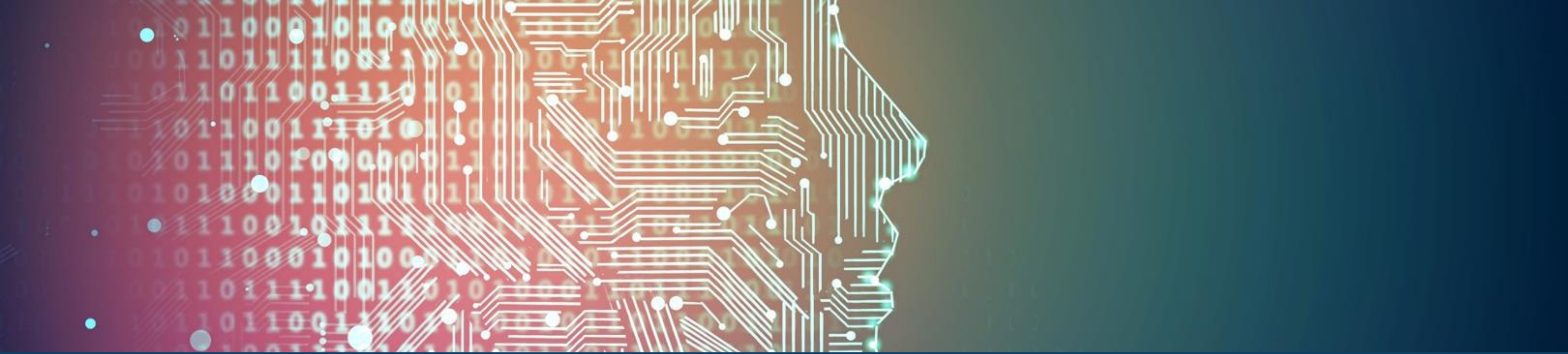


MATLAB과 함께하는 딥러닝 4주 완성 부트캠프

MathWorks®

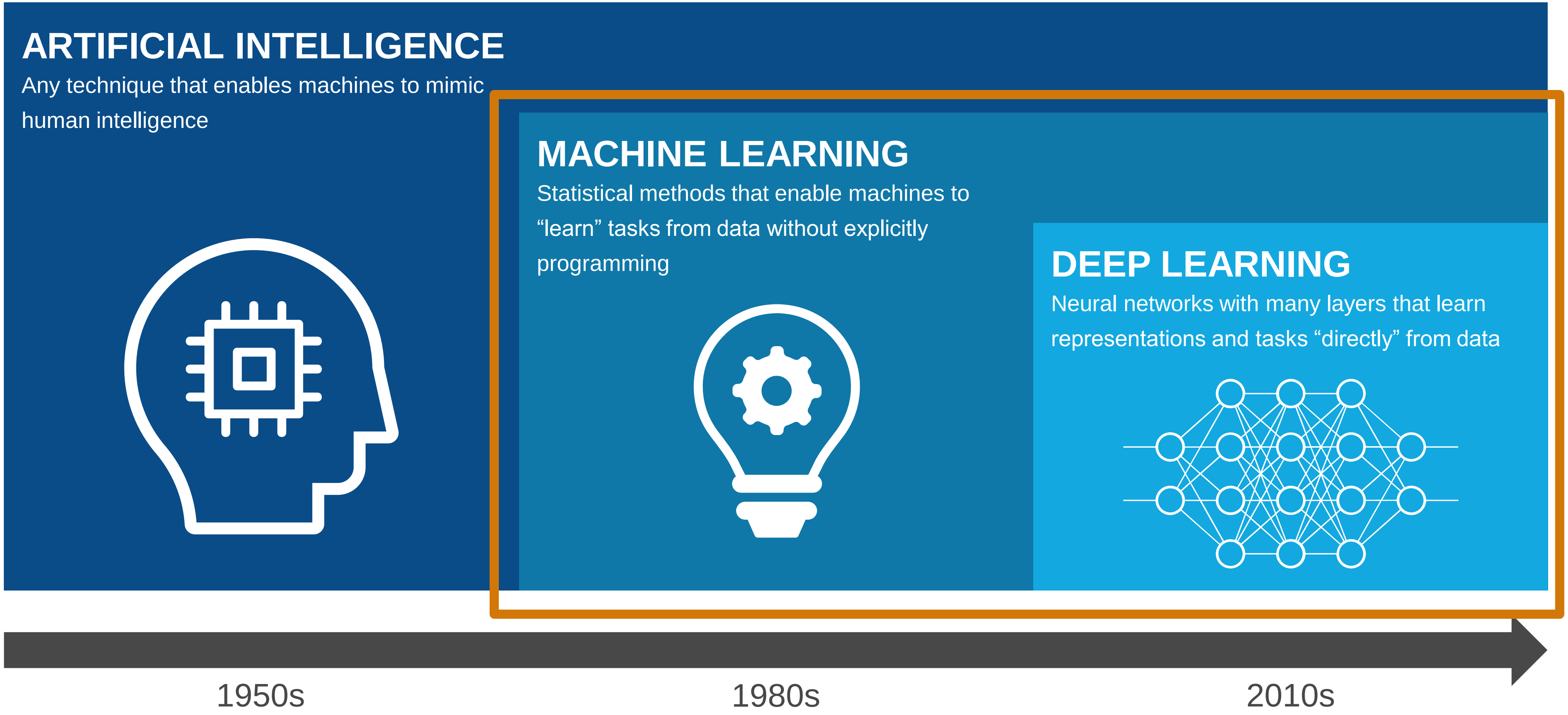
세션2. 신호처리를 위한 머신러닝과 딥러닝

MATLAB과 함께하는 딥러닝 4주 완성 부트캠프

김종남 부장
Application Engineer @ MathWorks
calebkim@mathworks.com

MathWorks®
Accelerating the pace of engineering and science

Understanding AI megatrend



ARTIFICIAL INTELLIGENCE

Any technique that enables machines to mimic human intelligence

MACHINE LEARNING

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

DEEP LEARNING

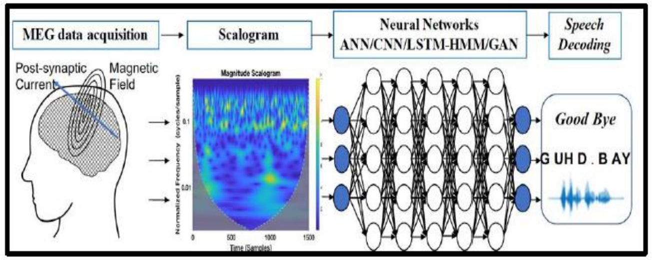
Neural networks with many layers that learn representations and tasks “directly” from data

1950s

1980s

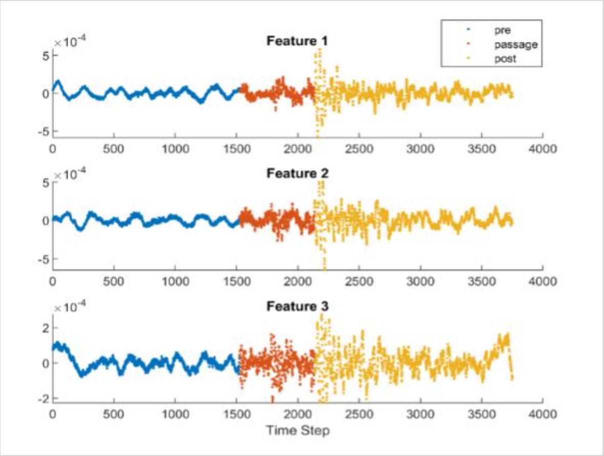
2010s

MATLAB AI used in Industries and Research



Converting brain waves to speech to help ALS patients communicate

UT Austin



Seismic Event Detection

Shell

Restoration of arm and hand control by processing brain signals

Battelle

Agenda

- AI Workflow
- Example 1: ECG Classification
- Example 2: Pavement Crack Identification
- Example 3: Human Activity Classification

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

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

Transforming raw data for useful analysis is a critical step

Data Preparation

Data cleansing and preparation

Human insight

Simulation-generated data

AI Modeling

Model design and tuning

Hardware accelerated training

Interoperability

Deployment

Embedded devices

Enterprise systems

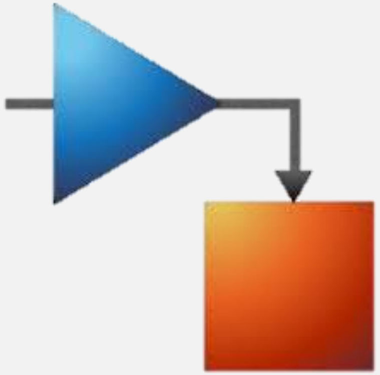
Edge, cloud, desktop

Synthetic Data Generation and Augmentation to deal with less data

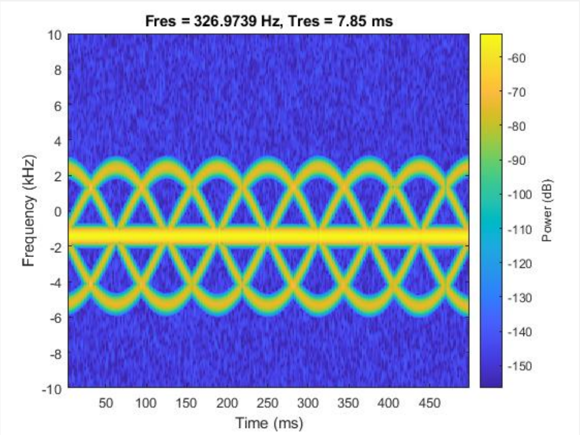
Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

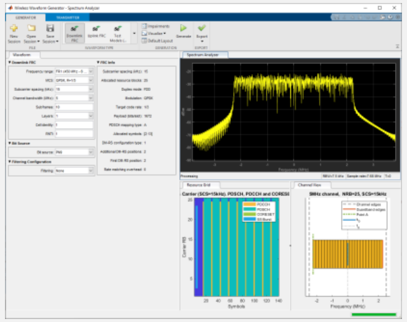
Simulate data using models and deep learning



Generate Radar Returns

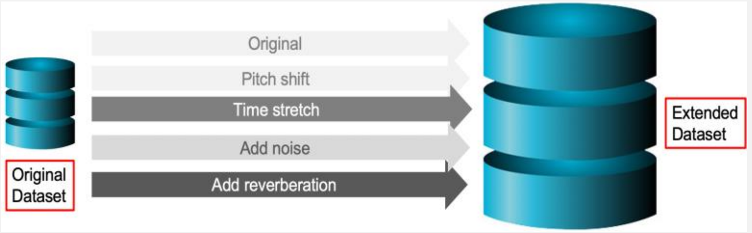


Generate wireless waveforms



Generate Audio Data

`text2speech`

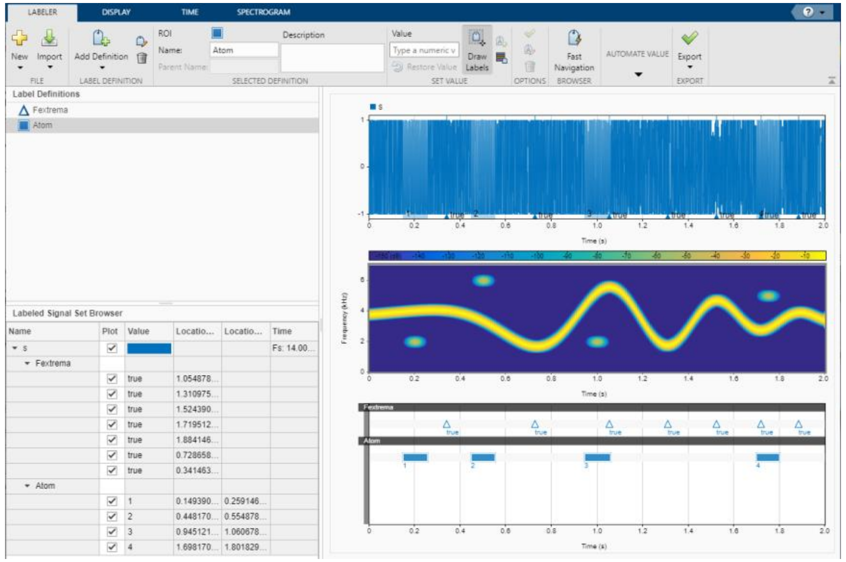


8

Use Signal Processing Apps to speed up Labeling and Preprocessing

Data Preparation

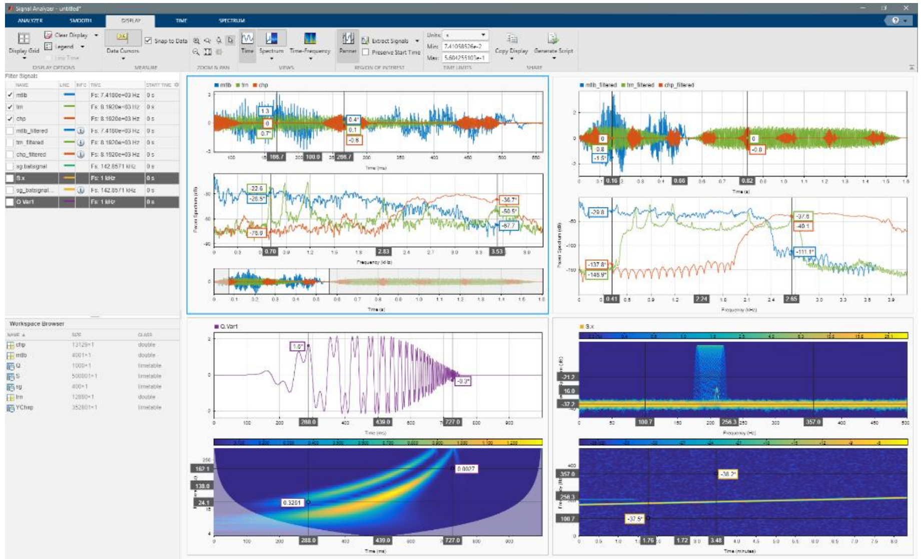
- Data cleansing and preparation
- Human insight
- Simulation-generated data



Labeled Signal Set Browser

Name	Plot	Value	Locatio...	Locatio...	Time
s	☑	🟦			Fs: 14.00...
Fextrema					
	☑	true	1.054878...		
	☑	true	1.310975...		
	☑	true	1.524390...		
	☑	true	1.719512...		
	☑	true	1.884146...		
	☑	true	0.728658...		
	☑	true	0.341463...		
Atom					
	☑	1	0.149390...	0.259146...	
	☑	2	0.448170...	0.554878...	
	☑	3	0.945121...	1.060678...	
	☑	4	1.698170...	1.801829...	

