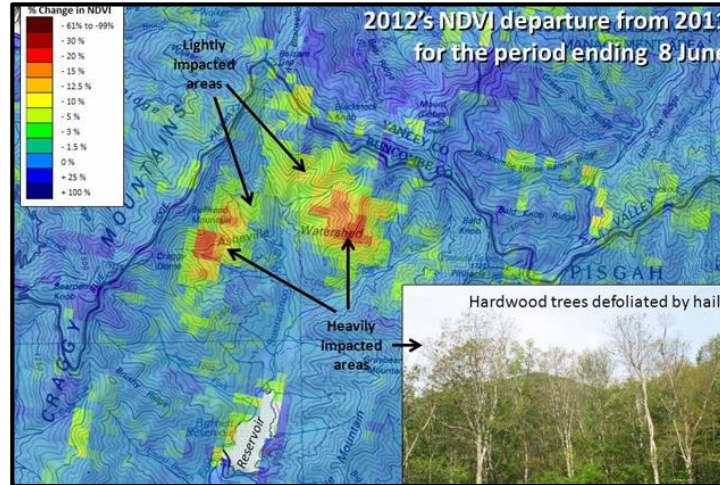


Image Processing and Computer Vision

IPCV Applications



NASA Develops Warning System for Detecting Forest Disturbances



Beth Israel Medical Center Improves MRI Accuracy



CNH Develops Intelligent Filling System for Forage Harvesters



FLIR Accelerates Development of Thermal Imaging FPGA



Veoneer (Autoliv) Builds Radar Sensor using LiDAR-Based Verification

Image Processing & Computer Vision

Import, Display, and
Exploration

Geometric Transform and
Image Registration

Image Filtering and
Enhancement

Image Segmentation and
Analysis

3D Volumetric Processing

Camera Calibration and
3D-Vision

Tracking and Motion
Estimation

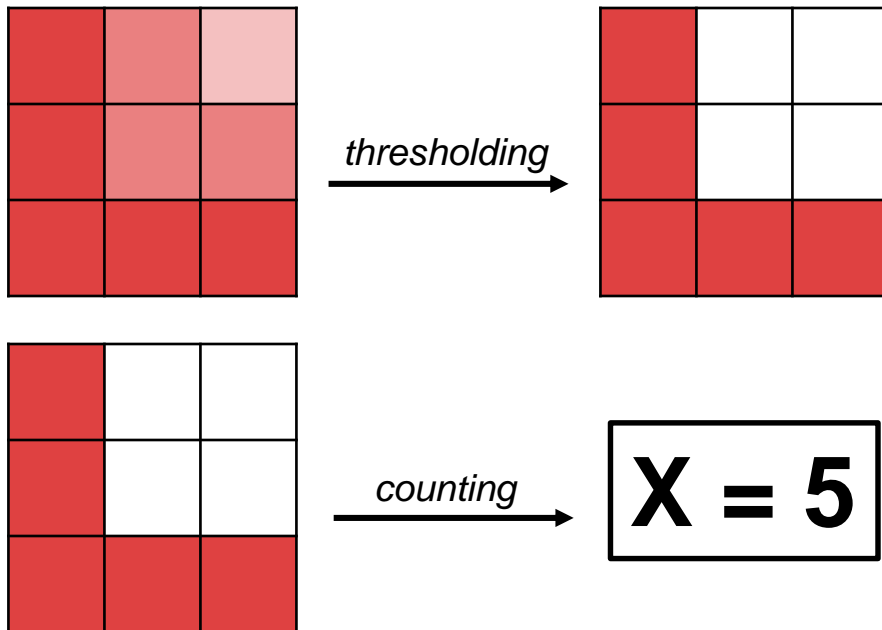
Feature Detection and
Extraction

LiDAR and Point Cloud
Processing

Deep Learning

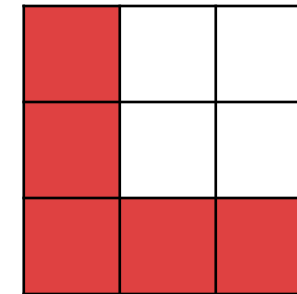
Image Processing

Manipulation of images to extract meaningful information



Computer Vision

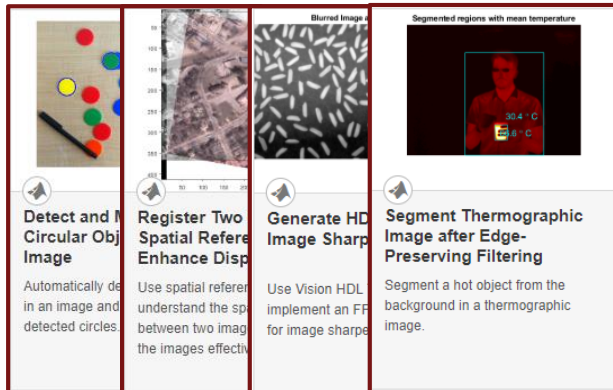
Teaching computers to understand images and video



This is the letter "L"

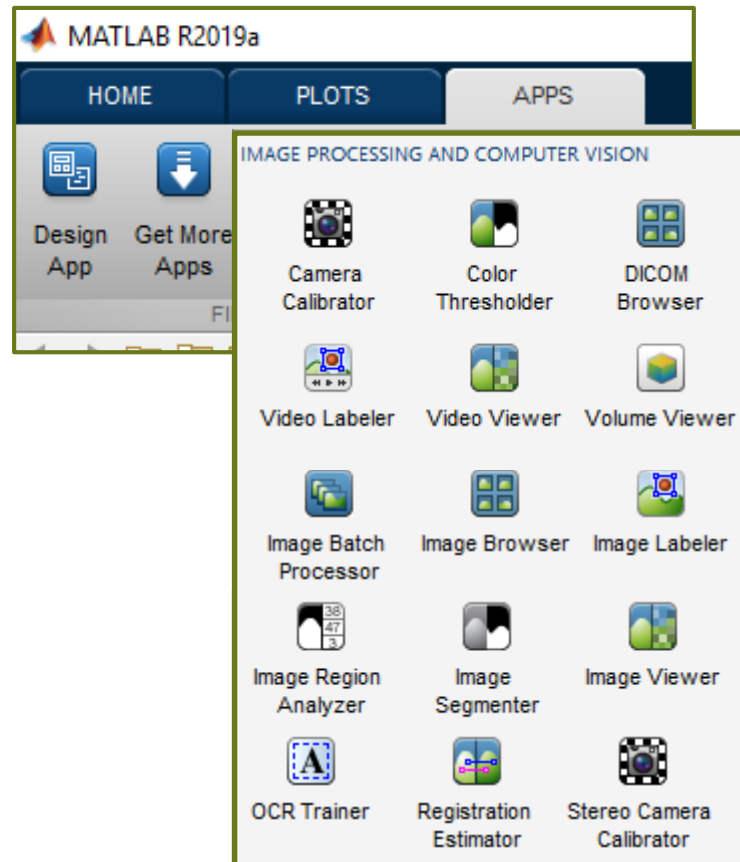
Why Use MATLAB?

Ease of Use and Thorough Documentation



(...)

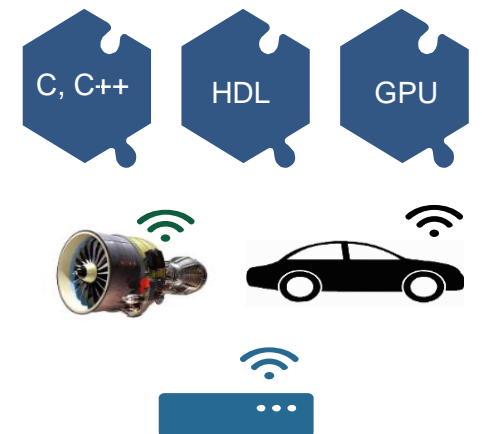
Rapid Prototyping and Algorithm Development



Code Generation for Embedded Deployment

MATLAB Code

Embedded Hardware



Need Technical Help?

