MATLAB EXPO 2019

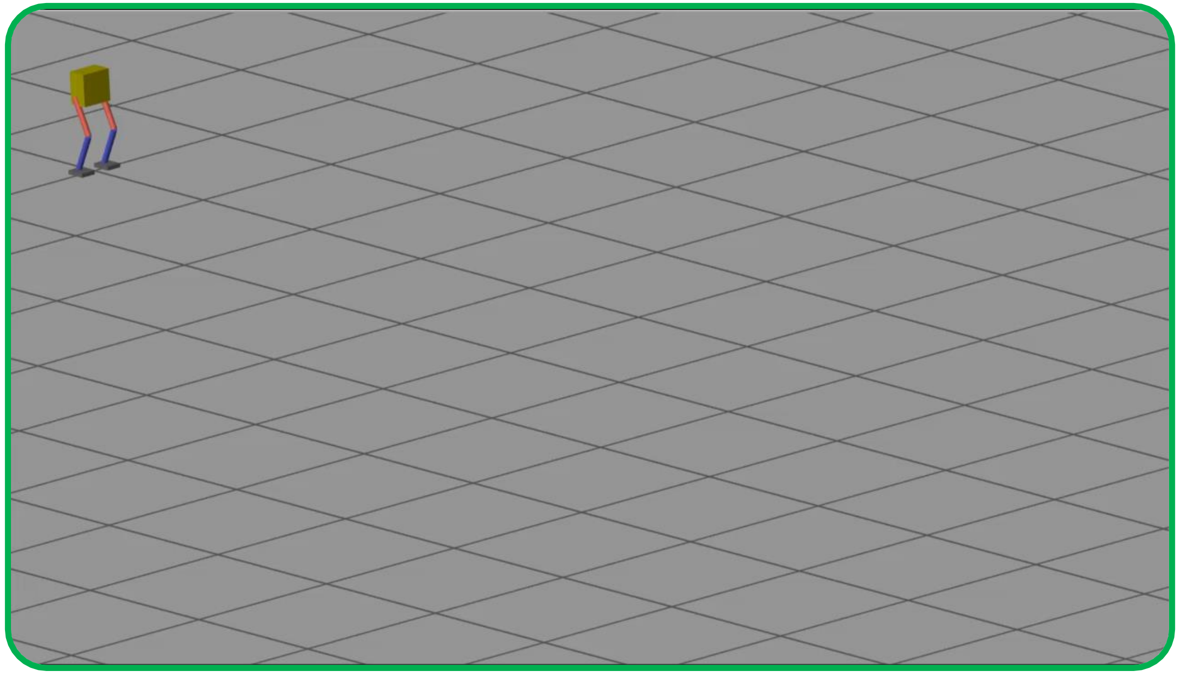
Deep Learning and Reinforcement Learning Workflows in A.I.

Avinash Nehemiah
Principal Product Manager
Deep Learning, Computer Vision, Automated Driving

Why MATLAB for Artificial Intelligence?

Artificial Intelligence

Development of computer systems to perform tasks that normally require human intelligence

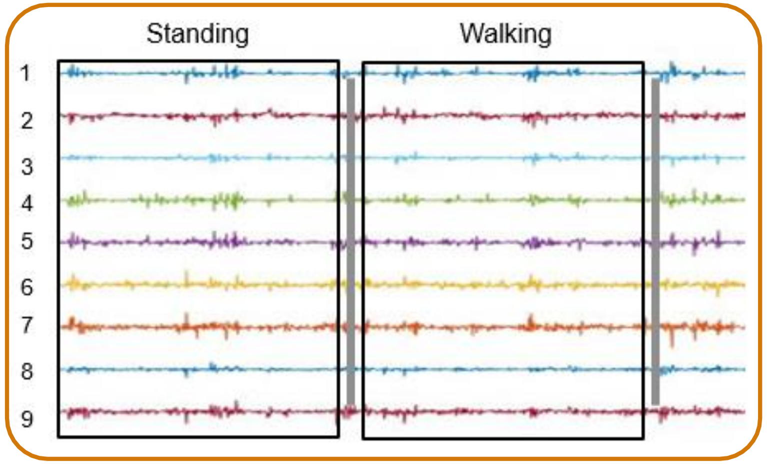


A.I. Applications

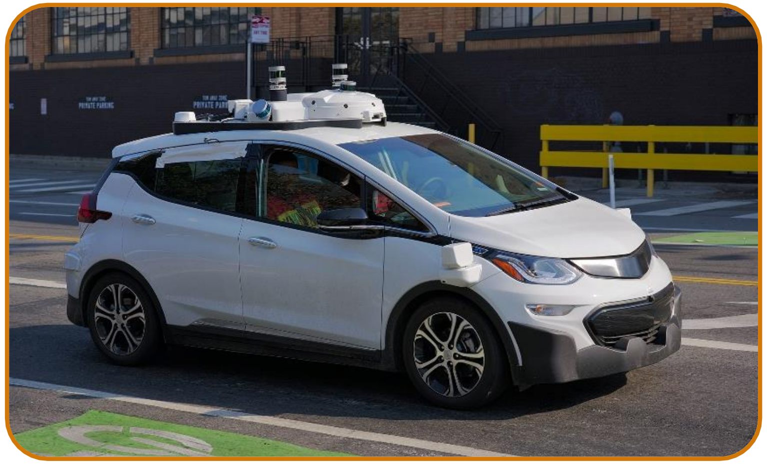
Object Classification

Speech Recognition

Predictive Maintenance



Signal Classification



Automated Driving

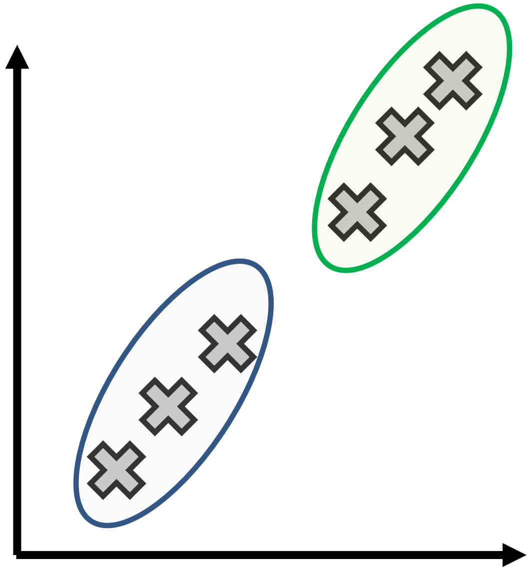
Stock Market Prediction

Machine Learning and Deep Learning

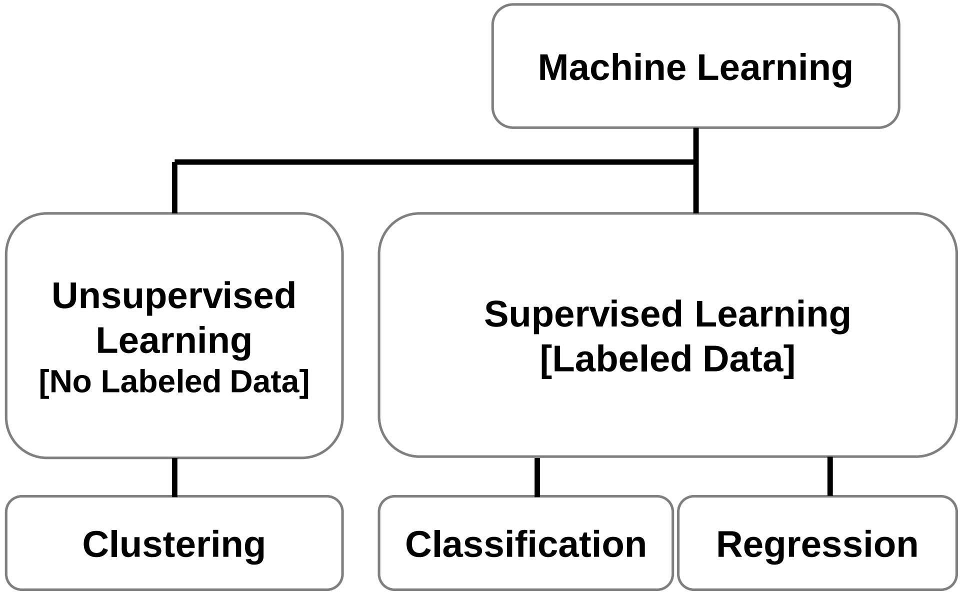
Machine Learning

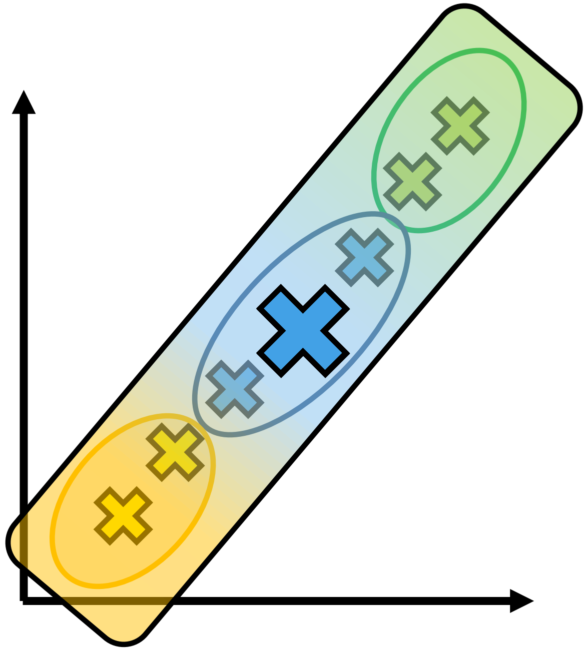
Unsupervised Learning
[No Labeled Data]

