MATLAB EXPO
INDIA

11 July 2024, Bengaluru

The Industrial AI Lifecycle

Dreaming, Designing, and Delivering in the Digital Age

Koustubh Shirke

Sammit Jain

Monalisa Pal

Nikita Pinto

Jayanth Balaji

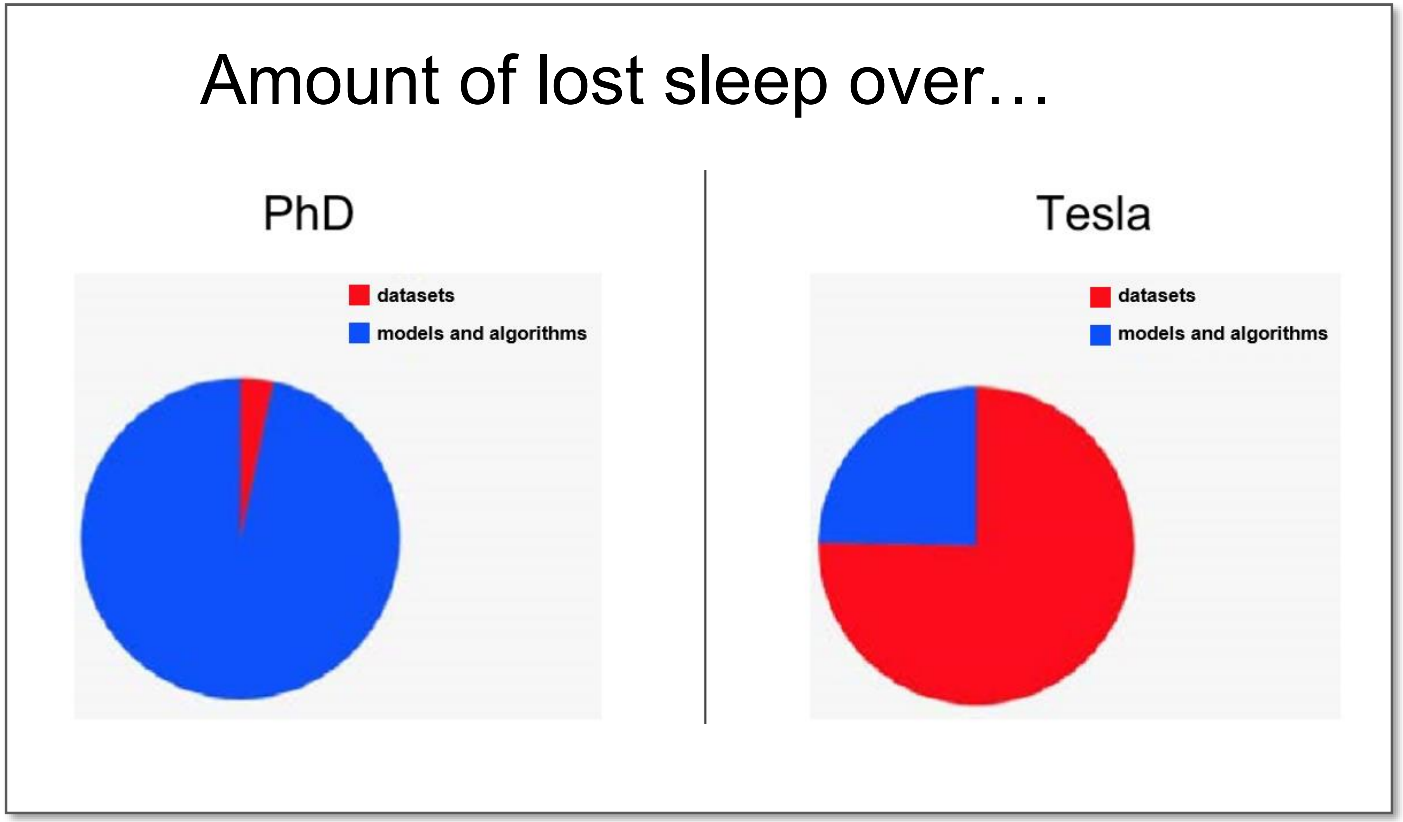
Peeyush Pankaj **(Moderator)**

MathWorks

Approaching (Almost) Any Data Science Project

Data Preparation	AI Modeling	Simulation & Test	Deployment
Data cleansing and preparation	Model design and tuning	Integration with complex systems	Embedded devices
Human insight	Hardware accelerated training	System simulation	Enterprise systems
Simulation-generated data	Interoperability	System verification and validation	Edge, cloud, desktop

Data preparation represents most of your AI effort

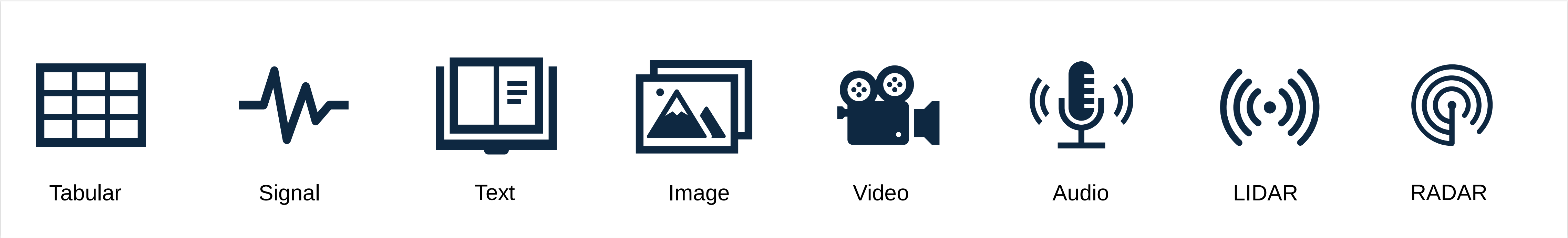


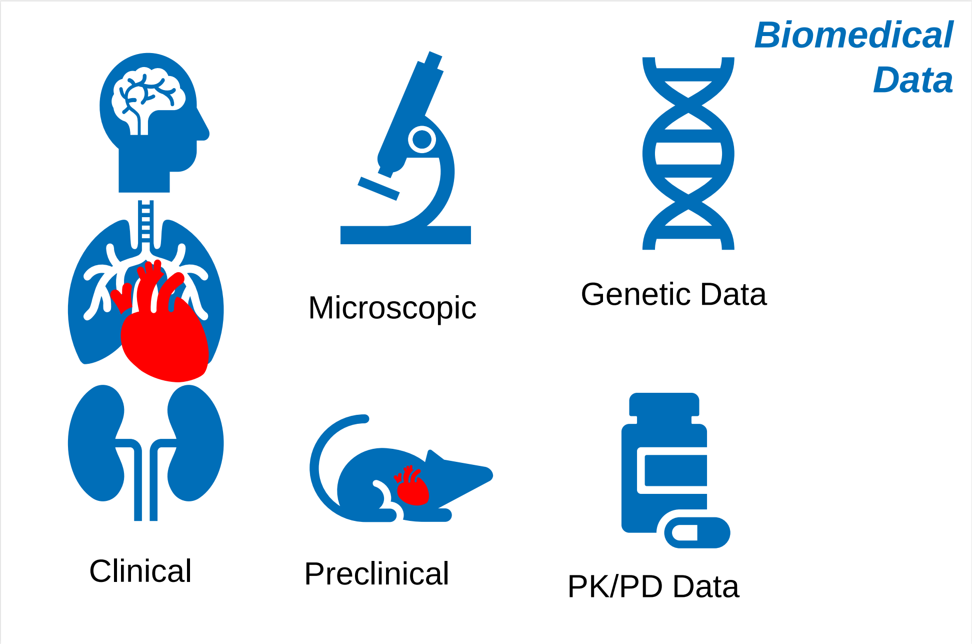
Source: Andrej Karpathy’s slide from TrainAI 2018

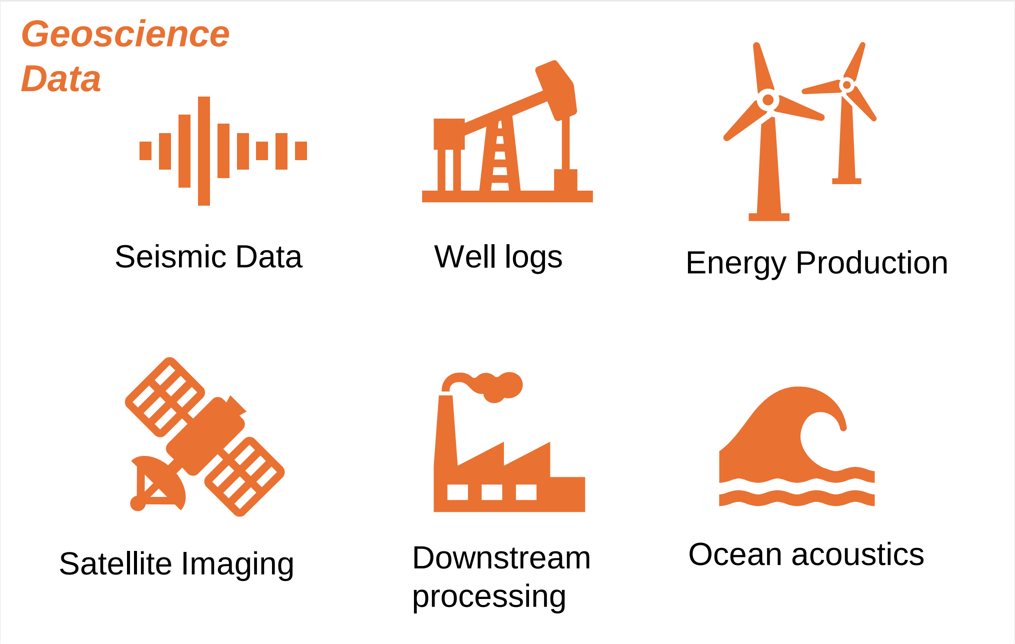
Challenges in data preparation

- Variety, velocity and volume of data
- Quality and quantity of data annotation
- Leveraging domain expertise
- Lack of data

MATLAB makes it easy to handle different data formats across industries



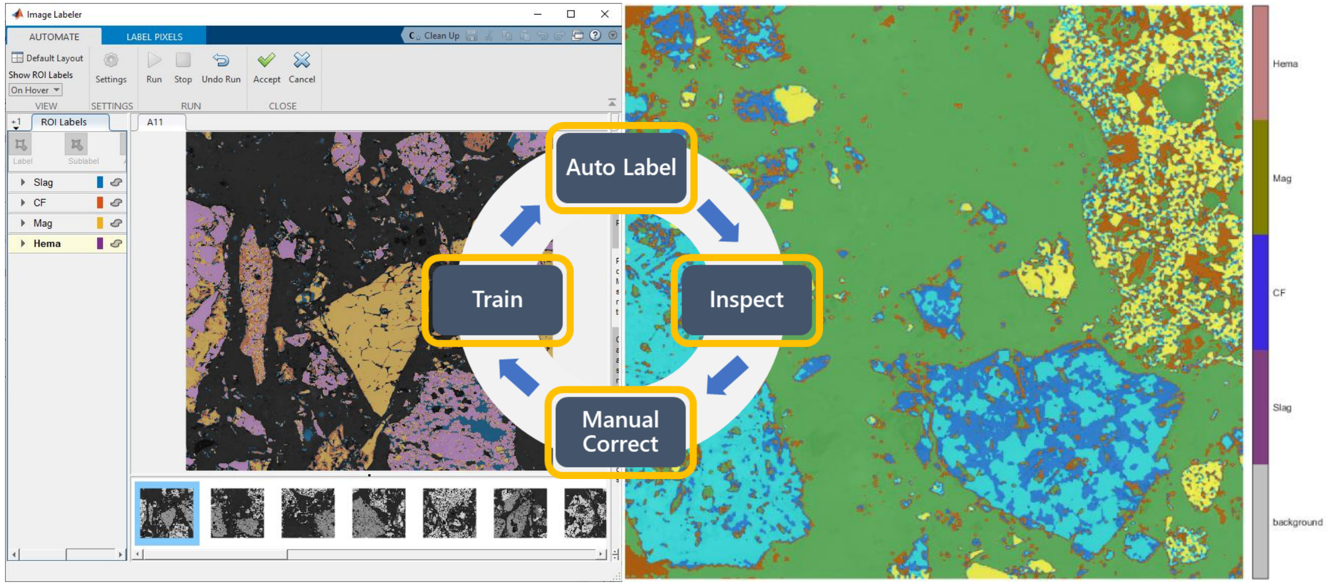




Reduce human supervision and development time

"Even though I had limited knowledge on Image processing and Deep Learning, I could successfully adopt deep learning for my project. With evaluation support from MathWorks, we could prototype our approach easily with limited time bound."

- Created a custom labeling algorithm for automatic labelling material
- Improved prediction accuracy using deep learning
- Partnered with MathWorks to leverage the full benefits of MATLAB



Automated labeling apps save you weeks to months

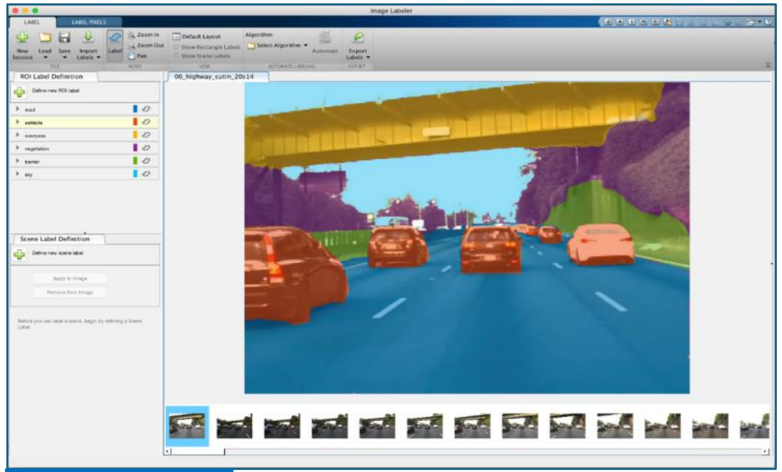
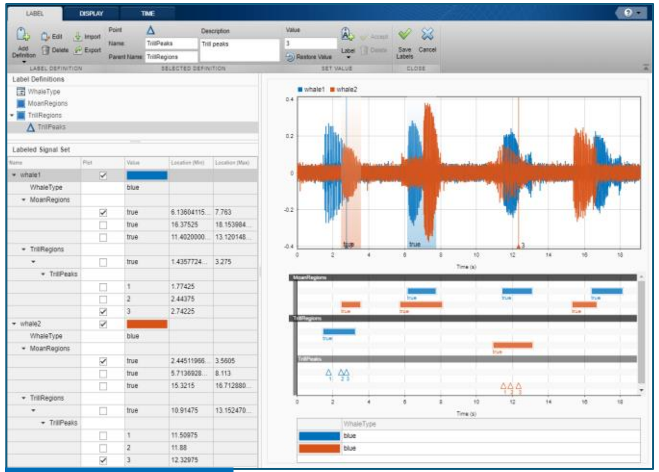
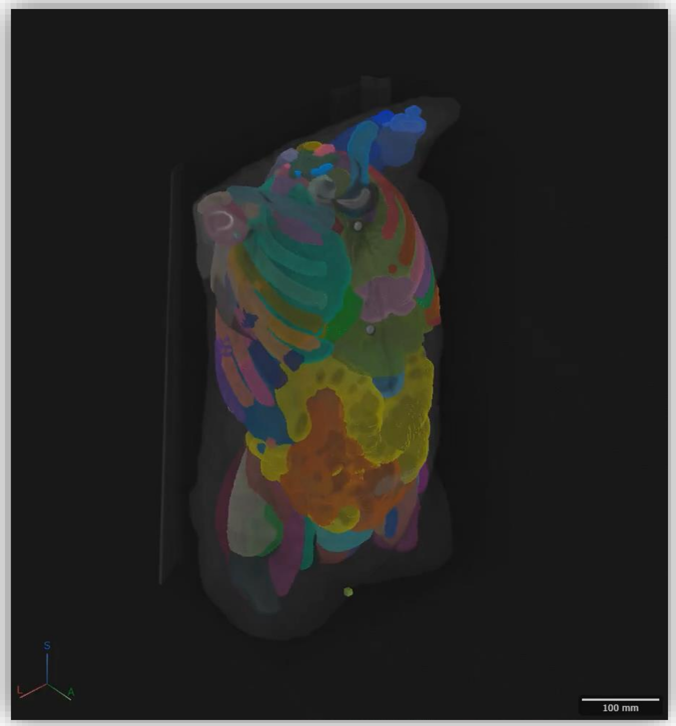
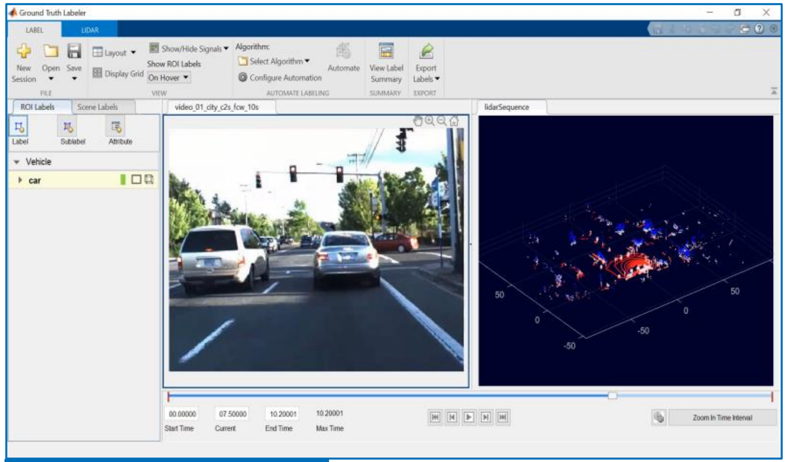