Signal Labeler app



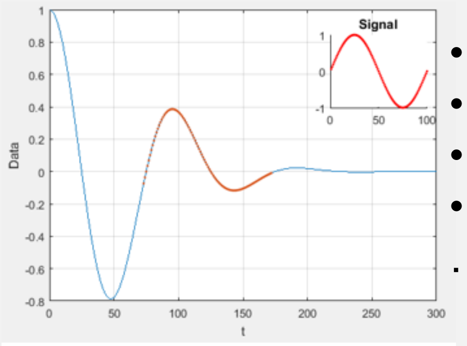
Signal Analyzer App

Easily Extract Features from Signals

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

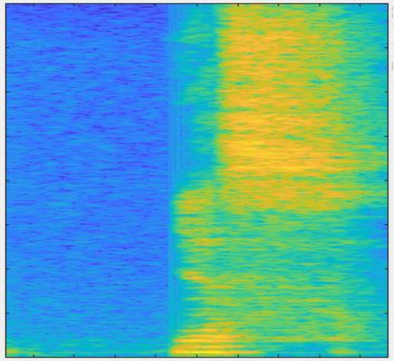
Time-Domain Features



- Signal Patterns
- Changepoints
- Peaks
- Signal Envelope

……

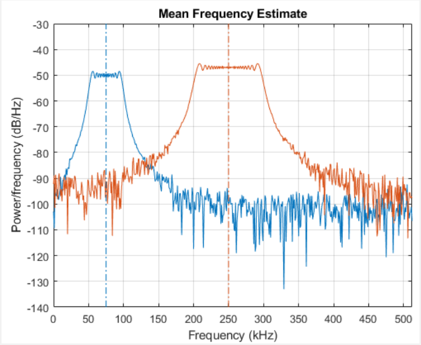
Time-Frequency features



- Spectrogram
- MFCC
- Constant-Q Transform
- Scalogram

……

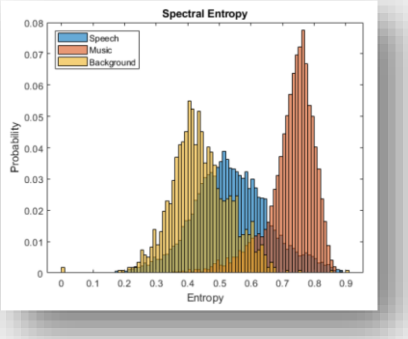
Frequency-Domain Features



- BW measurements
- Spectral Statistics
- Octave Spectrum

……

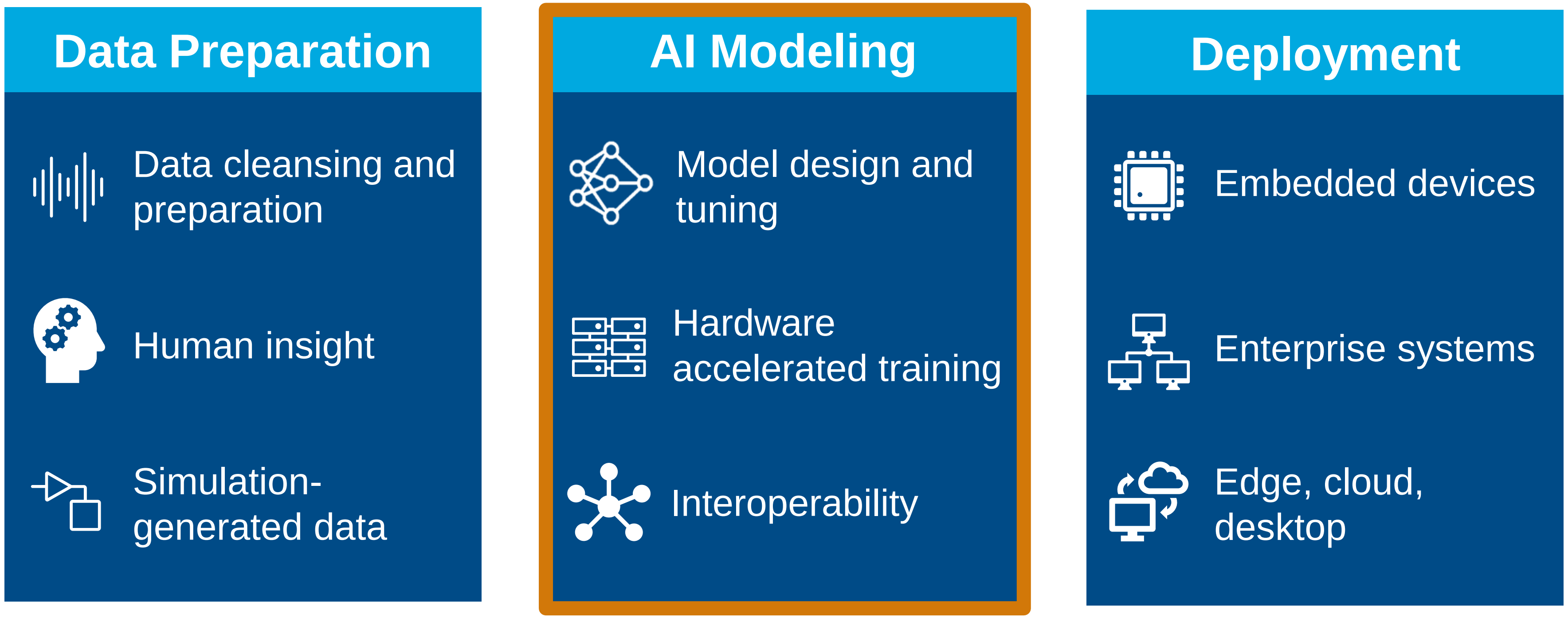
Domain-Specific Features



- Speech and audio
- Navigation and Sensor Fusion
- Radar
- Communication

……

Selecting and training the right model is important



Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

Deployment

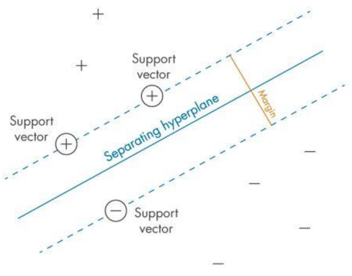
- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

Understand the approaches to creating AI models

AI Modeling

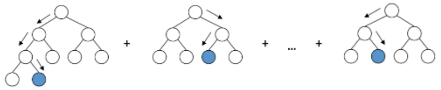
- Model design and tuning
- Hardware accelerated training
- Interoperability

