

MathWorks
**AUTOMOTIVE
CONFERENCE 2024**

Building Confidence through Design in AI-driven Engineered Systems

Avinash Nehemiah, MathWorks

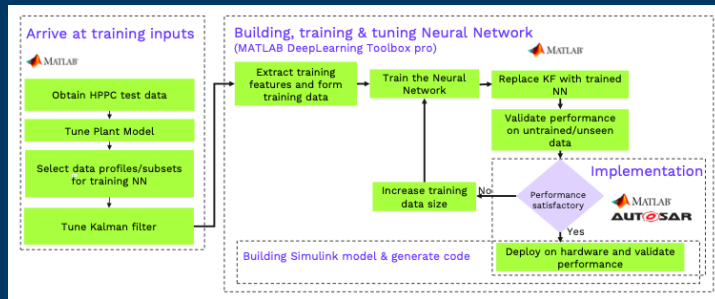


Applications of **AI in automotive** industry

Top **questions and concerns**

5 techniques to **build confidence** through design

The use of AI for engineered systems in Automotive continues to grow



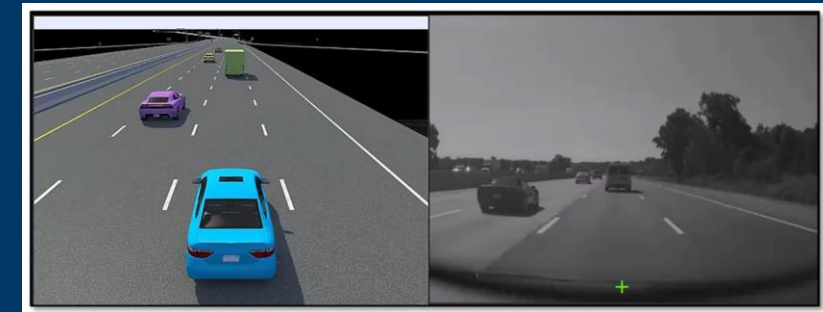
(KPIT) Virtual Sensor

Battery SOC and SOH Estimation using a Hybrid Machine Learning Approach



(Subaru) ROM

AI ROM to decrease time for Transmission Control system analysis



(APTIV) Data Generation

Converting test data to simulation using AI

MathWorks **AUTOMOTIVE CONFERENCE 2024** India
Register at mathworks.com/indiamac #MathWorksAutoConf

ServiceSage : A Gen AI-Based RCA Chat Assistant

19 November | Pune

 Bhakti Kalghatgi, Tata Motors

 Shubham Gupta, Tata Motors



MathWorks **AUTOMOTIVE CONFERENCE 2024** India
Register at mathworks.com/indiamac #MathWorksAutoConf

Embedded AI for Body Applications with MATLAB and Simulink

19 November | Pune

 Athulya Thazha, Mercedes-Benz Research and Development India Private Limited

 Shubham Kale, Mercedes-Benz Research and Development India Private Limited



MathWorks **AUTOMOTIVE CONFERENCE 2024** India
Register at mathworks.com/indiamac #MathWorksAutoConf

Tech Talk: Accelerating AI Adoption: From Design to Deployment in Mobility

19 November | Pune

 Myrtle Bini Rajendrababu, MathWorks

 Jayanth Balaji Avanasilingam, MathWorks

 Koustubh Shirke, MathWorks

 Nikita Pinto, MathWorks



Questions and concerns we need to answer to build confidence

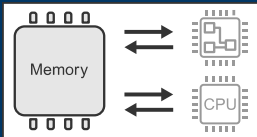
1. Are we using the best available AI model ?
2. How can we test impact of adding AI-based component to my system design ?
3. Do I have enough compute and memory to use an accurate AI model ?
4. Is it possible to verify and validate the AI model ?
5. Can I use my target hardware during development?



AI Modeling



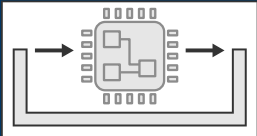
System-level Simulation



Memory and Execution Time

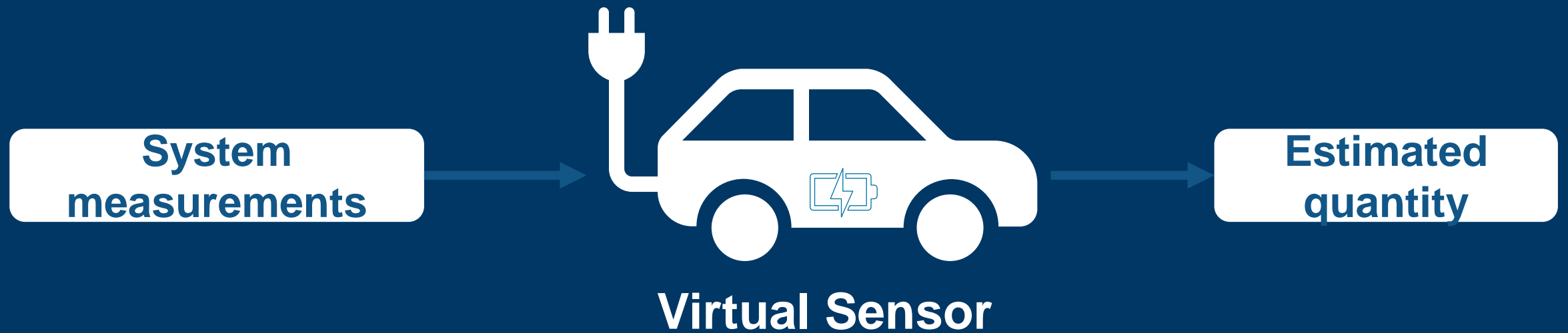


Verification and Validation



Testing on Hardware

Today we'll focus on Virtual Sensor modeling use case

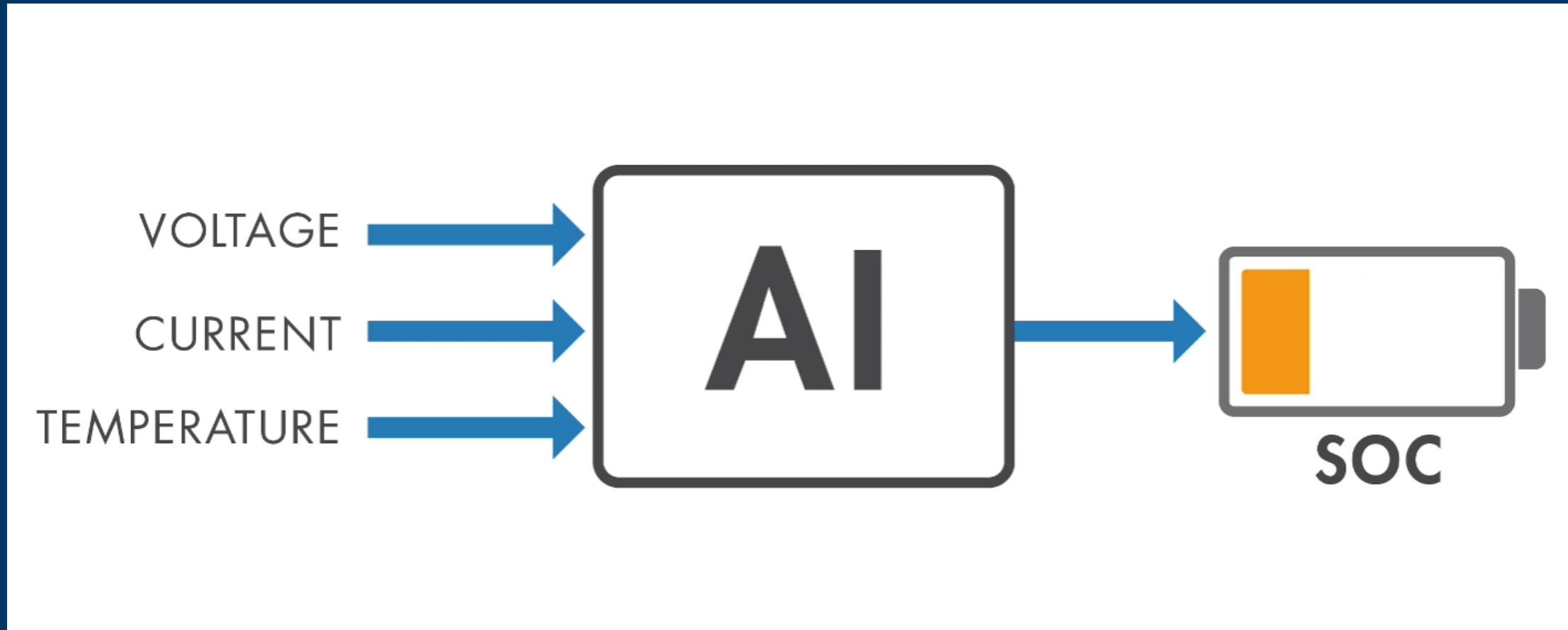


A Virtual Sensor mimics a physical sensor using data from other measurements to estimate the quantity of interest

Why are virtual sensors relevant ?

- A **physical sensor** may be:
- Expensive
 - Slow
 - Noisy
 - Unreliable
 - Not feasible
 - Unmanufacturable
 - Degrading over time
 - Requiring redundancy
 - etc.

AI Virtual Sensor for Battery State-of-Charge Estimation



Questions and concerns we need to answer to build confidence

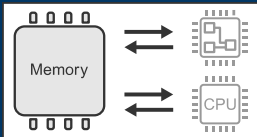
5 keys to building confidence



AI Modeling



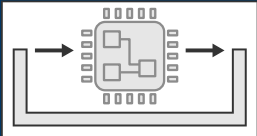
System-level Simulation



Memory and Execution Time

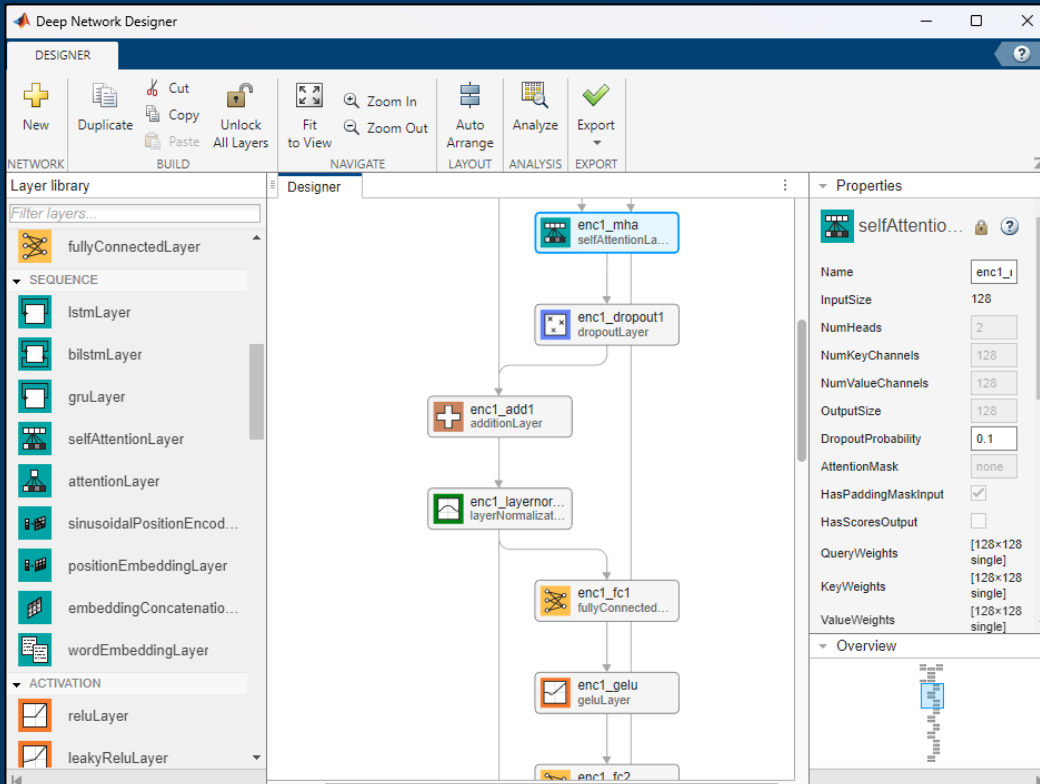


Verification and Validation



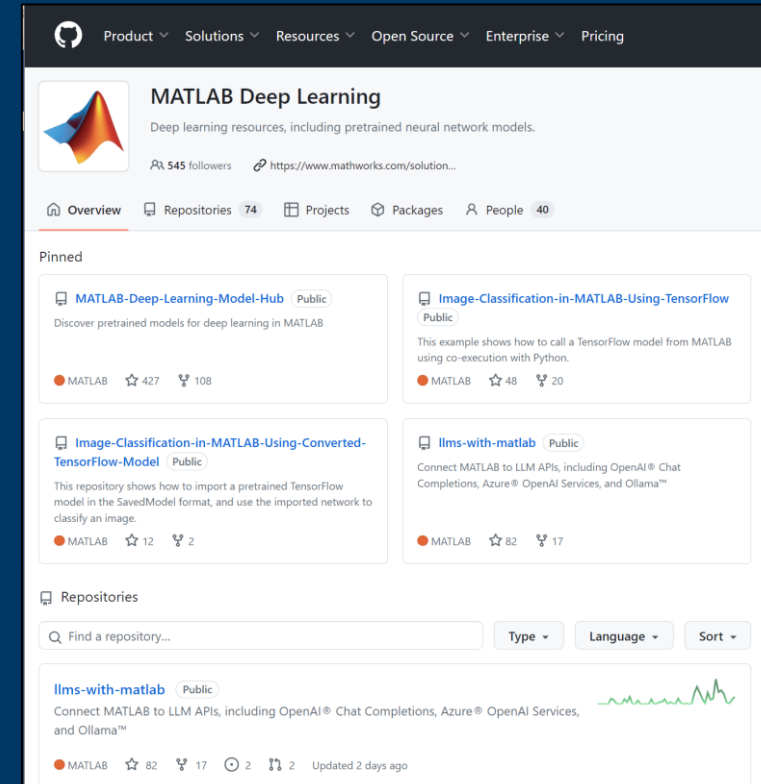
Testing on Hardware

Two ways of leveraging AI in MATLAB and Simulink



The screenshot shows the Deep Network Designer software interface. On the left is a 'Layer library' with categories like 'SEQUENCE' (lstmLayer, bilstmLayer, gruLayer, selfAttentionLayer, attentionLayer, sinusoidalPositionEncod..., positionEmbeddingLayer, embeddingConcatenati..., wordEmbeddingLayer) and 'ACTIVATION' (reluLayer, leakyReluLayer). The main 'Designer' area shows a flowchart of a neural network with layers: enc1_mha selfAttentionLa..., enc1_dropout1 dropoutLayer, enc1_add1 additionLayer, enc1_layernor... layerNormalizat..., enc1_fc1 fullyConnected..., enc1_gelu geluLayer, and enc1_fc2. A 'Properties' panel on the right is open for the 'selfAttention...' layer, showing settings like Name, InputSize (128), NumHeads (2), NumKeyChannels (128), NumValueChannels (128), OutputSize (128), DropoutProbability (0.1), AttentionMask (none), HasPaddingMaskInput (checked), HasScoresOutput (unchecked), QueryWeights ([128x128 single]), KeyWeights ([128x128 single]), and ValueWeights ([128x128 single]).