Image Labeler

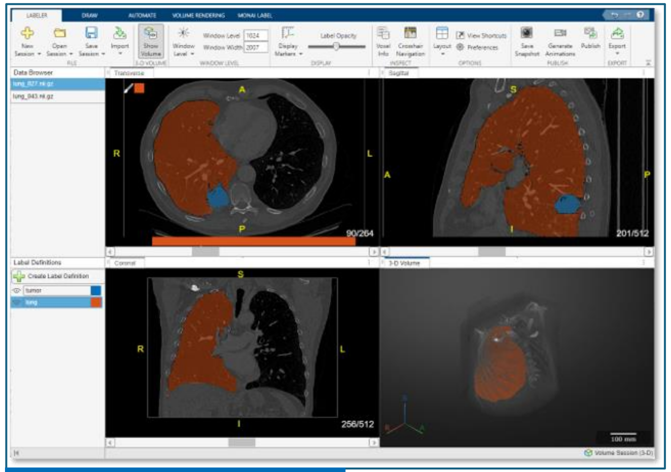


Signal Labeler



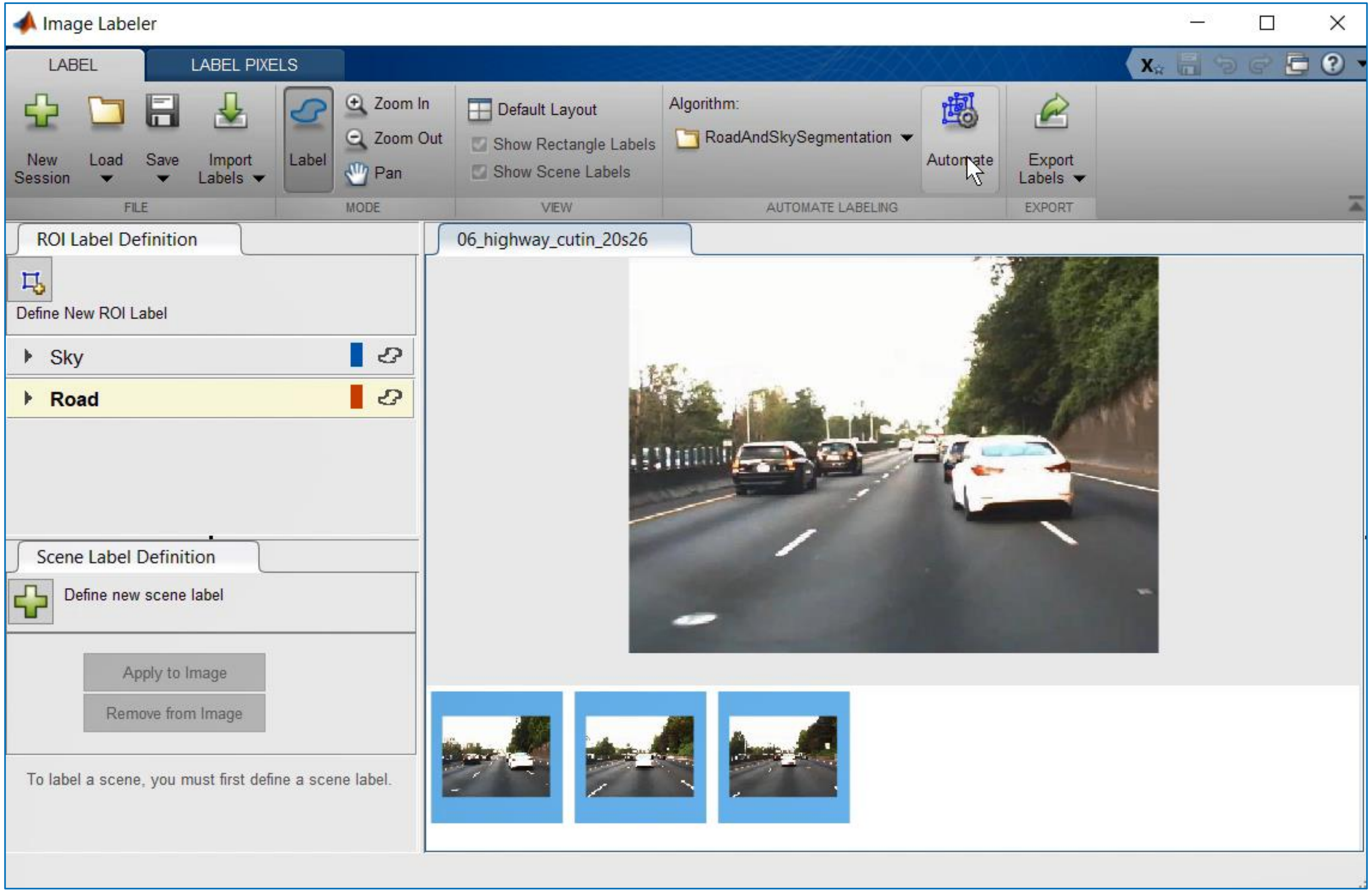


Ground Truth Labeler

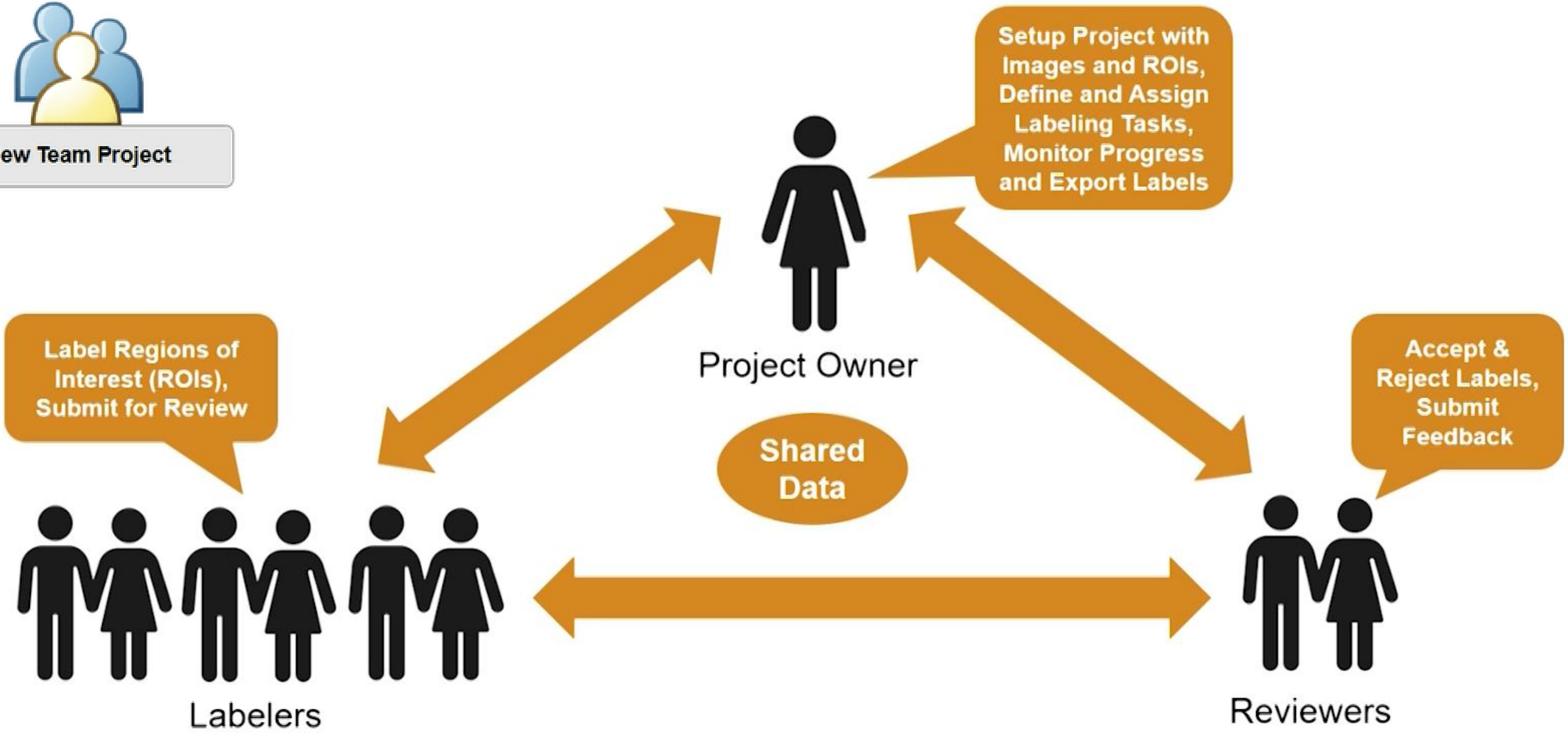


Medical Image Labeler

Label data faster with automated workflows

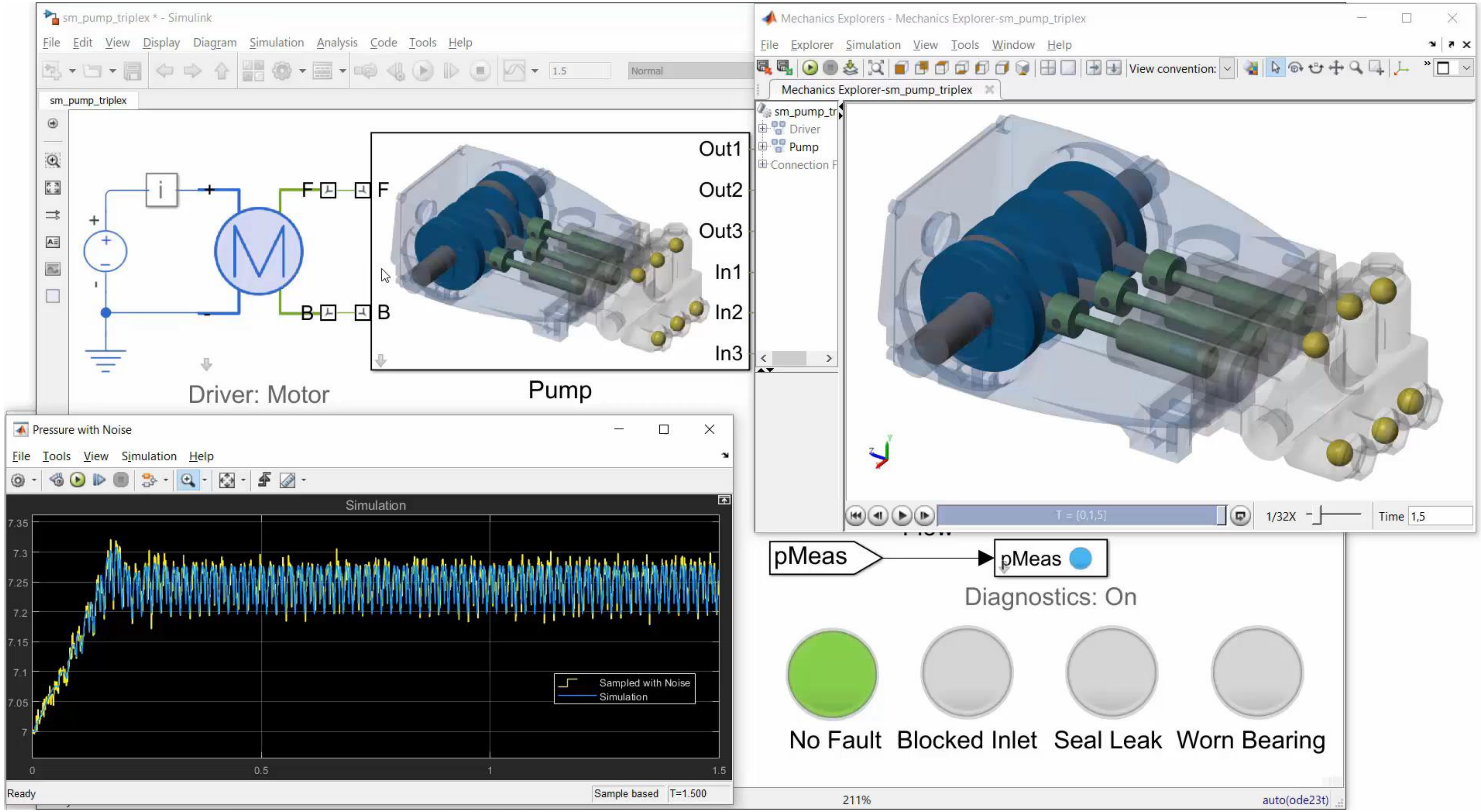


Label data faster with Team-based labeling



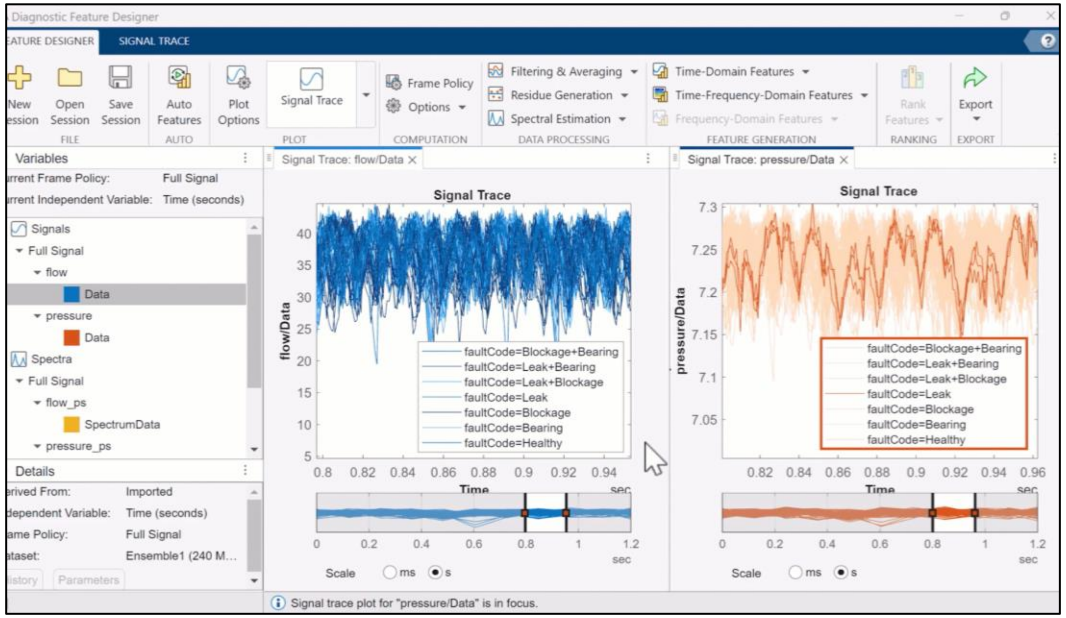
Big Data with No Data?

