【信号処理技術で約40年の実績】MATLABで信号処理
～机上シミュレーションから実装までを実例でご紹介～

Mathworks Japan
アプリケーションエンジニアリング部

全体のAgenda

1. **MATLABで信号処理**
   - なぜ、信号処理が重要なのか？
   - 信号処理ツールとしてMATLABが選ばれてきた理由
   - 信号処理の各種例題のご紹介

2. **MATLABで簡単Raspberry Pi実装**
   - サポートパッケージで簡単セットアップ + 外部ハードとの接続
   - まずはLチカ！+ MATLABおよびSimulinkならではのプラスアルファ
   - エッジAIとして利用する際の留意点と構成例

3. **MATLABで楽々FPGA/ASIC実装**
   - MATLABおよびSimulinkを用いたレーダーシステムのシミュレーション
   - サポートパッケージで楽々環境構築！
   - シミュレーションから簡単にHDLを生成、実装、検証

Session1：
MATLABで信号処理

Mathworks Japan
アプリケーションエンジニアリング部
竹本　佳充

Agenda

1. なぜ、信号処理が重要なのか？
2. 信号処理ツールとしてMATLABが選ばれてきた理由
3. 信号処理の各種例題のご紹介

Agenda

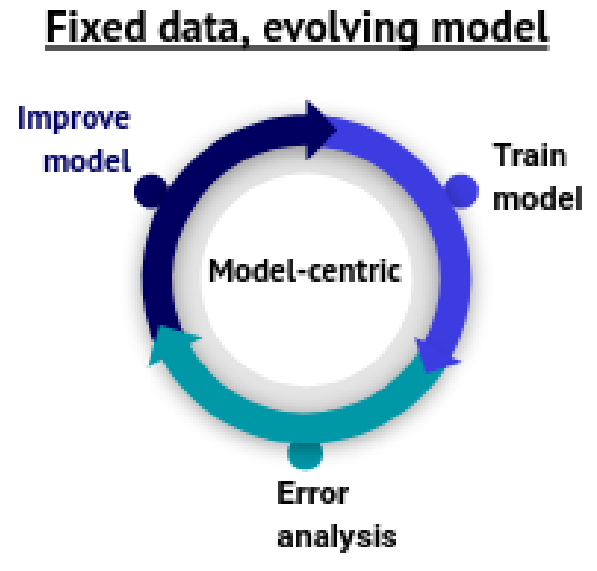
1. なぜ、信号処理が重要なのか？
2. 信号処理ツールとしてMATLABが選ばれてきた理由
3. 信号処理の各種例題のご紹介

データセントリックAI

ENTERPRISEAI
ADVANCED COMPUTING IN THE AGE OF AI

AI Modeling Reinvented: It's Time to Shift to Better Data, Rather Than Just Building Better Models

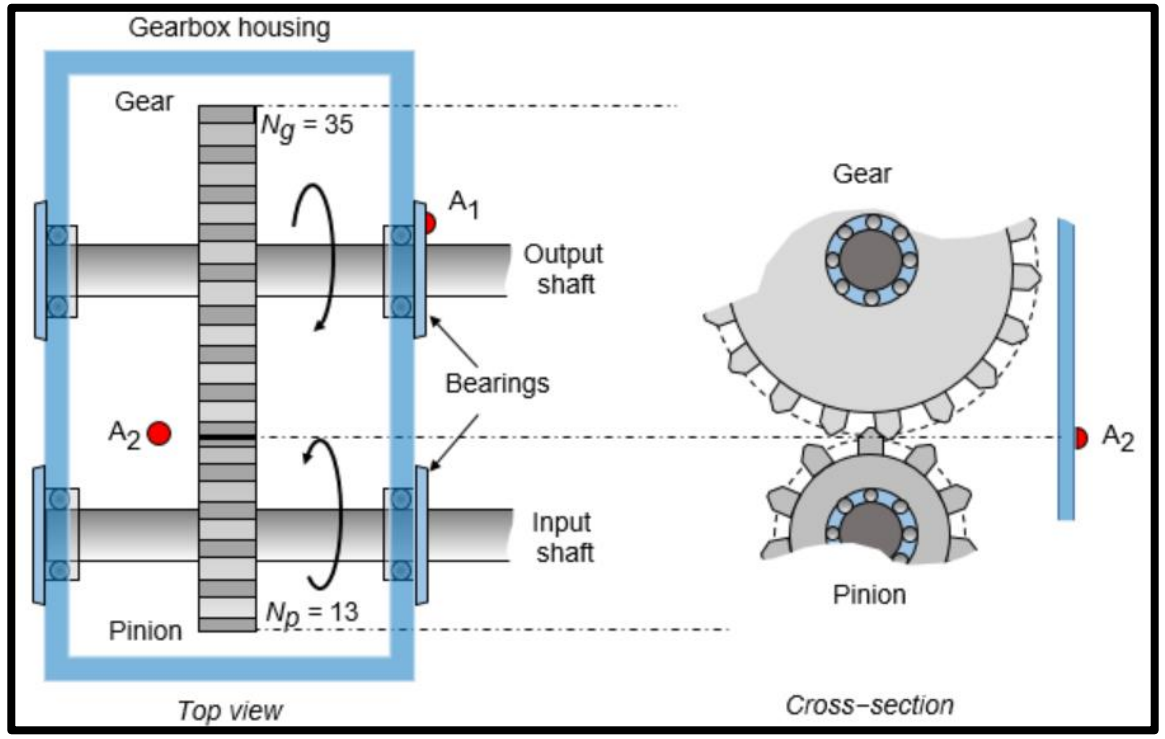
October 8, 2021 by Todd R. Weiss



"In many industries where giant data sets simply don't exist, I think the focus has to shift from big data to good data. Having 50 thoughtfully engineered examples can be sufficient to explain to the neural network what you want it to learn."

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

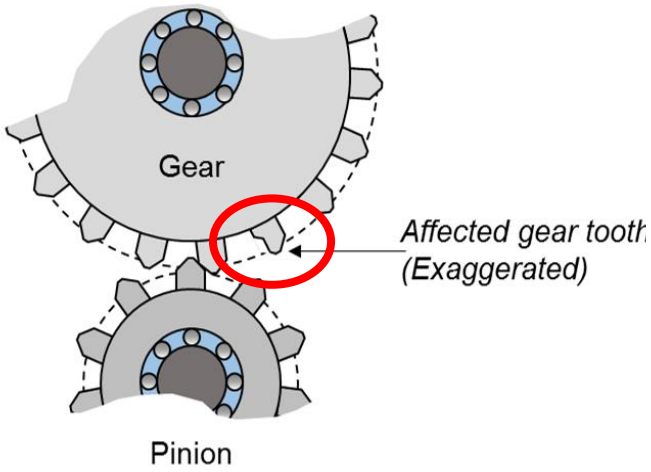
ギアの異常判別例：官能評価から自動判定へ



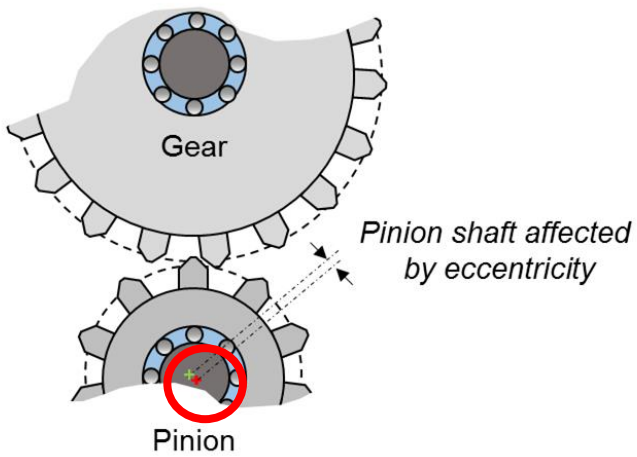
正常音

官能評価から自動判定へ
⇒波形を眺めてみる

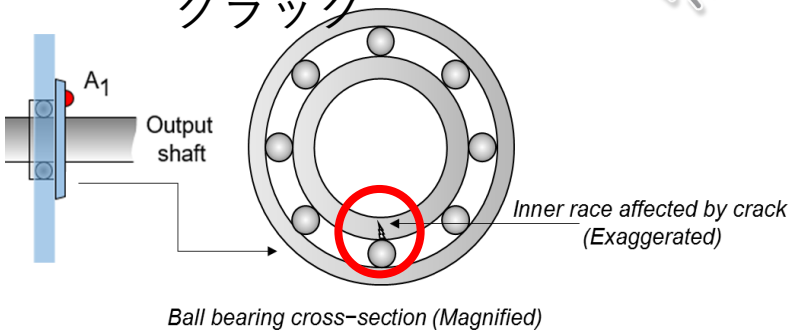
①大歯の欠け



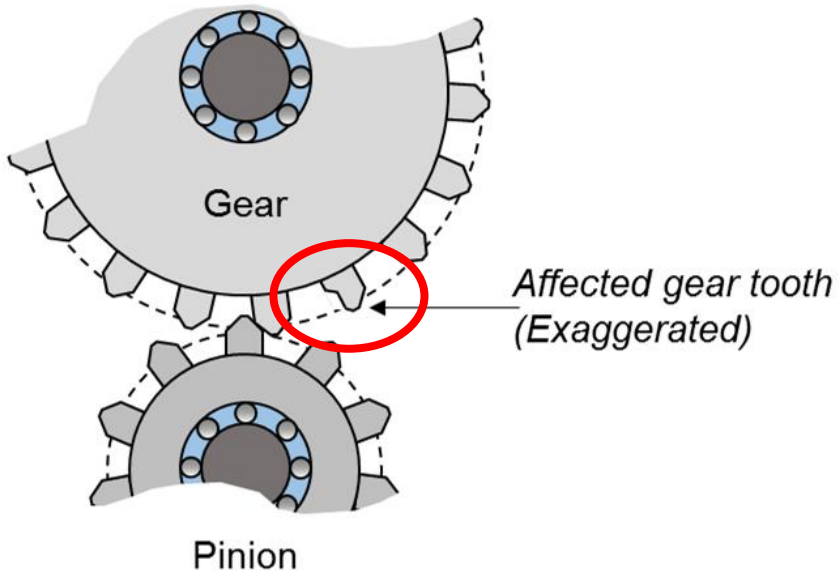
②小歯の偏芯



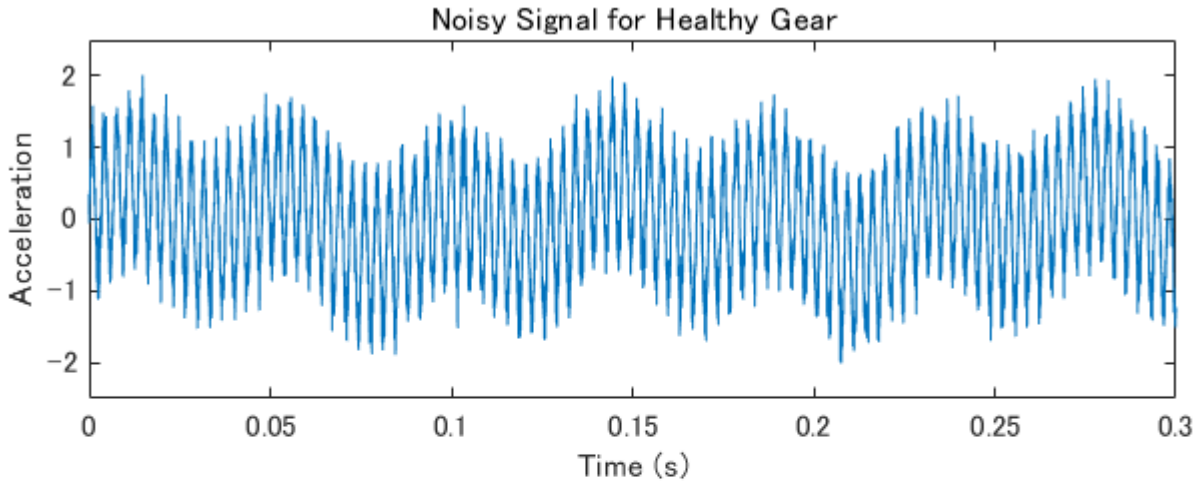
③ベアリングの
クラック



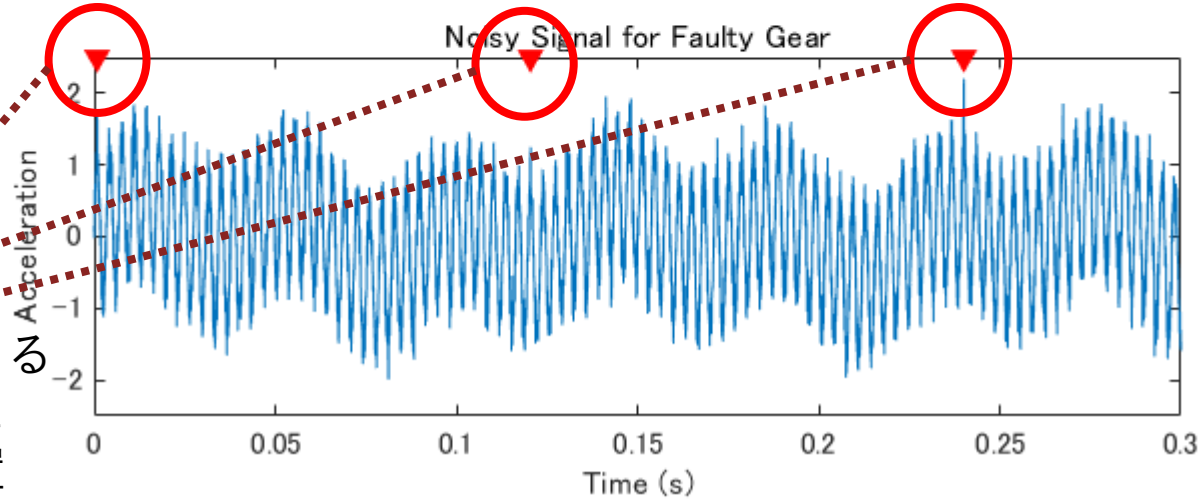
まずは時間波形で波形を観測



1/fMesh (0.25 ミリ秒) の 8% の時間で
2 kHz の振動信号が発生すると仮定



破損なし



破損あり

歯の破損による
高周波成分に
相当する位置

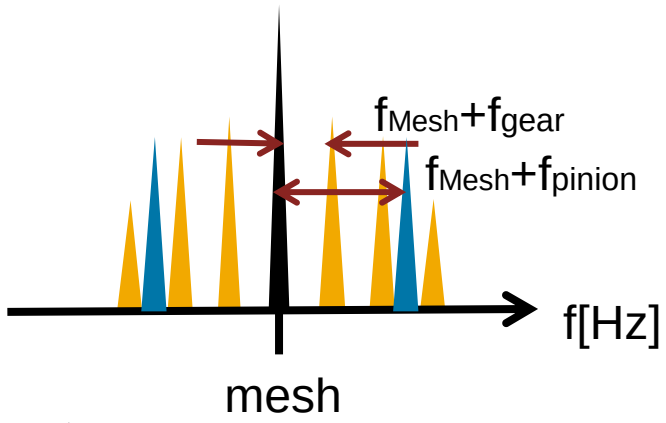
時間軸では判別が困難
⇒周波数軸で眺めてみる

信号処理の活用：周波数領域から眺める

破損に起因する周波数成分

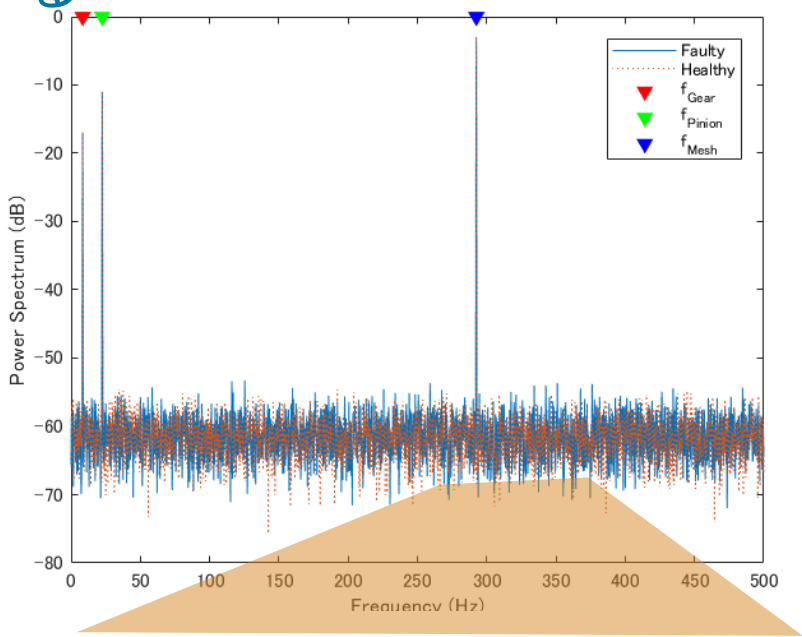
$$f_{\text{sideband,Pinion}} = f_{\text{Mesh}} \pm m \times f_{\text{Pinion}} \qquad \forall m \in \{1, 2, 3, ..\}$$

$$f_{\text{sideband,Gear}} = f_{\text{Mesh}} \pm m \times f_{\text{Gear}} \qquad \forall m \in \{1, 2, 3, ..\}$$

