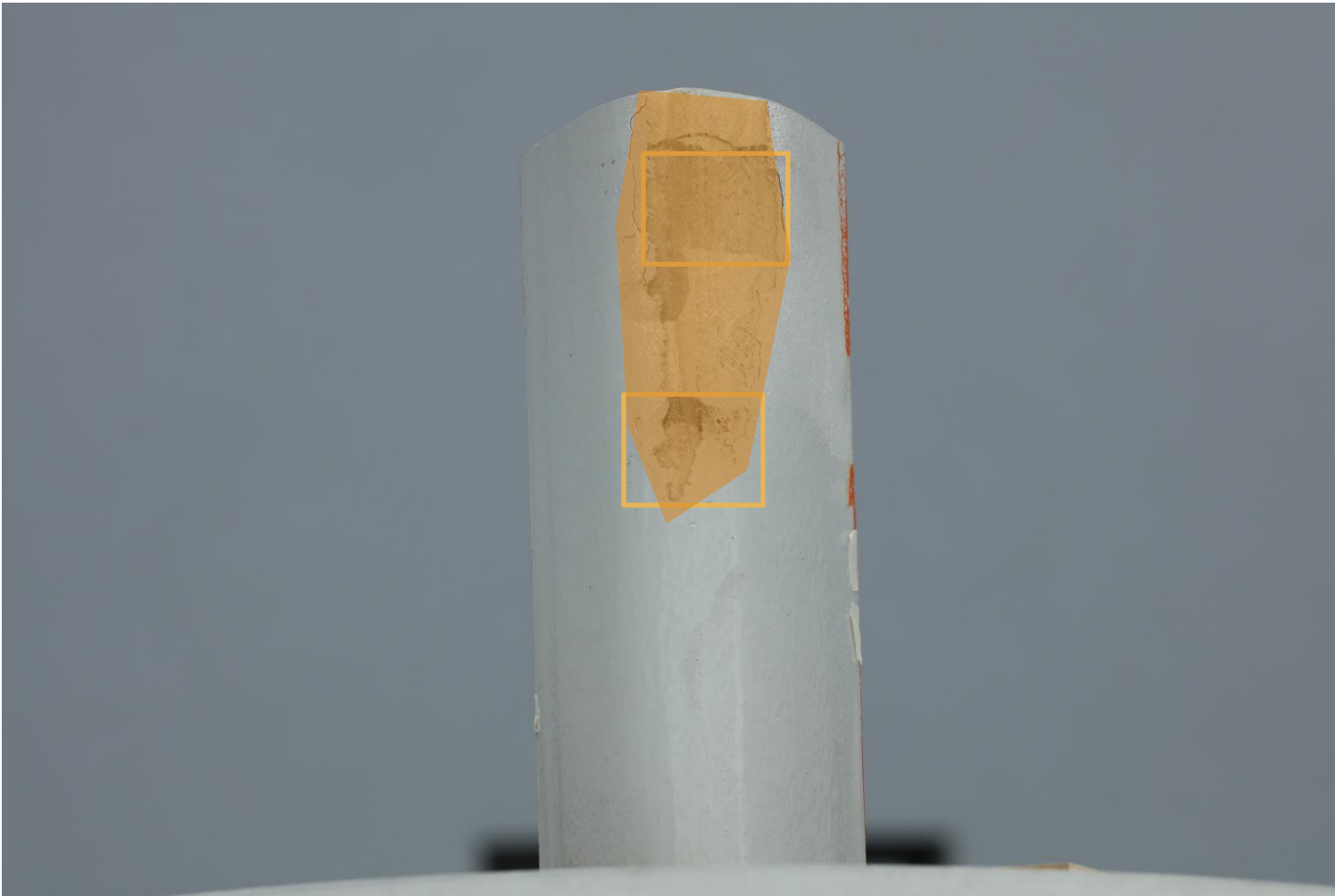
Damaged Turbine Blade

Visual inspection of wind turbine blades

Our dataset



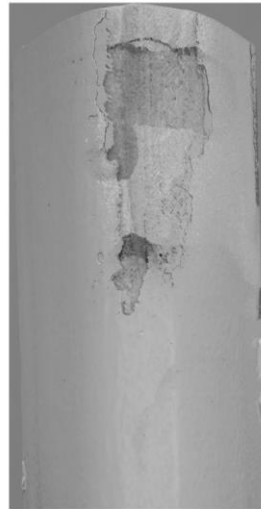
Identifying blade damage



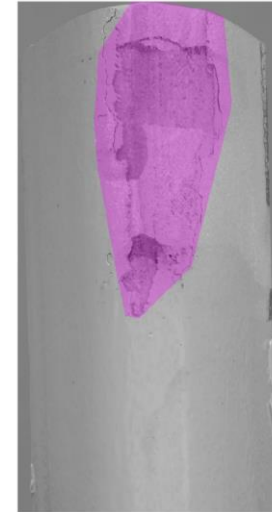
Problem Outline



Importing Data

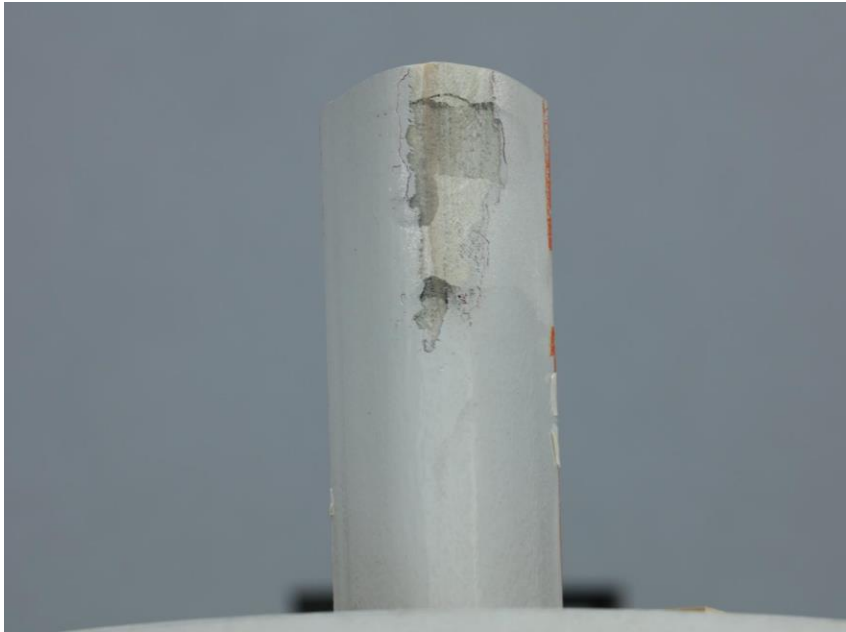


Crop to Blade



Detect Damaged Area

Importing the data



```
img = imread("2C8A0207.JPG");  
imshow(img)
```

Image read

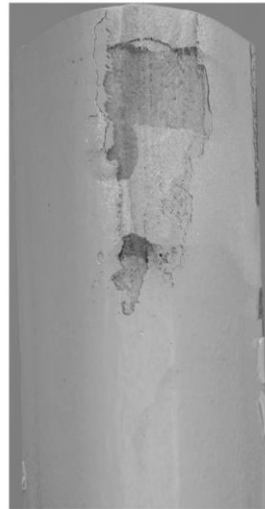


Scaling this up?