Clustering

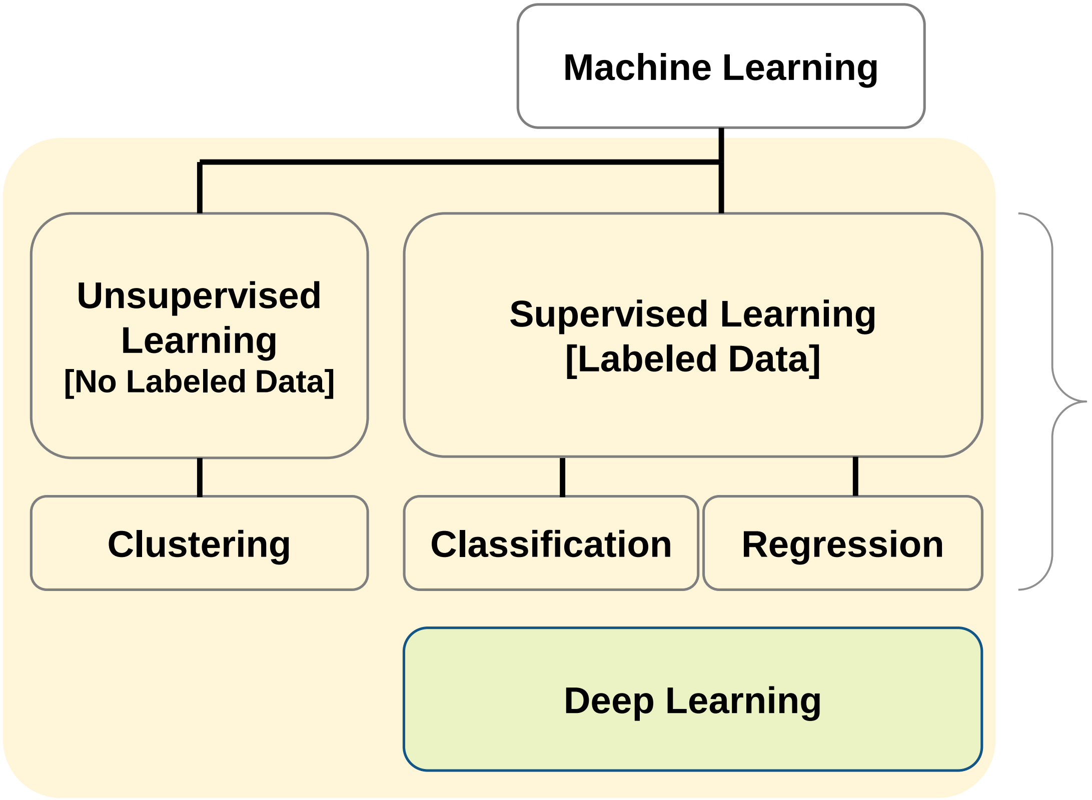


Machine Learning and Deep Learning



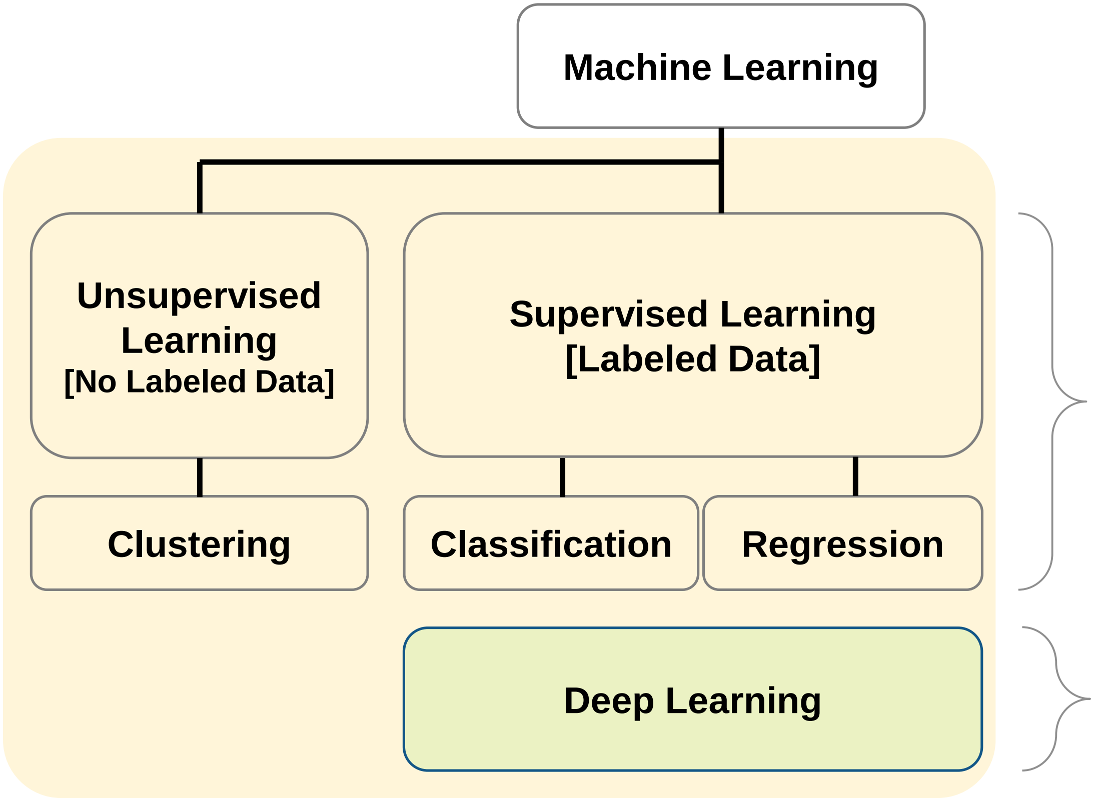


Machine Learning and Deep Learning



Supervised learning typically involves feature extraction

Machine Learning and Deep Learning

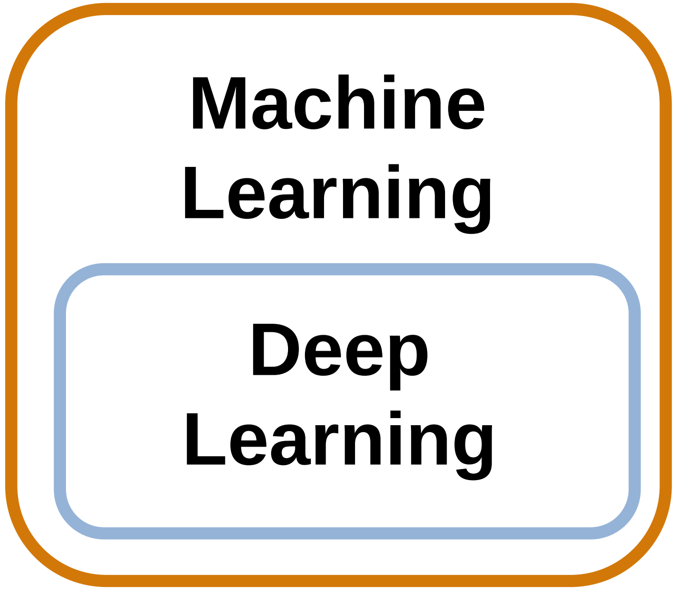


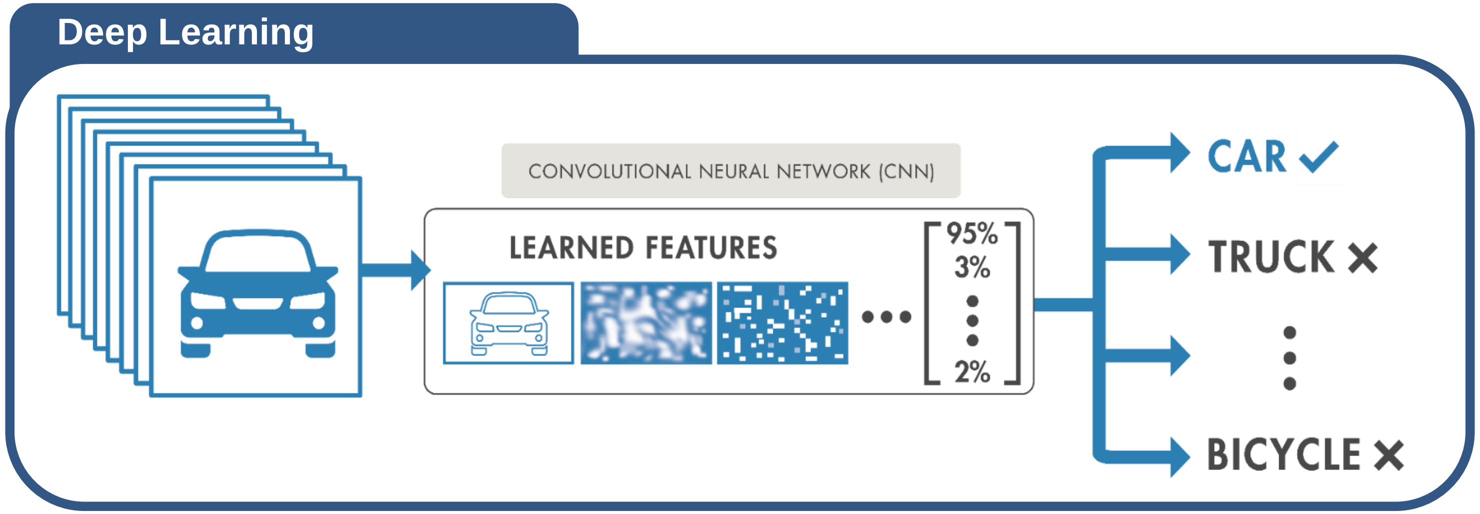
Supervised learning typically involves feature extraction

Deep learning typically does not involve feature extraction

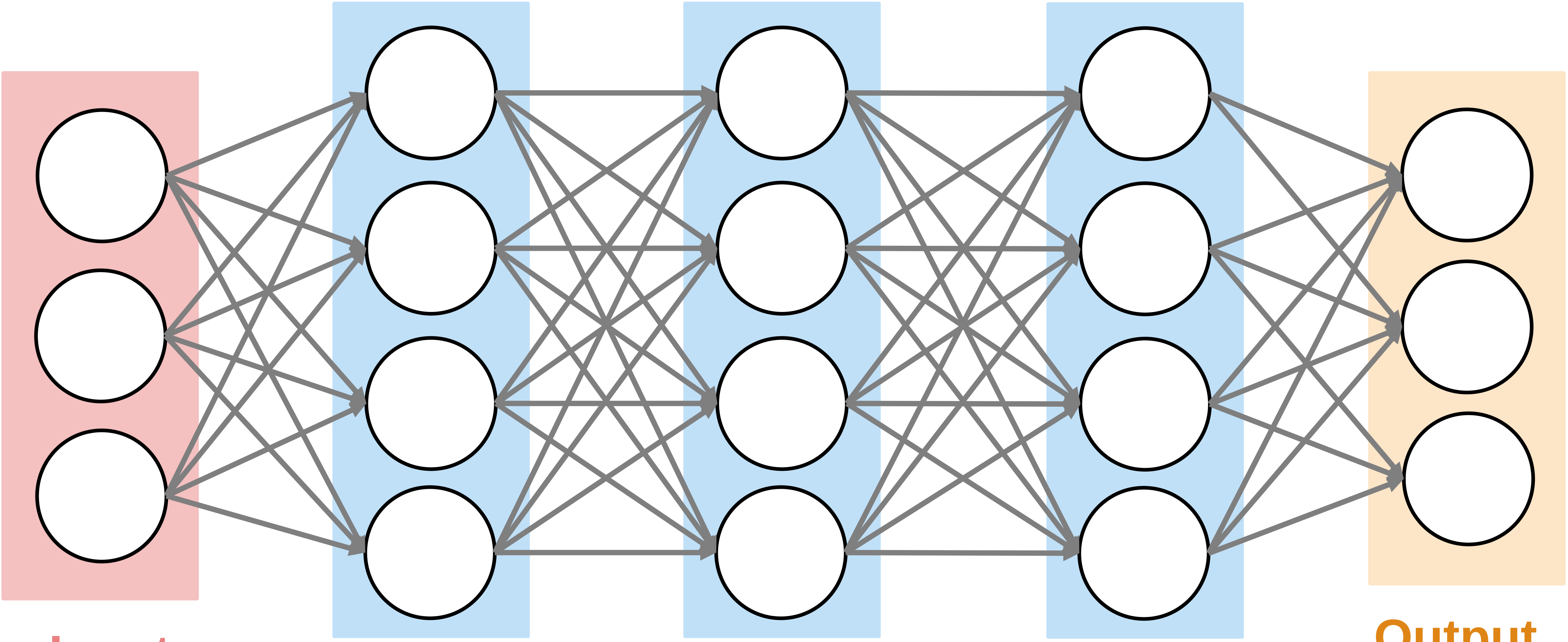
Deep Learning

- Subset of machine learning with automatic feature extraction
  - Learns features and tasks directly from data
  - More Data = better model



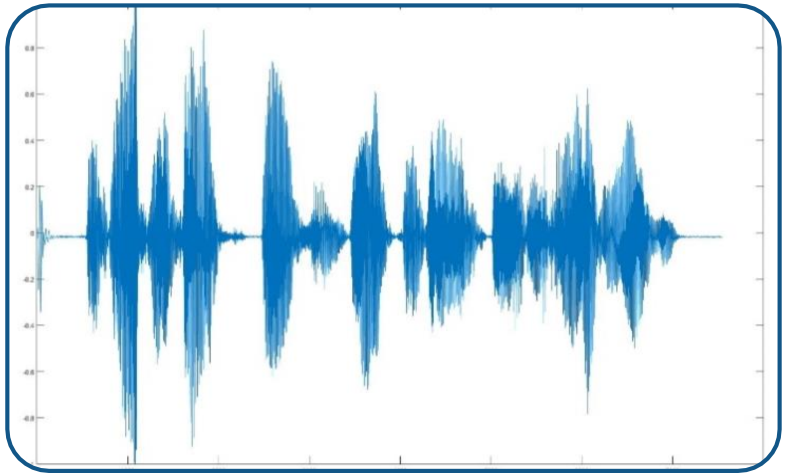


Deep Learning Uses a Neural Network Architecture



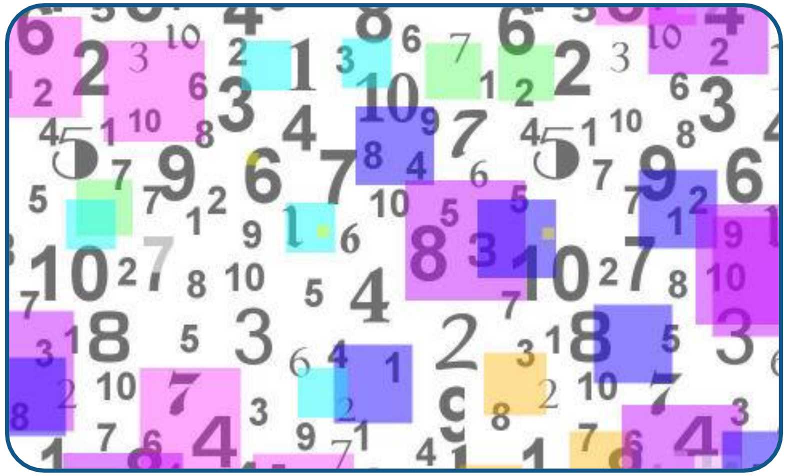
Deep Learning Datatypes

Image



Signal

Numeric



Text

Deep Learning Workflow

Prepare Data

- Data access and preprocessing
- Ground truth labeling

Train Model

- Model design, Hyperparameter tuning
- Model exchange across frameworks
- Hardware-accelerated training

Deploy

- Multiplatform code generation (CPU, GPU)
- Edge deployment
- Enterprise Deployment

Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Generate simulation data for complex models and systems

Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains

Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Labeling

Training

Model Exchange

Full A.I. workflows that cannot be easily replicated by other toolchains

Labeling for deep learning is repetitive, tedious, and time-consuming…

but necessary

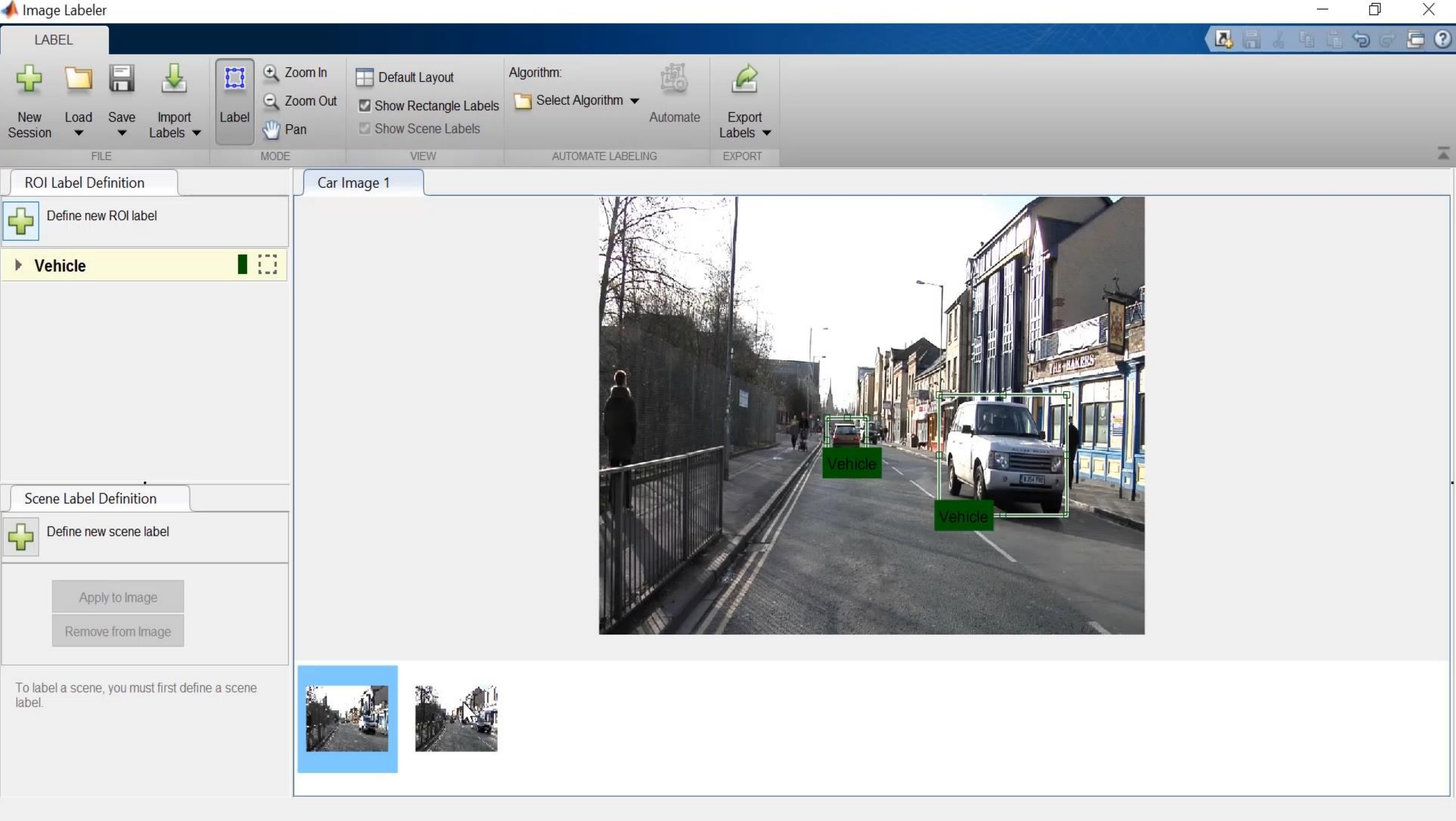
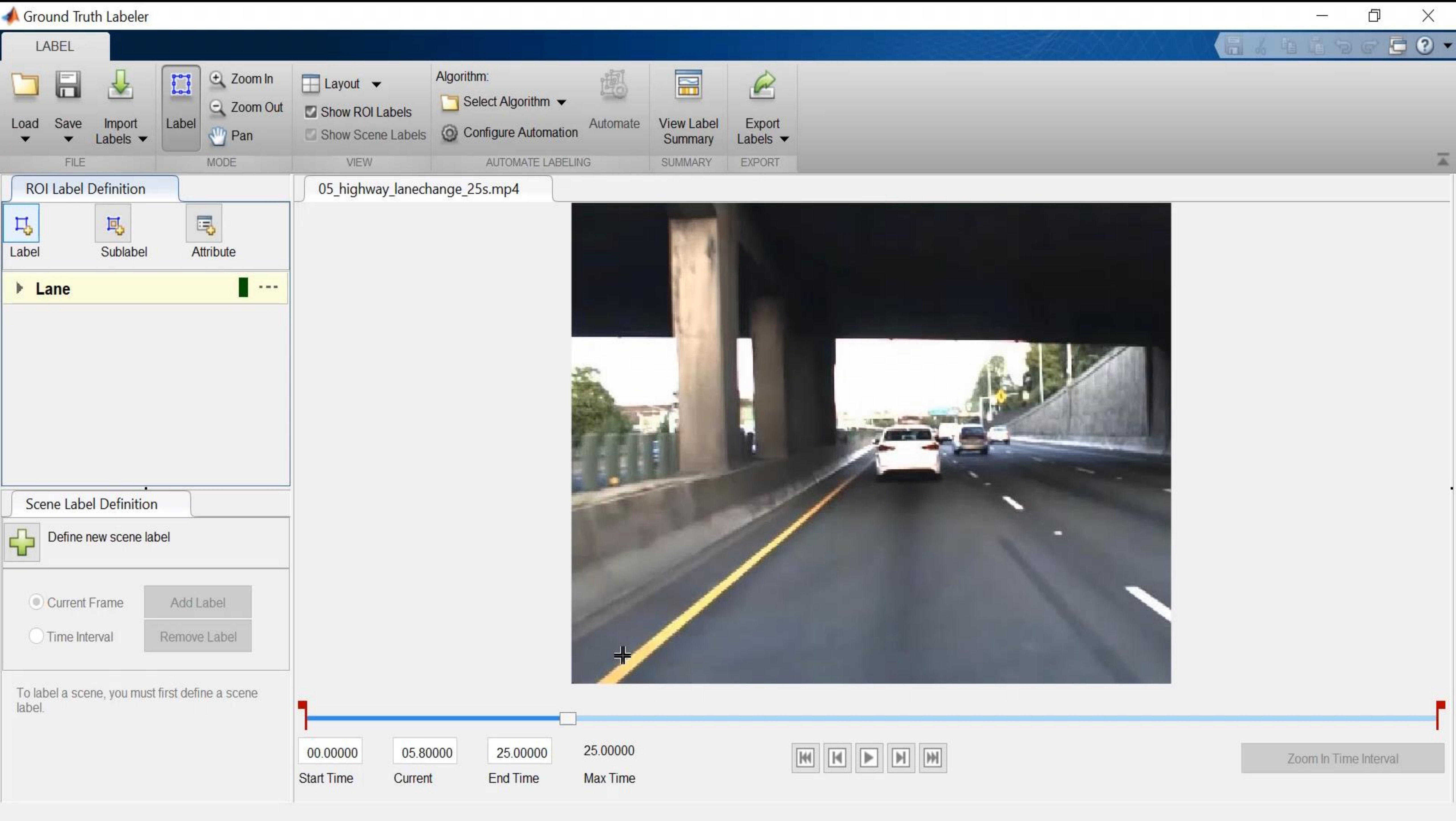
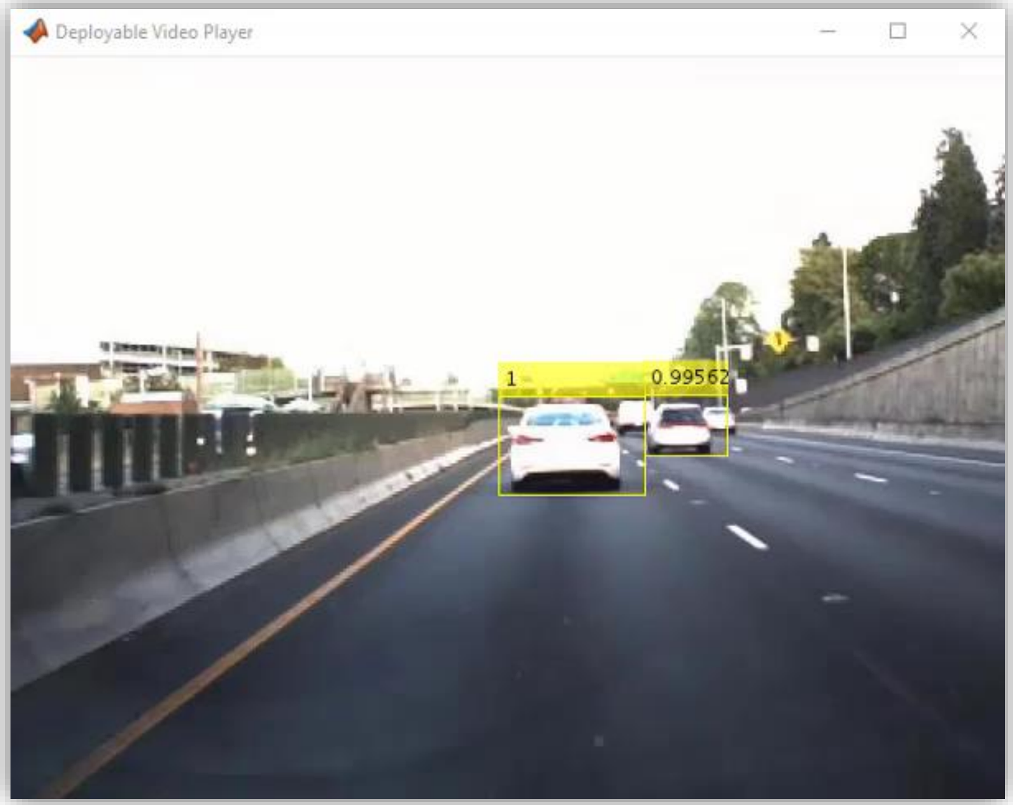


Image Labeler
LABEL
New Session
Load
Save
Import Labels
FILE
Label
Zoom In
Zoom Out
Pan
MODE
Default Layout
Show Rectangle Labels
Show Scene Labels
VIEW
Algorithm:
Select Algorithm
Automate
AUTOMATE LABELING
Export Labels
EXPORT
ROI Label Definition
Define new ROI label
Vehicle
Scene Label Definition
Define new scene label
Apply to Image
Remove from Image
To label a scene, you must first define a scene label.
Car Image 1
Vehicle
Vehicle


Ground Truth Labeler
LABEL
Load
Save
Import Labels
Label
Zoom In
Zoom Out
Pan
Layout
Show ROI Labels
Show Scene Labels
Algorithm:
Select Algorithm
Configure Automation
Automate
View Label Summary
Export Labels
FILE
MODE
VIEW
AUTOMATE LABELING
SUMMARY
EXPORT
ROI Label Definition
Label
Sublabel
Attribute
Lane
Scene Label Definition
Define new scene label
Current Frame
Time Interval
Add Label
Remove Label
To label a scene, you must first define a scene label.
05_highway_lanechange_25s.mp4
00.00000
Start Time
05.80000
Current
25.00000
End Time
25.00000
Max Time
Zoom In Time Interval

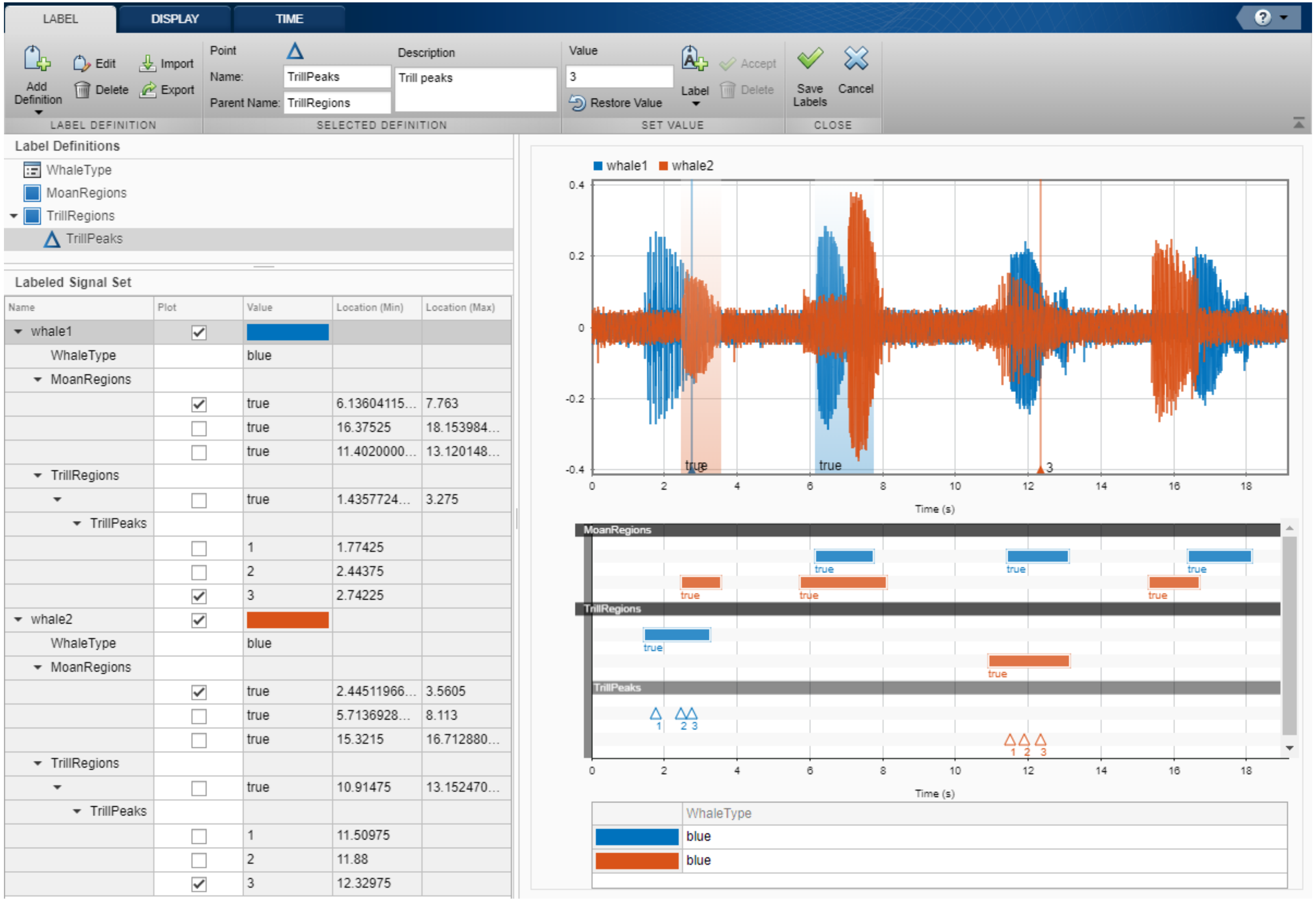

Deployable Video Player
1
0.99562


RICE HOUSE

Signal Labeling – annotate signals with labels/sublabels, export to workspace for training

Define Labels

View properties of labels

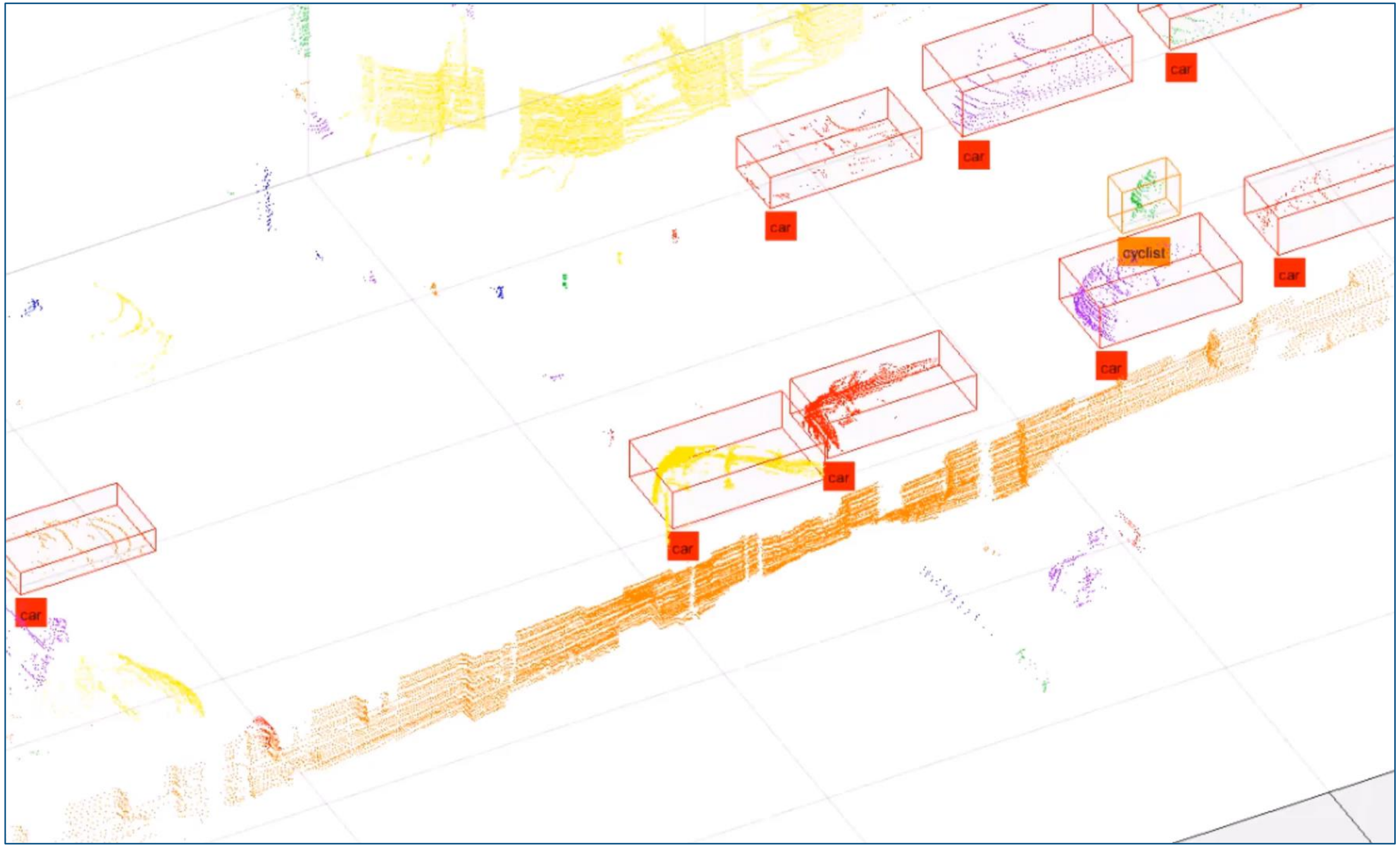


Interactively Label Signals

User Story – Veoneer (Autoliv)

- Automotive
  - Software and hardware for active safety, autonomous driving, occupant protection, and brake control
- Building radar sensor – check accuracy using LiDAR-based verification
- Human analyzes hours of recorded data
- Used MATLAB to semi-automate labeling and tracking of 3D LiDAR point clouds.