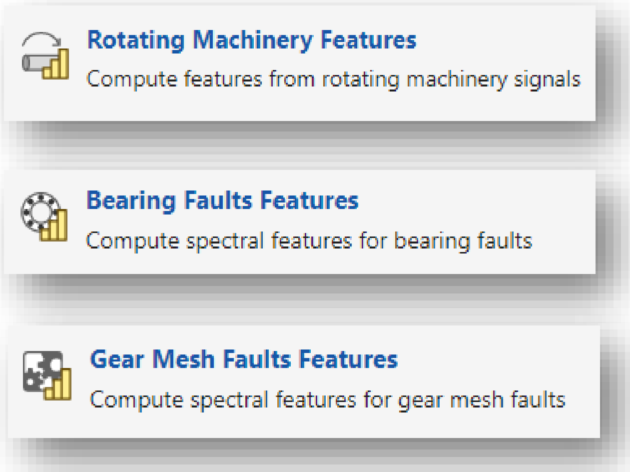
Simulate rare system failures to avoid them in the real world



App-based feature extraction and selection

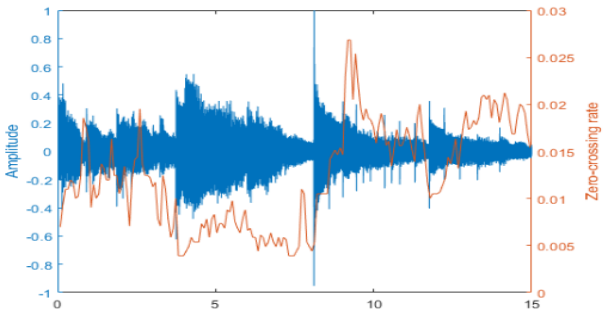
Diagnostic Feature Designer App



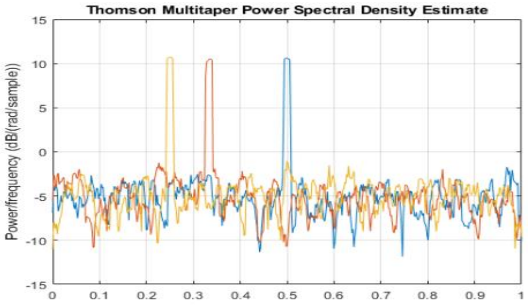


- Extract, visualize, and rank features
- Explore techniques without MATLAB coding
- Handle out-of-memory data
- Generate MATLAB code to automate tasks
- Physics-based features for rotating machines

Time Domain



Frequency-Domain



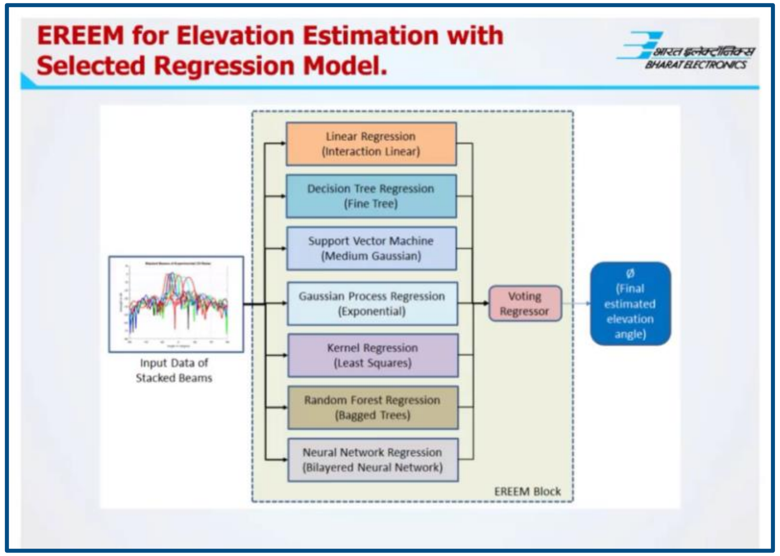
Bharat Electronics Applies AI to Elevation Estimations from 3D Radar

Using MATLAB, the Regression Learner app, and the Sensor Array Analyzer app, the Bharat Electronics team was able to model more accurate and robust predictions of target elevation angles from radar data.

Key Outcomes/Advantages:

- The Sensor Array Analyzer app enabled custom sensor array design and visualization without requiring extra time to code complex simulations for generating data sets to train AI models
- The curve-fitting tool in MATLAB simplified the process of calculating elevation angle estimates and delivered a more accurate result
- The Regression Learner app evaluated data with multiple regression methods in order to find the best fit for accurate predictions

Link to user story



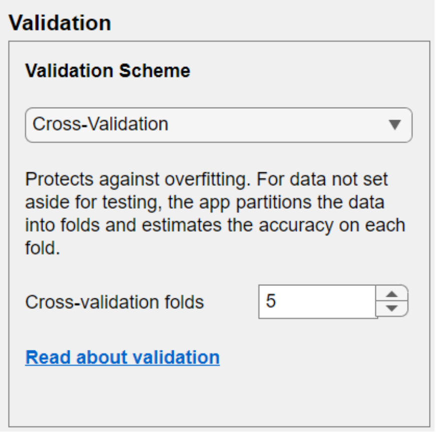
Complete workflow for the elevation estimation with a selected regression model.

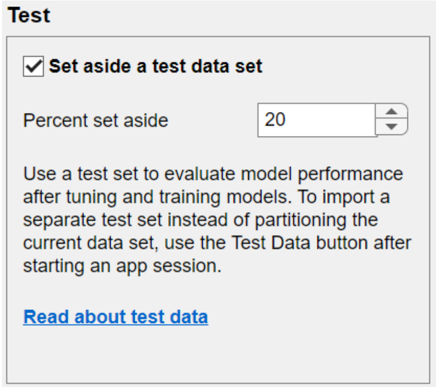
"With the help of AI, a lot more can be done. We have found that if more data is not available, then simulated data can also be generated with the help of MATLAB."

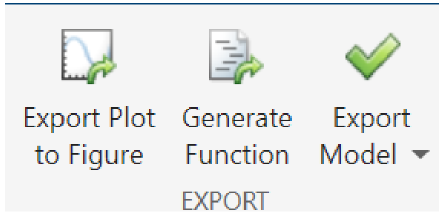
- Ram Pravesh, Bharat Electronics Limited

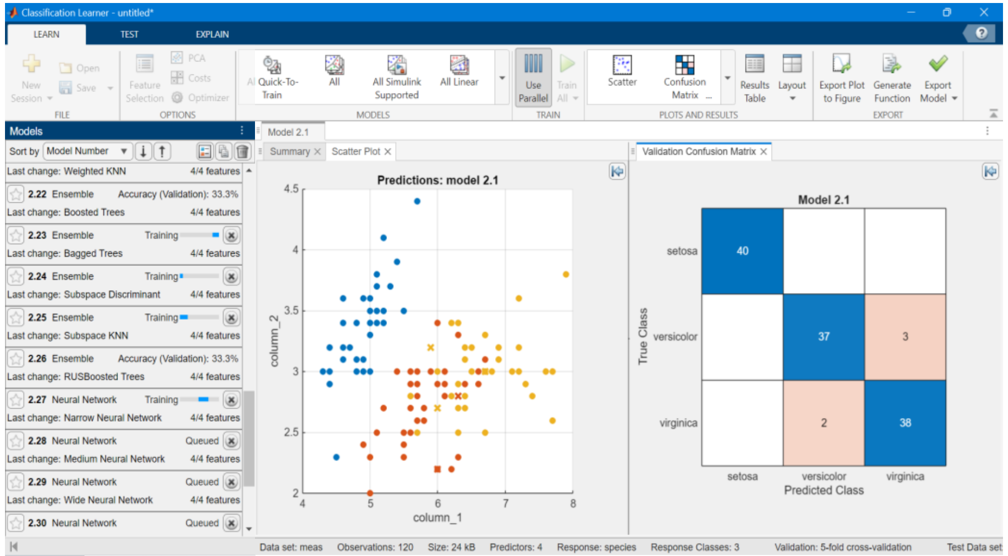
Interactive Apps for AI Workflow in MATLAB

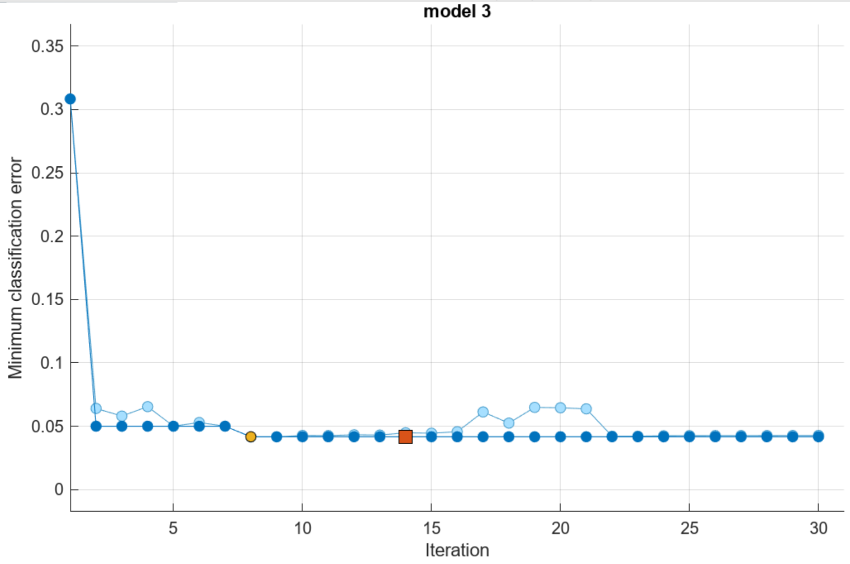
- Split Data
- Visualize Data
- Train and Compare Models
- Tune hyperparameters
- Export Trained Model or Generate a Function











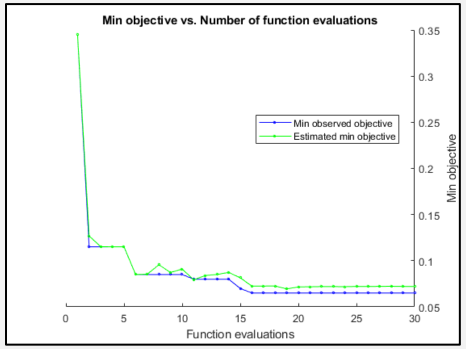
Improve Productivity with AutoML

- Productivity: build accurate models faster
- Narrow skill gap between engineers and data scientists
- Validate success of your “traditional” model building