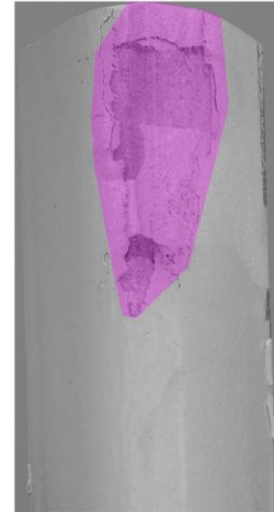
Problem Outline



Importing Data

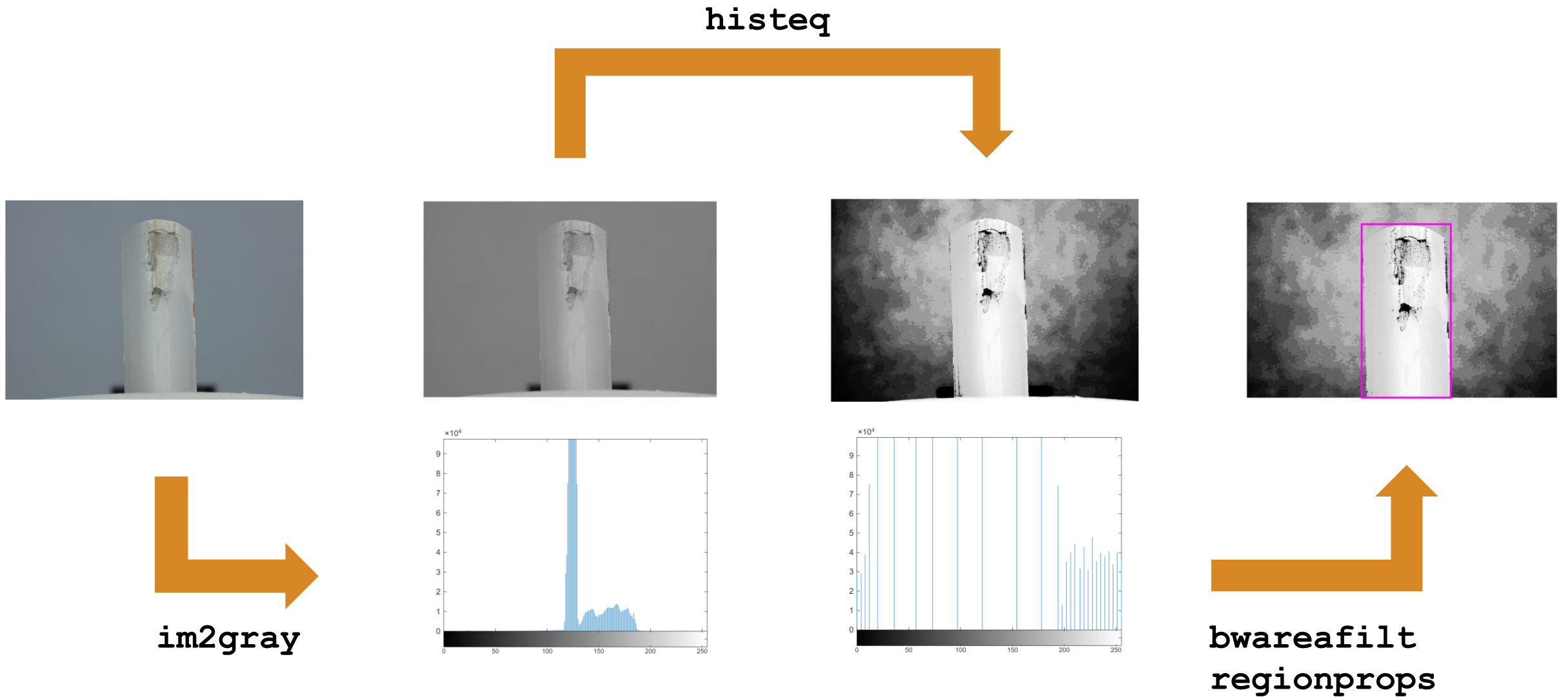


Crop to Blade



Detect Damaged Area

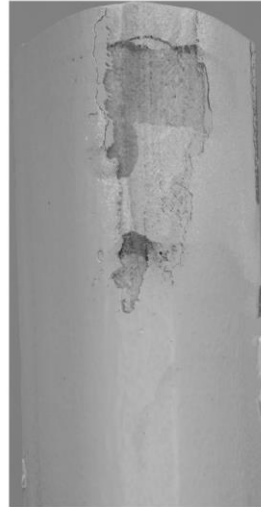
Cropping out the background



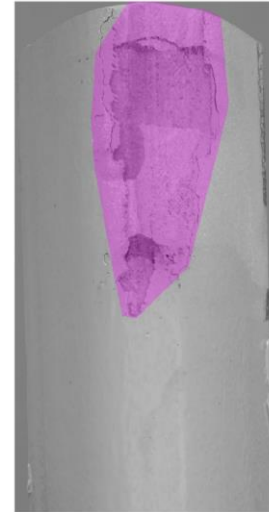
Problem Outline



Importing Data

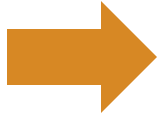
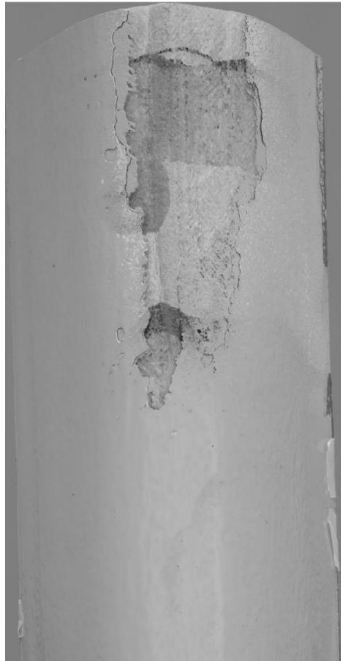


Crop to Blade

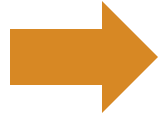


Detect Damaged Area

Detecting damage



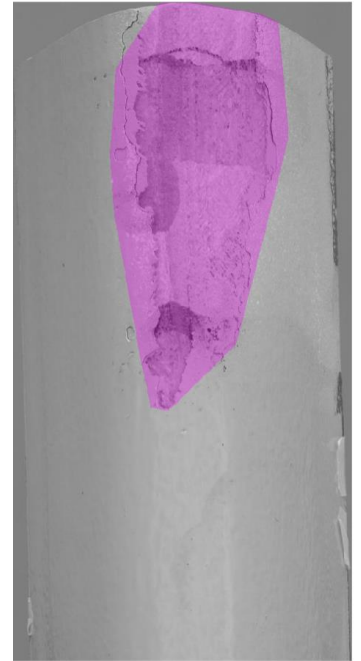
histeq



entropyfilt



prctile

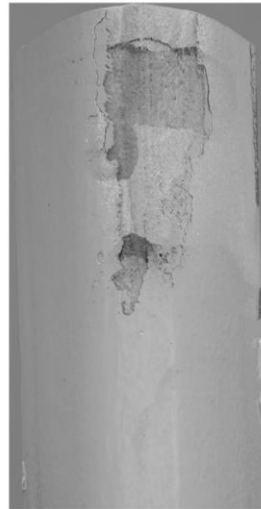


bwconvhull

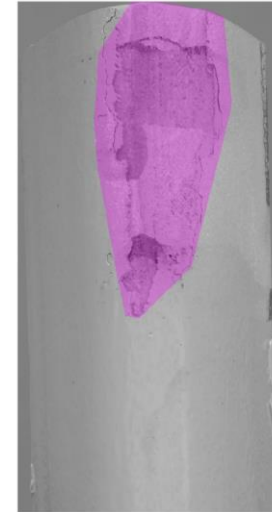
Problem Outline



Importing Data



Crop to Blade



Detect Damaged Area

Validation?

Image Batch Processor

[More about the image batch processor](#)

The screenshot displays the MATLAB Image Batch Processor interface. The window title is "Image Batch Processor". The main toolbar includes a "PROCESS" tab, a "Function Name" dropdown set to "cropBlade", and buttons for "Add", "Edit", "Open", "Use Parallel", "Process Selected", "Stop", "Link Axes", "Default Layout", and "Export". Below the toolbar, the "IMPORT" section shows a "test" folder containing a grid of image thumbnails. The thumbnail for "2C8A1549.JPG" is selected with a blue border. The main workspace is split into two panes: "Input Image" showing the full image "2C8A1549.JPG" and "croppedImage" showing the processed result. The "Results" panel at the bottom shows the "solidity" value as 0.8050.

PROCESS

Function Name: cropBlade

Buttons: Add, Edit, Open, Use Parallel, Process Selected, Stop, Link Axes, Default Layout, Export

IMPORT: test

Show All

Grid of image thumbnails (e.g., 2C8A0204, 2C8A0209, 2C8A0210, 2C8A0221, 2C8A0226, 2C8A0227, 2C8A0231, 2C8A0241, 2C8A0244, 2C8A0245, 2C8A1549, 2C8A1550, 2C8A1555, 2C8A1561, 2C8A1566, 2C8A1577, 2C8A1581, 2C8A1584, 2C8A1588, 2C8A1589, 2C8A1592, 2C8A1603, 2C8A1609, 2C8A1610, 2C8A1618, 2C8A1620, 2C8A1624, 2C8A1630)

Input Image: 2C8A1549.JPG

croppedImage

Results

solidity

0.8050

Validation

- Compute overlap between computed bounding box and ground truth
- Map detected damage convex hull back to original image co-ordinates and compute overlap with ground truth
- **This requires labeling to obtain a ground truth**

