MATLAB EXPO

신호처리 응용프로그램을 위한 데이터 중심 AI (Data-Centric AI)

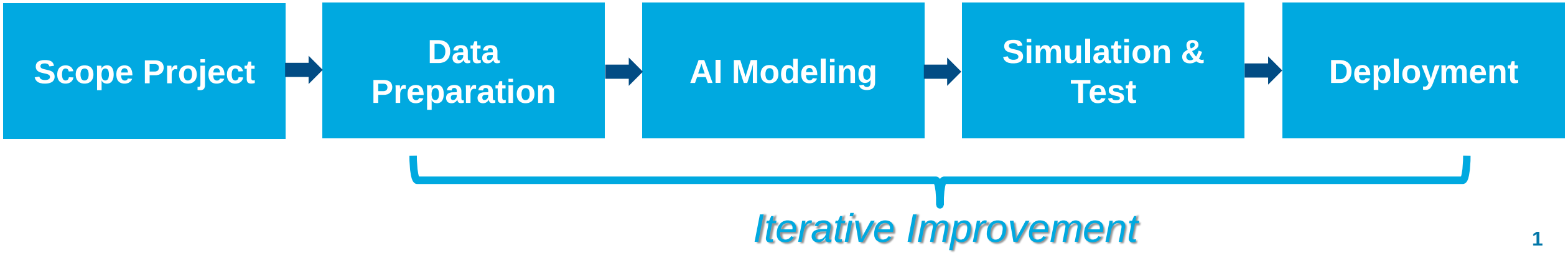
송완빈 과장, 매스웍스코리아

MathWorks®

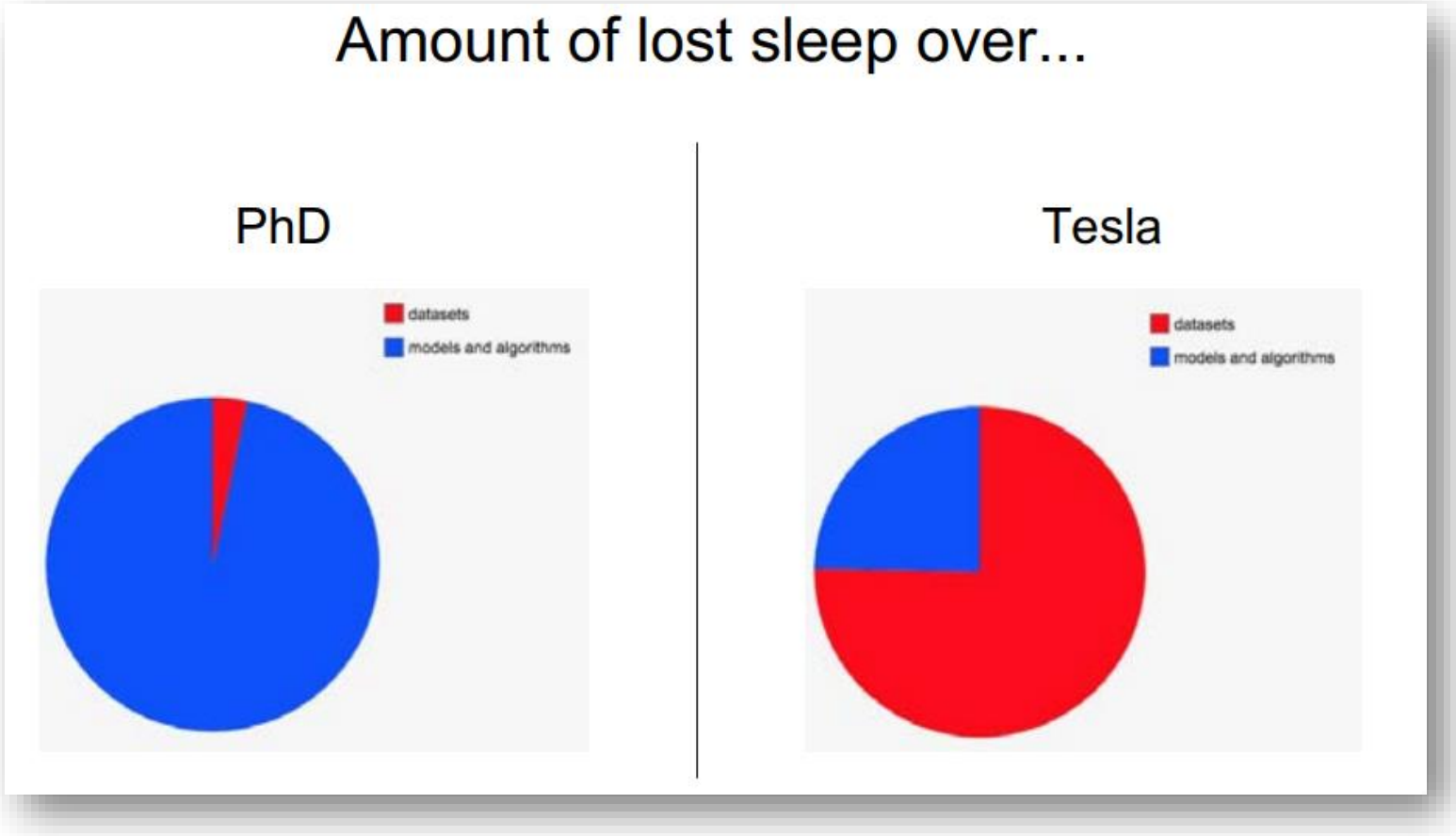
Software System Design

- Traditional Software

- AI Software = **Code (Model/Algorithm)** + **Data**



Industry and Research Invest in AI in Different Ways
Better Models or Better Data?

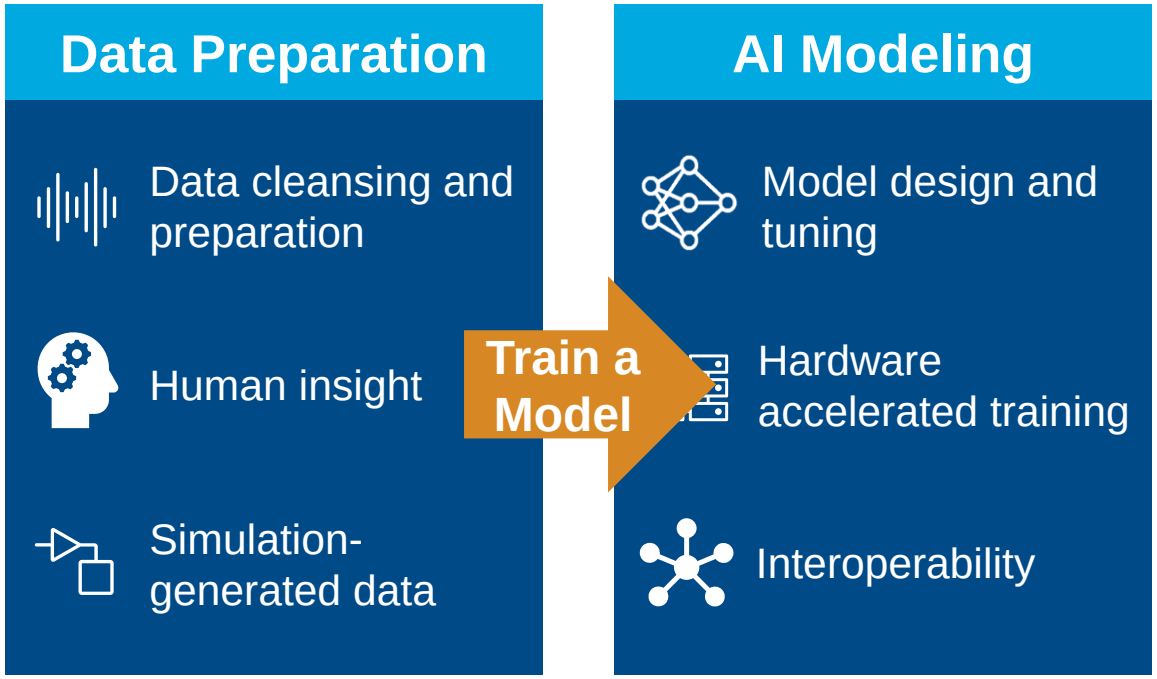


Andrej Karpathy – Building the Software 2.0 Stack (Spark+AI Summit 2018)

Data-Centric AI in 2022 – Trend Gaining Pace and Visibility

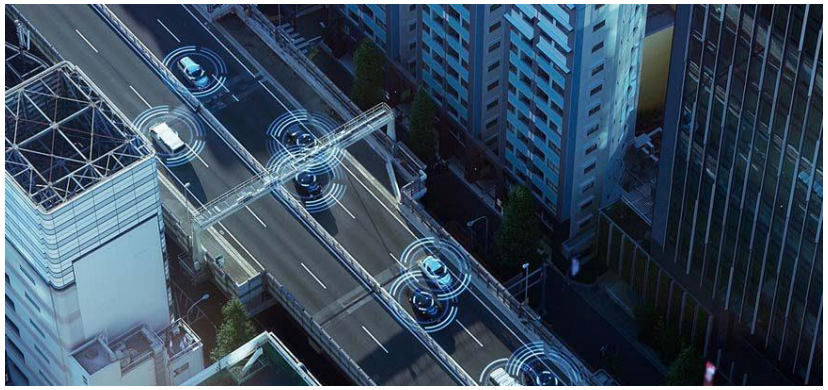
https://spectrum.ieee.org/andrew-ng-data-centric-ai

What is Data-Centric AI?



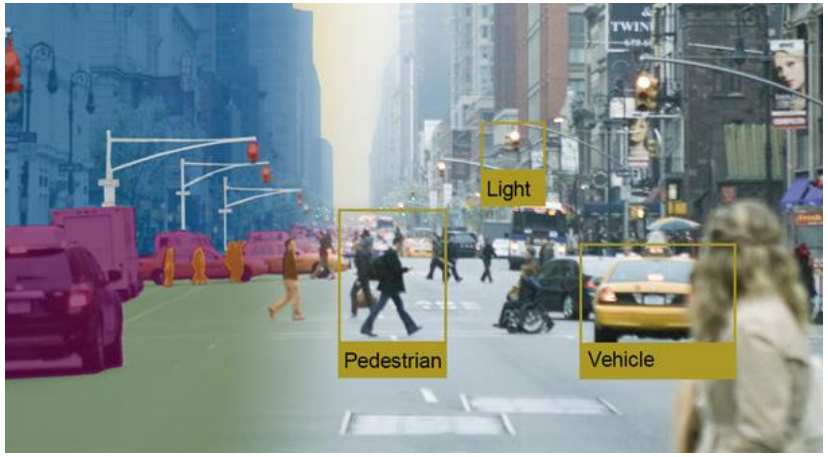
- Error Analysis
  - ***Model-centric***
    - How can I tune the model architecture?
  - ***Data-centric***
    - How can I modify my data?
- Data-centric AI
  - The consistency of the data is paramount. Use tools to improve the data quality
  - Hold the code fixed and iteratively improve the data.

