



Deep Learning Overview

Signals and Time-Series



Agenda

- I. Deep learning in engineering and science
- II. Developing a deep learning solution in MATLAB
- III. MathWorks deep learning support



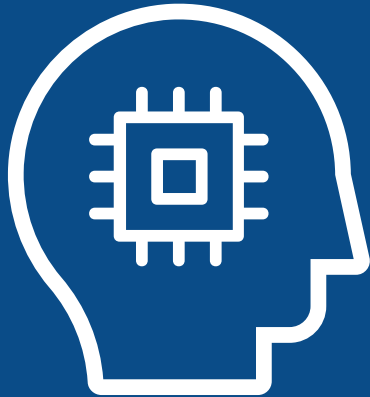
Agenda

- I. Deep learning in engineering and science
- II. Developing a deep learning solution in MATLAB
- III. MathWorks deep learning support

Deep learning is a key technology driving the AI megatrend

ARTIFICIAL INTELLIGENCE

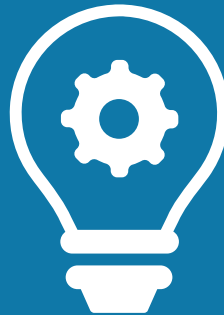
Any technique that enables machines to mimic human intelligence



1950s

MACHINE LEARNING

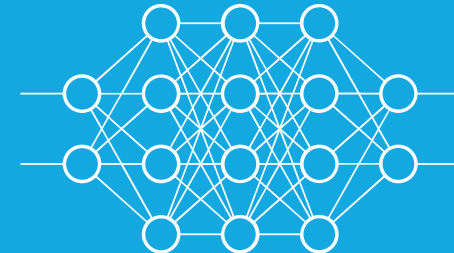
Statistical methods that enable machines to “learn” tasks from data without explicitly programming



1980s

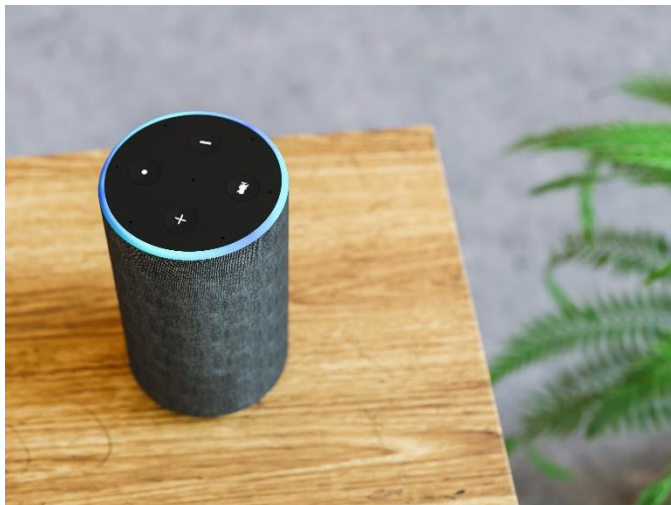
DEEP LEARNING

Neural networks with many layers that learn representations and tasks “directly” from data



2010s

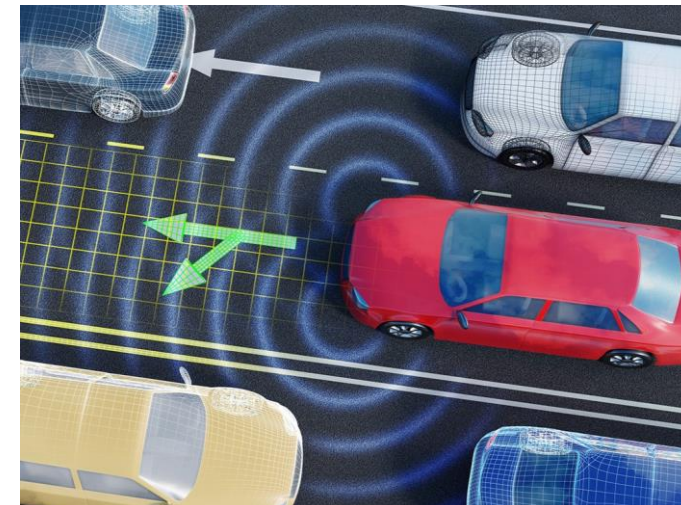
Deep learning is part of our everyday lives