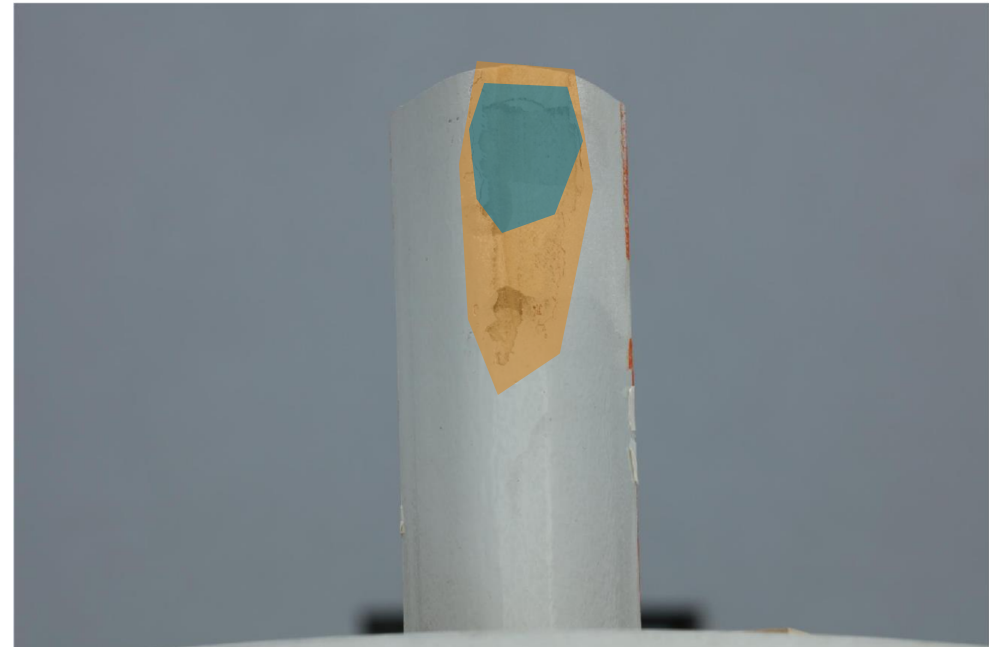
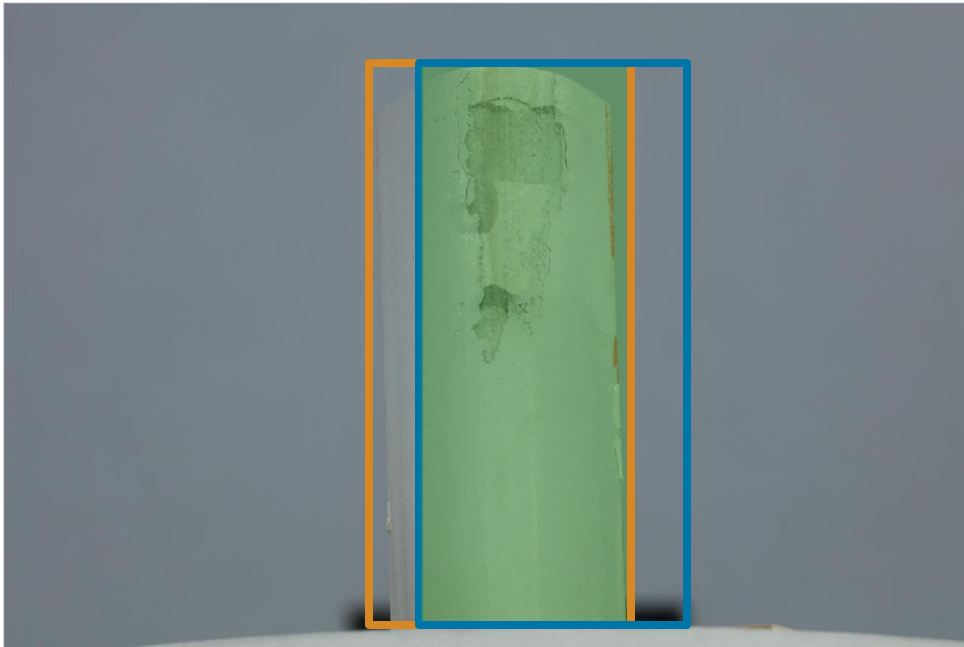


Image Labeler

[More about the image labeler](#)

The screenshot displays the MATLAB Image Labeler interface for a project named "LabelingProject.prj". The interface is divided into several sections:

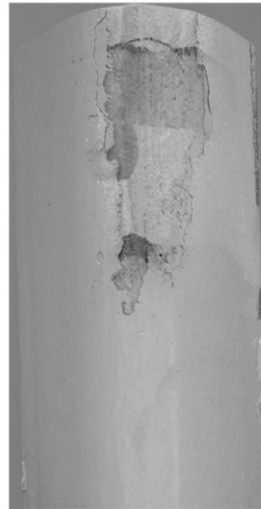
- Top Bar:** Includes "IMAGE LABELER" and "VISUALIZATION" tabs, along with navigation icons.
- Toolbox:** Contains various actions such as "New Project", "Open Project", "Save Project", "Import", "Add Label", "Sublabel", "Edit", "Delete", "Attribute", "Algorithm", "Automate", "View Label Summary", "Keyboard Shortcuts", "Tutorials", and "Export".
- ROI Label Definitions:** A list of labels for the current image. "blade" is selected, and "damage" is also visible.
- Scene Label Definitions:** A section for defining scene labels, with a note: "To label a scene, you must first define a scene label." and buttons for "Apply to Image" and "Remove from Image".
- Image Viewer:** The central area showing a photograph of a white cylindrical object with a damaged section. A green bounding box is drawn around the object, and a blue polygonal mask is applied to the damaged area.
- View Labels, Sublabels and Attributes:** A panel on the right showing a list of labels: "blade" and "damage".
- Image Browser:** A strip at the bottom showing a sequence of images from the project.

Problem Outline

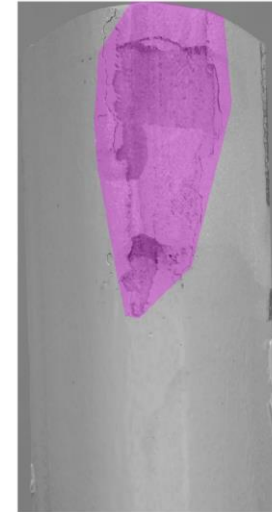
Can we do better using deep learning?