- Technical Support
- Application Engineers

Agenda

Image Processing Workflow

Morphology and Segmentation

Object Detection and Tracking

Deep Learning

Summary + Next Steps

Agenda

Image Processing Workflow

Morphology and Segmentation

Object Detection and Tracking

Deep Learning

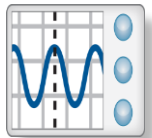
Summary + Next Steps

Image Processing Workflow

Access Data



Files

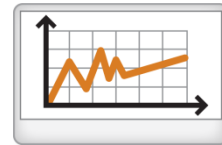


Hardware



Software

Explore and Discover



Data Analysis and Modeling

```
for k=1:max
    x = fft(dat
    y = 20*log1
```

Algorithm Development



Application Creation

Share Results



Embedded Code



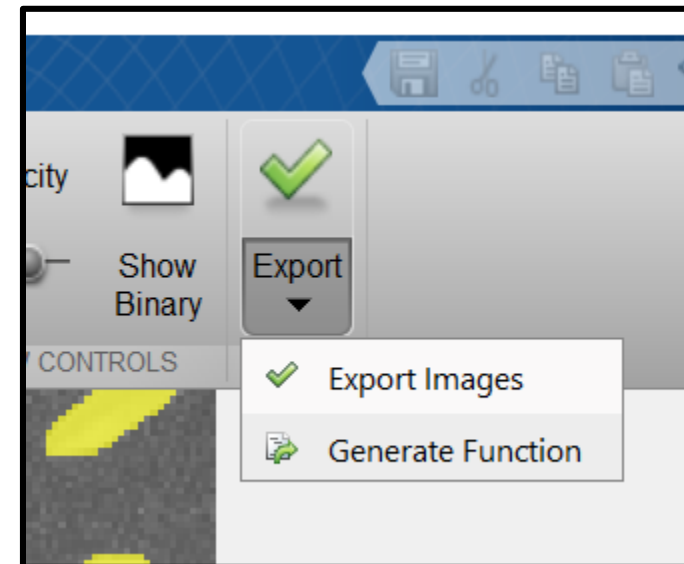
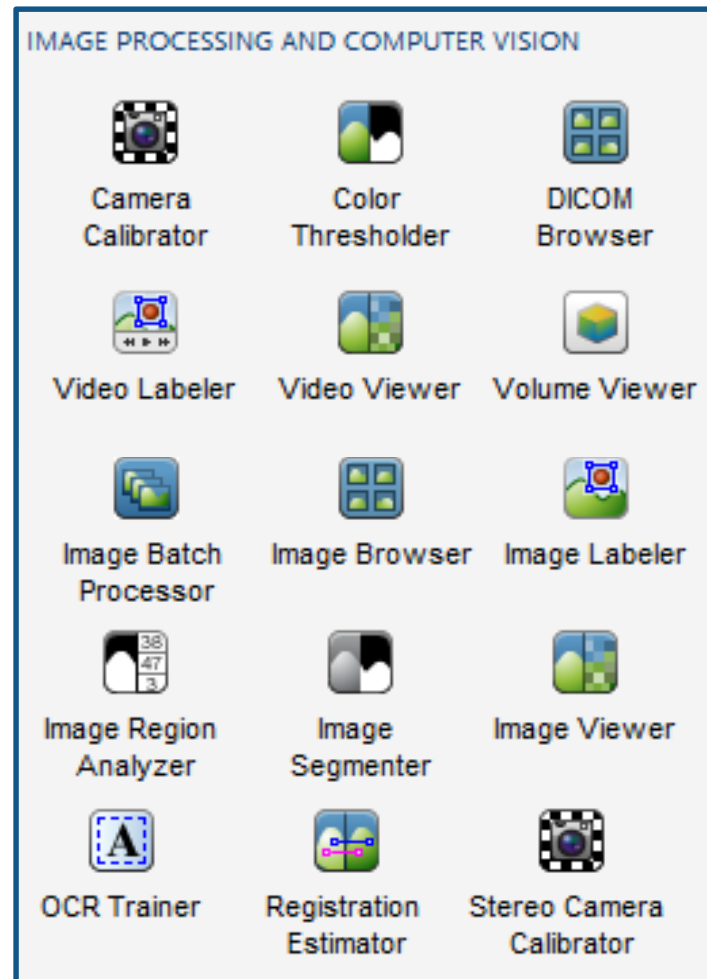
Enterprise Application



Reports

Iterate and Automate

IPCV Apps Accelerate Workflow



Instructor Demo

IPCVParasitologyWorkflow.mlx

Takeaways

- Data management with **imageDatastore**
- **IPCV apps** perform a variety of processes and can generate MATLAB Code
 - *(Additional Apps to generate embedded code)*