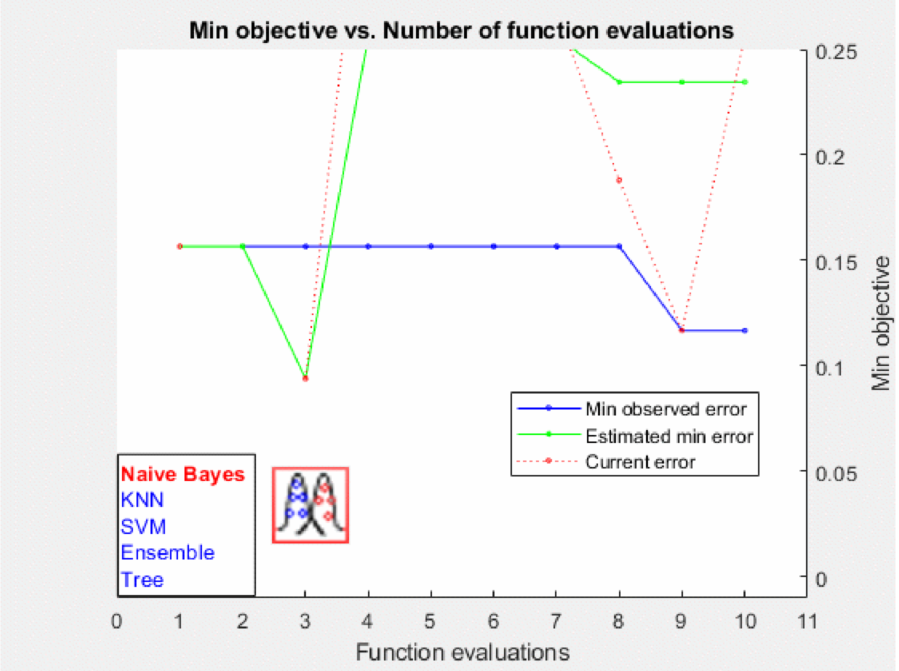
Model Selection

`fitcauto/fitrauto`

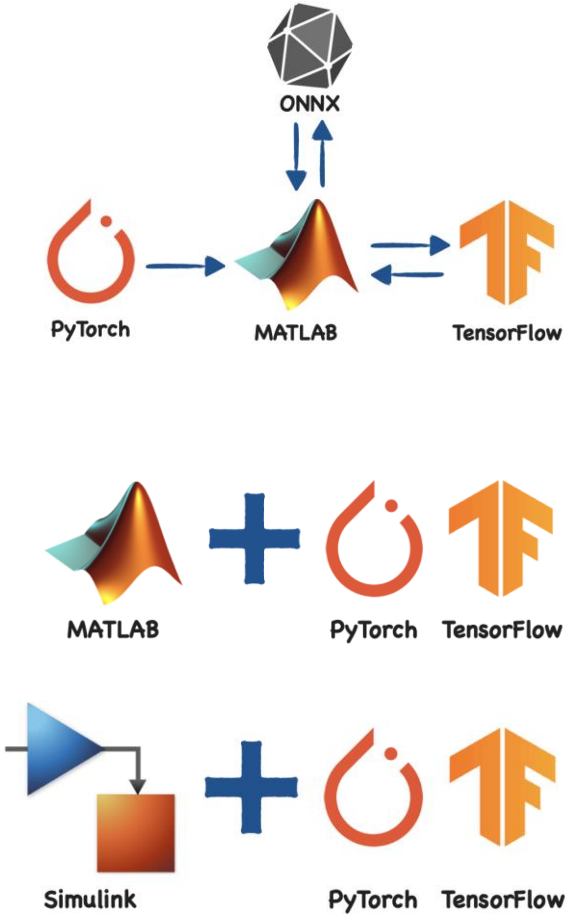
Hyper-parameter Optimization

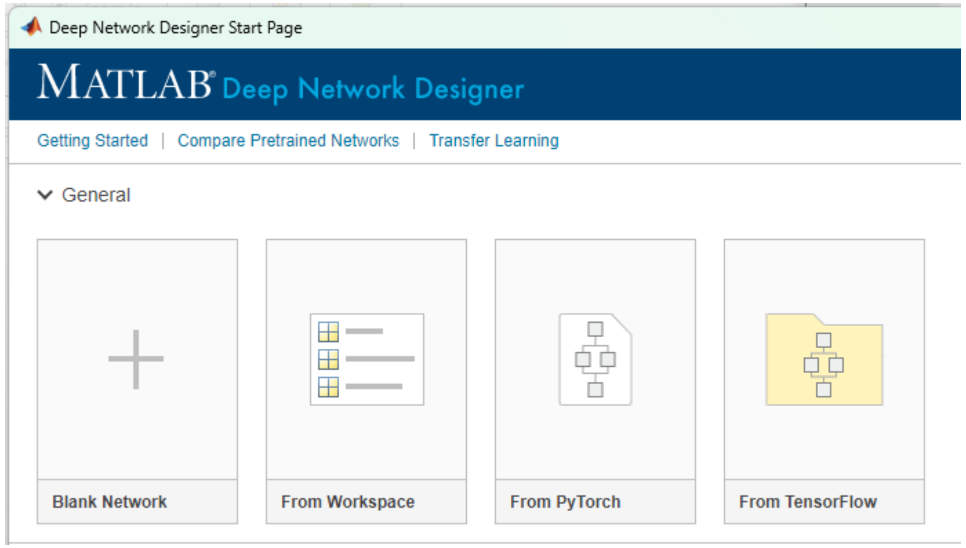


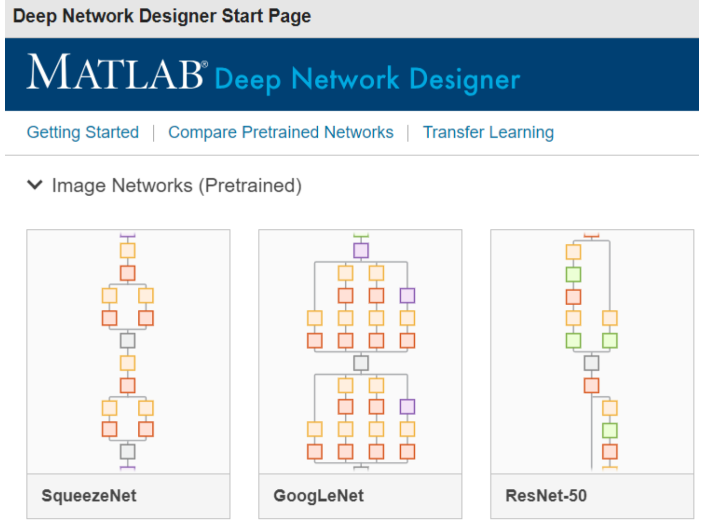
Decision Tree?
SVM?
KNN?
Ensemble?
…?



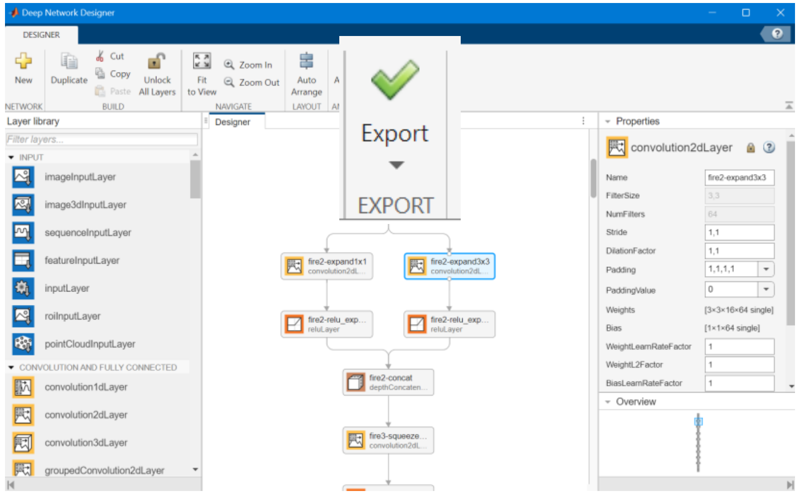
Analyzing the Deep Neural Networks



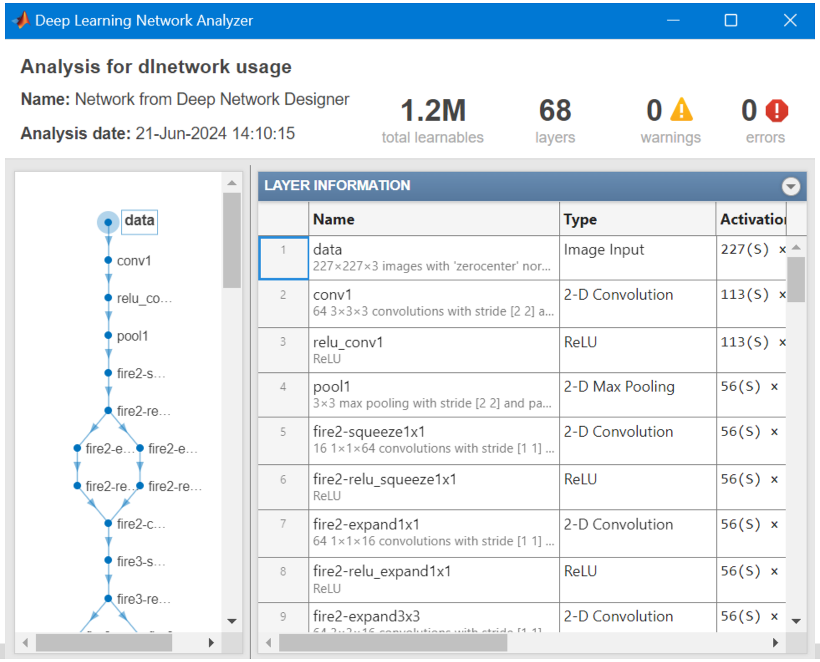




Interactively design networks



Analyze the network



master | 1 Branch | 5 Tags | Go to file | Code

jianghaw Merge pull request #17 from sureshmaddina/master 54ee7d2 · 2 months ago 52 Commits

Images	Adding a section called "Path planning with Motion Planni...	2 months ago
LICENSE	Initial Commit	3 years ago
MATLABDeepLearningModelHub.mlx	updated MATLABDeepLearningModelHub.mlx	last year
README.md	Adding a section called "Path planning with Motion Planni...	2 months ago
SECURITY.md	Initial Commit	3 years ago
viewDeepLearningModelHubGitHub.m	Script for launching the Deep Learning Model hub in a bro...	2 years ago

README | License | Security

MATLAB Deep Learning Model Hub

Discover pretrained models for deep learning in MATLAB.

Models

About

Discover pretrained models for deep learning in MATLAB

www.mathworks.com/solutions/deep-lea...

awesome deep-learning model-zoo matlab awesome-list pretrained-models matlab-deep-learning

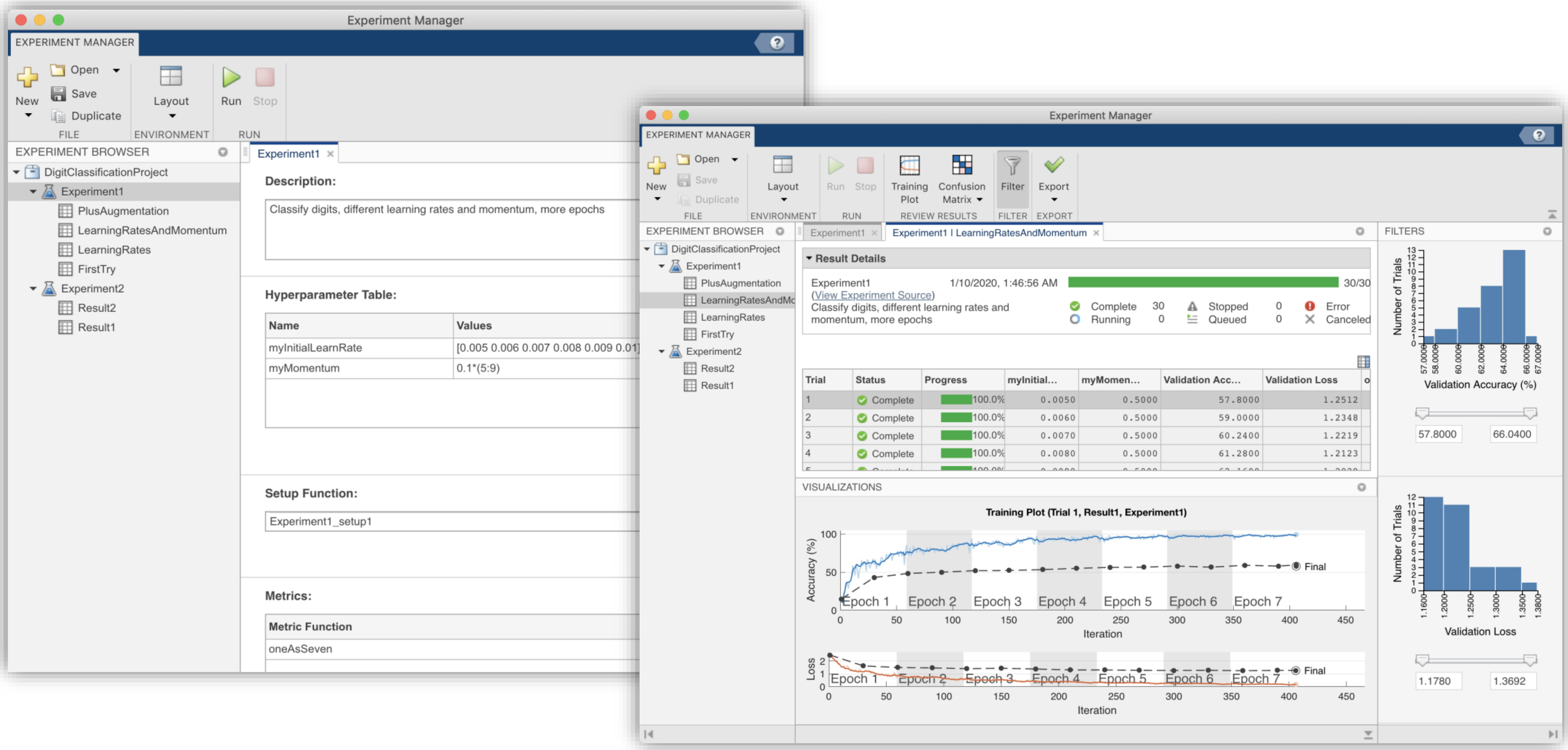
- Readme
- View license
- Security policy
- Activity
- Custom properties
- 418 stars
- 27 watching
- 107 forks

Report repository

Releases 4

R2024a Latest

Compare models and tune hyperparameters with Experiment Manager



MATLAB EXPO

Mercedes-Benz Simulates Hardware Sensors with Deep Neural Networks

Challenge

Simulate automotive hardware sensors with deep neural networks

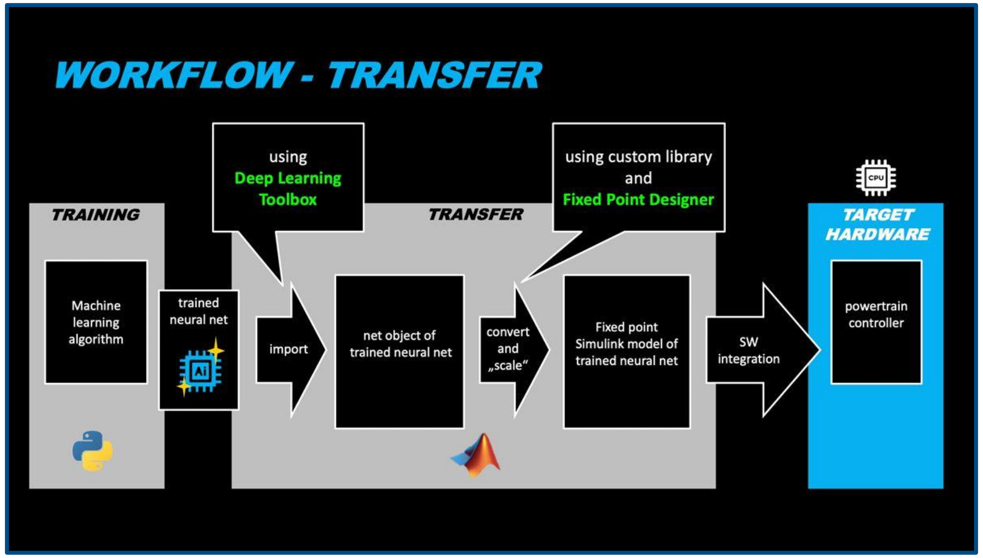
Solution

Use MATLAB, Simulink, Deep Learning Toolbox, and Fixed-Point Designer to convert Qkeras deep learning models into code that can be deployed to an automotive ECU

Results

- CPU, memory, and performance requirements met
- Flexible process established
- Development speed increased 600%

Link to user story



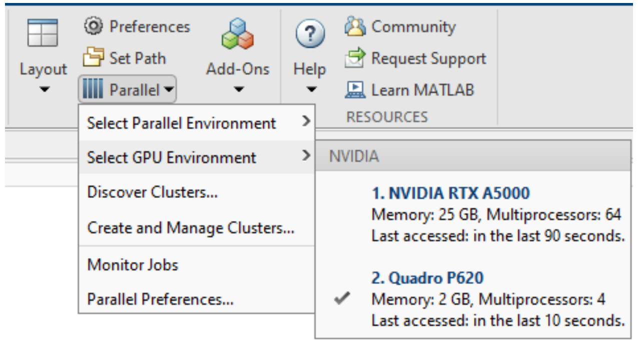
Automated workflow for deploying virtual sensors to powertrain ECU.

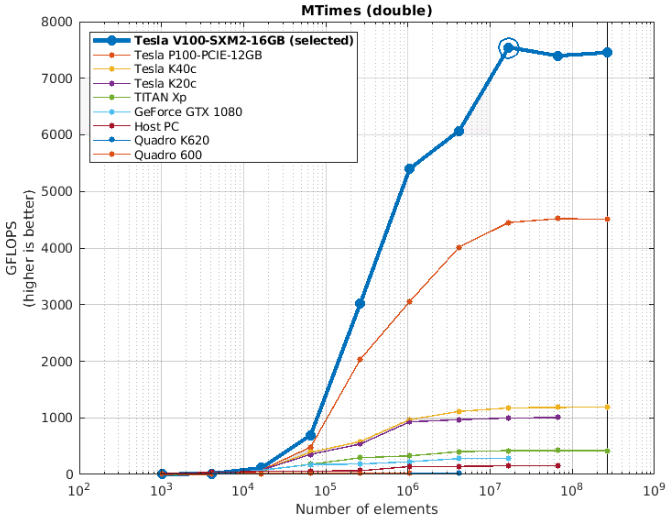
“This was the first time we were simulating sensors with neural networks on one of our powertrain ECUs. Without MATLAB and Simulink, we would have to use a tedious manual coding process that was very slow and error-prone.”