Speech Recognition



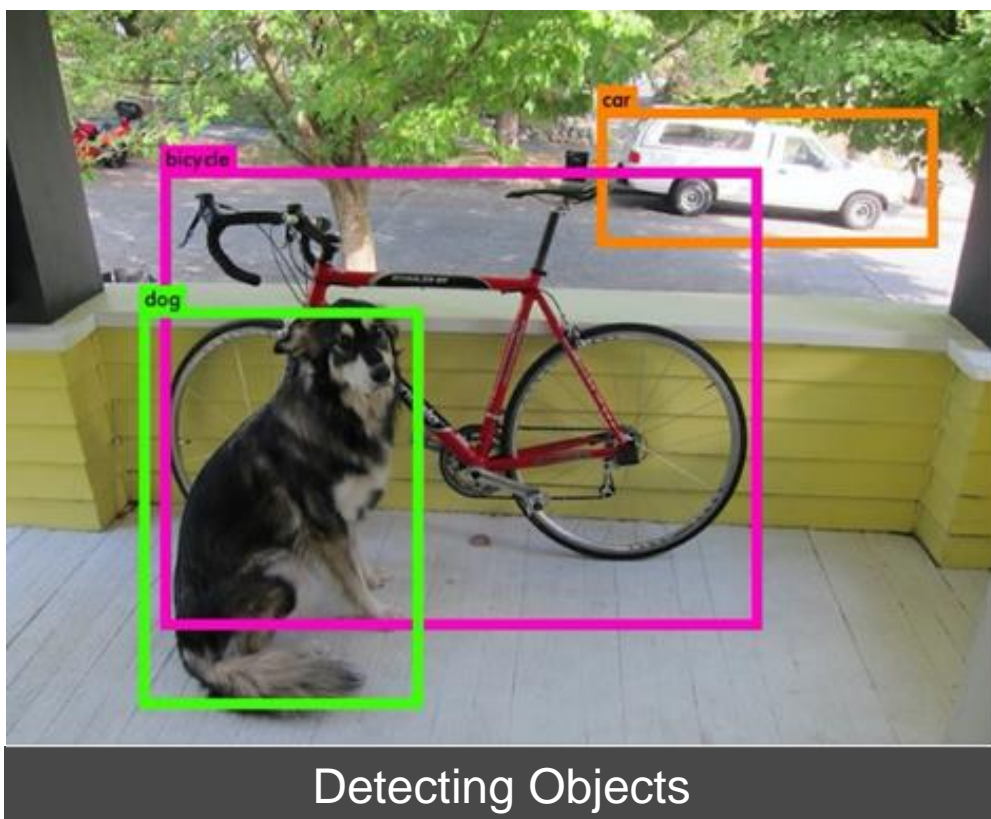
Face Detection



Automated Driving

Deep learning applications: mainstream vs. engineering

Mainstream



Engineering and Science

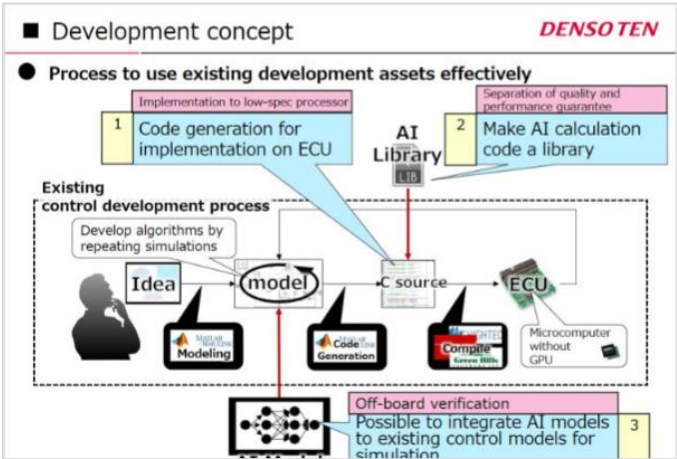


Deep Learning Detection

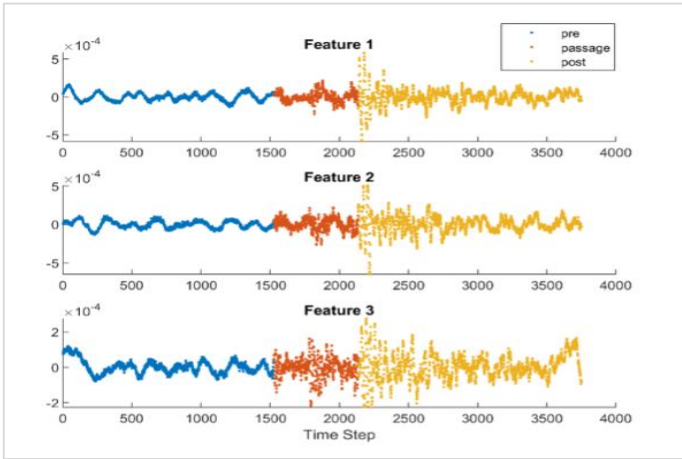
MATLAB Deep Learning used in Industry



Automatic Defect Detection
Airbus

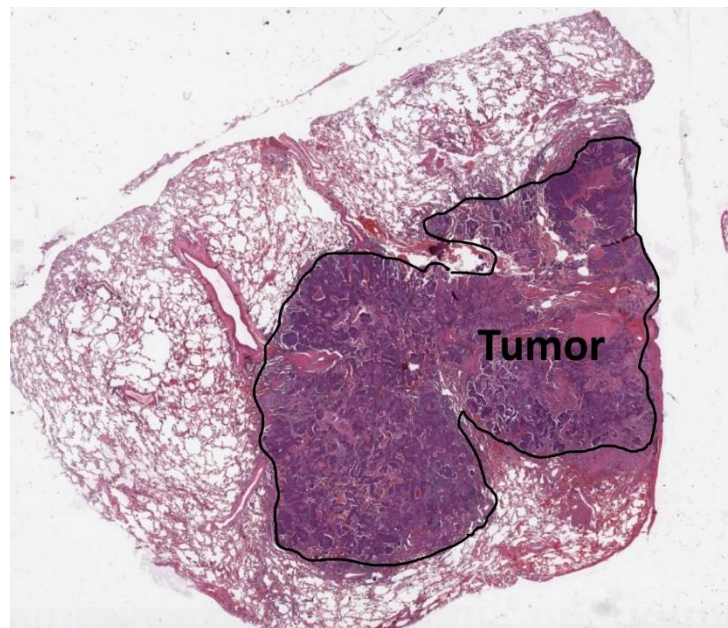


ECU Vehicle Control
Denso

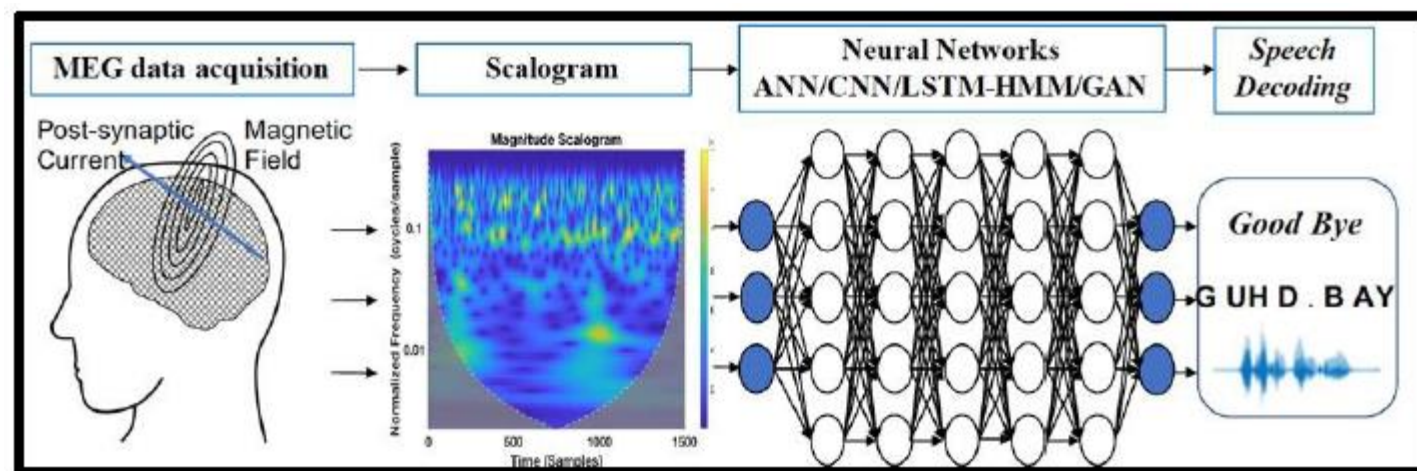


Seismic Event Detection
Shell

MATLAB Deep Learning used in Research



Predicting gastrointestinal cancer (July 2019)



Converting brain waves to speech to help ALS patients communicate (Nov 2019)

Evolution of Deep Learning in MATLAB

2016

CNN's
Pretrained Models
Caffe Importer

2017

Name Change

- Neural Network Toolbox to Deep Learning Toolbox

Algorithms

- LSTM's
- Directed Acyclic Graphs
- Multi-GPU Training

Code Generation

- GPU Coder

Apps

- Image Labeler

Interoperability

- TensorFlow-Keras Importer

2018

Examples

- Signal Processing
- Audio
- Text Analytics

Algorithms

- Wavelet Scattering

Code Generation

- MATLAB Coder C++

Apps

- Deep Network Designer
- Video Labeler
- Audio Labeler

Interoperability

- ONNX Support

2019

- Reinforcement Learning

Algorithms

- Automatic Differentiation
- Custom Training Loops
- Weight Sharing
- Big Image

Examples

- GANs
- Siamese Network
- Autoencoders
- 3-D support
- **Explainable AI**
 - Occlusion
 - Grad-CAM

Code Generation

- MATLAB Coder (ARM)

Apps

- Signal Labeler

2020

- Deep Learning Data Sets

Apps

- Experiment Manager
- Lidar Labeler

Examples

- 5G / Wireless
- Lidar

- **Over 200+ examples!**

Algorithms

- Point Cloud

Explainable AI

- **Lime**

Code Generation

- Quantization
- Deep Learning HDL Coder

Model Based Design

- Deep Learning in Simulink

Applications of deep learning for images and video

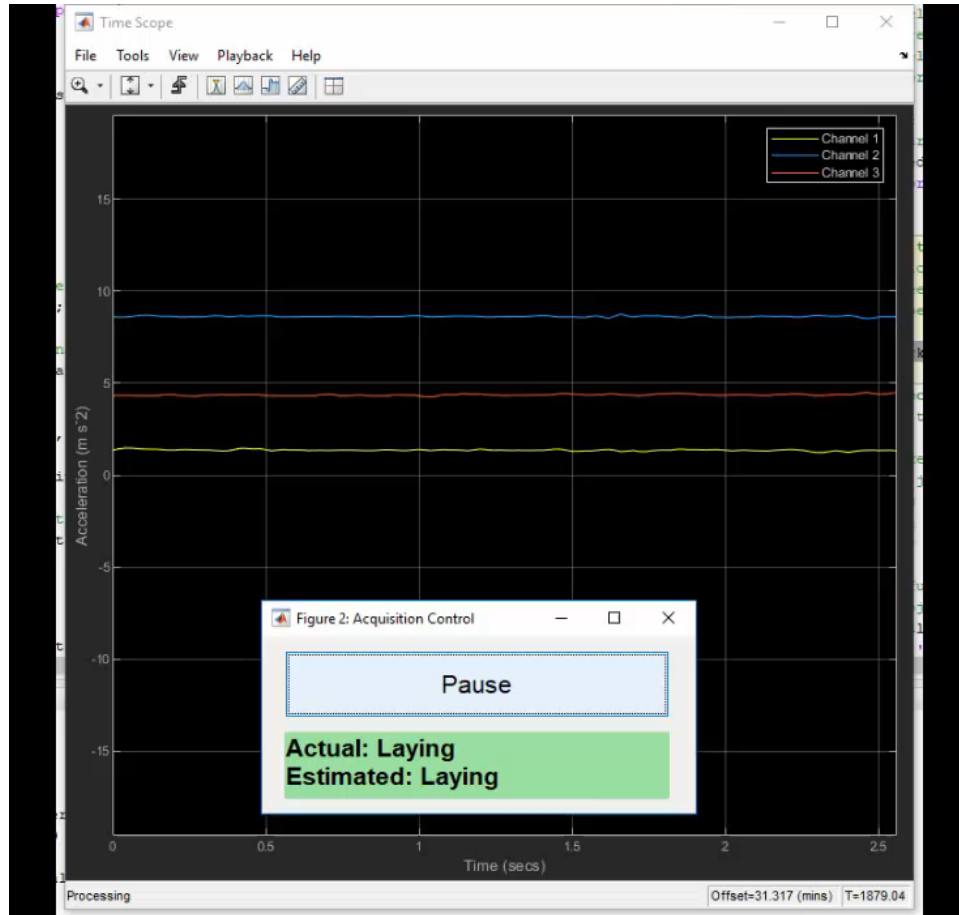


YOLO v2 (You Only Look Once)

