

# MATLAB EXPO

## The Key Role of Data in Modern AI-Powered Systems – Spotting Voice Keywords and Beyond

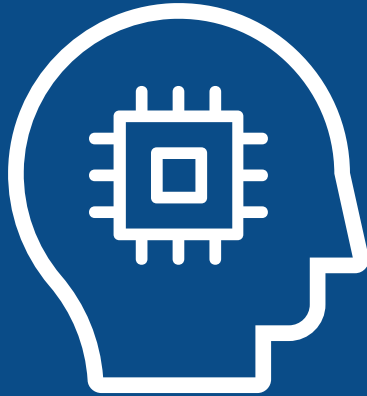
*Gabriele Bunkheila, MathWorks*



# Deep learning is a key technology driving the AI megatrend

## ARTIFICIAL INTELLIGENCE

Any technique that enables machines to mimic human intelligence



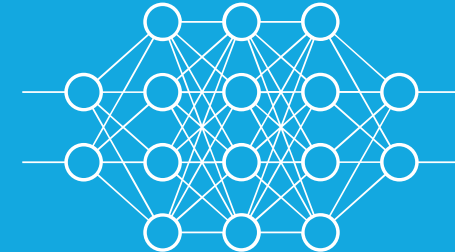
## MACHINE LEARNING

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

What does it take to develop an effective real-world deep learning system for signal processing applications?

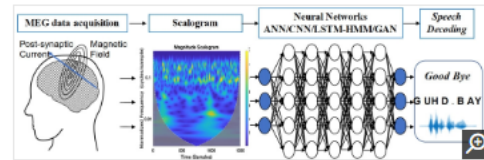


# Deep learning use in signal processing applications is growing rapidly

## UT Austin Researchers Convert Brain Signals to Words and Phrases Using Wavelets and Deep Learning

"MATLAB is an industry-standard tool, and one that you can trust. It is easier to learn than other languages, and its toolboxes help you get started in new areas because you don't have to start from scratch."

— Dr. Jun Wang, UT Austin



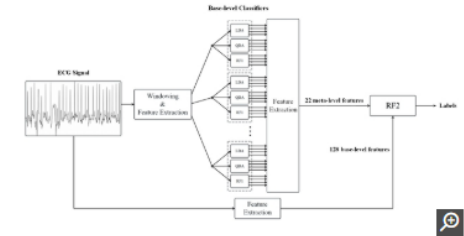
Classifying the brain signals corresponding to the imagined word "goodbye" using feature extraction and deep neural networks.

[https://www.mathworks.com/company/user\\_stories/ut-austin-researchers-convert-brain-signals-to-words-and-phrases-using-wavelets-and-deep-learning.html](https://www.mathworks.com/company/user_stories/ut-austin-researchers-convert-brain-signals-to-words-and-phrases-using-wavelets-and-deep-learning.html)

## MATLAB Based Algorithm Wins the 2017 PhysioNet/CinC Challenge to Automatically Detect Atrial Fibrillation

"I don't think MATLAB has any strong competitors for signal processing and wavelet analysis. When you add in its statistics and machine learning capabilities, it's easy to see why nonprogrammers enjoy using MATLAB, particularly for projects that require combining all these methods."

— Ali Bahrami Rad, Aalto University



Block diagram for Black Swan's atrial fibrillation detection algorithm.

[https://www.mathworks.com/company/user\\_stories/matlab-based-algorithm-wins-the-2017-physionet-cinc-challenge-to-automatically-detect-atrial-fibrillation.html](https://www.mathworks.com/company/user_stories/matlab-based-algorithm-wins-the-2017-physionet-cinc-challenge-to-automatically-detect-atrial-fibrillation.html)

## Shell performs Seismic Event Detection with Deep Learning

### Challenges

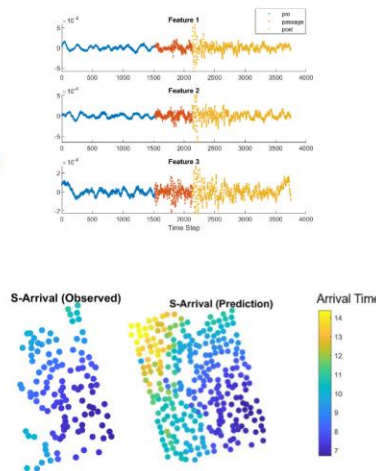
- Terabytes of passive seismic data from geophones
- Traditional methods time/labor intensive (5 months &~ \$100K)
- Event detection inconsistent/unreliable in 'low' signal to noise records