Machine Learning Models



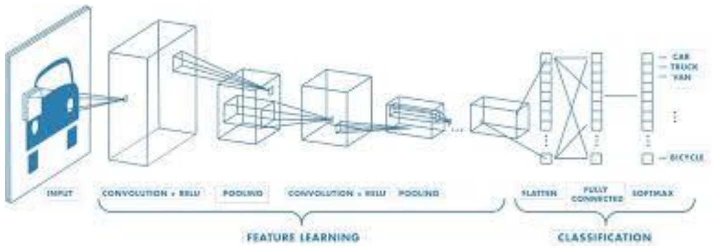
SVMs

Clustering

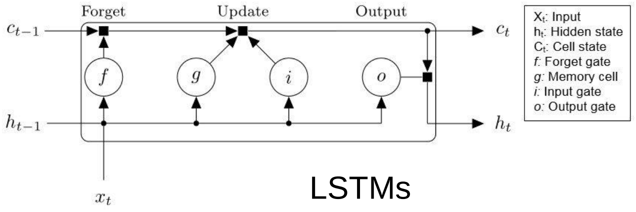


Decision trees

Deep Learning from scratch

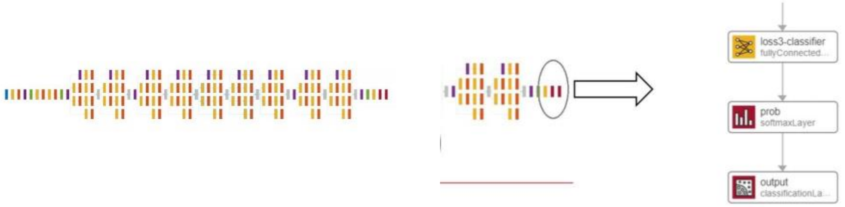


CNNs



LSTMs

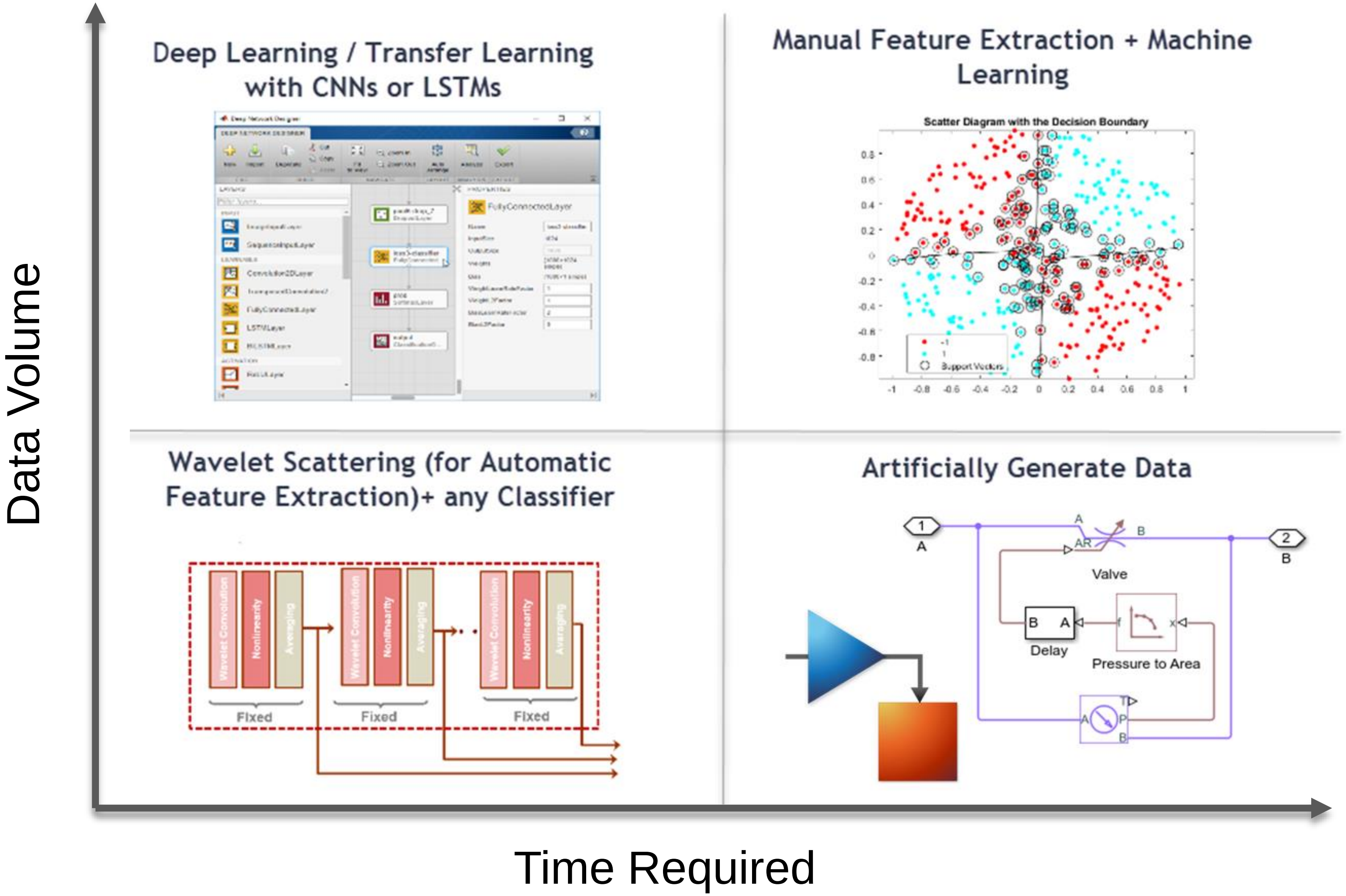
Deep Learning using Transfer Learning



Understanding tradeoffs to building models

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability



Deep Learning / Transfer Learning with CNNs or LSTMs

Manual Feature Extraction + Machine Learning

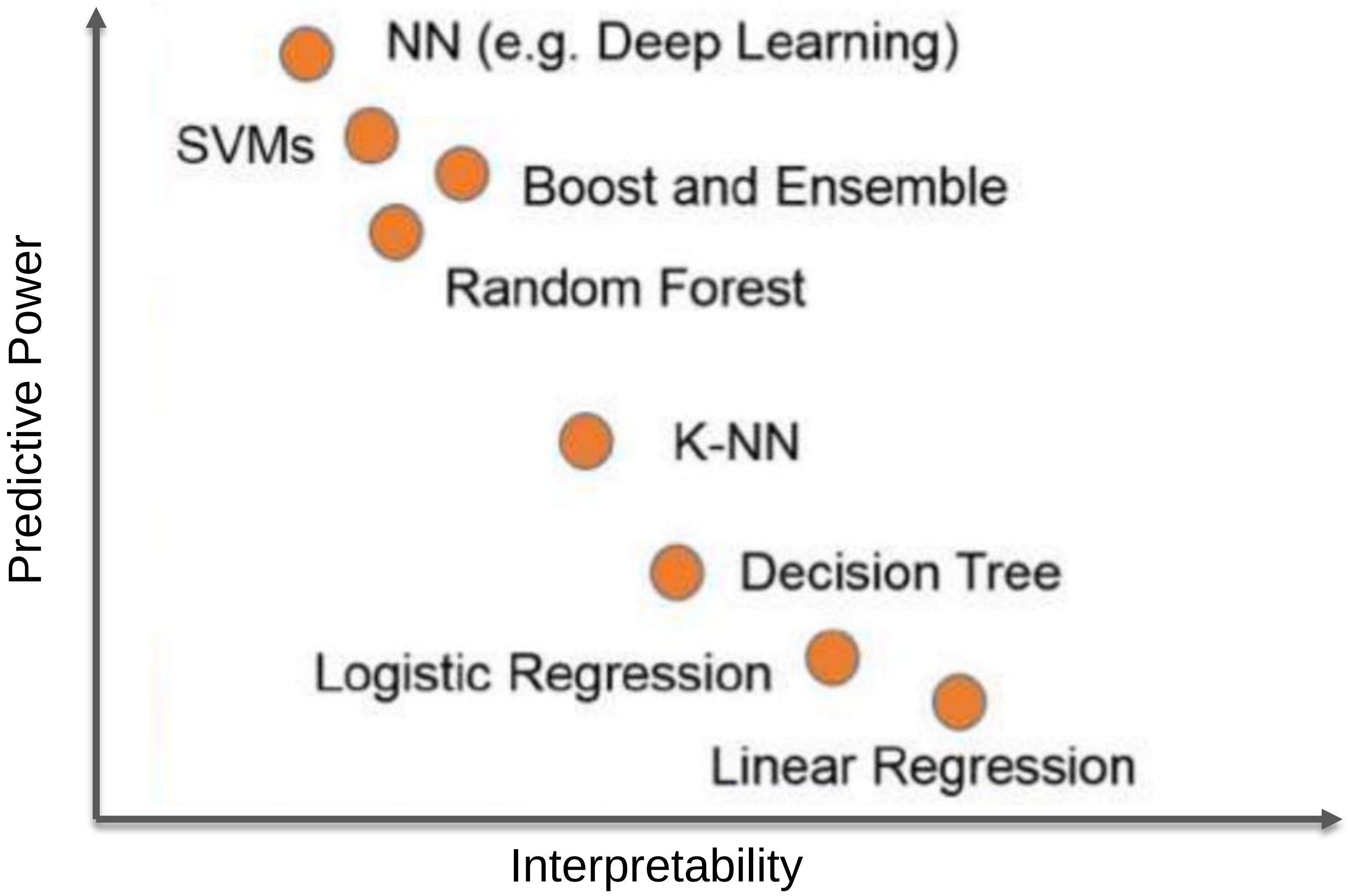
Wavelet Scattering (for Automatic Feature Extraction)+ any Classifier

Artificially Generate Data

Understanding tradeoffs to building models

AI Modeling

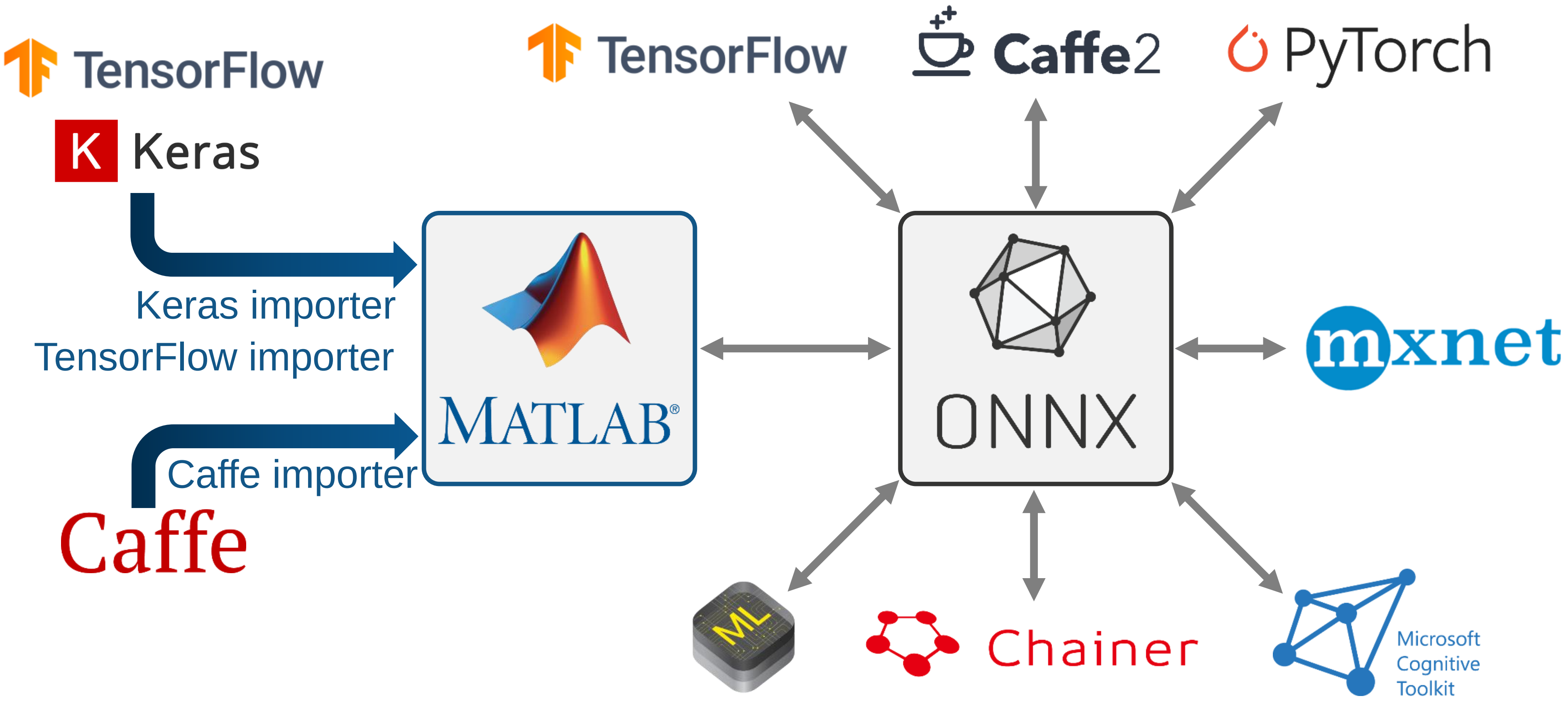
- Model design and tuning
- Hardware accelerated training
- Interoperability



MATLAB interoperates with other frameworks

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability



TensorFlow

Keras

Keras importer

TensorFlow importer

Caffe importer

Caffe

MATLAB

ONNX

TensorFlow

Caffe2

PyTorch

mxnet

ML

Chainer

Microsoft Cognitive Toolkit

Hardware acceleration and scaling are critical for training

Azure

intel

NVIDIA

aws

docker

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

Completing your AI workflow with deployment

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

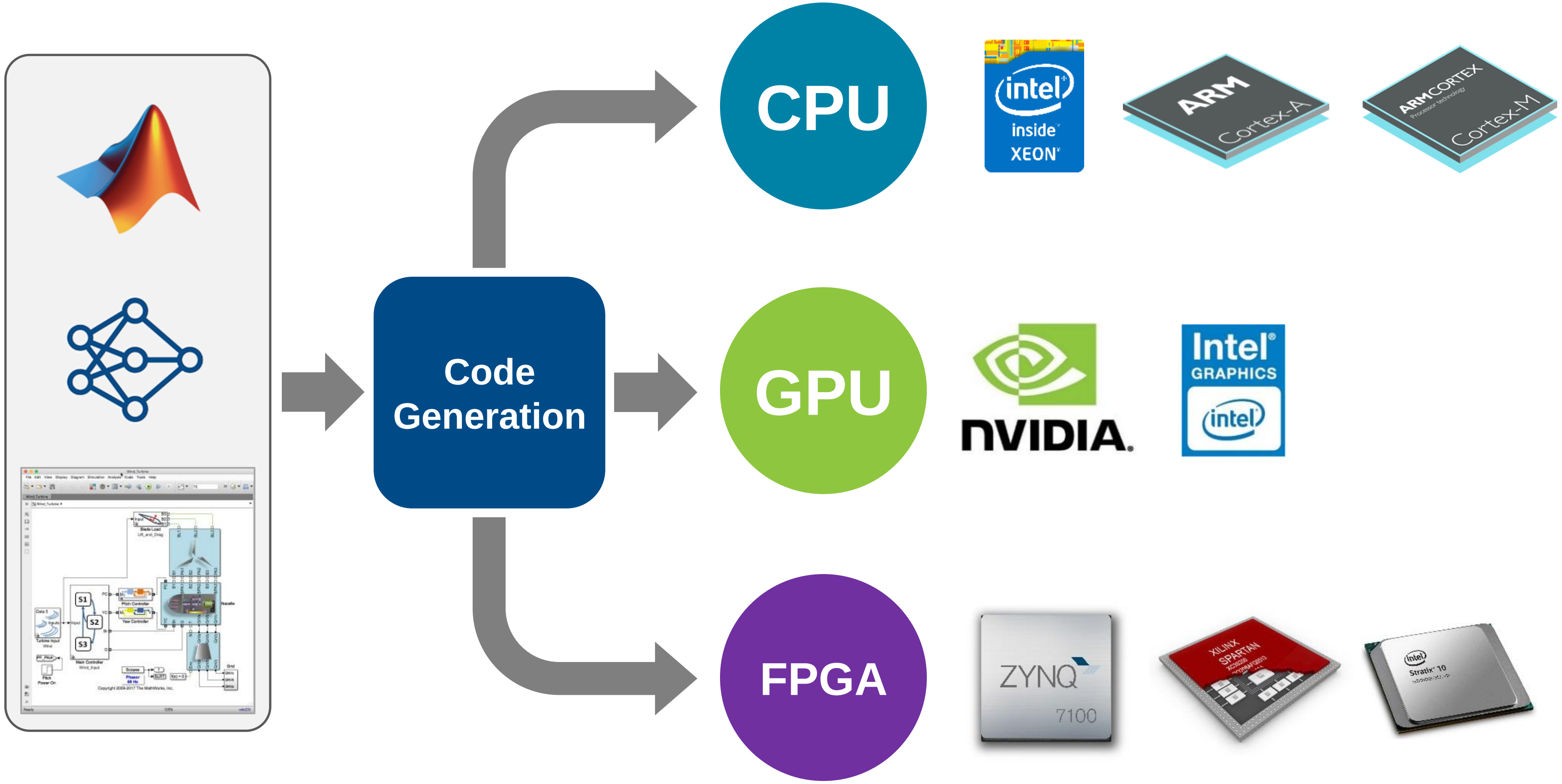
Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

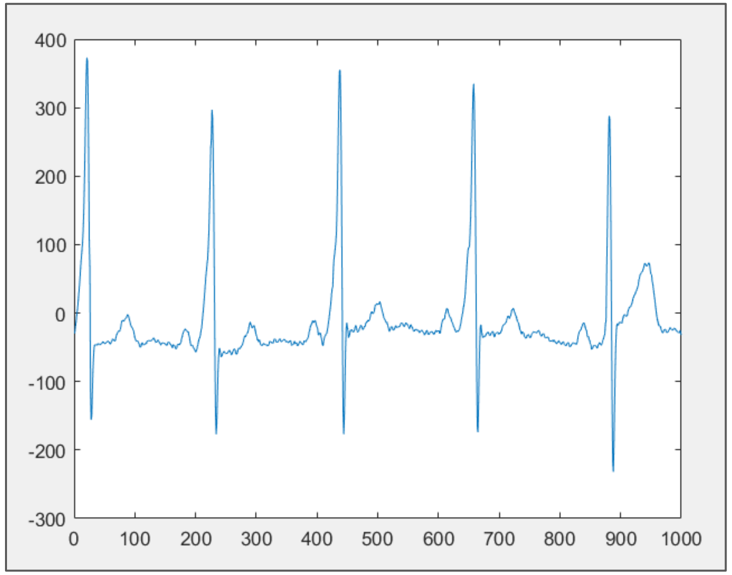
Deploy to any processor with best-in-class performance