Build your own models



The screenshot shows the MATLAB Deep Learning repository page on GitHub. The page title is 'MATLAB Deep Learning' and it includes a description: 'Deep learning resources, including pretrained neural network models.' It shows 545 followers and a link to the solution page. The 'Pinned' section features several repositories:

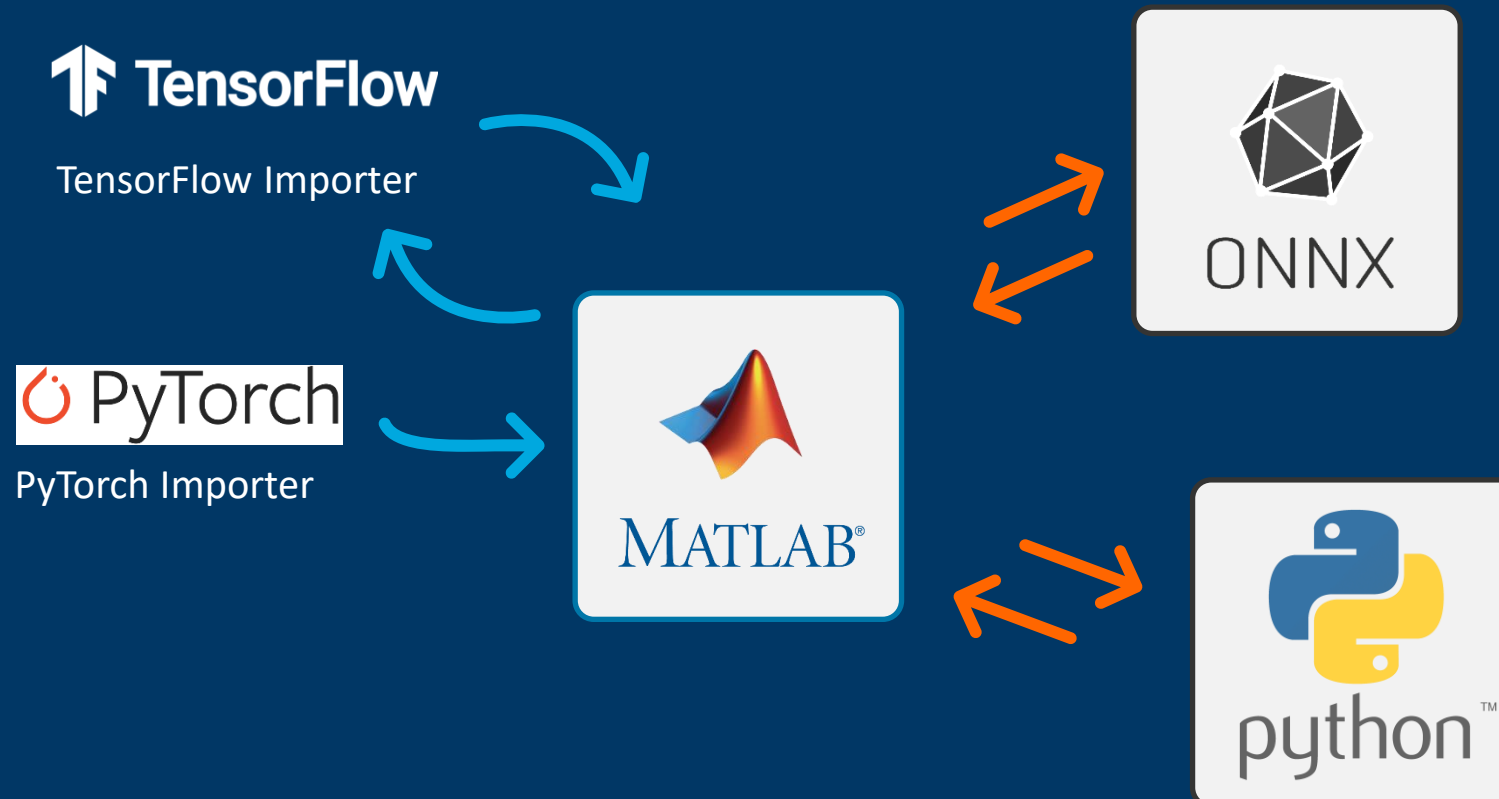
- MATLAB-Deep-Learning-Model-Hub** (Public): Discover pretrained models for deep learning in MATLAB. 427 stars, 108 forks.
- Image-Classification-in-MATLAB-Using-TensorFlow** (Public): This example shows how to call a TensorFlow model from MATLAB using co-execution with Python. 48 stars, 20 forks.
- Image-Classification-in-MATLAB-Using-Converted-TensorFlow-Model** (Public): This repository shows how to import a pretrained TensorFlow model in the SavedModel format, and use the imported network to classify an image. 12 stars, 2 forks.
- llms-with-matlab** (Public): Connect MATLAB to LLM APIs, including OpenAI® Chat Completions, Azure® OpenAI Services, and Ollama™. 82 stars, 17 forks.

Repositories section includes a search bar and filters for Type, Language, and Sort.

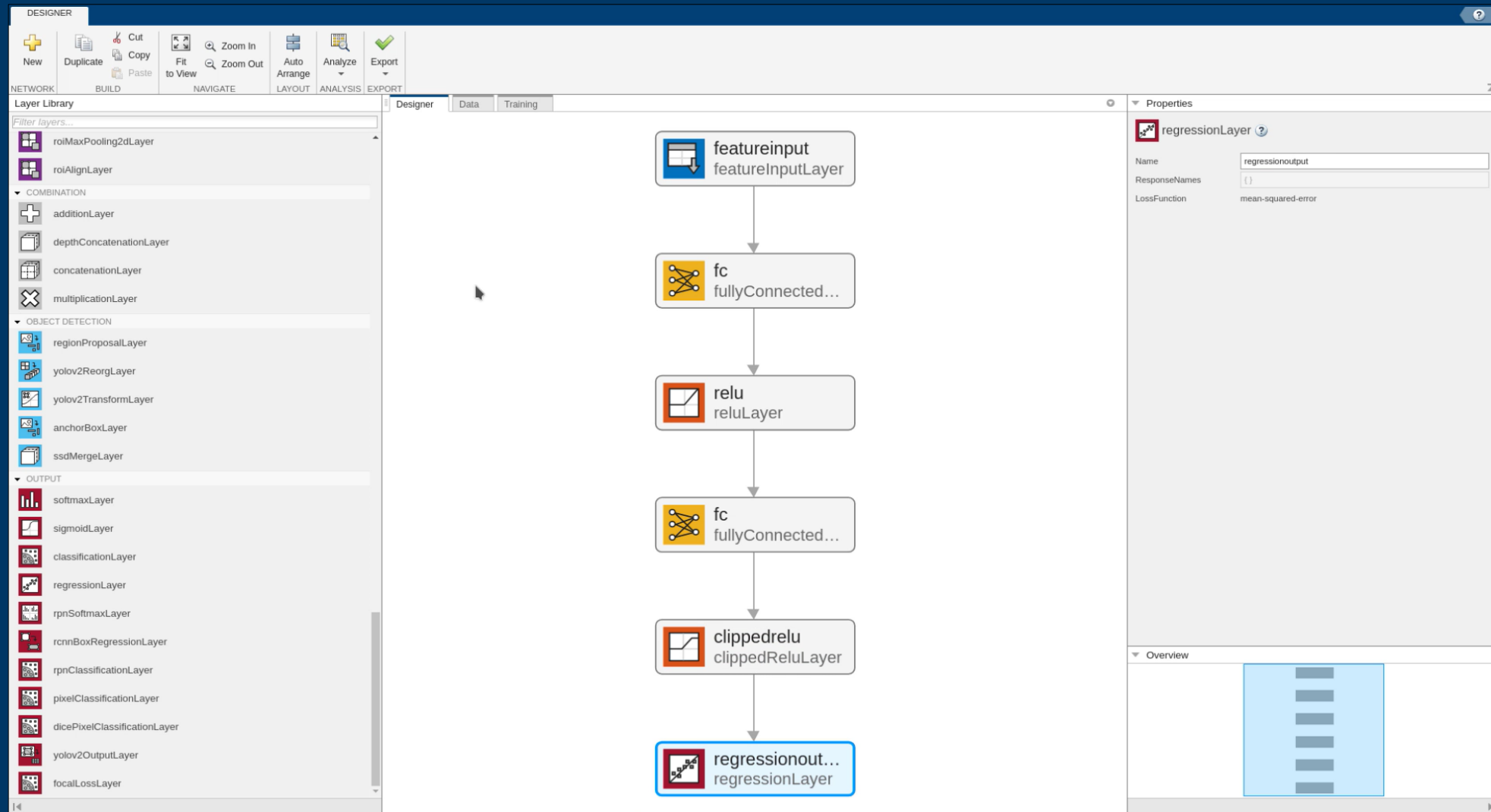
Leverage models from research



Build, run, or fine-tune models locally

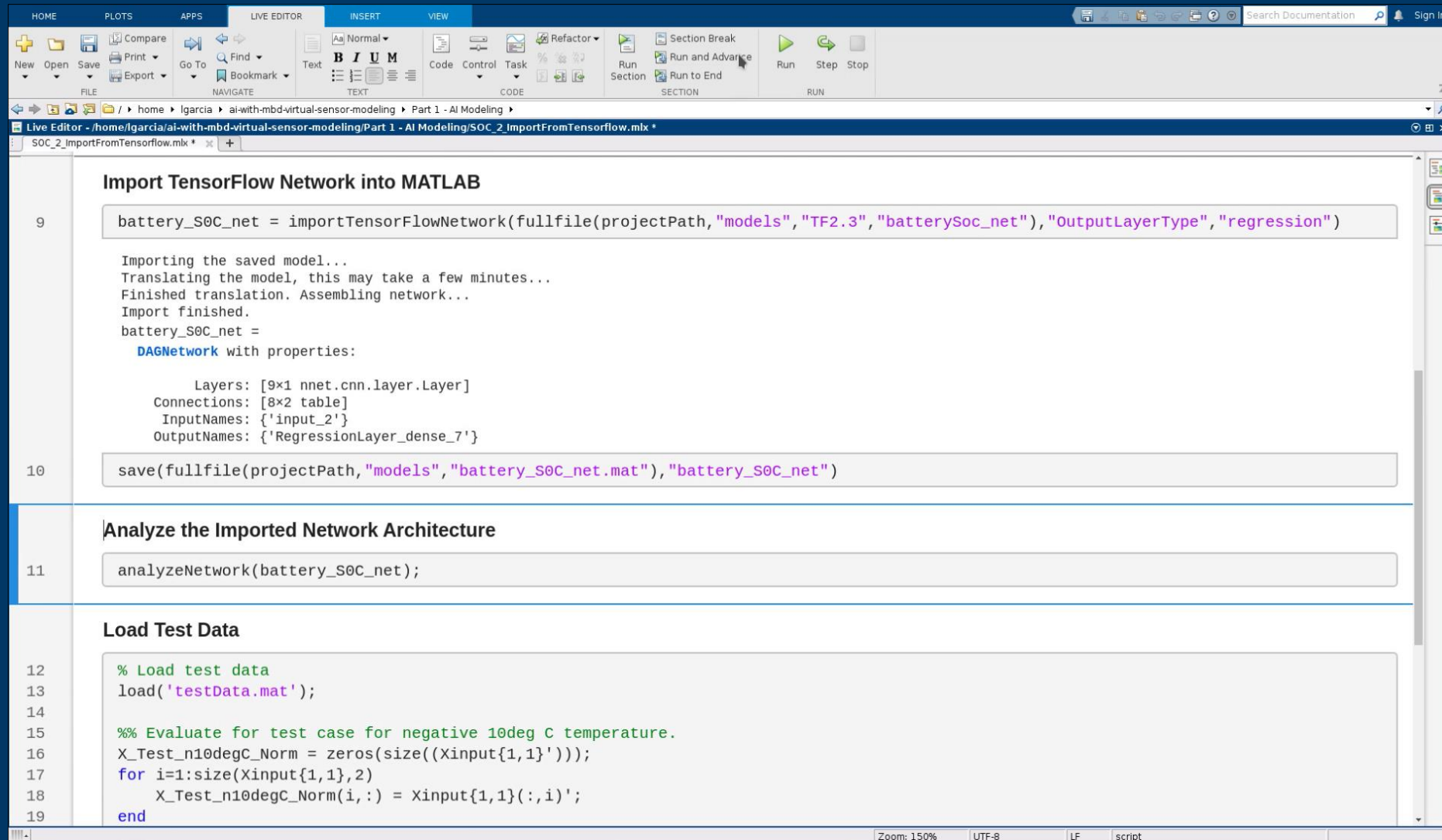


Create a white-box model from scratch in MATLAB



Low Code AI Model Development

Programmatically create model in MATLAB



The screenshot displays the MATLAB Live Editor interface. The top menu bar includes HOME, PLOTS, APPS, LIVE EDITOR, INSERT, and VIEW. The toolbar contains icons for file operations (New, Open, Save, Print, Export), navigation (Go To, Find, Bookmark), text formatting (Normal, Bold, Italic, Underline, Monospace), code execution (Run, Step, Stop), and other tools (Refactor, Section Break, Run and Advance, Run Section, Run to End). The address bar shows the current file path: `/home/lgarcia/ai-with-mbd-virtual-sensor-modeling/Part 1 - AI Modeling/SOC_2_ImportFromTensorflow.mlx`. The main editor area is divided into sections:

- Import TensorFlow Network into MATLAB**:
 - Line 9: `battery_S0C_net = importTensorFlowNetwork(fullfile(projectPath,"models","TF2.3","batterySoc_net"),"OutputLayerType","regression")`
 - Output text: `Importing the saved model...
Translating the model, this may take a few minutes...
Finished translation. Assembling network...
Import finished.
battery_S0C_net =
 DAGNetwork with properties:

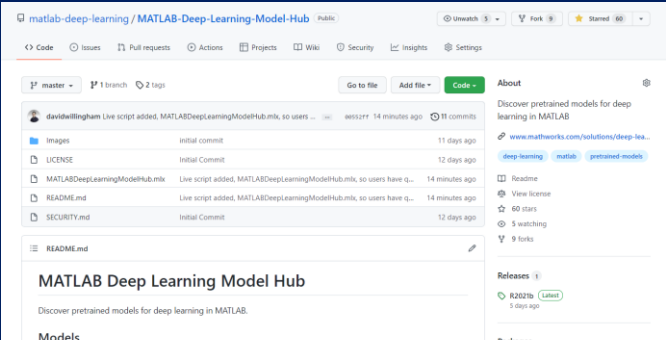
 Layers: [9x1 nnet.cnn.layer.Layer]
 Connections: [8x2 table]
 InputNames: {'input_2'}
 OutputNames: {'RegressionLayer_dense_7'}`
 - Line 10: `save(fullfile(projectPath,"models","battery_S0C_net.mat"),"battery_S0C_net")`
- Analyze the Imported Network Architecture**:
 - Line 11: `analyzeNetwork(battery_S0C_net);`
- Load Test Data**:
 - Line 12: `% Load test data`
 - Line 13: `load('testData.mat');`
 - Line 14: (blank)
 - Line 15: `%% Evaluate for test case for negative 10deg C temperature.`
 - Line 16: `X_Test_n10degC_Norm = zeros(size(Xinput{1,1}));`
 - Line 17: `for i=1:size(Xinput{1,1},2)`
 - Line 18: `X_Test_n10degC_Norm(i,:) = Xinput{1,1}(:,i)';`
 - Line 19: `end`

The status bar at the bottom indicates `Zoom: 150%`, `UTF-8`, `LF`, and `script`.