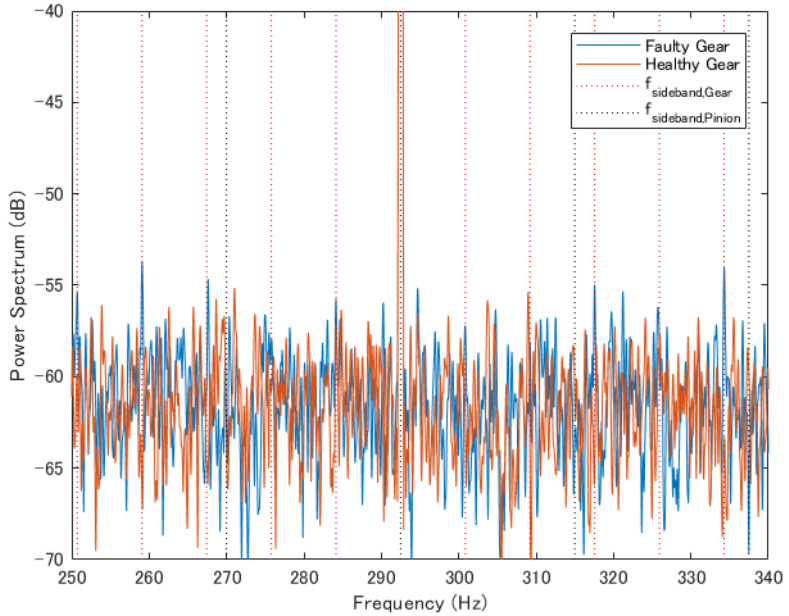



```
[Spect,f] = pspectrum(...
[vFaultNoisy' vNoFaultNoisy'],...
Fs,'FrequencyResolution',0.2, ...
'FrequencyLimits',[0 500]);
```

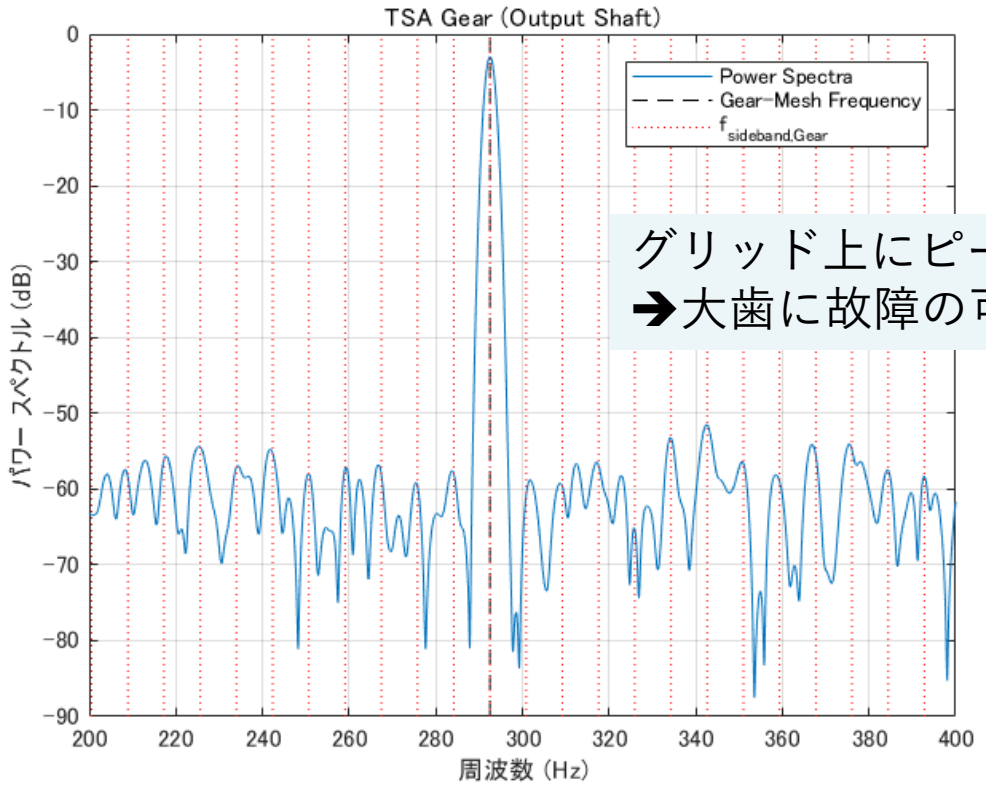
周波数軸でも、雑音に埋もれて観測は困難
⇒TSA(時間同期平均）による前処理



拡大



効果的な前処理により特徴が明らかに



グリッド上にピーク有り
➔大歯に故障の可能性

- 信号処理技術の使いどころ
  - 様々なドメインから眺める
  - 効果的な前処理で信号の本質を知る
  - 特徴的な成分を見極める
- 信号処理技術の効果
  - AIまかせからの脱却
  - 現場の知見を活かせる

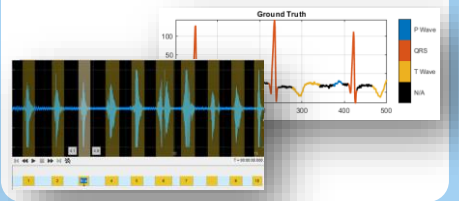
信号処理・AIシステム開発のワークフロー

1. データセットの作成

■ データソース

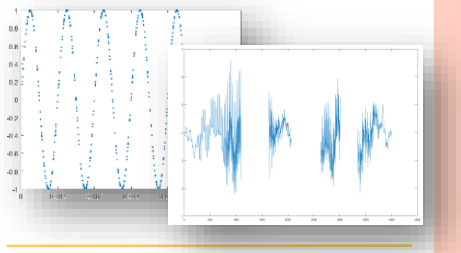
■ シミュレーション
■ オーグメンテーション

■ データラベリング

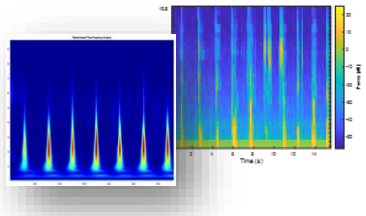


2. 前処理と変換

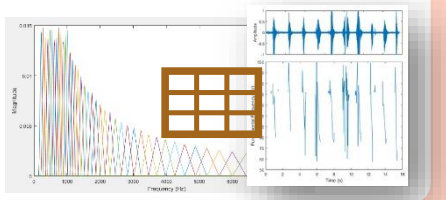
■ 前処理



■ 変換



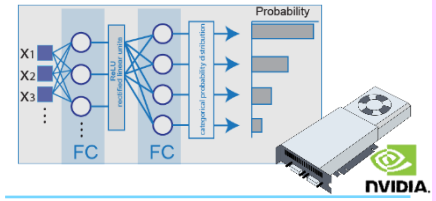
■ 特徴抽出



3. 予測モデルの開発

■ リファレンスモデルの流用
■ オリジナルモデルの作成

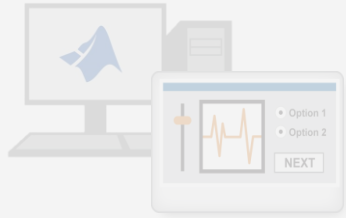
■ GPUによる高速な学習



■ ハイパーパラメータの
解析とチューニング

4. 実システムへの展開

■ デスクトップアプリ



■ エンタープライズシステム

■ エッジ（組み込み）デバイス

Agenda

1. なぜ、信号処理が重要なのか？
2. 信号処理ツールとしてMATLABが選ばれてきた理由
3. 信号処理の各種例題のご紹介

MATLABが選ばれる理由

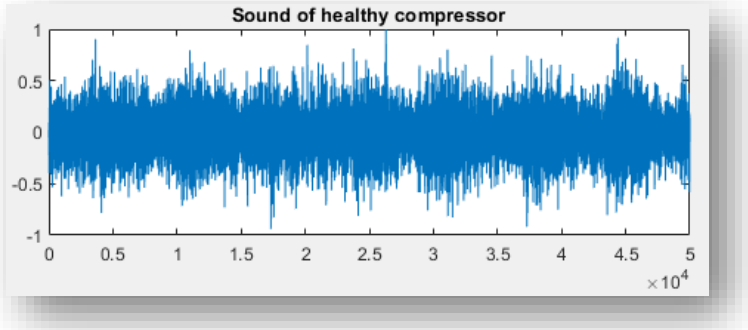
1. データセットの置き換えですぐ試せる

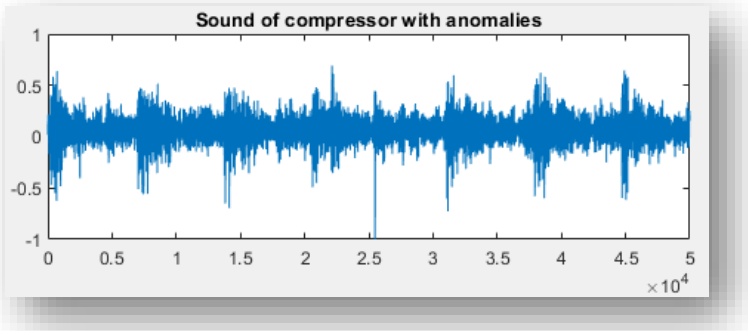
2. 具体的かつ実践的な例題

3. 便利なアプリの活用

エアーコンプレッサーの異音判別例

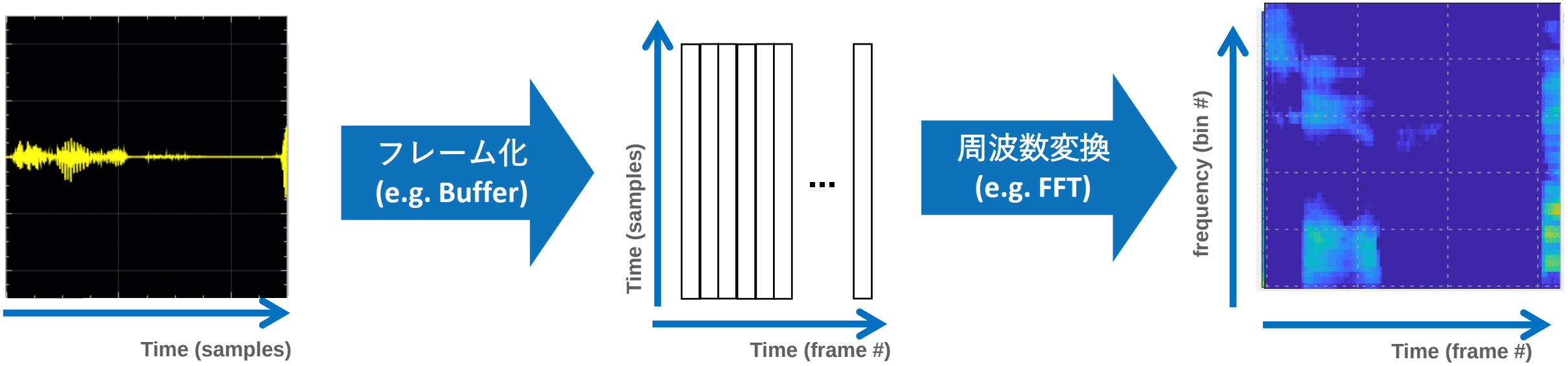
- ラベル付き録音データセット
- 1種類の正常データ
- 7種類の異音データ
  - ベアリング異常
  - バルブ異常
  - ピストン異常
  - その他4種
- 1800点の .wavファイル
- 各データにつき225ファイル(3秒程度)





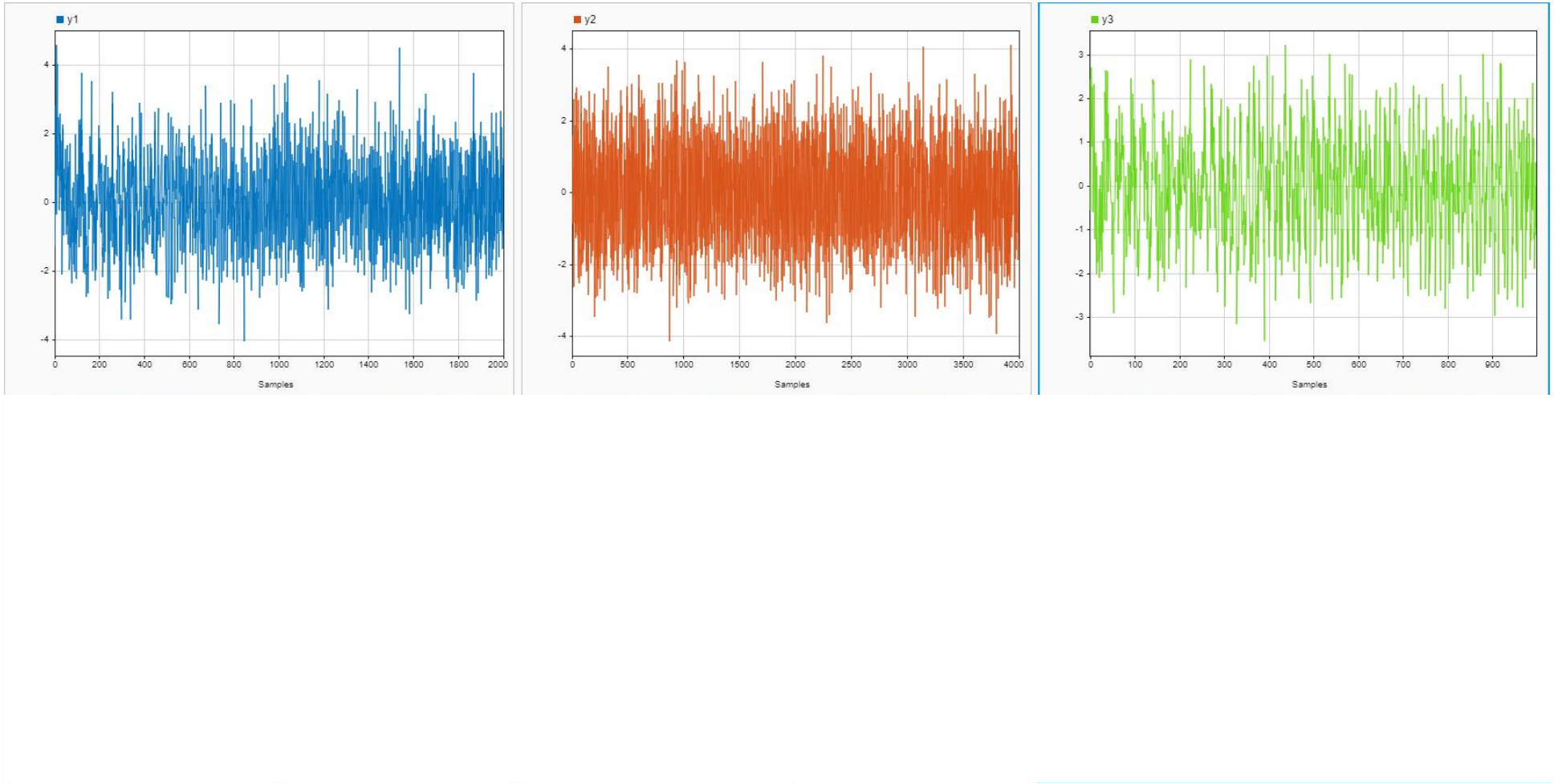
[Example: Transfer Learning with Pretrained Audio Networks in Deep Network Designer]