Agenda

Image Processing Workflow

Morphology and Segmentation

Object Detection and Tracking

Deep Learning

Summary + Next Steps

Morphology and Segmentation

Image processing techniques that facilitate the extraction of information



Original

Image Morphology

Structuring Element: 3-pixel line

*Rule: **Dilation** (max)*

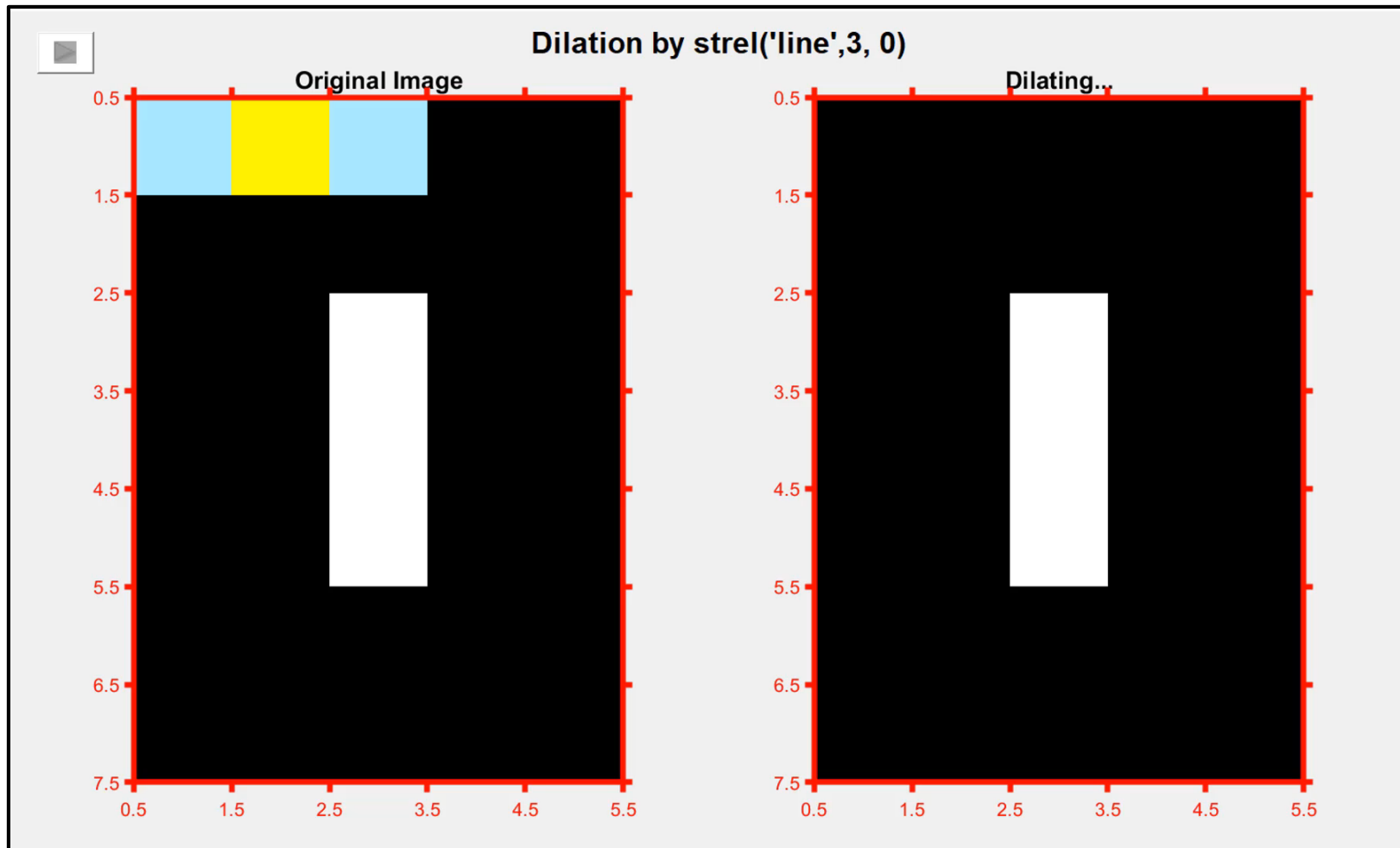


Image Morphology

Structuring Element: 3-pixel line

*Rule: **Erosion** (min)*

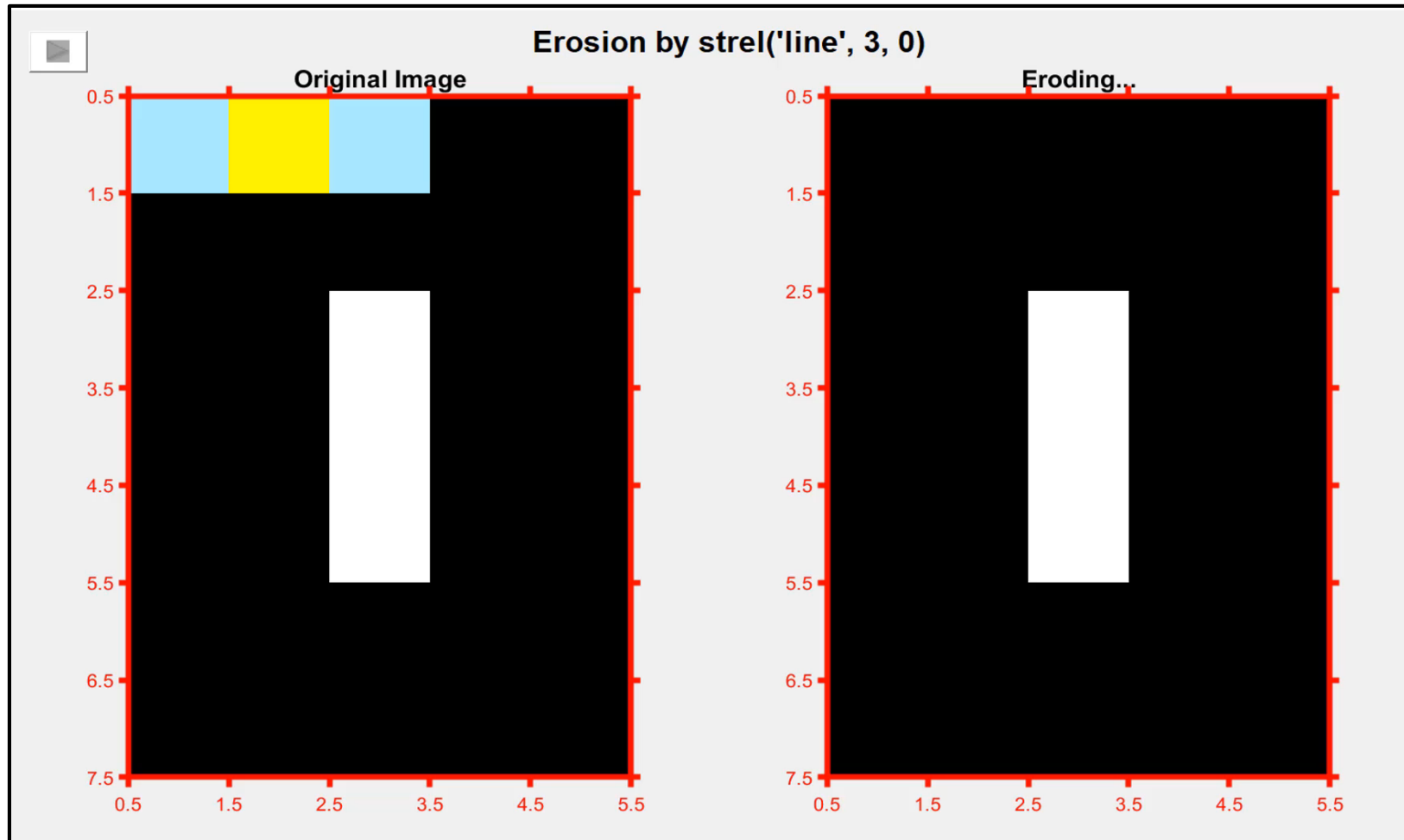
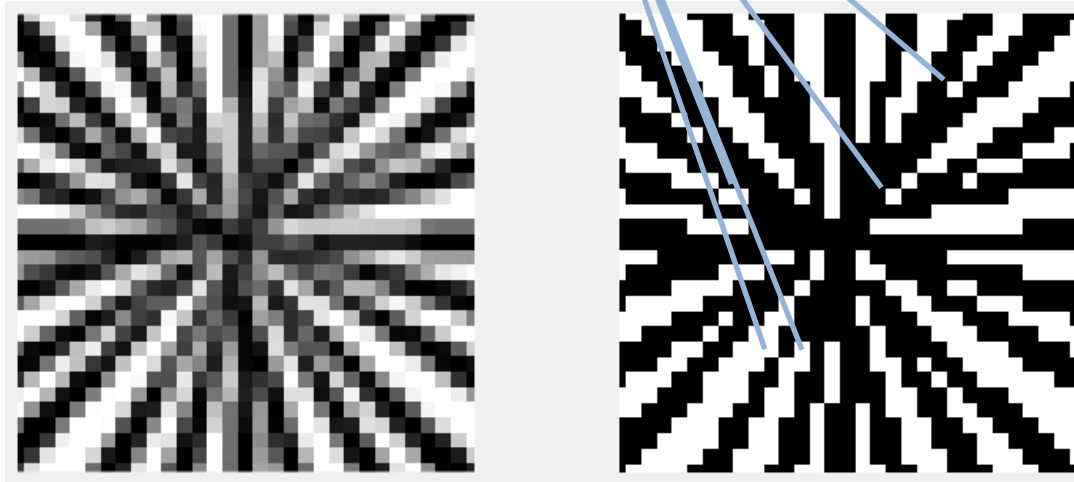
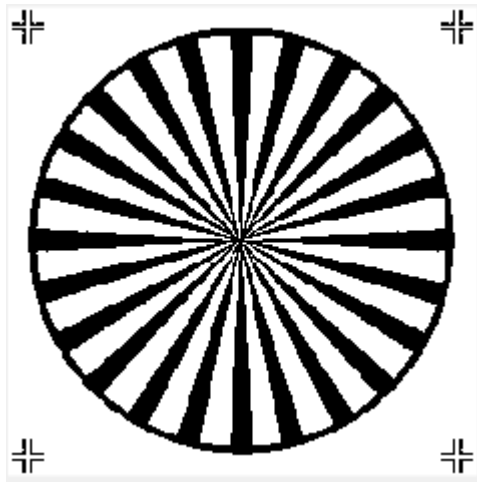


Image Morphology Example

Image Morphology

Count white spokes?

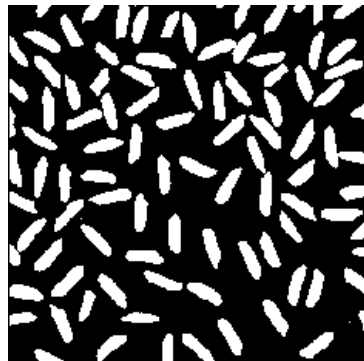
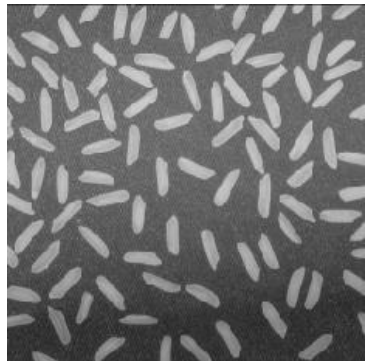


Binarization without morphology leads to connectivity issues, but `imclearborder` (morphology) provides clarity.

Image Segmentation

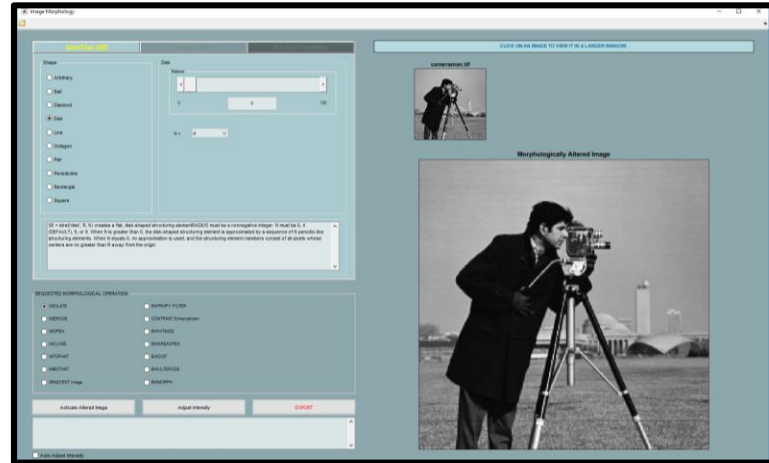
Creating masks where ROIs with similar traits are assigned the same value

Image Segmentation



Apps for Morphology and Segmentation

Image Morphology
(file exchange)