Discover the latest pretrained models – on GitHub

MATLAB Deep Learning Model Hub

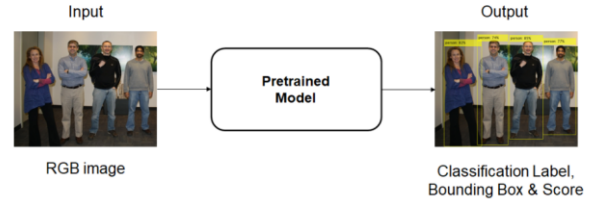
- Quick reference to over 60* pretrained models in:
 - Computer Vision
 - Natural Language Processing
 - Audio
 - Lidar
 - * New models added every month



Object Detection

Object detection is a computer vision technique used for locating instances of objects in images or videos. When humans look at images or video, we can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate this intelligence using a computer.

Inputs are RGB images, the output is the predicted label, bounding box and score:



Input: RGB image

Output: Classification Label, Bounding Box & Score

These networks have been trained to detect 80 objects classes from the COCO dataset. These models are suitable for training a custom object detector using transfer learning.

Network	Backbone Networks	Size (MB)	Mean Average Precision (mAP)	Object Classes	Location
EfficientDet-DO	efficientnet	15.9	33.7	80	GitHub
YOLO v4	yolov4-coco yolov4-tiny-coco	229 21.5	44.2 19.7	80	GitHub
YOLO v3	darknet53-coco tiny-yolov3-coco	220.4 31.5	34.4 9.3	80	Doc
YOLO v2	darknet19-VOC darknet19-COCO tiny-yolo_v2-coco	180 181 40	75.4 28.7 10.5	80	GitHub

Tips for selecting a model

Pretrained object detectors have different characteristics that matter when choosing a network to apply to your problem. The most important characteristics are mean average precision (mAP), speed, and size. Choosing a network is generally a tradeoff between these characteristics.

Run experiments to find the best AI model

The screenshot shows the Experiment Manager interface. The main area displays the following information:

Result Details

Baseline Tuning 2/7/2020, 12:53:36 PM 7/16 Trials

(View Experiment Source)

✔ Complete 7 ⚠ Stopped 0 ! Error 0
○ Running 1 ⏸ Queued 8 ✖ Canceled 0

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

Spectrum of ways to leverage AI models



Build, run, or fine-tune
models locally

Connect to top performing
models in the cloud

Connect to top performing models in the cloud

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization
1	Gemini-1.5-Pro-Exp-0801	1300	+6/-5	12672	Google
2	GPT-4o-2024-05-13	1286	+3/-2	69832	OpenAI
2	GPT-4o-mini-2024-07-18	1280	+6/-4	12047	OpenAI
4	Claude 3.5 Sonnet	1271	+3/-4	40174	Anthropic
4	Gemini-Advanced-0514	1266	+3/-4	50686	Google
4	Meta-Llama-3.1-405b-Instruct	1262	+6/-7	8454	Meta



Leaderboard: <https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard>



matlab-deep-learning/
llms-with-matlab

Connect MATLAB to the OpenAI Chat Completions API (which powers ChatGPT)

5 Contributors 4 Issues 62 Stars 14 Forks

GitHub - matlab-deep-learning/llms-with-matlab: Connect MATLAB to the OpenAI Chat Completions API (which powers ChatGPT)

github.com • 5 min read

Text Generation Models

Create internal assistants that have knowledge about internal documentation

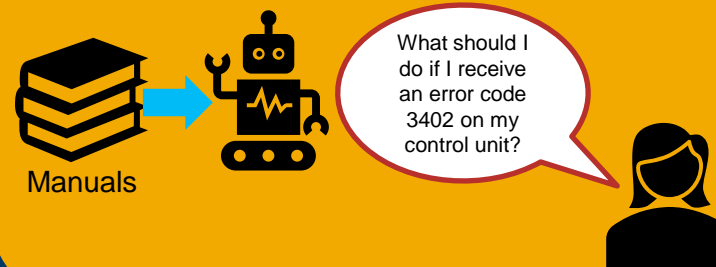


Image Generation Models

Generate images to test robustness of systems



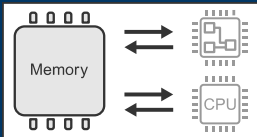
Questions and concerns we need to answer to build confidence



AI Modeling



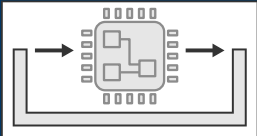
System-level Simulation



Memory and Execution Time



Verification and Validation



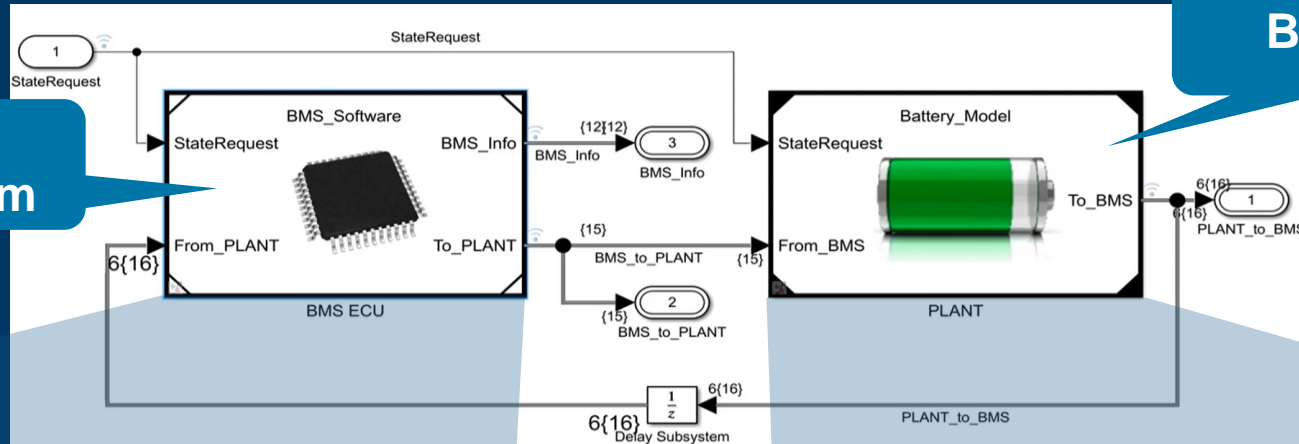
Testing on Hardware

5 keys to building confidence

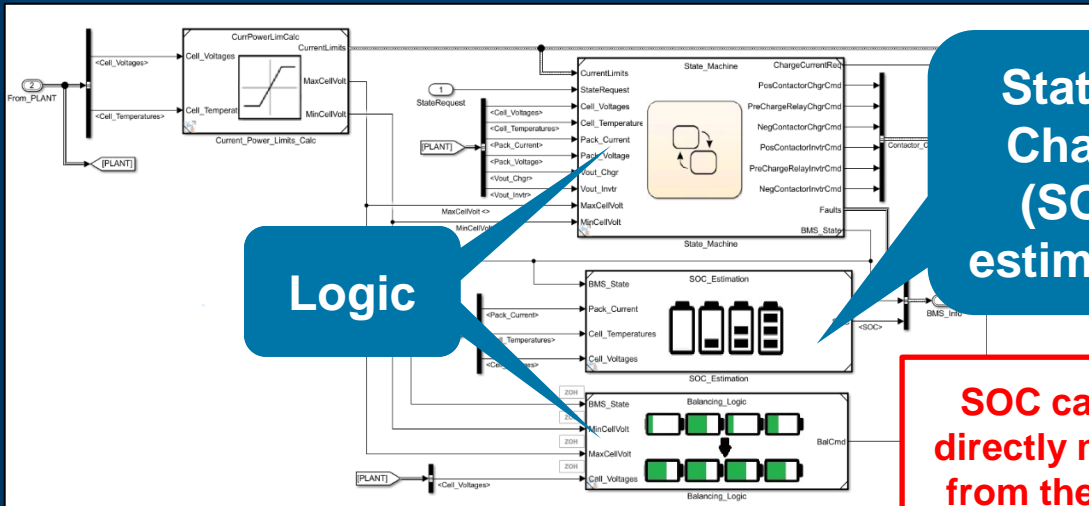
1. Access and explore state of art AI models

Use Simulink to create a system-level model that incorporates the AI virtual sensor

ECU Battery Management System



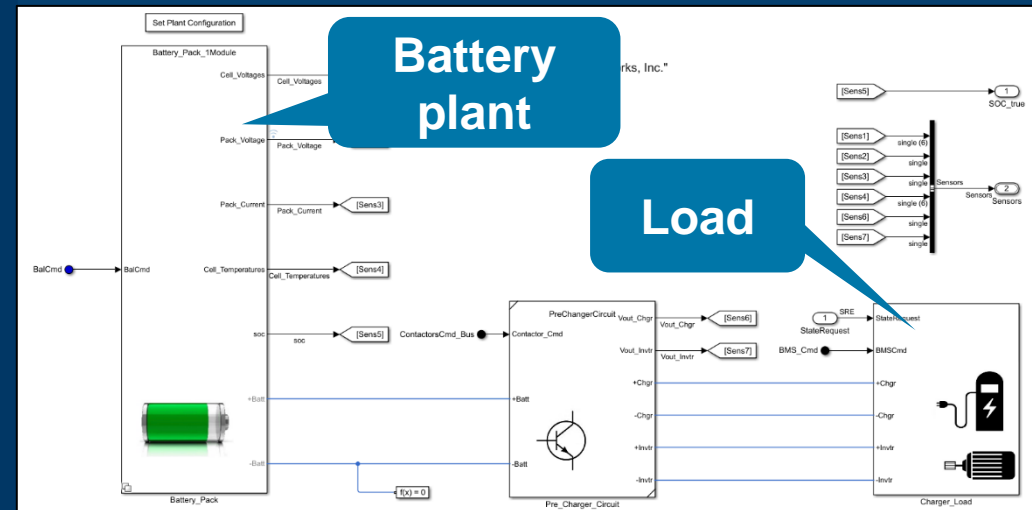
Battery



Logic

State of Charge (SOC) estimation

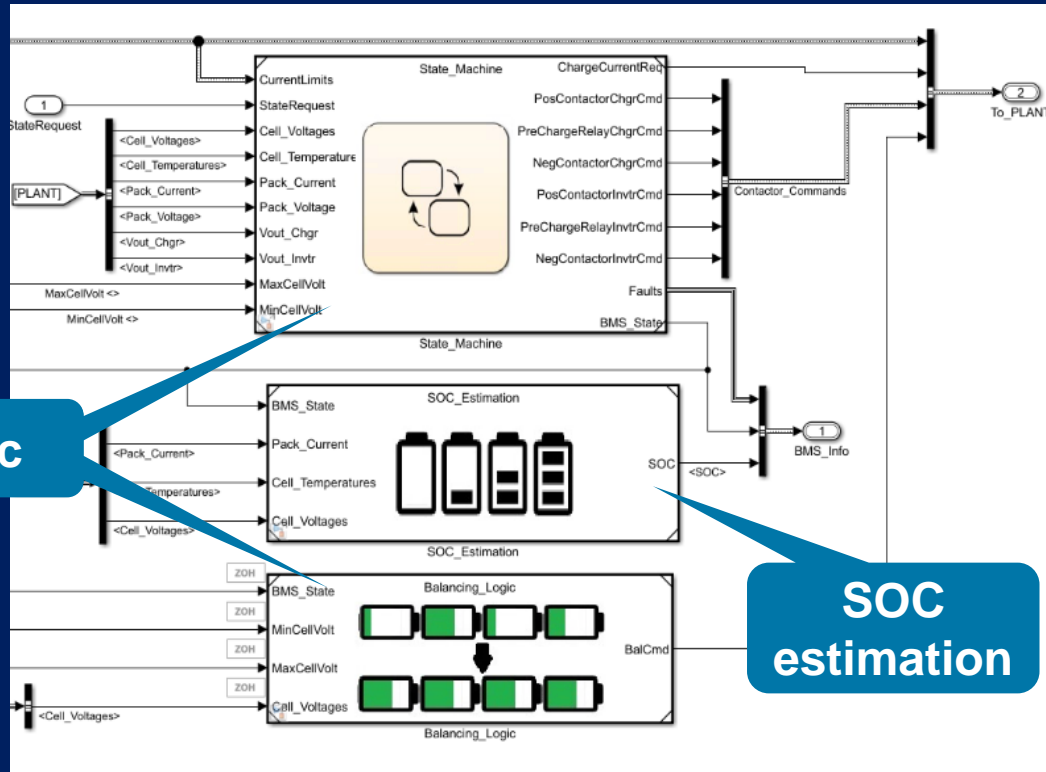
SOC cannot be directly measured from the battery!



Battery plant

Load

Virtual sensor for Battery State of Charge (SOC) estimation

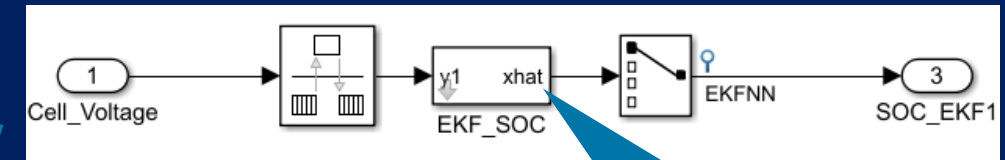


Logic

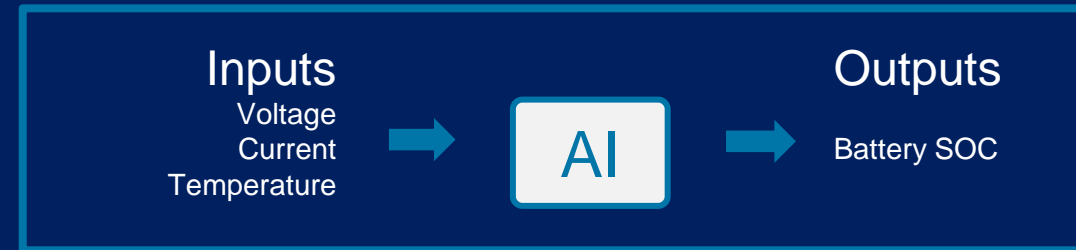
SOC estimation

Why AI over Kalman filtering?