Read Data



Crop to Blade



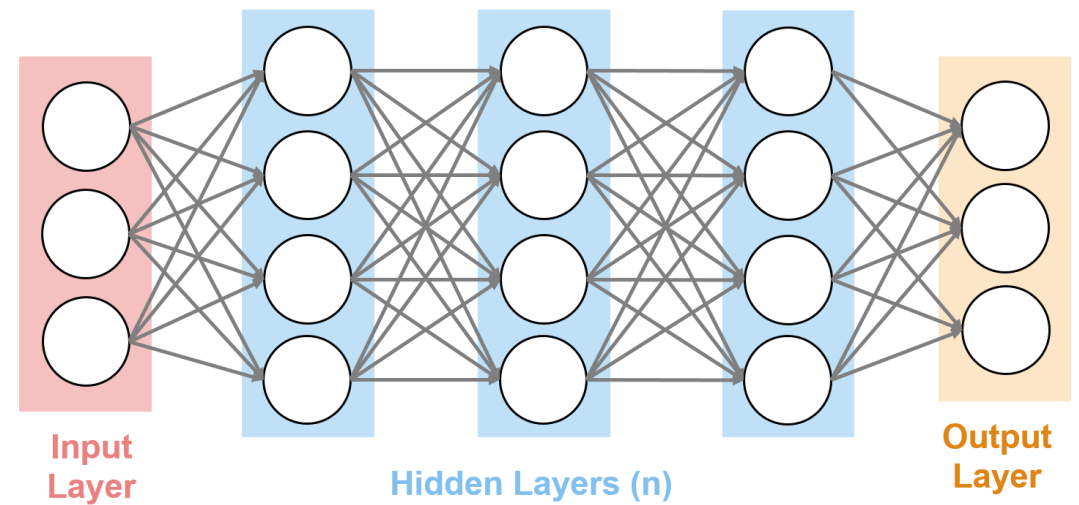
Detect Damaged Area

Deep learning workflow

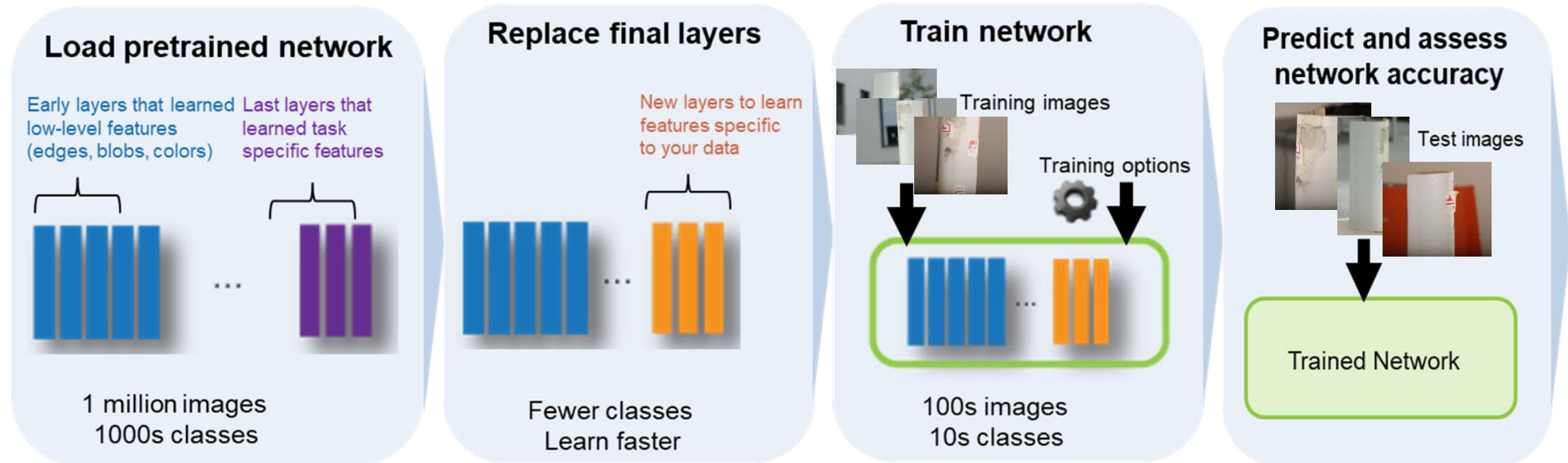


Neural Network = Model

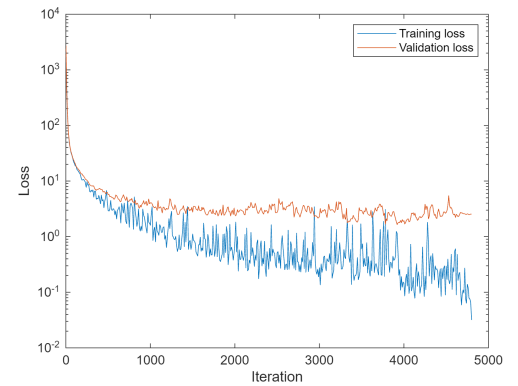
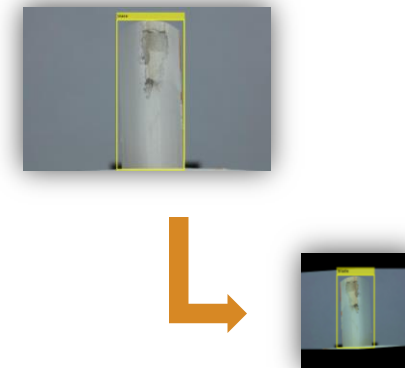
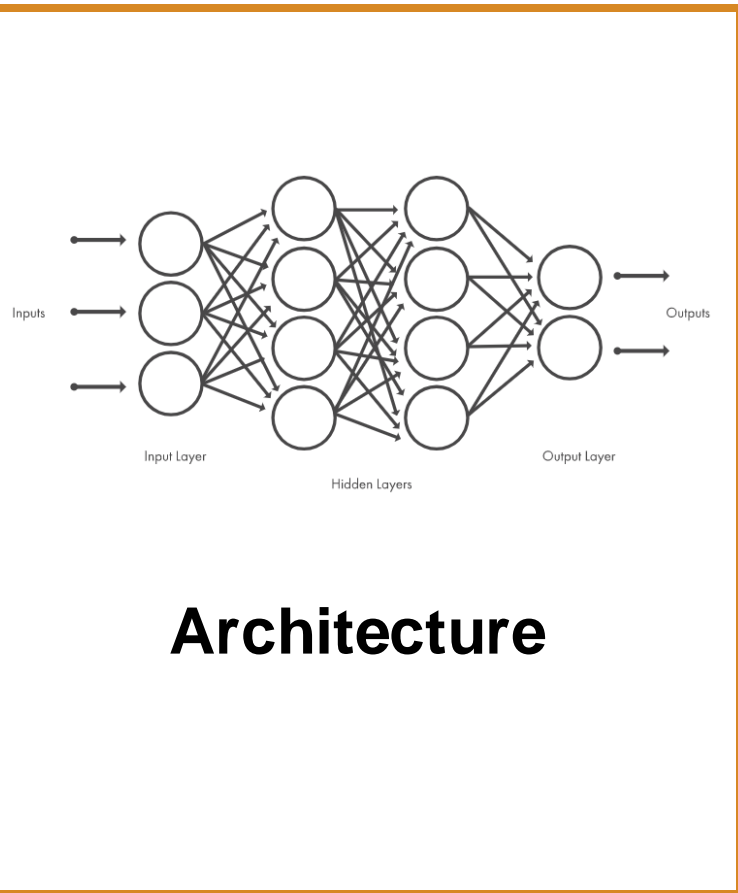
- Designing the architecture
- Training and validating the model
- Tuning training options