Most existing AI resources support few applications



Autonomous driving

Language modeling

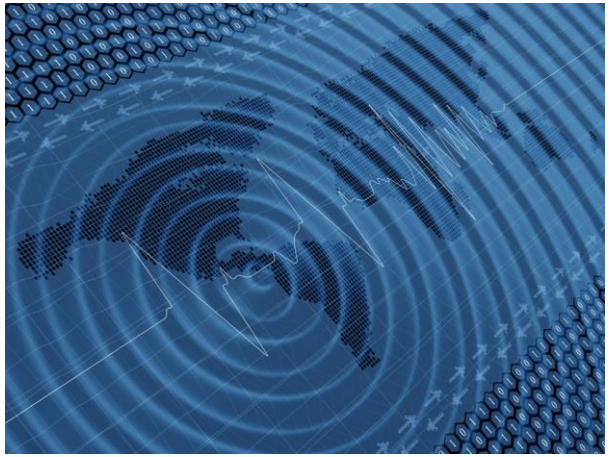


Computer vision

Speech recognition

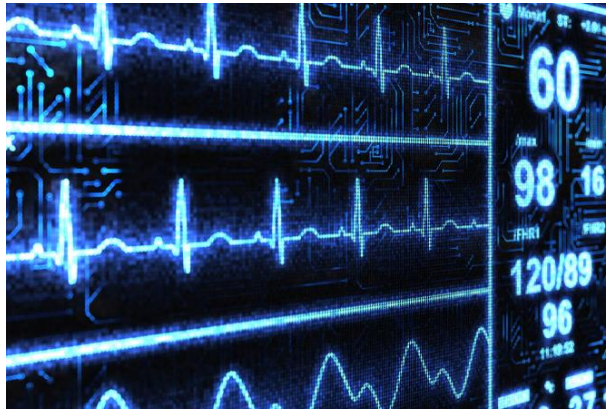
Most signal processing applications cannot count on many AI resources

Vibration analysis



Seismic analysis

Predictive maintenance



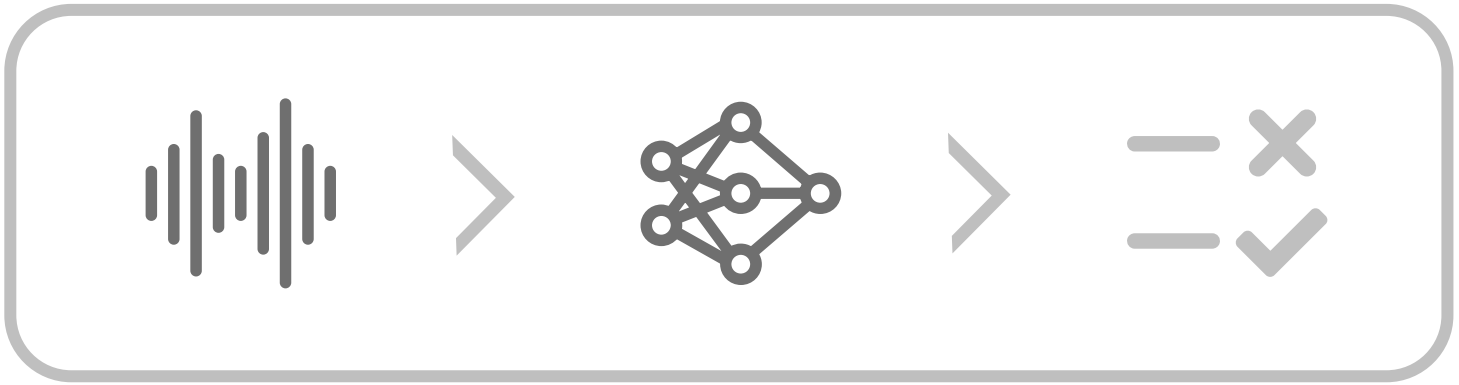
Digital health

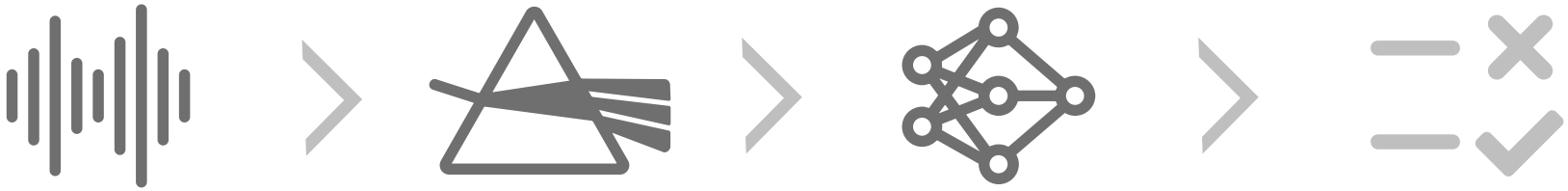
Machine health

...

Which of these best describes your AI-related challenges

- ○ Model Complexity
- ○ Data Complexity
- ○ AI Expertise

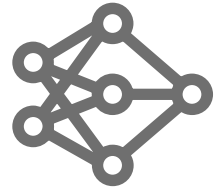




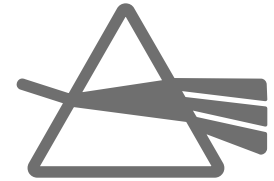
Data-Centric AI in Signal Processing Applications

Agenda – Three Practical Engineering Approaches

1. Transfer learning with pre-trained AI models



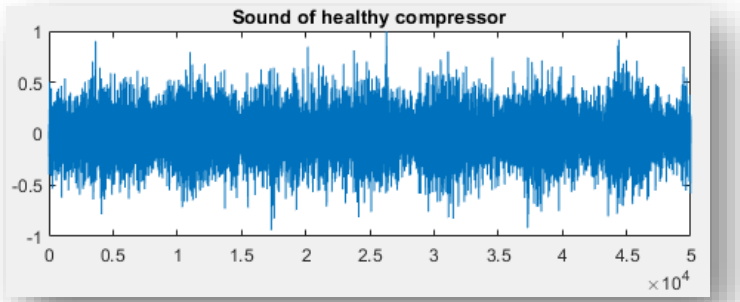
2. Feature extraction with simpler and smaller AI models

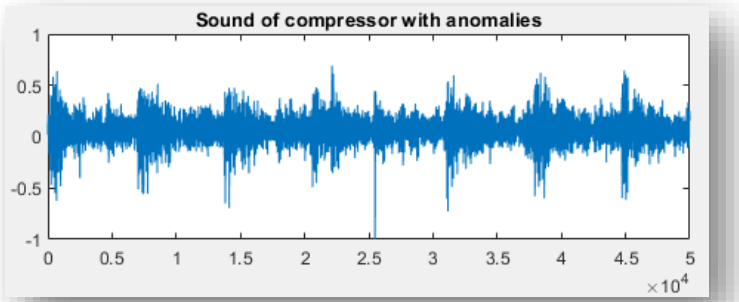


3. Better signal datasets, real or simulated

How can I apply transfer learning to detecting faults in an air compressors based on their noise

- Have dataset with labeled sound recordings
- One “healthy” class
- 7 different classes of faults
- 1800.wav files, 225 per class

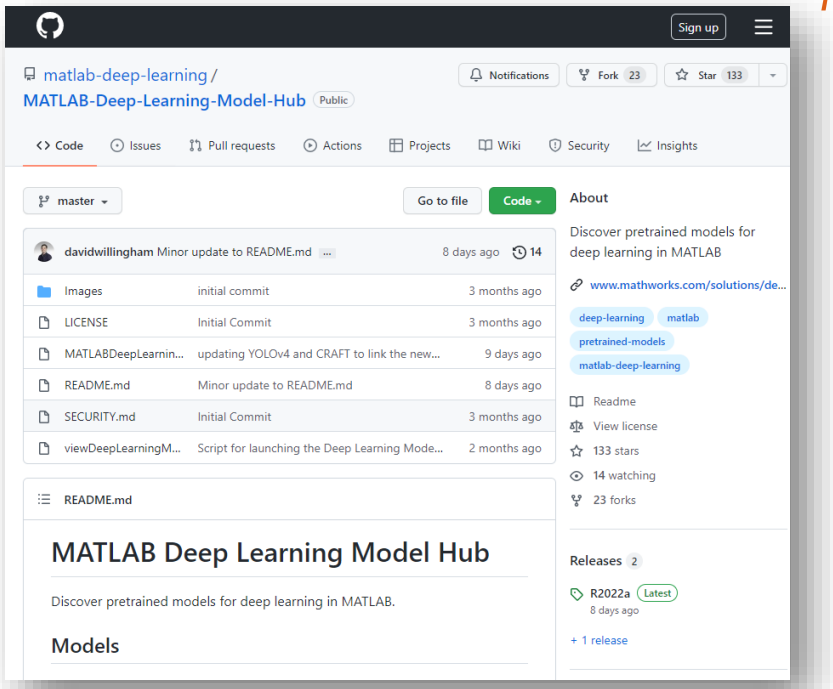




[Example: Transfer Learning with Pretrained Audio Networks in Deep Network Designer]

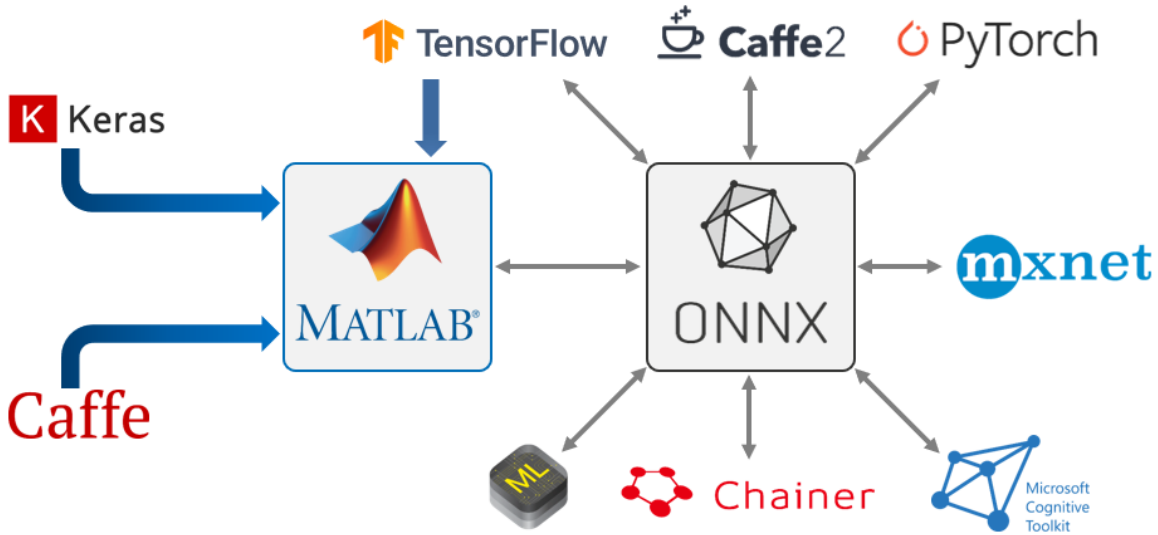
Finding a pre-trained deep learning network for Transfer Learning