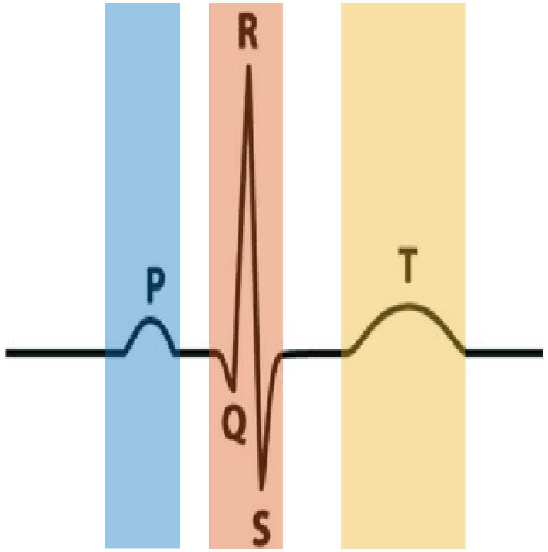
Deployment

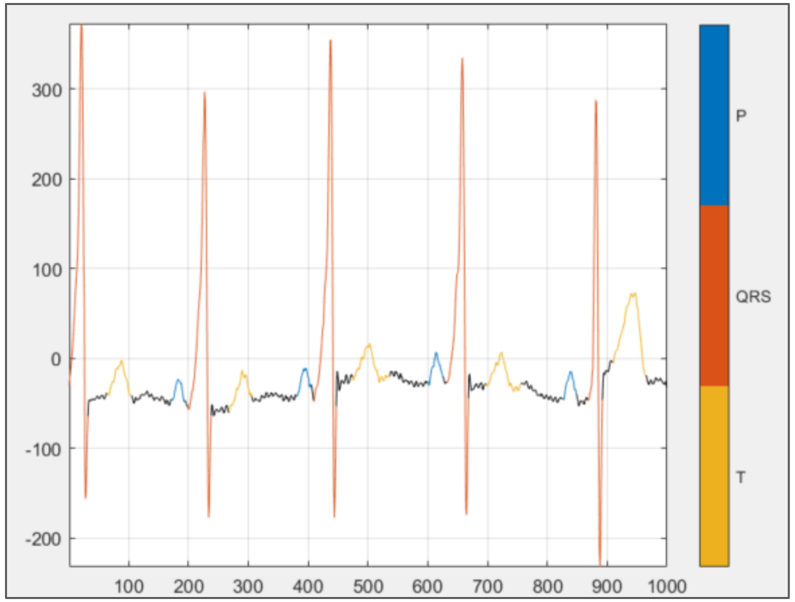
- Embedded devices
- Enterprise systems
- Edge, cloud, desktop



ECG Waveform Segmentation







- ECG signals contain P,QRS and T waves. Identifying these can help with diagnosis and classification.

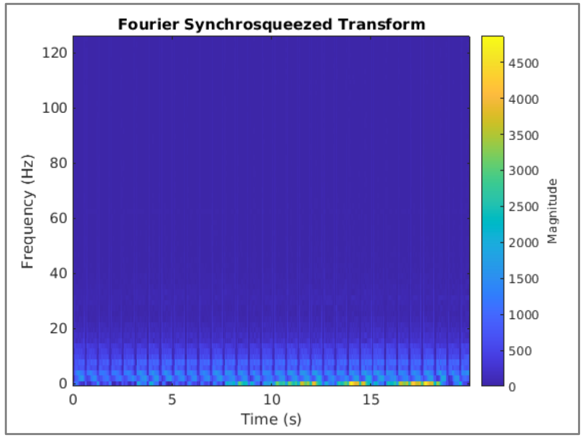
- Dataset contains 210 ECG signals ,~ 15 minutes long, labeled by cardiologist

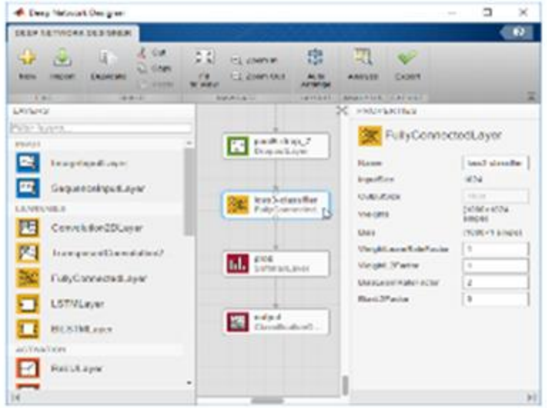
ECG Segmentation using a LSTM Network

Preprocess Data and Feature Extraction

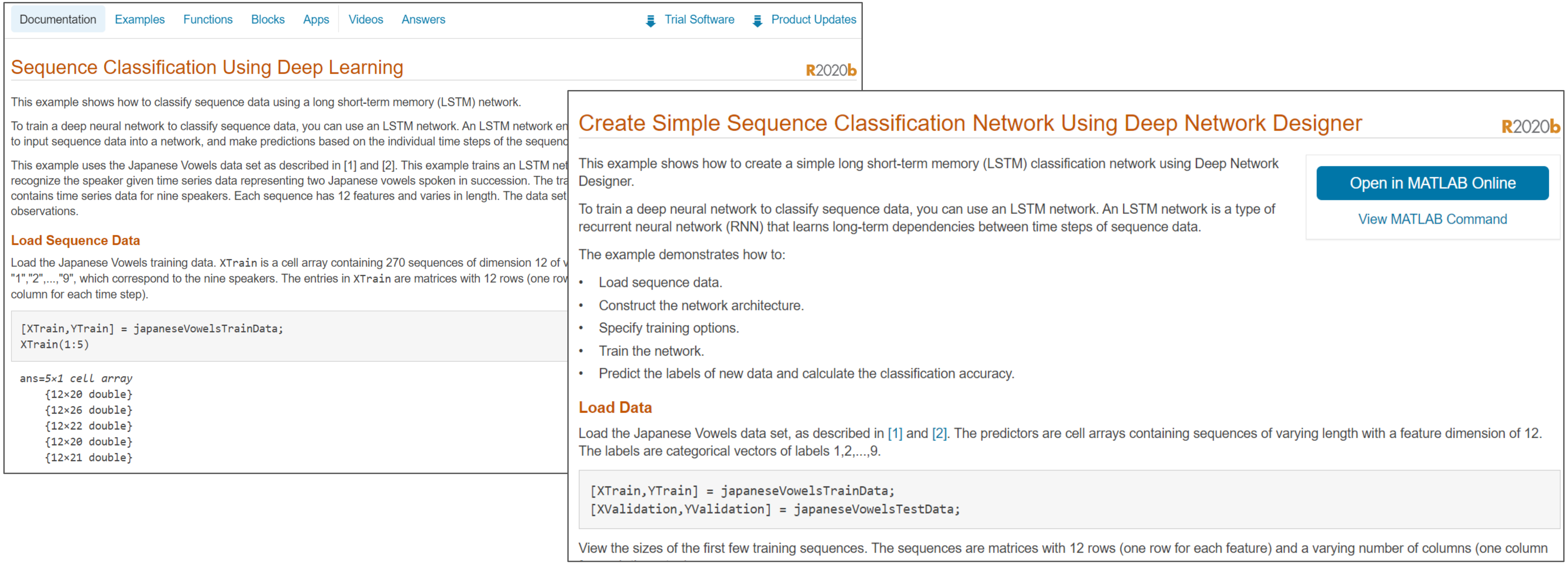
Build Deep Network from Scratch

Deploy & Integrate





Building a Deep Learning Model from Scratch



Sequence Classification Using Deep Learning

R2020b

This example shows how to classify sequence data using a long short-term memory (LSTM) network.

To train a deep neural network to classify sequence data, you can use an LSTM network. An LSTM network en
to input sequence data into a network, and make predictions based on the individual time steps of the sequenc

This example uses the Japanese Vowels data set as described in [1] and [2]. This example trains an LSTM net
recognize the speaker given time series data representing two Japanese vowels spoken in succession. The tra
contains time series data for nine speakers. Each sequence has 12 features and varies in length. The data set
observations.

Load Sequence Data

Load the Japanese Vowels training data. `XTrain` is a cell array containing 270 sequences of dimension 12 of v
"1","2",...,"9", which correspond to the nine speakers. The entries in `XTrain` are matrices with 12 rows (one row
column for each time step).

```
[XTrain,YTrain] = japaneseVowelsTrainData;
XTrain(1:5)
```

```
ans=5×1 cell array
    {12×20 double}
    {12×26 double}
    {12×22 double}
    {12×20 double}
    {12×21 double}
```

Create Simple Sequence Classification Network Using Deep Network Designer

R2020b

This example shows how to create a simple long short-term memory (LSTM) classification network using Deep Network Designer.

To train a deep neural network to classify sequence data, you can use an LSTM network. An LSTM network is a type of recurrent neural network (RNN) that learns long-term dependencies between time steps of sequence data.

The example demonstrates how to:

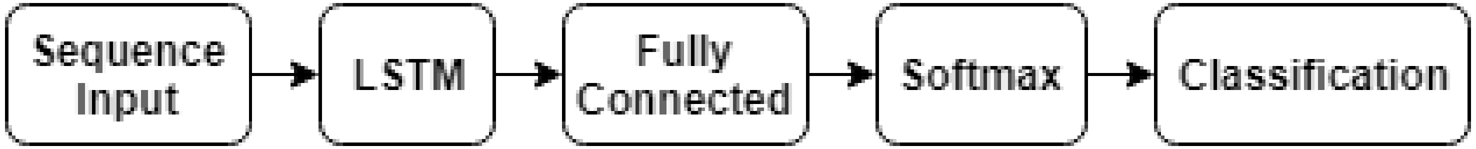
- Load sequence data.
- Construct the network architecture.
- Specify training options.
- Train the network.
- Predict the labels of new data and calculate the classification accuracy.

Load Data

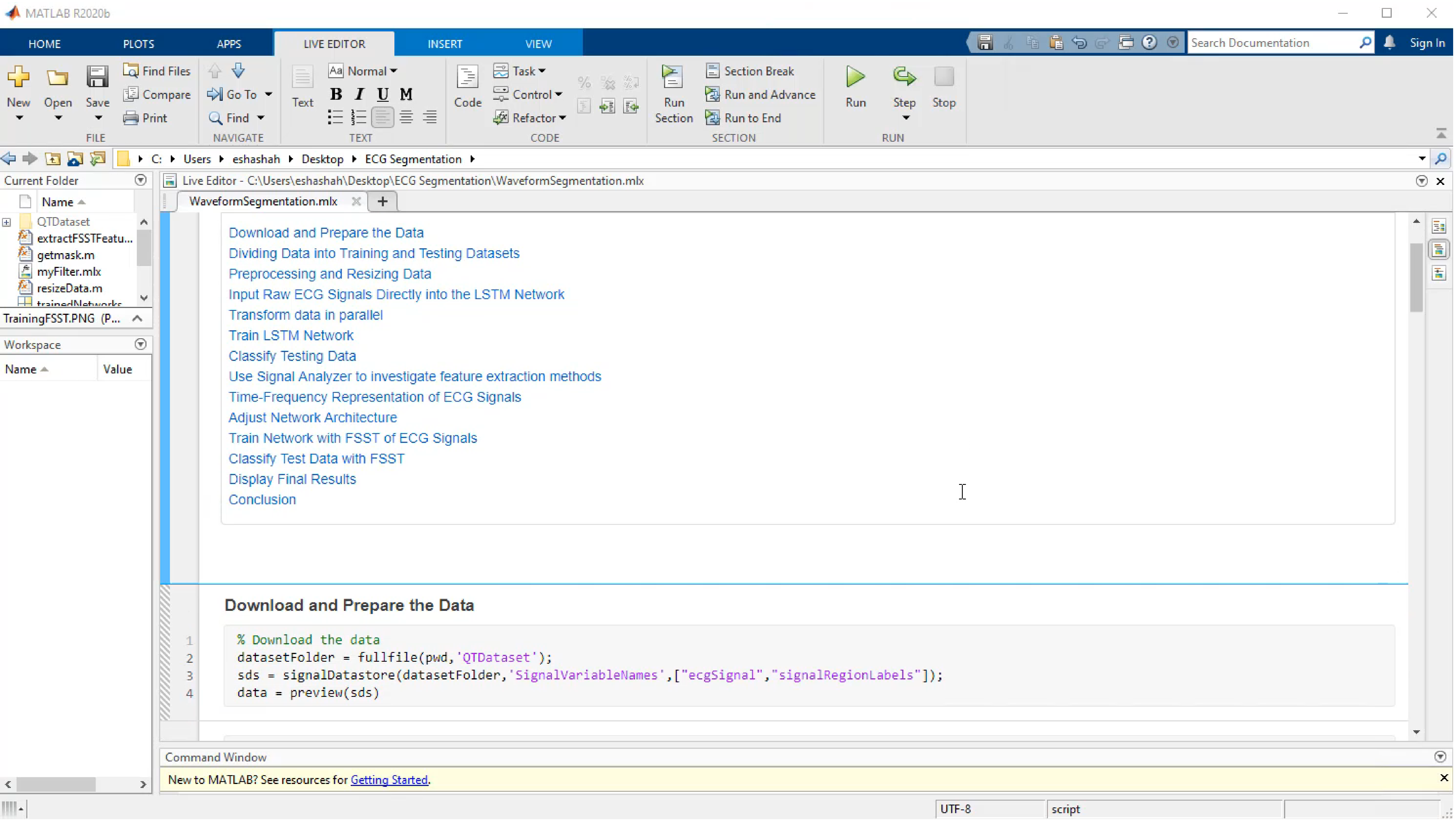
Load the Japanese Vowels data set, as described in [1] and [2]. The predictors are cell arrays containing sequences of varying length with a feature dimension of 12. The labels are categorical vectors of labels 1,2,...,9.

```
[XTrain,YTrain] = japaneseVowelsTrainData;
[XValidation,YValidation] = japaneseVowelsTestData;
```