Transfer learning



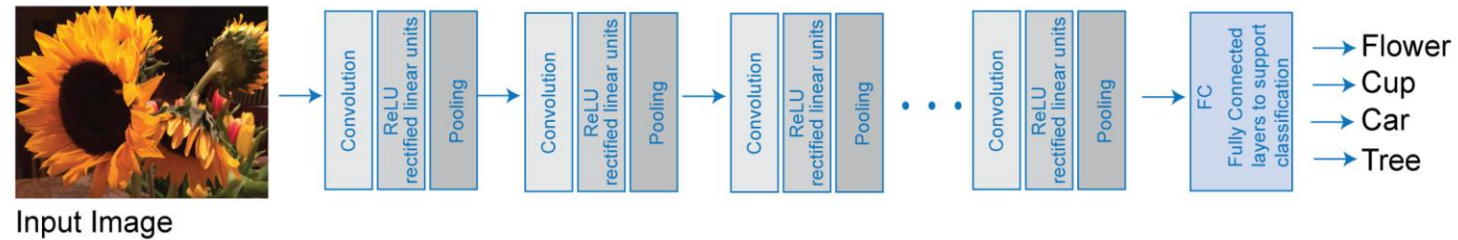
Solving the cropping sub-problem



Architecture

Classification

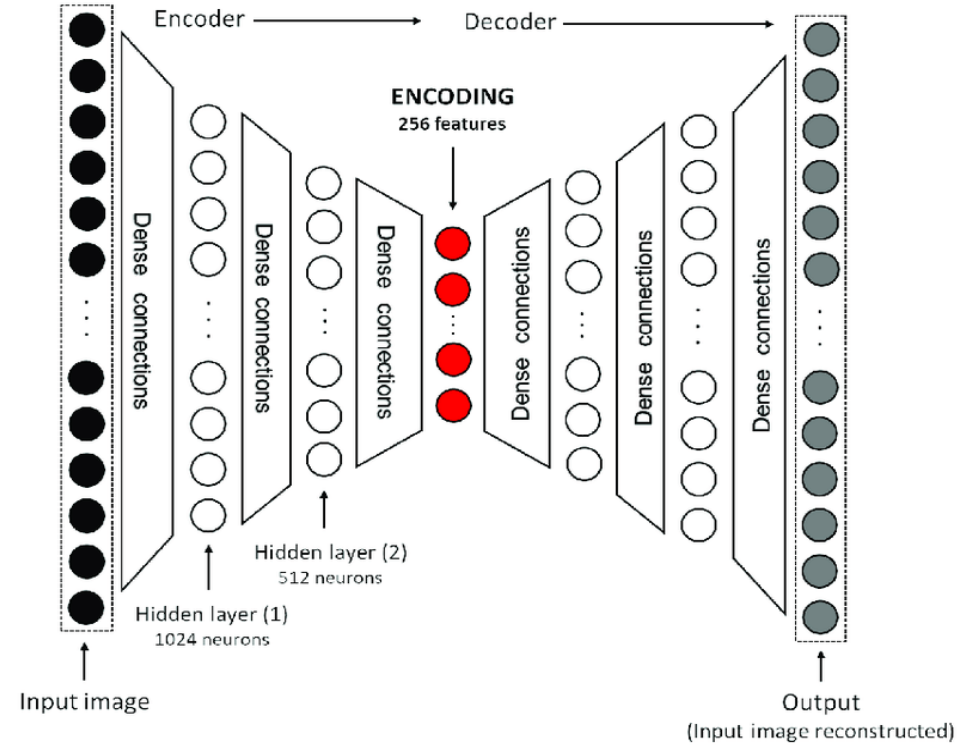
Convolutional Neural Networks



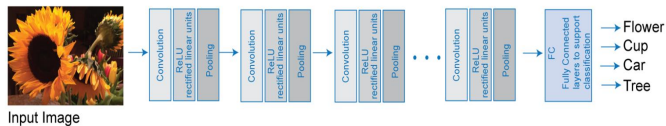
Architecture

Anomaly Detection

Deep Autoencoder



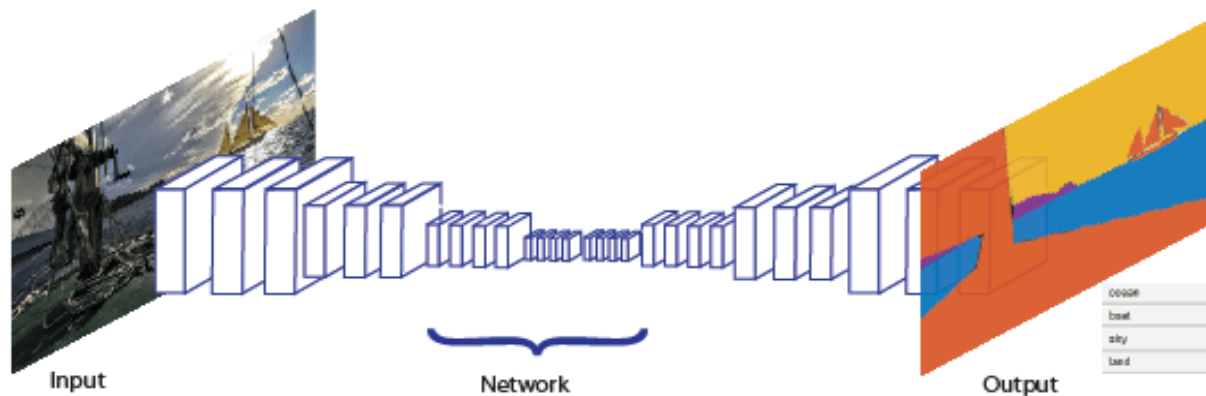
Convolutional Neural Networks



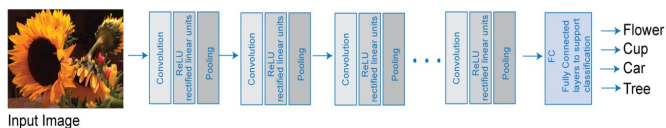
Architecture

Image Segmentation

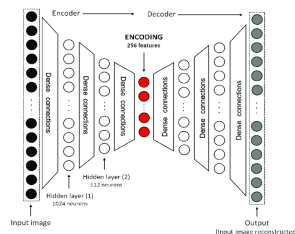
Semantic Segmentation Network



Convolutional Neural Networks



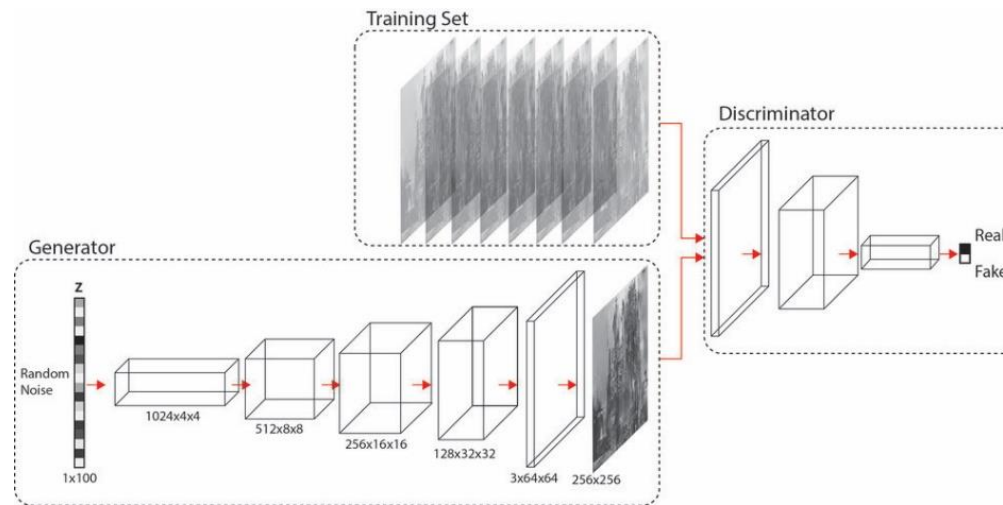
Deep Autoencoder



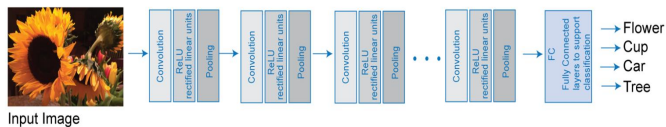
Architecture

Denoising, Synthetic data generation

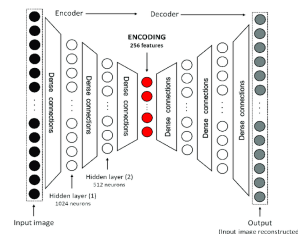
Generative Adversarial Network (GAN)



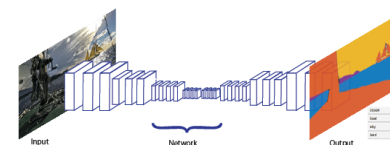
Convolutional Neural Networks



Deep Autoencoder



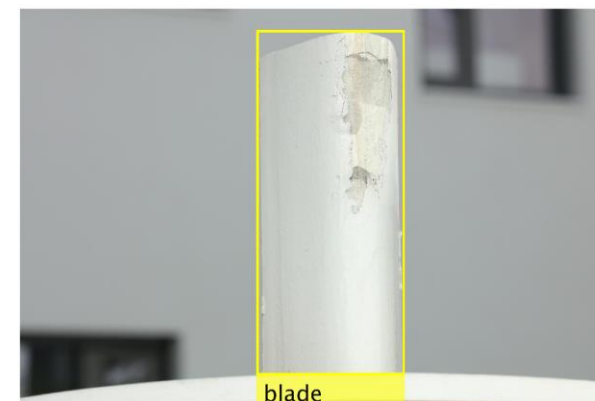
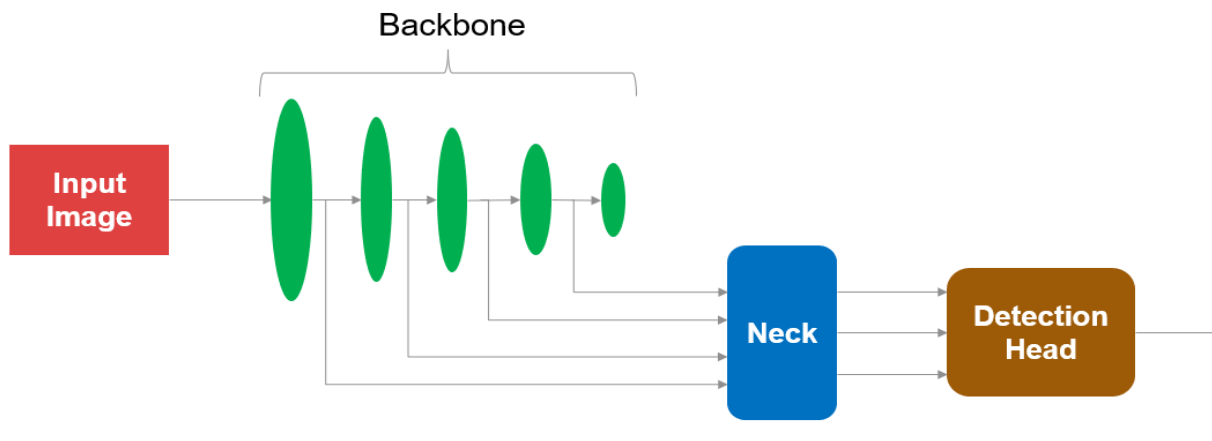
Semantic Segmentation Network