- Find one directly in MATLAB



https://github.com/matlab-deep-learning/MATLAB-Deep-Learning-Model-Hub

- Import it from a known non-MATLAB repository



14:00 - 14:40 (KST)

딥러닝을 위해 MATLAB과 TensorFlow/PyTorch 함께 사용하기

MATLAB®과 Simulink®는 TensorFlow™/PyTorch와 같은 딥러닝 프레임워크와 함께 사용함으로써 인공지능 모델을 설계하고 학습하기 위한 향상된 기능을 제공합니다. 상호 운용성을 통해 MATLAB 개발 환경을 최대한 활용하고 이를 오픈 소스 커뮤니티에서 개발한... 더보기

인공지능

MATLAB R2022a

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

Level A Update Cleanup Search Documentation Gabriele

New Open Save Compare Print Export FILE | Go To Find Bookmark NAVIGATE | Text Normal B I U M TEXT | Code Control Task Refactor CODE | Run Section Section Break Run and Advance Run to End SECTION | Run Step Stop RUN

C: ▸ Users ▸ gbunkhei ▸ OneDrive - MathWorks ▸ Documents ▸ MATLAB ▸ Examples ▸ R2022a ▸ deeplearning_shared ▸ TransferLearningWithAudioNetworkInDeepNetworkDesignerExample

Current Folder | Command History

Name	Size	Date Modified
Live Script		
Transf...	425 KB	07/04/2022 ...

Details

Workspace

Name	Value
ads	*1x1 audioData*
adsTest	*1x1 audioData*
adsTrain	*1x1 audioData*
adsValidation	*1x1 audioData*
datasetLocation	'C:\Users\gbun
downloadFolder	'C:\Users\gbun
tdsTrain	*1x1 Transform*
tdsValidation	*1x1 Transform*
url	'https://www.m

Live Editor - C:\Users\gbunkhei\OneDrive - MathWorks\Documents\MATLAB\Examples\R2022a\deeplearning_shared\TransferLearningWithAudioNetworkInDeepNetworkDesignerExample\TransferLearningWithAudioNetworkInDeepNetw...

TransferLearningWithAudioNetworkInDeepNetworkDesignerExample.mlx

Transfer Learning with Pretrained Audio Networks in Deep Network Designer

This example shows how to interactively fine-tune a pretrained network to classify new audio signals using Deep Network Designer.

Transfer learning is commonly used in deep learning applications. You can take a pretrained network and use it as a starting point to learn a new task. Fine-tuning a network with transfer learning is usually much faster and easier than training a network with randomly initialized weights from scratch. You can quickly transfer learned features to a new task using a smaller number of training signals.

This example retrains YAMNet, a pretrained convolutional neural network, to classify a new set of audio signals. This example requires Audio Toolbox™ and Deep Learning Toolbox™.

Load Data

Download and unzip the air compressor data set [1]. This data set consists of recordings from air compressors in a healthy state or one of 7 faulty states.

```
url = 'https://www.mathworks.com/supportfiles/audio/AirCompressorDataset/AirCompressorDataset.zip';
downloadFolder = fullfile(tempdir,'aircompressordataset');
datasetLocation = tempdir;

if ~exist(fullfile(tempdir,'AirCompressorDataSet'),'dir')
    loc = websave(downloadFolder,url);
    unzip(loc,fullfile(tempdir,'AirCompressorDataSet'))
end
```

Create an `audioDatastore` object to manage the data and split it into training, validation, and test sets.

Command Window

fx >>

Zoom: 100% UTF-8 LF script

Transfer Learning – Handouts

Choosing the right model for transfer learning

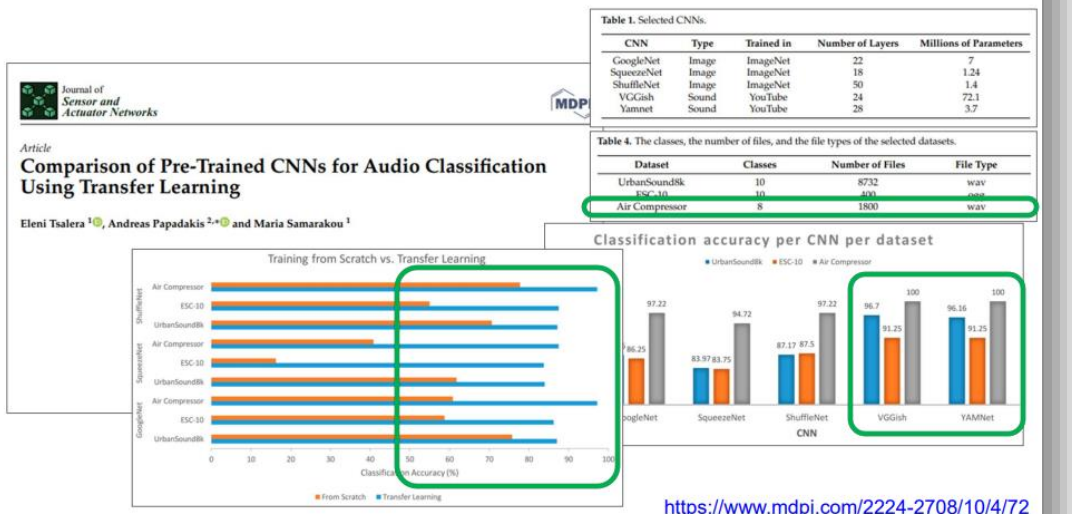


Table 1. Selected CNNs.

CNN	Type	Trained in	Number of Layers	Millions of Parameters
GoogleNet	Image	ImageNet	22	7
SqueezeNet	Image	ImageNet	18	1.24
ShuffleNet	Image	ImageNet	50	1.4
VGGish	Sound	YouTube	24	72.1
Yamnet	Sound	YouTube	28	3.7

Table 4. The classes, the number of files, and the file types of the selected datasets.

Dataset	Classes	Number of Files	File Type
UrbanSound8k	10	8732	wav
ESC-10	10	400	ogg
Air Compressor	8	1800	wav

Journal of Sensor and Actuator Networks

Article

Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning

Eleni Tsalera [1], Andreas Papadakis [2,*] and Maria Samarakou [1]

Training from Scratch vs. Transfer Learning

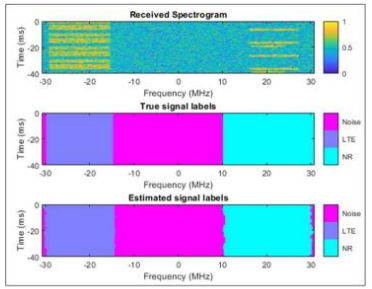
Classification accuracy per CNN per dataset

https://www.mdpi.com/2224-2708/10/4/72

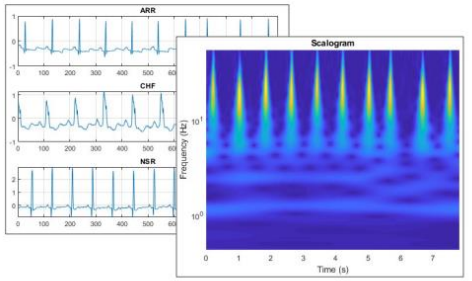
[Download @ Journal of Sensors and Actuator Networks]

Transfer Learning with models pre-trained on different types of data

- Spectrum Sensing with Deep Learning to Identify 5G and LTE Signals
- Network: ResNet-50 (Image segmentation)
- Input: 256-by-256-by-3 images
- Features: `spectrogram` of baseband waveforms



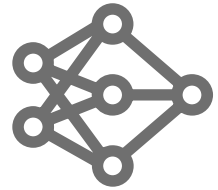
- Classify Time Series Using Wavelet Analysis and Deep Learning
- Network: GoogLeNet (Image object classification)
- Input: 224-by-224-by-3 images
- Features: `cwt` (scalogram) of ECG signals



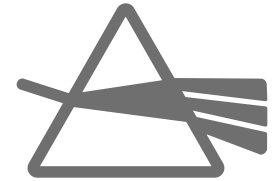
Data-Centric AI in Signal Processing Applications

Agenda – Three Practical Engineering Approaches

1. Transfer learning with pre-trained AI models



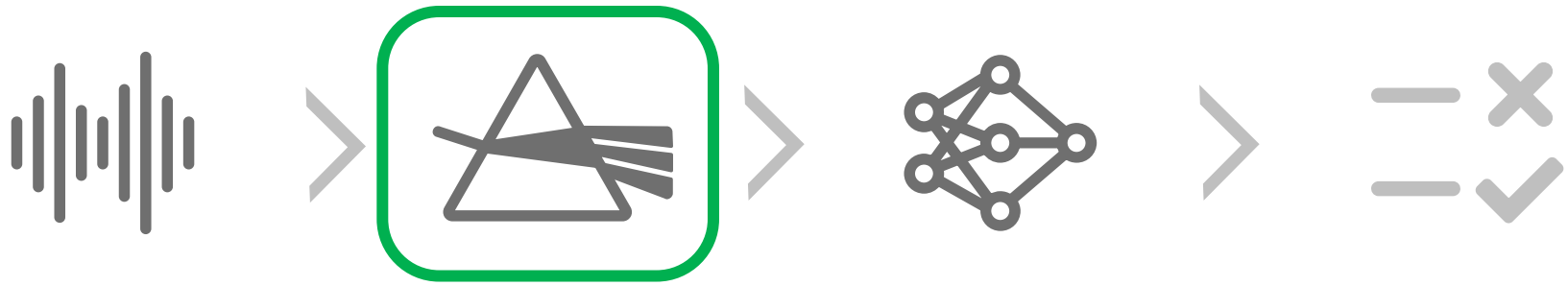
2. Feature extraction with simpler and smaller AI models