Manual Labeling for 25 events took over 20 minutes.
After full automation with MATLAB’s tools, it took 5 minutes

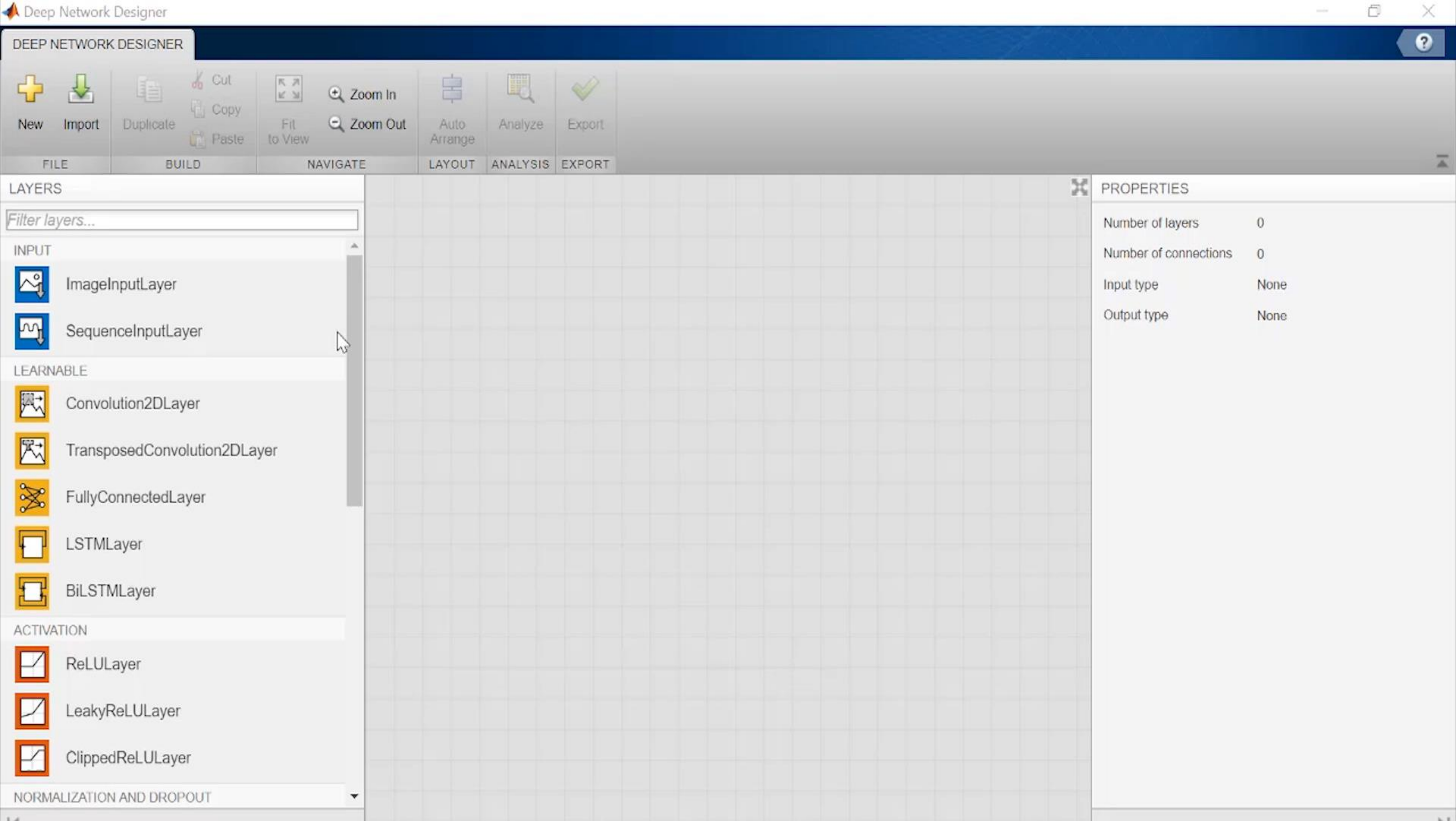


Why MATLAB for A.I. Tasks?

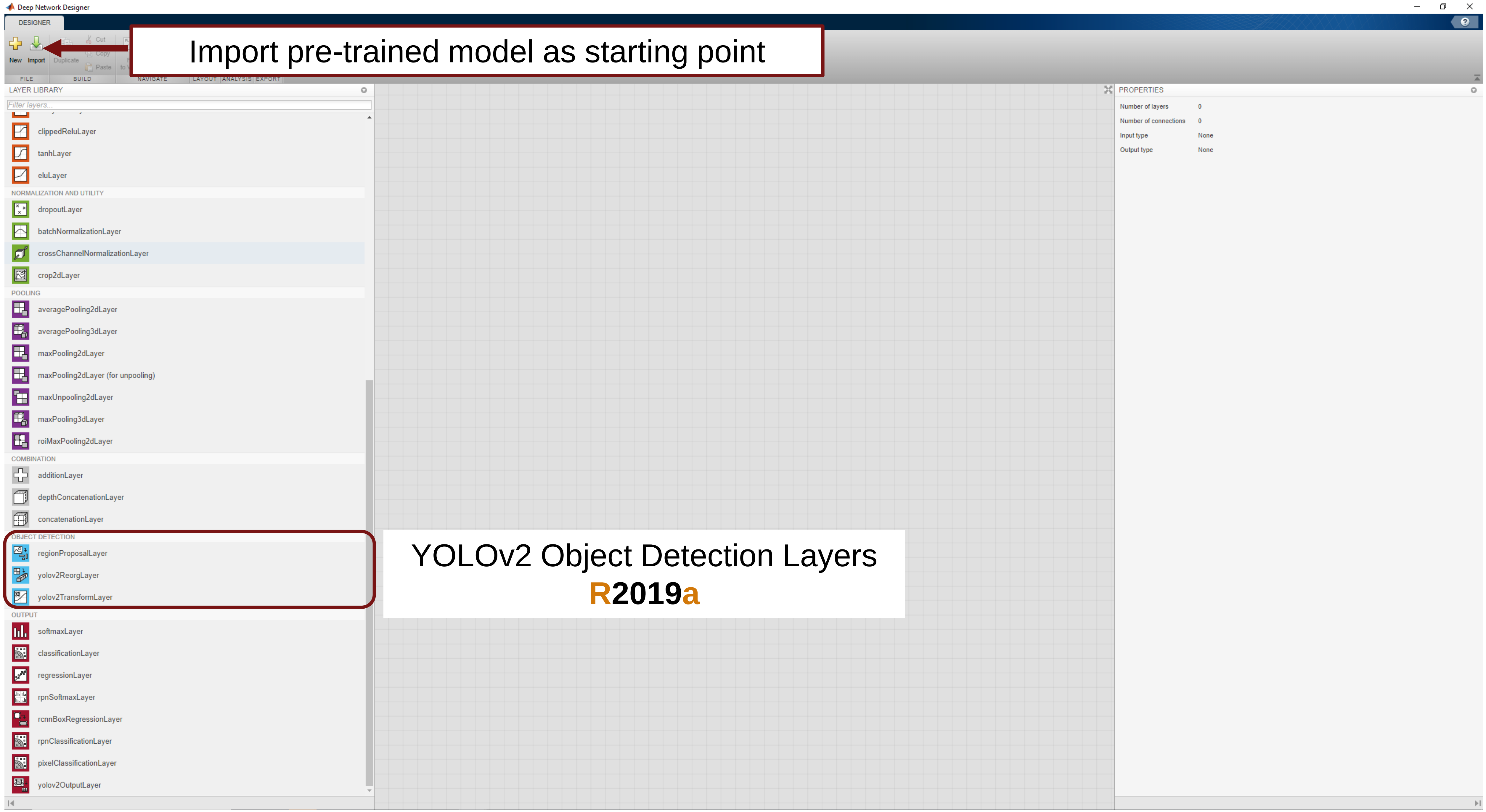
Increased productivity with interactive tools

Labeling | **Training** | **Model Exchange**

Full A.I. workflows that cannot be easily replicated by other toolchains


Deep Network Designer
DEEP NETWORK DESIGNER
New
Import
Duplicate
Cut
Copy
Paste
Fit to View
Zoom In
Zoom Out
Auto Arrange
Analyze
Export
FILE
BUILD
NAVIGATE
LAYOUT
ANALYSIS
EXPORT
LAYERS
Filter layers...
INPUT
ImageInputLayer
SequenceInputLayer
LEARNABLE
Convolution2DLayer
TransposedConvolution2DLayer
FullyConnectedLayer
LSTMLayer
BiLSTMLayer
ACTIVATION
ReLULayer
LeakyReLULayer
ClippedReLULayer
NORMALIZATION AND DROPOUT
PROPERTIES
Number of layers 0
Number of connections 0
Input type None
Output type None

Import pre-trained model as starting point



YOLOv2 Object Detection Layers
R2019a

Import Pre-trained Models for Transfer Learning

- Inception-v3
- ResNet-101
- VGG-16
- Inception-ResNet-v2
- ResNet-18
- GoogLeNet
- DenseNet-201
- VGG-19
- SqueezeNet
- AlexNet
- ResNet-50
- Places365-GoogLeNet
- MobileNet-v2
- Xception

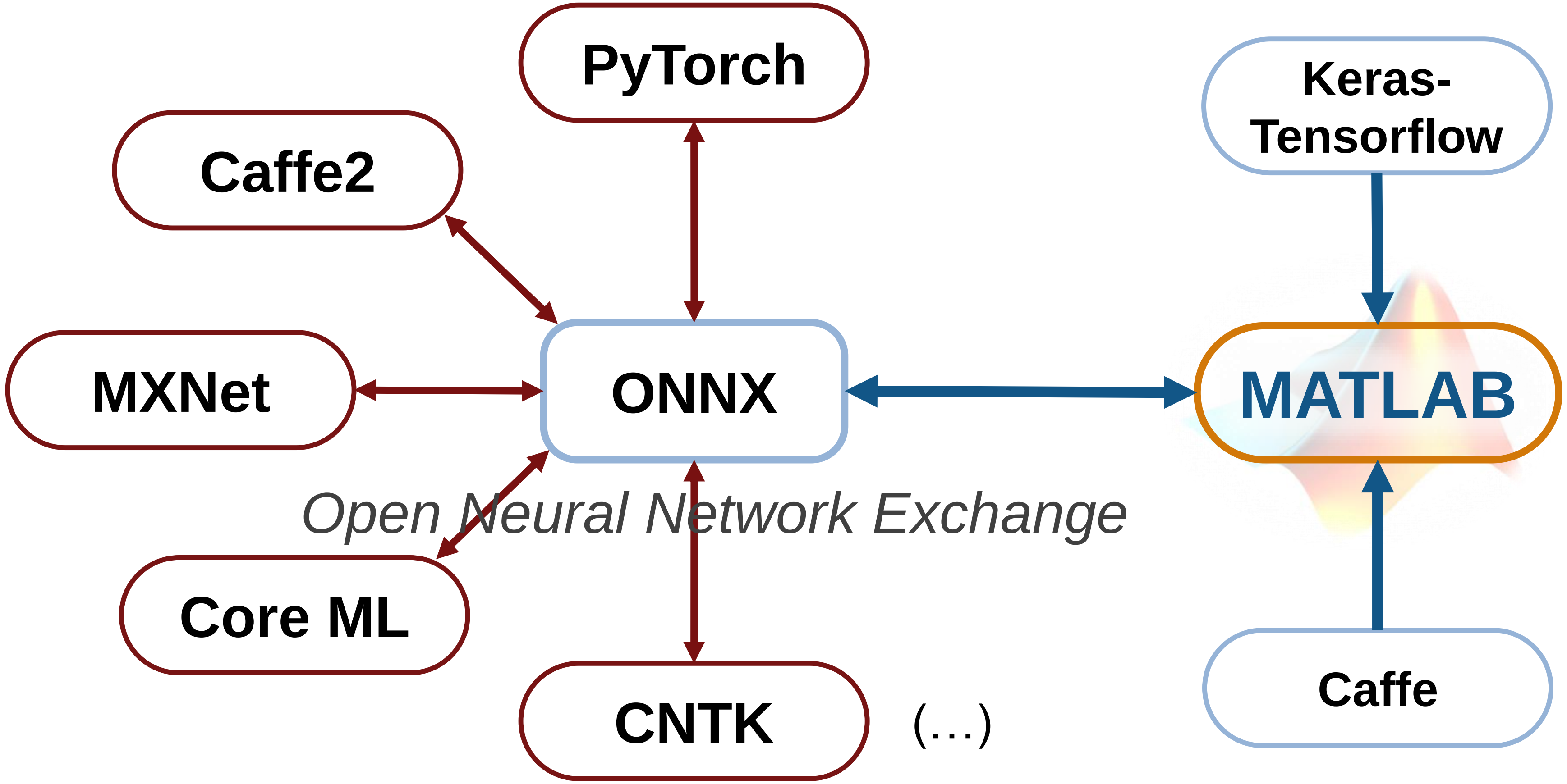
Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Labeling | **Training** | **Model Exchange**

Full A.I. workflows that cannot be easily replicated by other toolchains

Model Exchange with MATLAB



Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Generate simulation data for complex models and systems

Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains

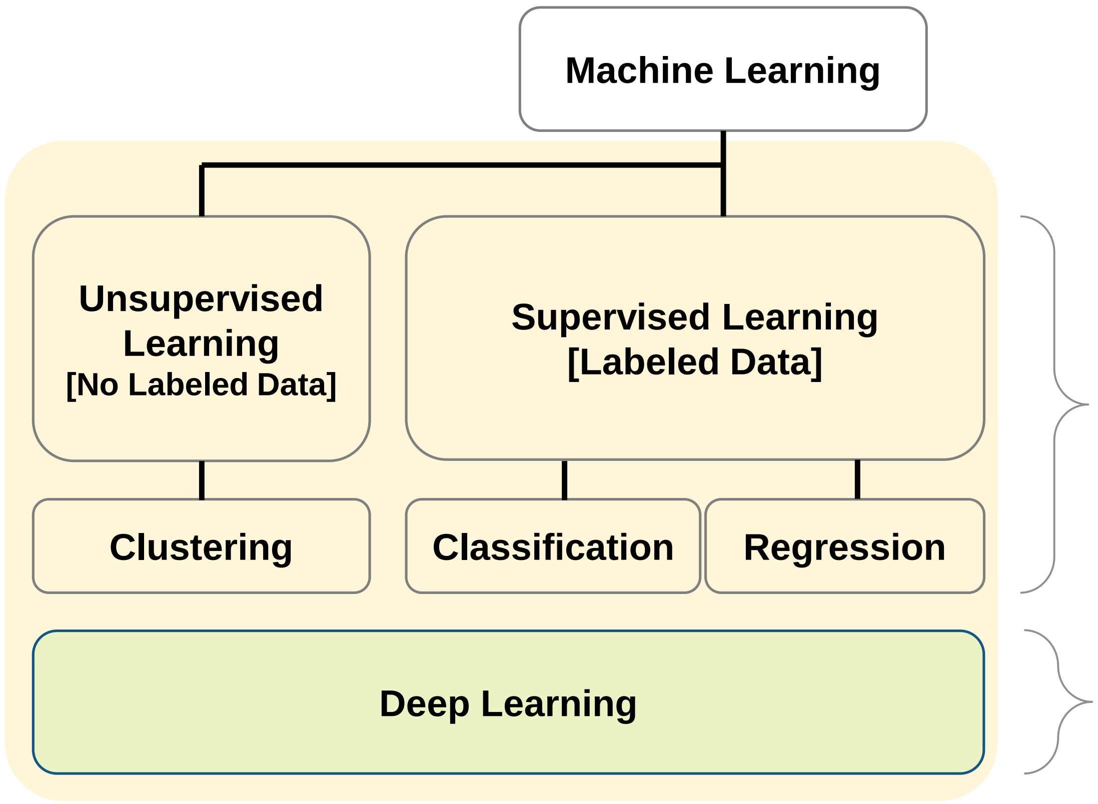
Why MATLAB for A.I. Tasks?