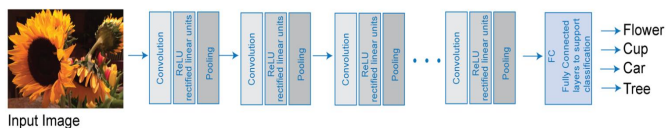
Architecture

Object Detection

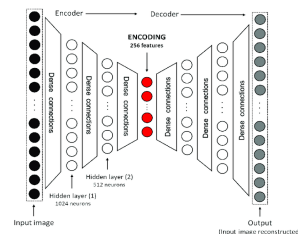
YOLO – You Only Look Once



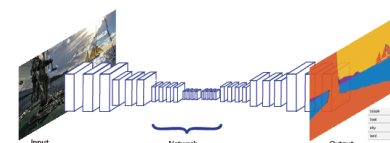
Convolutional Neural Networks



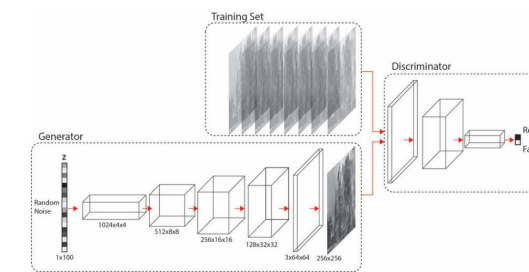
Deep Autoencoder



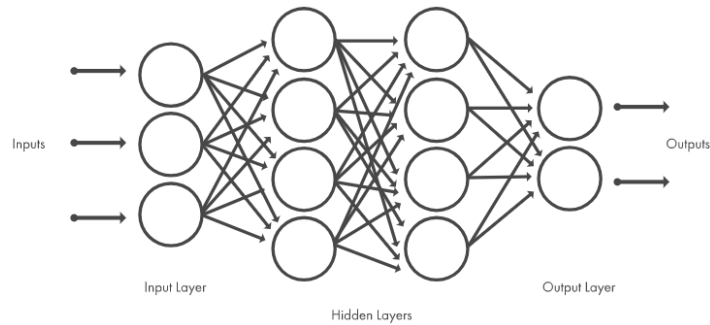
Semantic Segmentation Network



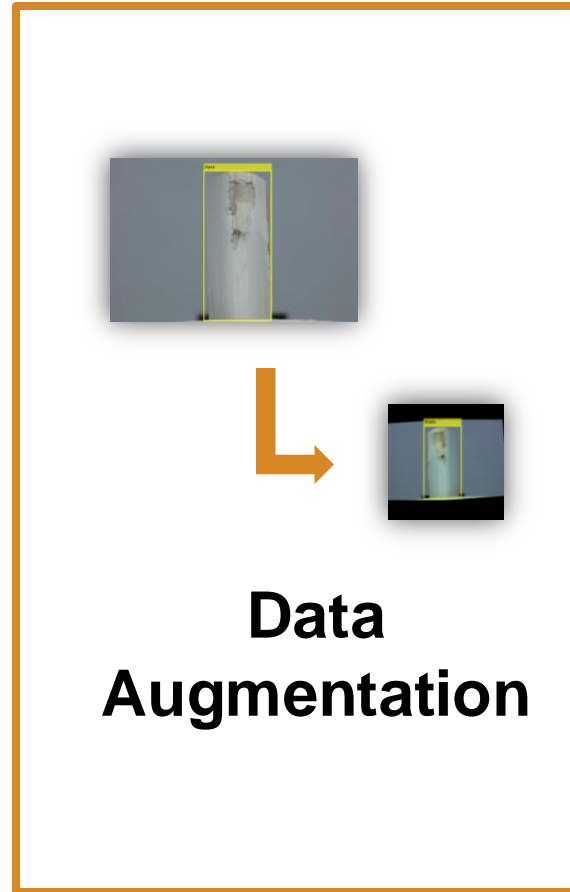
Generative Adversarial Network (GAN)



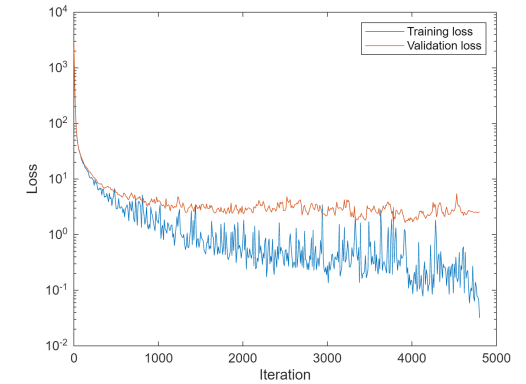
Solving the cropping sub-problem



Architecture

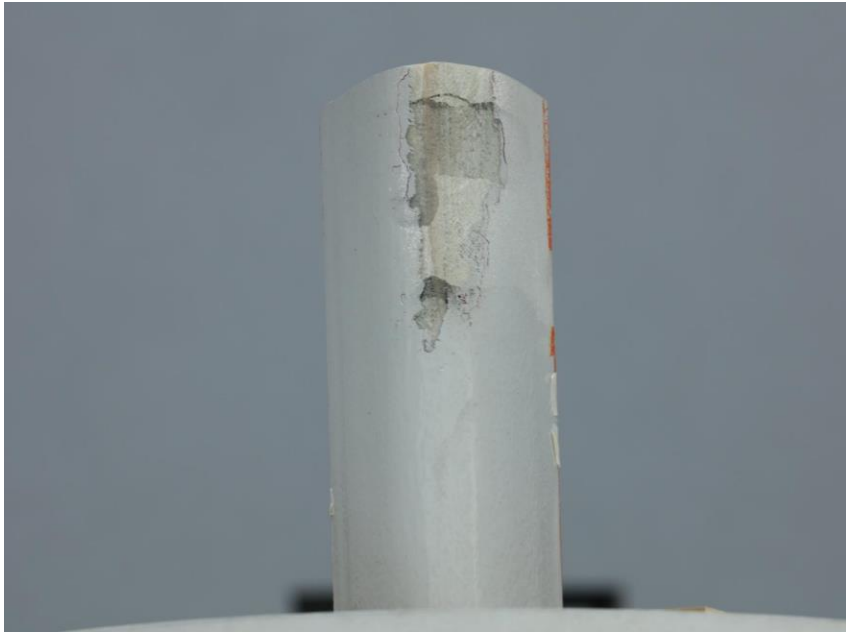


**Data
Augmentation**



Train & Evaluate

Importing the data



```
img = imread("2C8A0207.JPG");  
imshow(img)
```

Image read



Scaling this up?

Importing the data and data augmentation



```
f = figure();
img = imread("2C8A0207.JPG");
imshow(img)
```

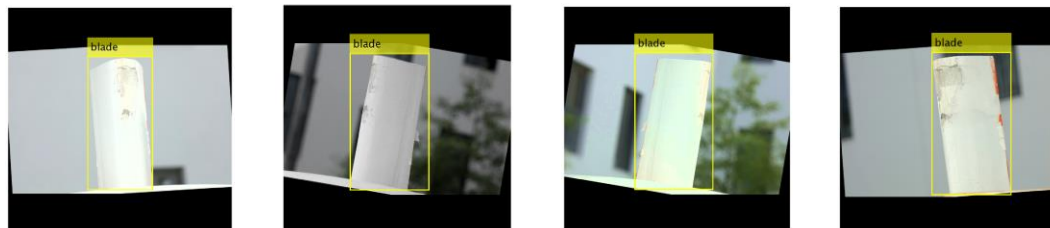
Image read

Datastore



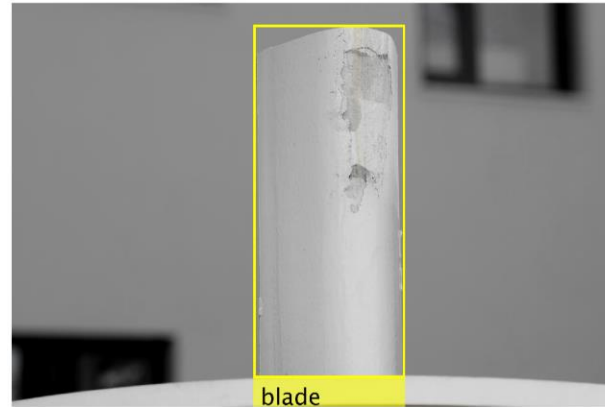
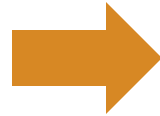
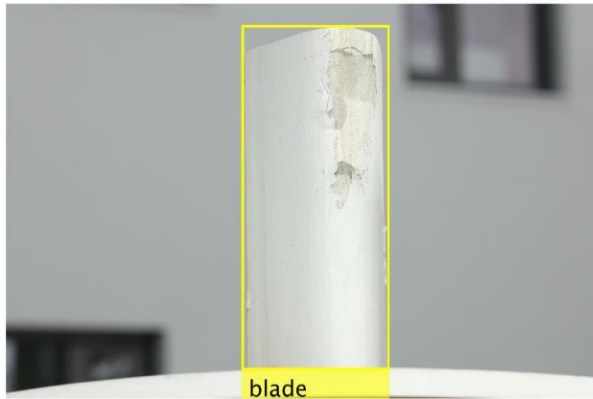
Hard Drive

```
data = trainDS.read();
```