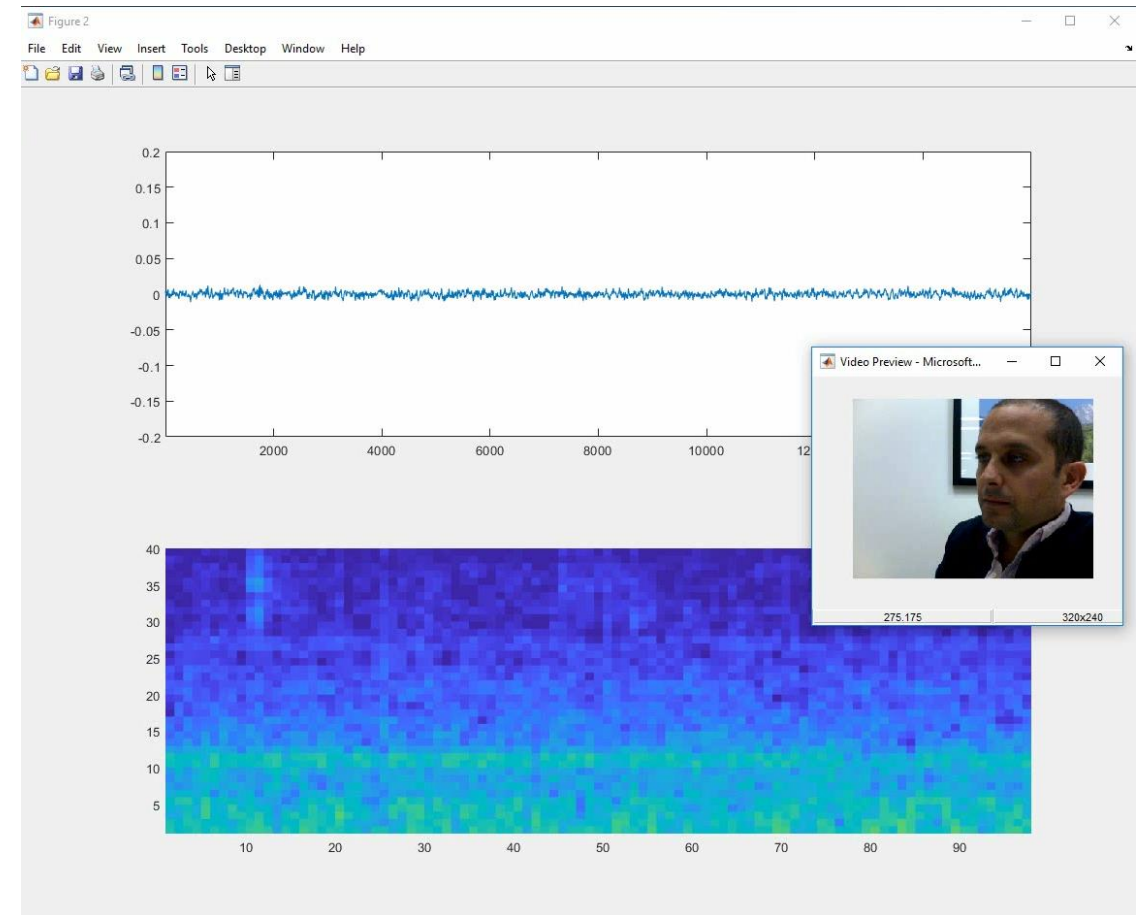


Semantic Segmentation using SegNet

Applications of deep learning for signal processing

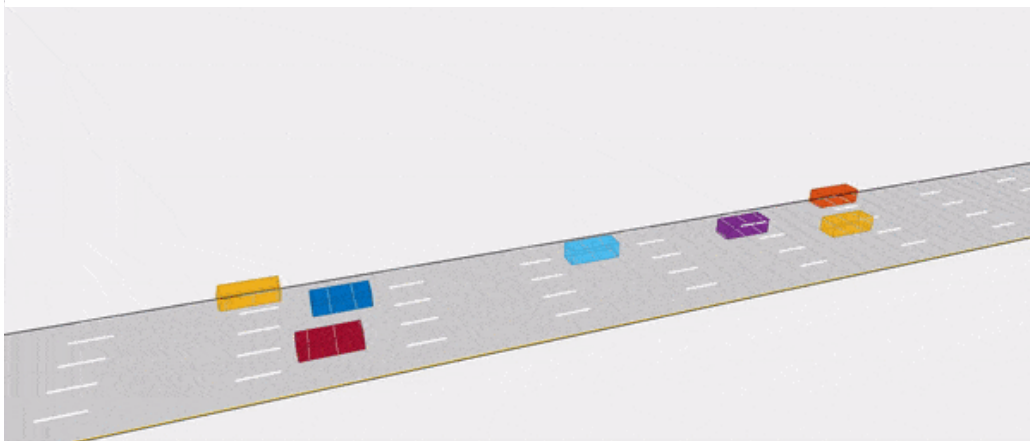


Signal Classification using LSTMs

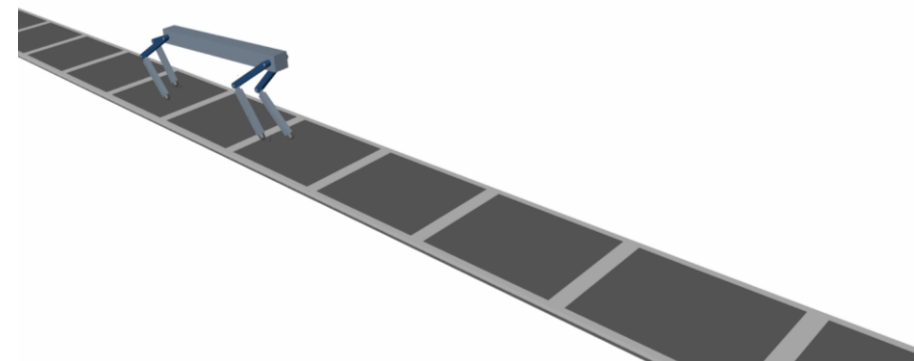


Speech Recognition using CNNs

Applications of reinforcement learning



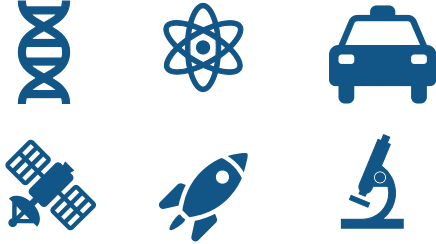
Teach a car to navigate traffic



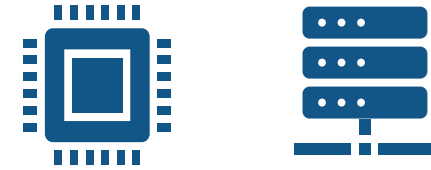
Train a Quadruped robot to walk

Why MATLAB & MathWorks for Deep Learning?

Domain-specialized workflows
for **engineering and science**



Multi-platform **deployment** of
full applications and systems



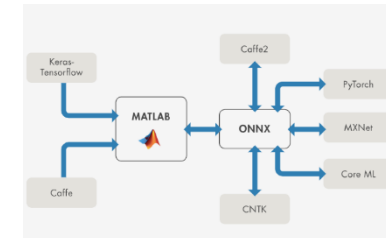
People



Platform productivity



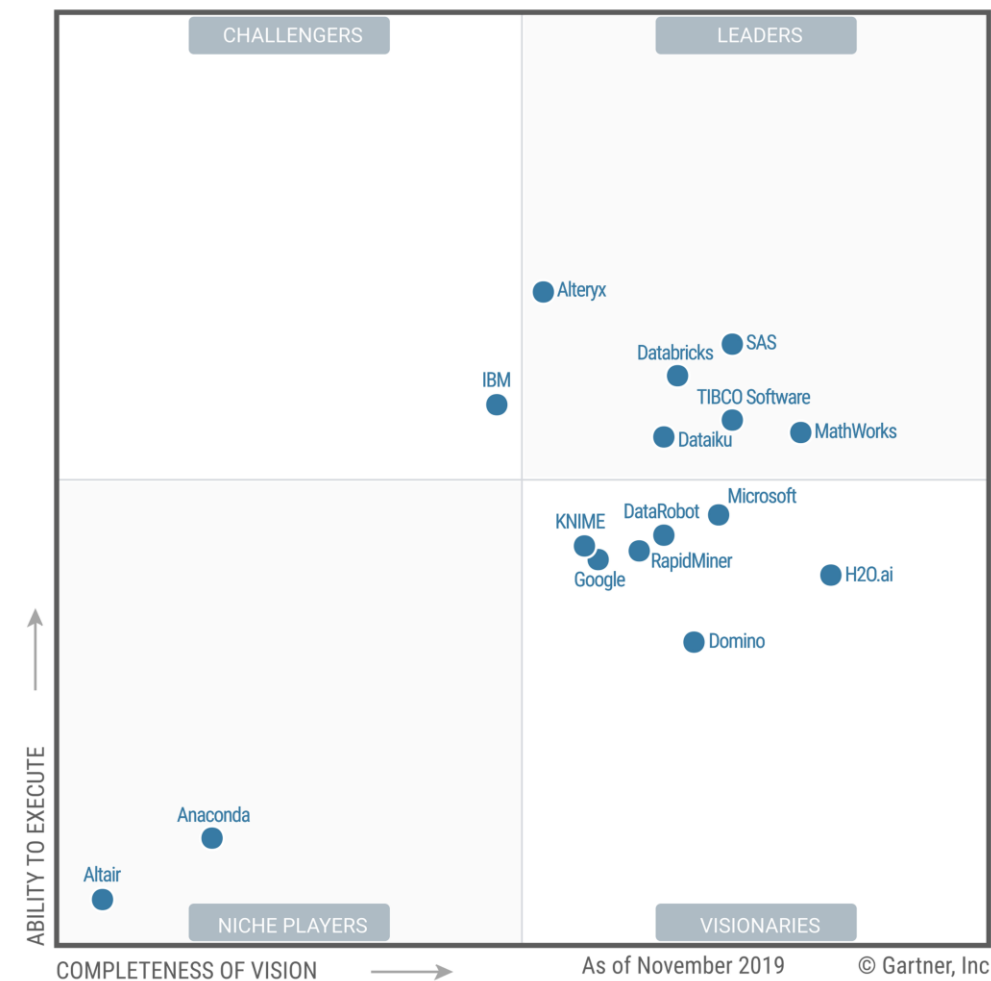
Interoperability with
TensorFlow and PyTorch