- Katja Deuschl, AI developer at Mercedes-Benz

Hardware acceleration and scaling are critical for training

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

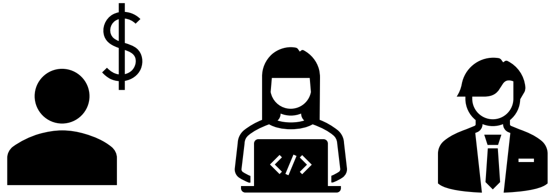




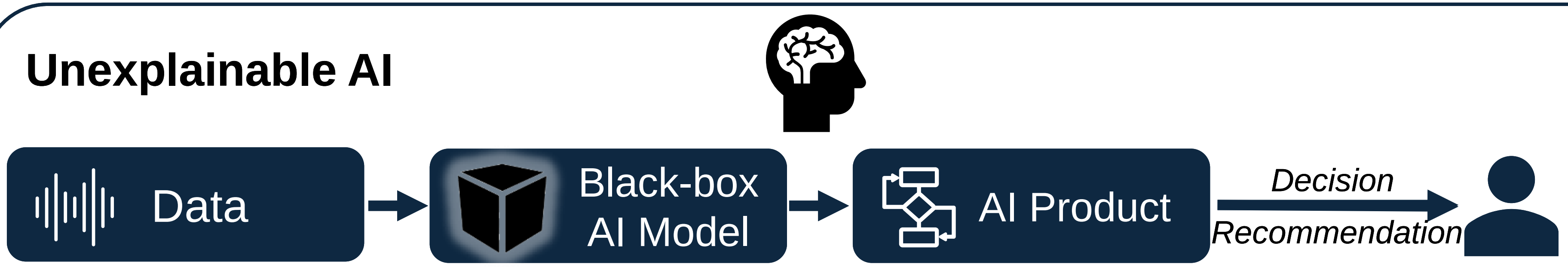
There is a desire to explain, verify and validate AI in production

AI provides the best results
for many tasks

Explainable AI facilitates informed decisions

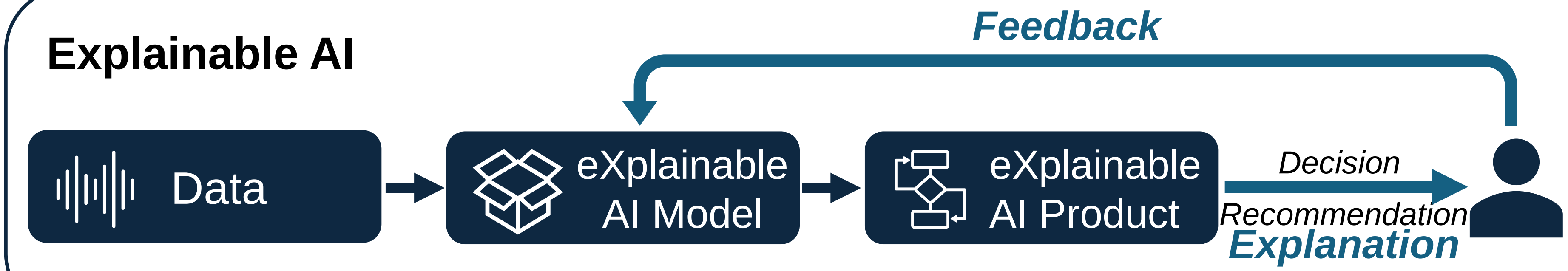


Unexplainable AI



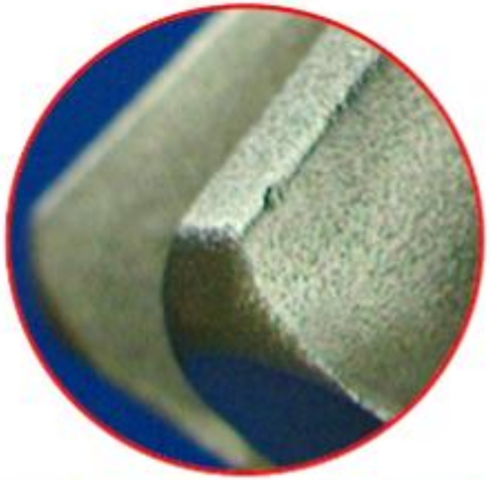
- Why did you do that?
- Why did you not do that?
- When do you succeed or fail?
- When can I trust you?
- How do I correct an error?

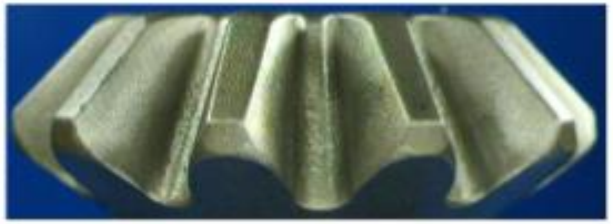
Explainable AI



- I understand why
- I understand why not
- I know why you succeed or fail
- I know when to trust you
- I know why you erred

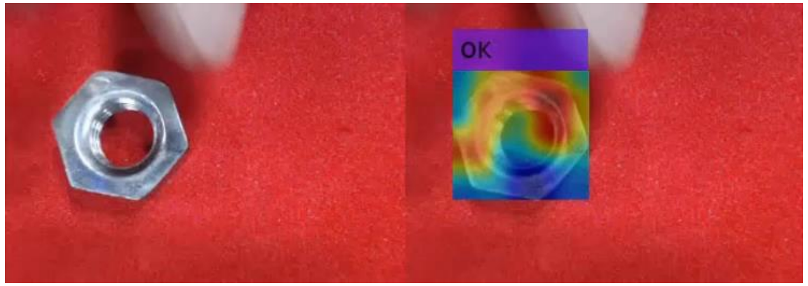
Musashi Seimitsu Industry Uses Deep Learning for Visual Inspection of Automotive Parts





> *Using camera connection, preprocessing, and various pretrained models in MATLAB enabled us to work on the entire workflow. Through discussions with consultants, our team gained many tips for solving problems, growing the skills of our engineers.*

Class Activation Mapping



How XAI was used:

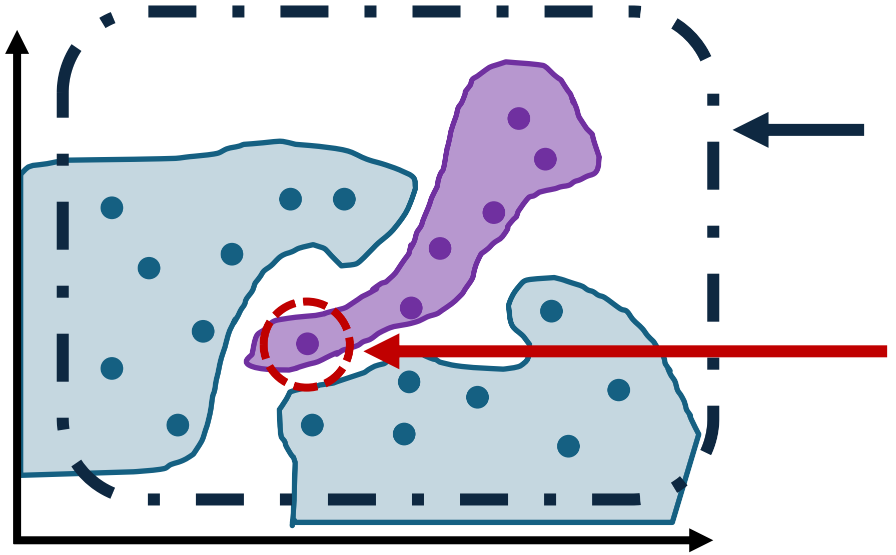
Estimate and visualize the defect area
using Class Activation Mapping

A taxonomy of Explainable AI

Explainable AI

Scope

- Global
- Local



A taxonomy of Explainable AI

Explainable AI

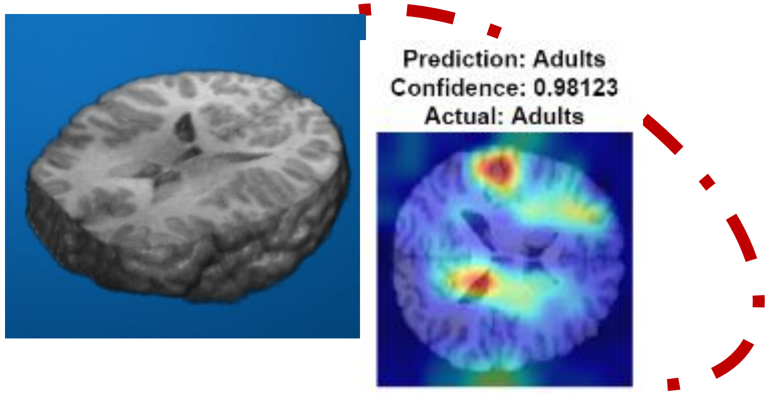
- Scope
  - Global
  - Local
- Stage
  - Pre-model
  - Intrinsic
  - Post-hoc

Post-hoc: Explain model behaviour post-training

Partial Dependence Plot
Individual Conditional Exp

LIME
Shapley values
KNN
Fuzzy Logic

Occlusion Sensitivity, CAM, Grad-CAM, Activations



A taxonomy of Explainable AI

Explainable AI

Scope

Globa

Loca

Model-specific: leverage the inner workings of the model to explain results.

Occlusion Sensitivity, CAM, Grad-CAM, Activations

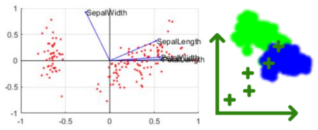
Regression

Tree Models

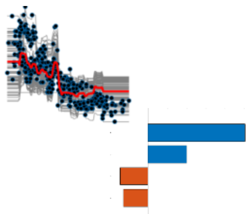
Naive Bayes

Clustering

Model-agnostic: analyze the feature input space and its relationship with the output.



Feature engineering and selection, statistical methods



Partial dependence, LIME, Shapley values, Fuzzy logic, KNN

Model

Model-specific

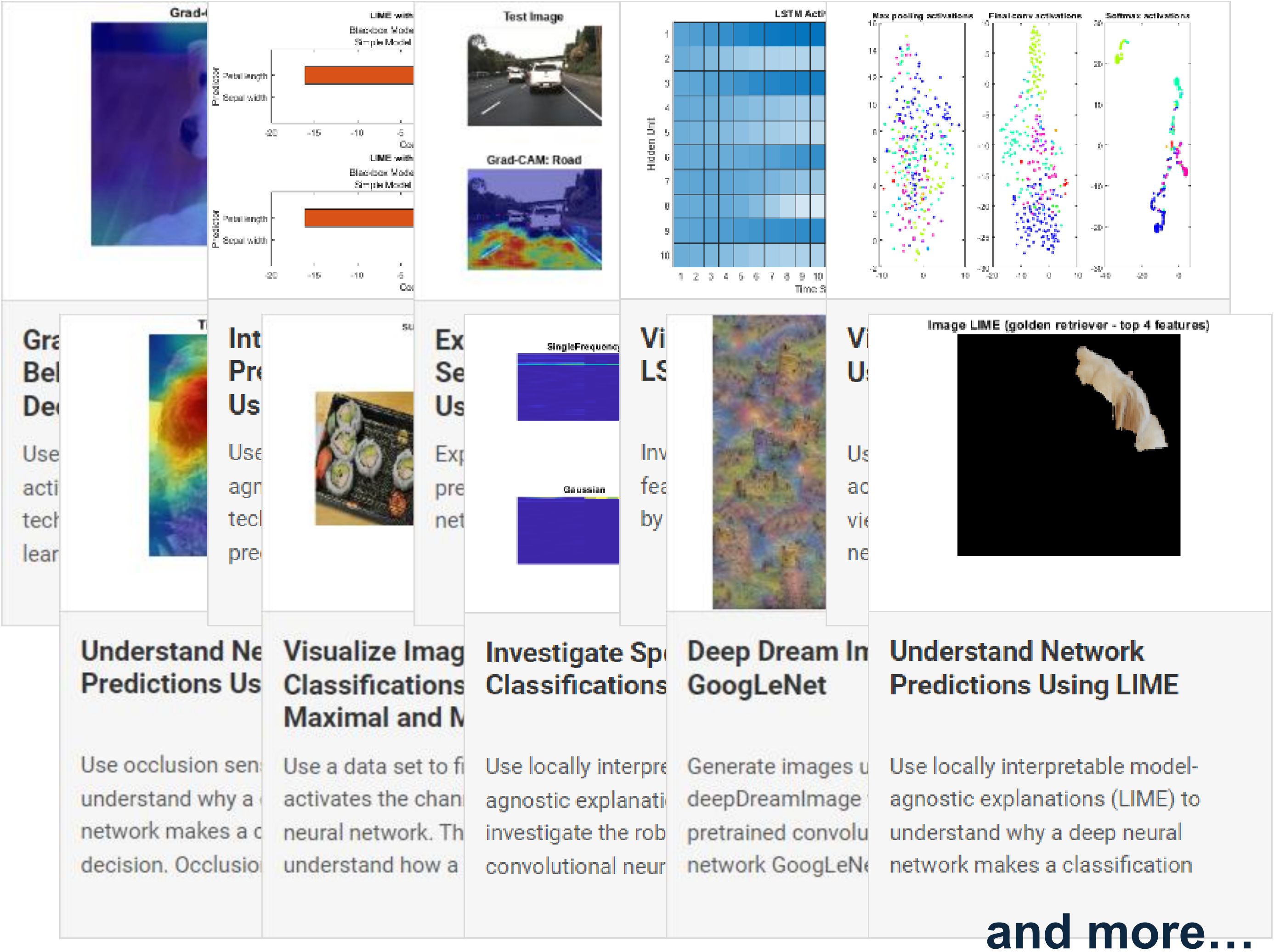
Model-agnostic

Different Explainability methods

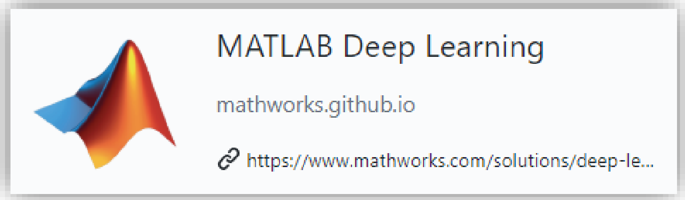
Global methods
Local methods

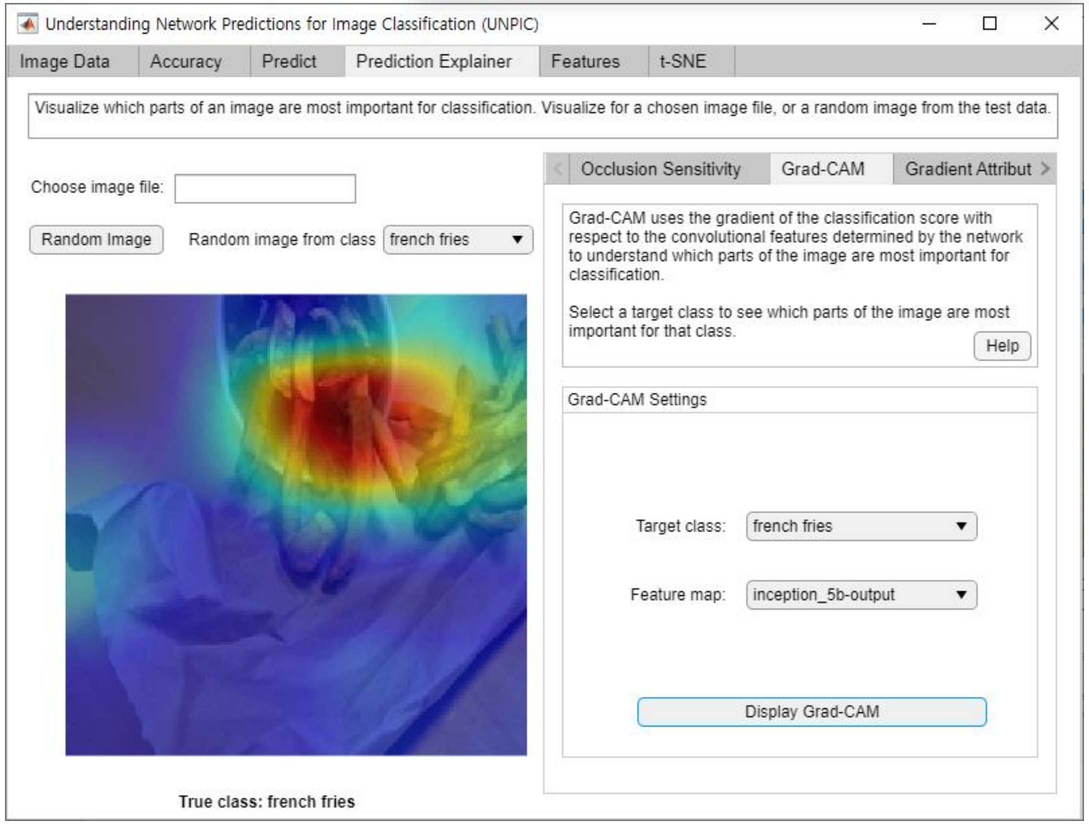
	Pre-model	Intrinsic	Post-hoc
Model-specific		Regression Tree Models Naive Bayes Clustering	Predictor Importance (trees & ensemble) Occlusion Sensitivity CAM, Grad-CAM, Activations
Model-agnostic	PCA tSNE MRMR Correlation		Partial Dependence Plot Individual Conditional Exp LIME Shapley values KNN Fuzzy Logic