Generate simulation data for complex models and systems

Reinforcement Learning

Full A.I. workflows that cannot be easily replicated by other toolchains

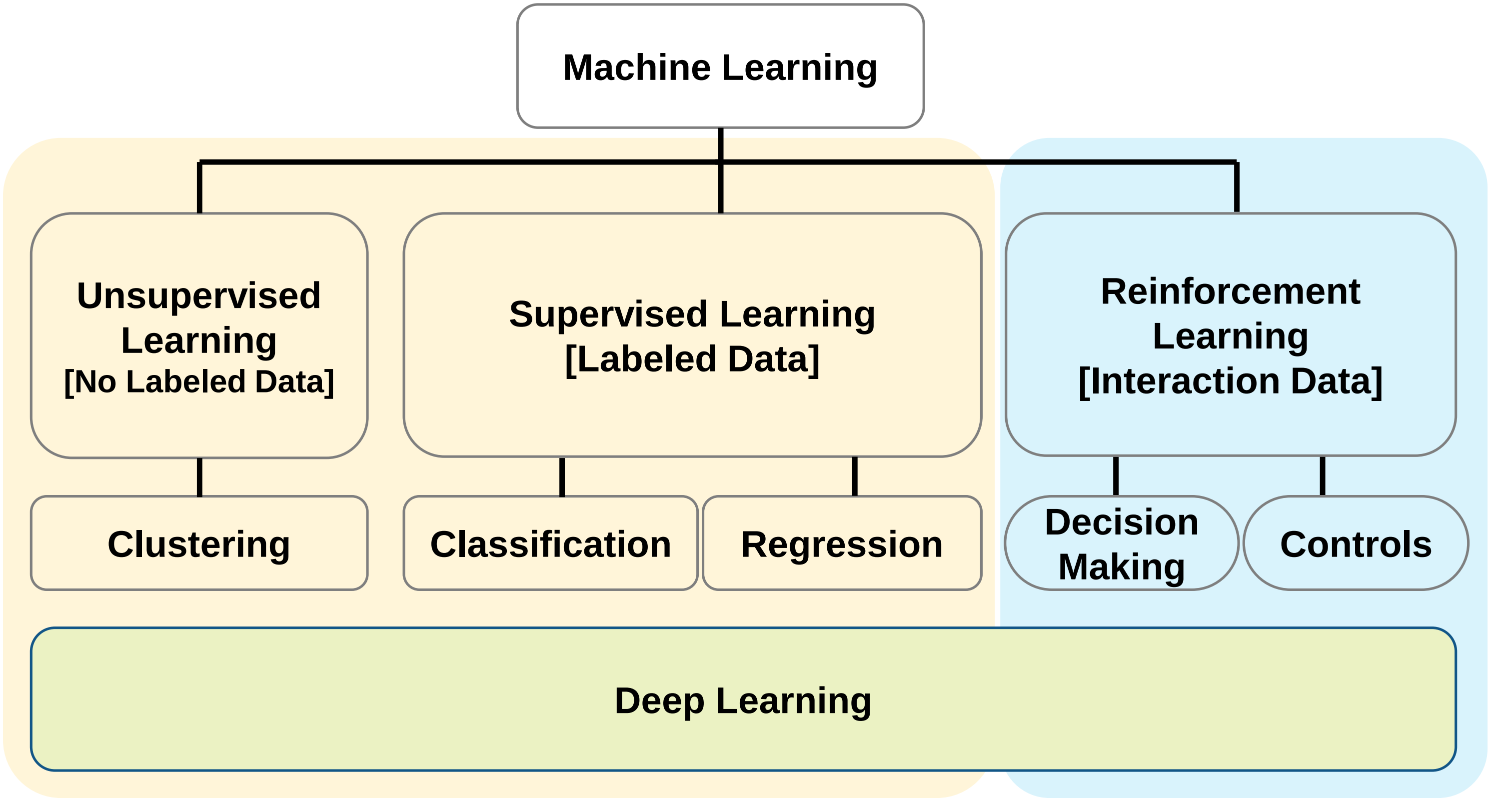
Reinforcement Learning vs Machine Learning vs Deep Learning



Supervised learning typically involves feature extraction

Deep learning typically simplifies feature extraction

Reinforcement Learning vs Machine Learning vs Deep Learning



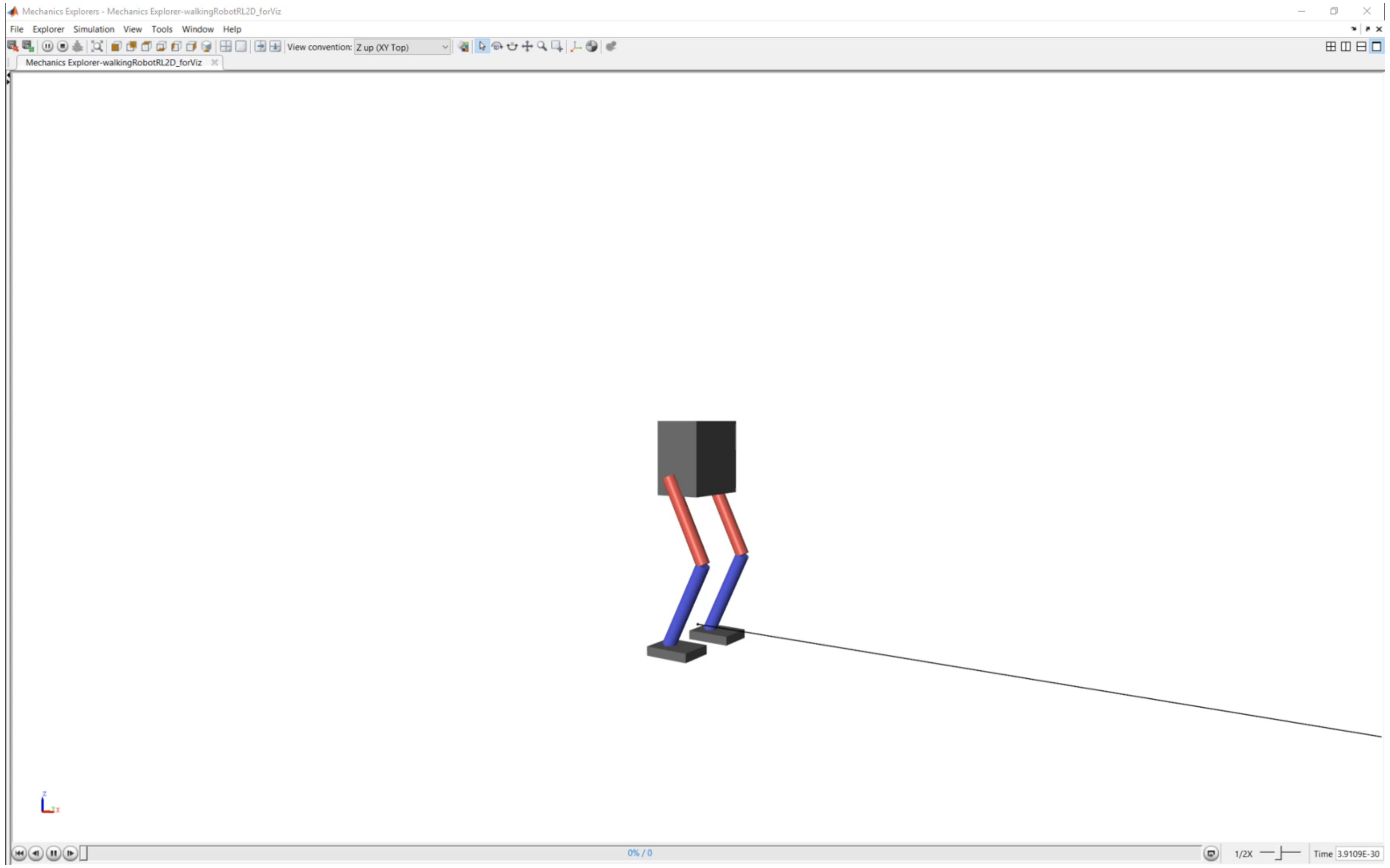
Reinforcement learning:

- Learning through trial & error [*interaction*]
- It's about learning a **behavior** or accomplishing a **task**

What is Reinforcement Learning?

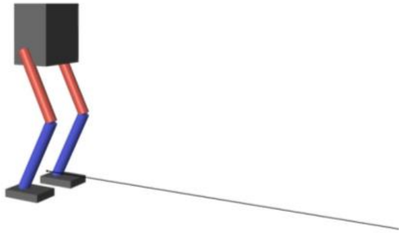
- What is Reinforcement Learning?
  - Type of machine learning that trains an ‘**agent**’ through repeated interactions with an environment

- How does it work?
  - Through a trial & error process that uses a reward system to maximize success

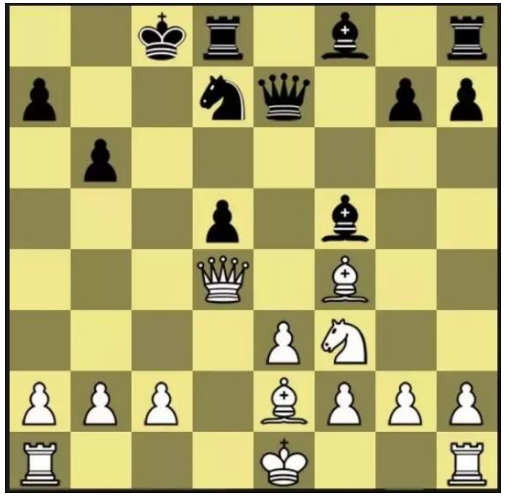


Reinforcement Learning enables the use of Deep Learning for Controls and Decision Making Applications