is a **Leader** in the Gartner Magic Quadrant for 2020 Data Science and Machine Learning Platforms

Figure 1. Magic Quadrant for Data Science and Machine Learning Platforms



Source: Gartner (February 2020)

*Gartner Magic Quadrant for Data Science and Machine Learning Platforms, Peter Krensky, Erick Brethenoux, Jim Hare, Carlie Idoine, Alexander Linden, Svetlana Sicular, 11 February 2020 .

This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from MathWorks. Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, express or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.



A 2020 Gartner Peer Insights Customers' Choice for Data Science and Machine Learning Platforms



is named a 2020 Gartner Peer Insights Customers' Choice for Data Science and Machine Learning Platforms.



Figure 1. Gartner Peer Insights "Voice of the Customer" Data Science and Machine Learning Platforms Customers' Choice



Source: Gartner (July 2020)

Disclaimer: Gartner, Gartner Peer Insights 'Voice of the Customer': Data Science and Machine Learning Platforms, July 2020. This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from MathWorks.

The Gartner Peer Insights Customers' Choice badge is a trademark and service mark of Gartner, Inc., and/or its affiliates, and is used herein with permission. All rights reserved. Gartner Peer Insights Customers' Choice constitute the subjective opinions of individual end-user reviews, ratings, and data applied against a documented methodology; they neither represent the views of, nor constitute an endorsement by, Gartner or its affiliates.



AI-driven system design

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



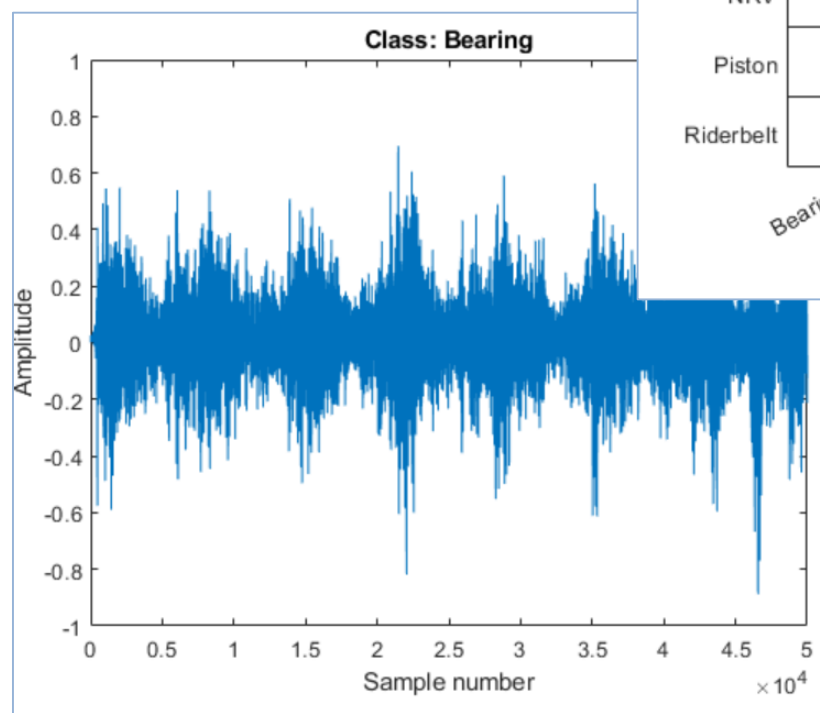
Agenda

- I. Deep learning in engineering and science
- II. Developing a deep learning solution in MATLAB
- III. MathWorks deep learning support

Featured Example: Classifying Air Compressor Sounds

Build, test, and deploy a deep learning solution that can classify sequences in signal data.

- [Long Short-Term Memory Networks](#)
- Classify sequence data
- Anomaly detection, natural language processing (NLP)



| True Class | Bearing | 20 | 4 | | | | | 1 | |
|------------|-----------|-----------------|----------|---------|-----|-----|-----|--------|-----------|
| | Flywheel | 2 | 18 | | | | | | |
| | Healthy | | | 21 | | | | | |
| | LIV | | | | 19 | 1 | 1 | | |
| | LOV | | | 1 | 2 | 19 | 1 | | |
| | NRV | | | | 1 | 1 | 19 | 1 | |
| | Piston | | | | | | 1 | 20 | |
| | Riderbelt | | | | | 1 | | | 22 |
| | | Bearing | Flywheel | Healthy | LIV | LOV | NRV | Piston | Riderbelt |
| | | Predicted Class | | | | | | | |

Data preparation represents most of your AI effort...

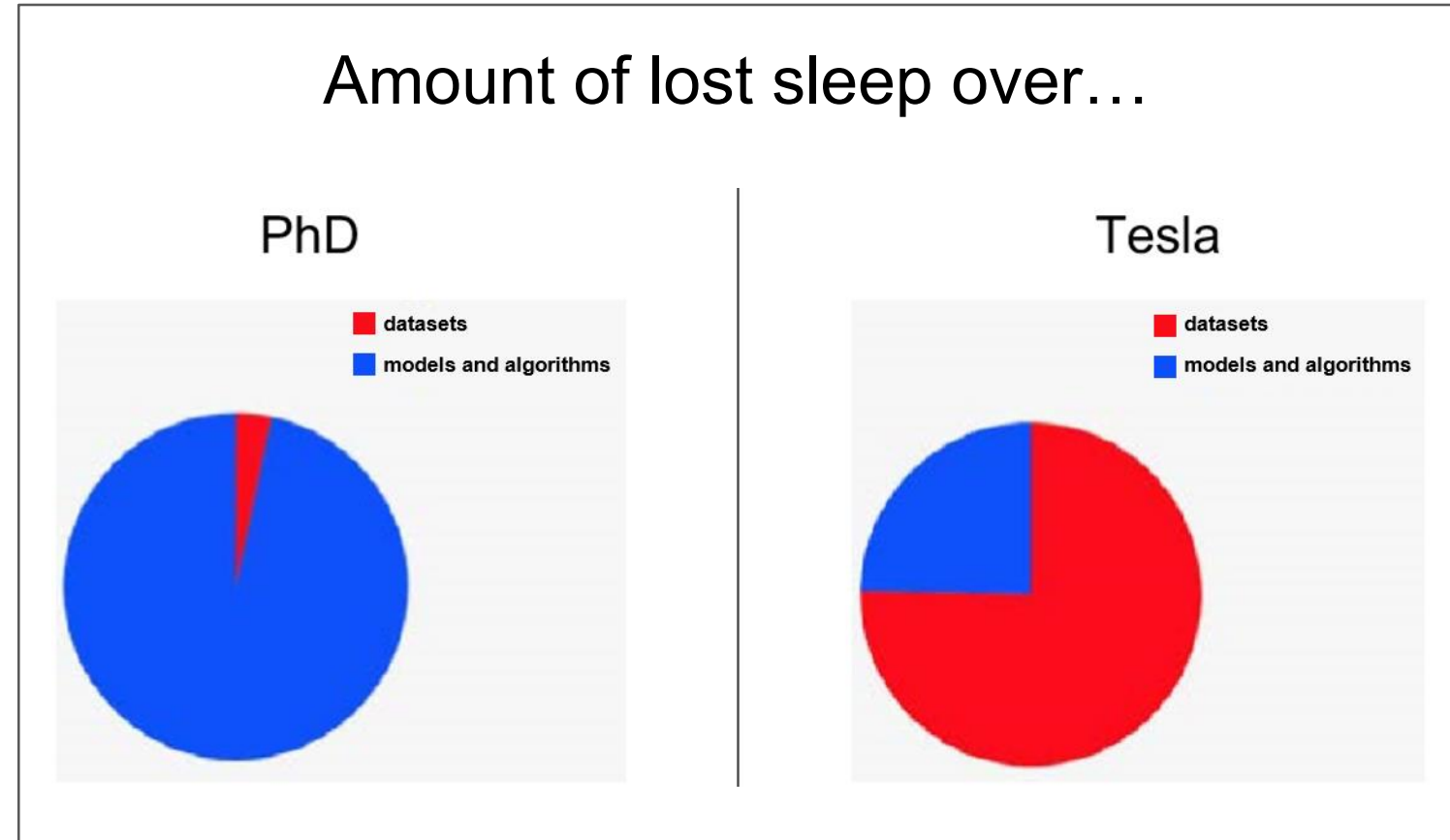
Transforming raw data for useful modeling and analysis is a critical step.