- No need of an internal battery model
- Training directly on measured data
- Capture very complex data relationships

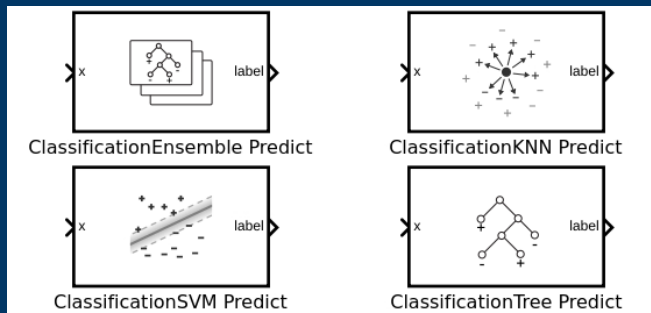
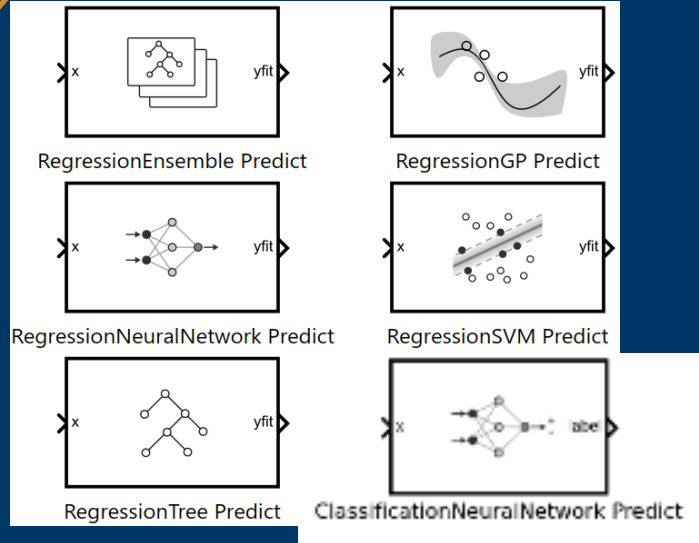


Extended Kalman filter

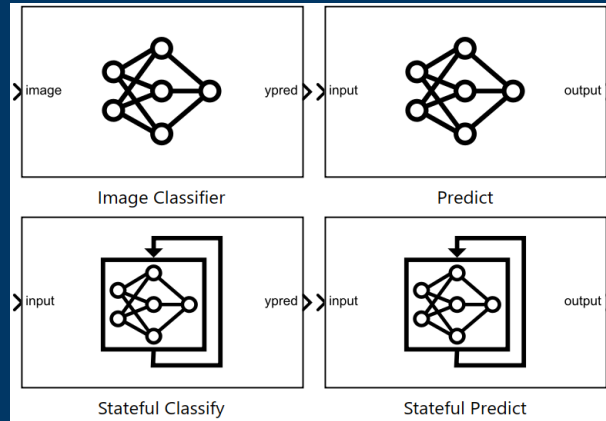


Simulink now includes more AI blocks for more applications

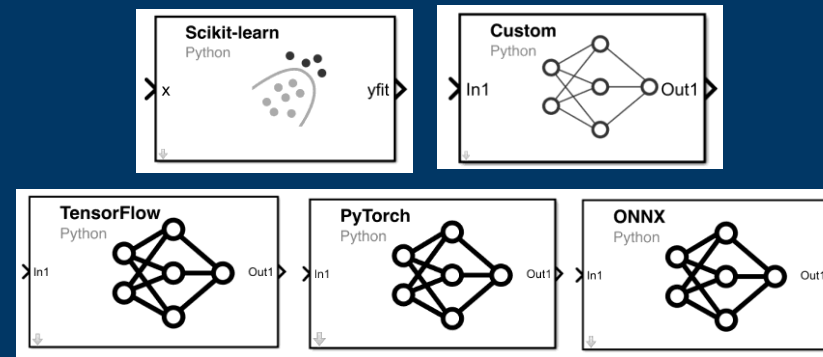
AI core



Regression and classification blocks
And more (incremental and cluster analysis blocks)
(Statistics and Machine Learning Toolbox)

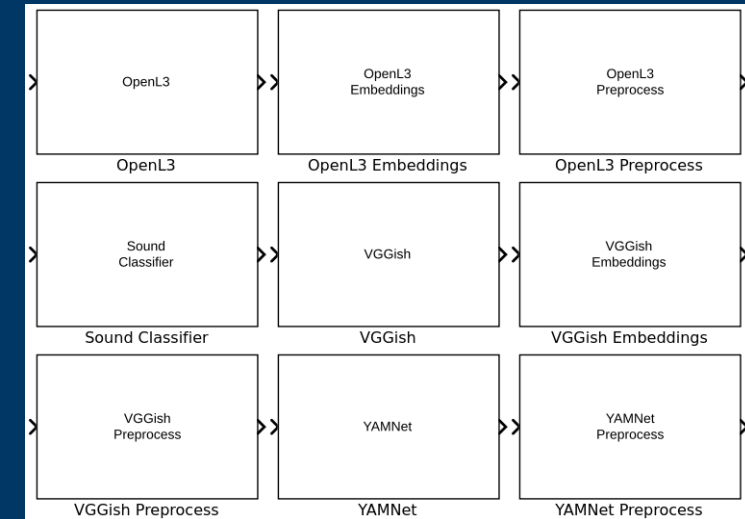


Neural network classify and predict blocks
And the list of layer blocks
(Deep Learning Toolbox)

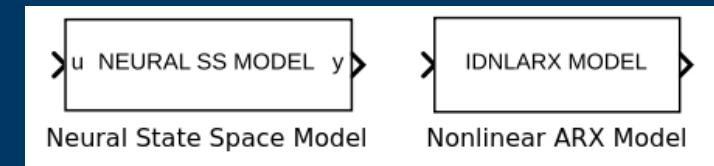


Co-execution blocks

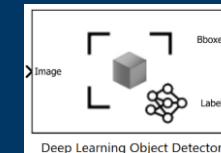
Specialized



Audio Toolbox



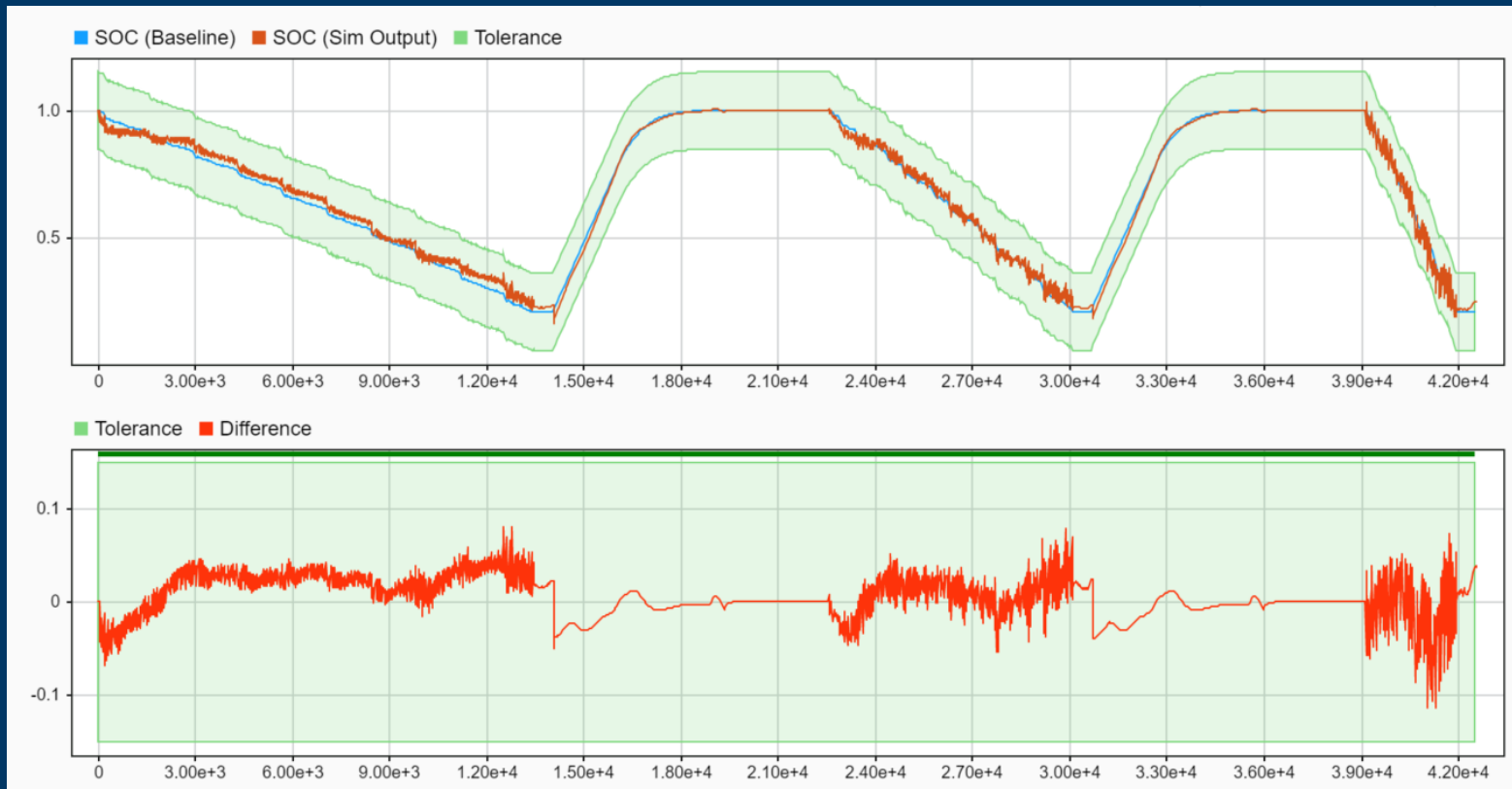
System Identification Toolbox



Computer Vision Toolbox

Perform baseline testing in simulation

Use Data-Driven Tests in Simulation to Assert **Accuracy**



Baseline Data

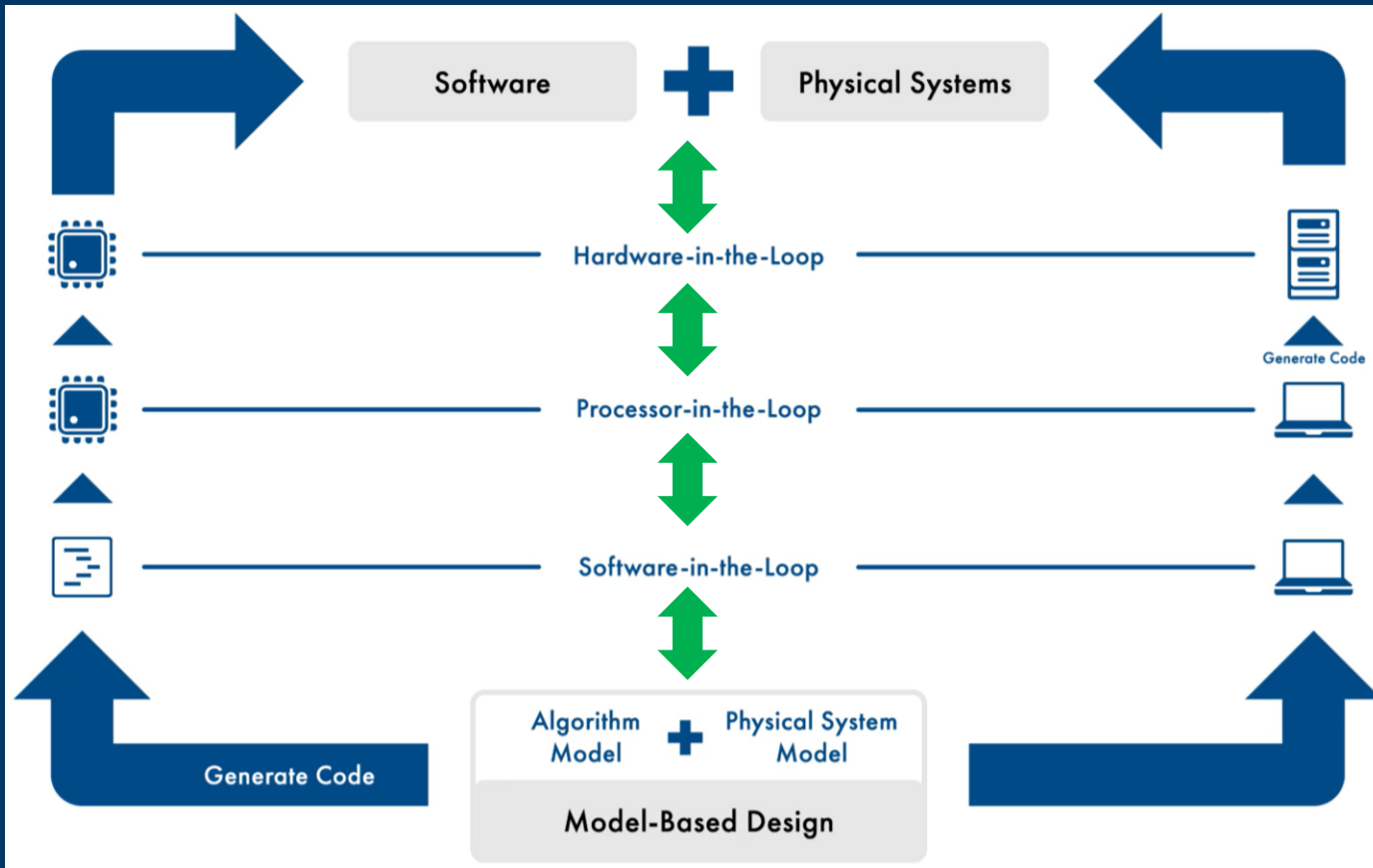
Ground truth
e.g., lab measurements

VS.