時間-周波数変換


フレーム化
(e.g. Buffer)
周波数変換
(e.g. FFT)
Time (samples)
Time (samples)
Time (frame #)
frequency (bin #)
Time (frame #)

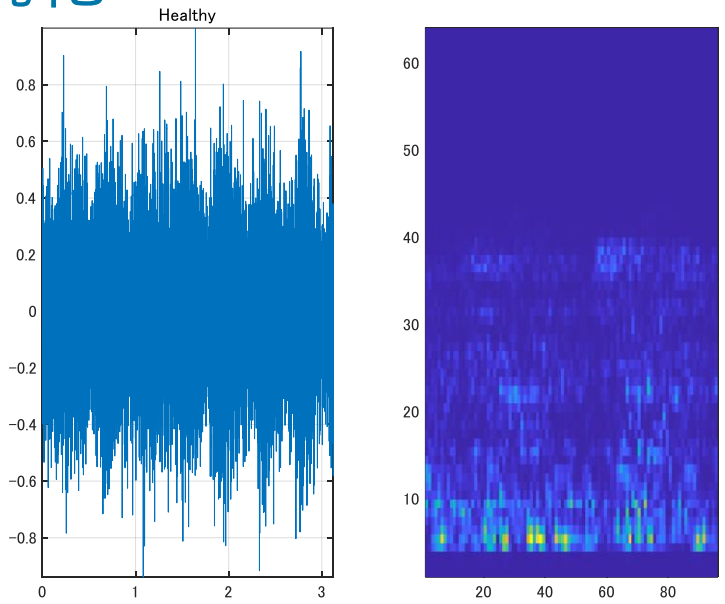
時間-周波数変換は、信号の特性を際立たせる

時間-周波数変換

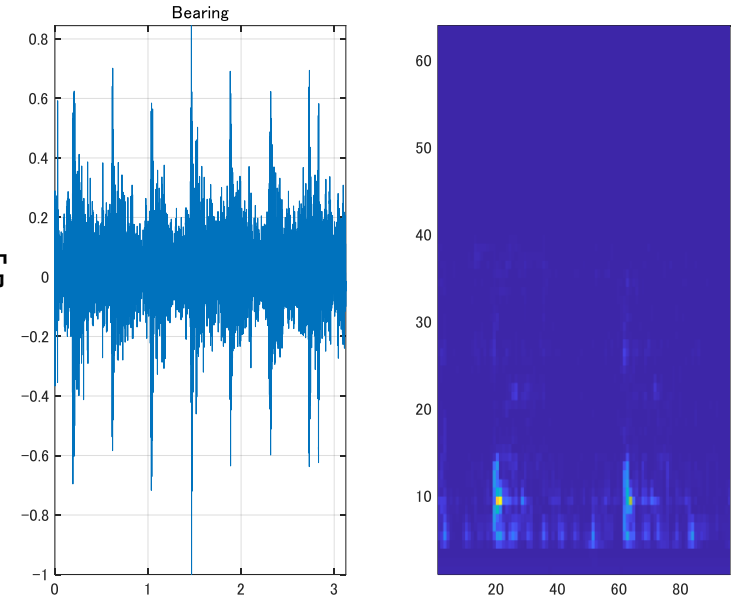


各波形の可視化

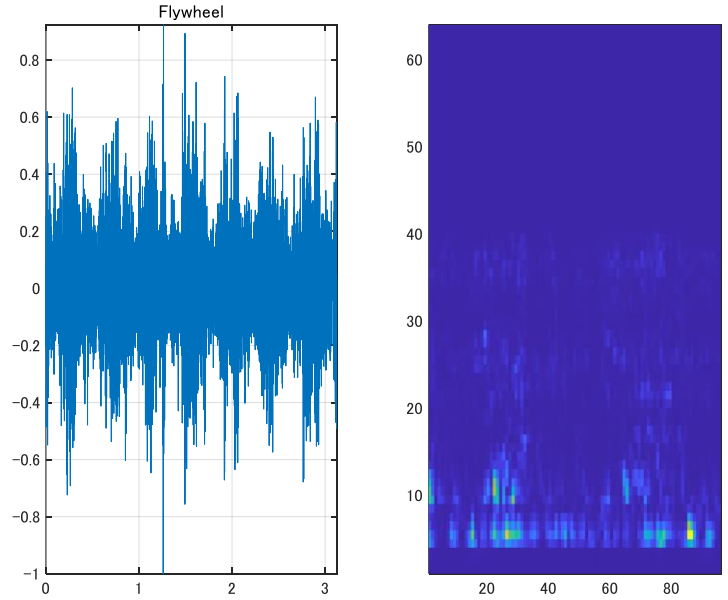
■正常



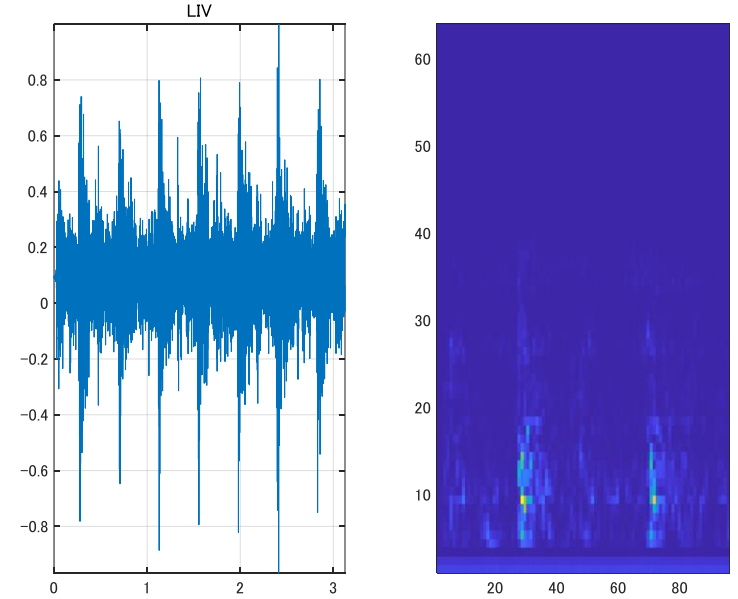
■ベアリング異常



■フライホイール異常

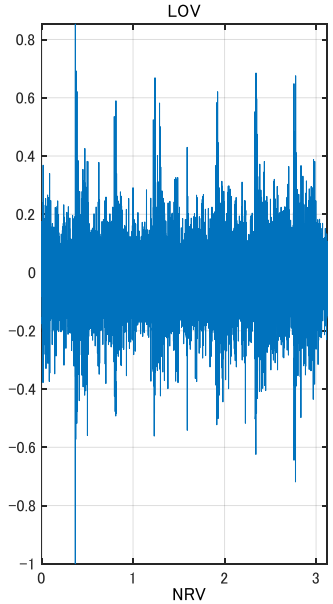


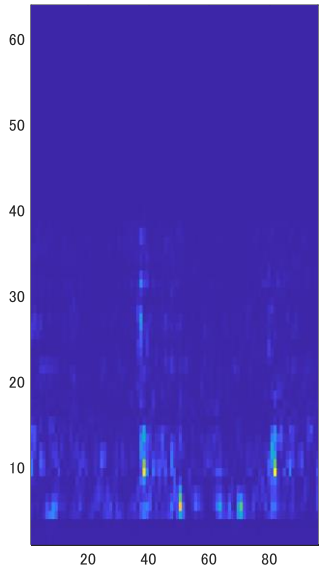
■インレット
バルブ漏れ異常



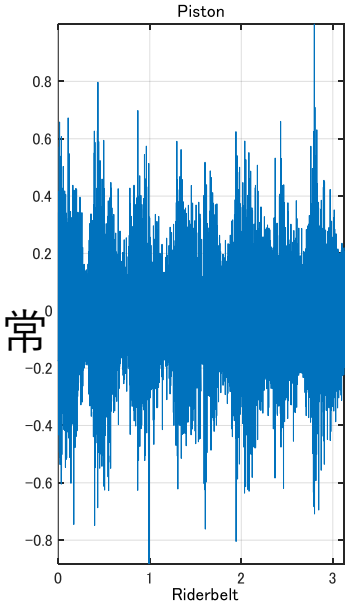
各波形の可視化（cont'd）

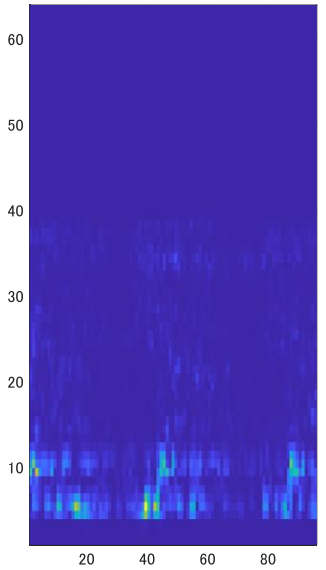
■アウトレット
バルブ漏れ異常



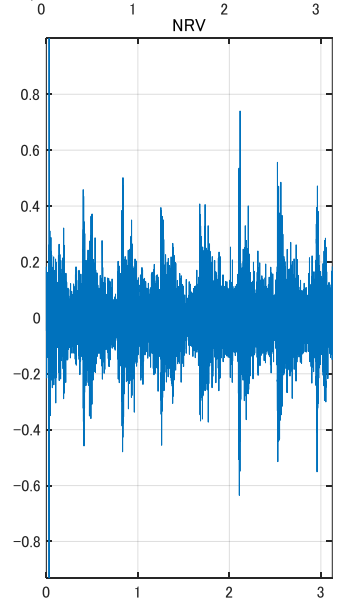


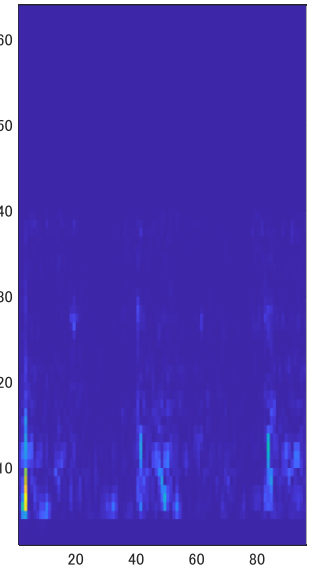
■ピストンリング異常



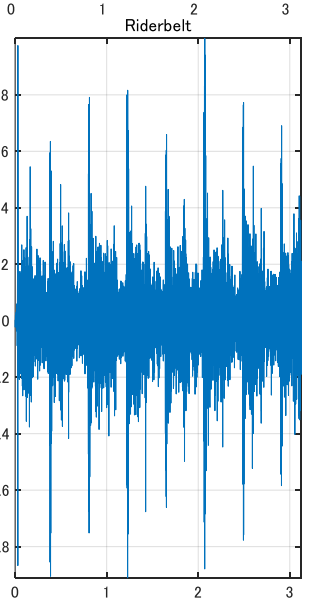


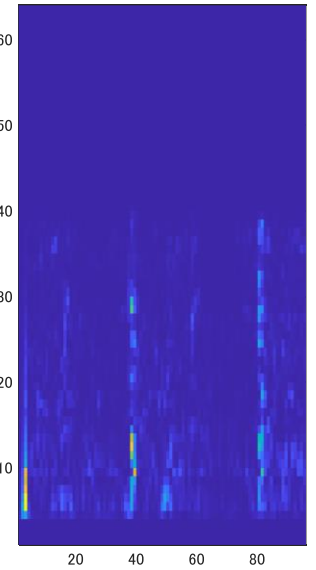
■ノンリターン
バルブ異常





■ライダーベルト異常





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

>>

Zoom: 100% UTF-8 LF script

混同行列による検証

学習済み
ネットワーク

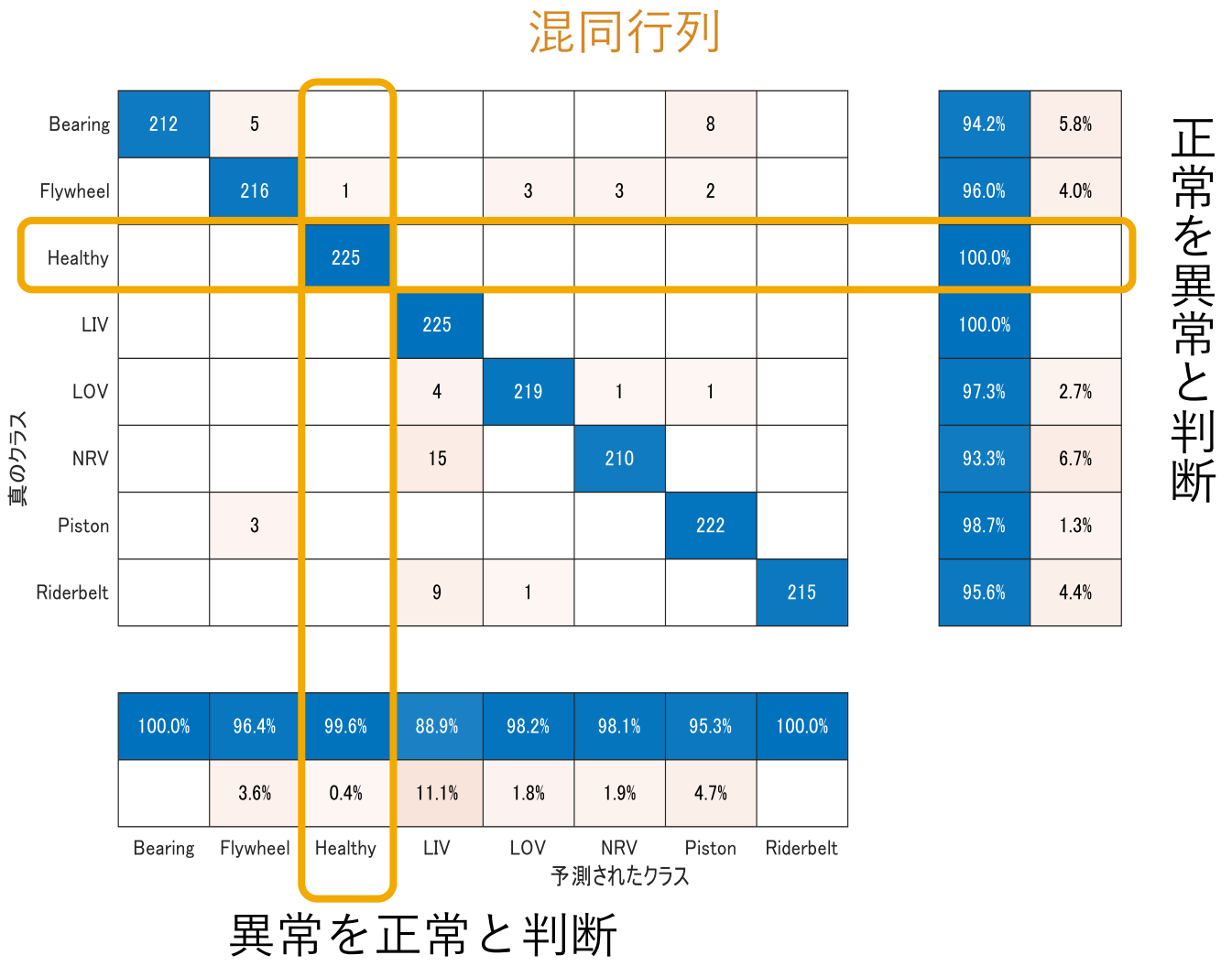
クラスが未知の
特徴量

```
% 学習済みネットワークによる分類
ValPred = classify(net, single(validationFeatures));
% 混同行列の表示
confusionchart(validationLabels, ValPred);
```