Color Thresholder

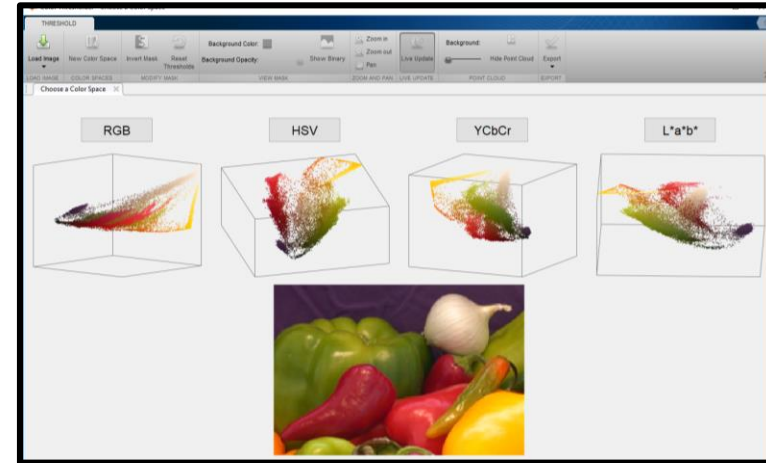


Image Segmenter

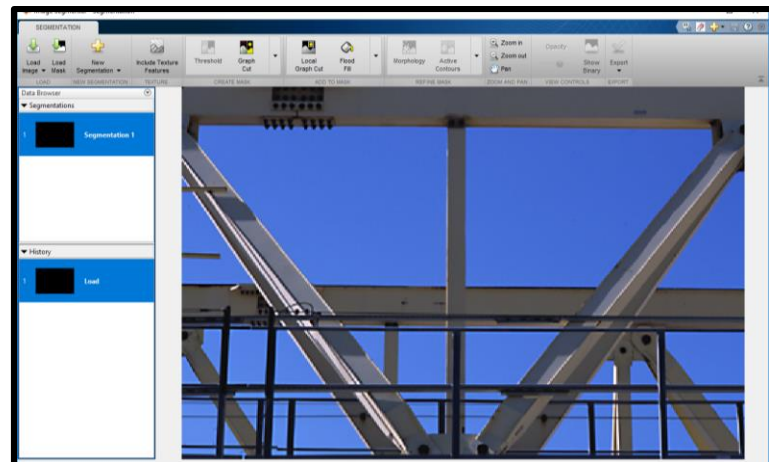
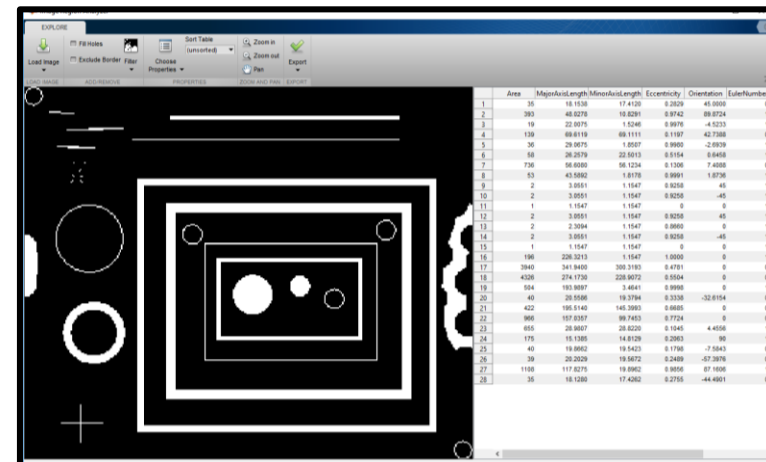


Image Region Analyzer



Instructor Demo

MorphologyAndSegmentation.mlx

Takeaways

- Dilation and erosion are building blocks of common morphological operations
- Morphology and segmentation often used together to extract meaningful information from images.

Agenda

Image Processing Workflow

Morphology and Segmentation

Object Detection and Tracking

Deep Learning

Summary + Next Steps

Computer Vision – Analyzing and Understanding Images at Higher Level



Object Detection Workflow

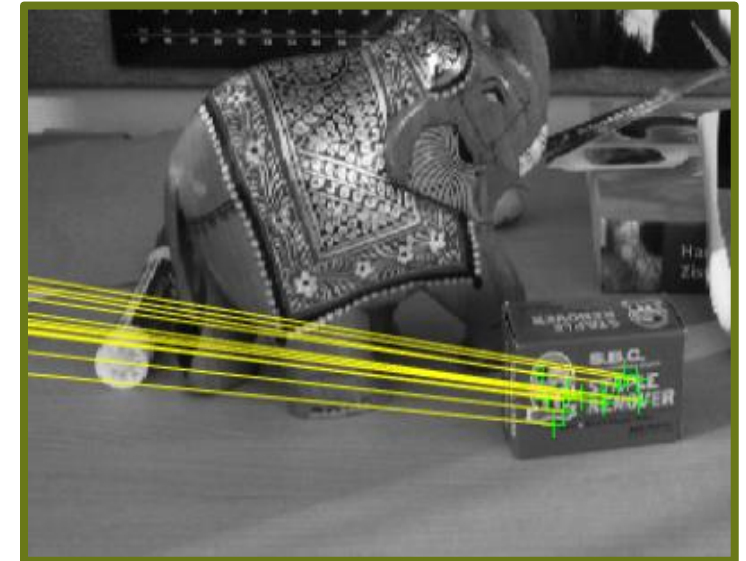
1. Detect



2. Extract

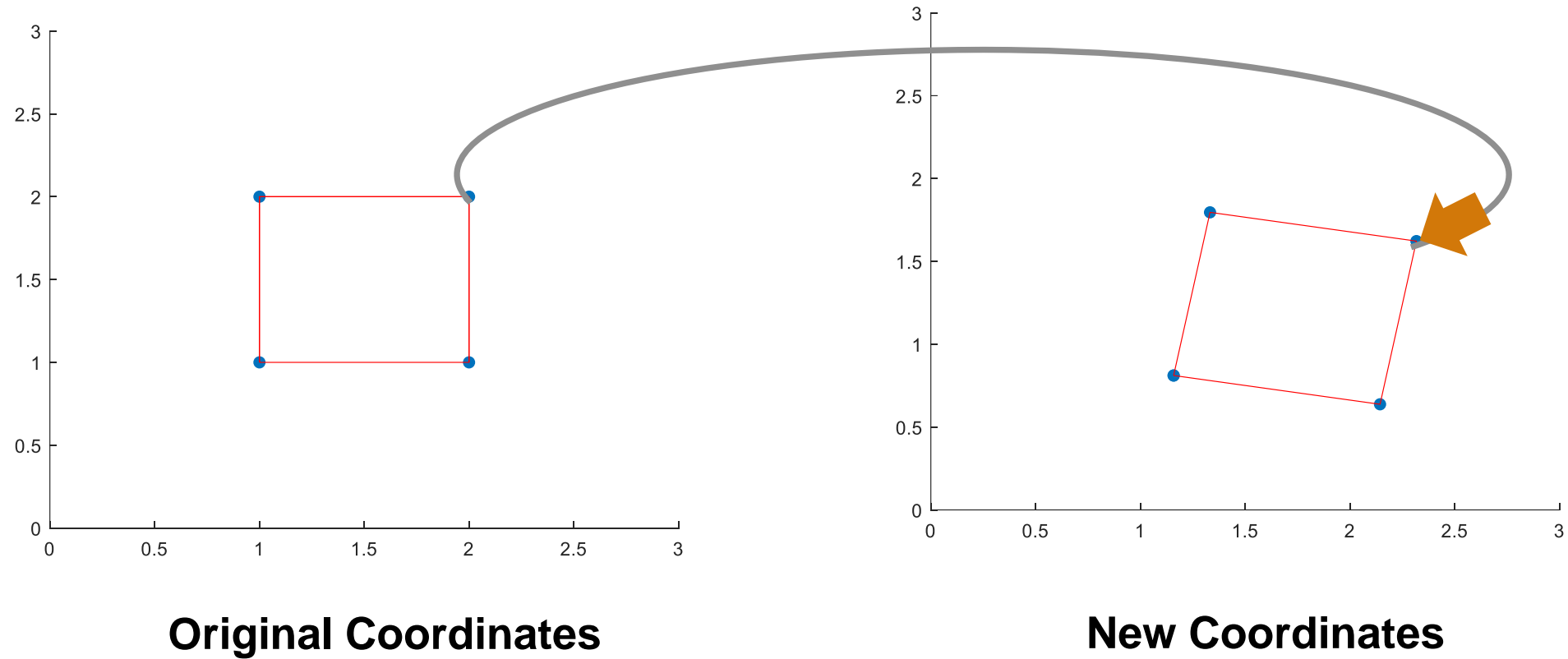
Compute a descriptor
for each feature

3. Match



Goal – to find a geometric transform between original image and target image

What is a Geometric Transform?



Instructor Demo

ObjectDetection.mlx

Object Tracking Workflow

1. Detect

2. Extract

3. Match

4. Track

**Maintain corresponding feature points from
frame to frame**

Why can't we detect objects in every frame?

- 1. Tracking allows you to maintain history*
- 2. Detection requires more computation than tracking*

What If You Don't Have a Reference Image?



Tracking in Action



Instructor Demo

Lecture_ObjectTracking.mlx

Takeaways

- Features are meaningful structures within images
- Computers use features to understand images
- Object Detection = Detect, Extract, Match
- Object Tracking = Detect, Extract Match, Track