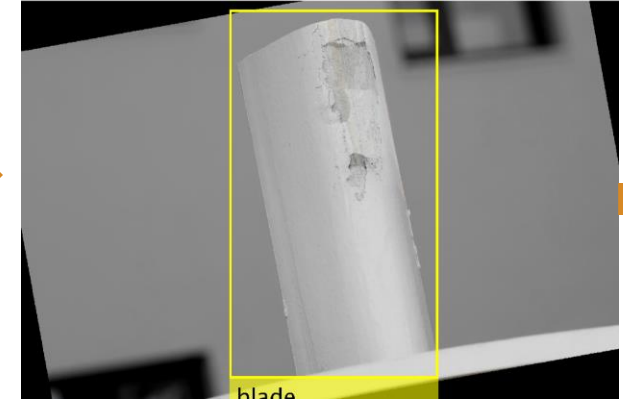


Datastores

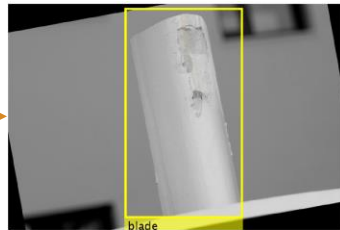
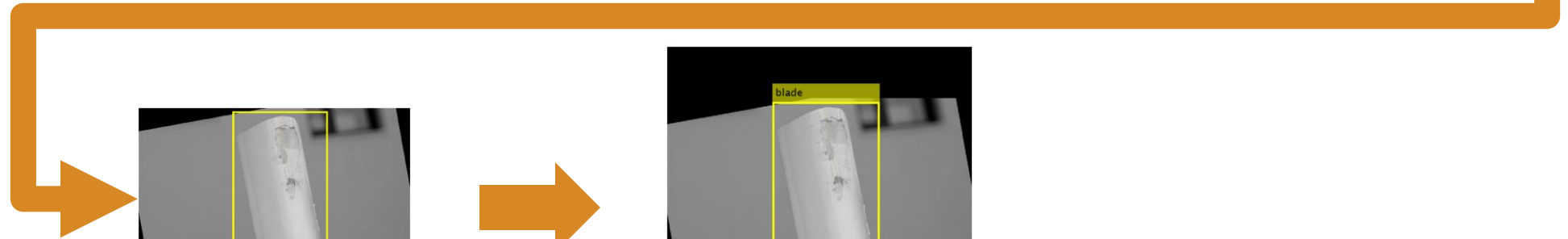
Data Augmentation



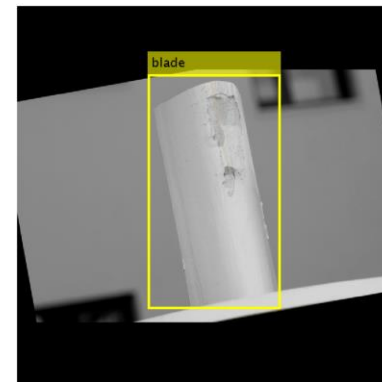
`jitterColorHSV`



`randomAffine2d`
`+ imwarp`

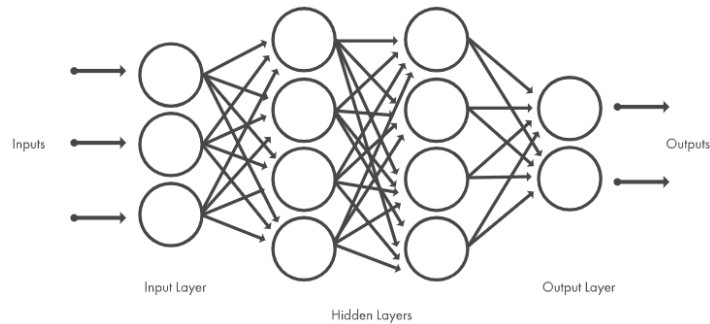


`imresize`

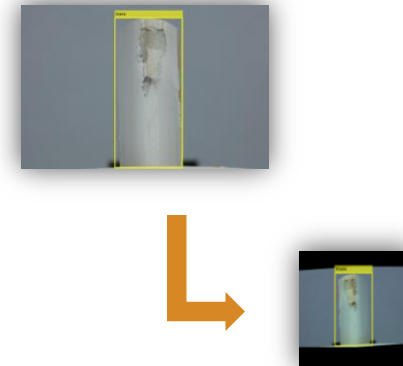


`padarray`

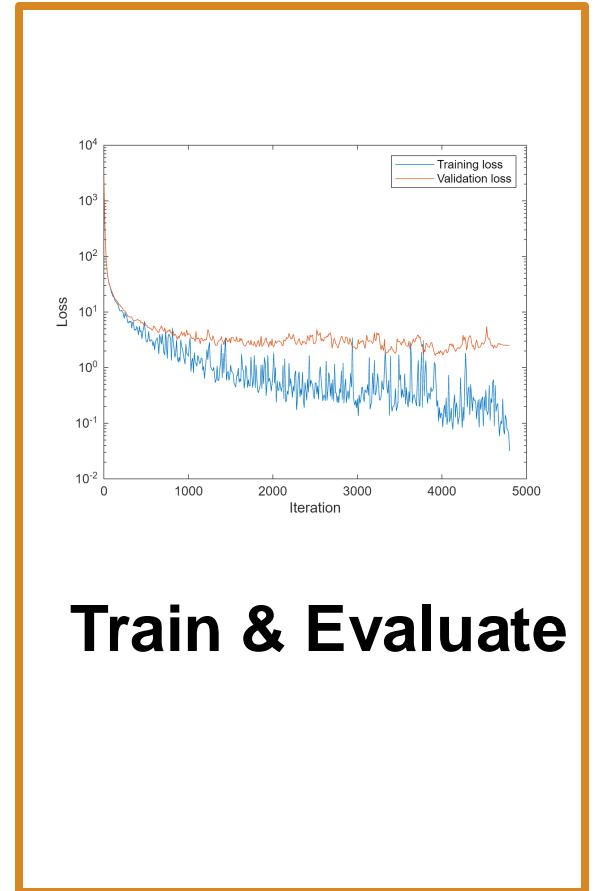
Solving the cropping sub-problem



Architecture



**Data
Augmentation**



Train & Evaluate

Training and evaluation

43

```
detector = yolov4ObjectDetector("tiny-yolov4-coco", "blade", anchorBoxes, InputSize=inputSize);
```

44

```
options = trainingOptions("adam", ...
```

45

```
    InitialLearnRate=0.001, ...
```

46

```
    MiniBatchSize=4, ...
```

47

```
    MaxEpochs=200, ...
```

48

```
    Shuffle="every-epoch", ...
```

49

```
    ValidationData=valDS);
```

50

```
detectorTrain = trainYOLOv4ObjectDetector(preprocessedTrainDS, detector, options);
```