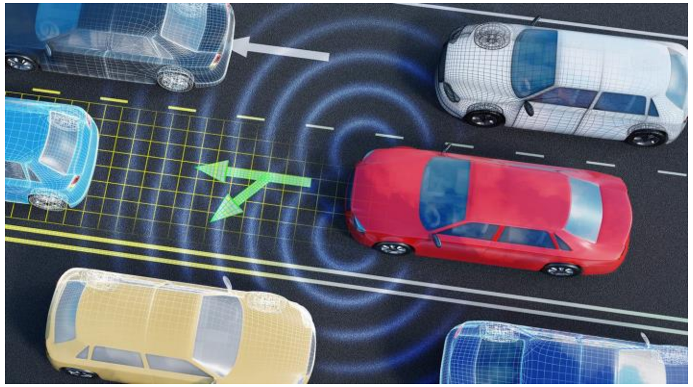
Controls



Robotics

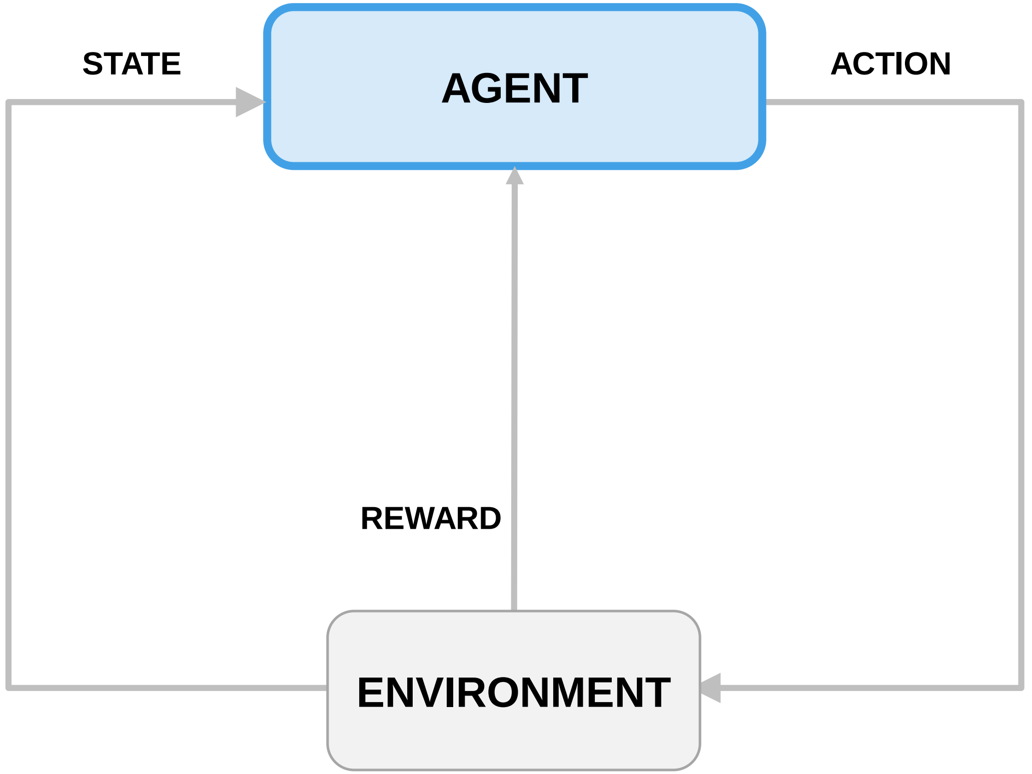


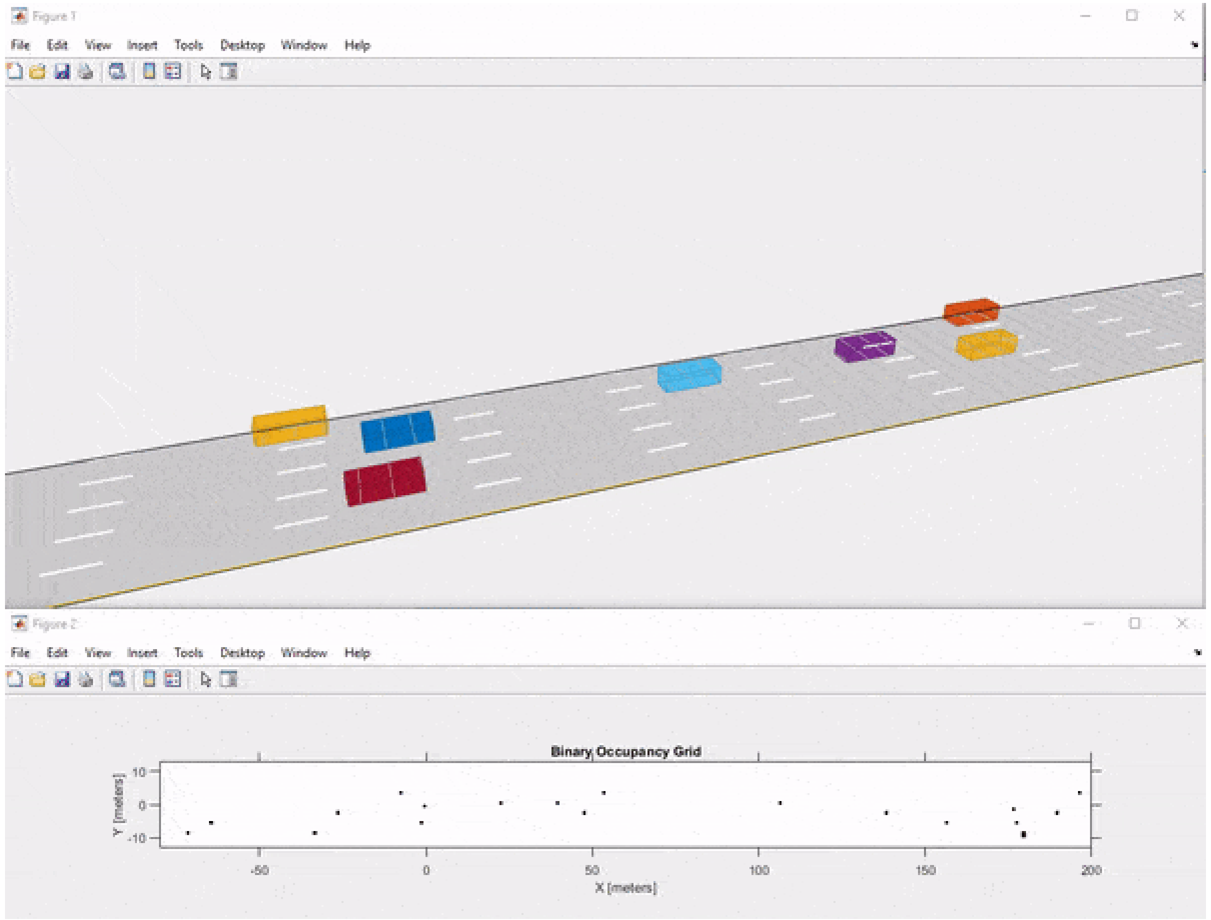
A.I. Gameplay



Autonomous driving

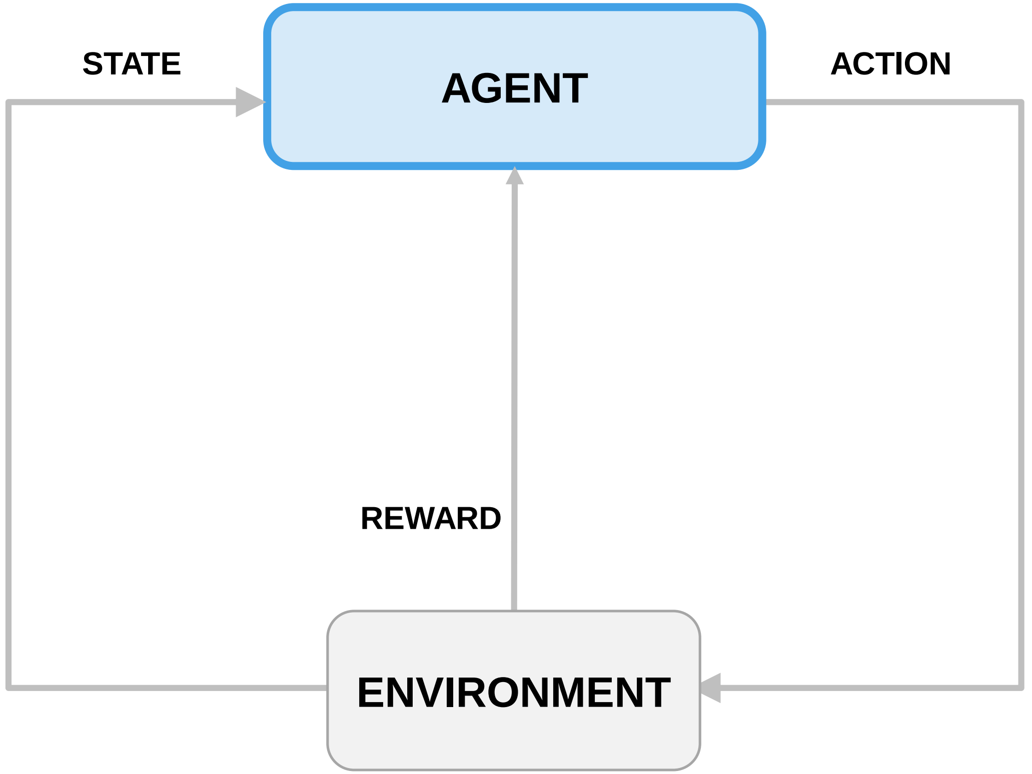
How Does Reinforcement Learning Work?





A Practical Example of Reinforcement Learning

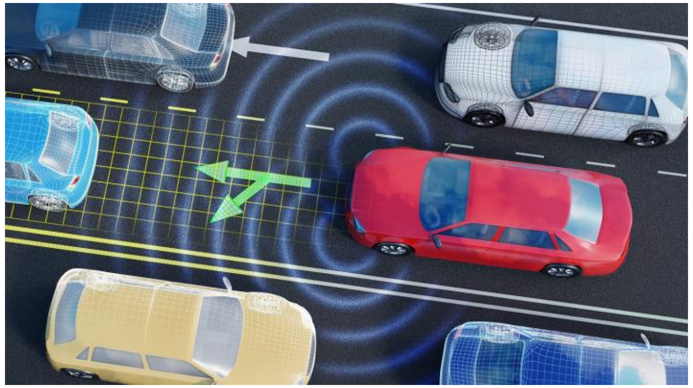
Training a Self-Driving Car



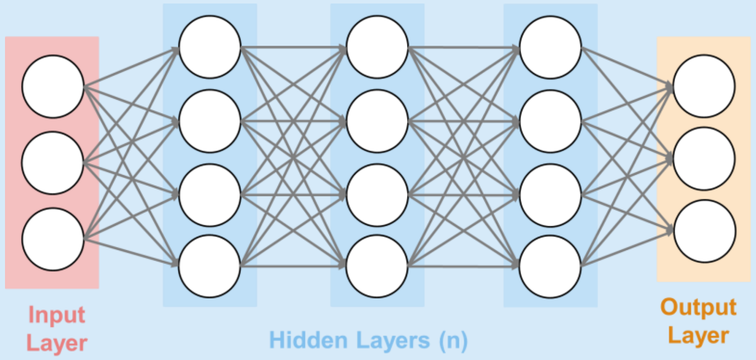
- Vehicle’s computer learns how to drive… (**agent**)
- using sensor readings from LIDAR, cameras,… (**state**)
- that represent road conditions, vehicle position,… (**environment**)
- by generating steering, braking, throttle commands,… (**action**)
- to avoid collisions and lane deviation… (**reward**).

The goal of Reinforcement learning is for the agent to find an optimal algorithm for performing a task

Deep Networks are commonly found in the agent, because they can model complex problems.



AGENT



- Turn left
- Turn right
- Brake
- Accelerate

Reinforcement Learning Workflow

Prepare Data

Data access and preprocessing

Ground truth labeling

Simulink – generate data for dynamic systems (planes, cars, robots, etc.)

Train Model

Reinforcement learning

Training agent to perform task

Developing reward system to optimize performance

Deployment

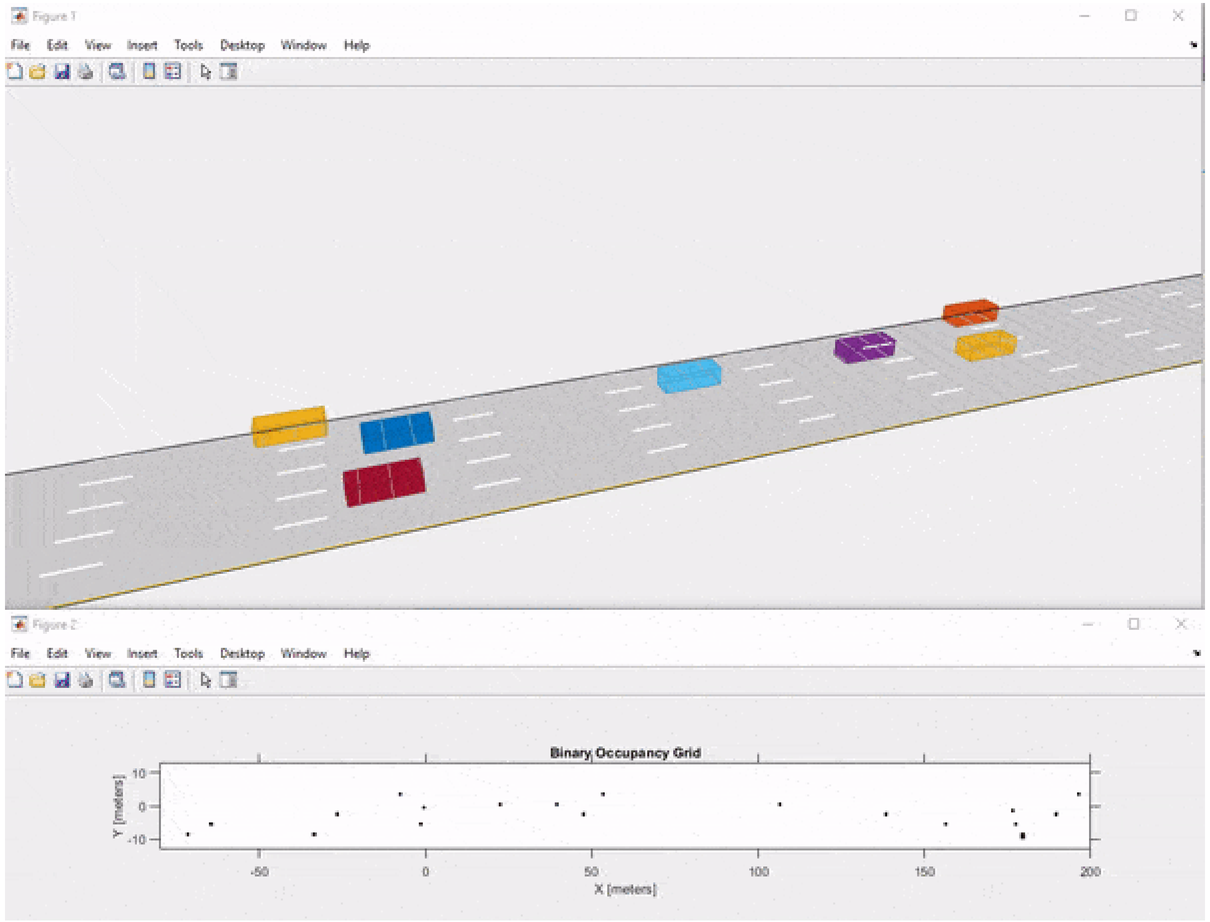
Multiplatform code generation (CPU, GPU)

Edge deployment

Enterprise Deployment

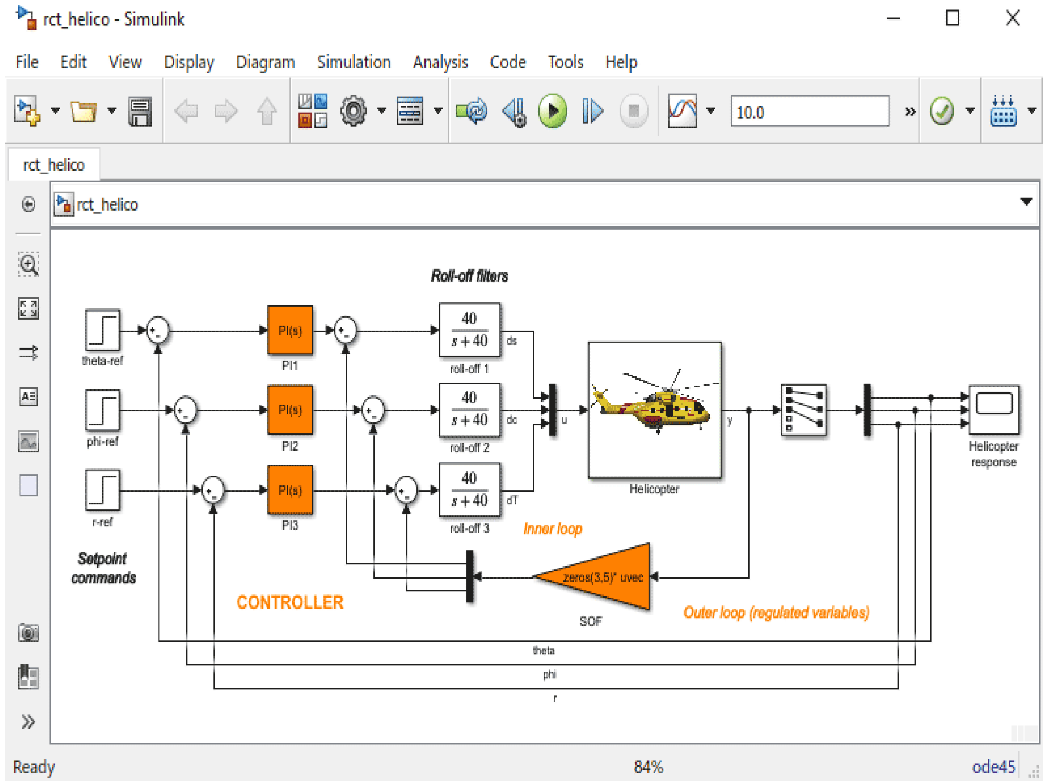
Why MATLAB and Simulink for Reinforcement Learning?