Data Preparation

 Data cleansing and preparation

 Human insight

 Simulation-generated data



Source: Andrej Karpathy slide from TrainAI 2018

Spend less time preprocessing and labeling data

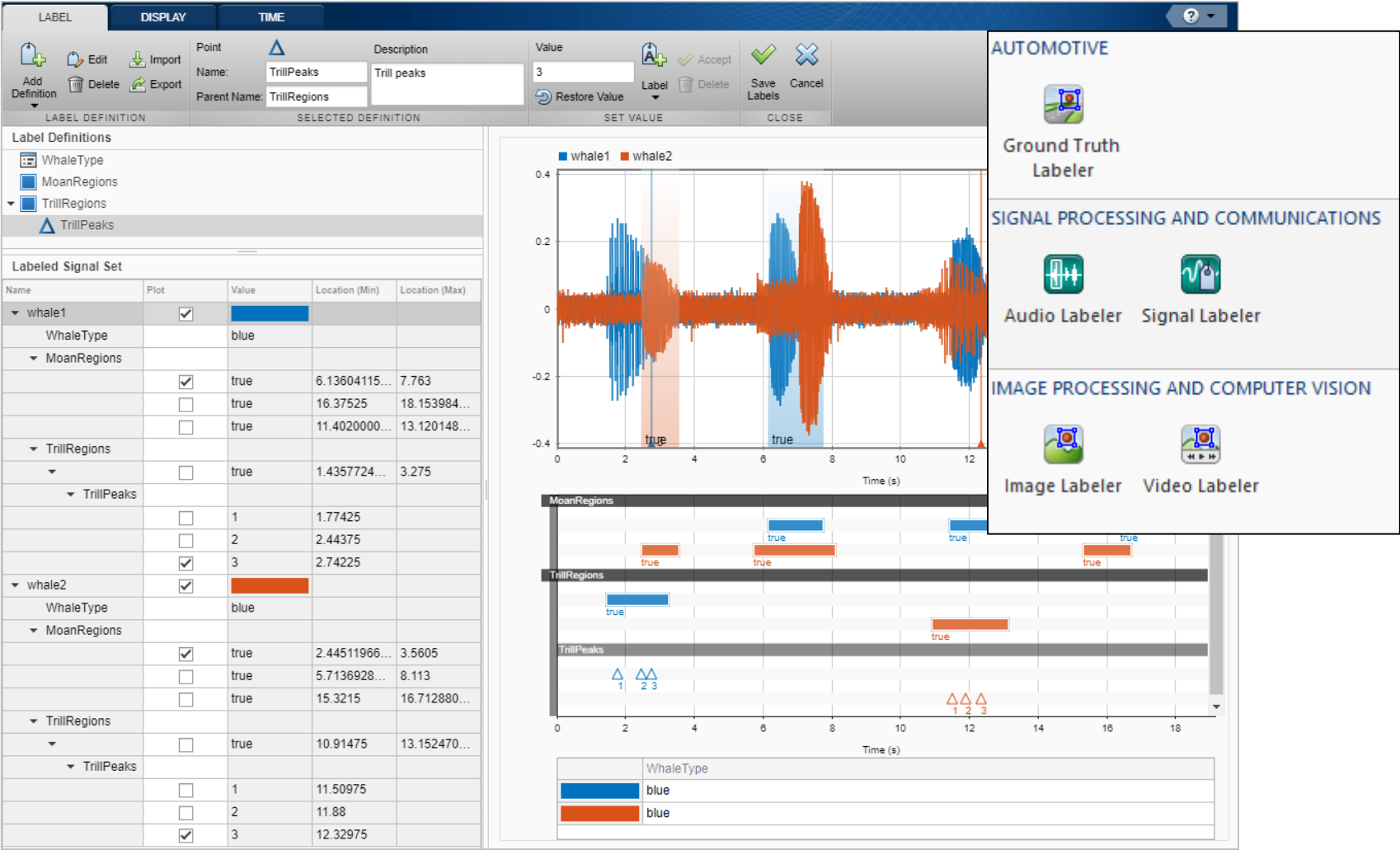
Synchronize disparate time series, filter noisy signals, automate labeling of video, and more.

Data Preparation

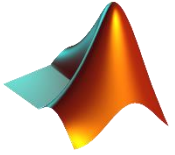
Data cleansing and preparation

Human insight

Simulation-generated data



Data Preparation Demo



Open Script
Part 1

Data Preparation



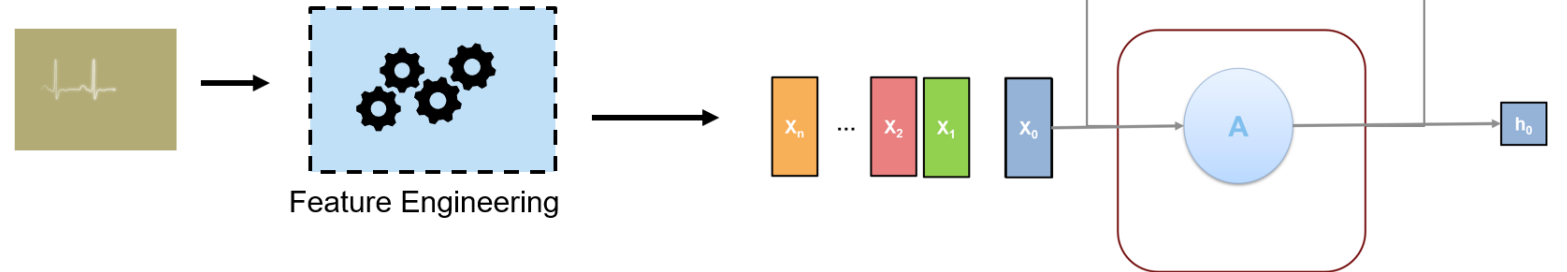
Data cleansing and
preparation



Human insight



Simulation-
generated data



Long Short Term Memory (**LSTM**) Networks

Start with a complete set of algorithms and pre-built models

AI Modeling



Model design and tuning



Hardware accelerated training



Interoperability

Algorithms

Machine learning

Trees, Naïve Bayes, SVM...

Deep learning

CNNs, GANs, LSTM, MIMO...

Reinforcement learning

DQN, A2C, DDPG...

Regression

Linear, nonlinear, trees...

Unsupervised learning

K-means, PCA, GMM...

Predictive maintenance

RUL models, condition indicators...

Bayesian optimization

Pre-built models

Image classification models

AlexNet, GoogLeNet, VGG, SqueezeNet, ShuffleNet, ResNet, DenseNet, Inception...

Reference examples

Object detection

Vehicles, pedestrians, faces...

Semantic segmentation

Roadway detection, land cover classification, tumor detection...

Signal and speech processing

Denoising, music genre recognition, keyword spotting, radar waveform classification...

...and more...

Increase productivity using Apps for design and analysis

Use MATLAB Apps to design deep learning networks, explore a wide range of classifiers, train regression models, train an optical character recognition model, and more.