View the sizes of the first few training sequences. The sequences are matrices with 12 rows (one row for each feature) and a varying number of columns (one column



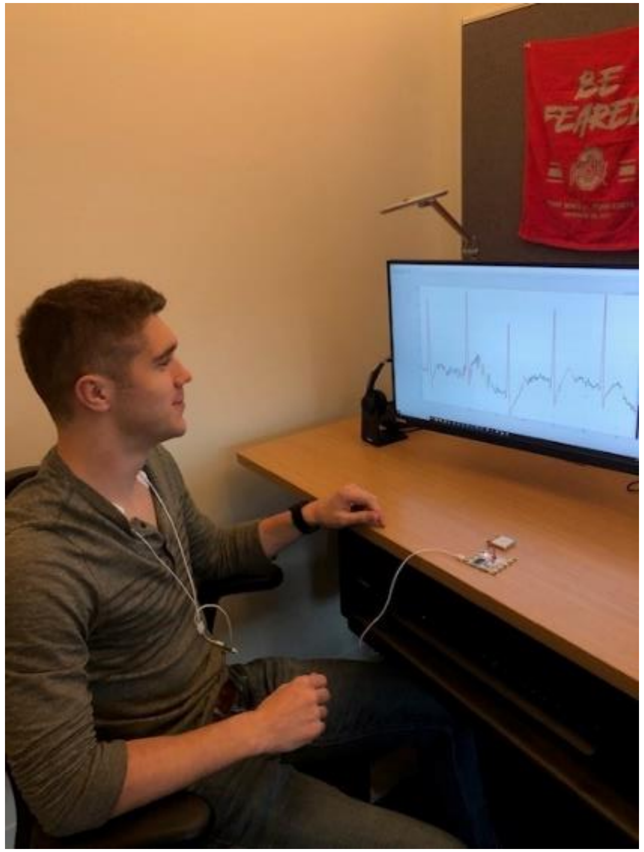
Sequence Input → LSTM → Fully Connected → Softmax → Classification

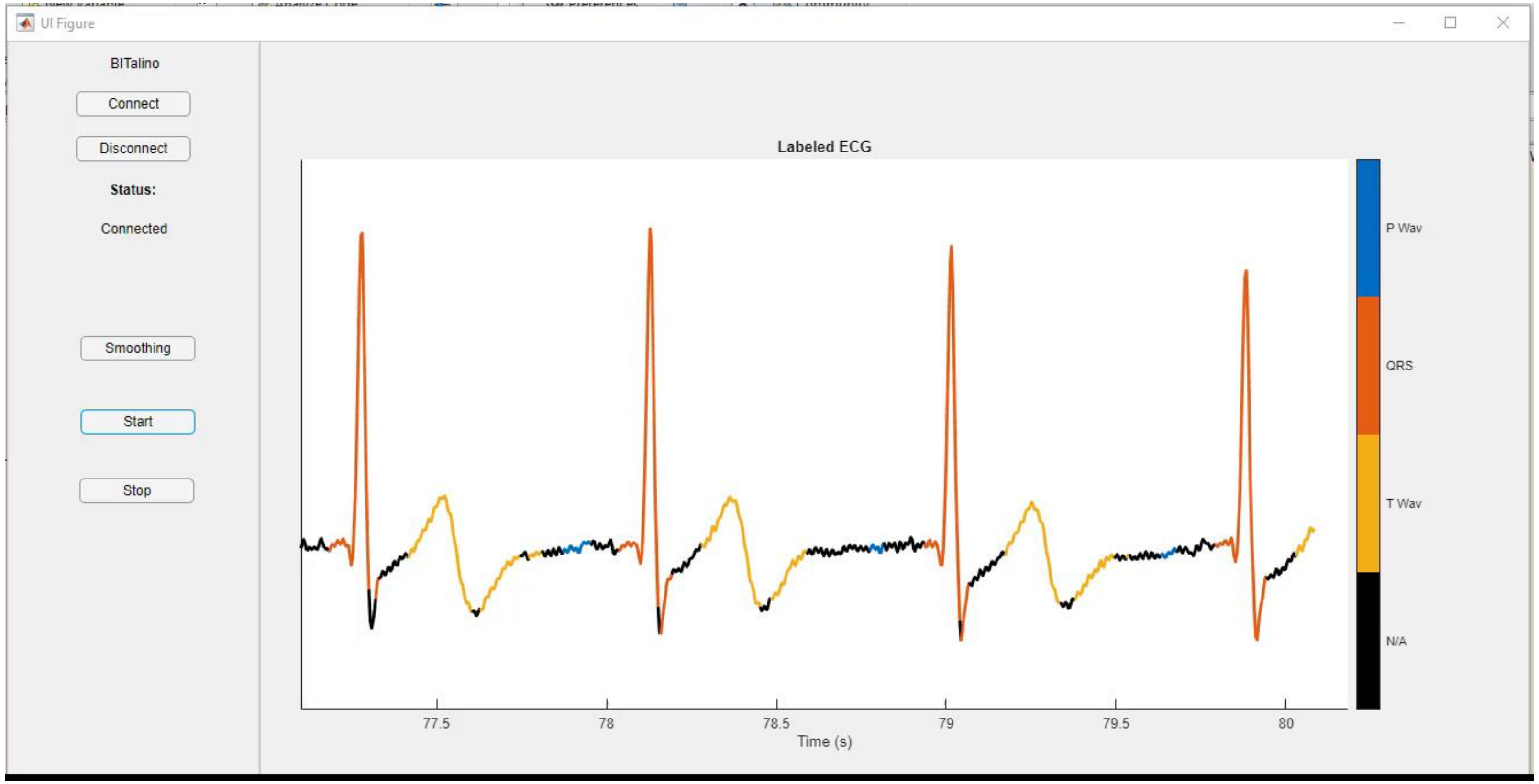


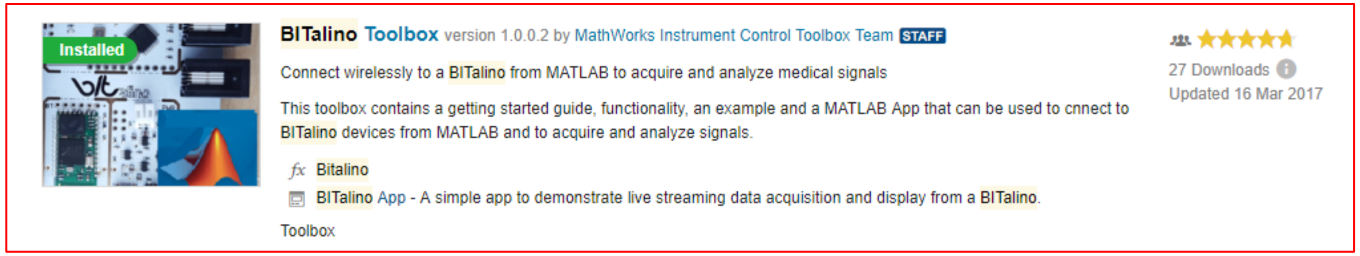
Download and Prepare the Data

```
% Download the data
datasetFolder = fullfile(pwd,'QTDataset');
sds = signalDatastore(datasetFolder,'SignalVariableNames',["ecgSignal","signalRegionLabels"]);
data = preview(sds)
```

Getting data from Bitalino card







BITalino Toolbox version 1.0.0.2 by MathWorks Instrument Control Toolbox Team

27 Downloads
Updated 16 Mar 2017

Connect wirelessly to a BITalino from MATLAB to acquire and analyze medical signals

This toolbox contains a getting started guide, functionality, an example and a MATLAB App that can be used to cnnect to BITalino devices from MATLAB and to acquire and analyze signals.

- *fx* Bitalino
- BITalino App - A simple app to demonstrate live streaming data acquisition and display from a BITalino.

Toolbox

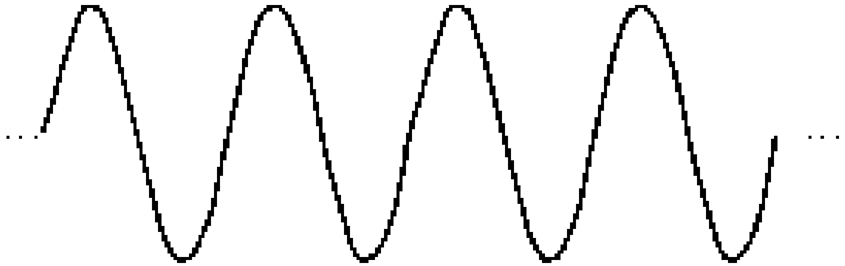
Deployment to a Raspberry Pi Board

https://www.mathworks.com/help/deeplearning/ug/code-generation-for-ecg-segmentation-using-deep-learning-on-raspberry-pi.html

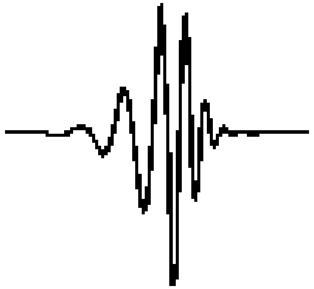
- Raspberry Pi 3 B+ with ARM Cortex A processor
- Compute Library from ARM
  - Repository of low-level optimized functions
  - For ARM Cortex A processors and ARM Mali family of GPUs

Using Wavelets for AI

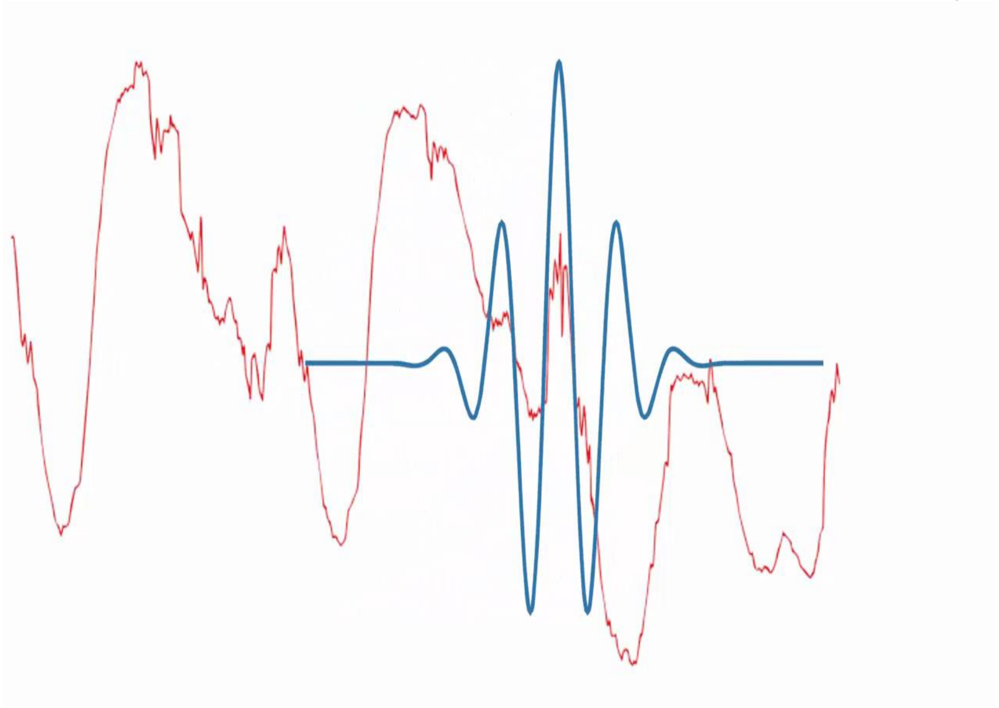
Why are wavelets and multiscale techniques useful?