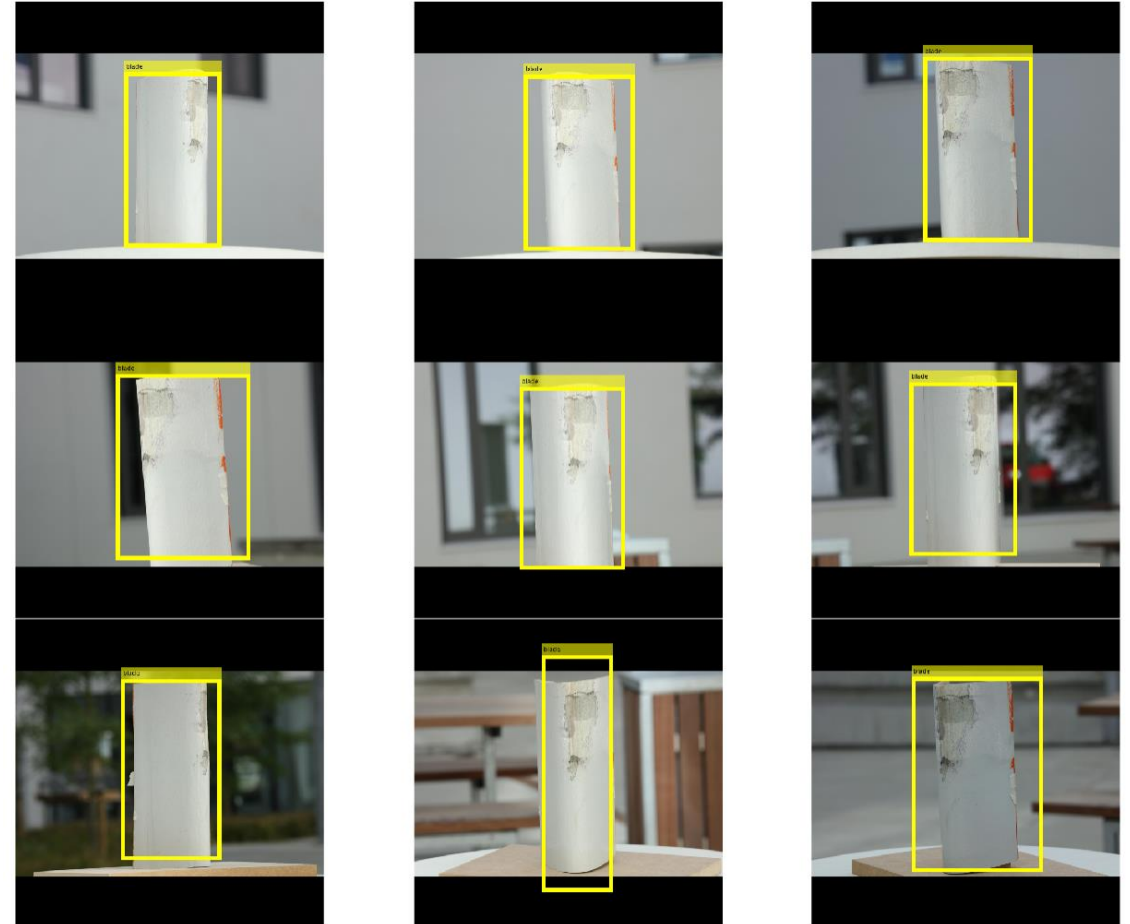
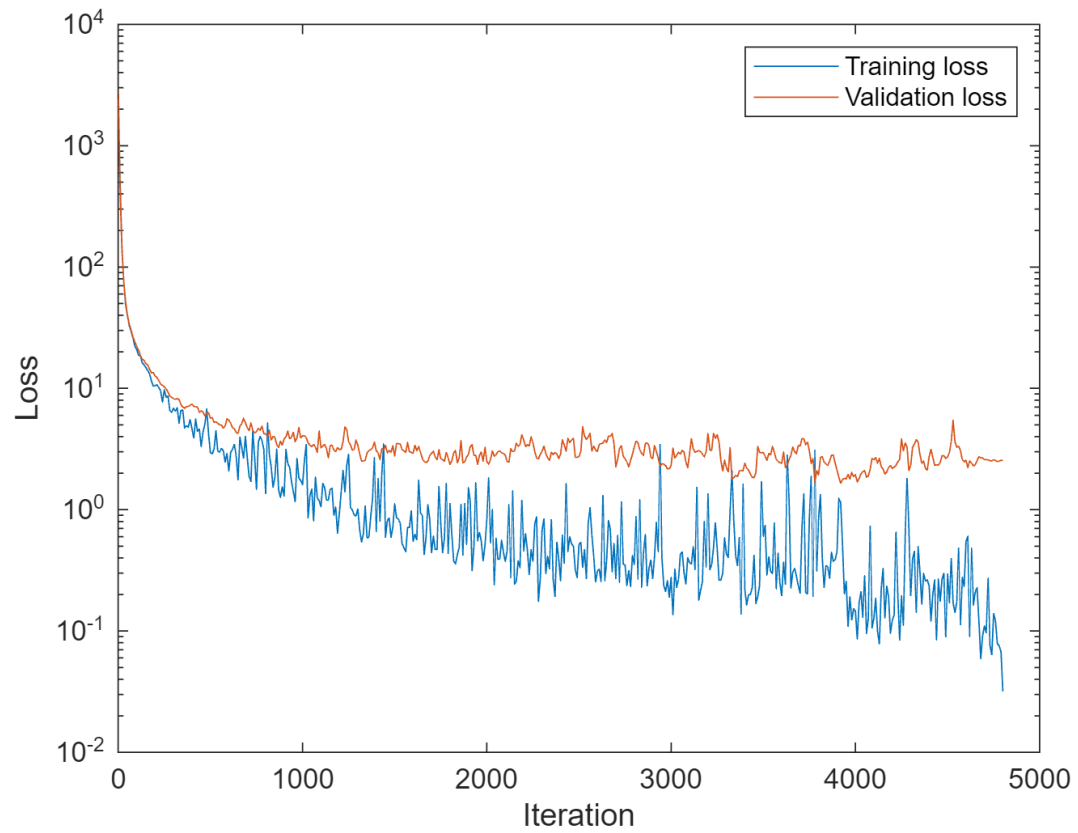
Computing Input Normalization Statistics.

Training a YOLO v4 Object Detector for the following object classes:

* blade

Epoch	Iteration	TimeElapsed	LearnRate	TrainingLoss	ValidationLoss
1	1	00:00:11	0.001	2407.7	2746
1	10	00:00:44	0.001	504.49	432.31
1	20	00:01:18	0.001	144.33	138.46
2	30	00:01:52	0.001	76.456	75.908
2	40	00:02:27	0.001	48.541	51.499
3	50	00:02:59	0.001	39.563	40.005
3	60	00:03:32	0.001	32.804	33.864
3	70	00:04:06	0.001	28.351	30.126

Training and evaluation



Take-Home Messages

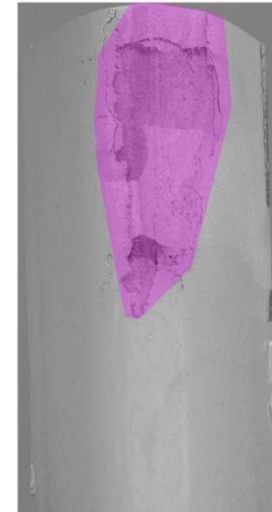
Control your data gathering environment



Importing Data



Crop to Blade



Detect Damaged Area

Classical image processing and deep learning can be complementary

	Classical Image Processing	Deep Learning
Data Requirements	Low	High
Data Labelling	Not required	Algorithm / workflow dependent <i>Time consuming</i>
Algorithm Development	Potentially complex	Learnt from data
Anomalies	Must account for manually <i>Likely to be missed once deployed</i>	Learnt from data <i>Depends on algorithm</i>
Explainability	Complete	Low by default <i>Tools exist to improve</i>
Execution Speed	High	Slow <i>Specialized hardware can improve</i>

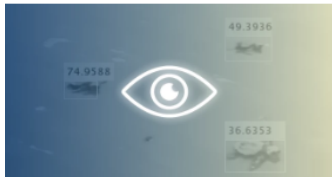
Onramps - Learn the basics in 2 hours or less



Image Processing Onramp

2 hours | [Languages](#)

Learn the basics of practical image processing techniques in MATLAB.



Computer Vision Onramp

1.5 hours | [Languages](#)

Learn the basics of computer vision to design an object detector and tracker.



Machine Learning Onramp

2 hours | [Languages](#)

Learn the basics of practical machine learning methods for classification problems.



Deep Learning Onramp

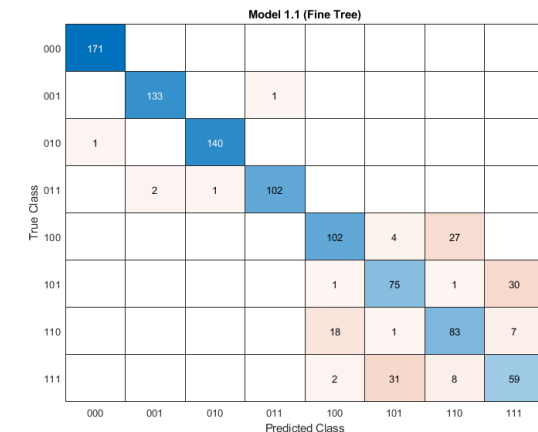
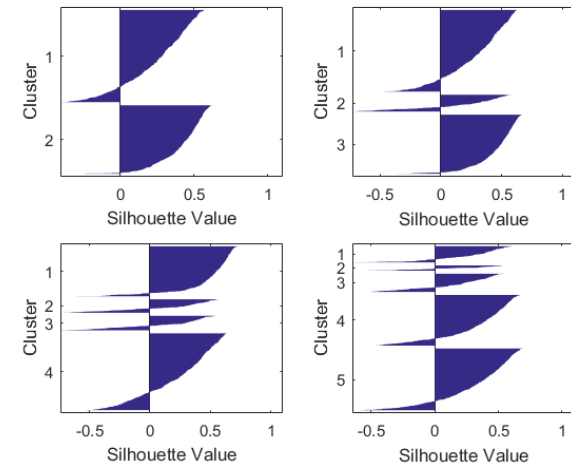
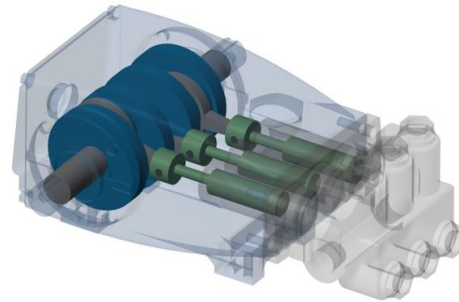
2 hours | [Languages](#)

Get started quickly using deep learning methods to perform image recognition.

Training Course: Predictive Maintenance with MATLAB

Topics included in this 2-day course:

- Importing and organizing data
- Creating custom visualizations
- Fault Detection/Classification
- Preprocessing to improve data quality, and extract time and frequency domain features
- Estimating Remaining Useful Life (RUL)
- Interactive workflows with apps



[See detailed course outline](#)



Achieve Results Faster with Predictive Maintenance Consulting

Our expert consultants can help you with the entire predictive maintenance workflow: Data Preprocessing, Exploratory Analysis, Predictive Modeling, and Operational Deployment



Transparent Approach

You will have full access to all our work throughout your project. Your self-sufficiency is our goal.



Customized Engagements

We'll work with you on a customized project plan aligned to your business goals.



Return on Investment

Reduce development time and cost, learn faster, and improve quality and collaboration.

Request a free consultation: www.mathworks.com/pmp

MATLAB EXPO

 UNITED KINGDOM



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