Simulation Output

Prediction results

Perform back-to-back testing on model behavior



Perform back-to-back testing

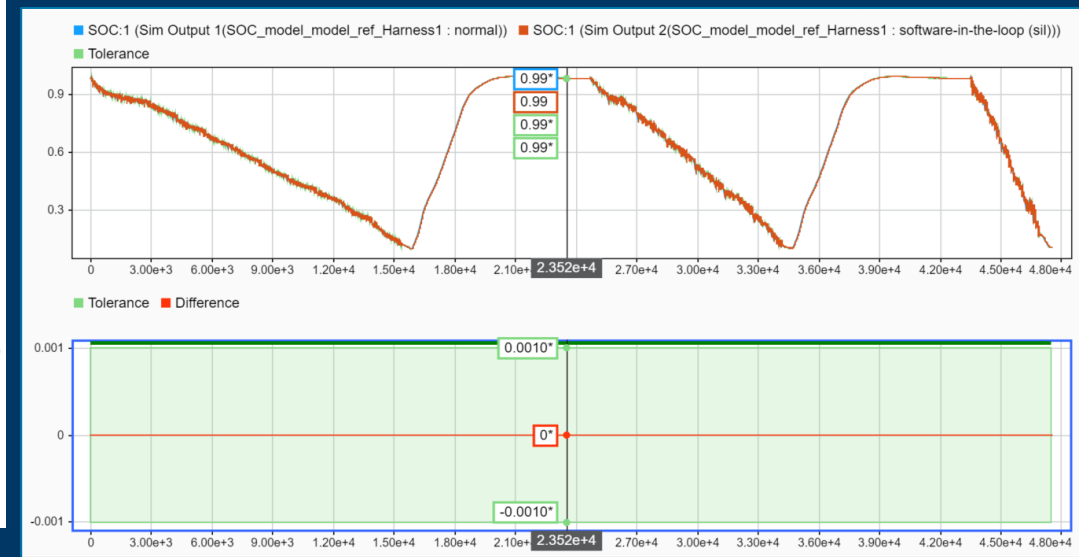
Set up a test to compare the component under test outputs in different simulation modes

Select simulation modes:

Simulation1:

Simulation2:

Different Simulation Modes



Test equivalence of model behavior from desktop simulations to SIL/PIL/HIL

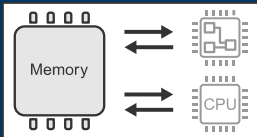
Questions and concerns we need to answer to build confidence



AI Modeling



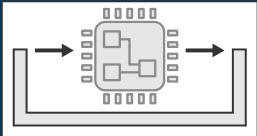
System-level Simulation



Memory and Execution Time



Verification and Validation

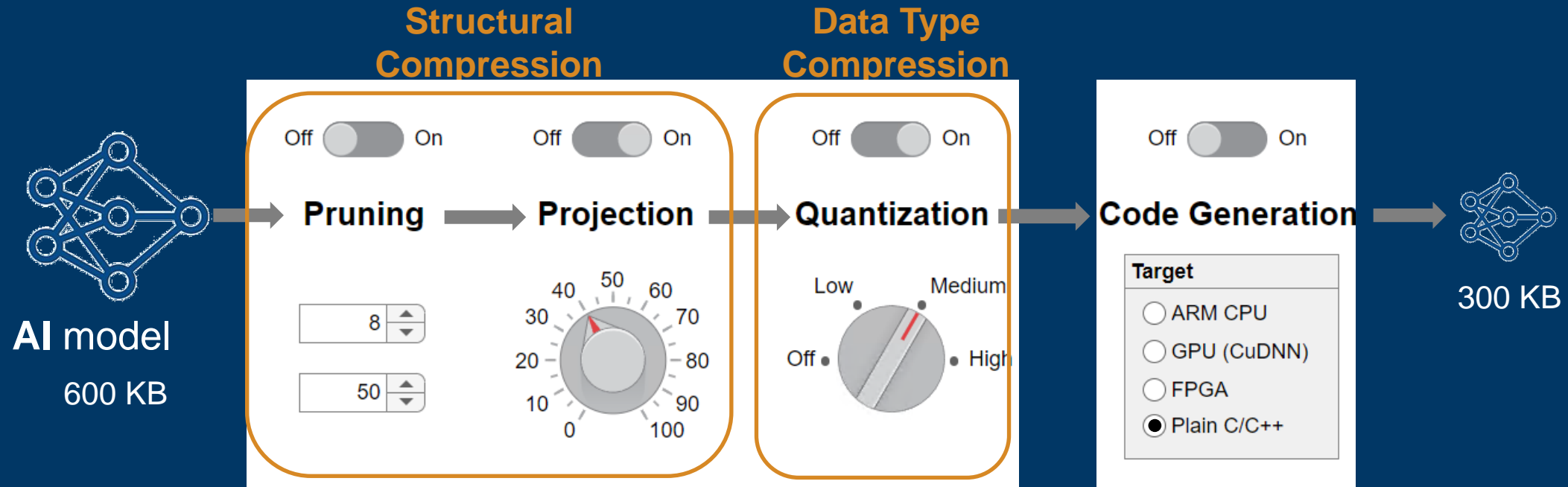


Testing on Hardware

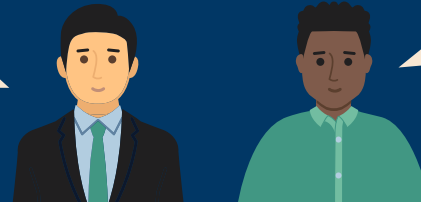
5 keys to building confidence

1. Access and explore state of art AI models
2. System-level testing to assess impact of adding AI component

Reduce model footprint and accelerate inference of DL models for deployment to the edge

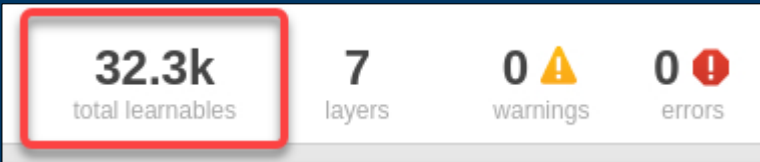
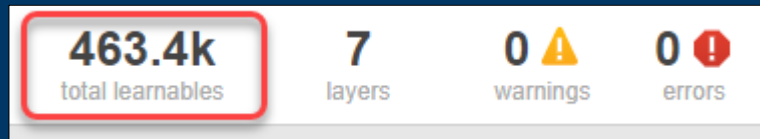


“my model is 40MB, I needed it to be less than 10MB to fit.”



“model is 600kb and want to reduce it to 300kb. If I'm not fitting it in, I don't have a working solution”

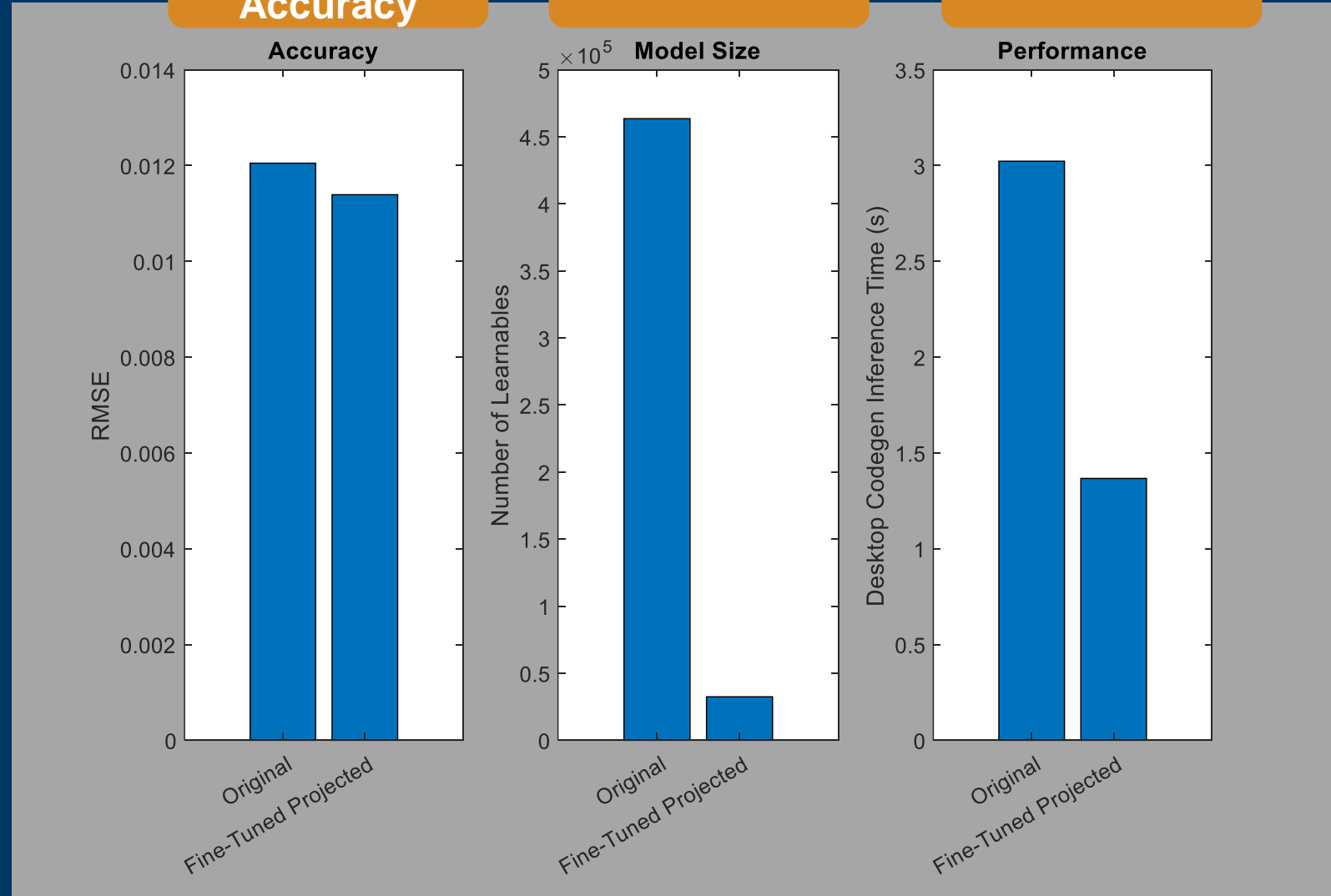
Fine tune using projection-based model compression techniques



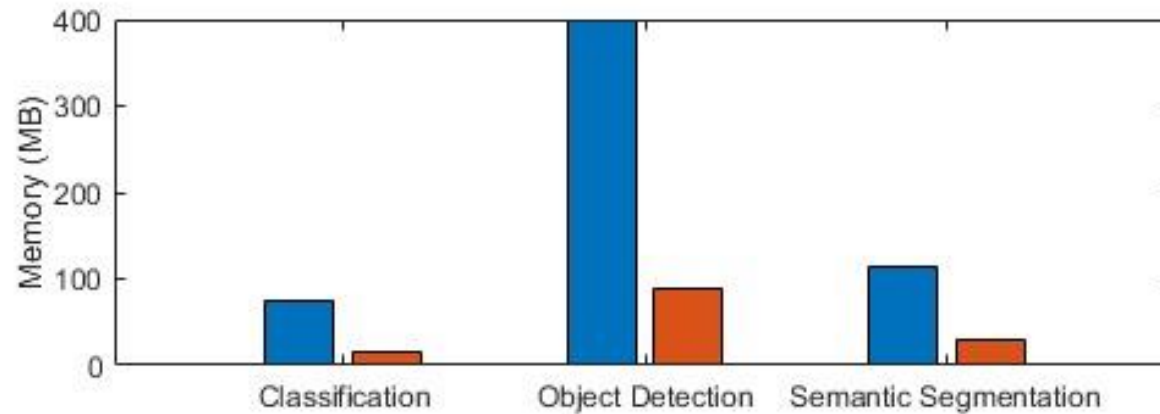
Similar Accuracy

93% Smaller

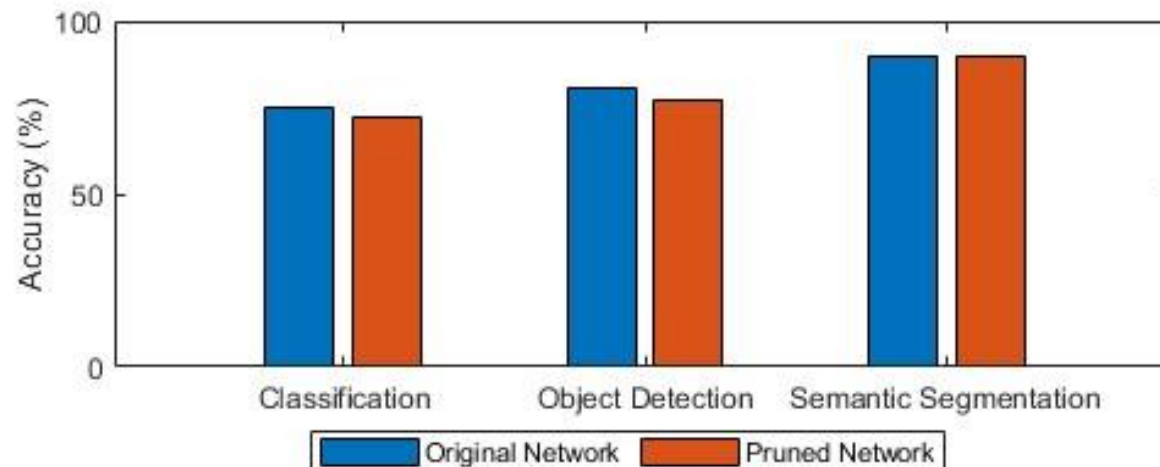
Faster



Apply pruning to a variety of AI model types to reduce memory footprint



<5% Accuracy Reduction



Pruning
~4x Memory Reduction

Equivalent Accuracy

Quantize deep networks to 8-bit scaled integer data types

DEEP NETWORK QUANTIZER

Calibration Data:

Validation Data:

FILE CALIBRATE QUANTIZE VALIDATE EXPORT

net - Layer Graph

Layer Graph showing a neural network structure with layers: data, conv1, relu_conv1, pool1, fire2-sque..., fire2-relu..., fire2-expa..., fire2-relu..., fire2-concat, fire3-sque..., fire3-relu..., fire3-expa..., fire3-relu...