Sine Wave



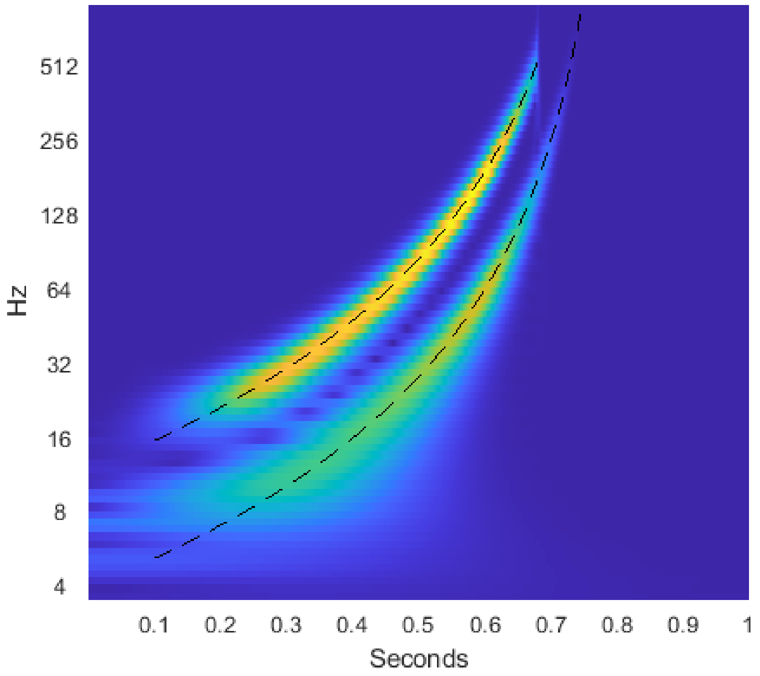
Wavelet (db10)



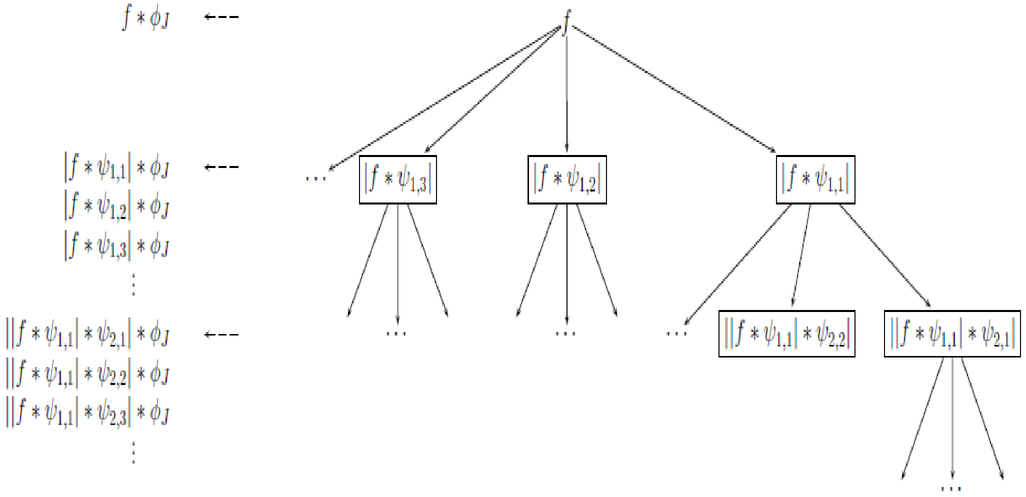
- Efficient representations for data characterized by transient components and long duration trends or oscillations

- Provides state-of-the- art results for :
  - Anomaly detection and Health Monitoring
  - Biomedical Signal Analysis
  - Seismic Analysis
  - Radar and Comms applications
  - Financial Analysis

Which wavelet techniques can be used for AI?

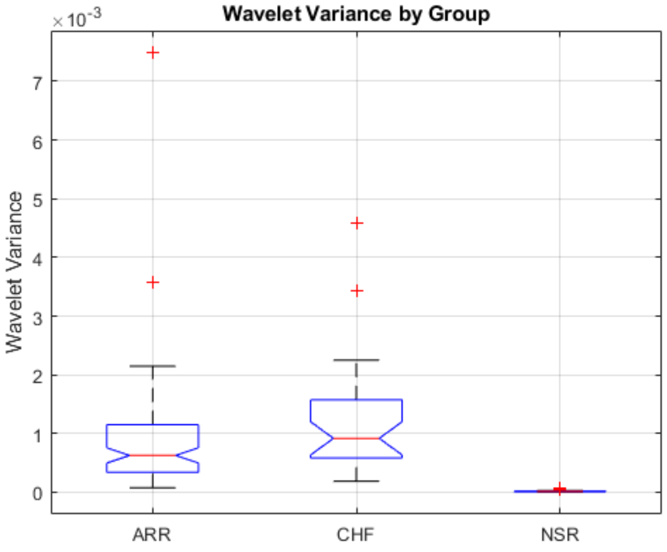


Continuous Wavelet Transform



$f * \phi_J$
$|f * \psi_{1,1}| * \phi_J$
$|f * \psi_{1,2}| * \phi_J$
$|f * \psi_{1,3}| * \phi_J$
$||f * \psi_{1,1}| * \psi_{2,1}| * \phi_J$
$||f * \psi_{1,1}| * \psi_{2,2}| * \phi_J$
$||f * \psi_{1,1}| * \psi_{2,3}| * \phi_J$
f
$|f * \psi_{1,3}|$
$|f * \psi_{1,2}|$
$|f * \psi_{1,1}|$
$||f * \psi_{1,1}| * \psi_{2,2}|$
$||f * \psi_{1,1}| * \psi_{2,1}|$

Wavelet Scattering



Wavelet Statistics

..and many more

Crack Identification in Pavements with Wavelets

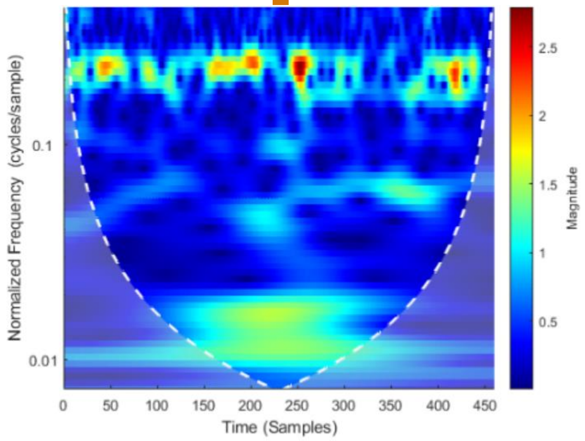
- Image analysis of pavements is not always feasible
- Data is collected using accelerometer sensor installed in cars
- Dataset: Mendeley Data open data repository – 327 samples
- Data is collected at different car speeds for varying sizes of pavement cracks

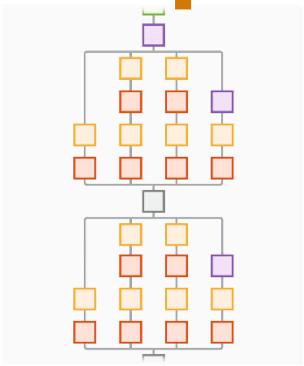
Crack Identification with Pretrained Network

Preprocess Data and Feature Extraction

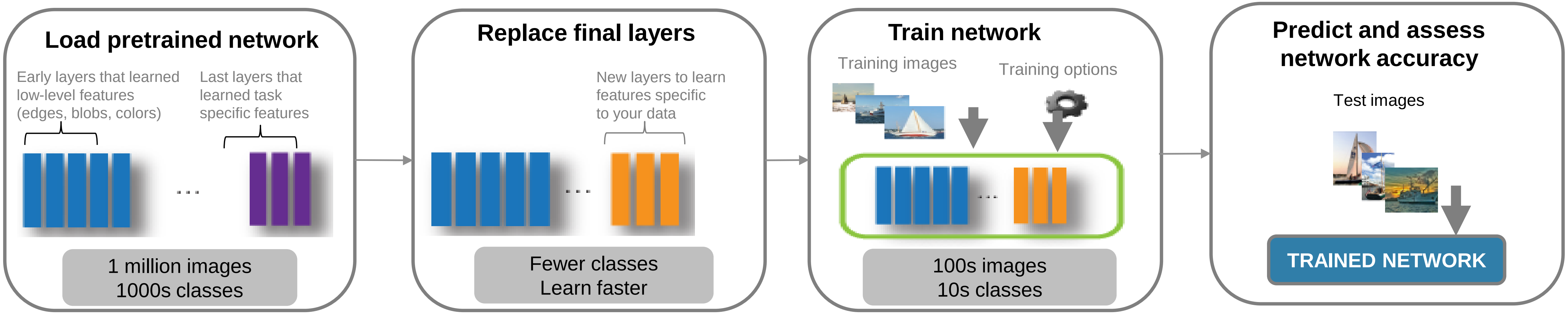
Use Deep Learning with Transfer Learning

Wavelet Transform





Using Transfer Learning



Load pretrained network

Early layers that learned low-level features (edges, blobs, colors)

Last layers that learned task specific features

1 million images
1000s classes

Replace final layers

New layers to learn features specific to your data

Fewer classes
Learn faster

Train network

Training images

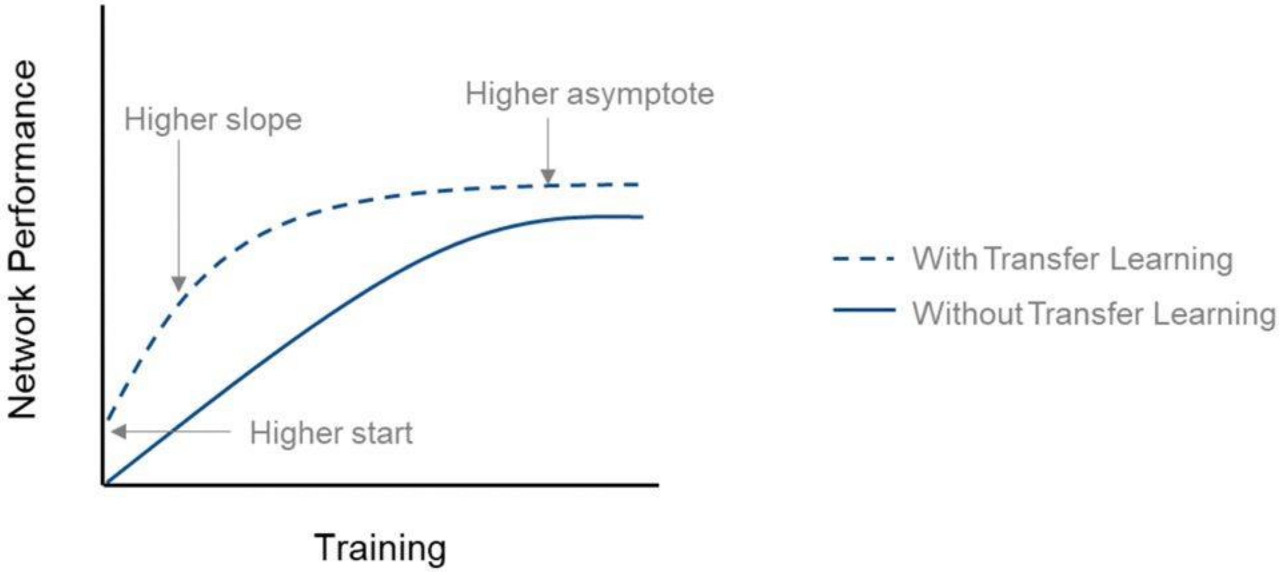
Training options

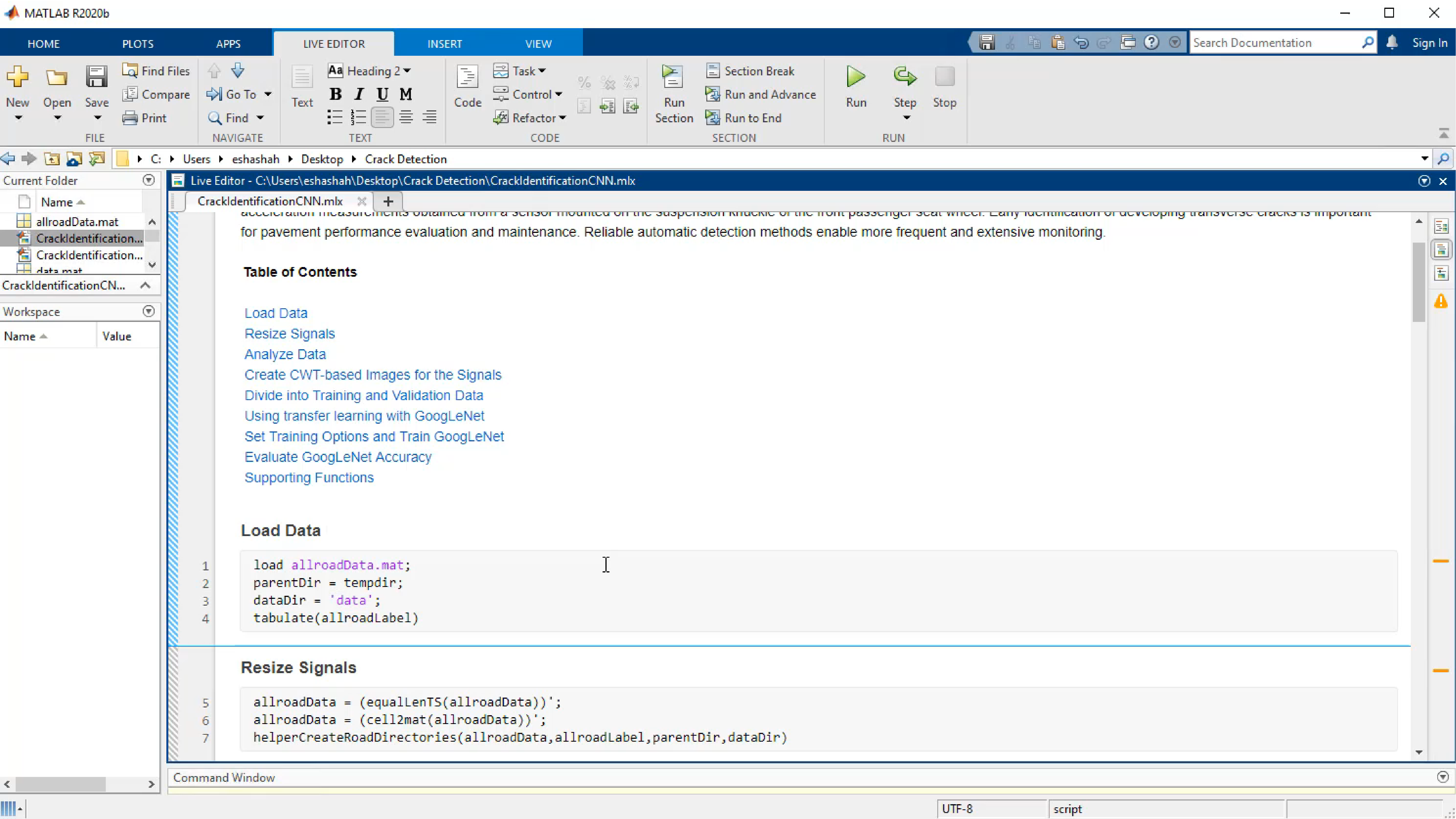
100s images
10s classes

Predict and assess network accuracy

Test images

TRAINED NETWORK





acceleration measurements obtained from a sensor mounted on the suspension knuckle of the front passenger seat wheel. Early identification of developing transverse cracks is important for pavement performance evaluation and maintenance. Reliable automatic detection methods enable more frequent and extensive monitoring.