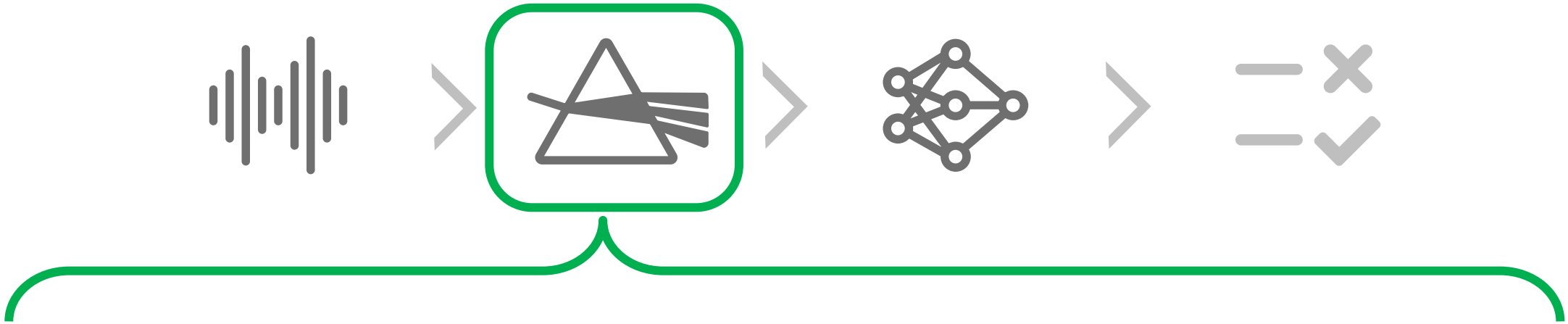
3. Better signal datasets, real or simulated

Extracting features from signals helps reduce complexity

- Smaller networks, faster to learn and run
- Easier to deploy and implement
- Smaller training datasets
- Better return from existing domain expertise



Extracting features from signals helps reduce complexity

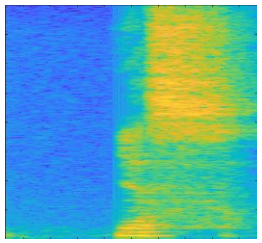


Time-frequency transforms

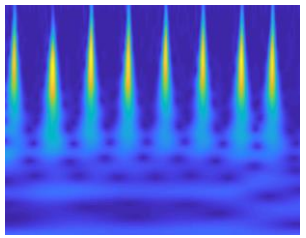
Scalar metrics

Vectors

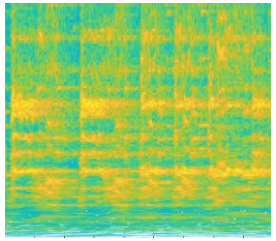
Automated extractors and “embeddings”



spectrogram

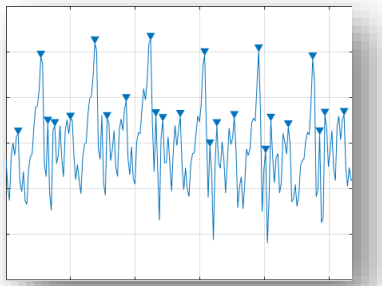


cwt

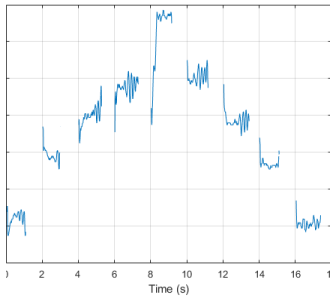


cqt

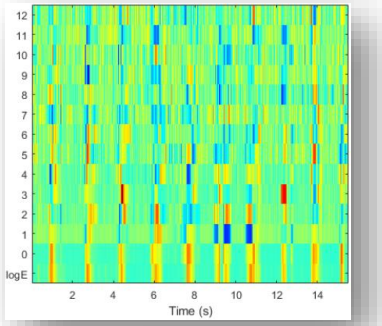
findpeaks

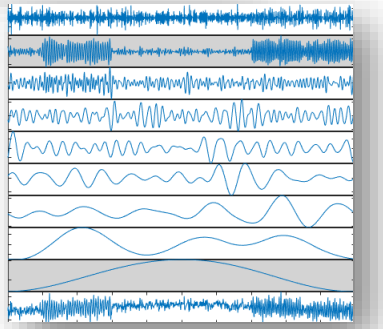


pitch



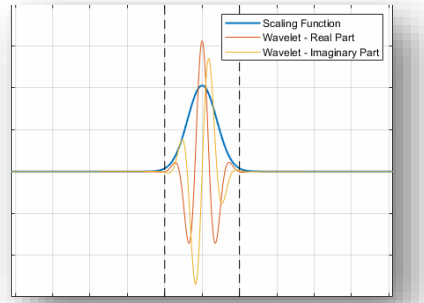
mfcc

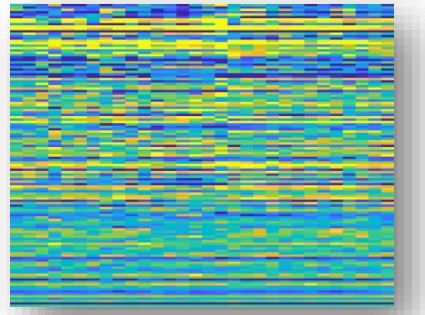




emd, vmd

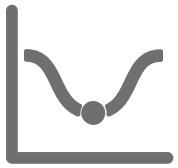
waveletScattering





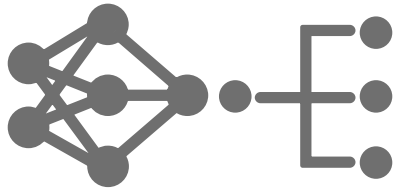
vggishEmbeddings

Time-frequency transforms



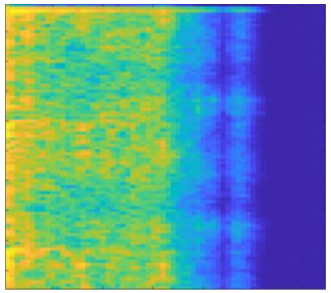
Scalar metrics

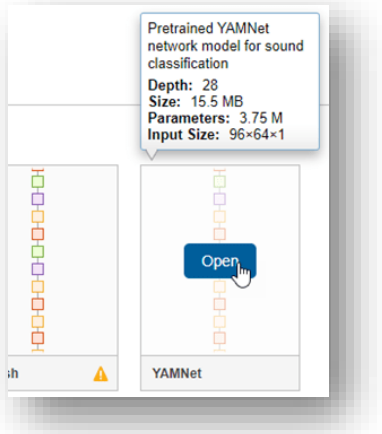
Vectors

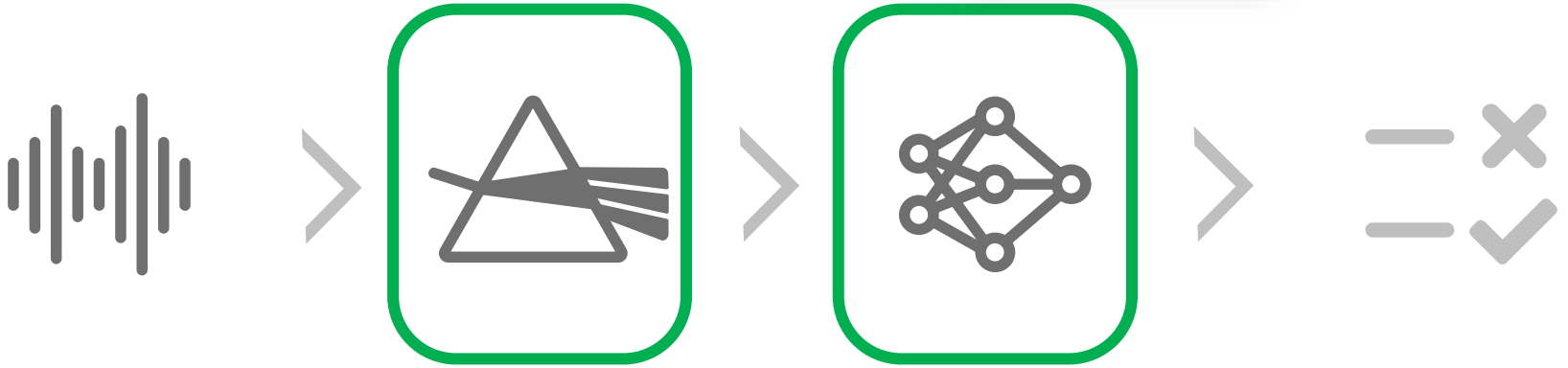


Automated extractors and “embeddings”

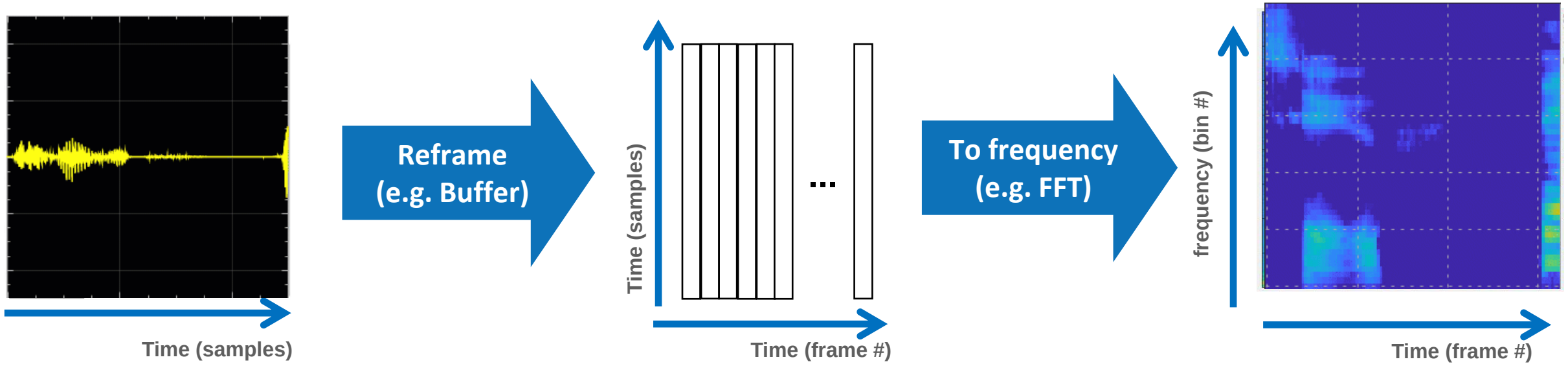
Deep networks most often don’t learn directly from raw signals