Virtual models allow you to simulate conditions hard to emulate in the real world.



Using MATLAB and Simulink for Reinforcement Learning

- Reinforcement learning is a dynamic process
- Decision making problems
  - Financial trading, calibration, etc.
- Controls-based problems
  - Lane-keep assist, adaptive cruise control, robotics, etc.



Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Generate simulation data for complex models and systems

Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains

Why MATLAB for A.I. Tasks?

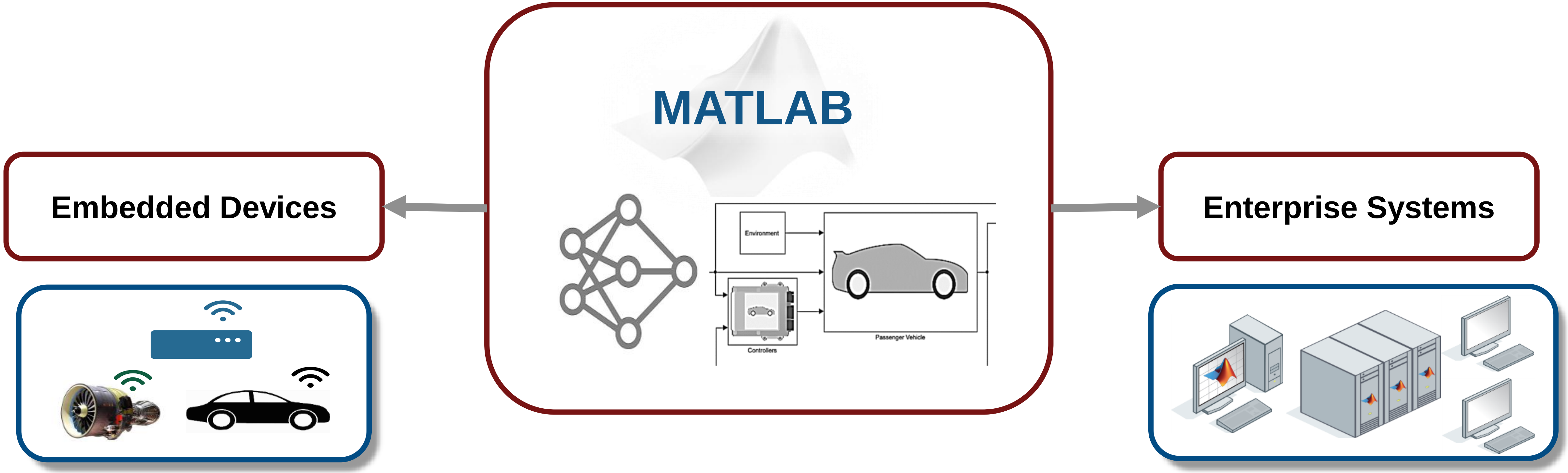
Increased productivity with interactive tools

Generate simulation data for complex models and systems

Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains

Deployment and Scaling for A.I.



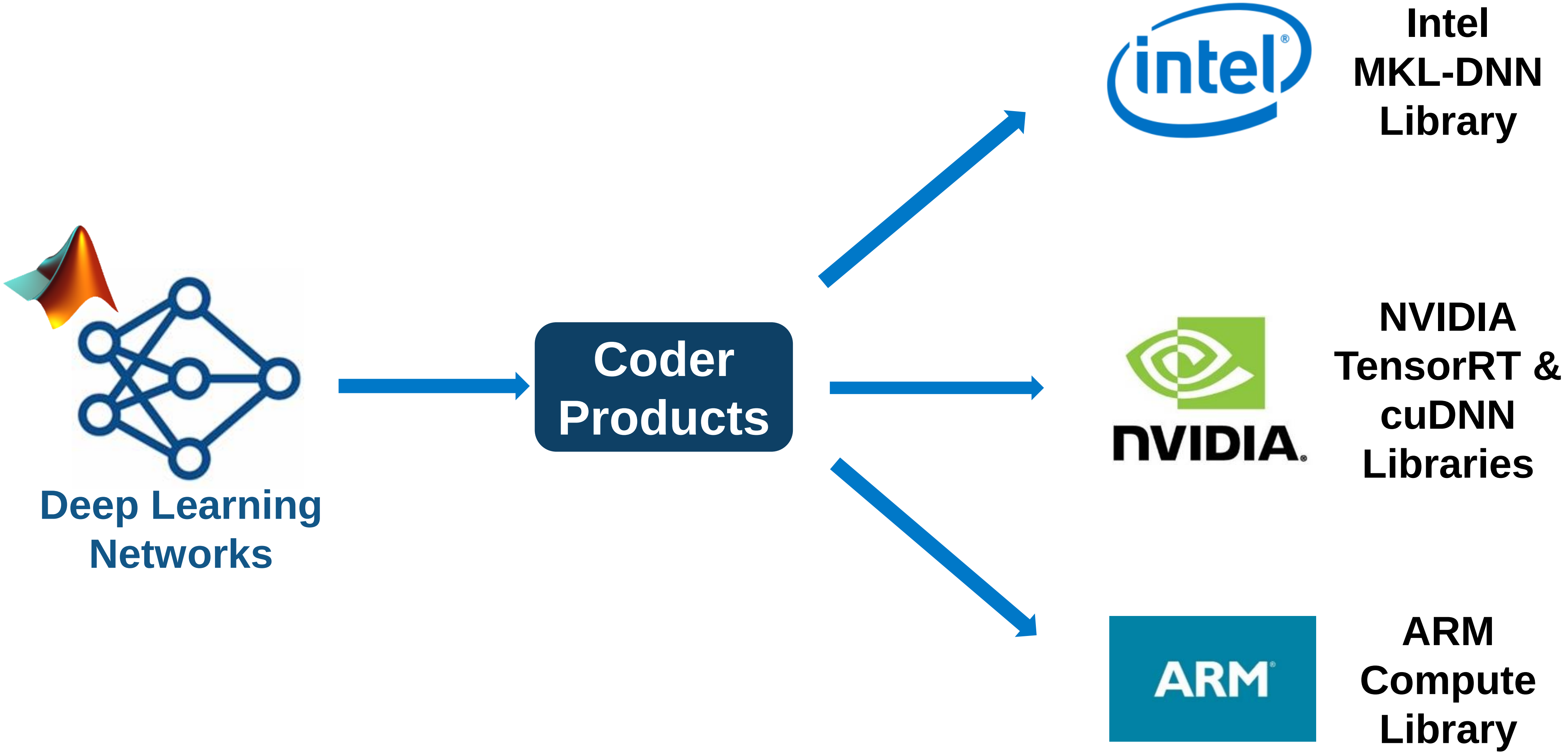
Embedded Devices – Automatic Code Generation

MATLAB Code

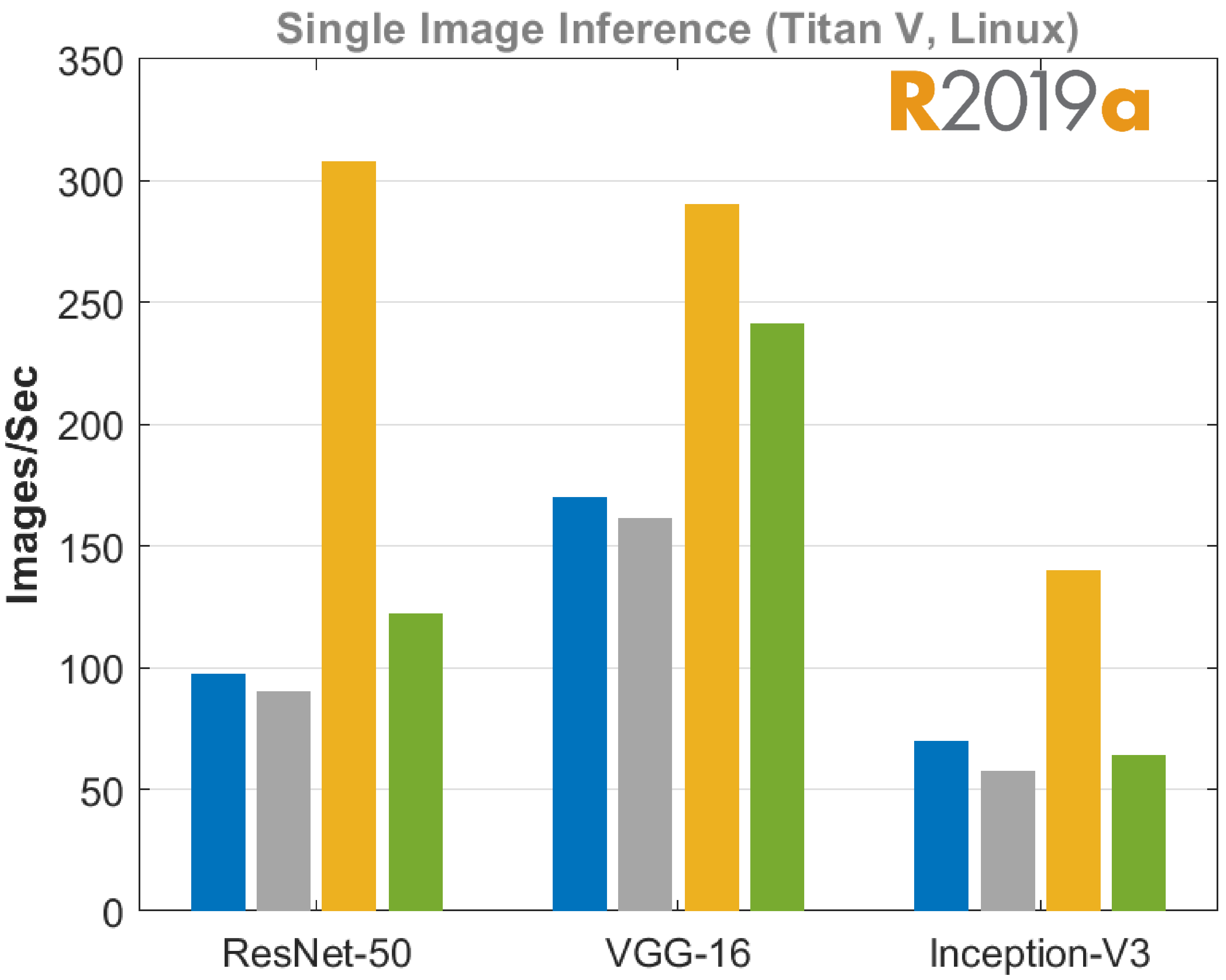
Auto-generated Code (C/C++/CUDA)