Table of Contents

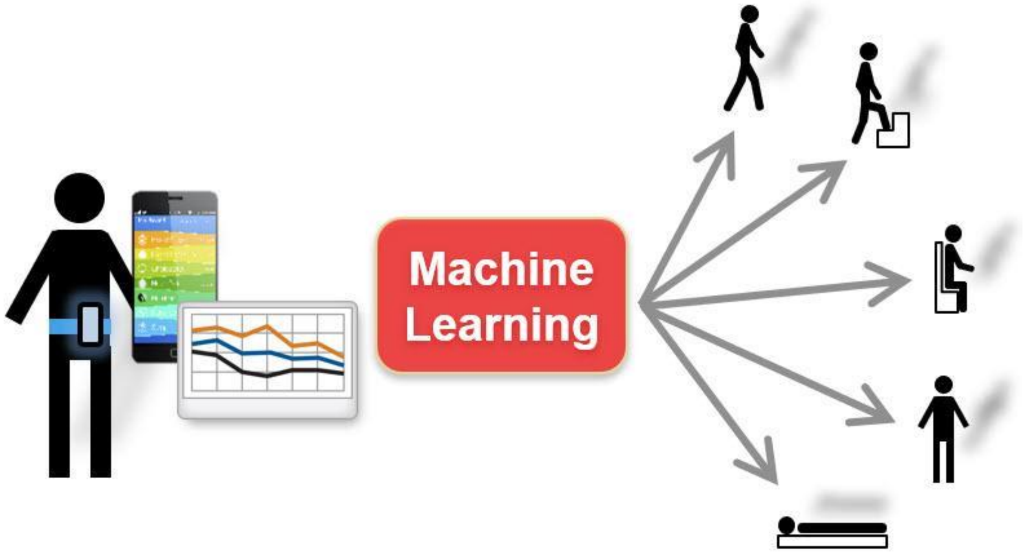
Load Data

```
load allroadData.mat;
parentDir = tempdir;
dataDir = 'data';
tabulate(allroadLabel)
```

Resize Signals

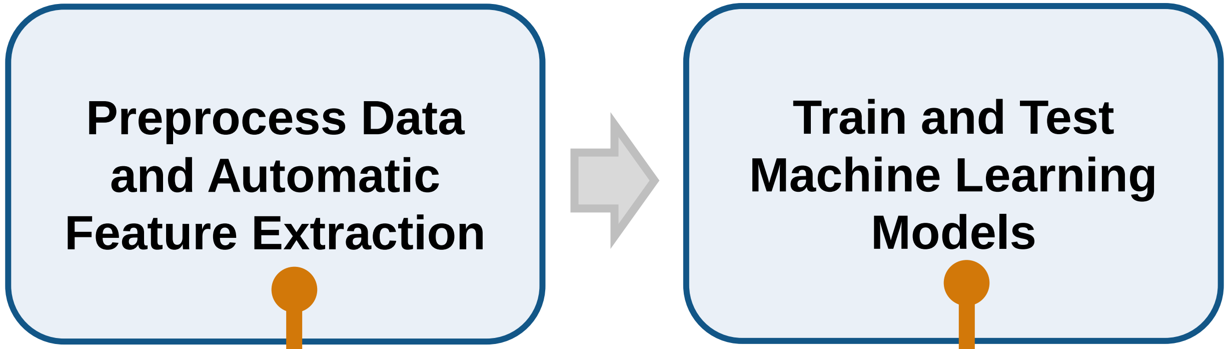
```
allroadData = (equalLenTS(allroadData))';
allroadData = (cell2mat(allroadData))';
helperCreateRoadDirectories(allroadData,allroadLabel,parentDir,dataDir)
```

Human Activity Classification



- Classify from 5 activities:
  - Walking
  - Climbing upstairs
  - Going downstairs
  - Lying down
  - standing
- Data captured from sensors on mobile phones

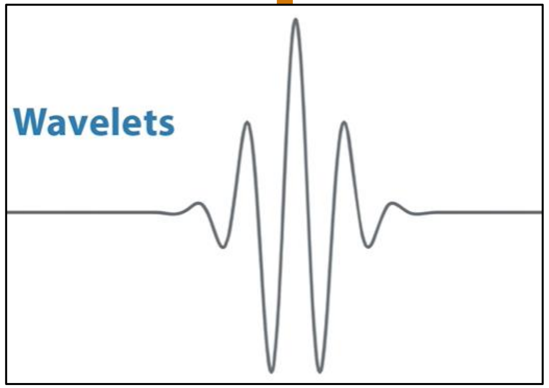
Human Activity Classification with AutoML

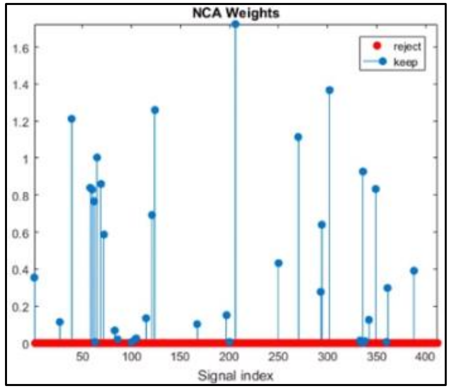


Preprocess Data and Automatic Feature Extraction

Train and Test Machine Learning Models

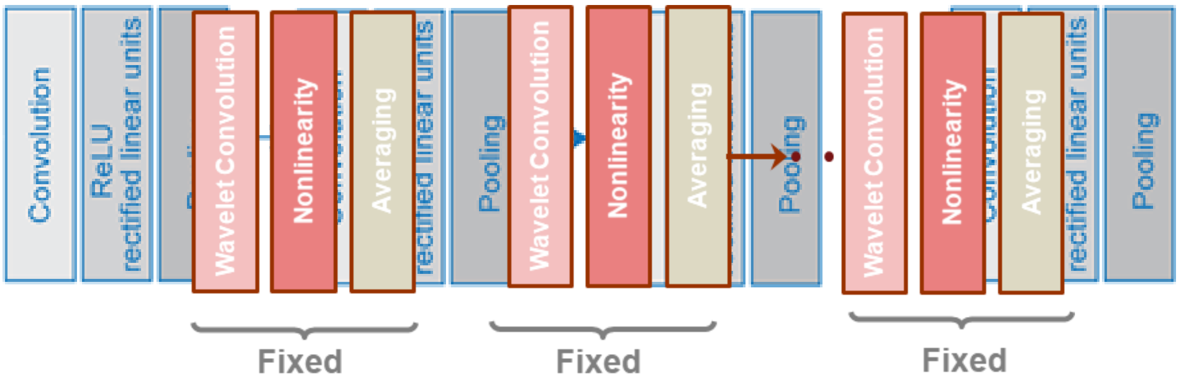
Wavelet Scattering

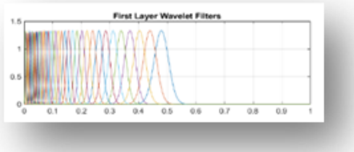


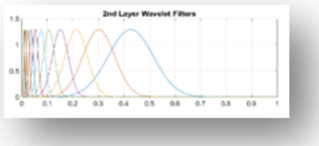


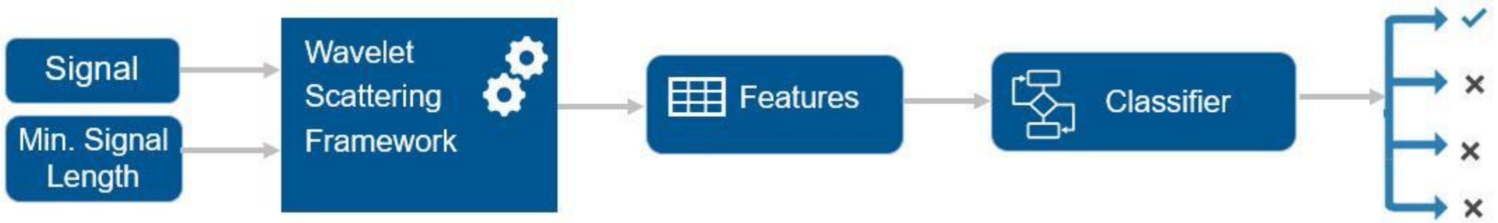
Feature and Model Selection

Wavelet Scattering









Signal

Min. Signal Length

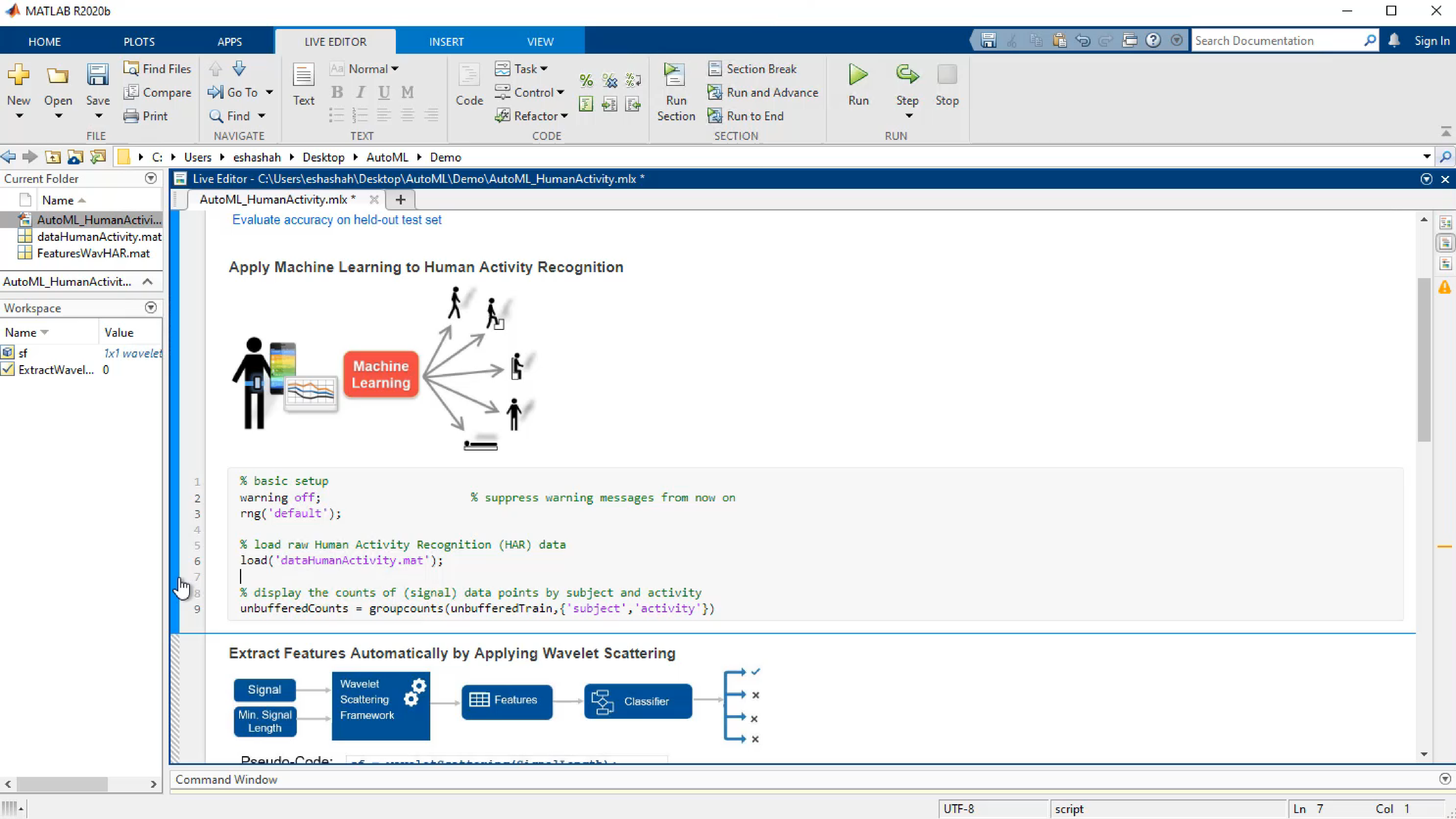
Wavelet Scattering Framework

Features

Classifier

Pseudo-Code:

```
sf = waveletScattering(SignalLength);
Loop over signal
   waveletFeature = featureMatrix(sf,signal)
   Append waveletFeature to feature table
   Add labels
end
```