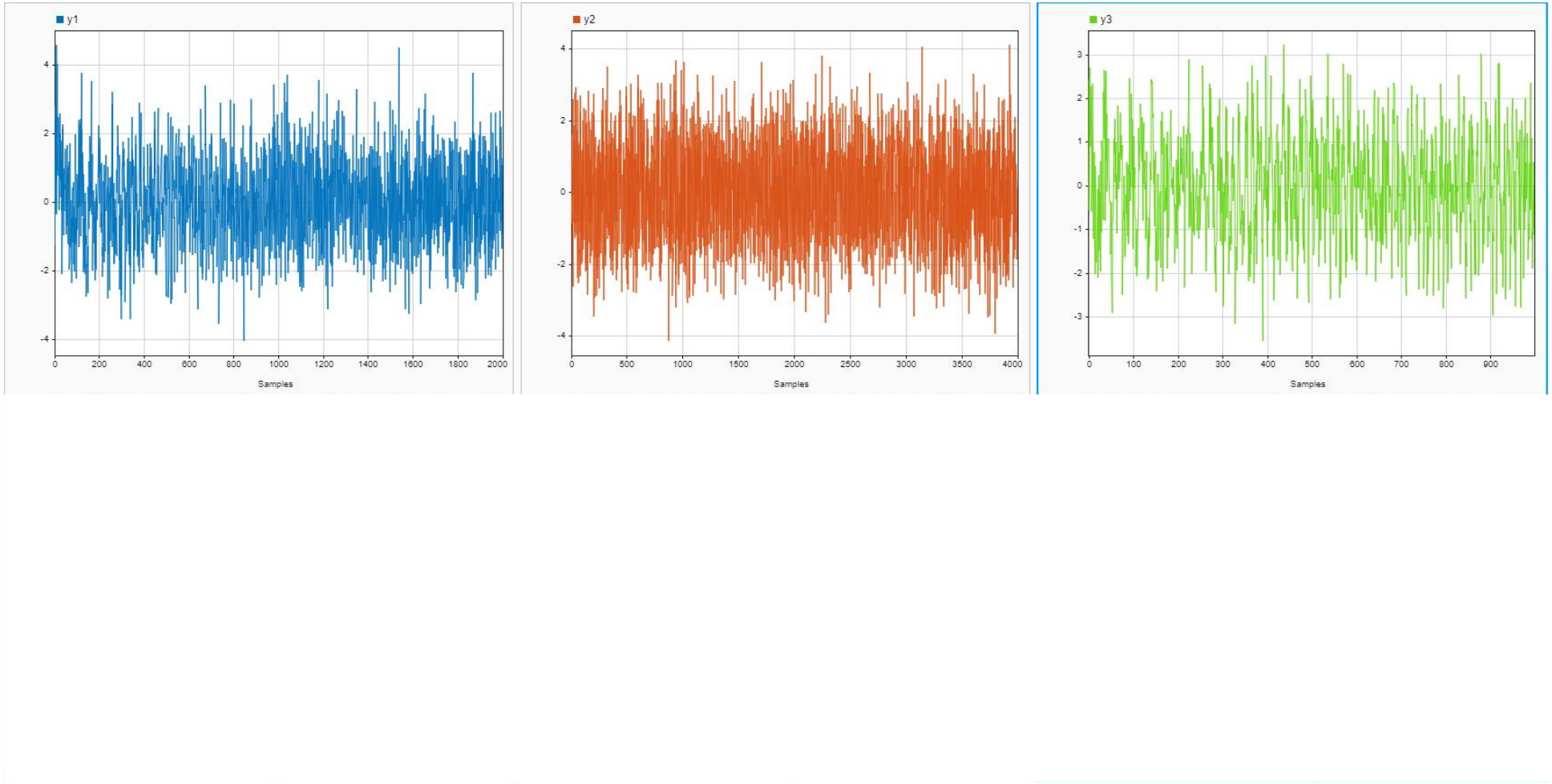


Time-frequency transformations are popular feature extraction methods



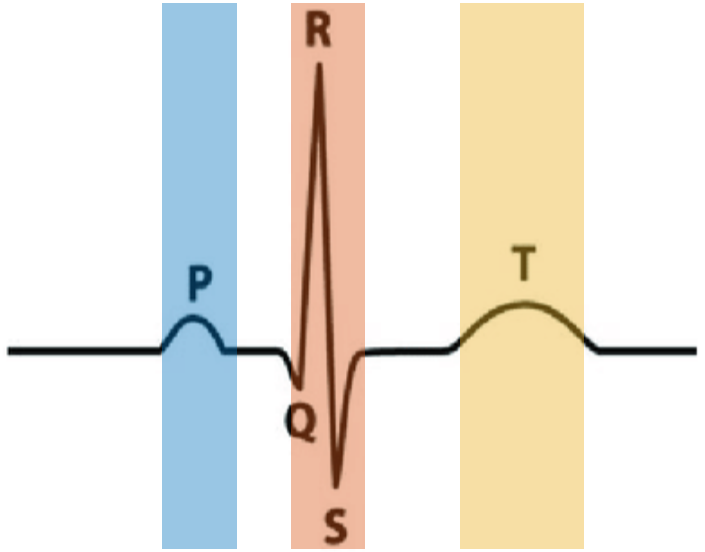
Time-frequency transforms make signal characteristics more evident

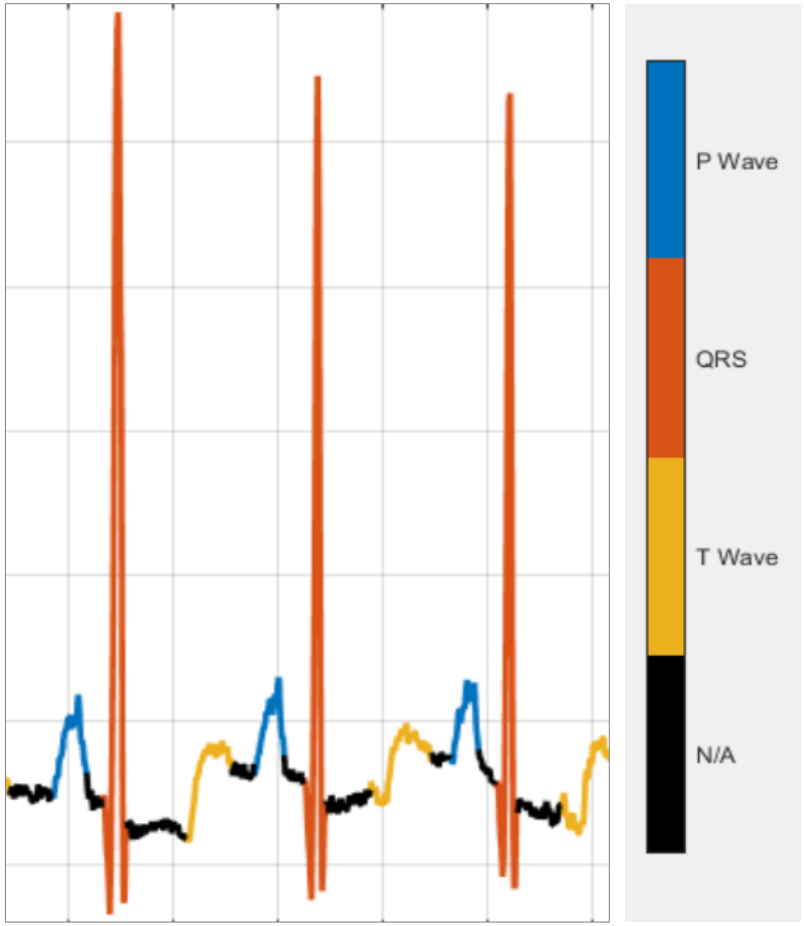
Time-frequency transform



How to use feature extraction to segment ECG signals?

- Have dataset with signals labeled by cardiologists
- 3 types of wave events
- 210 ECG recordings (total ~15 minutes)



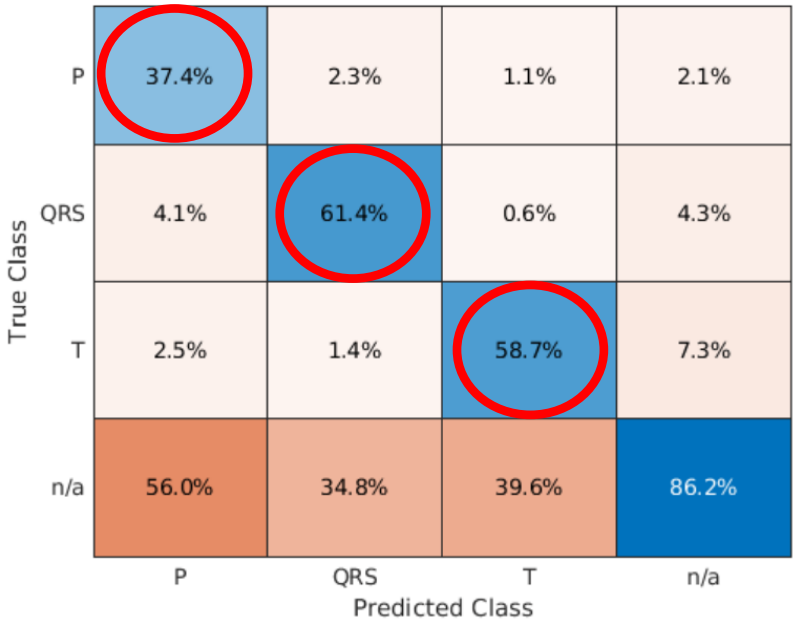


Example: Waveform Segmentation Using Deep Learning

Training simple recurrent neural networks directly with raw signals most often yields unsatisfactory performance