Agenda

Image Processing Workflow

Morphology and Segmentation

Object Detection and Tracking

Deep Learning

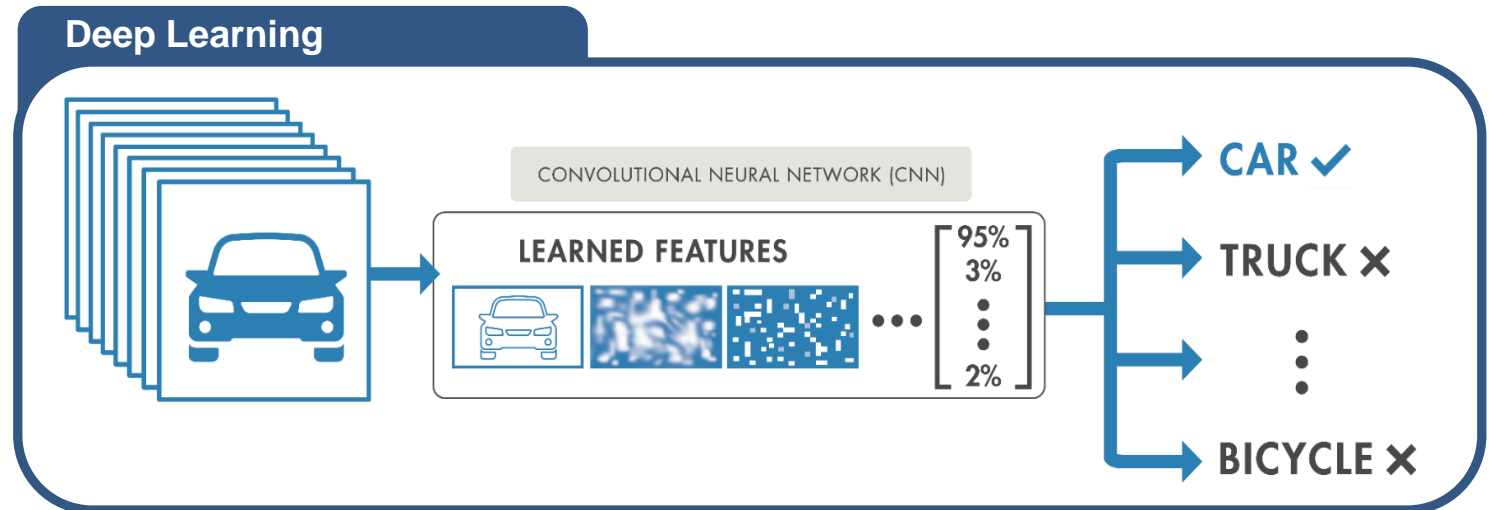
Summary + Next Steps

What is Deep Learning?

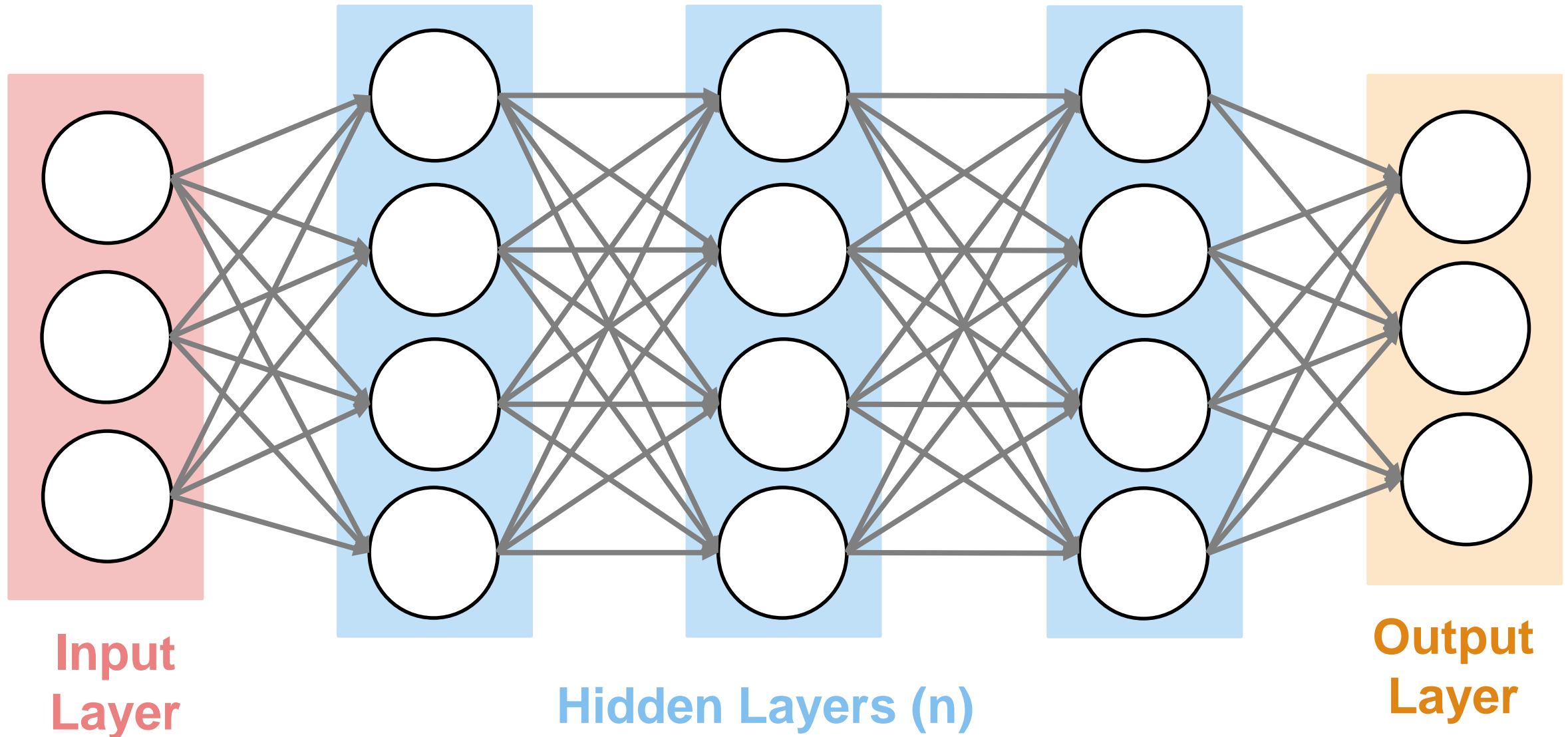
- Neural networks to learn tasks directly from data
- (Automatic feature extraction)

**Machine
Learning**

**Deep
Learning**



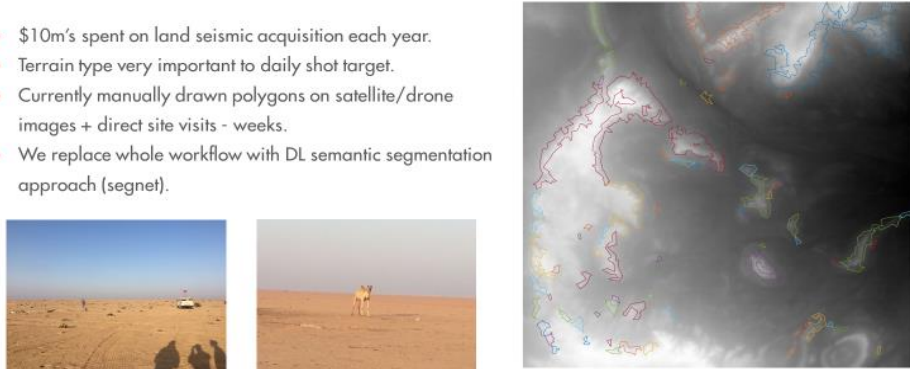
Deep Learning Uses a Neural Network Architecture



Applications of Deep Learning

Shell

- \$10m's spent on land seismic acquisition each year.
- Terrain type very important to daily shot target.
- Currently manually drawn polygons on satellite/drone images + direct site visits - weeks.
- We replace whole workflow with DL semantic segmentation approach (segnet).

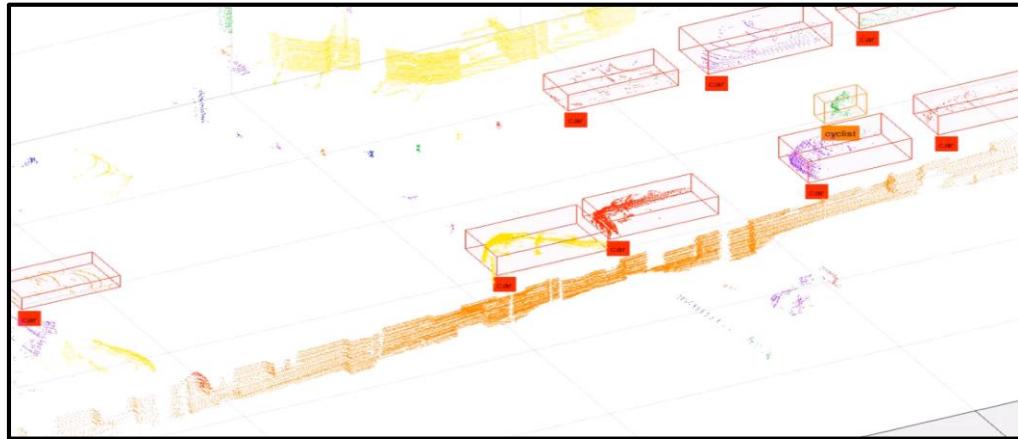


Copyright of Shell Global Solutions (UK)

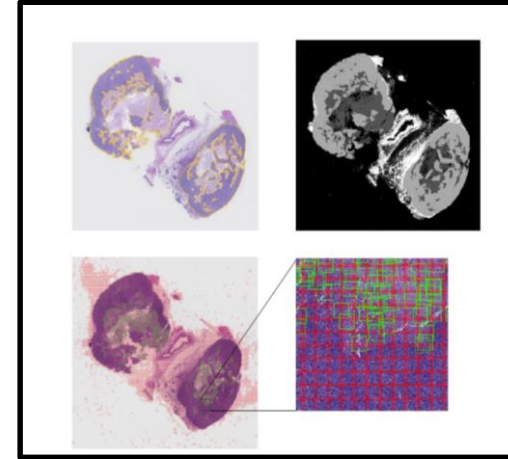
Radar image with rough polygons overlaid