真のクラス

予測クラス

ご自身のデータに置き換えて
是非お試しください

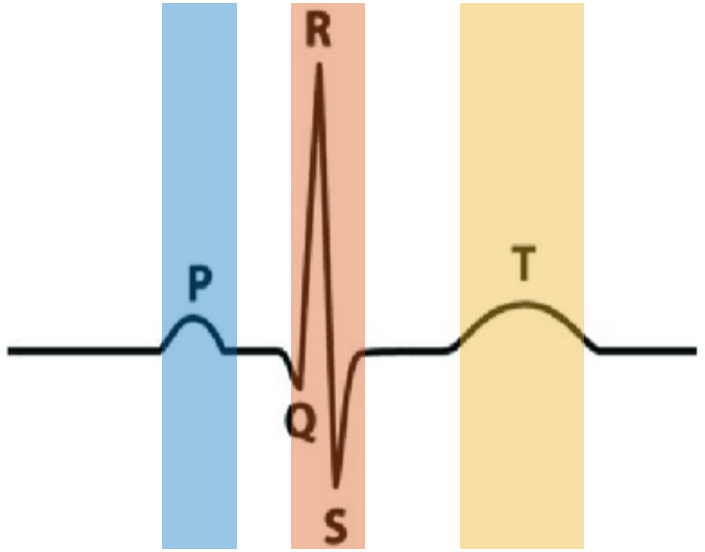


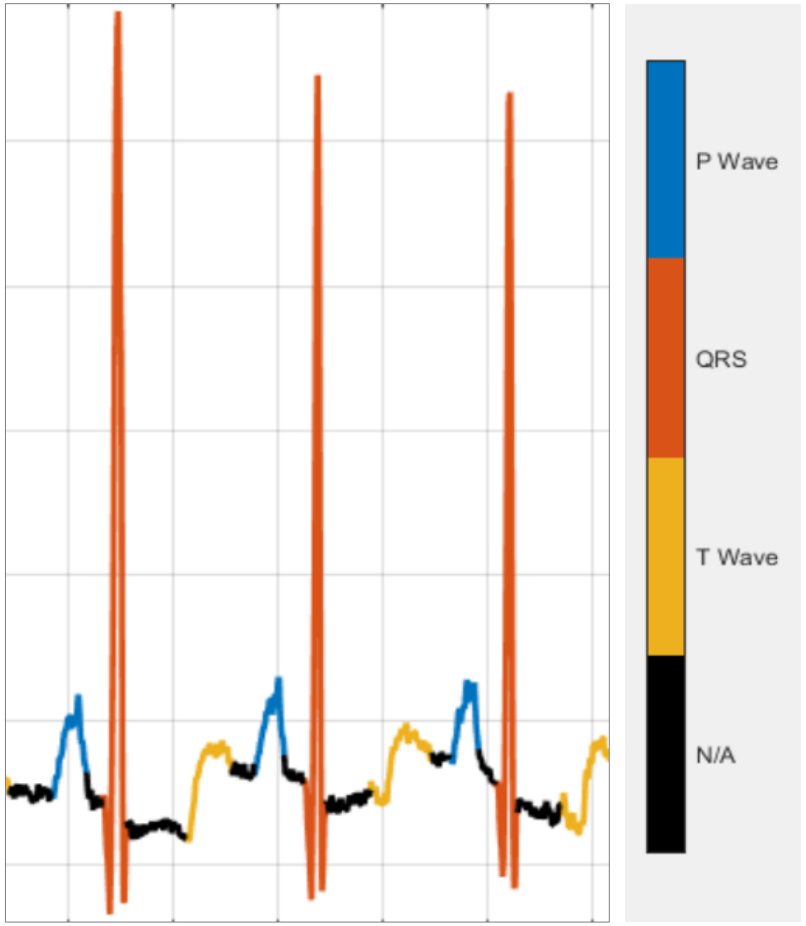
MATLABが選ばれる理由

1. データセットの置き換えですぐ試せる
2. 具体的かつ実践的な例題
3. 便利なアプリの活用

信号の領域分割の自動化例　～性能改善の手順～

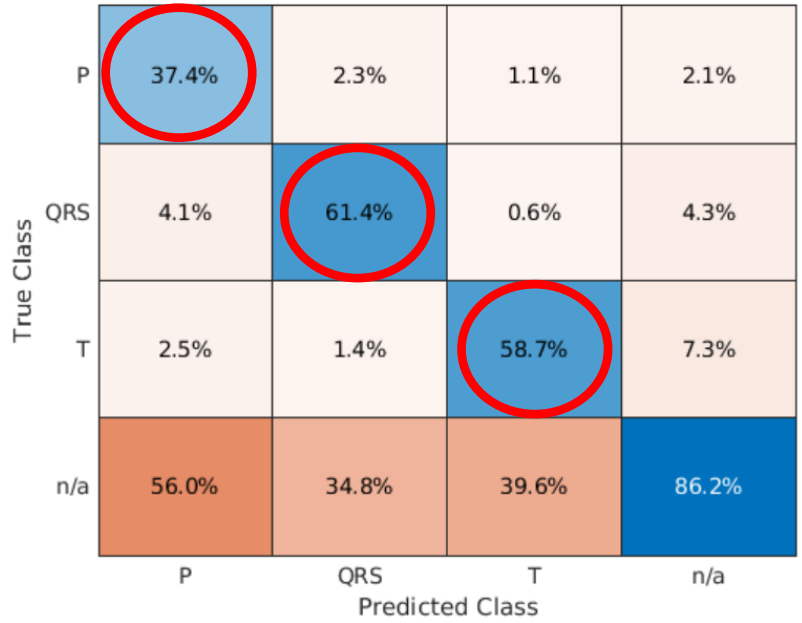
- 専門医がラベル付けしたデータセット
- 3種類の波形領域
- 210のECGデータ(~15 分)





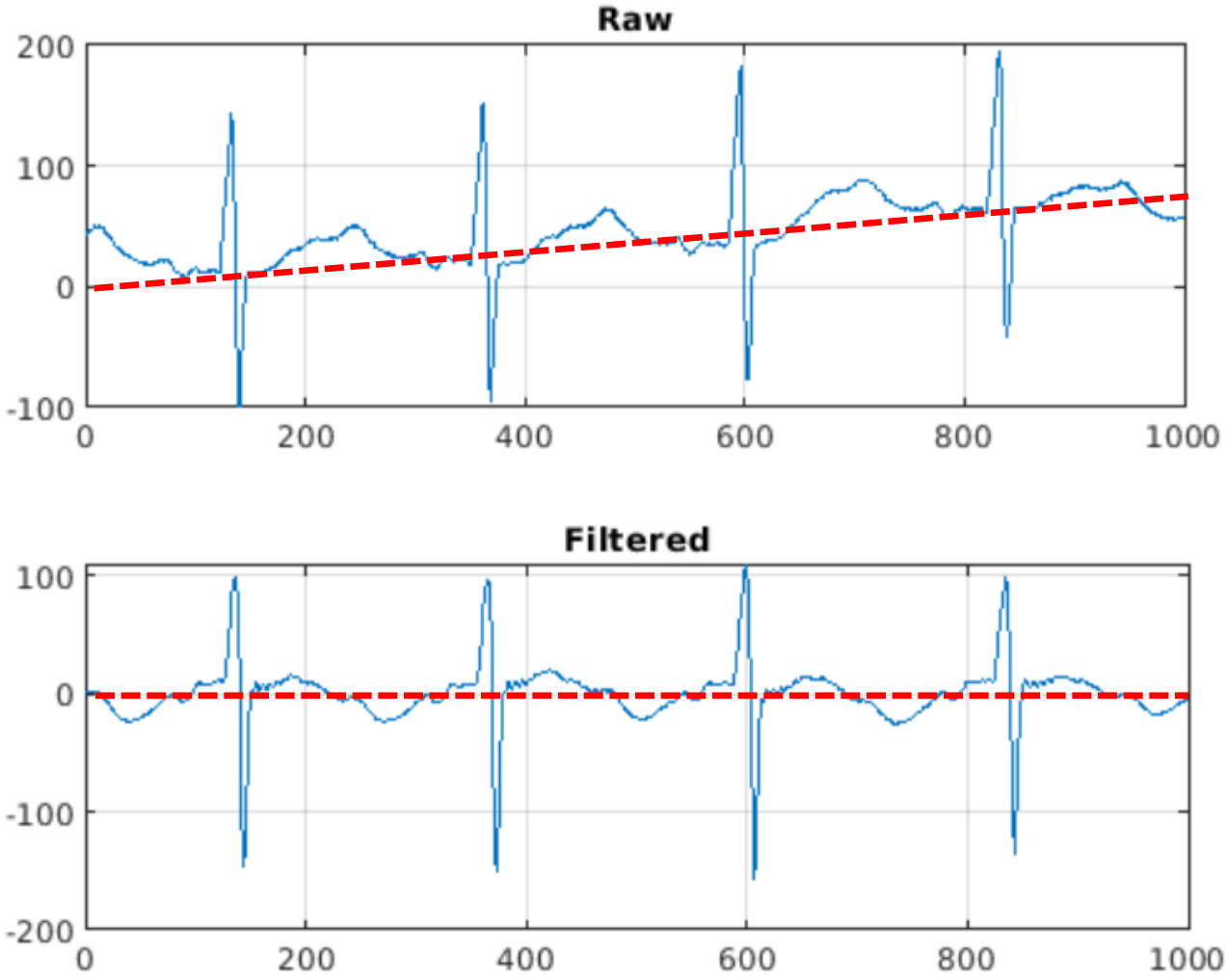
[Example: Waveform Segmentation Using Deep Learning]

生データを用いても満足する性能は得られない



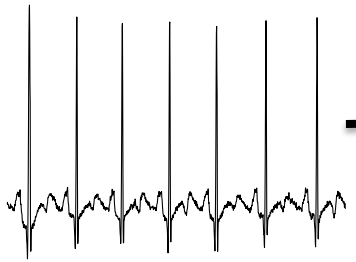

```
layers = [ ...
    sequenceInputLayer(1)
    lstmLayer(200,'OutputMode','sequence')
    fullyConnectedLayer(4)
    softmaxLayer
    classificationLayer];
```