AI Modeling



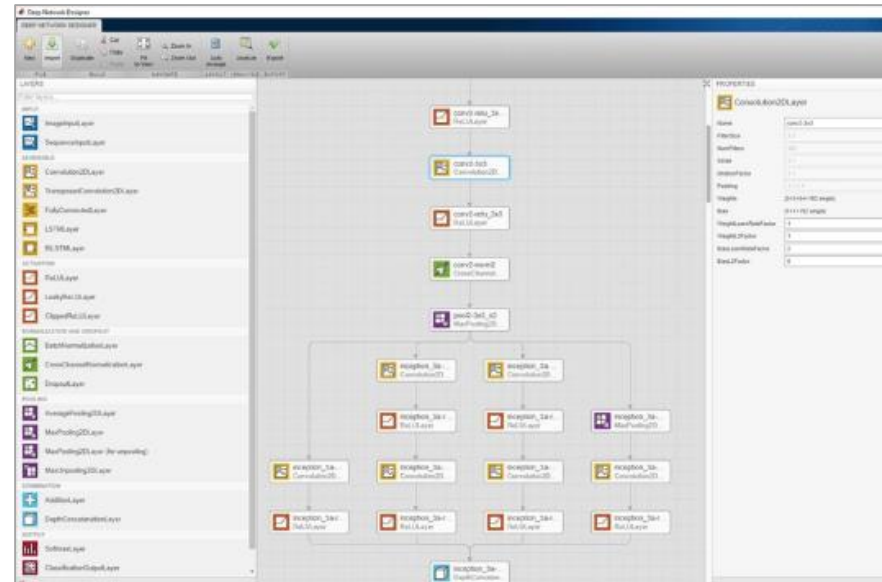
Model design and tuning



Hardware accelerated training



Interoperability



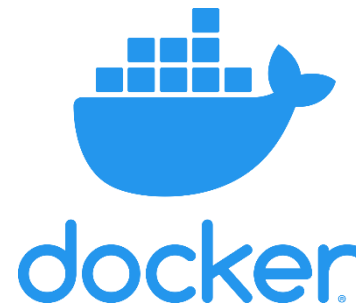
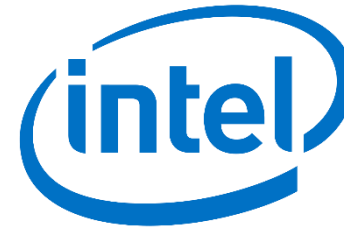
Deep Network Designer app to build, visualize, and edit deep learning networks

| 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 | 2.4441 | 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 | 2.1233 | 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 | | | |


Experiment Manager app to manage multiple deep learning experiments, analyze and compare results and code

Hardware acceleration and scaling are critical for training


MATLAB accelerates AI training on GPUs, cloud, and datacenter resources without specialized programming.




AI Modeling



Model design and tuning



Hardware accelerated training






Interoperability

Performance is Continuously Improving

Development continuously measures, tracks, and improves performance for training and inference

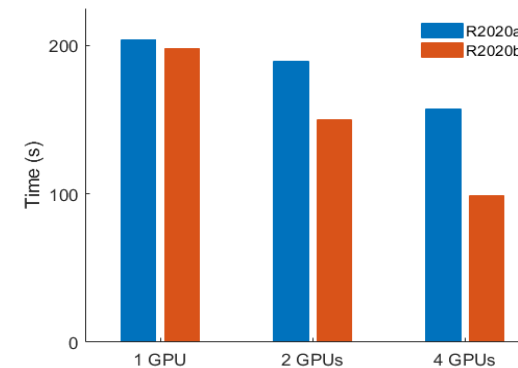
AI Modeling

-  Model design and tuning
-  Hardware accelerated training
-  Interoperability

Training

R2020a-R2020b

Multi-GPU 1.6x speedup



Windows 10, Intel® Xeon® E5-2623 v4 @ 2.60 GHz
NVIDIA® Titan V 12GB GPUs

Inference

R2018b-R2020b

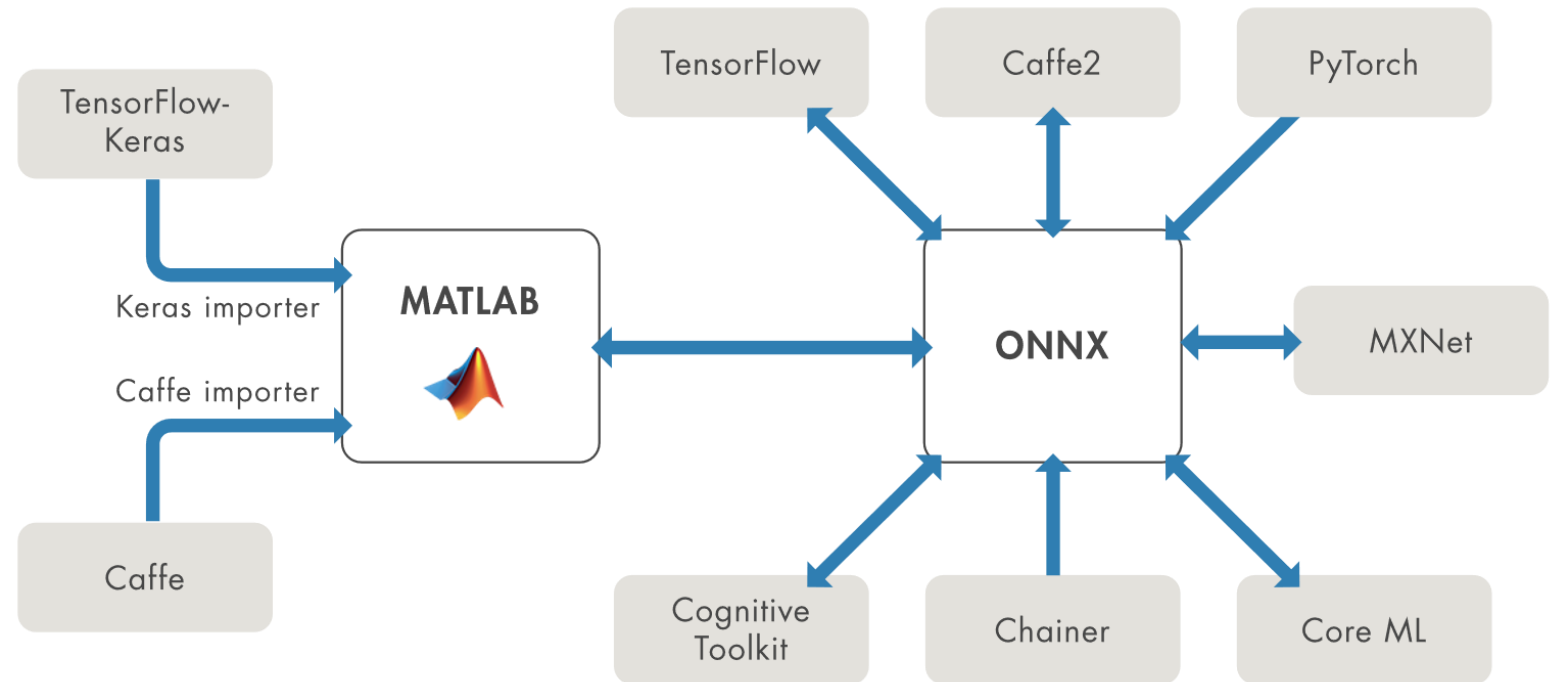
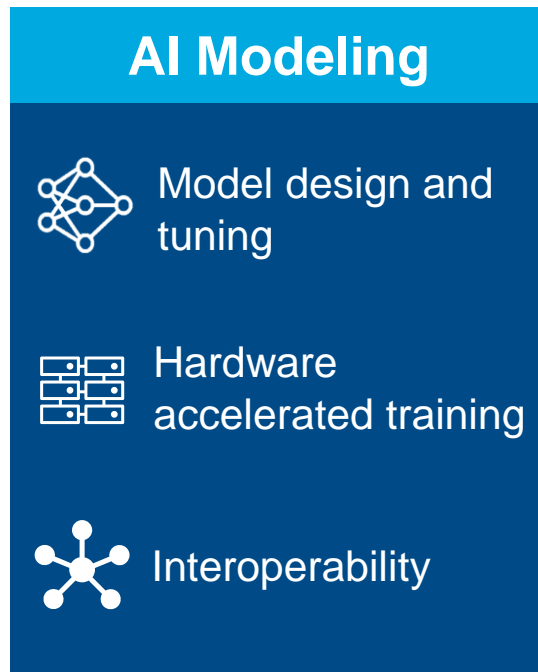
GPU 2.8x speedup

CPU 3x speedup

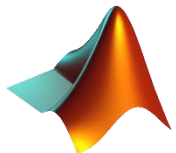
ResNet-50, Batch Size 32, Intel® Xeon® CPU 3.6 GHz – Titan V. NVIDIA

MATLAB interoperates with other frameworks

Supports ONNX and can exchange models with PyTorch, TensorFlow, and other frameworks.



Modeling Demo



Open Script
Part 2

AI Modeling

 Model design and tuning

 Hardware accelerated training

 Interoperability

| True Class | Bearing | Flywheel | Healthy | LIV | LOV | NRV | Piston | Riderbelt |
|-----------------|---------|----------|---------|-----|-----|-----|--------|-----------|
| | 20 | 4 | | | | | 1 | |
| | 2 | 18 | | | | | | |
| | | | 21 | | | | | |
| | | | | 19 | 1 | 1 | | |
| | | | 1 | 2 | 19 | 1 | | |
| | | | | 1 | 1 | 19 | 1 | |
| | | | | | | 1 | 20 | |
| Predicted Class | | | | | 1 | | | 22 |
| | Bearing | Flywheel | Healthy | LIV | LOV | NRV | Piston | Riderbelt |

Models need to exist within a complete system

In automated driving systems, AI for perception must integrate with algorithms for path planning, braking, acceleration, and other controls.