ECG

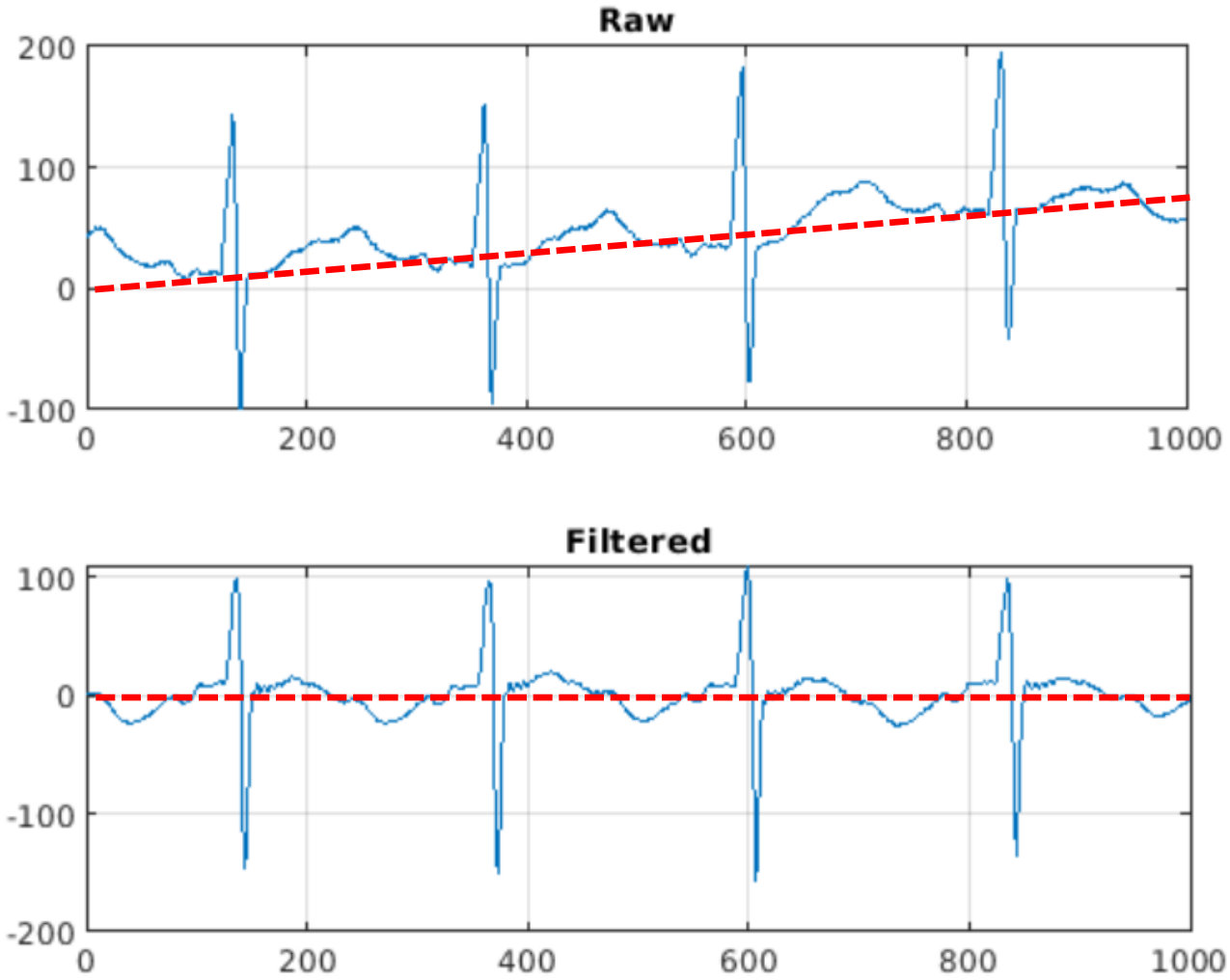
Segmented ECG

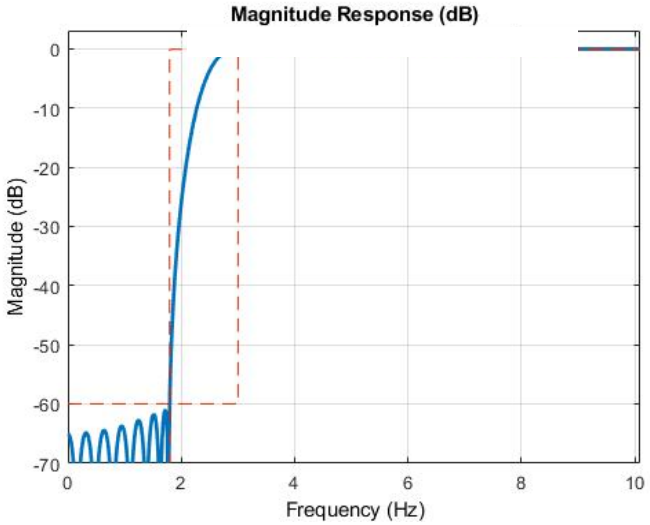



```
layers = [ ...
    sequenceInputLayer(1)
    lstmLayer(200,'OutputMode','sequence')
    fullyConnectedLayer(4)
    softmaxLayer
    classificationLayer];
```

Step 1

Pre-process raw signals to eliminate known patterns that you don't want the AI model to re-learn



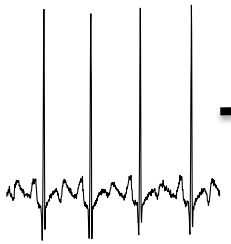


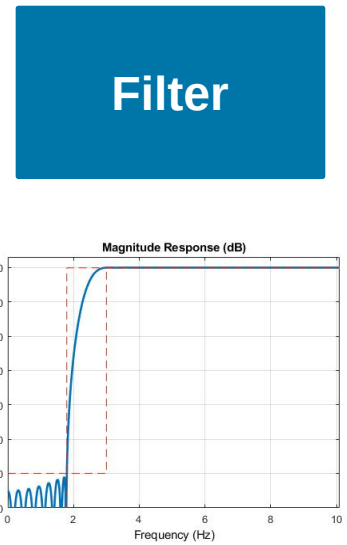
Filter baseline "wander"

Step 1

Pre-process raw signals to eliminate known patterns that you don't want the AI model to re-learn

Raw ECG



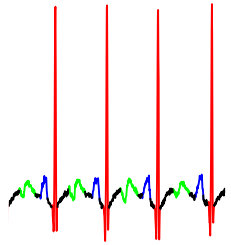


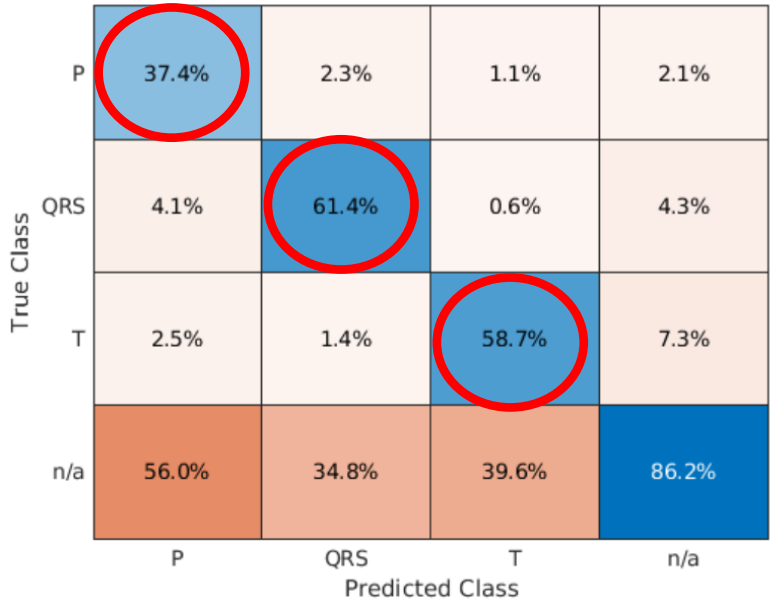
Filter baseline "wander"