Dynamic Range Statistics

Layer Name	Min Value	Max Value	Quantized
data			<input type="checkbox"/>
Activations	0.0000	255.0000	
data_normalization			<input type="checkbox"/>
Activations	-124.4884	171.6490	
conv1			<input checked="" type="checkbox"/>
Activations	-850.5910	728.3384	
Weights	-0.9198	0.8849	

Dynamic Range of Calibrated Layers

Validation Summary

Validation Results

Number of samples: 20

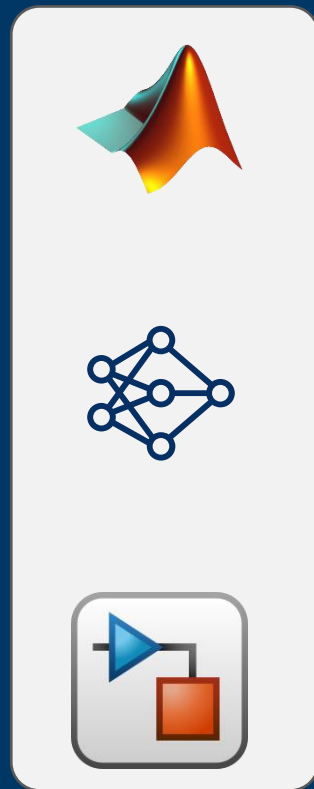
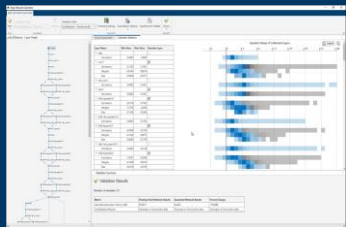
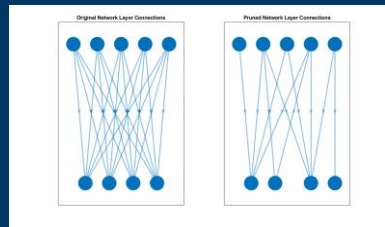
Metric	Floating-Point Network Results	Quantized Network Results	Percent Change
Learnable parameter memory (MB)	2.9003	0.7339	-74.6943
Top-1 Accuracy	1.0000	1.0000	0.0000

Deploy compressed model using automatic code generation

Model compression
(pruning, projection
and quantization)

Generate code for AI
model + pre- and post-
processing

Performant multi-processor deployment




Code
Generation

CPU


Any CPU
No Library needed


oneDNN
Library


ARM Compute
Library

GPU


NVIDIA

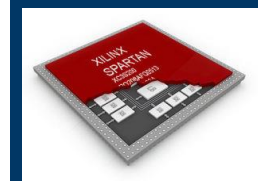
μC


NXP

life..augmented

 **TEXAS
INSTRUMENTS**

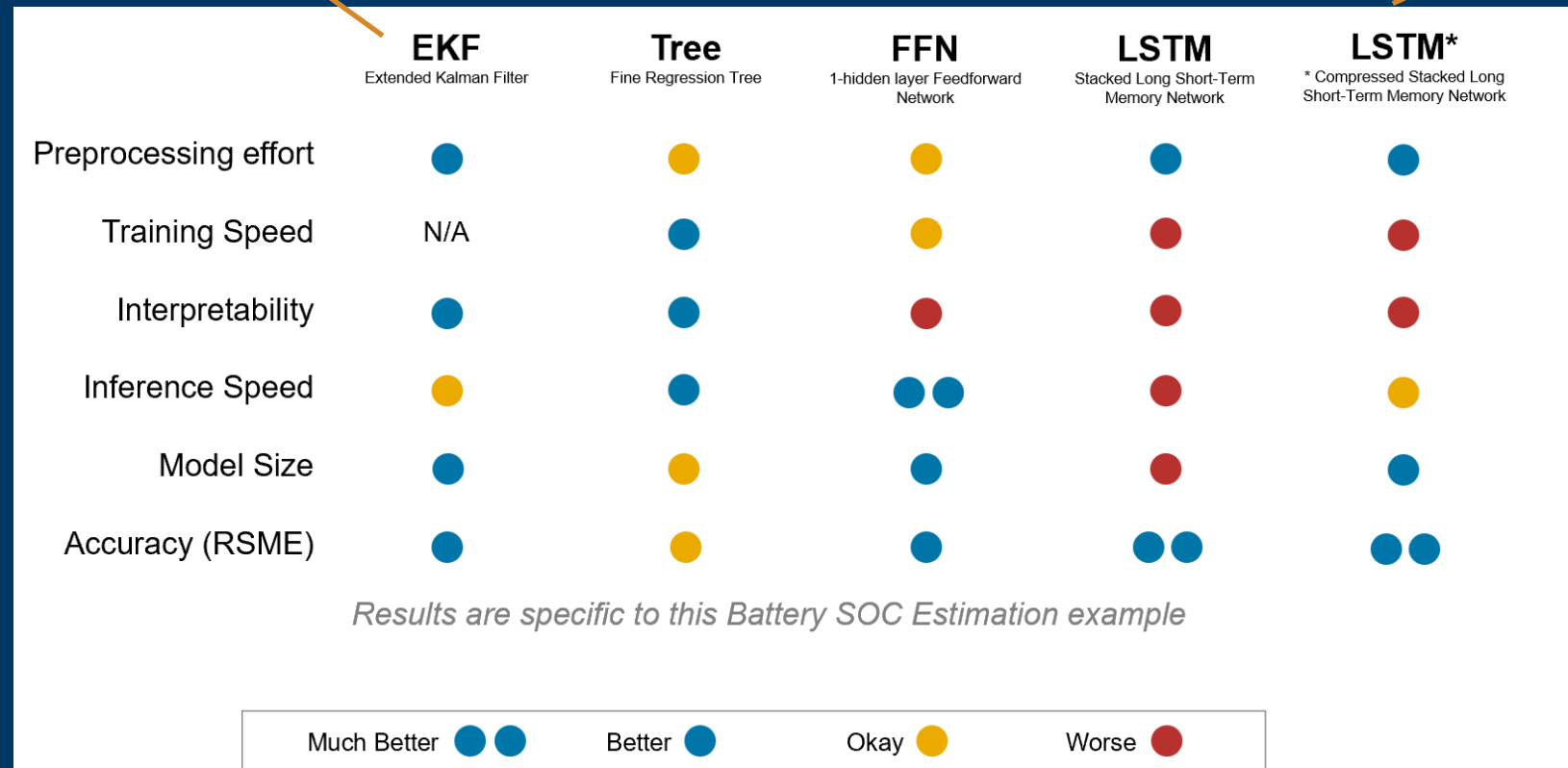
FPGA



Perform trade-off analysis between AI and traditional techniques

Kalman Filter

AI Model



Trade-Off Analysis of Different Models (Non-AI vs. Different AI Models)

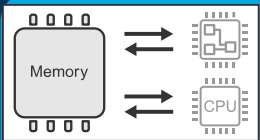
Questions and concerns we need to answer to build confidence



AI Modeling



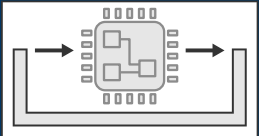
System-level Simulation



Memory and Execution Time



Verification and Validation



Testing on Hardware

5 keys to building confidence

1. Access and explore state of art AI models
2. System-level testing to assess impact of adding AI component
3. Model compression, trade-off studies, and leverage automatic code generation

Neural networks can misclassify inputs due to small imperceptible changes

Bicycle



+

δ

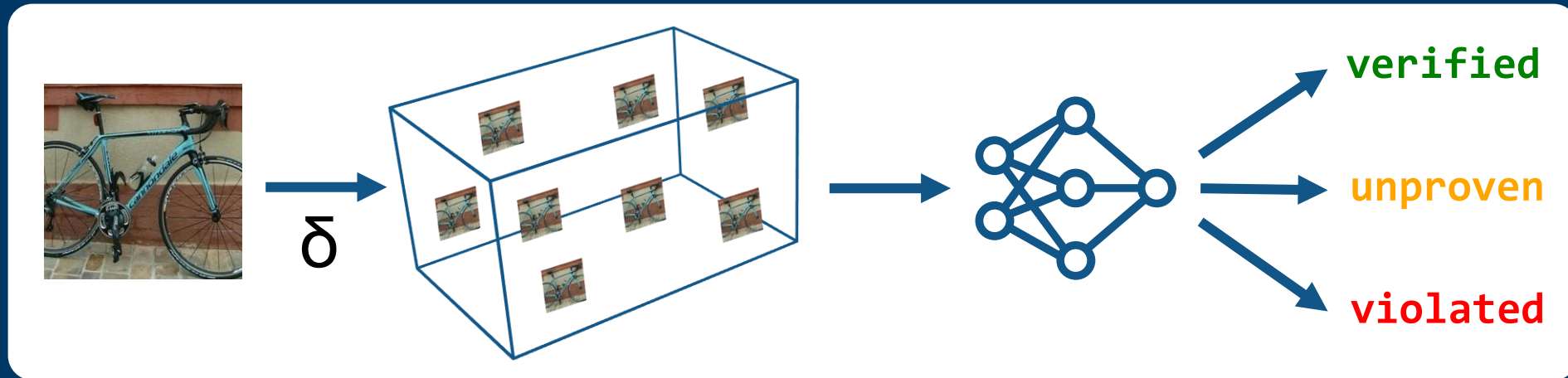


=

Pole



Neural network verification provides formal evidence that your model behaves as intended



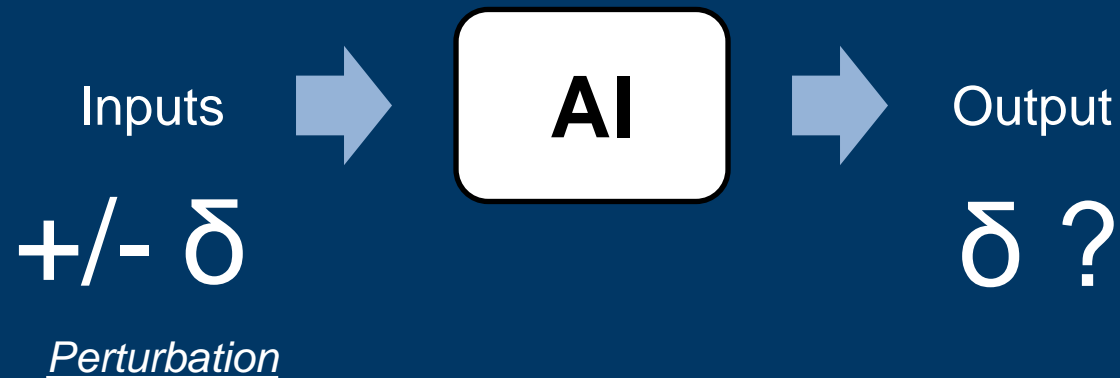
Deep Learning Toolbox Verification Library

by MathWorks Deep Learning Toolbox Team **STAFF**

Verify and test robustness of deep learning networks

Perform formal verification to test robustness

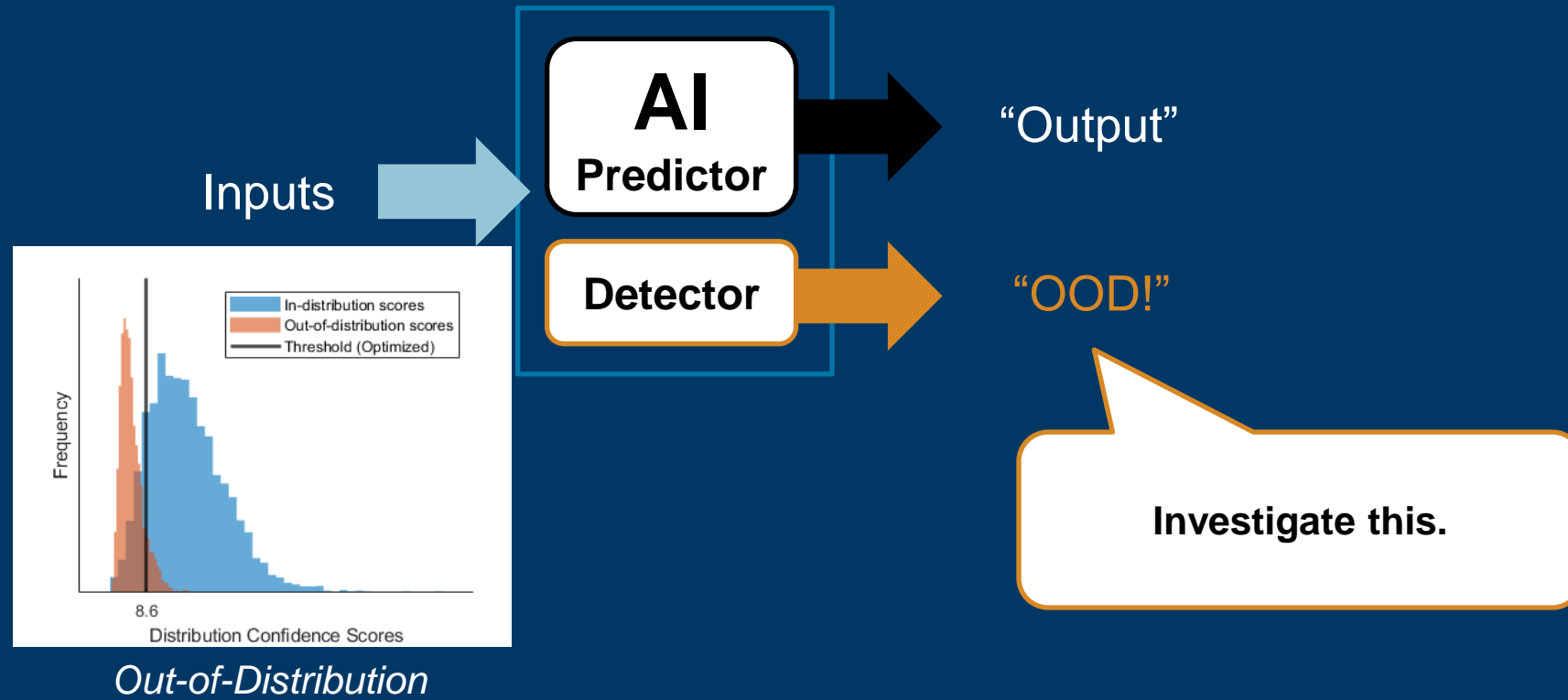
Robustness against Input Perturbation



```
d1X = dlarray(X, 'BC');  
XLower = d1X-perturbation;  
XUpper = d1X+perturbation;  
dlnet = dlnetwork(trainedNetwork.Layers(1:end-1));  
[YLower, YUpper] = estimateNetworkOutputBounds(dlnet, XLower, XUpper);  
YLower = extractdata(YLower)';  
YUpper = extractdata(YUpper)';  
maxDeviation = max([abs(YLower-Y_pred); abs(YUpper-Y_pred)]);
```