Simulation & Test

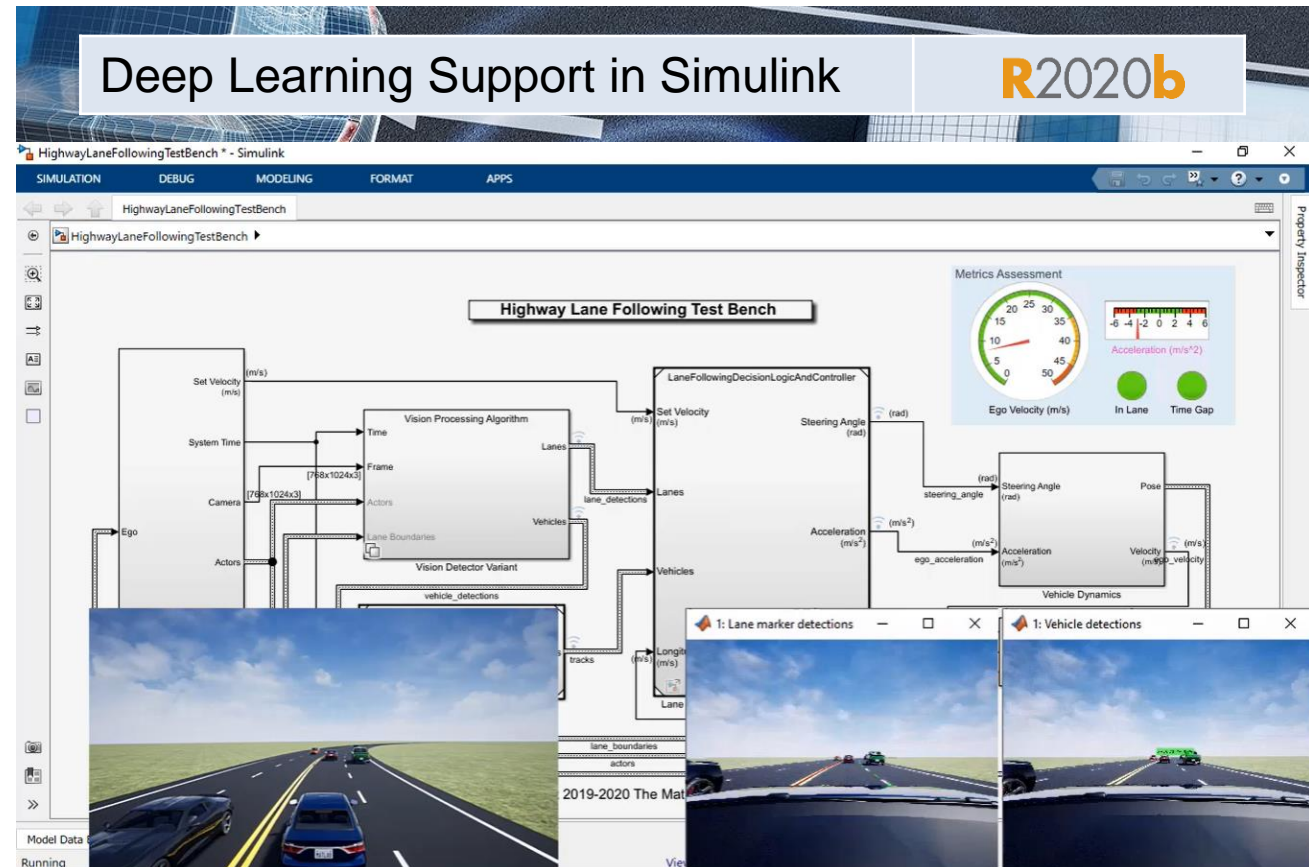


Integration with complex systems



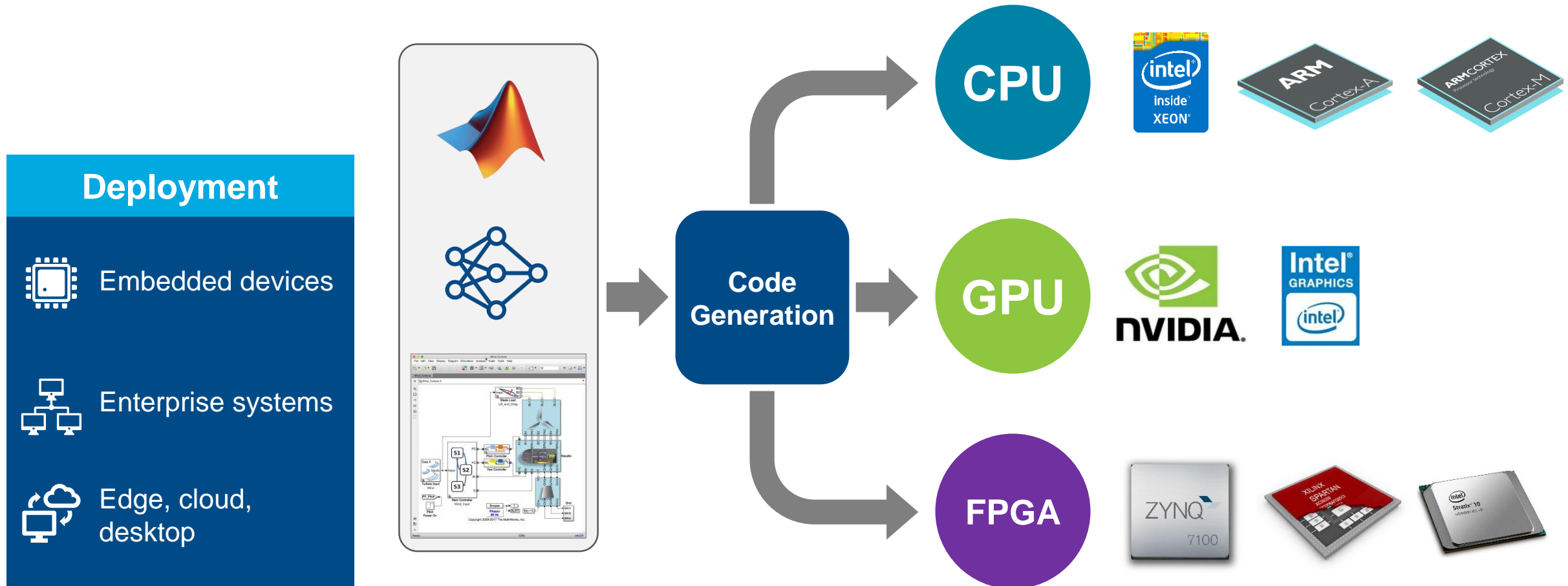
System simulation

- ✗ System verification
- ✓ and validation

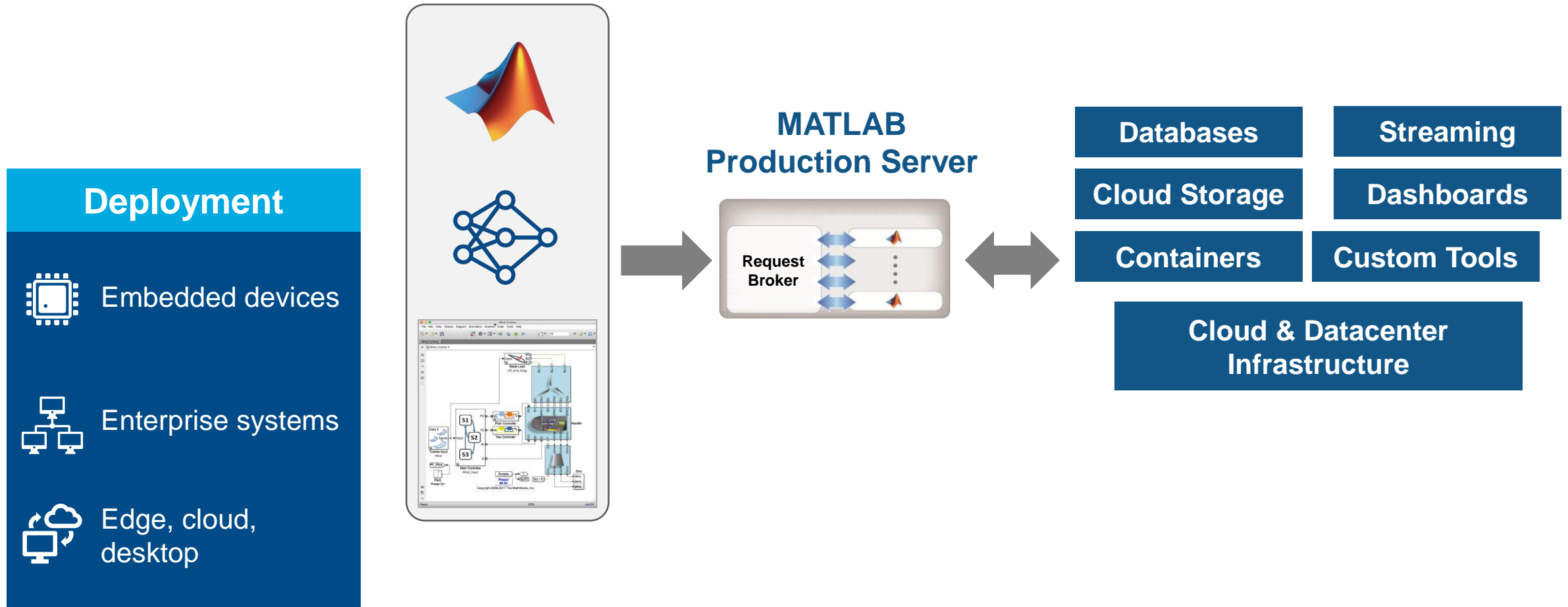


Deploy to any processor with best-in-class performance

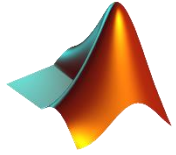
AI models in MATLAB and Simulink can be deployed on embedded devices, edge devices, enterprise systems, the cloud, or the desktop.