Terrain Recognition with Hyperspectral Data

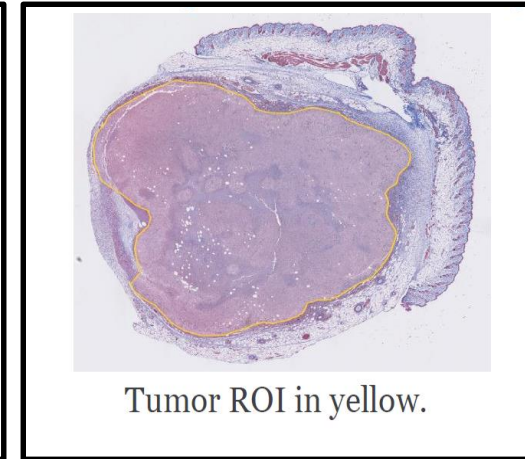
Veoneer



LiDAR-Based Sensor Verification



CNNs for Digital Pathology Analysis

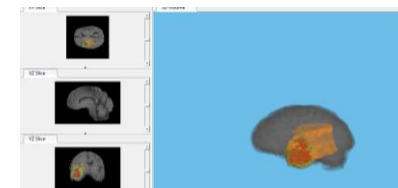
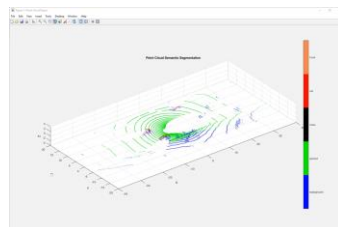
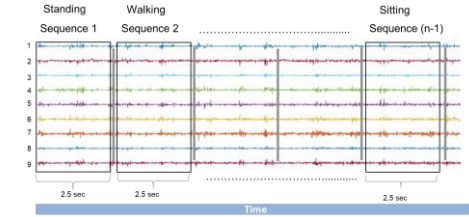
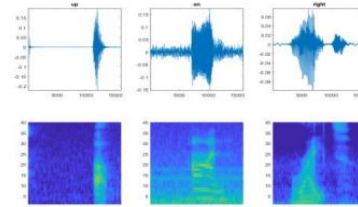
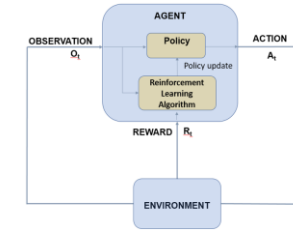
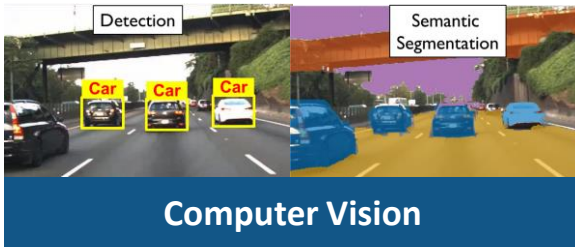