信号の”質”を改善



必要な情報だけを
ネットワークに学習させる！

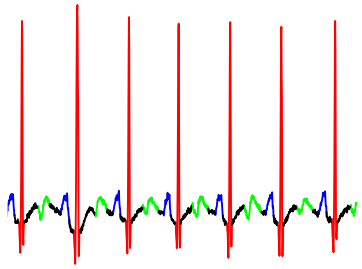
信号の”質”を改善



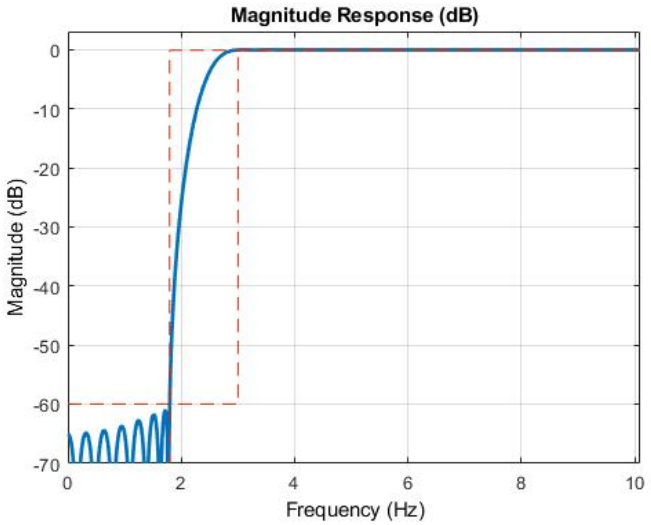
ECG

Filter

LSTM

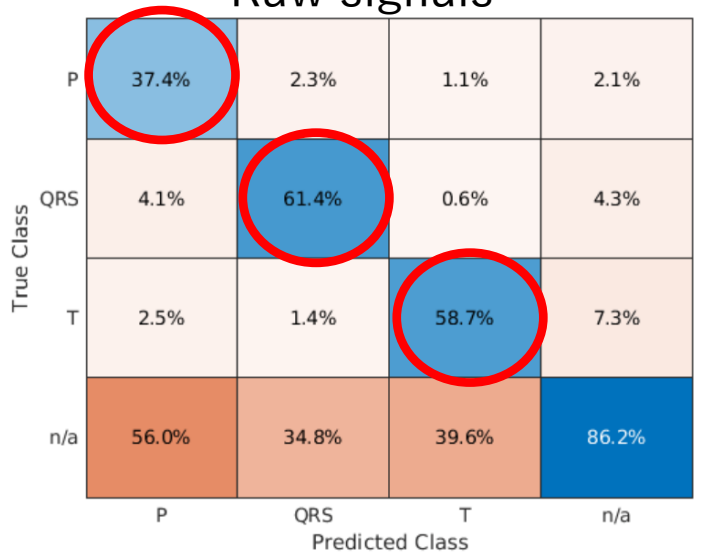


Labeled ECG

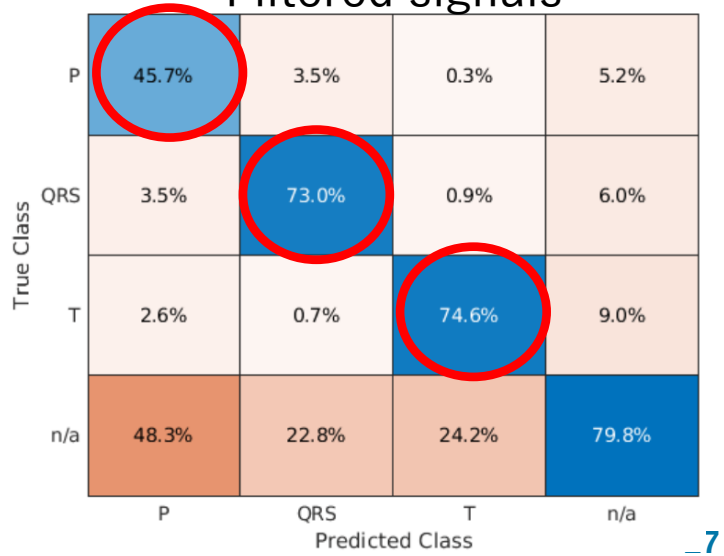


Filter baseline wander

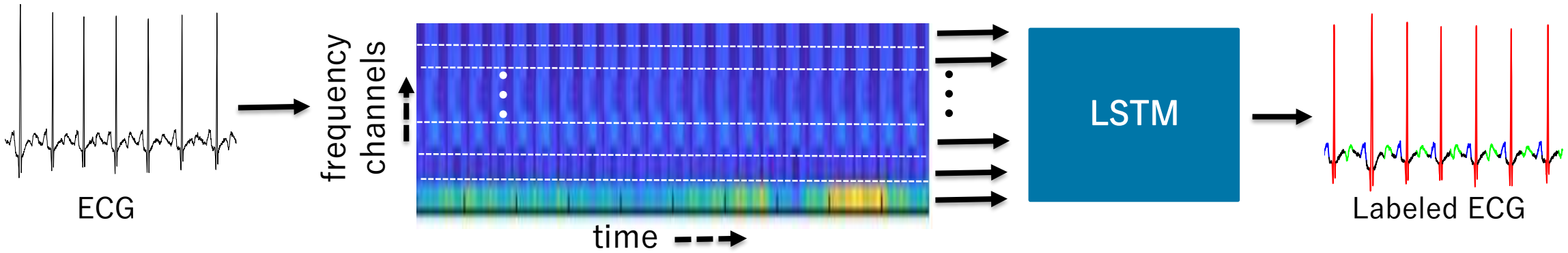
Raw signals



Filtered signals

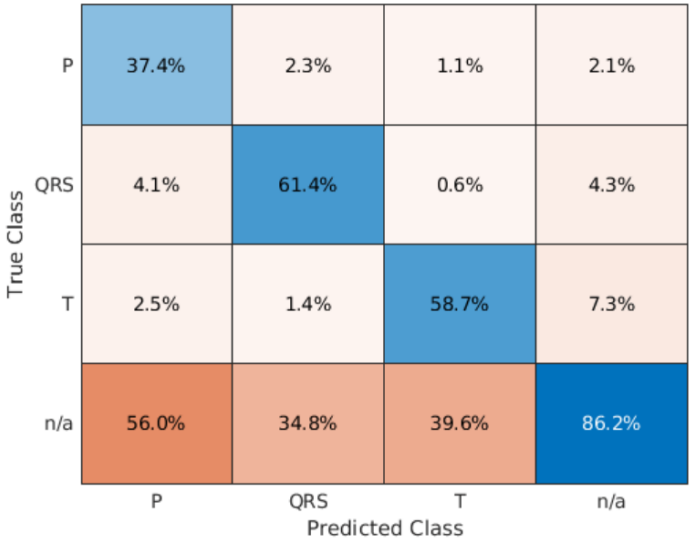


信号の特徴が、より際立つように変換処理を加える

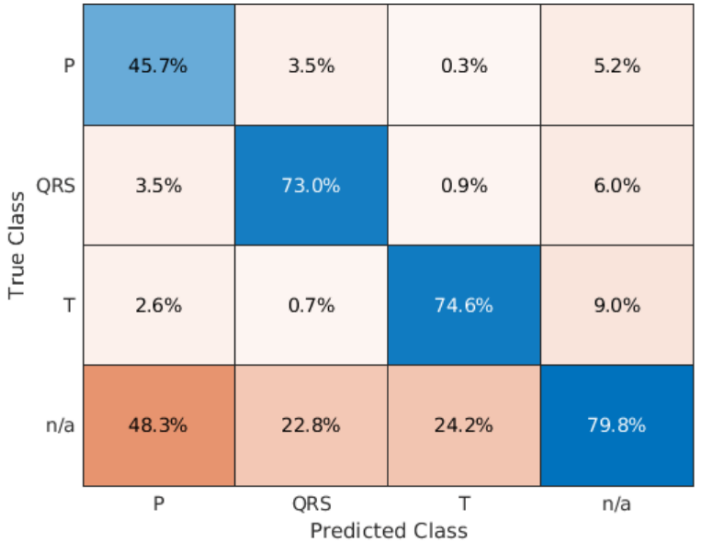


信号の特徴が、より際立つように変換処理を加える

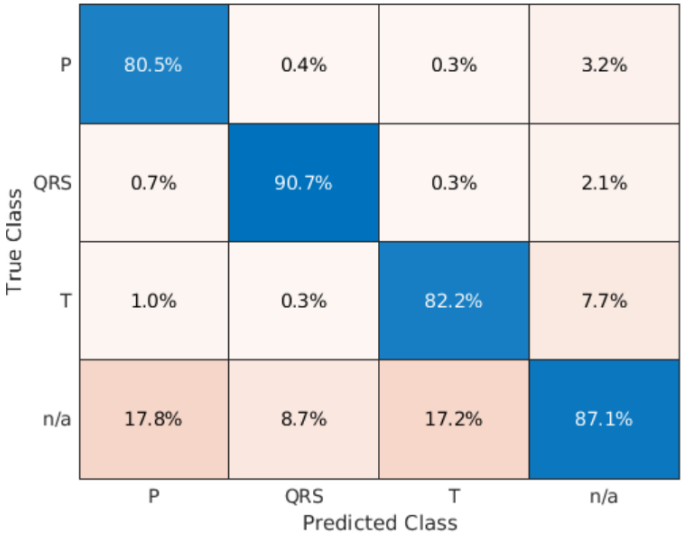
源信号（未処理）

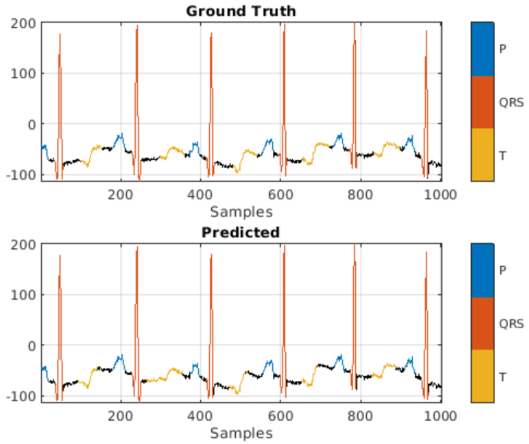


フィルタ後の信号



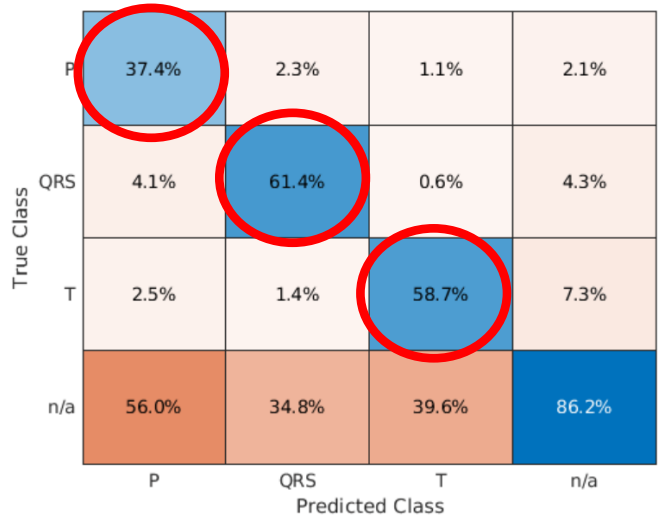
FSST

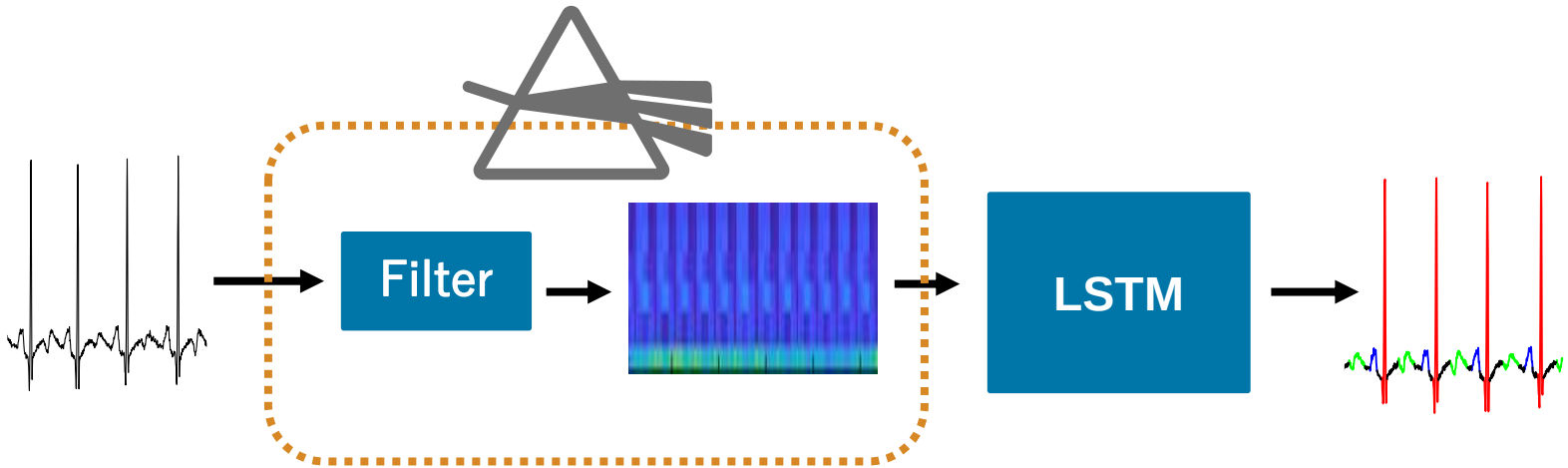


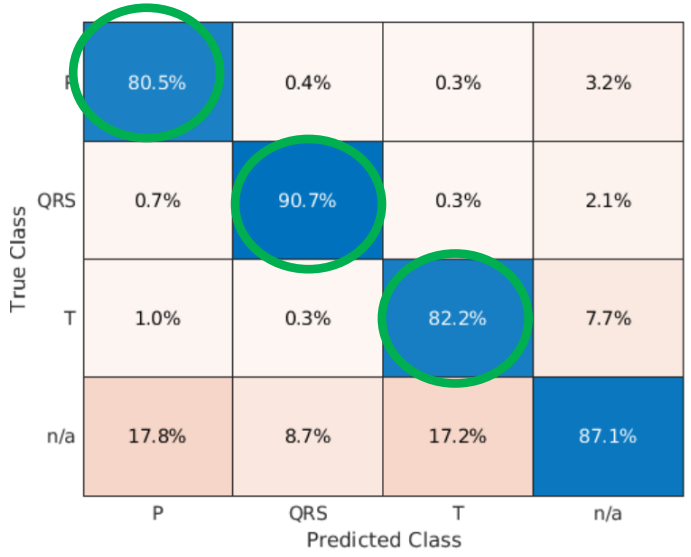


Region	Raw Signals Accuracy (%)	FSST Accuracy (%)
P	37.4	80.5
QRS	61.4	90.7
T	58.7	82.2