Deploy to enterprise IT infrastructure



Deployment Demo



Open Script
Part 3

Deployment



Embedded devices



Enterprise systems



Edge, cloud,
desktop

The screenshot displays the MATLAB Coder interface. The left pane shows the 'MATLAB SOURCE' section with a 'Function List' containing 'yolov2_detect.m' and 'cudnnApi.p'. The 'GENERATED CODE' section shows 'Source Files' including 'DeepLearningNetwork', 'MWAdditionLayer.c', 'MWAdditionLayer.h', 'MWAdditionLayer.m', 'MWBatchNormaliza', and 'MWBatchNormaliza'. The main editor shows the 'Function: yolov2_detect' with the following code:

```
1 function outImg = yolov2_detect(in)
2
3 % Copyright 2018-2019 The MathWorks, Inc.
4 persistent yolov2obj;
5
6 if isempty(yolov2obj)
7     yolov2obj = coder.loadDeepLearningNetwork('Yolov2UsingResNet50_ONNX.mat');
8 end
9
10 % pass in input
11 [bboxes,~,labels] = yolov2obj.detect(in,'Threshold',0.5);
12
13 % convert categorical labels to cell array of character vectors for MATLAB
14 % execution
15 if coder.target('MATLAB')
16     labels = cellstr(labels);
17 end
18
19 % Annotate detections in the image.
20 outImg = insertObjectAnnotation(in,'rectangle',bboxes,labels);
21
22
```

The bottom pane shows the 'SUMMARY' tab with the message 'Code generation successful'. The generated code details are:

- Generated on: 17-Sep-2019 14:21:46
- Build type: MEX Function
- Output file: C:\Users\shmitra\Work\Deep_Learning\Seminar\19b\ResNetImportYolov2\HelperFilesAndFunctions\yolov2_detect_mex.mexw64
- Processor: Generic->MATLAB Host Computer



Agenda

- I. Deep learning in engineering and science
- II. Developing a deep learning solution in MATLAB
- III. MathWorks deep learning support



MathWorks is your Deep Learning partner



The Platform

MATLAB, Simulink, and over 100 add-on products for specialized applications



Your People

Helping you build an agile workforce today and preparing tomorrow's engineers



Our Expertise

From onboarding and implementation to solving advanced engineering challenges

MathWorks Engineering Support



Training



Guided Evaluations



Onsite Workshops



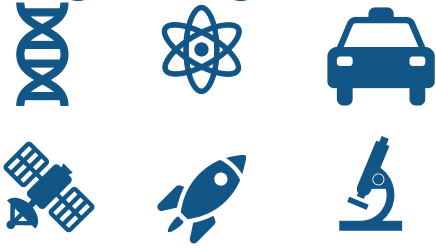
Consulting



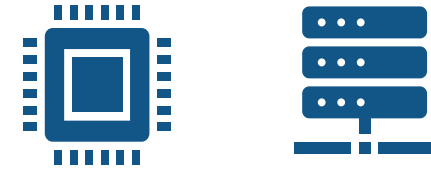
Technical Support

Why MATLAB & MathWorks for Deep Learning?

Domain-specialized workflows
for **engineering and science**



Multi-platform **deployment** of
full applications and systems



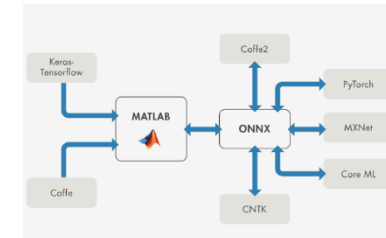
People



Platform productivity

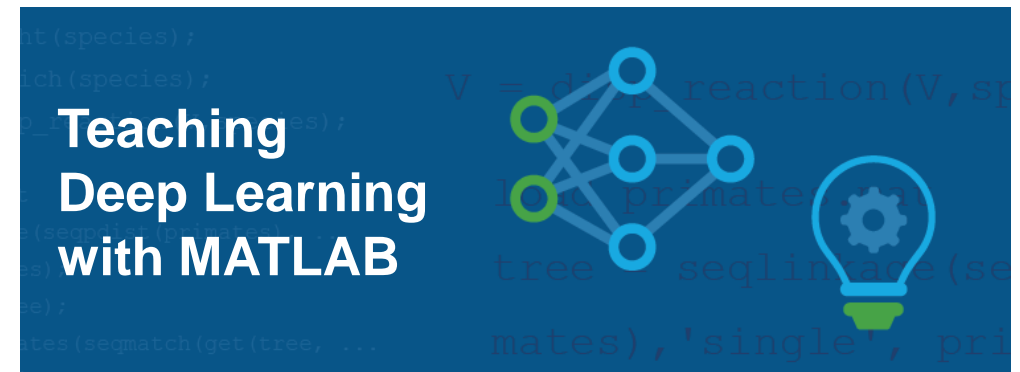
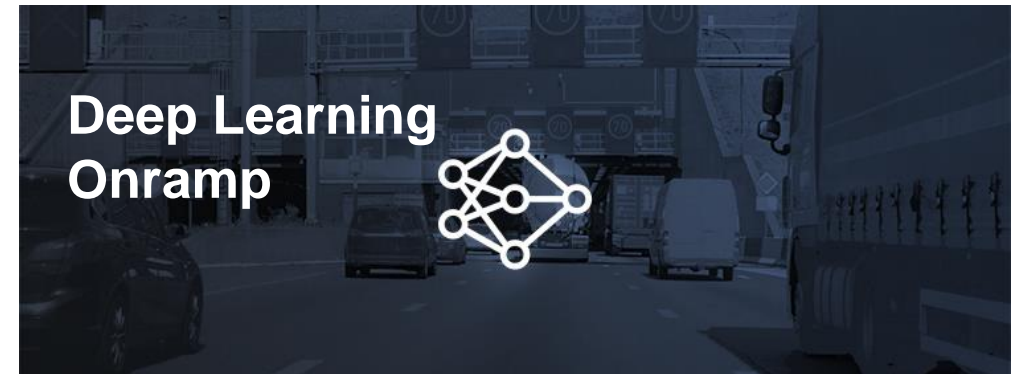


Interoperability with
TensorFlow and PyTorch



Further Learning & Teaching

- [Deep Learning Onramp](#)
 - 2 hr online tutorial
- Deep Learning Workshop
 - 3 hr hands on session
 - Contact us to schedule
- [Deep Learning Training](#)
 - 16 hr in depth course
 - Online or Instructor Lead
- [Teaching Deep Learning with MATLAB](#)
 - Curriculum support



Where to find this content on GitHub

■ <https://tinyurl.com/deeplearningmatlabsignal>



Learn Git and GitHub without any code!
Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

matlab-deep-learning / Fault-Detection-Using-Deep-Learning-Classification

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

This demo shows the full deep learning workflow for an example of signal data. We show how to prepare, model, and deploy a deep learning LSTM based classification algorithm to identify the condition or output of a mechanical air compressor. [Edit](#)

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor View license

Branch: master New pull request Create new file Upload files Find file [Clone or download](#)

| File | Initial push | Latest commit |
|---------------------------------|--------------|----------------|
| HelperFiles | initial push | 10 minutes ago |
| Images | initial push | 10 minutes ago |
| SavedData | initial push | 10 minutes ago |
| resources/project | initial push | 10 minutes ago |
| .gitignore | initial push | 10 minutes ago |
| AirCompressorDataset.rights | initial push | 10 minutes ago |
| AirCompressorClassification.prj | initial push | 10 minutes ago |
| LICENSE | initial push | 10 minutes ago |
| Part01_DataPreparation.mlx | initial push | 10 minutes ago |
| Part02_Modeling.mlx | initial push | 10 minutes ago |
| Part03_Deployment.mlx | initial push | 10 minutes ago |
| README.md | initial push | 10 minutes ago |

README.md

Fault Detection Using LSTM Deep Learning Classification

This demo shows the full deep learning workflow for an example of signal data. We show how to prepare, model, and deploy a deep learning LSTM based classification algorithm to identify the condition or output of a mechanical air compressor.

Class: Bearing

Amplitude

Sample number $\times 10^4$