Deployment Target

Deploying Deep Learning Models for Inference



With GPU Coder, MATLAB is fast

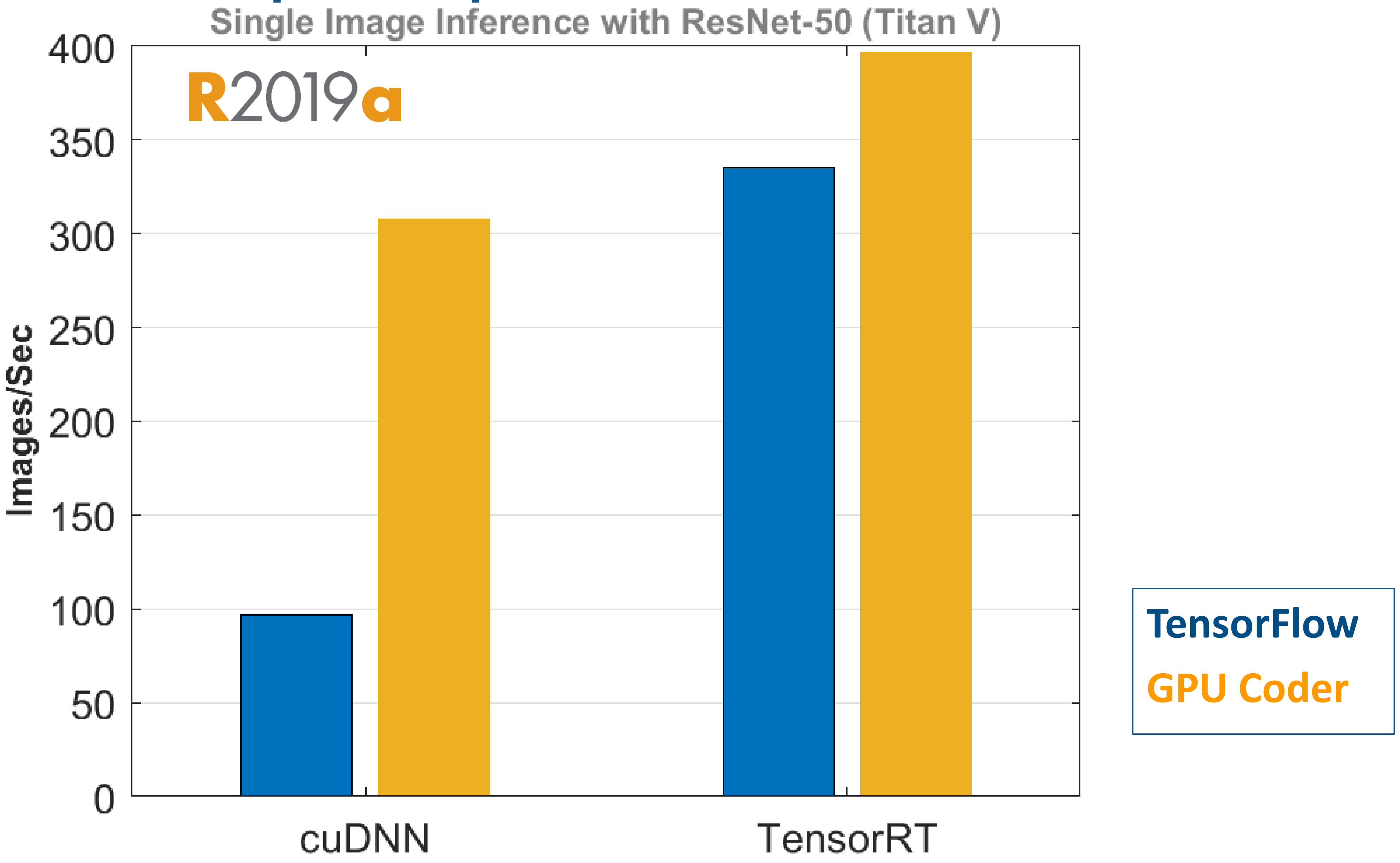


GPU Coder is faster than TensorFlow, MXNet and Pytorch

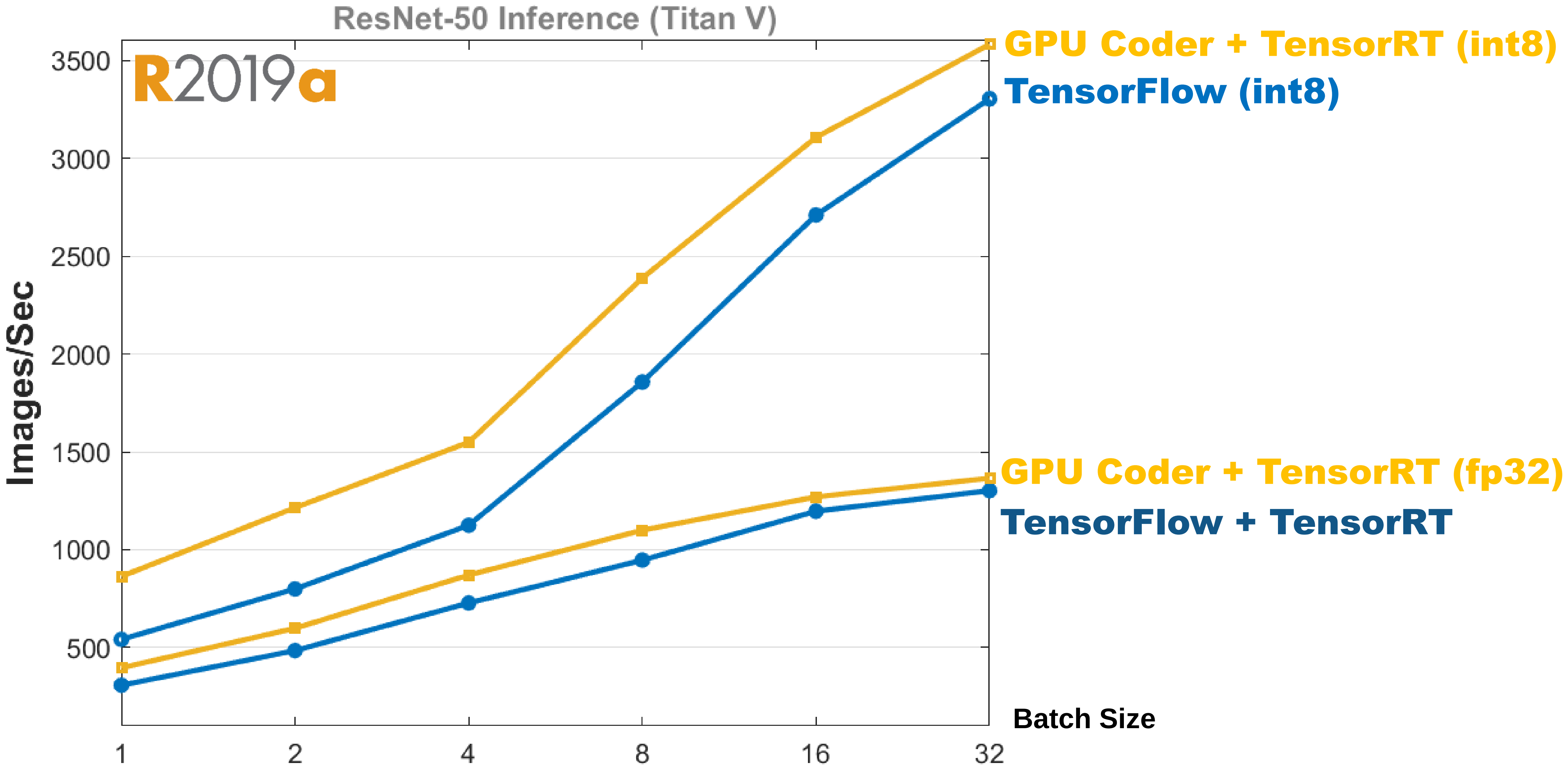
- TensorFlow
- MXNet
- GPU Coder
- PyTorch

Intel® Xeon® CPU 3.6 GHz - NVIDIA libraries: CUDA10 - cuDNN 7 - Frameworks: TensorFlow 1.13.0, MXNet 1.4.0 PyTorch 1.0.0

TensorRT speeds up inference for TensorFlow and GPU Coder

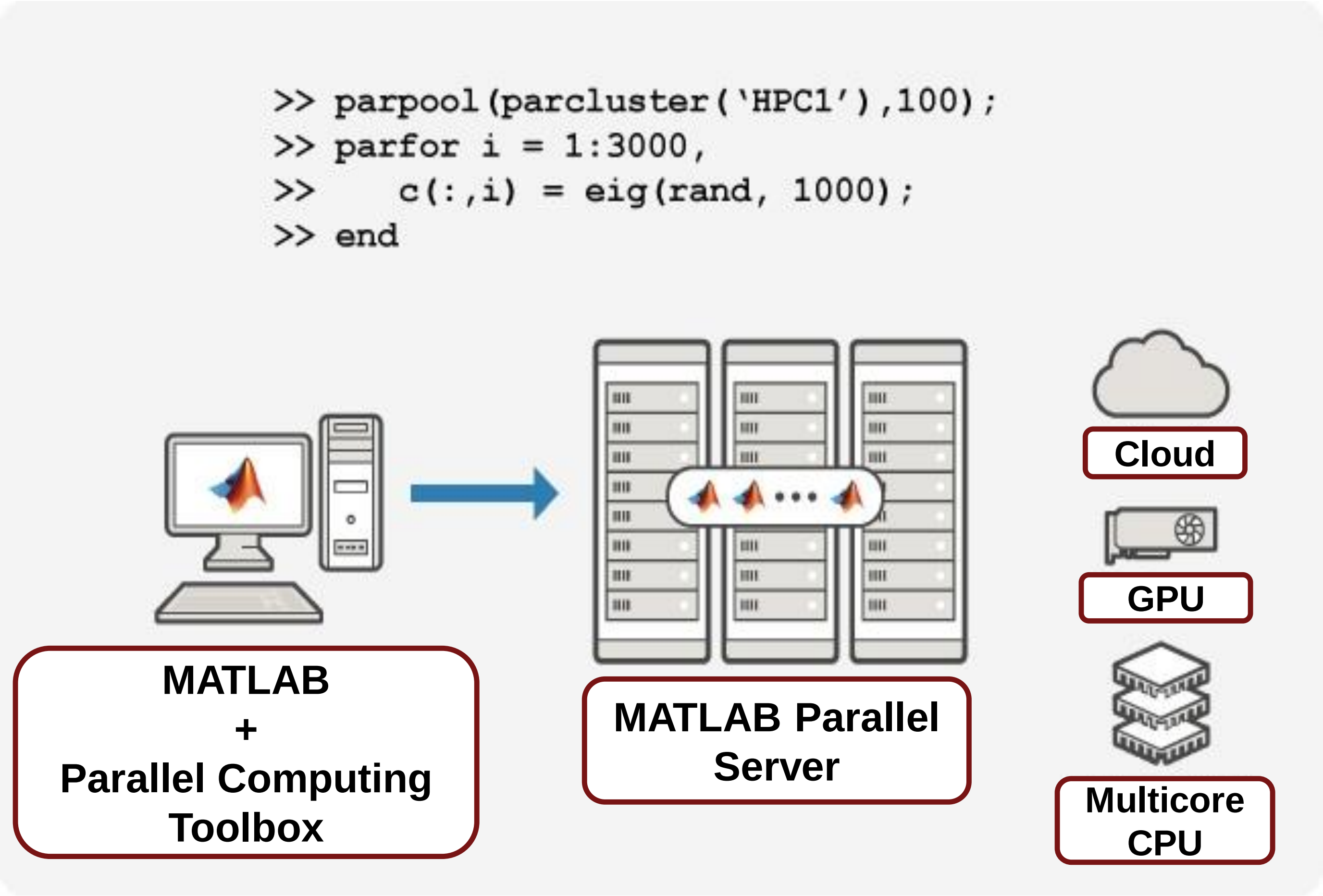


Even higher Speeds with Integer Arithmetic (int8)



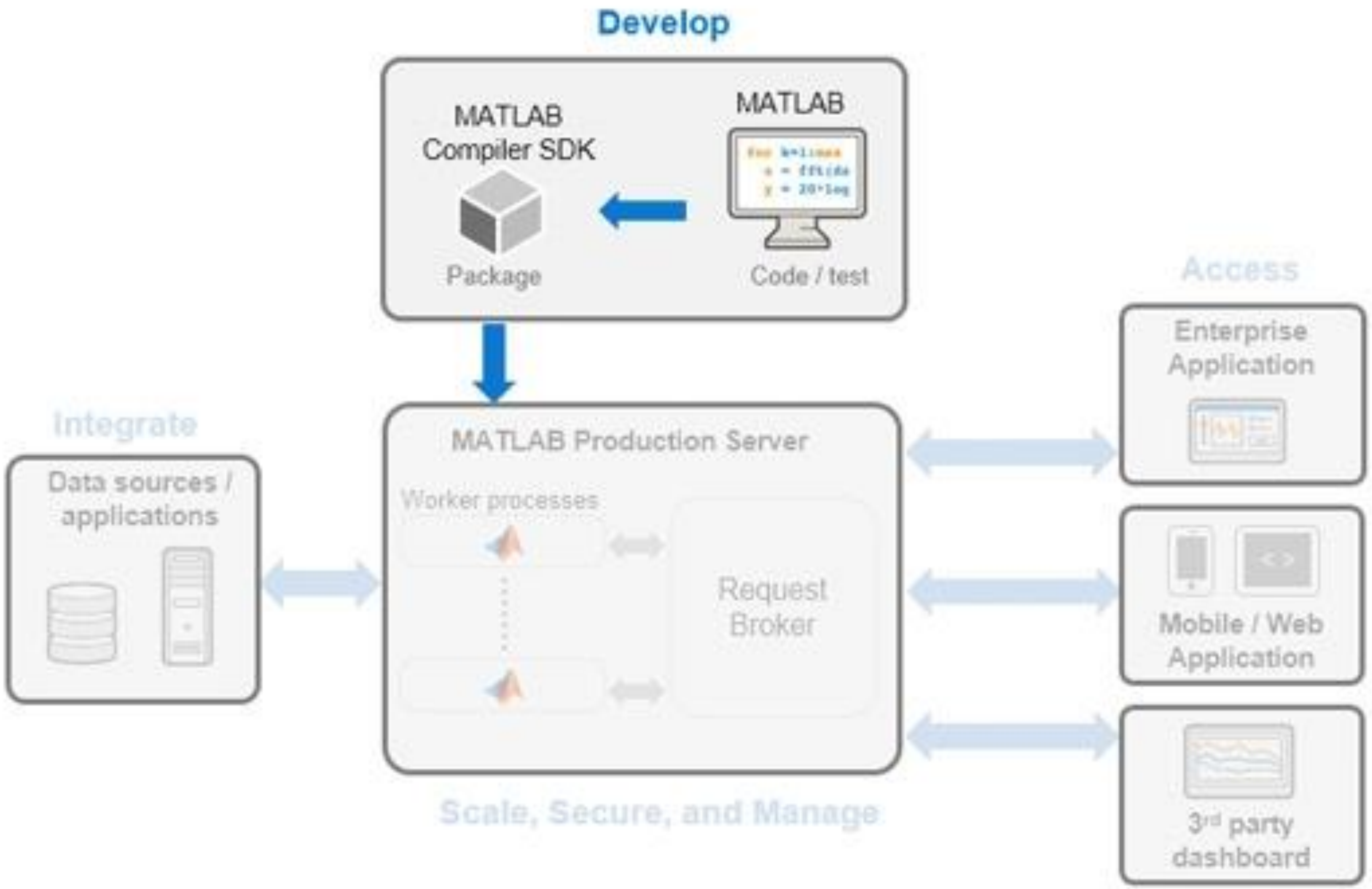
Intel® Xeon® CPU 3.6 GHz - NVIDIA libraries: CUDA10 - cuDNN 7 – Tensor RT 5.0.2.6. Frameworks: TensorFlow 1.13.0, MXNet 1.4.0 PyTorch 1.0.0

Enterprise Deployment



Run thousands of simulations in parallel with MATLAB Parallel Server to save hours of training time.

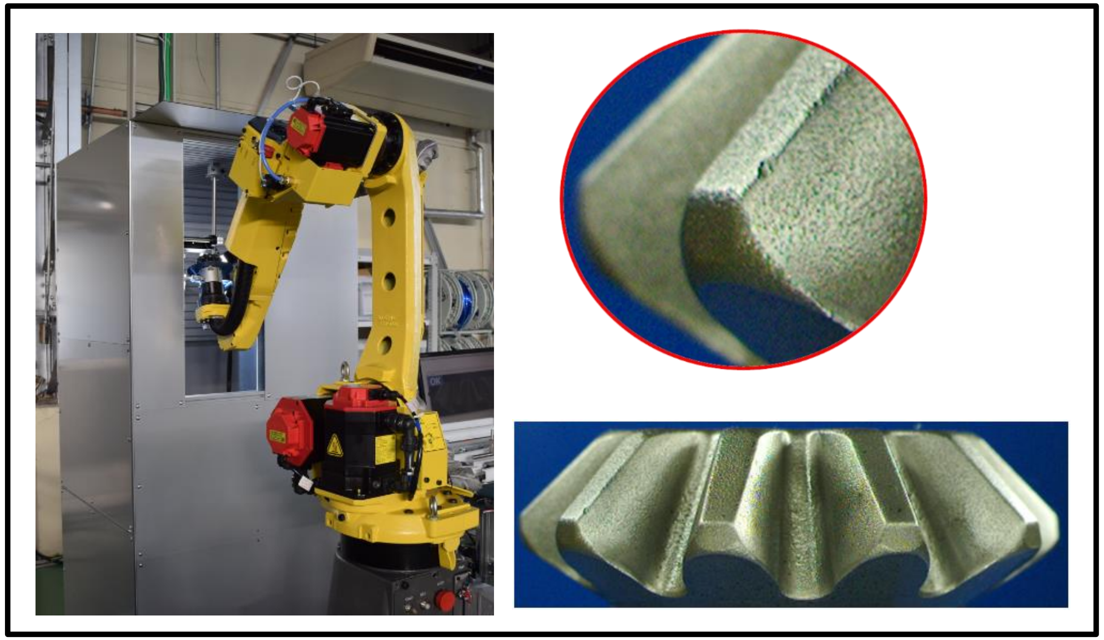
Enterprise Deployment



Deployment to the cloud with MATLAB Compiler and MATLAB Production Server

Musashi Seimitsu Industry Co.,Ltd.

Detect Abnormalities in Automotive Parts



Automated visual inspection of 1.3 million bevel gear per month

MATLAB use in project:

- Preprocessing of captured images
- Image annotation for training
- Deep learning based analysis
  - Various transfer learning methods (Combinations of CNN models, Classifiers)
  - Estimation of defect area using Class Activation Map (CAM)
  - Abnormality/defect classification
- Deployment to NVIDIA Jetson using GPU Coder

Why MATLAB for A.I. Tasks?

Increased productivity with interactive tools

Generate simulation data for complex models and systems

Ease of deployment and scaling to various platforms

Full A.I. workflows that cannot be easily replicated by other toolchains