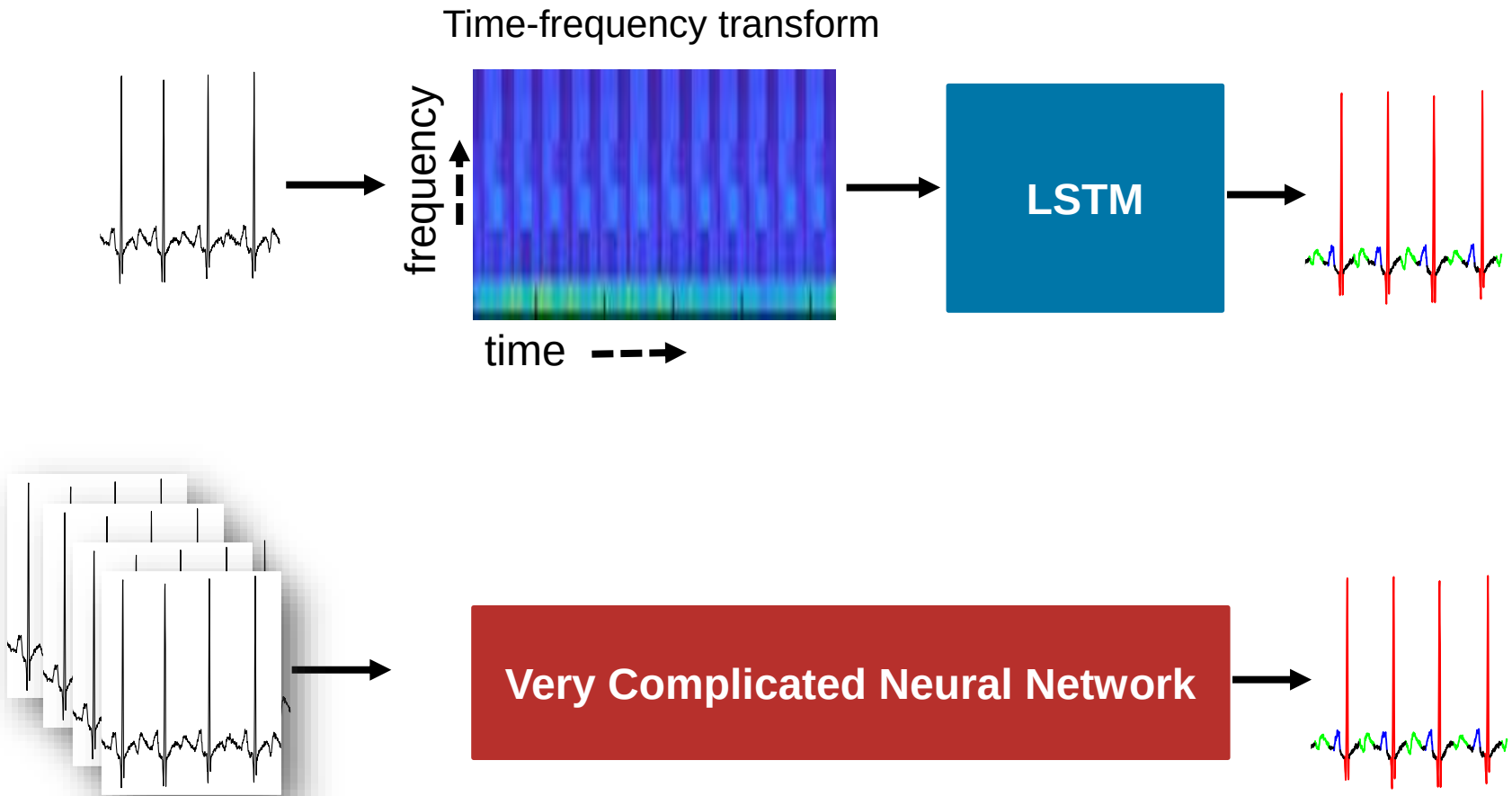
特徴抽出により、シンプルなAIモデルでも精度は上がる

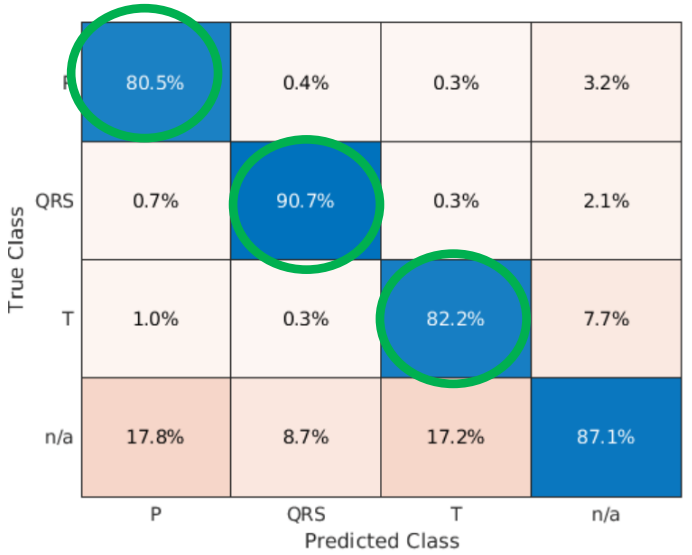






（逆に）特徴抽出により、モデルの複雑さを軽減





推論ロジックがコンパクトになれば…
⇒実装コストの低減

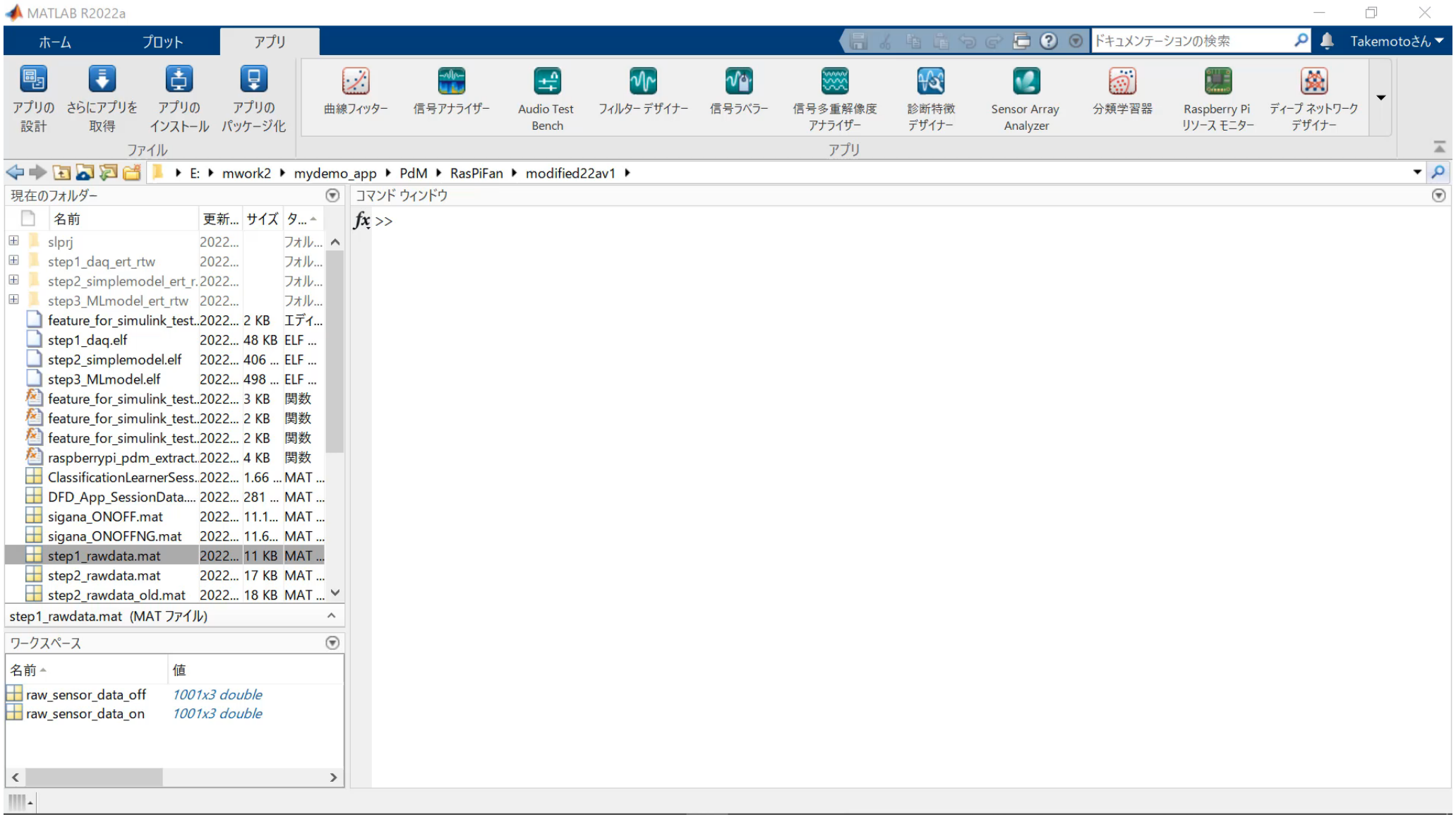
MATLABが選ばれる理由

1. データセットの置き換えですぐ試せる

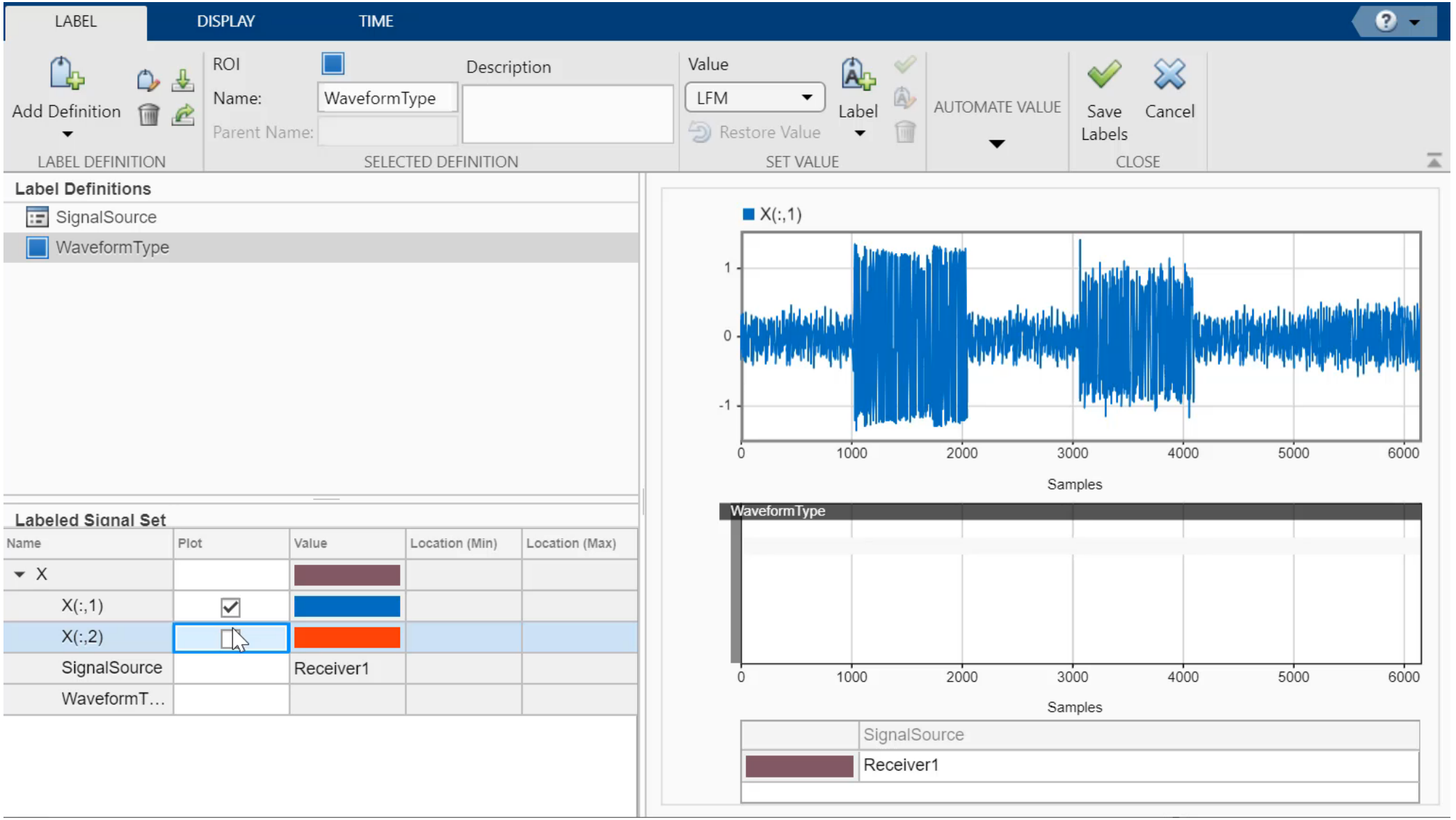
2. 具体的かつ実践的な例題

3. 便利なアプリの活用

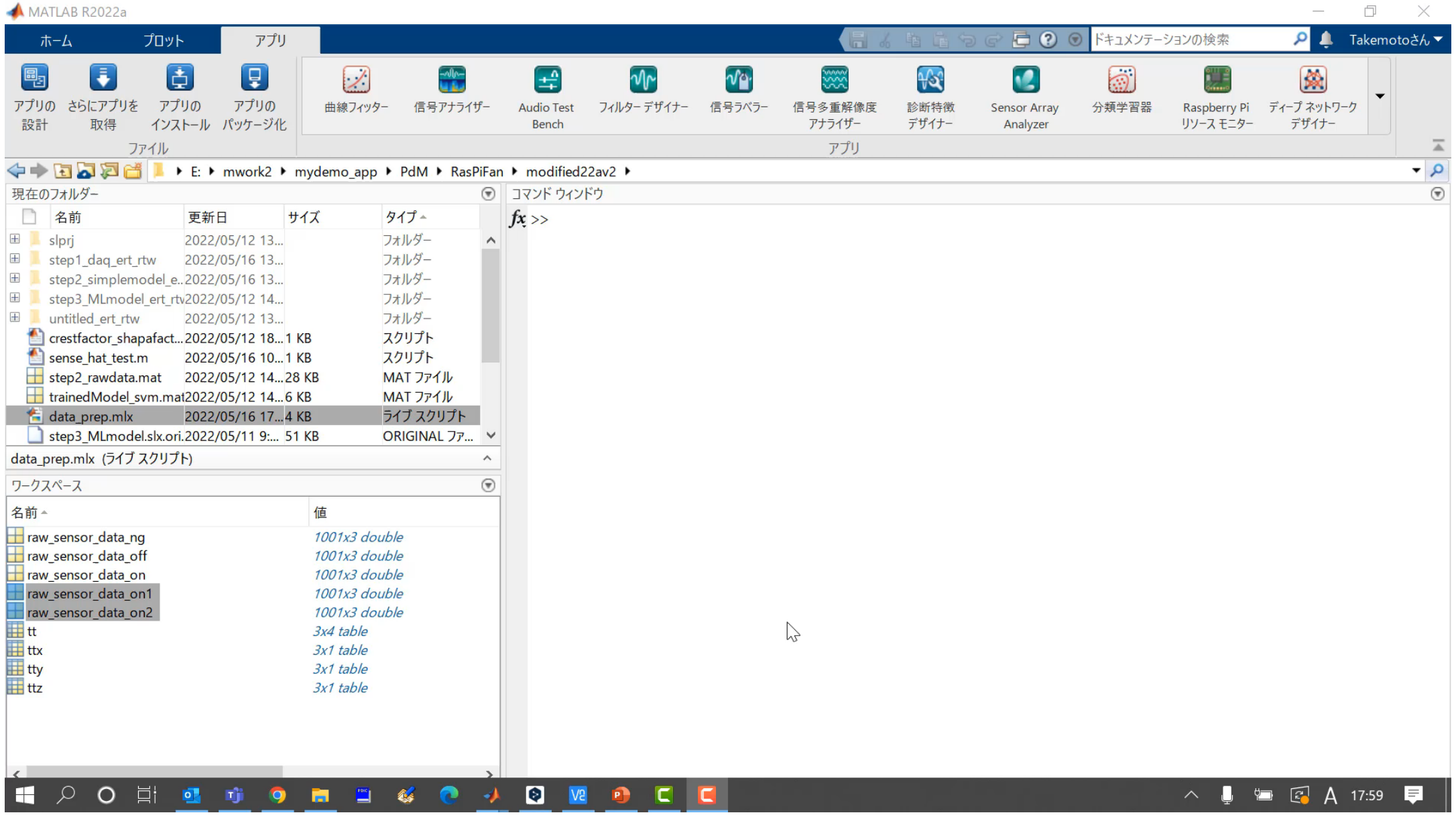
信号アナライザー（センサーデータの可視化、解析）



信号ラベラー（ラベリング作業の効率化）



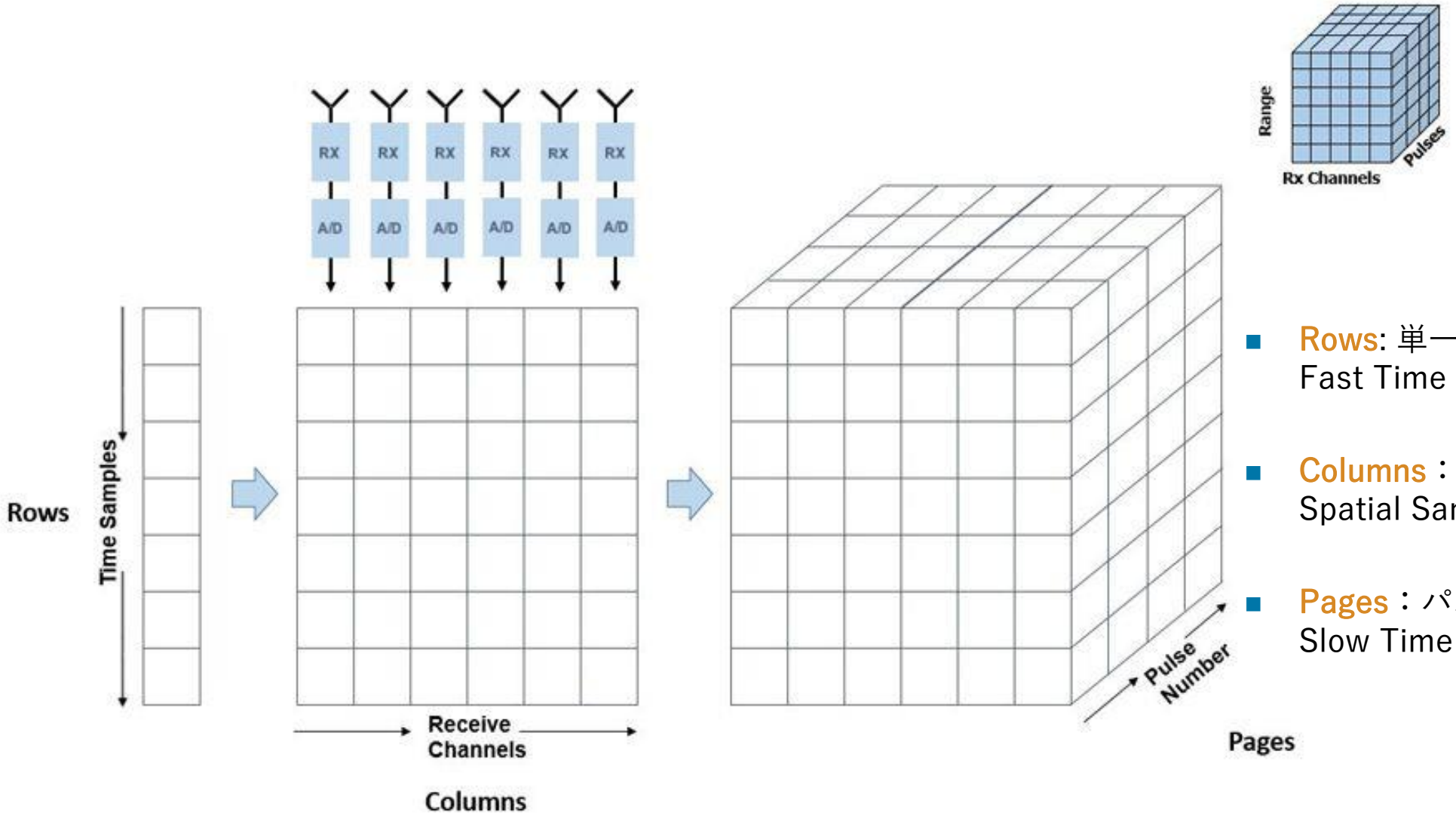
診断特徴デザイナー/分類学習器（特徴抽出、ランキング、学習）



Agenda

1. なぜ、信号処理が重要なのか？
2. 信号処理ツールとしてMATLABが選ばれてきた理由
3. 信号処理の各種例題のご紹介

レーダー信号の概要



- Rows: 単一レーダの反射
  Fast Time Sampling（距離）
- Columns：各素子からの反射
  Spatial Sampling（方向）
- Pages：パルス数
  Slow Time Sampling （速度）