Check for out of distribution values to ensure safety

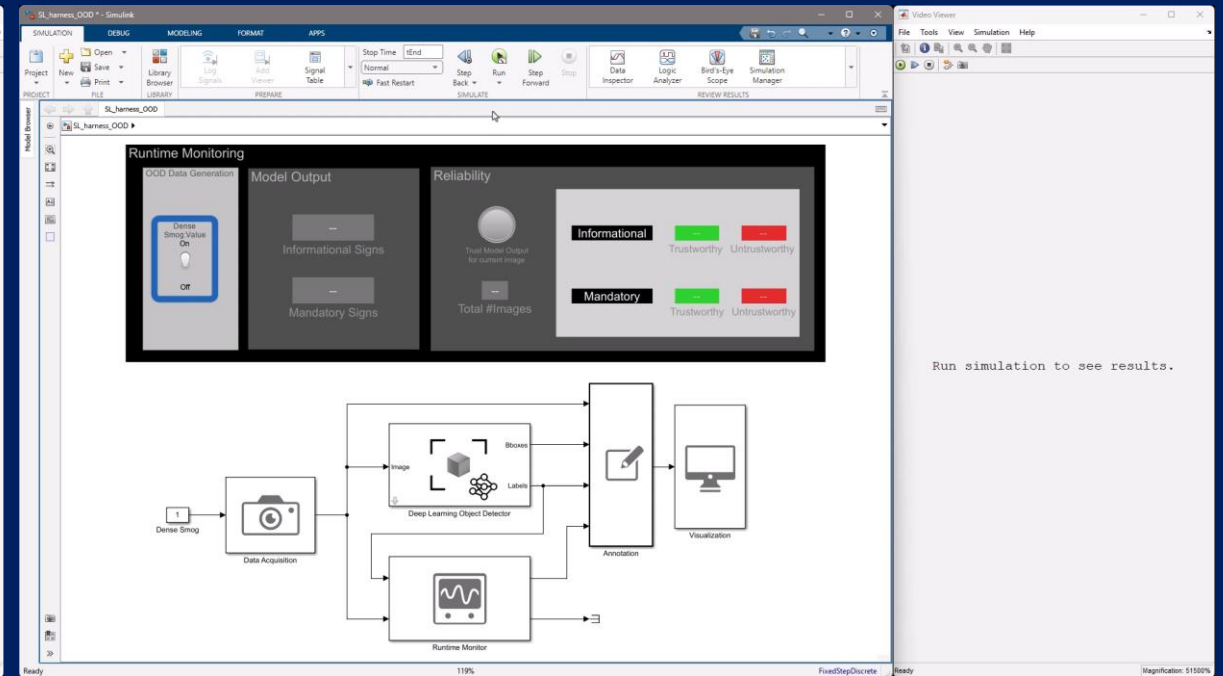
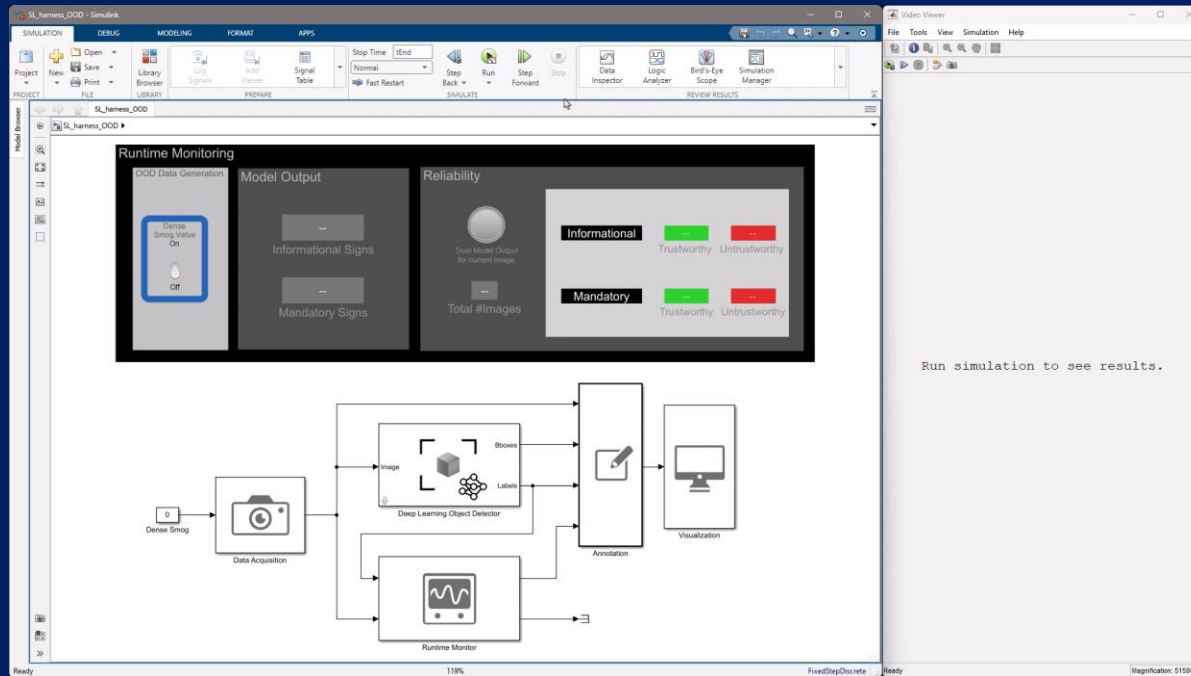
Out-of-Distribution Detection



Integrate and test your AI model with Simulink with a Runtime Monitoring System

In-distribution ✓

Out-of-distribution ✗



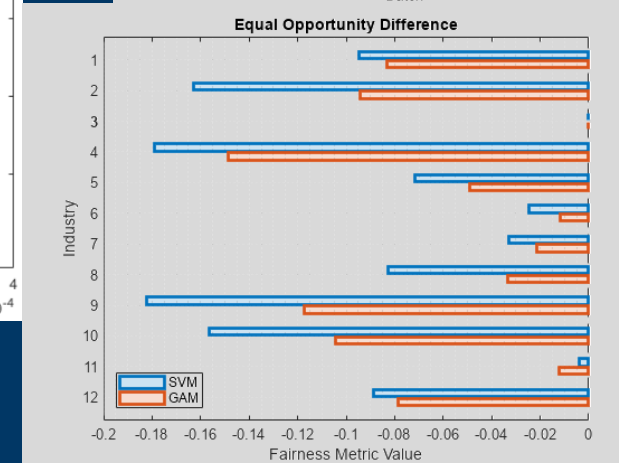
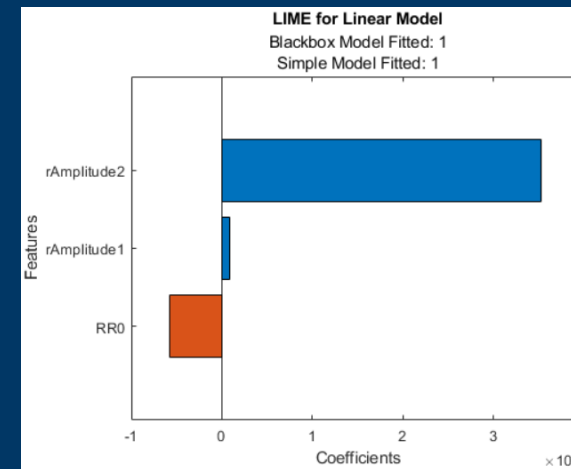
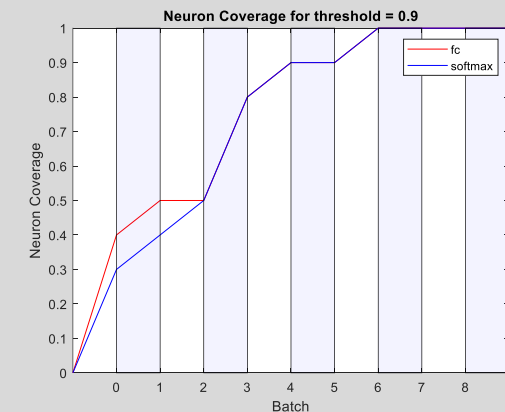
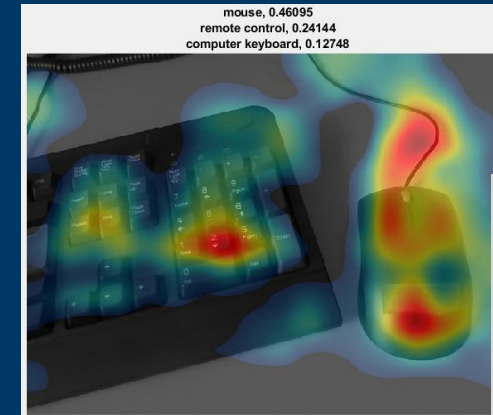
Explain & Visualize how a Model Works

There are visualization techniques to investigate and explain model predictions.

Which features did a model use to make a prediction?

What are the main features found in each of the layers?

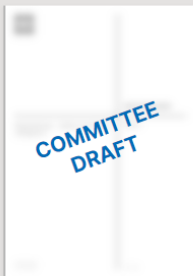
Class Activation Mapping	R2019a
Occlusion	R2019b
Local Interpretable Model-agnostic Explanation (LIME)	R2020b
GRAD-CAM, Shapley Values	R2021a
GRAD-CAM for 1-D	R2022b
Fairness Metrics, Neuron Coverage	R2022b
Bias mitigation using fairness thresholder for binary classification	R2023a
Fairness metrics comparisons	R2023a
Multi-point Shapley values/plots	R2024a
Explain object detectors using D-RISE	R2024a



There is an increased focus in government regulation and certification efforts



Automotive



ISO/CD PAS 8800

Road Vehicles — Safety and artificial intelligence

Under development

A draft is being reviewed by the committee.



Aerospace

WIP 2023-06-26



Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI ARP6983

This document discusses guidelines for the development of Aircraft Systems leveraging AI capabilities, taking into account the overall aircraft operating environment and functions. This includes validation of requirements and verification of the design implementation for certification and product assurance and guidelines with the assessment of safety. It provides practices for showing compliance with the regulations and serves to assist a company in developing and meeting its own internal standards by considering the guidelines herein.



Medical Devices

← [Software as a Medical Device \(SaMD\)](#)

Artificial Intelligence and Machine Learning in Software as a Medical Device

May 13, 2024 update: 191 Artificial Intelligence and Machine Learning (AI/ML)-Enabled Medical Devices were added to the list below. With this update, the FDA has authorized 882 AI/ML-enabled medical devices. Of those newly added to the list, 151 are devices with final decision dates between August 1, 2023, and March 31, 2024, and 40 are devices from prior periods identified through a further refinement of methods used to generate this list.

European Parliament

2019-2024

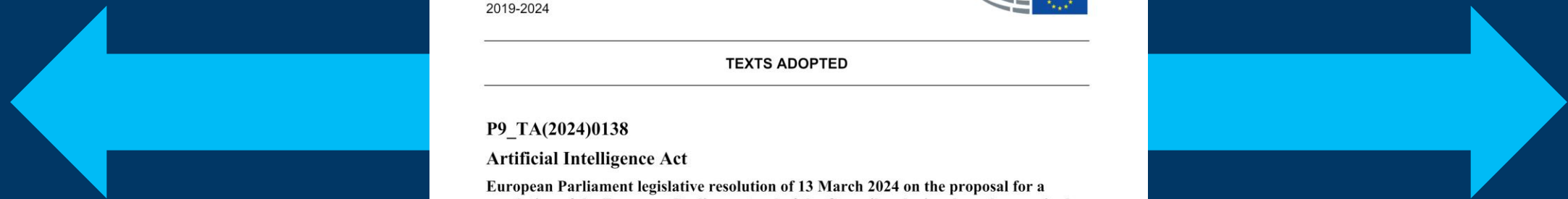


TEXTS ADOPTED

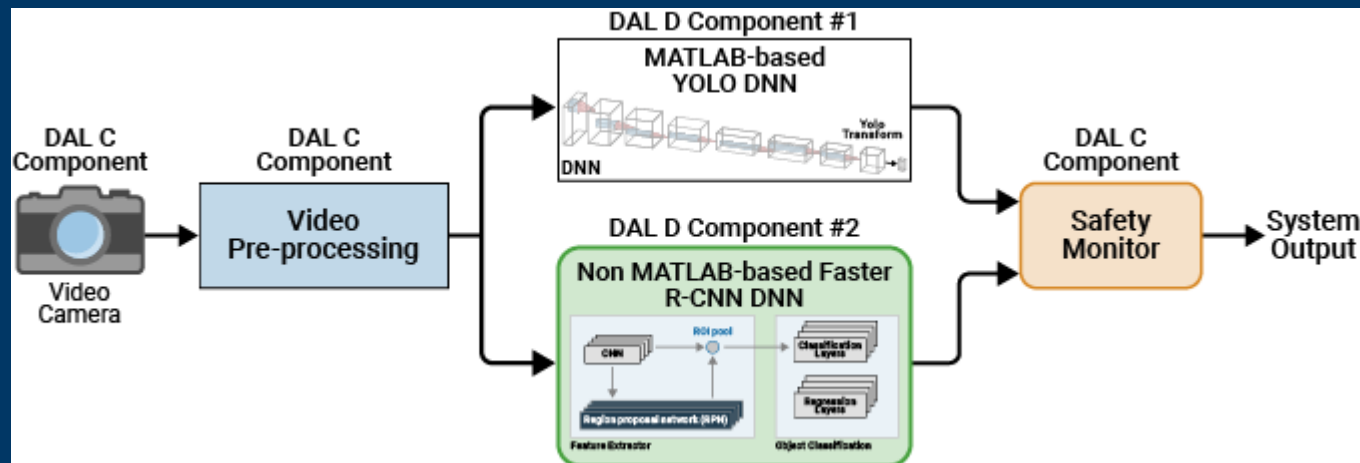
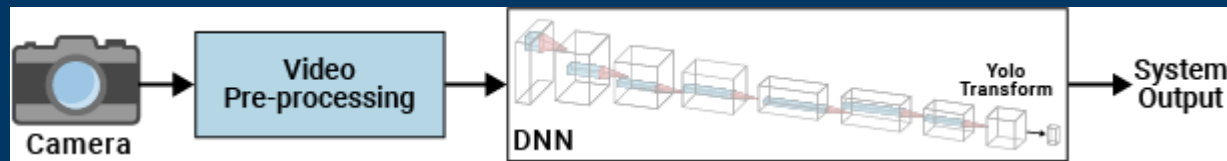
P9_TA(2024)0138

Artificial Intelligence Act

European Parliament legislative resolution of 13 March 2024 on the proposal for a regulation of the European Parliament and of the Council on laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts (COM(2021)0206 – C9-0146/2021 – 2021/0106(COD))



Published first reference application on developing system with AI component to comply with DO-178



Runway Sign Classifier: Certify an Airborne Deep Learning System

Demonstrates the certification of airborne deep learning system.