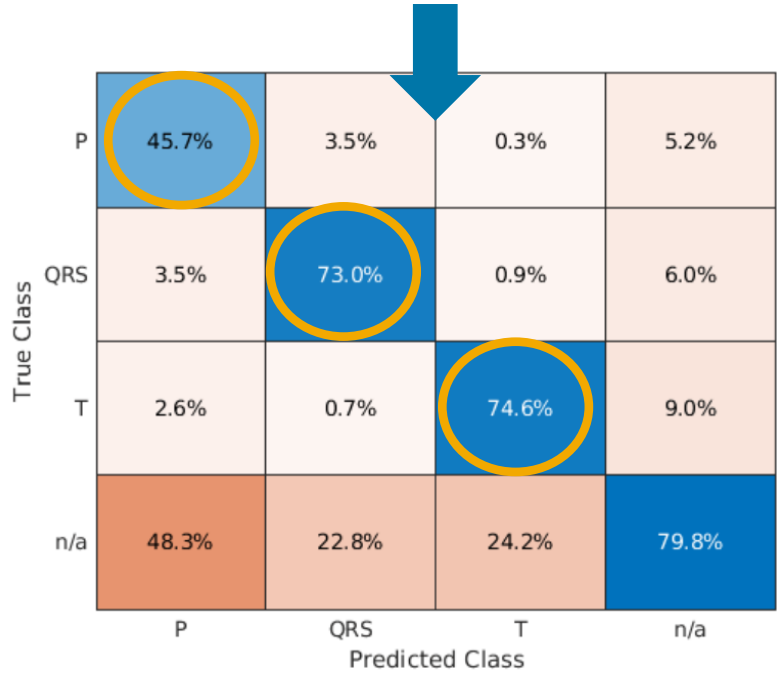
Filter

LSTM

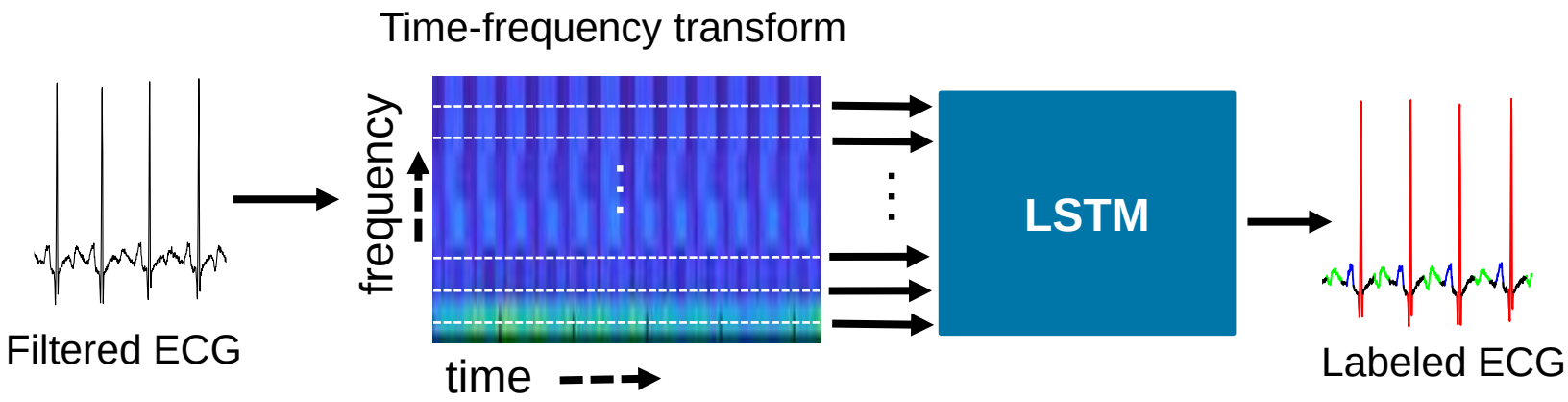
Segmented ECG

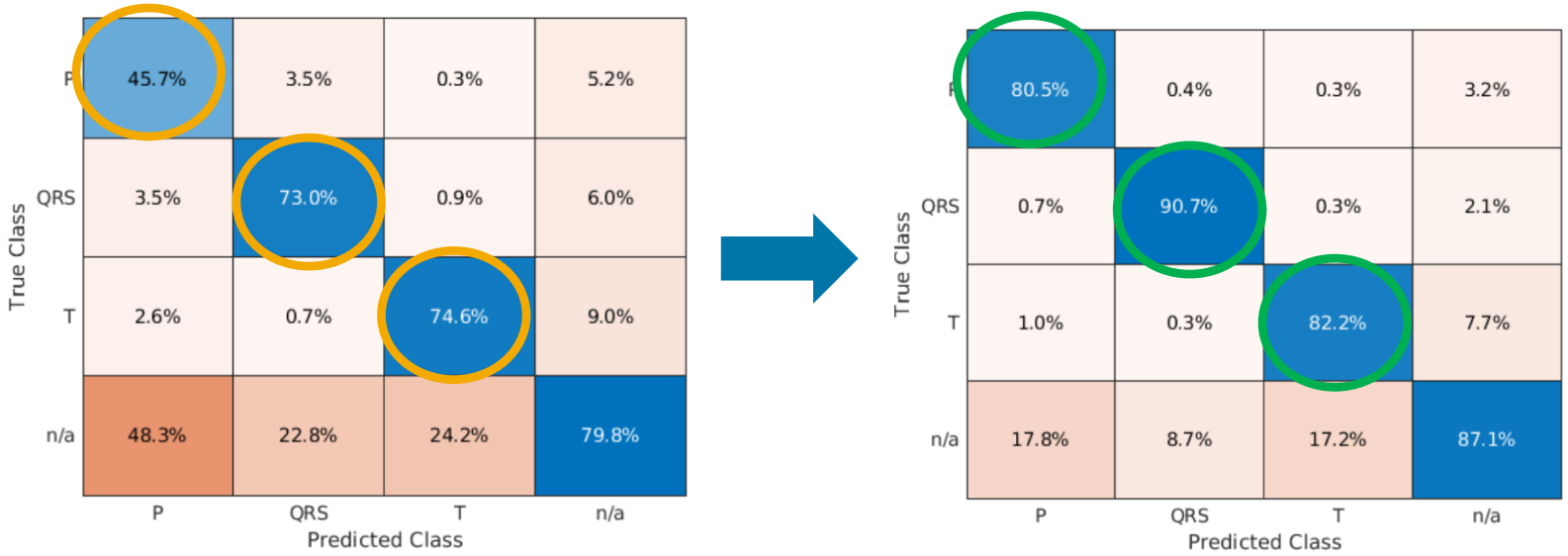






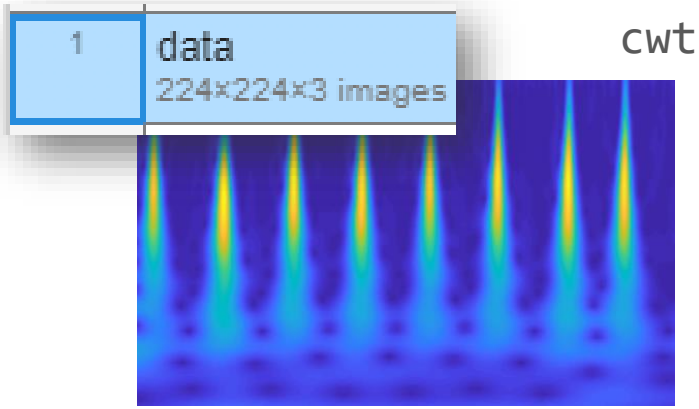
Step 2
Extract features that highlight true variability





Domain experts are best placed to select feature extraction algorithm

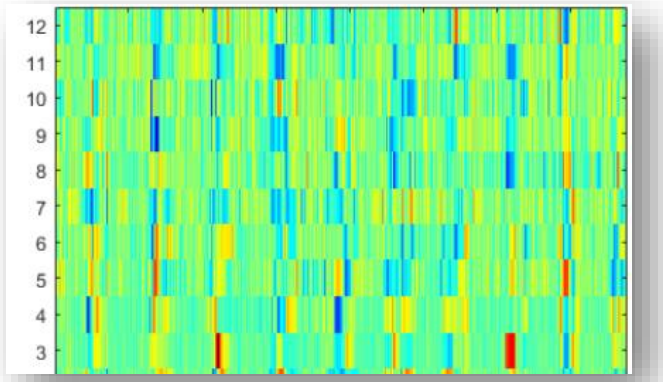
Model size, signal patterns



cwt

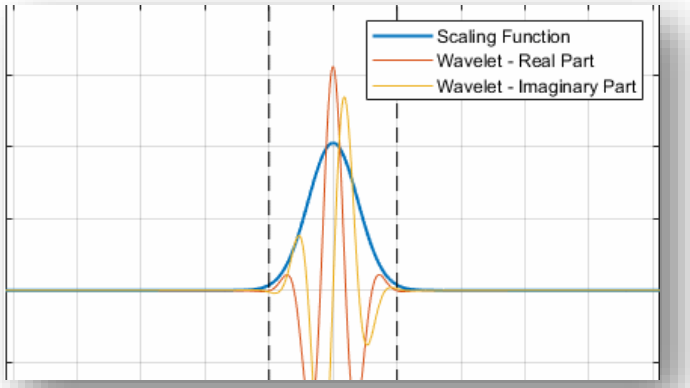
Application and signal type

mfcc



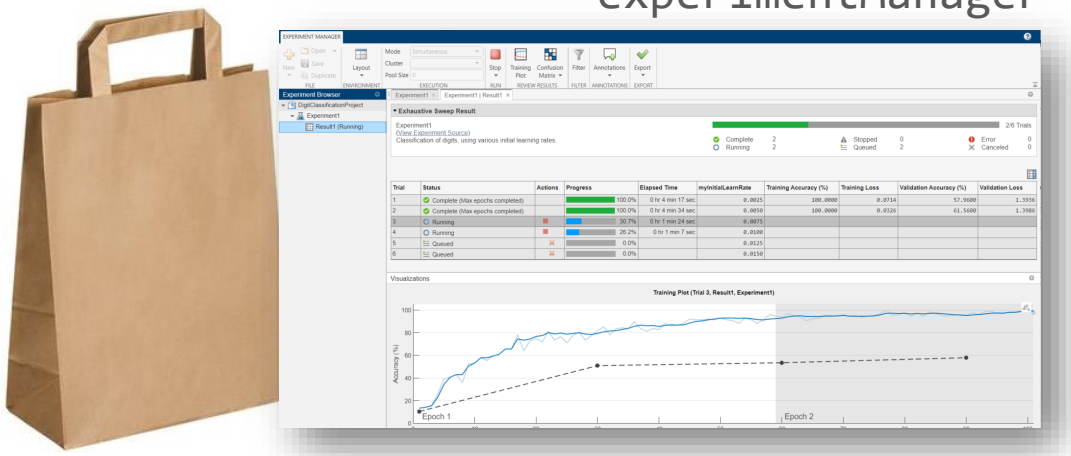
Automated methodology

waveletScattering



Test-based experiments

experimentManager



signalFrequencyFeatureExtractor

Feature Extraction – Handouts

MathWorks Wins Geoscience AI GPU Hackathon

The following post is from Akhilesh Mishra, Mil Shastri and Samvith V. Rao from MathWorks here to talk about their participation and in a Geoscience hackathon. Akhilesh and Mil are Applications Engineers and Samvith is the Industry Marketing Manager supporting the Oil and Gas industry.

Background

SEAM (SEG Advanced Modeling Corp.) is a petroleum geoscience industry body that fosters collaborations among industry, government, and academia to address major Geological challenges. Their latest event was a hackathon (SEAM AI Applied Geoscience GPU Hackathon) that sought to explore the use of AI to improve both qualitative and quantitative interpretation of geophysical images of Earth's interior, and speed up the applications using NVIDIA GPUs.

A total of 7 teams participated from all over the world, including commercial companies (Chevron, Total, Petrobras) and a mix of industry and university students. Each team was assigned a mentor who is an expert geoscientist working for a top oil and gas company.

The Challenge

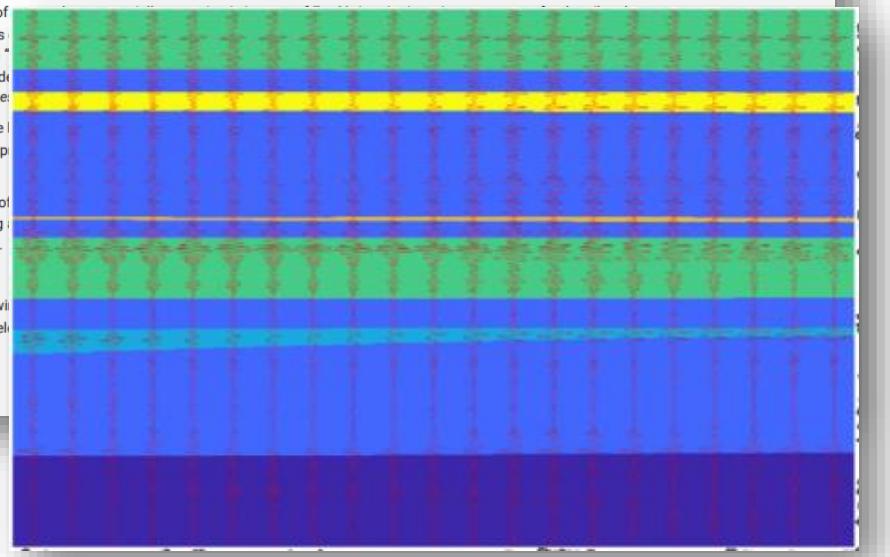
Geologic interpretation of
industry. Seismic images
summarized by the term "
and abandonment of unde
often called *seismic facie*

This process is still done
display. Successful interp
features.

The problem statement o
automatically, producing
up human interpretation.

The Data

We were given the followi
public and has been label



[MathWorks Deep Learning Blog Post]

Daihatsu Uses AI to Classify Engine Sounds

Challenge

Develop an AI solution that can judge the level of engine knocking sound, which only skilled workers could judge

Solution

Create classification models and easy-to-use interface with MATLAB, making it possible to examine features multiple times

Key Outcomes

- Performed knocking sound analysis with the same accuracy as skilled workers
- Increased AI expertise through MATLAB training
- Promoted visualization of AI and increased awareness of AI

Daihatsu used AI to identify knocking sounds from its engines.