レーダーシステムにおける信号処理（受信処理、後処理）

変位観測
対象物

TX

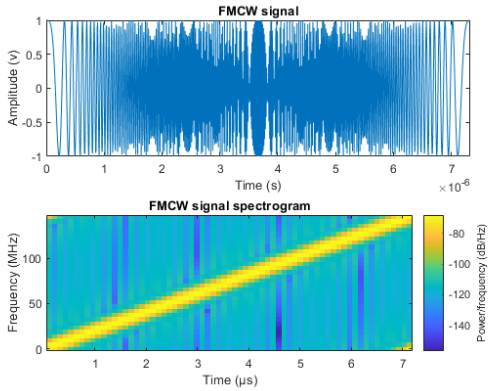
RX

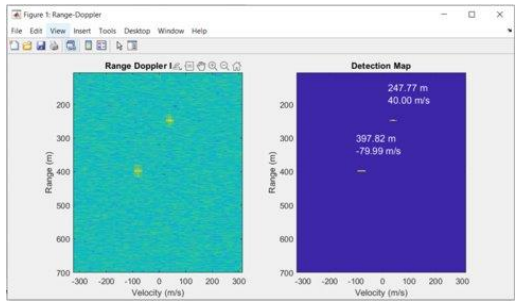
TRX IC

I/F

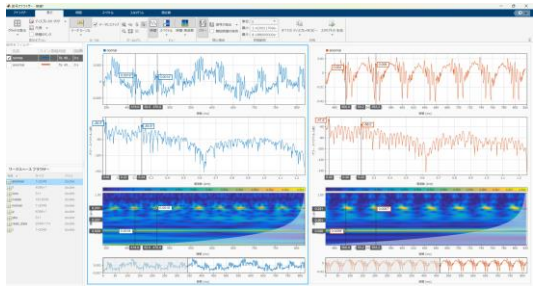
I/F信号への
信号処理

Ethernet





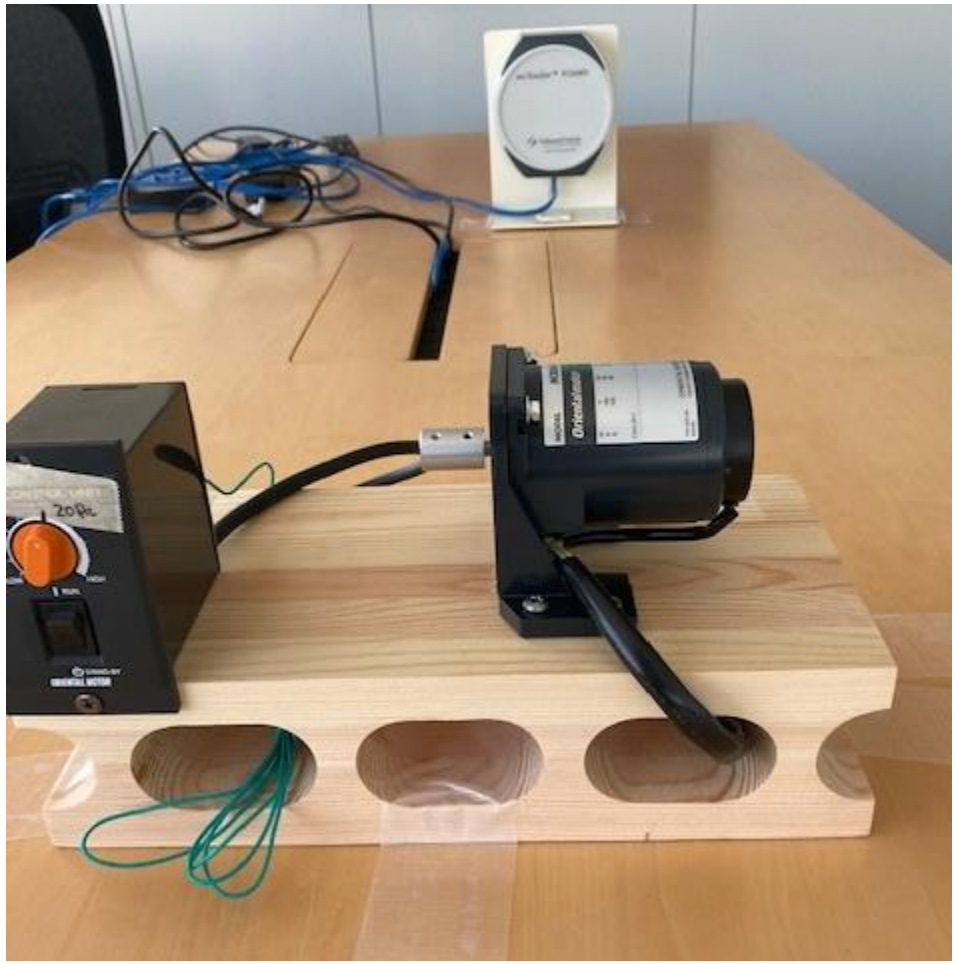
距離、速度、方向推定のための
信号処理



変位データに対する
信号処理

変位データによる異常判別（実験概要）

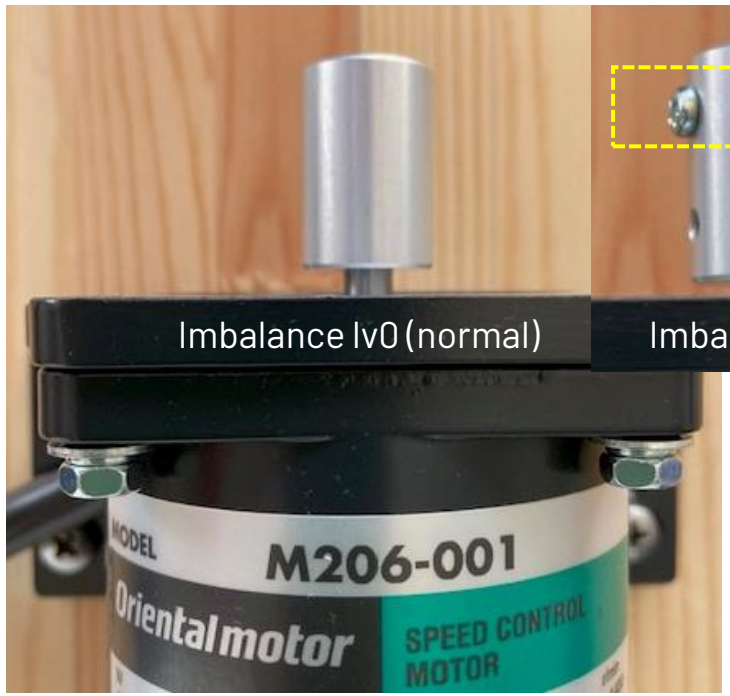
- モーター仕様
  - メーカー: オリエンタルモーター社
    - AC コントローラ: US206-001
  - 回転速度: 20rps
- レーダー仕様
  - 79-81GHz FMCW レーダー (ADAR6902)
  - FMCW ramp-to-ramp period: 600µs (1667Hz)
  - Burst ramps: 2048
- モーター偏心 (次ページ)
  - Imbalance lv0: Normal
  - Imbalance lv1: 1 screw
  - Imbalance lv2: 1 screw + 1 nut
  - Imbalance lv3: 1 screw + 2 nuts
  - Imbalance lv4: 1 screw + 3 nuts
  - Imbalance lv5: 1 screw + 4nuts

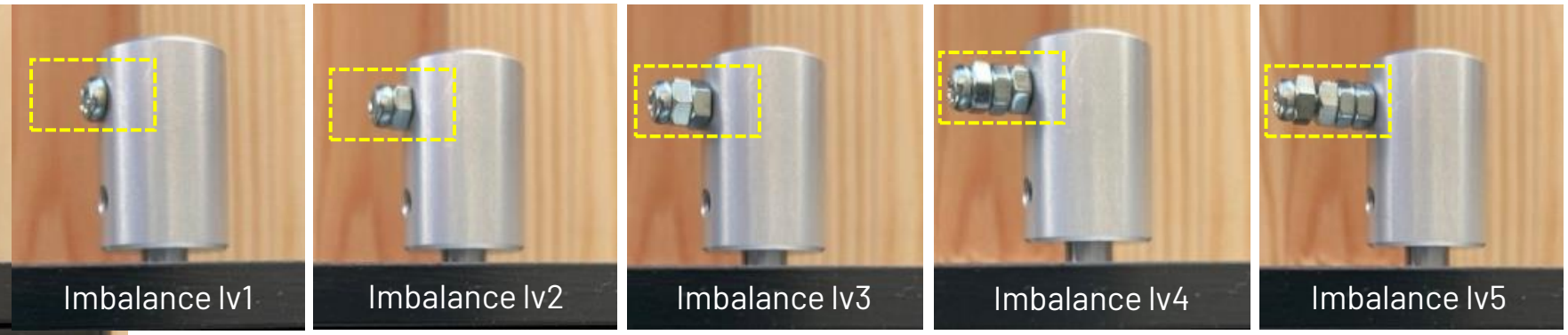


正常・異常（偏心）のレベル

normal

anormal





■ 正常・異常の度合いを
全6クラスで定義

信号アナライザによる特徴の吟味

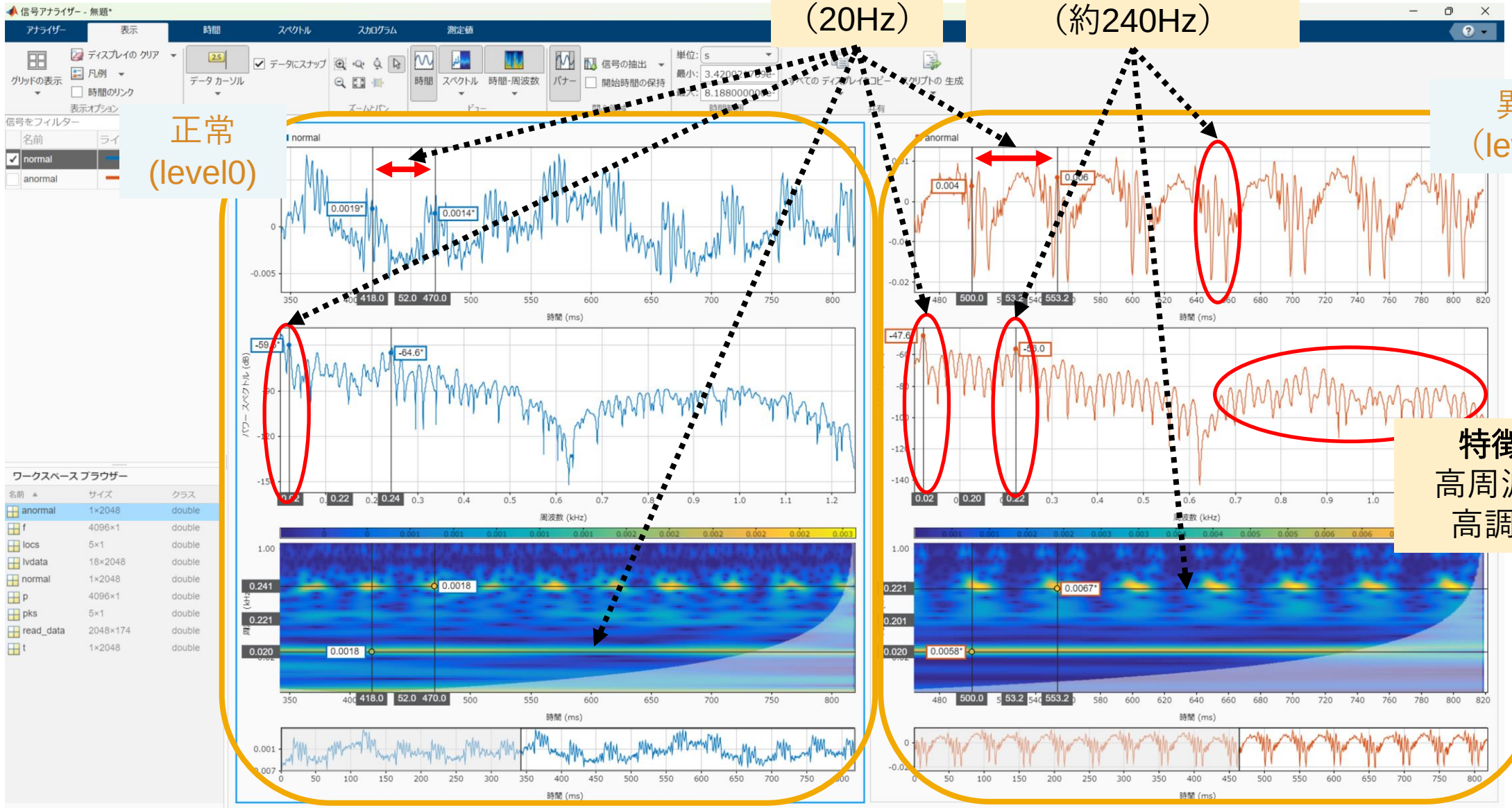
特徴1
回転周期
（20Hz）

特徴2
周期的に現れる成分
（約240Hz）

正常
(level0)

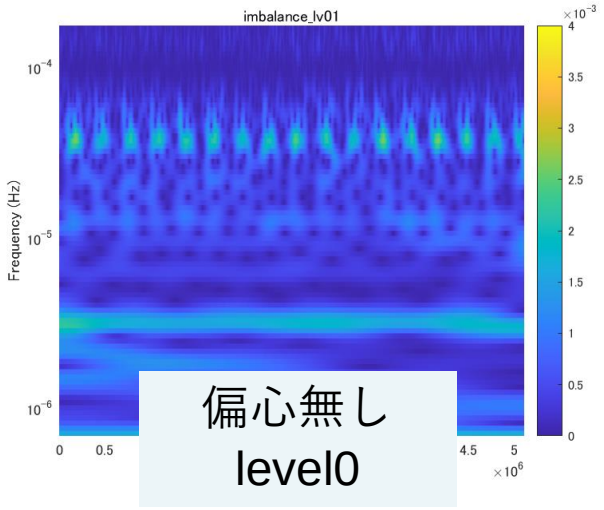
異常
（level5）

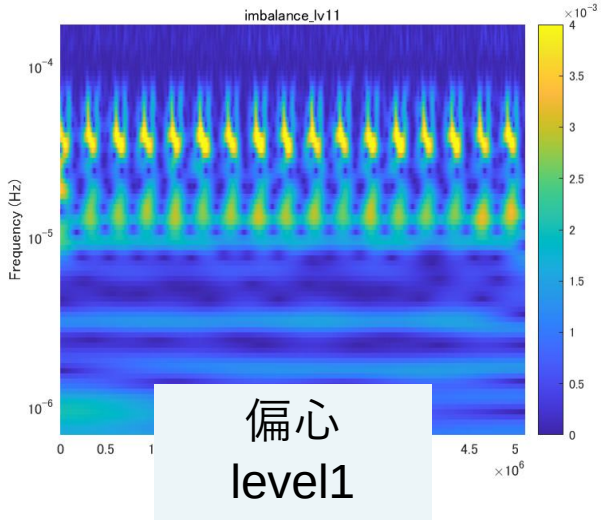
特徴3
高周波帯
高調波

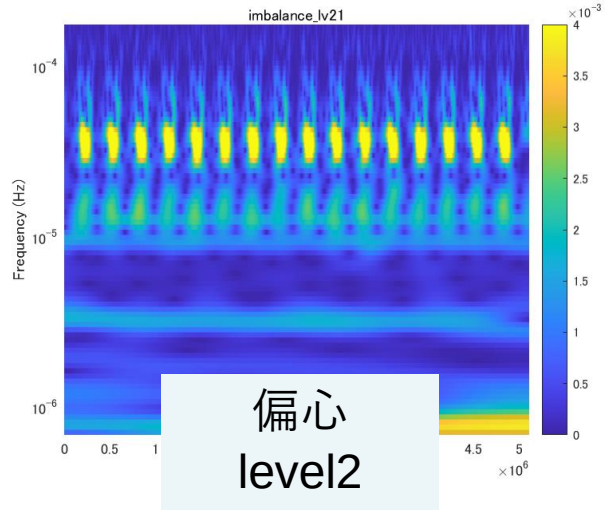


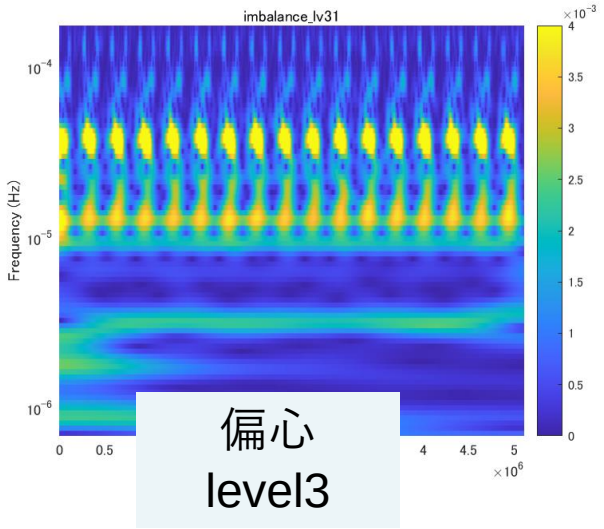
6クラスの波形比較（時間軸、周波数軸）

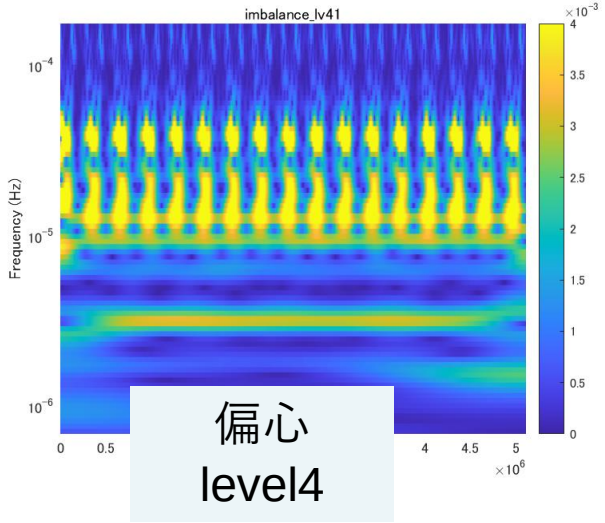
時間-周波数特性（ウェーブレット）

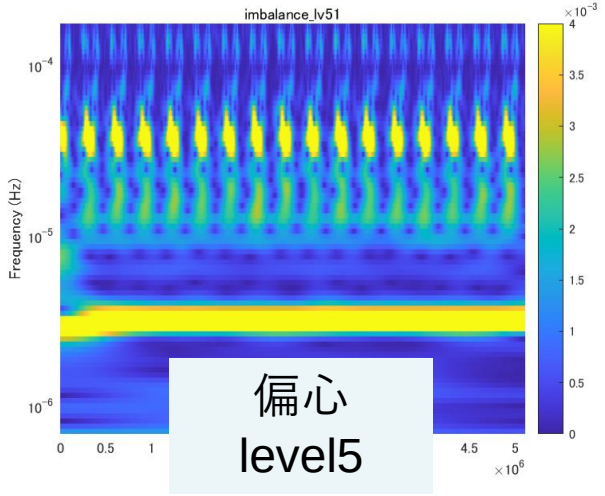












各レベルで
特徴に違いあり

異常度の増加に
伴う単調増加
とは限らない

単純な閾値判定
では判別は困難

ディープラーニン
グの得意分野

学習用データセットの用意

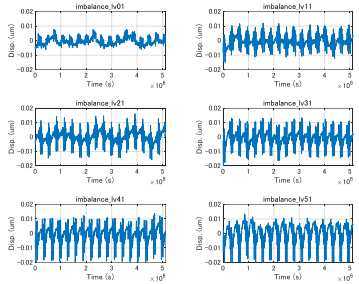
CSVファイルの読み込み

readmatrix

	FI	FJ	FK	FL	FM	FN	FO	FP	FQ	FR	FS
2342											
2343	-0.00507	-0.00239	-0.00394	0.001004	0.001955	0.001602	-0.003	-0.00173	-0.0036	0.000087	
2344	-0.00562	-0.00307	-0.00491	0.000876	0.001992	0.000628	-0.00365	-0.00228	-0.00412	-0.0006	
2345	-0.00699	-0.00526	-0.00619	-5.7E-05	0.000743	-0.00105	-0.00542	-0.00394	-0.00529	-0.00109	
2346	-0.0068	-0.00443	-0.00406	0.001695	0.000767	-0.00096	-0.00512	-0.00358	-0.0045	-0.00045	
2347	-0.00673	-0.00417	-0.00167	0.002174	-0.00014	-0.00147	-0.00508	-0.00404	-0.00464	-0.00048	
2348	-0.00561	-0.00416	0.000621	0.002667	-0.00154	-0.00052	-0.00459	-0.00416	-0.00482	0.000742	