Genentech

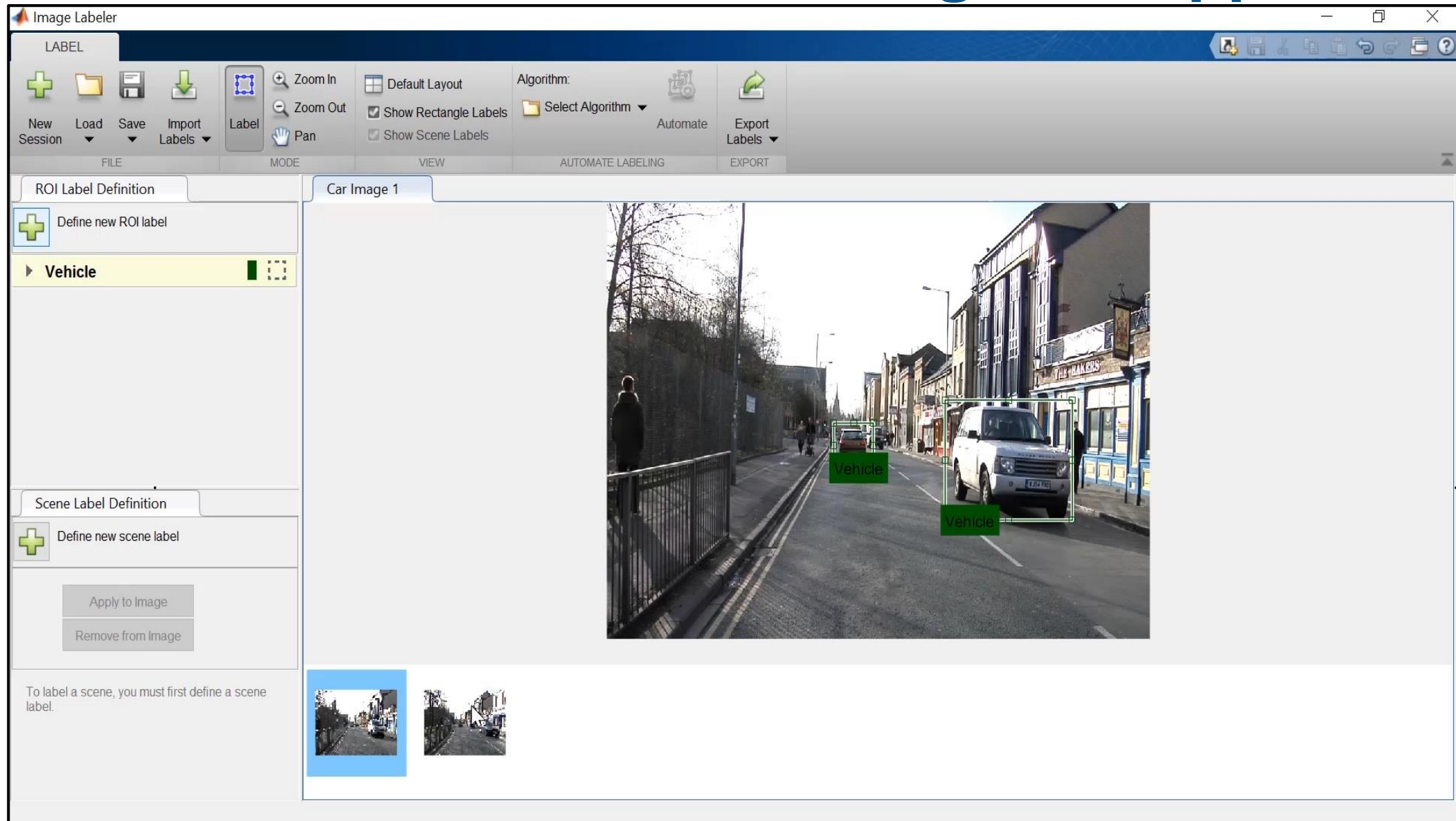


Equipment Classification

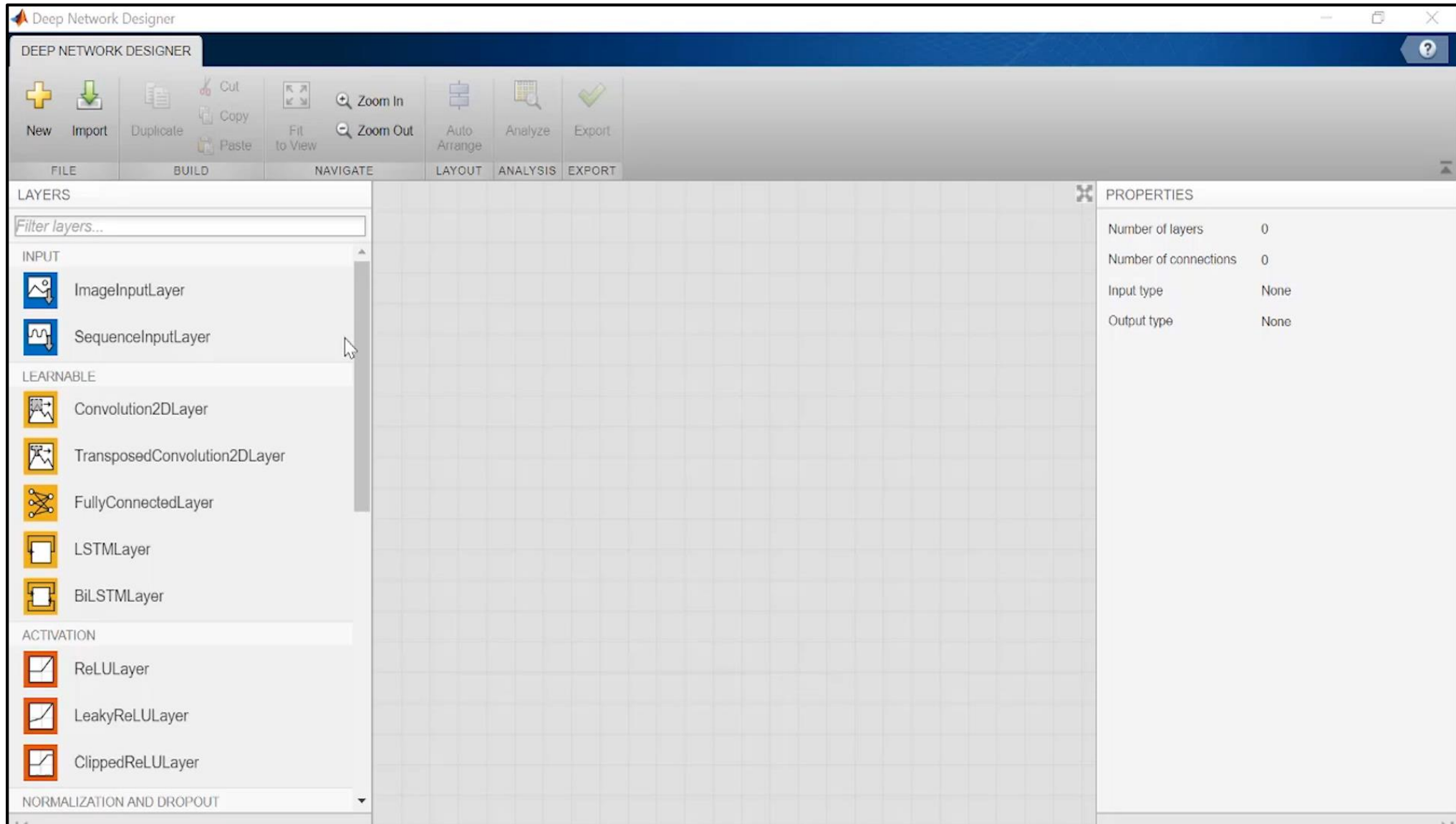
Caterpillar



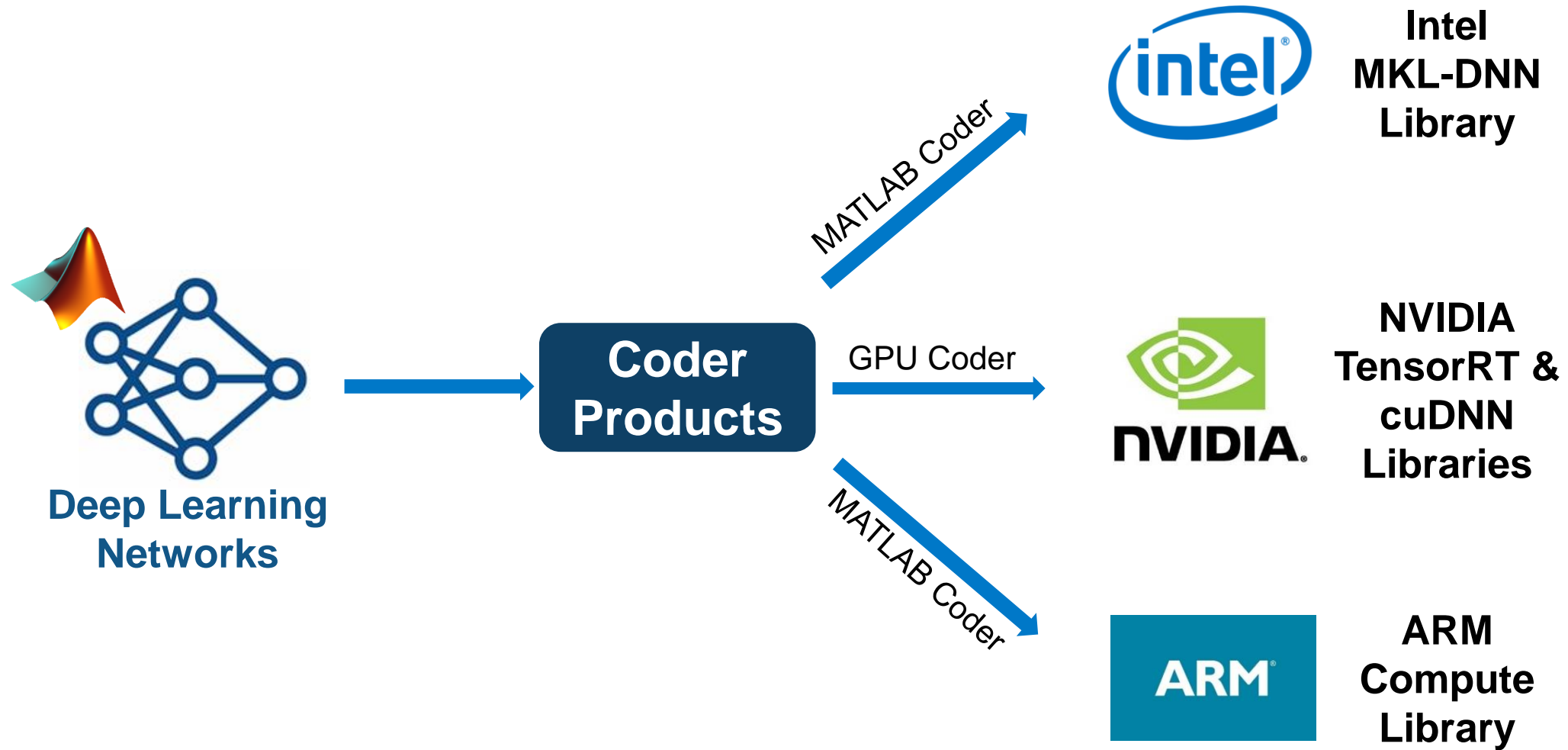
Semi-Automated Labeling with Apps



Interactive Deep Network Creation



Deploying Deep Learning Models for Inference



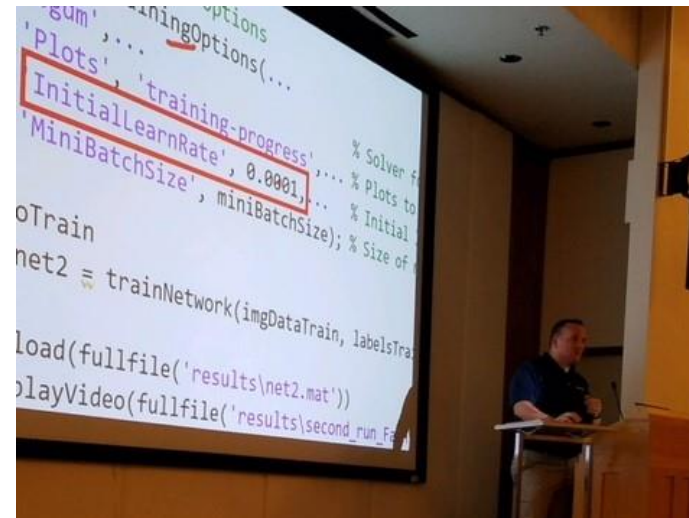
Instructor Demo

DeepLearning.mlx

Hands-on Deep Learning Workshop

Learn deep learning techniques and best practices in MATLAB

- Train deep networks using GPUs.
- Create networks from scratch and with transfer learning.
- Import and export models from Python frameworks (Keras, PyTorch)
- Learn techniques for improving network accuracy
- Label, train, and deploy YOLOv2 Object detector.



Agenda

Image Processing Workflow

Morphology and Segmentation

Object Detection and Tracking

Deep Learning

Summary + Next Steps

Core Modules

1. IPCV Workflow
2. Morphology and Segmentation
3. Object Detection and Tracking
4. Intro to Deep Learning

Filter and Analyze Mask Regions

We have segmented our cells, but the mask is still messy. We will clean up the mask using the **Image Region Analyzer** App.

| | Area | MaxXLength/MinXLength | MaxYLength/MinYLength | Perimeter | Orientation | SubNumber |
|----|------|-----------------------|-----------------------|-----------|-------------|-----------|
| 1 | 127 | 20 | 2000 | 8.0000 | 0.0000 | -480.0000 |
| 2 | 2631 | 60 | 1017 | 64.0070 | 0.2875 | -480.7500 |
| 3 | 2255 | 60 | 2240 | 47.0000 | 0.0000 | -74.7500 |
| 4 | 184 | 30 | 841 | 13.0000 | 0.0000 | -75.0000 |
| 5 | 1000 | 100 | 3070 | 80.0070 | 0.0000 | -2.7500 |
| 6 | 1000 | 60 | 8000 | 30.0000 | 0.0000 | 0.0000 |
| 7 | 3070 | 60 | 6300 | 80.0000 | 0.0000 | -480.0000 |
| 8 | 3102 | 60 | 7100 | 80.0000 | 0.0000 | -480.0000 |
| 9 | 1000 | 60 | 1400 | 40.0000 | 0.0000 | -480.0000 |
| 10 | 2400 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 11 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 12 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 13 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 14 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 15 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 16 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 17 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |
| 18 | 2000 | 60 | 6700 | 40.0000 | 0.0000 | -480.0000 |

Exercise:

1. Open the **Image Region Analyzer** from the Apps tab and load **cellMask**.
2. Fill in any "holes" in the cell mask.
3. Remove or "filter" the detected cells by size (**area**) to remove any little flecks and noise from the mask.
4. Save the processing algorithm by selecting "Export Function" under the Export options.
5. Name the function **filterCellRegions**.