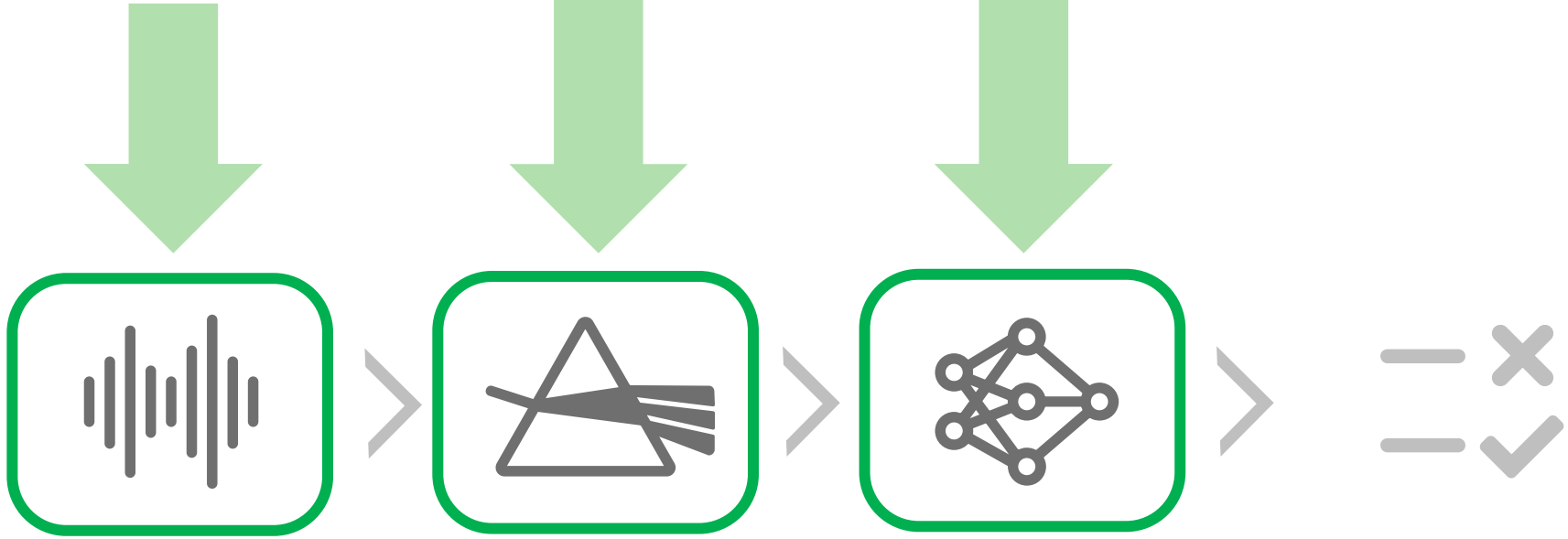
"Although we tried other programming languages, it was hard to implement. We decided to use MATLAB, which allows us to easily import the necessary data by dragging and dropping, and we could easily see the result by ourselves."
- Takuya Kumagae, Daihatsu Motor Co., Ltd.

Link to case study

[Daihatsu User Story]

Requiring smaller datasets multiplies the impact of data engineering

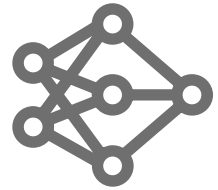
- Using **transfer learning**…
- …or **feature extraction** with simple models…
- …leads to requiring **much smaller labeled datasets** for model training



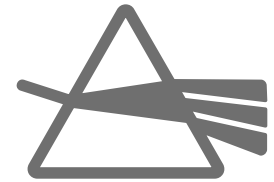
Data-Centric AI in Signal Processing Applications

Agenda – Three Practical Engineering Approaches

1. Transfer learning with pre-trained AI models



2. Feature extraction with simpler and smaller AI models



3. Better signal datasets, real or simulated

Strategy for preparing Good Data

Cover of important cases
Good coverage of inputs x

Sized appropriately

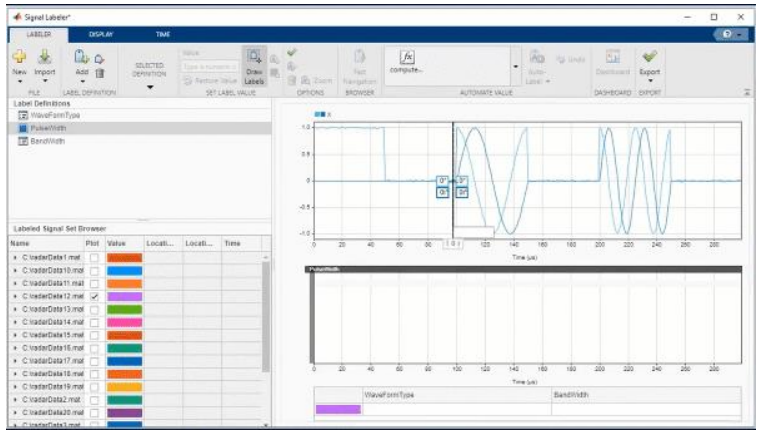
Defined consistently
Definition of labels y is unambiguous

- Change inputs x
  - Use tools to improve the data quality
  - Data augmentation, data generation or data collection

- Change labels y
  - Give more consistent definition for labels if they were found to be ambiguous

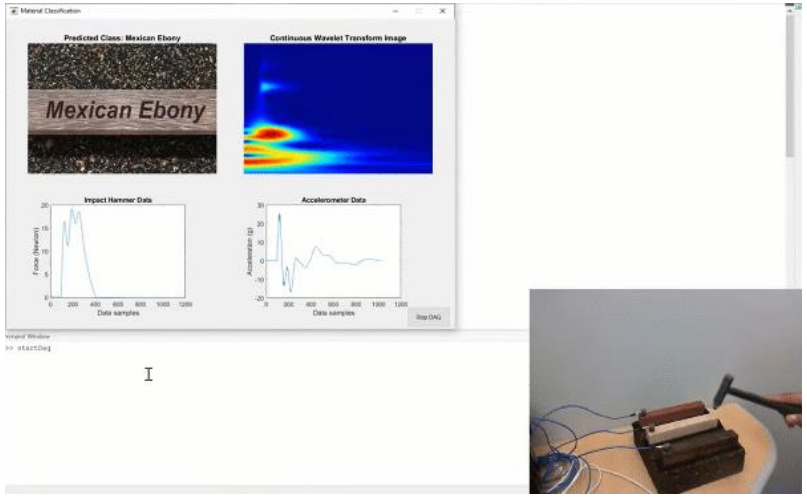
How can I enhance the quality of my training signal data?

Define accurate data labels



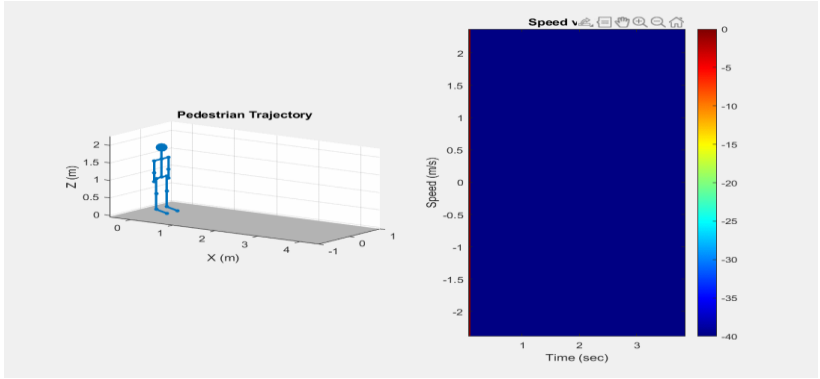
`signalLabeler`

Record and label new data via Apps and Hardware

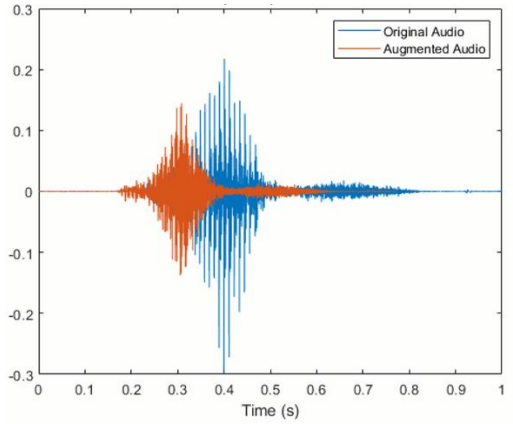


`App Designer`

Synthesize data via simulation



Augment data via signal processing

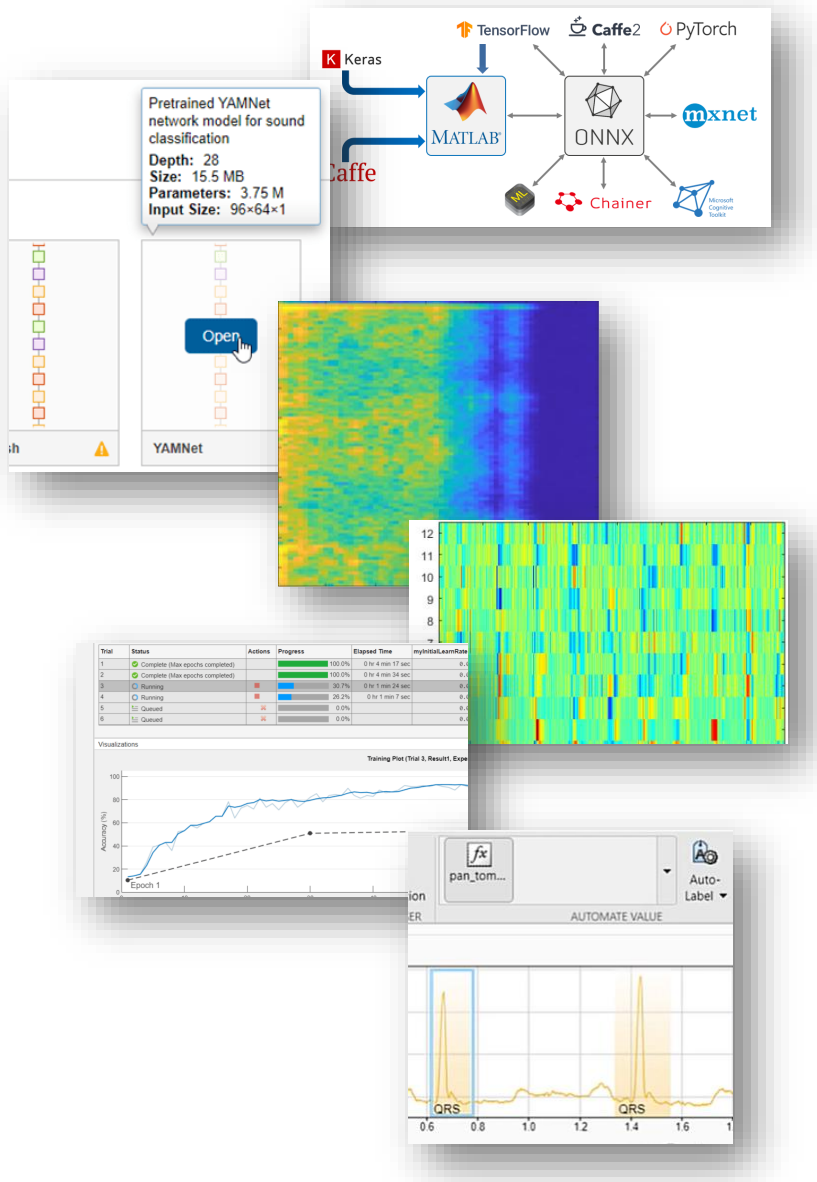


`audioDataAugmenter`

Data-Centric AI accelerate AI adoption by domain experts

The “unbiggen AI” effect

- ~~Model Complexity~~ » MATLAB
- ~~Data Complexity~~ » Signal Processing
- ~~AI Expertise~~ » Domain Expertise



AI vs. Signal Processing

MATLAB EXPO

Thank you