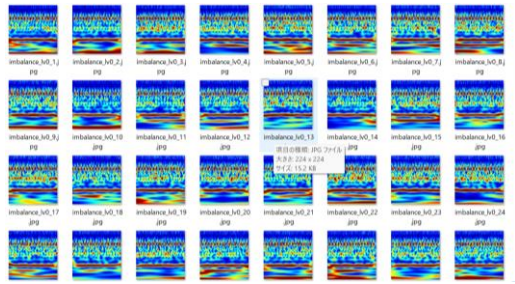
変位データ（時系列データ）の切り出し

- サンプリング周波数：400μs
- データ点数：1024点
- データの種類：6種類（正常:1、異常:5）
- ファイル数：3132



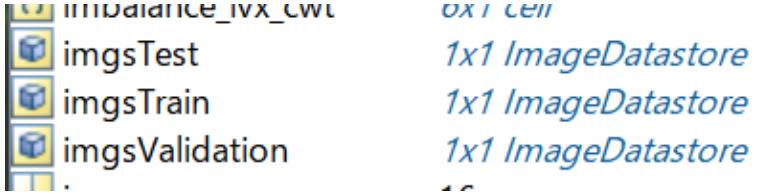
全データについてウェーブレット変換後.jpg形式でクラス別に保存

- jpegフォーマットとして保存
- ディレクトリ名: imbalance_lv* (*はレベル0~5）
- ファイル数は各クラス同数（3132/6 = 522)



imageDatastore定義後、学習用、検証用、テスト用に分割

- 学習用：2190 (3132の約70%）
- 検証用：630 (3132の約20%)
- テスト用：312 (3132の約10%)



学習パラメータの設定、学習

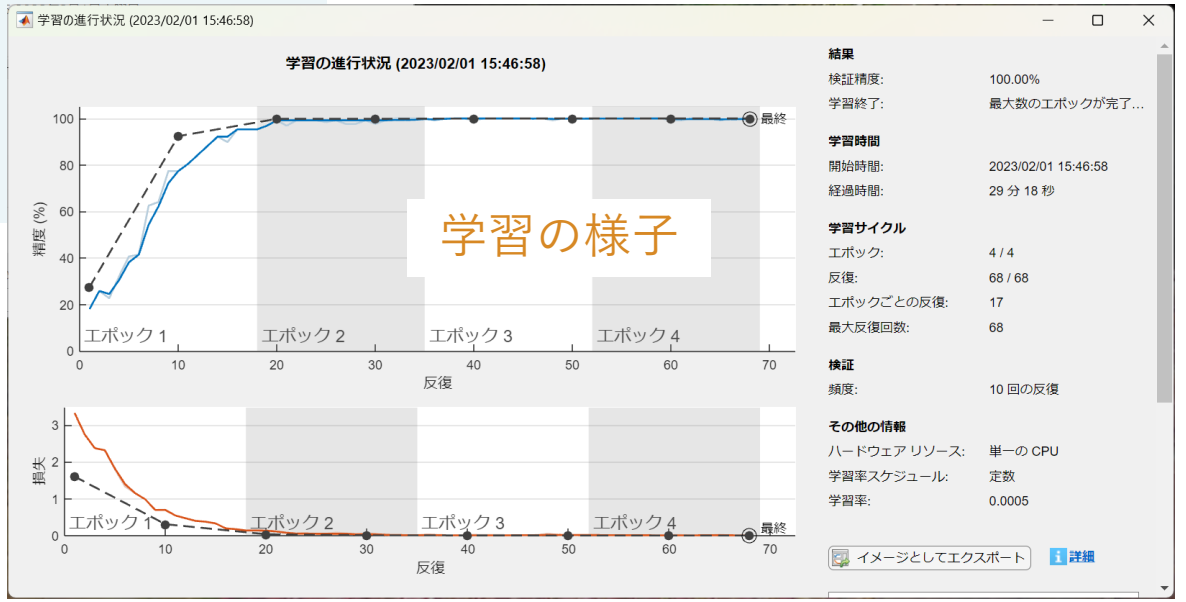
GoogLeNetの編集

```
net = googlenet;

lgraph = layerGraph(net);
numberOfLayers = numel(lgraph.Layers);

newDropoutLayer = dropoutLayer(0.6,'Name','new_Dropout'); % 過適合対策
lgraph = replaceLayer(lgraph,'pool5-drop_7x7_s1',newDropoutLayer);

newConnectedLayer = fullyConnectedLayer(6,'Name','new_fc',... % 分類数で置き換え
'WeightLearnRateFactor',5,'BiasLearnRateFactor',5);
lgraph = replaceLayer(lgraph,'loss3-classifier',newConnectedLayer);

newClassLayer = classificationLayer('Name','new_classoutput');
lgraph = replaceLayer(lgraph,'output',newClassLayer);
```

学習オプション

```
options = trainingOptions('sgdm',...
'MiniBatchSize',128,...
'MaxEpochs',4,...
'InitialLearnRate',5e-4,...
'ValidationData',imgsValidation,...
'ValidationFrequency',10,...
'Verbose',1,...
'ExecutionEnvironment','cpu',...
'Plots','training-progress');
```

学習の様子

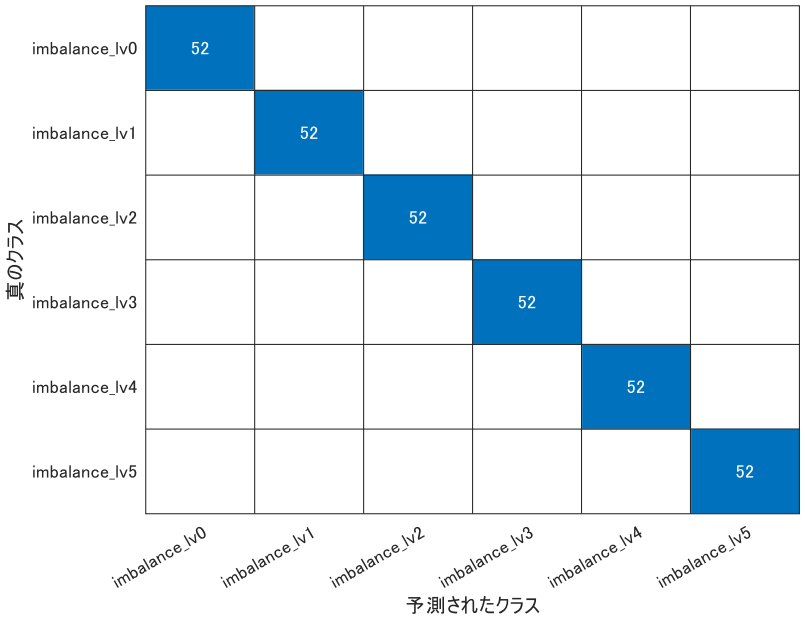


GoogLeNetの再学習

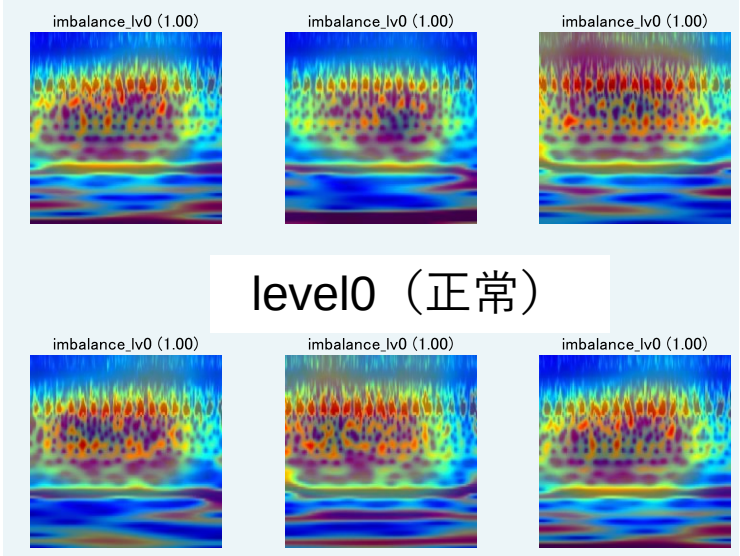
```
>>trainedGN = trainNetwork(imgsTrain,lgraph,options);
```

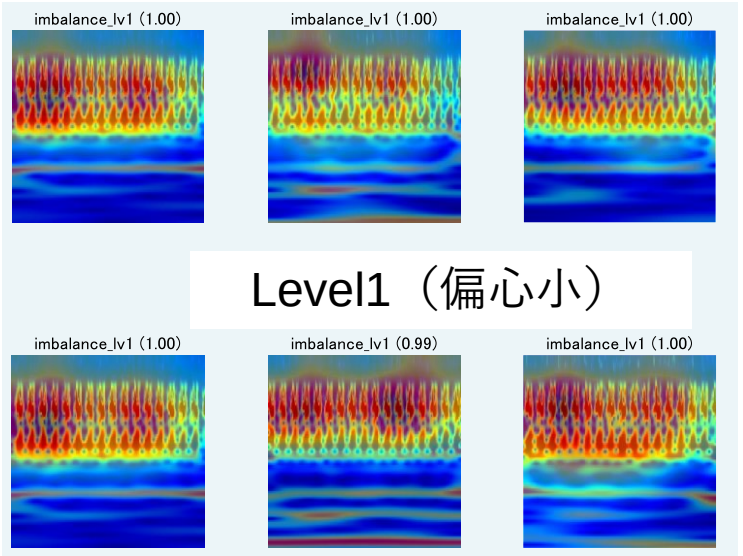
混同行列、Grad-CAMによる検証

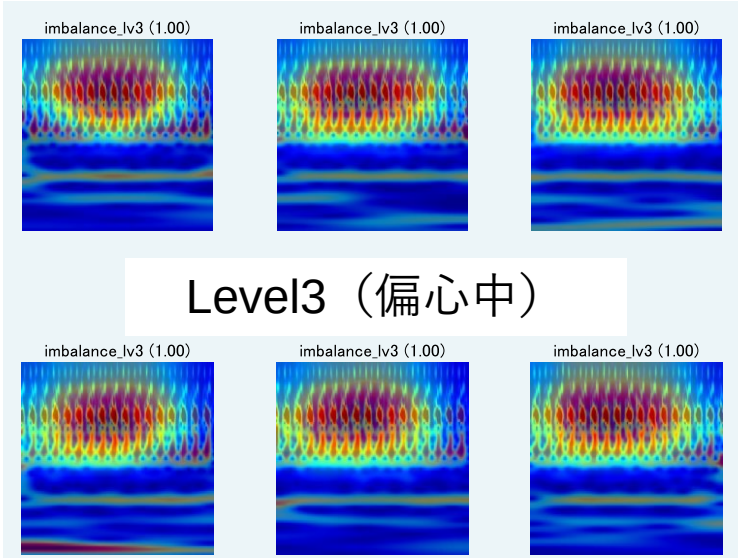
混同行列

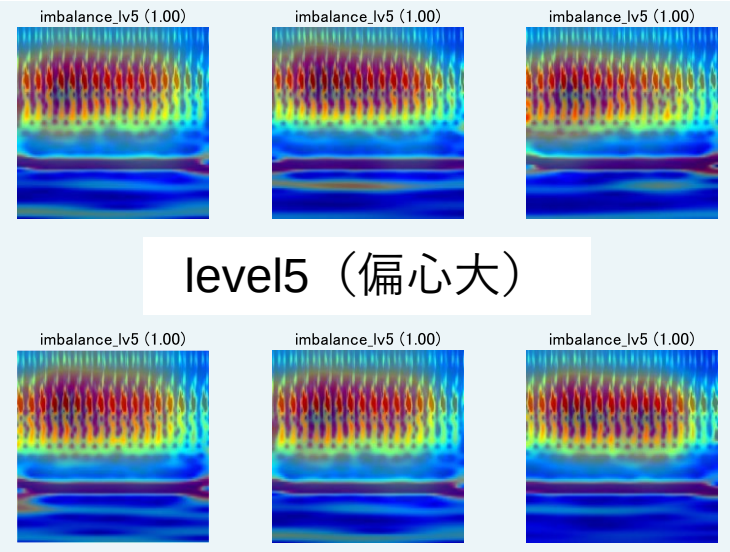


Grad-CAM









AI 関連例題

51 Vertical Application Examples

アプリケーション	Analyze	Label	Synthesize	Engineer Features	Classify (AI)	Regress (AI)	Deploy
Biosignals	2	2	1	12	8	2	4
Radar/Comms	2	1	3	1	2	1	1
Vibrations	4		4	3	3	2	
Power Systems	3		3	1	1		
Speech & Audio	2	2		5	3	1	
Predictive Maintenance*			1	1	1		
Earth & Ocean	1	1					
Auto				1	1		
Finance	1						

** excl. Predictive Maintenance Tbx examples*

- Analyze ：データアクセス、データ管理、可視化、データクレンジング
- Label ：信号のらベリンツ（マニュアルおよび自動ラベリング）
- Synthesize ：信号生成、データの水増し、GANによる信号合成
- Deploy ： CPUおよびGPU向け自動コード生成

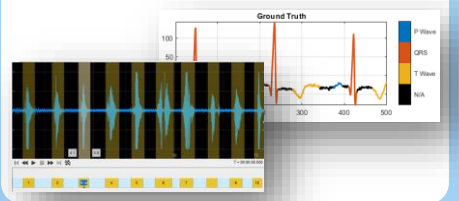
まとめ：信号処理・AIシステム開発のワークフロー（再掲）

1. データセットの作成

■ データソース

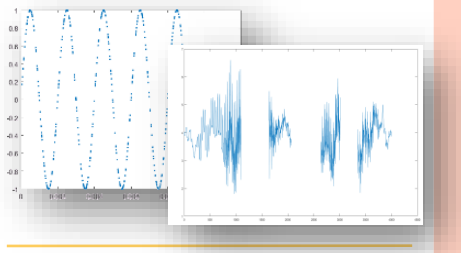
■ シミュレーション
■ オーグメンテーション

■ データラベリング

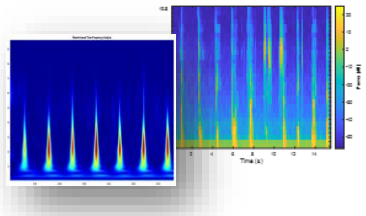


2. 前処理と変換

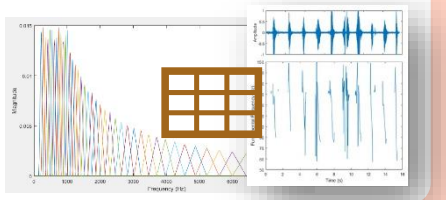
■ 前処理



■ 変換



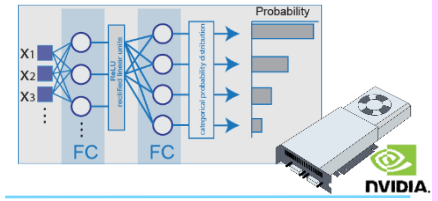
■ 特徴抽出



3. 予測モデルの開発

■ リファレンスモデルの流用
■ オリジナルモデルの作成

■ GPUによる高速な学習

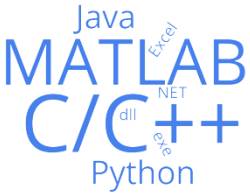


■ ハイパーパラメータの
解析とチューニング

4. 実システムへの展開

■ デスクトップアプリ

■ エンタープライズシステム



■ エッジ（組み込み）デバイス