The Fundamental Rules: Dilation and Erosion

In **dilation**, the value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In binary dilation, neighborhood of the of the origin of the strel pixels is set to the value 1, and the output pixel is set to 1.

As the origin of the structuring element is positioned over each pixel in an input image, the pixel value in an output image is modified that location the maximum of the input image's neighborhood.

```
animateOperation('Dilation', 'strel(''line'',3,0));
```

Match Features

Find the corresponding pairs of matching points in each image.

```
boxPairs = matchFeatures(boxFeatures, sceneFeatures);
if isempty(boxPairs)
    error("Not enough features - try a different combination.")
end

matchedBoxPoints = boxPoints(boxPairs(:,1), :);
matchedScenePoints = scenePoints(boxPairs(:,2), :);
figure;
showMatchedFeatures(boxImage, sceneImage, matchedBoxPoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```

Task 1: Create Network Architecture

Use the **Deep Network Designer** App to network architecture below. When you are finished, export the code to your workspace.

Use Default Parameters Except for the Following Layers

Imageinput
InputSize = 28,28,1

Conv
FilterSize = 5,5
NumFilters = 20
Padding = (0,0,0,0)

Maxpool
PoolSize = 2,2
Stride = 2,2

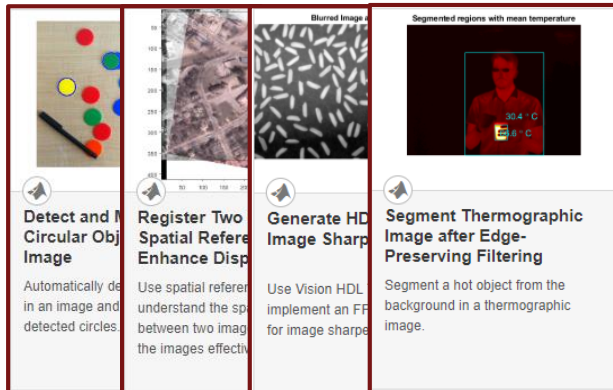
```
layers = [ imageInputLayer([28 28 1])
    convolution2dLayer(5,20)
    reluLayer
    maxPooling2dLayer(2,'Stride', 2)
    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer()
];
```

Task 2: Set Training Options

Use the following command and name value pairs to configure your training options.

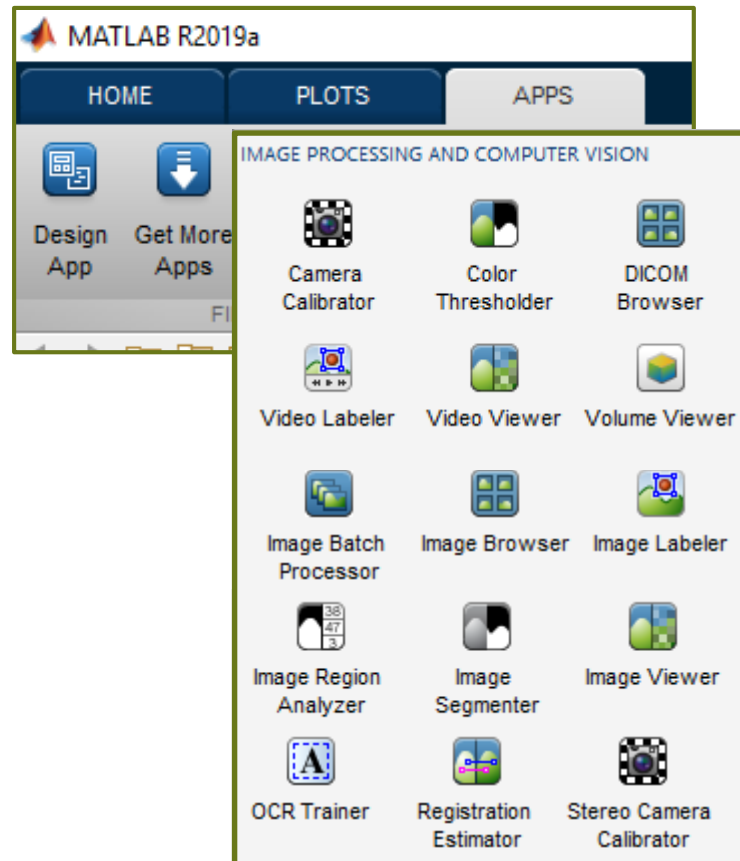
Why Use MATLAB?

Ease of Use and Thorough Documentation



(...)

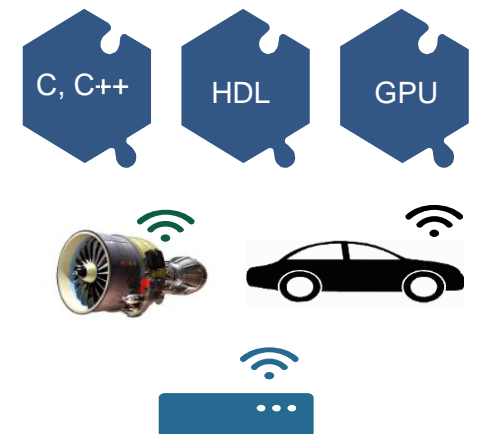
Rapid Prototyping and Algorithm Development



Code Generation for Embedded Deployment

MATLAB Code

Embedded Hardware



Need Technical Help?

- Technical Support
- Application Engineers

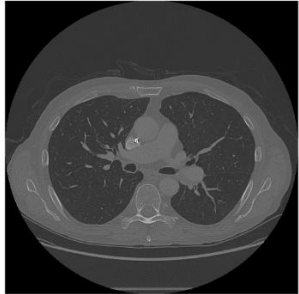
Prepare the Data

This part of the example loads the human chest CT scan data into the MATLAB workspace.

```
% Load the 3-D volumetric CT scan data into the MATLAB workspace
load chestVolume;

% Convert the CT scan data from int16 to single to normalize the values between [0 1]
V = im2single(V);

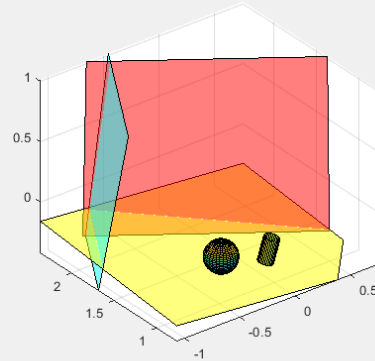
% sliceNum = 250
sliceNum = 199;
imshow(V(:,:,sliceNum),[])
```



```
h3.FaceAlpha = 0.5;
```

```
plot(sphereMdl)
plot(cylinderMdl)
```

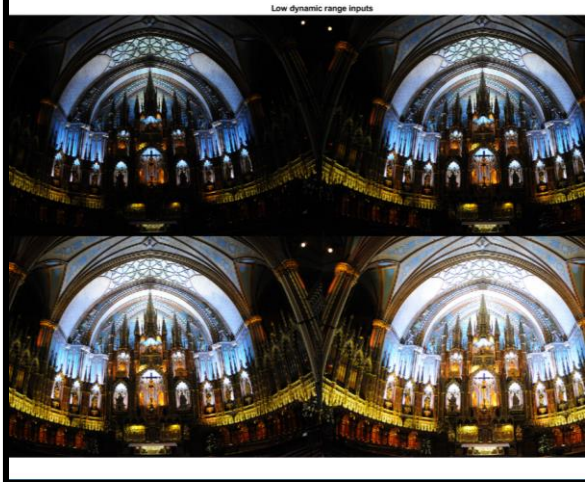
```
% You can turn off some graphic objects
set(h.Children(6),'visible','off')
```



Additional Modules

- 3D Vision and Point Cloud Processing
- Image Registration
- 3D Image Processing and Visualization
- Camera Pipeline Workflow
 - Image formation
 - LDR to HDR
 - Haze reduction and low-light enhancement
- Deep Learning Workshop

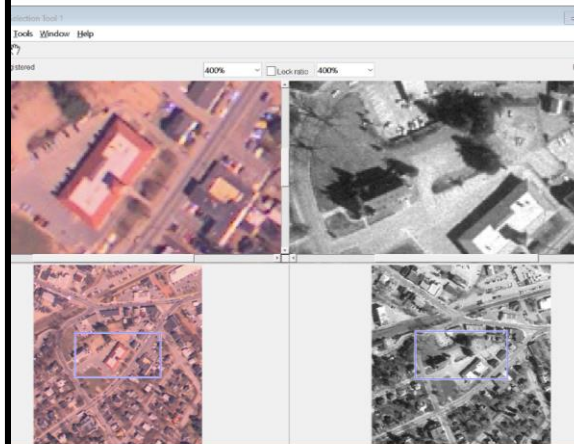
```
lowDynamicRangeInputs = imread('LowDynamicRangeInputs.mat');
imshow(lowDynamicRangeInputs,[]);
title('Low dynamic range inputs')
```



```
last four corresponding points in the images. Try to get a sampling of points from different parts of the image.
are finished selecting points, use File > Export Points to Workspace to access the control points later.
```

In the control point selection procedure can be found in the Image Processing Toolbox documentation.

```
istered, fixed)
```



Information matrix from the first image to the second