Start explaining your AI model with Golden References Today

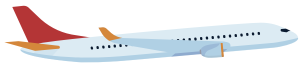


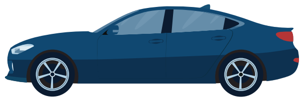
and more…

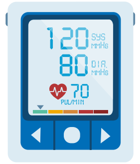


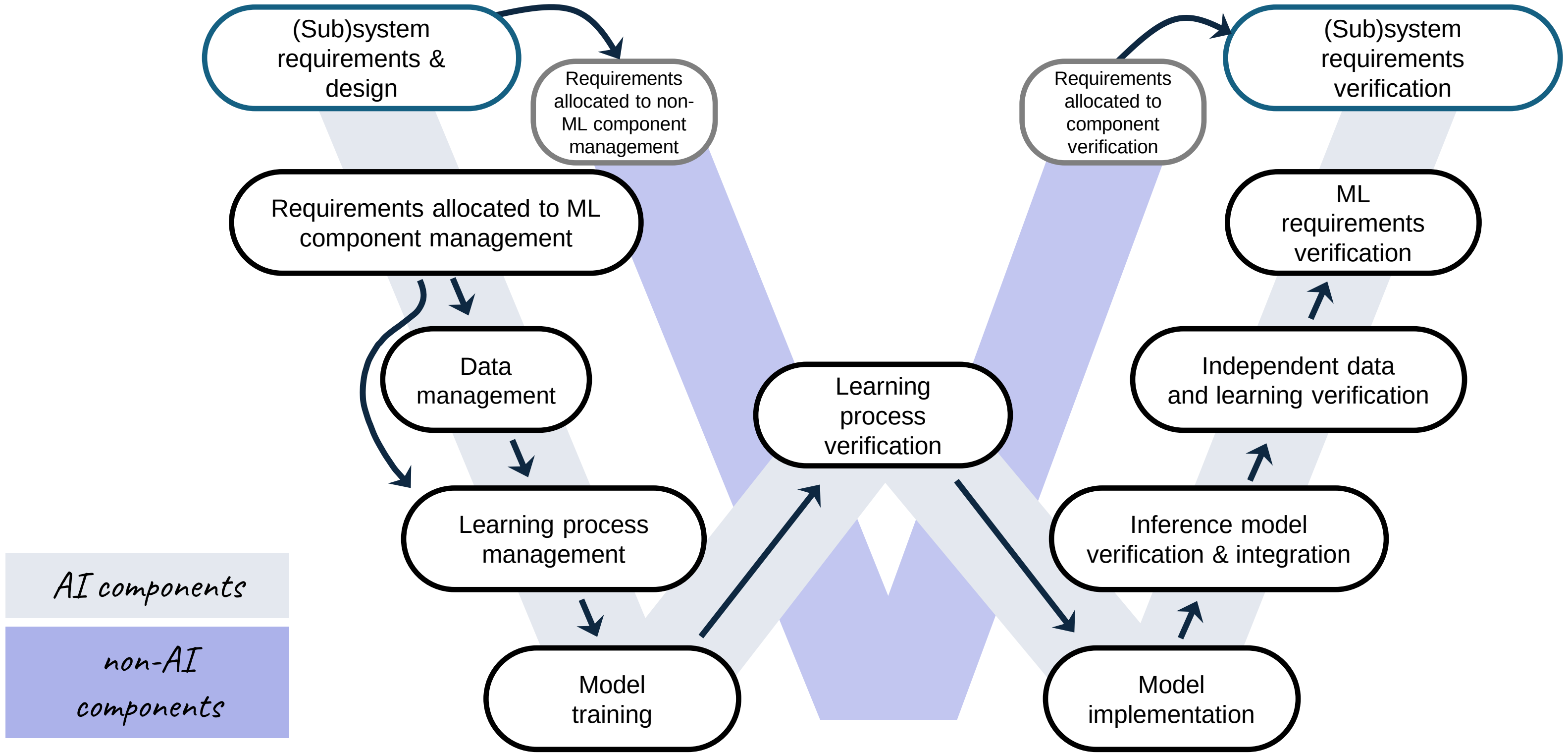


Industries are making progress on verifying AI in systems and its inevitable









Credit: EASA

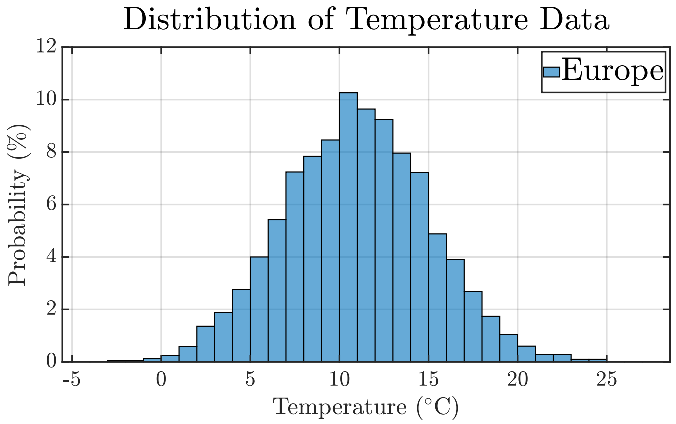
MathWorks has capabilities addressing each area of the W-diagram

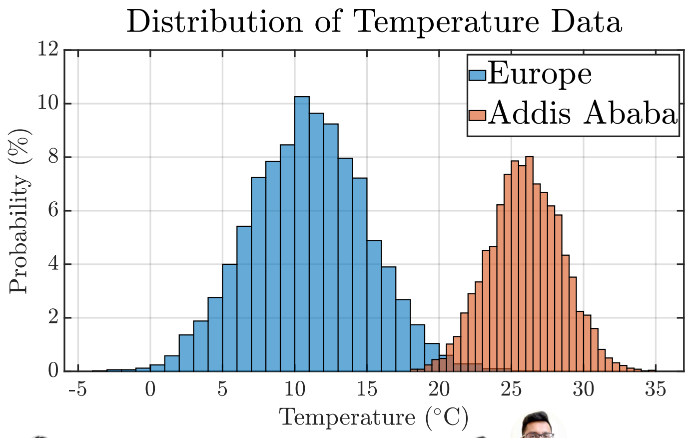
MathWorks is actively engaging with research groups and certification bodies

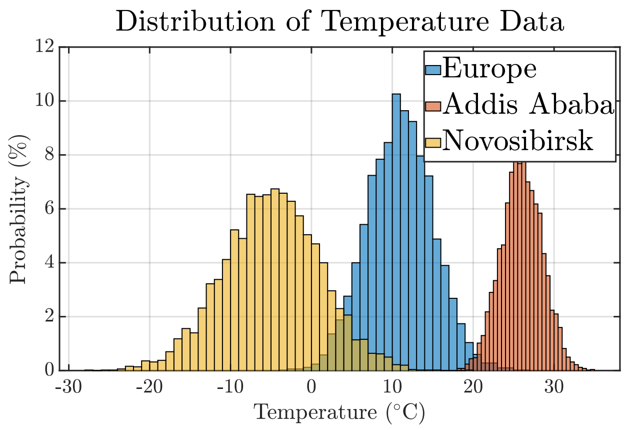
EUROCAE

SAE INTERNATIONAL

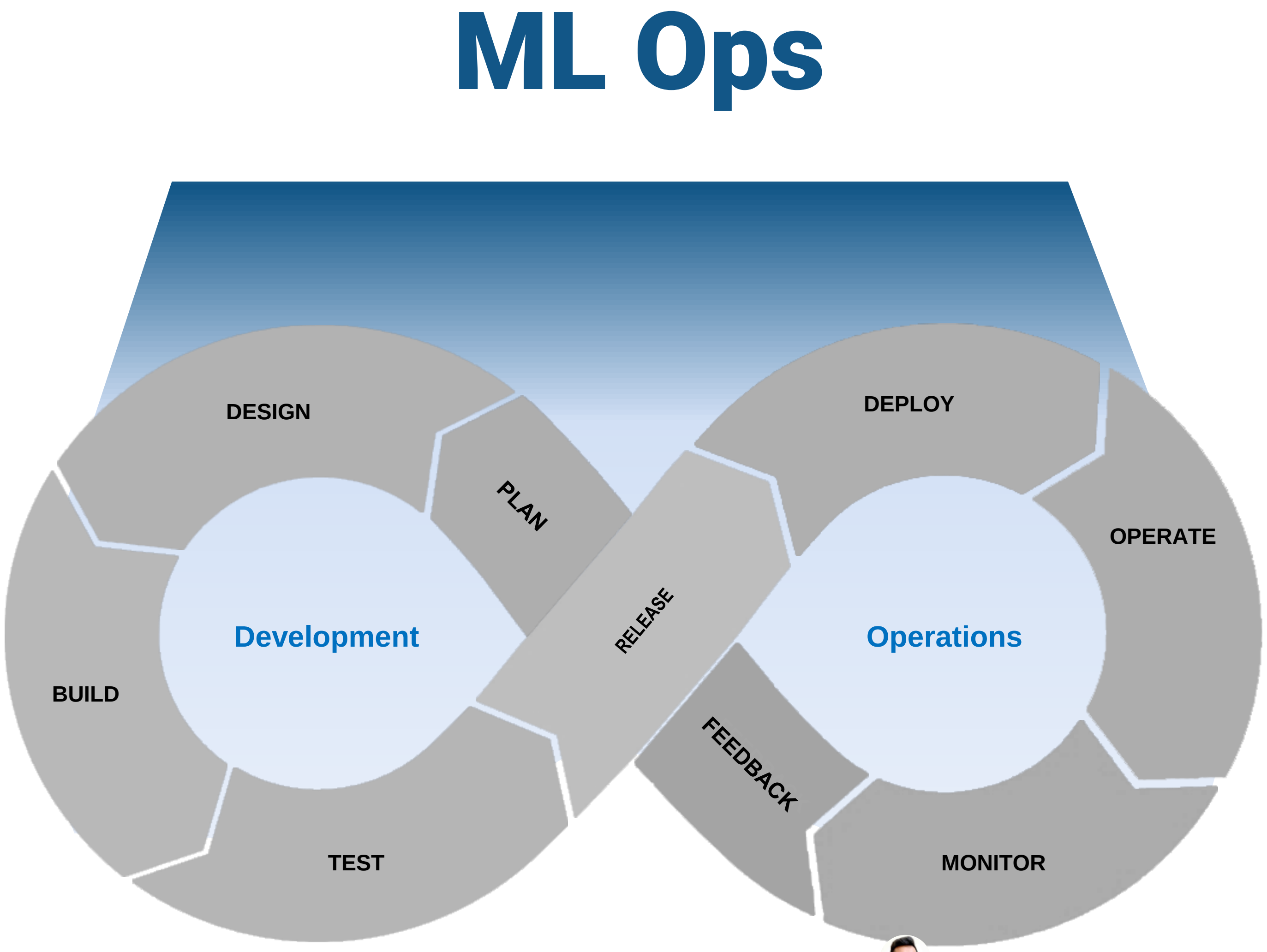
Statistical changes in the input data may decrease a model's predictive or classification accuracy.







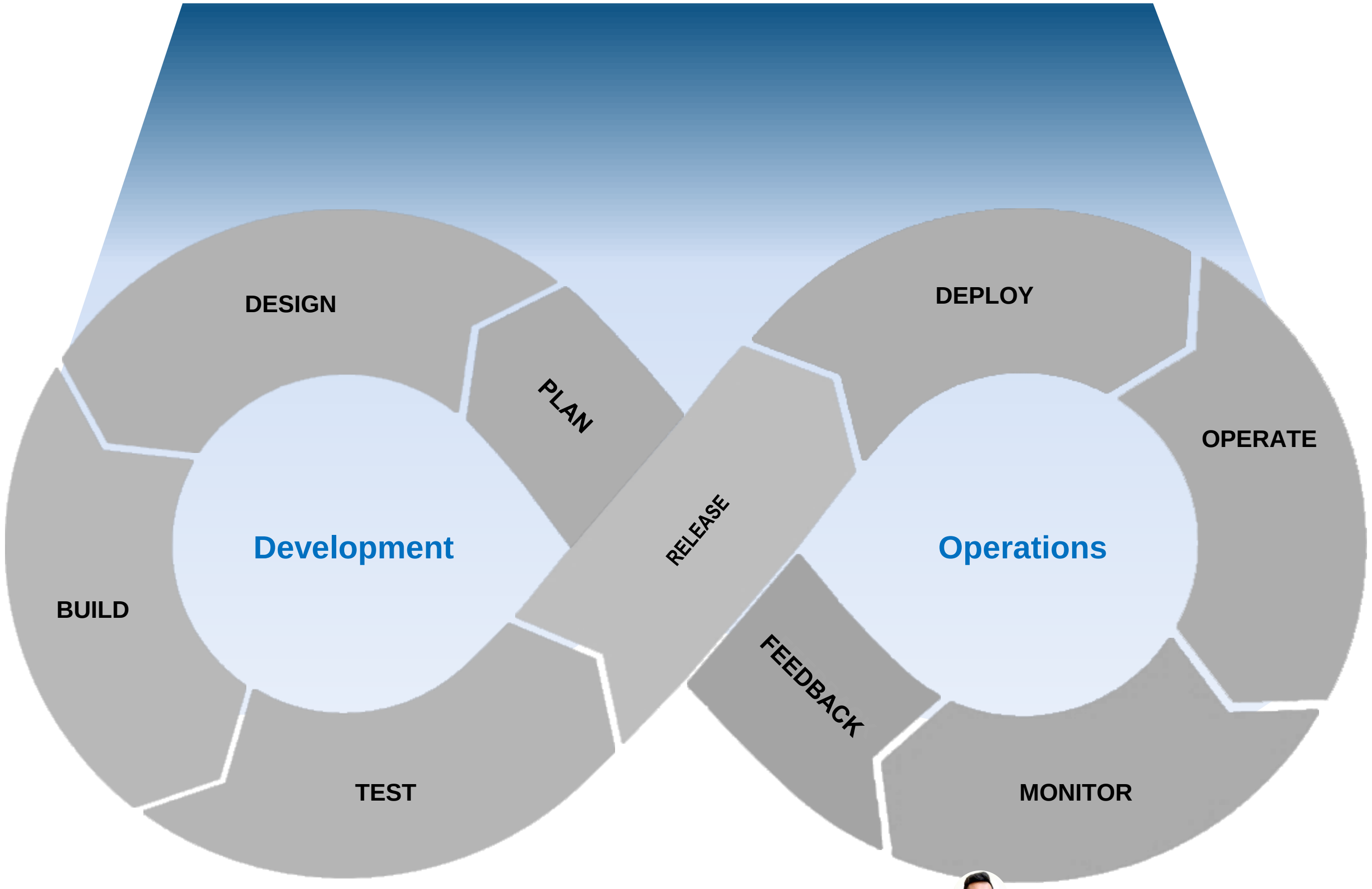
Prompt calibration and model evaluation produces better models and more accurate decisions.