Evaluate accuracy on held-out test set

Apply Machine Learning to Human Activity Recognition

```
% basic setup
warning off;                        % suppress warning messages from now on
rng('default');

% load raw Human Activity Recognition (HAR) data
load('dataHumanActivity.mat');

% display the counts of (signal) data points by subject and activity
unbufferedCounts = groupcounts(unbufferedTrain,{'subject','activity'})
```

Extract Features Automatically by Applying Wavelet Scattering

Pseudo-Code:

MATLAB supports the entire AI-driven system design

Data Preparation

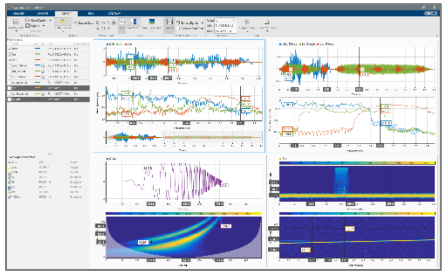
- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

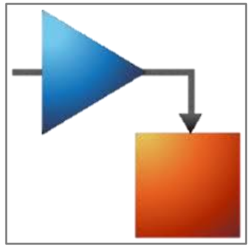
- Model design and tuning
- Hardware accelerated training
- Interoperability

Deployment

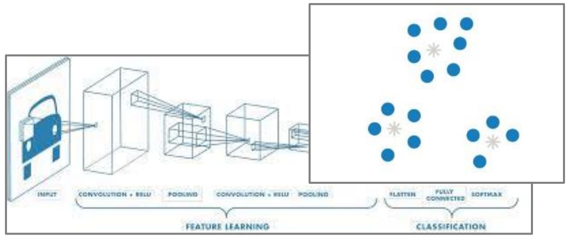
- Embedded devices
- Enterprise systems
- Edge, cloud, desktop



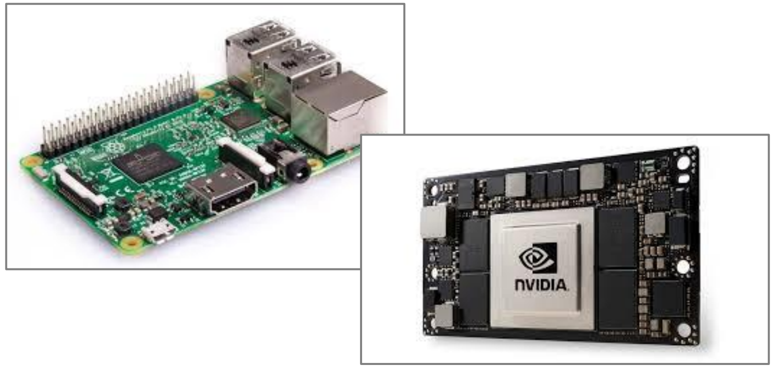
Signal Processing apps



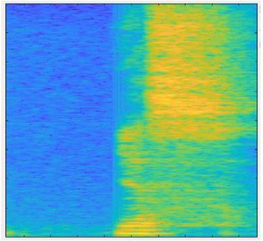
Generate Data



Quickly build models

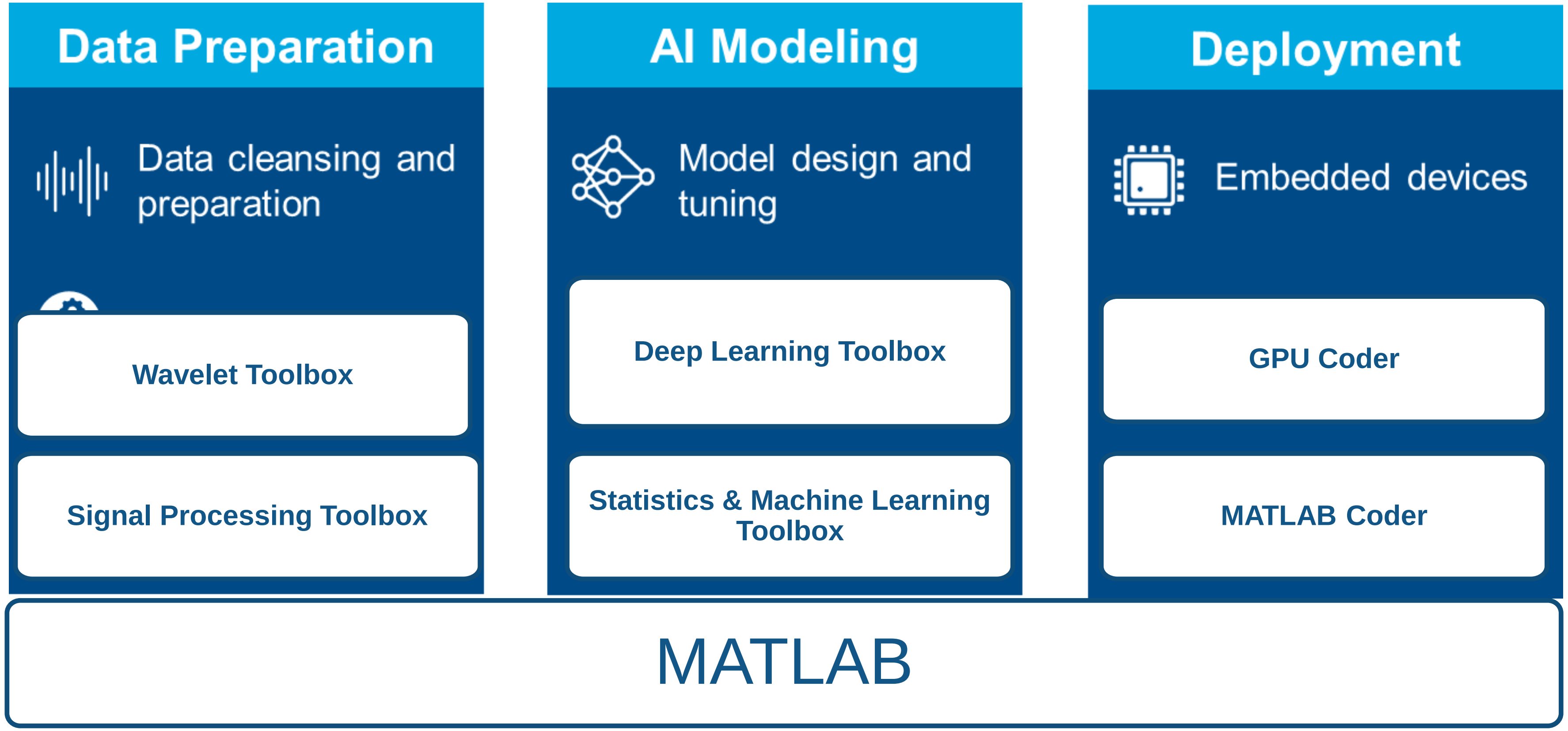


Deploy to targets with code generation



Feature Extraction Techniques

Accelerate training



Data Preparation

Data cleansing and preparation

Wavelet Toolbox

Signal Processing Toolbox

AI Modeling

Model design and tuning

Deep Learning Toolbox

Statistics & Machine Learning Toolbox

Deployment

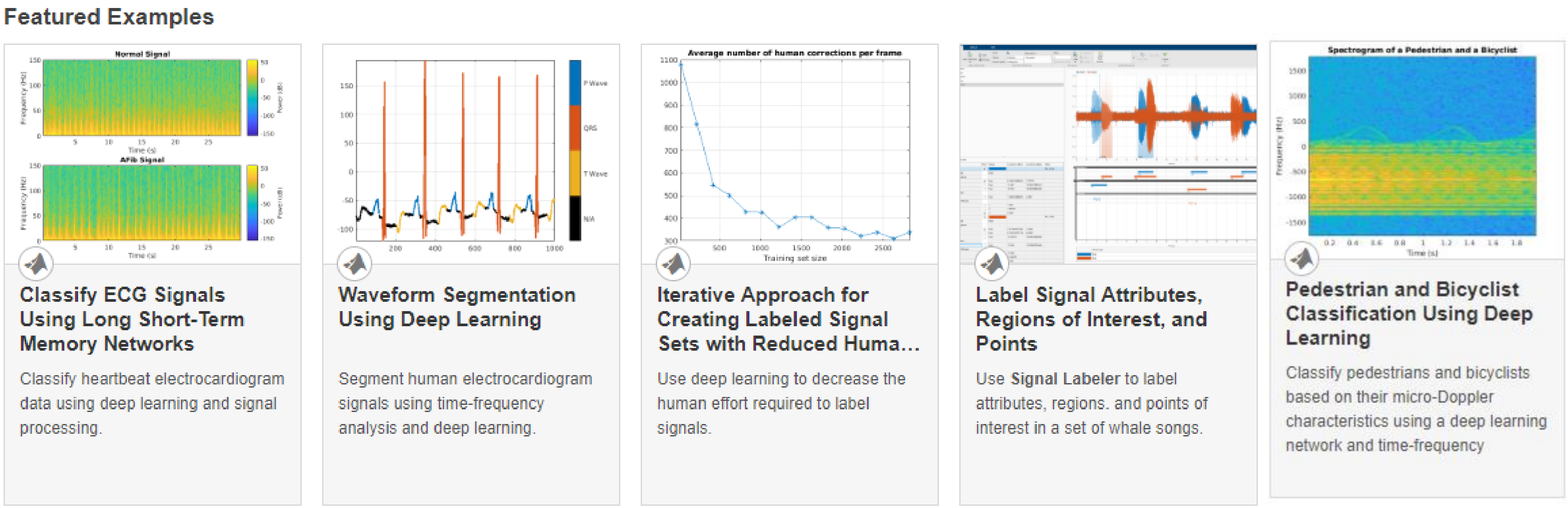
Embedded devices

GPU Coder

MATLAB Coder

MATLAB

Many resources to get started with



Featured Examples

Classify ECG Signals Using Long Short-Term Memory Networks

Classify heartbeat electrocardiogram data using deep learning and signal processing.

Waveform Segmentation Using Deep Learning

Segment human electrocardiogram signals using time-frequency analysis and deep learning.

Iterative Approach for Creating Labeled Signal Sets with Reduced Huma…

Use deep learning to decrease the human effort required to label signals.

Label Signal Attributes, Regions of Interest, and Points

Use **Signal Labeler** to label attributes, regions, and points of interest in a set of whale songs.

Pedestrian and Bicyclist Classification Using Deep Learning

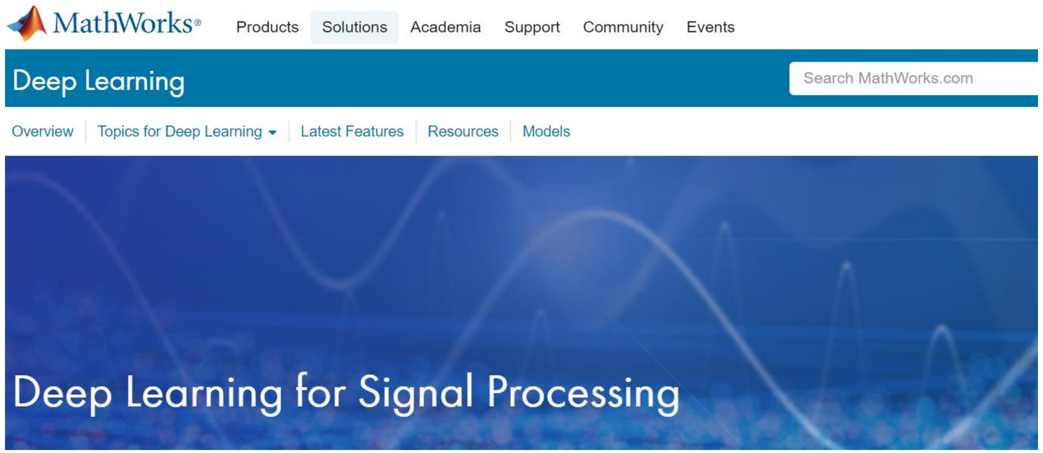
Classify pedestrians and bicyclists based on their micro-Doppler characteristics using a deep learning network and time-frequency

WHITE PAPER

Deep Learning for Signal Processing with MATLAB

MathWorks

https://www.mathworks.com/help/signal/examples.html
https://www.mathworks.com/help/wavelet/examples.html



MathWorks® Products Solutions Academia Support Community Events

Deep Learning

Search MathWorks.com

Overview | Topics for Deep Learning ▾ | Latest Features | Resources | Models

Deep Learning for Signal Processing

https://www.mathworks.com/solutions/deep-learning/deep-learning-signal-processing.html

MATLAB과 함께하는 딥러닝 4주 완성 부트캠프 일정

7월 7일	7월 14일	7월 21일	7월 22일	7월 28일
영상 분석을 위한 딥러닝	신호처리를 위한 머신 러닝과 딥러닝	딥러닝 프로젝트를 위한 데이터 준비 기법	MATLAB Deep Learning Day 2021	MATLAB으로 시작하는 강화학습

딥러닝 부트캠프 응용편			
딥러닝 기반 이상탐지 기법	라이다 및 레이다를 위한 딥러닝	무선통신을 위한 딥러닝	바이오, 의료분야를 위한 딥러닝

https://bit.ly/3hfSm24

오늘 등록하세요!

감사합니다