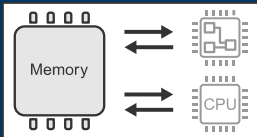
Questions and concerns we need to answer to build confidence



AI Modeling



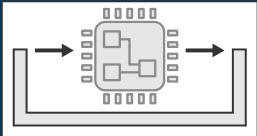
System-level Simulation



Memory and Execution Time



Verification and Validation

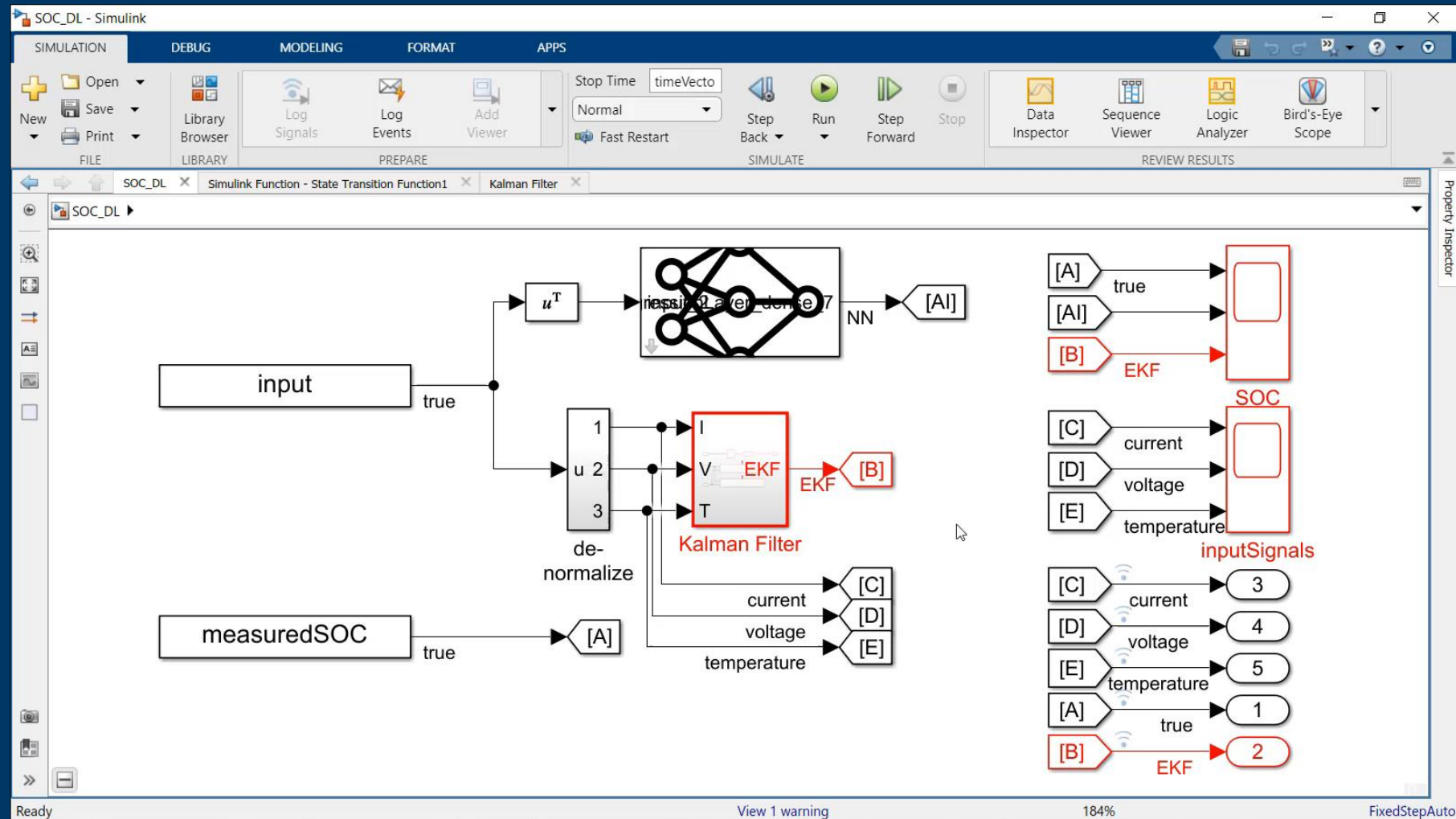


Testing on Hardware

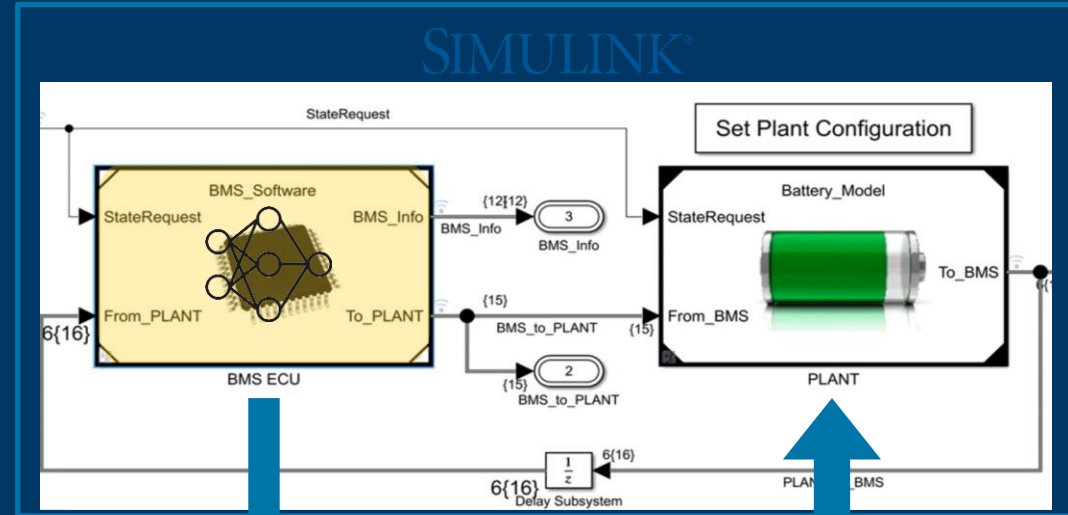
5 keys to building confidence

1. Access and explore state of art AI models
2. System-level testing to assess impact of adding AI component
3. Model compression, trade-off studies, and leverage automatic code generation
4. Verify robustness and test for out of distribution

Generate Library-Free C/C++ Code for AI Models



Processor-in-the-Loop Testing on ARM Cortex-M7 Processor



Code generation from algorithm

Deployed code communicates with simulated plant

NXP S32K344 board



Arm® Cortex®-M

Processor-in-the-Loop Testing on ARM Cortex-M7 Processor

The screenshot displays the MATLAB/Simulink environment for Processor-in-the-Loop (PIL) testing. On the left, a photograph shows the physical hardware board. The main workspace contains a Simulink model with an 'SOC Estimation (PIL)' block. A 'Download finished' dialog box indicates the executable file has been downloaded to the S32K344 board. The 'inputSignals' block shows connections for true SOC, current, voltage, and temperature. The 'Diagnostic Viewer' at the bottom shows the following terminal output:

```

165004 3008 27096 195188 2fa74 ./SOC_Estimation.elf
### Done invoking postbuild tool.
### Invoking postbuild tool "ELF To Binary Converter" ...
arm-none-eabi-objcopy -O binary ./SOC_Estimation.elf ../.././SOC_Estimation.bin
### Done invoking postbuild tool.
### Successfully generated all binary outputs.

C:\Users\jgazarr\OneDrive - MathWorks\Work\Projects\AI_MBD\SOC_Estimation\work\slprj\vert\SOC_Estimation\pil>exit /B 0
### Updating code generation report with PIL files ...
### Starting application: 'work\slprj\vert\SOC_Estimation\pil\SOC_Estimation.elf'
    
```

On the right, three plots are shown: 'true, estim' (SOC), 'current', 'voltage', and 'temperature'. The top plot shows true SOC (blue) and estimated SOC (red) over time. The middle plot shows current (yellow) and the bottom plot shows voltage (yellow) and temperature (green) signals.

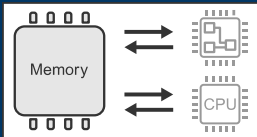
Questions and concerns we need to answer to build confidence



AI Modeling



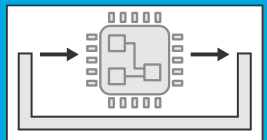
System-level Simulation



Memory and Execution Time



Verification and Validation



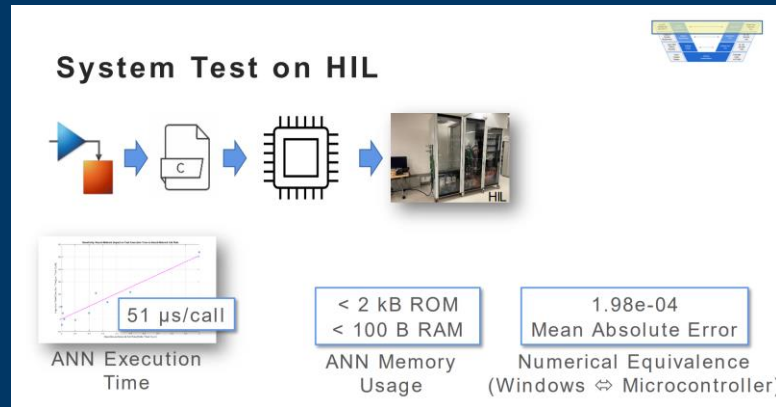
Testing on Hardware

5 keys to building confidence

1. Access and explore state of art AI models
2. System-level testing to assess impact of adding AI component
3. Model compression, trade-off studies, and leverage automatic code generation
4. Verify robustness and test for out of distribution
5. Perform PIL and HIL testing

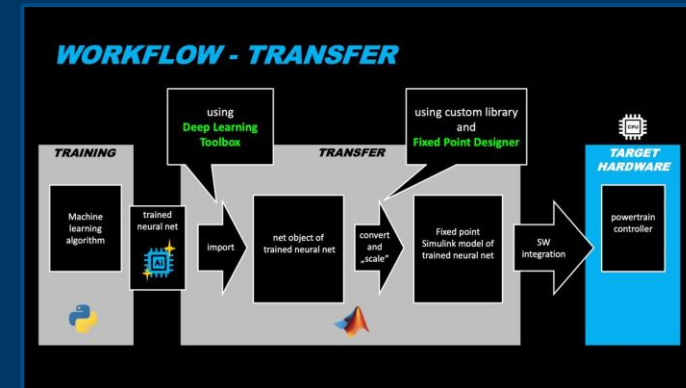
Learn about other automotive uses of AI virtual sensors

Gotion



Battery pack SOC estimation with neural network

Mercedes-Benz

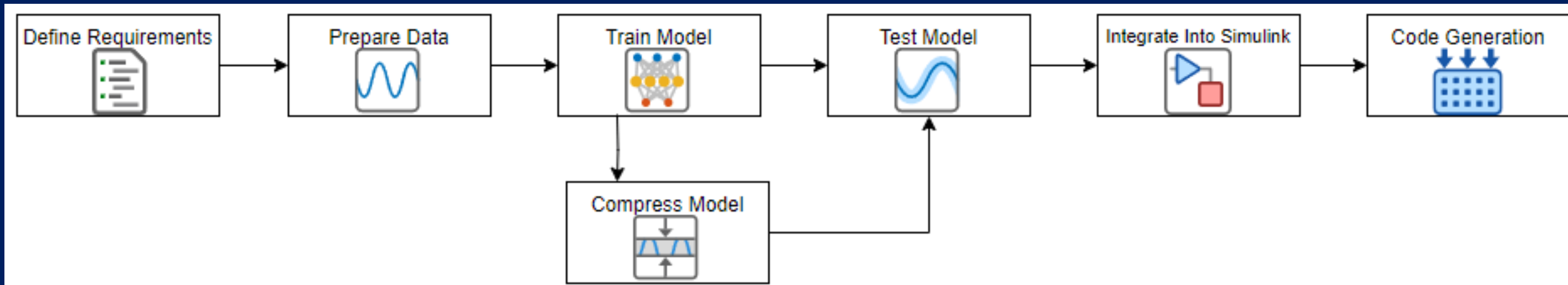


Engine piston pressure estimation with deep neural network

5 keys to building confidence through design for AI-driven systems

1. Access and **explore state of art models**
2. **System-level simulation** to test impact of adding AI component
3. Model **compression, trade-off studies**, and leverage automatic **code generation**
4. Verify **robustness** and test for **out of distribution**
5. Perform **PIL and HIL** testing

Get started quickly with end-end reference examples



This example shows how to perform these steps:

1. [Define Requirements for Battery State of Charge Estimation](#)
2. [Prepare Data for Battery State of Charge Estimation Using Deep Learning](#)
3. [Train Deep Learning Network for Battery State of Charge Estimation](#)
4. [Compress Deep Learning Network for Battery State of Charge Estimation](#)
5. [Test Deep Learning Network for Battery State of Charge Estimation](#)
6. [Integrate AI Model into Simulink for Battery State of Charge Estimation](#)
7. [Generate Code for Battery State of Charge Estimation Using Deep Learning](#)

How Generative AI will Impact Engineering Workflows

Augment Existing Workflows

- Learn while doing
- Create code, analyses, models, etc. using NLP
- Check, verify, validate



Empower MATLAB and Simulink Users to Build

- Access popular models for text, images, video, etc.
- Build custom transformer models
- Easy from options from platforms like Hugging Face



Generative AI in Engineered Systems

- Apply LLM innovations to time-series sensor data
- Real-time and near-real-time systems
- Safety-critical



Now, Near-term

Future

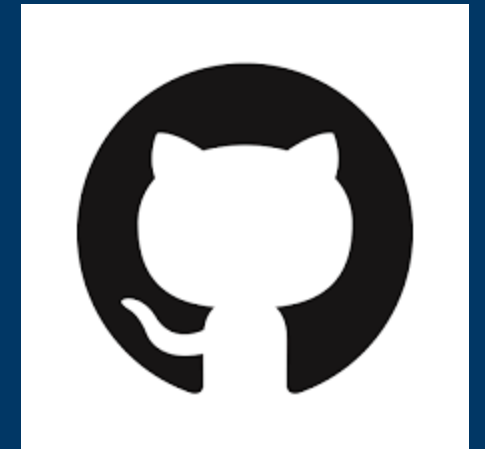
MATLAB AI Chat Playground

- Experiment with an AI assistant alongside MATLAB
- Generate first-draft MATLAB code or ask questions
- Powered by the ChatGPT API's

The screenshot shows the MATLAB AI Chat Playground web interface. The browser address bar displays `mathworks.com/matlabcentral/playground/new`. The page title is "AI Chat Playground". A "Clear" button is visible at the top left. The main chat area shows a message from "AI" with a plus icon: "Hello! Whether you're checking out MATLAB for the first time or have been using it for years, I'm here to answer your questions and provide coding tips." Below this is a warning box: "Please keep in mind that AI sometimes writes code and text that seems accurate, but isn't. AI does not yet have knowledge of features delivered after June 2021 and only limited knowledge of Simulink and specialized toolboxes. This is a space for experimentation. Try it, verify any resulting code, and kindly give thumbs up or thumbs down on the results to help improve the responses." Three suggested prompts are listed: "Determine whether a matrix is sparse", "Create some data, construct a grid of query points, interpolate on the grid, and plot the results", and "Highlight contours at particular levels". A "Shuffle" button is located to the left of the second prompt. At the bottom, there is a text input field with the placeholder "Send a message..." and a send button. A footer note states: "Generated code and text might be inaccurate. Validate before use. [About](#)". On the right side, there is a code editor area with "CODE" and "RUN" tabs and a green play button icon.

Use Large Language Models with MATLAB

- GitHub Repo: [llms-with-matlab](#) within matlab-deep-learning
- Connect to models via the OpenAI™ API and the Azure® OpenAI Service
- Connect to local and remote models hosted with Ollama
- Examples of chatbots, text summarization, retrieval augmented generation, image generation



Let us catch up after session to discuss MathWorks investments

Please join our other AI sessions today

MathWorks **AUTOMOTIVE CONFERENCE 2024** India

Register at mathworks.com/indiamac
#MathWorksAutoConf

ServiceSage : A Gen AI-Based RCA Chat Assistant

19 November | Pune



Bhakti Kalghatgi,
Tata Motors



Shubham Gupta,
Tata Motors



MathWorks **AUTOMOTIVE CONFERENCE 2024** India

Register at mathworks.com/indiamac
#MathWorksAutoConf

Embedded AI for Body Applications with MATLAB and Simulink

19 November | Pune



Athulya Thazha,
Mercedes-Benz Research and Development
India Private Limited



Shubham Kale,
Mercedes-Benz Research and Development
India Private Limited



MathWorks **AUTOMOTIVE CONFERENCE 2024** India

Register at mathworks.com/indiamac
#MathWorksAutoConf

Tech Talk: Accelerating AI Adoption: From Design to Deployment in Mobility

19 November | Pune



Myrtle Bini Rajendrababu,
MathWorks



Jayanth Balaji Avanashilingam,
MathWorks



Koustubh Shirke,
MathWorks



Nikita Pinto,
MathWorks



MathWorks
**AUTOMOTIVE
CONFERENCE 2024**

Thank you