MATLAB EXPO

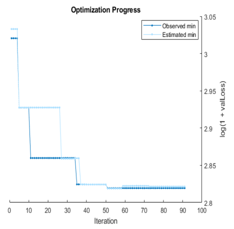
Understanding the lifecycle of a machine learning solution lets you know if you've automated all of it.

ML Ops

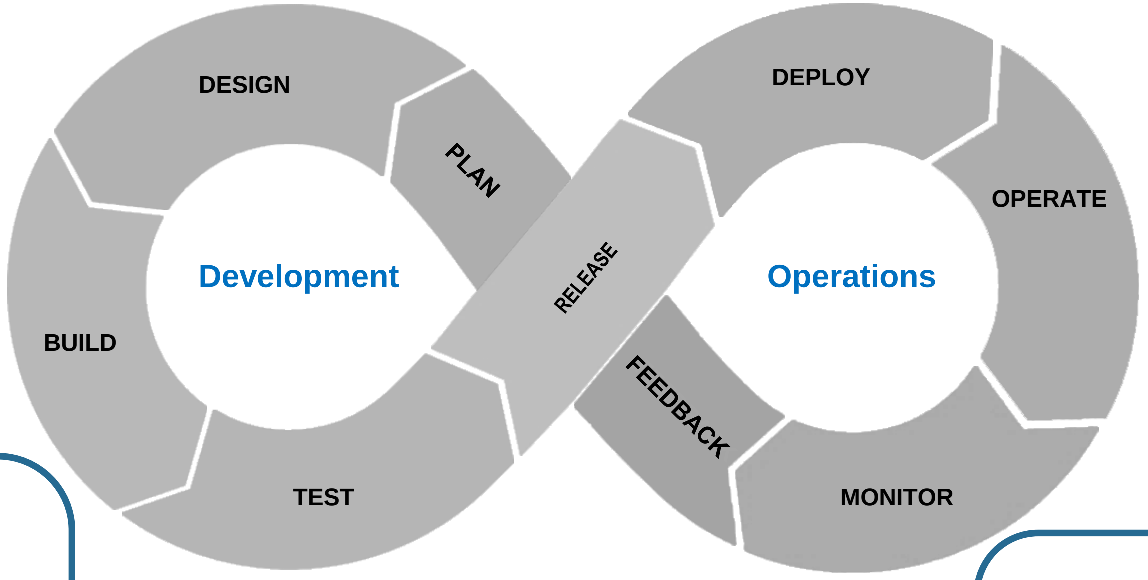


Lots of Moving components and need a strong template to get started

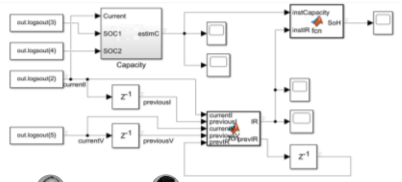
"AutoML"



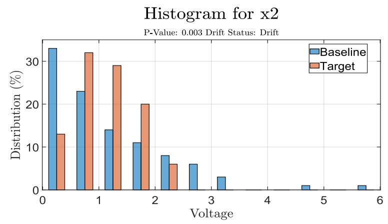
$\hat{f}(x)$



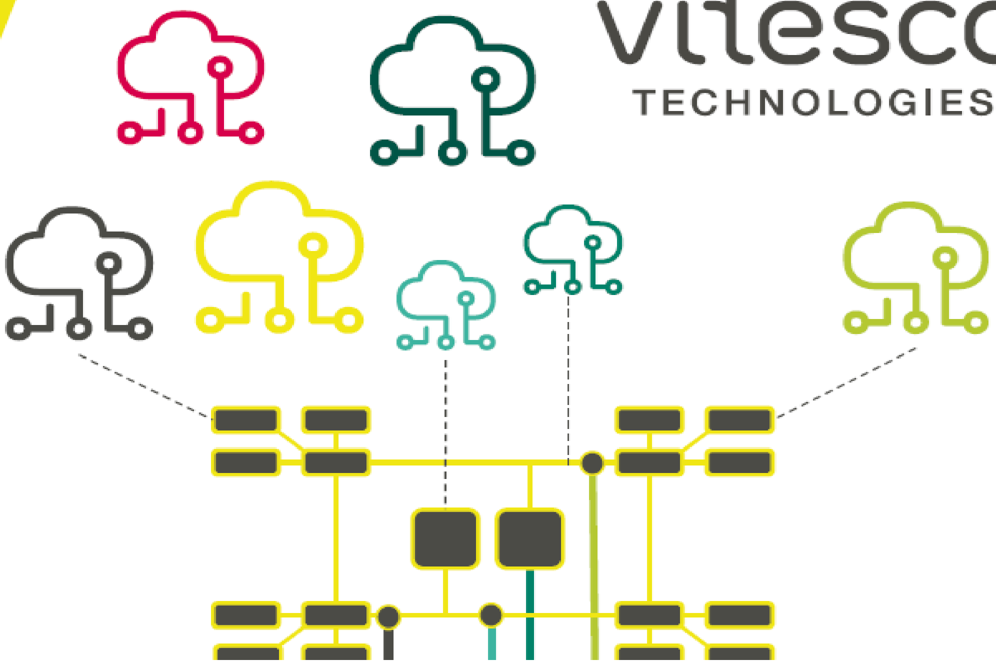
Model-based labeling



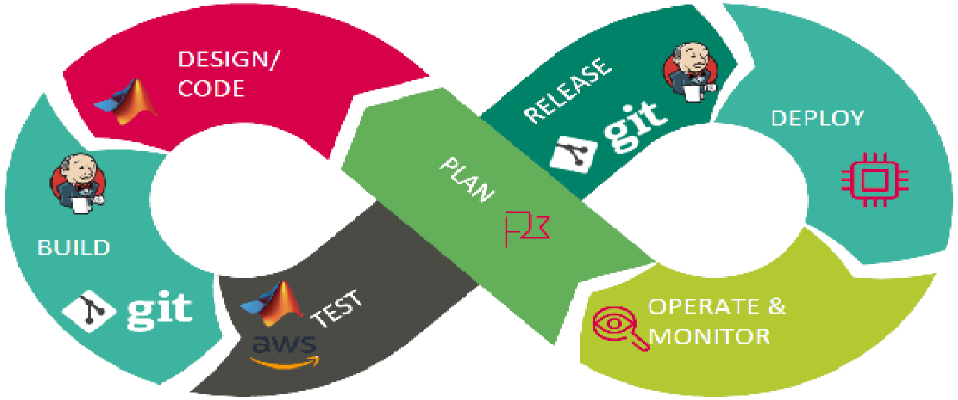
Drift Detection



VITESCO TECHNOLOGIES

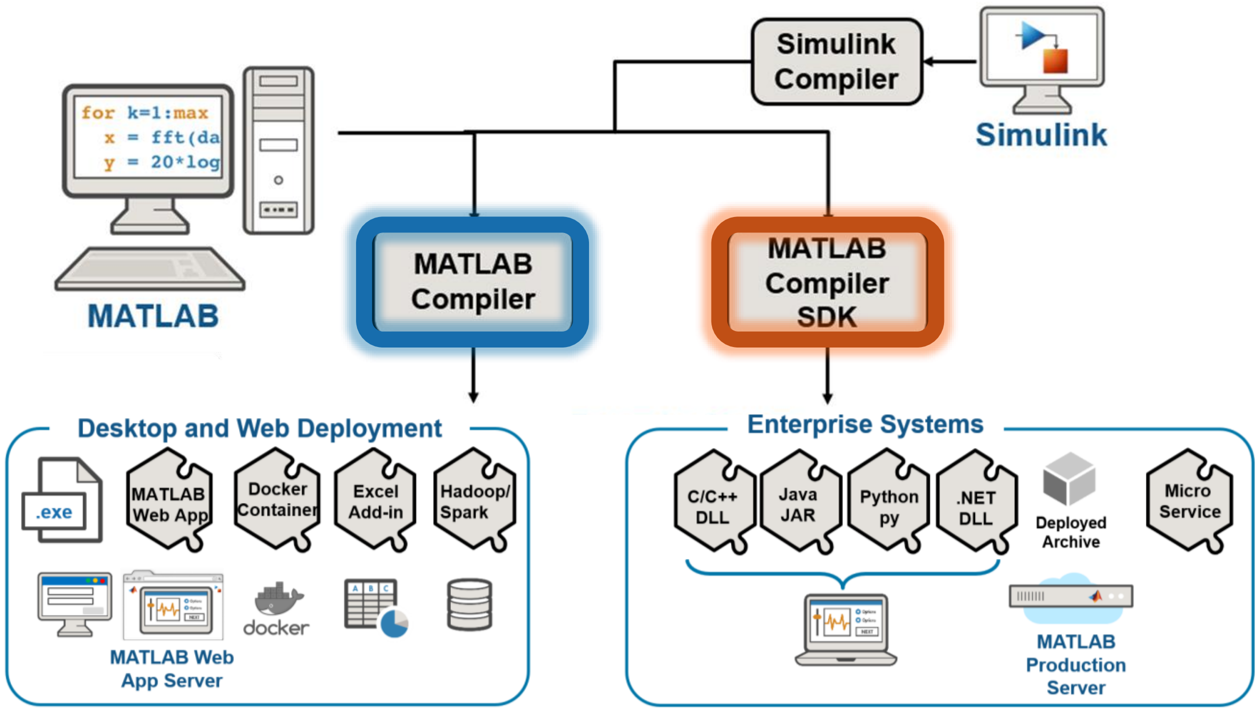


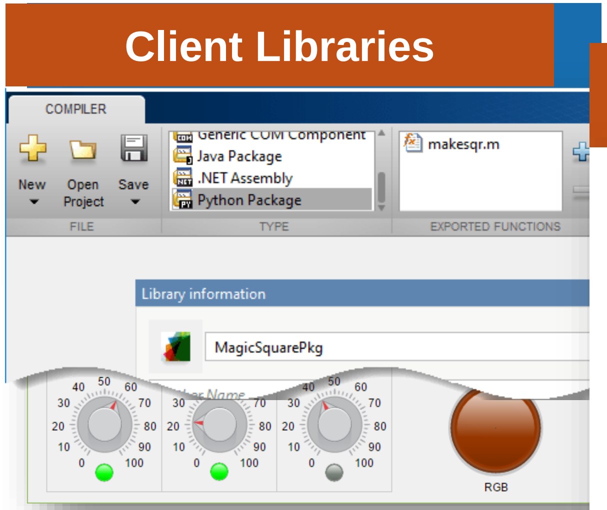
MACHINE LEARNING AND CLOUD FOR EV SYSTEM DEVELOPMENT

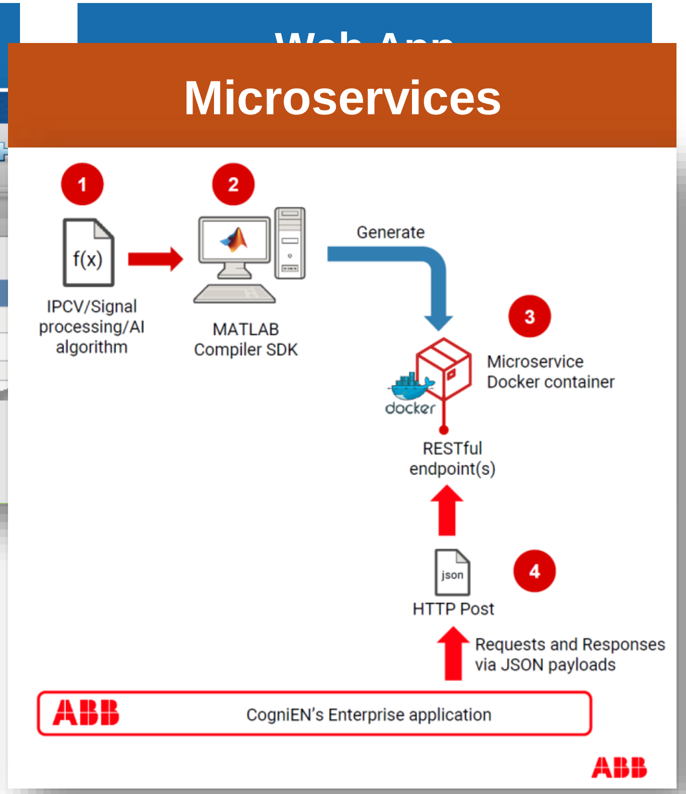


V.Venkobarao,A.Konstantin,M.Wutz, M.Khan,P.Patil,S.Pittan /
Vitesco Technologies /External

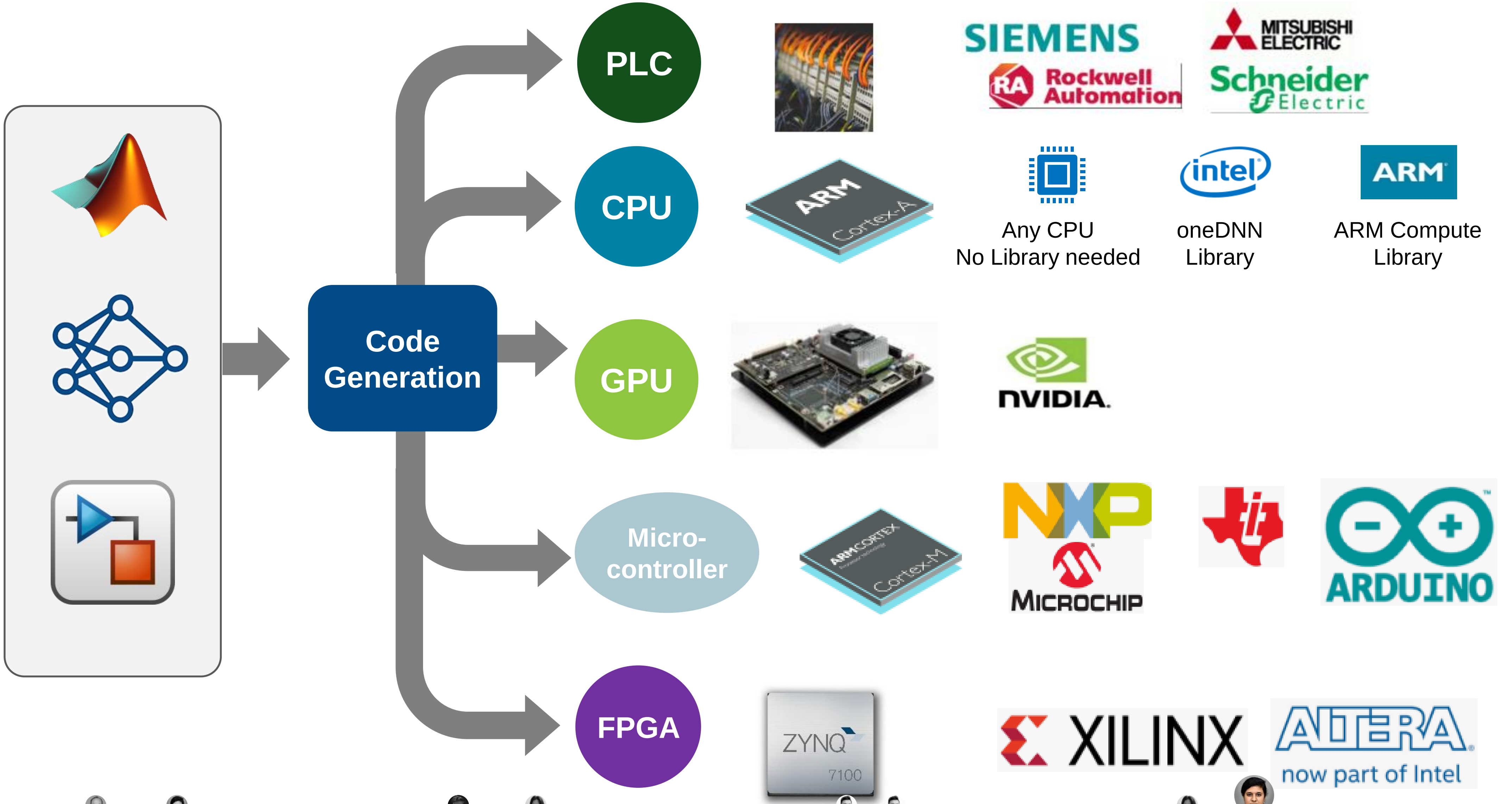
AI Deployment on Enterprise Systems







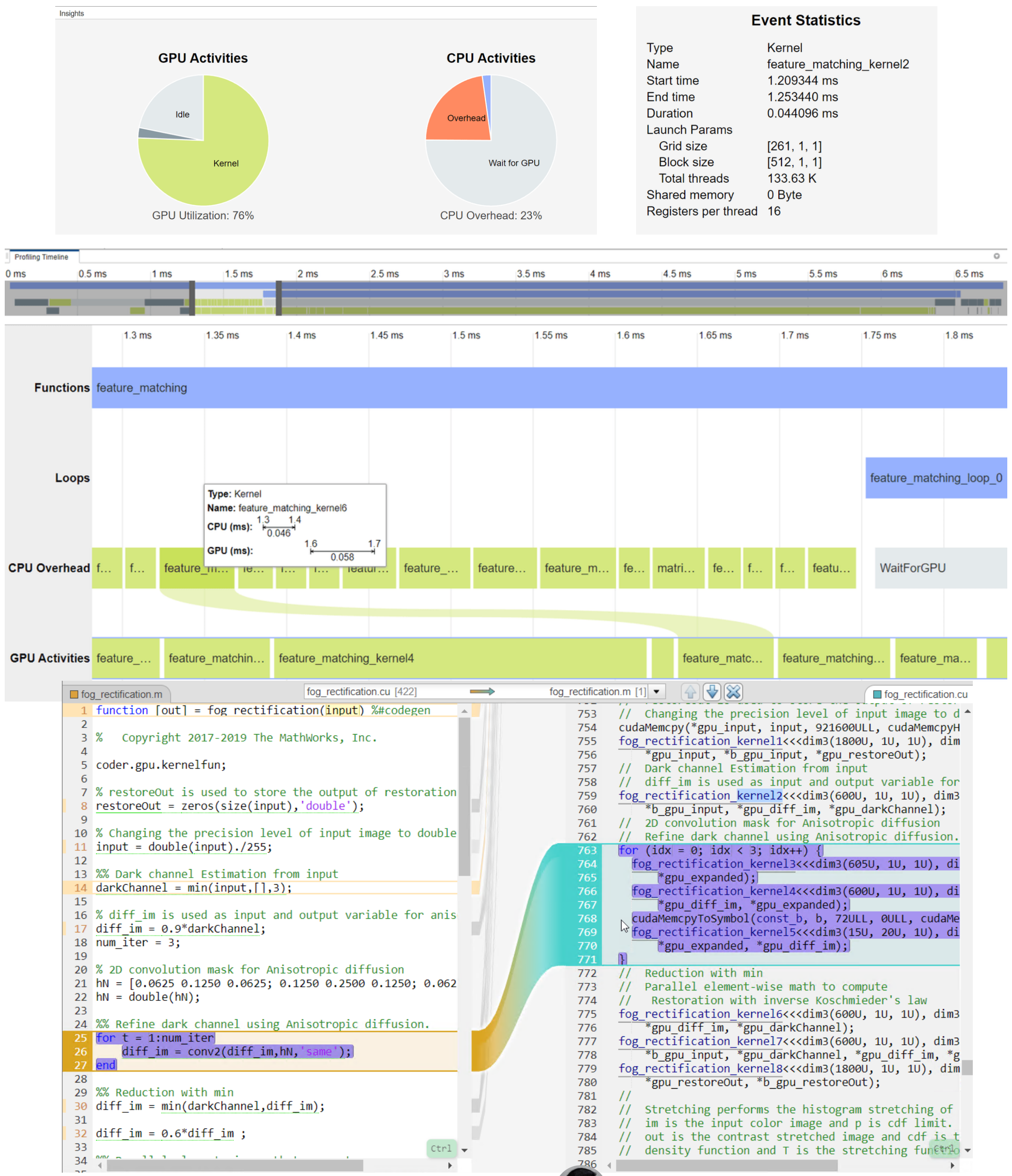
One Codebase – Many Embedded Deployment targets



How to Optimize Further?

Profiling and Bidirectional Traceability Tools

- Use the GPU Coder Performance Analyzer to profile the generated CUDA code
  - Identify bottlenecks & opportunities to optimize performance
- Use bidirectional traceability to map to/from CUDA code back to MATLAB code
  - Helps you to understand how CUDA kernels are created from your MATLAB algorithms



Analyze and Find Issues in Deep Learning Network for Code Generation

- Use analyzeNetworkForCodegen function

```
targetLibraries = {'none','arm-compute','arm-compute-mali',...
    'mkldnn','cmsis-nn','cudnn', 'tensorrt'};
S = analyzeNetworkForCodegen(dlnet,TargetLibrary = targetLibraries);
```

- Detects issues
  - including unsupported layers for code generation, network issues, built-in layer-specific issues, and issues with custom layers

	Supported	NetworkDiagnostics	
none	"Yes"	""	
arm-compute	"Yes"	""	
arm-compute-mali	"No"	"Found 1 issue(s). View network diagnostics."	"Found 2 unsupported layer type(s). View
mkldnn	"Yes"	""	
cmsis-nn	"No"	"Found 1 issue(s). View network diagnostics."	"Found 2 unsupported layer type(s). View
cudnn	"Yes"	""	
tensorrt	"Yes"	""	

```
S(5).LayerDiagnostics
```

ans=4×3 *table*

LayerName	LayerType	Diagnostics
"add"	"AdditionLayer"	"Unsupported layer type."
"relu_body1"	"ReLULayer"	"Unsupported layer type."
"relu_body3"	"ReLULayer"	"Unsupported layer type."
"output2"	"ReLULayer"	"Unsupported layer type."

- Requires MATLAB Coder Interface for Deep Learning Libraries and GPU Coder Interface for Deep Learning Libraries Support Packages

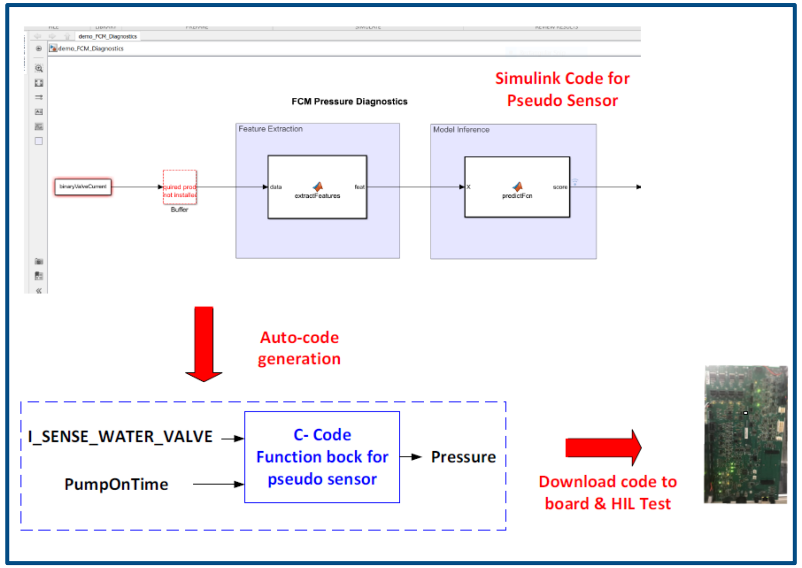
- **Analyze Network for Code Generation**

Coca-Cola Develops Virtual Pressure Sensor with Machine Learning to Improve Beverage Dispenser Diagnostics

Using MATLAB and Simulink, Coca-Cola designed and deployed a machine learning algorithm that serves as a virtual pressure sensor, improving field diagnostics and eliminating the need to retrofit 10,000 Freestyle beverage dispensers with costly sensors.

Key Outcomes/Advantages:

- Transformed a standard flow control module into a diagnostics-capable smart component
- Eliminated the need to retrofit thousands of existing dispensers with costly sensors
- Achieved up to 91% accuracy in pressure predictions



Modeling, deployment, and testing of a virtual ("pseudo") pressure sensor.

With the help of MathWorks, the team was able to reduce the footprint of this code so that it will fit nicely in the ARM-Cortex M microprocessor. It has transformed the flow control module into a smart component.

Link to user story

Deploying AI is difficult

Four specific challenges

- **Integrate AI** model with an Embedded systems
- Fit large AI models on **limited hardware memory**
- **Error-free Code** Generation
- Achieving **Real-Time Performance** post-deployment

AI Compression Techniques

Minimize model size and speed up inference for deploying AI models

Reduce memory and power needs of deployed models

Quantization

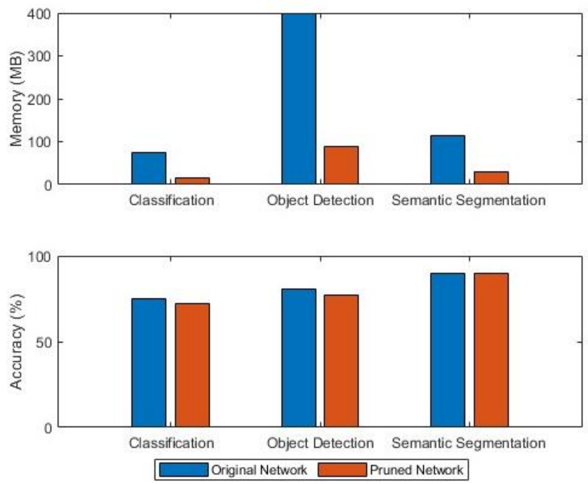
- Convert learnable from floating point to fixed point

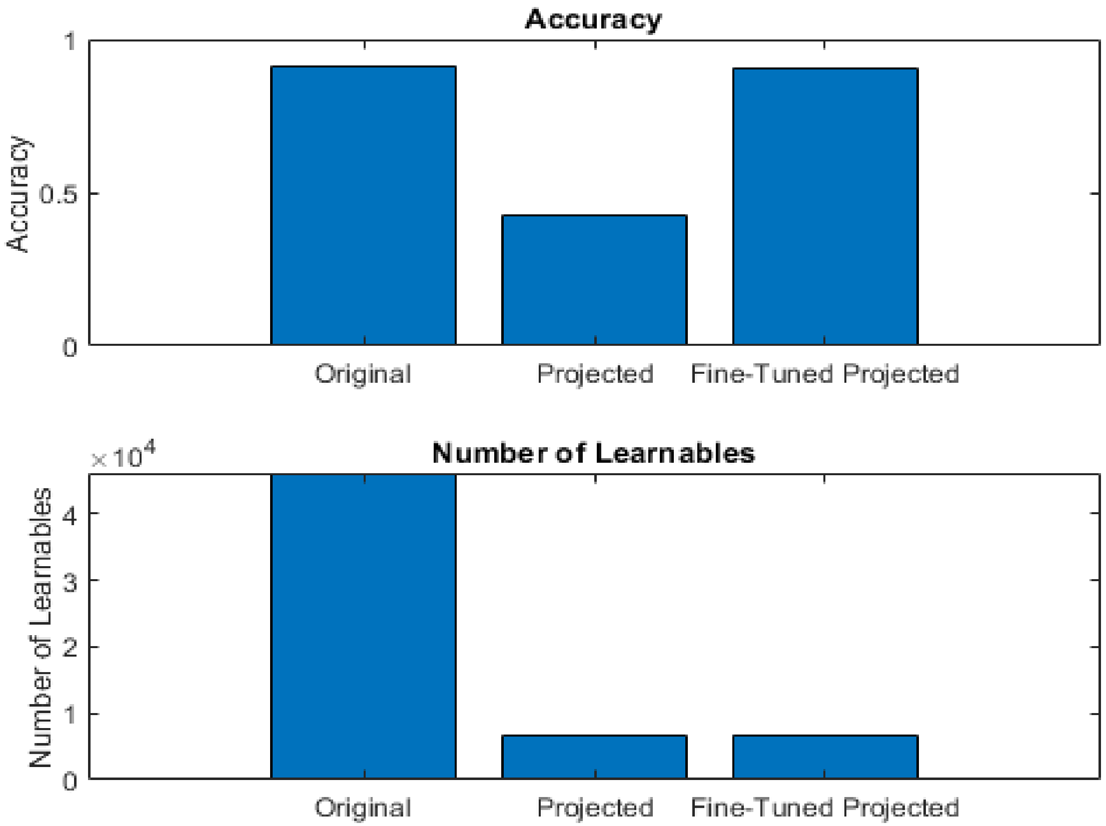
Pruning

- Remove unimportant parts of the network

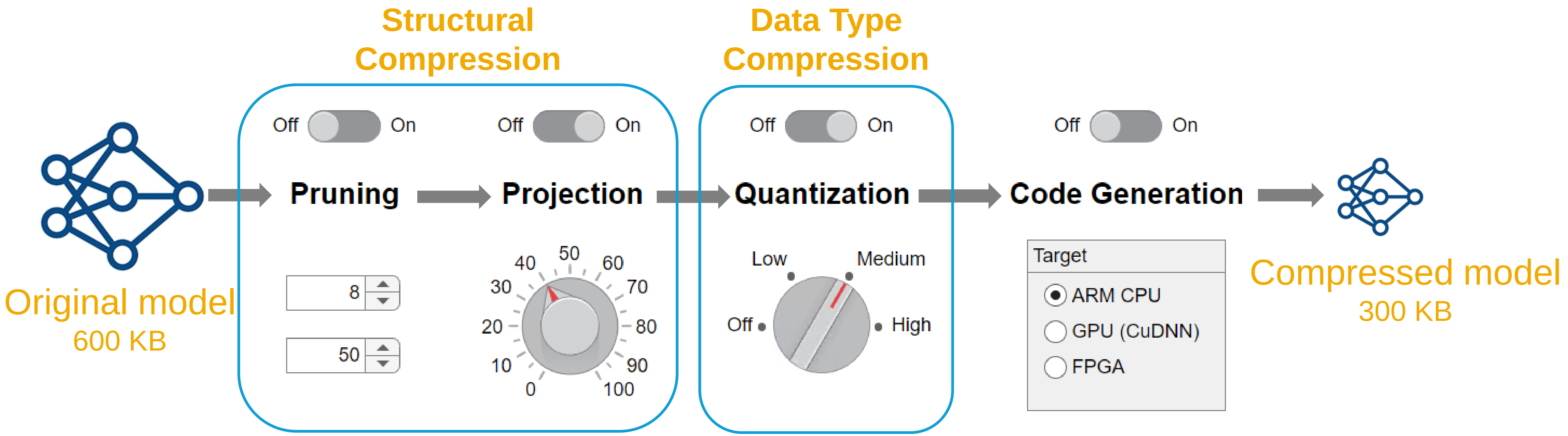
Projection

- Perform principal component analysis to identify redundancies

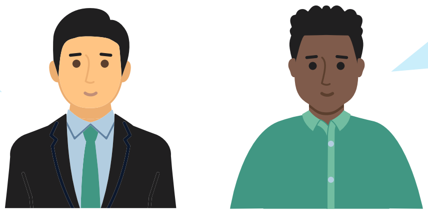




Reduce model footprint and accelerate inference of DL models for deployment to the edge



*"original [network] was 40MB, was told needed 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"*


MathWorks
@MathWorks
Share the EXPO experience
#MATLABEXPO
MATLAB
EXPO

MATLAB EXPO

INDIA

MathWorks®