### Solution

- Train LSTM network to detect P-wave and S-wave arrivals via sequence-to-sequence classification

### Results

- >98% accuracy for arrival prediction
- Networks generalizes to other data (sites, source mechanisms)



<https://library.seg.org/doi/pdf/10.1190/segam2019-3215081.1>



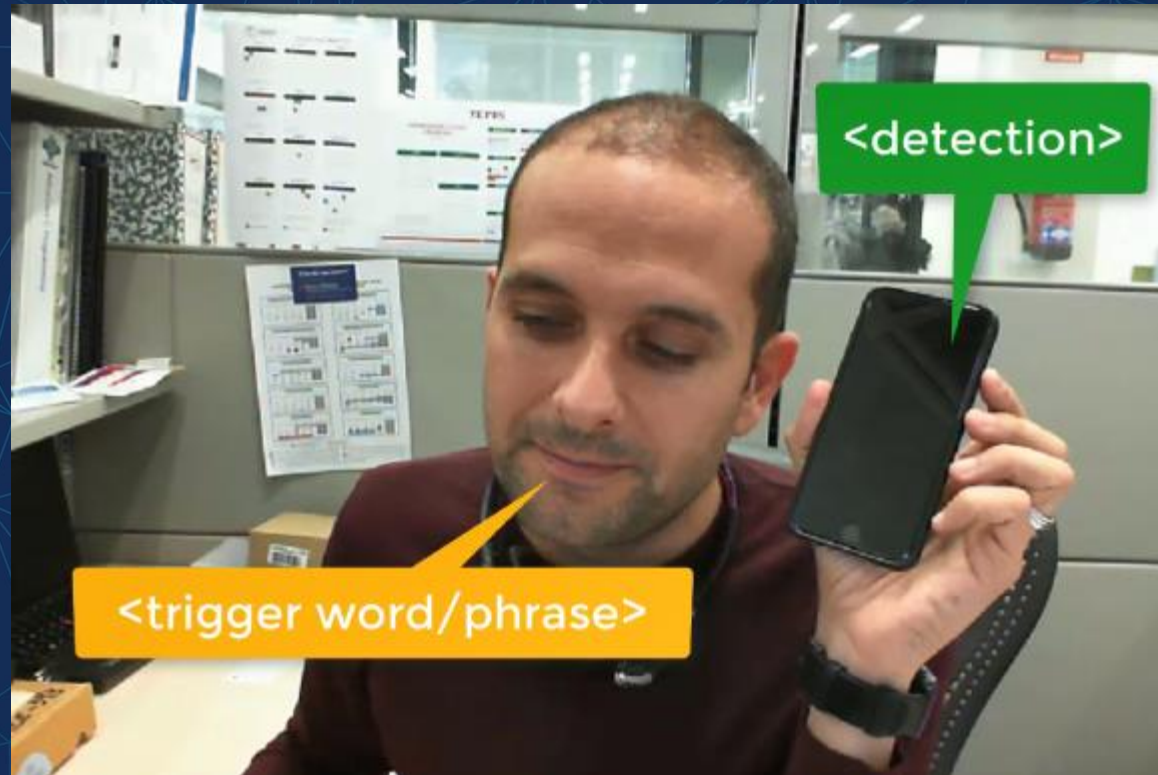
## Voice Interface: The Touchscreen of the Next Century

How AI and Signal Processing Came Together to Track the DNA of Sound

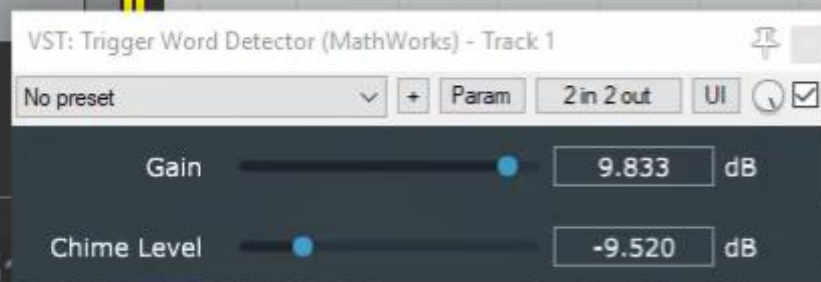
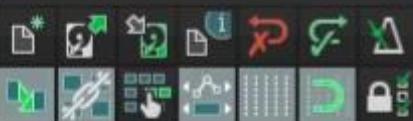
<https://www.mathworks.com/company/mathworks-stories/ai-signal-processing-for-voice-assistants.html>

# A Practical Example: Trigger Word Detection

(The embedded gateway to your cloud-based voice assistant)





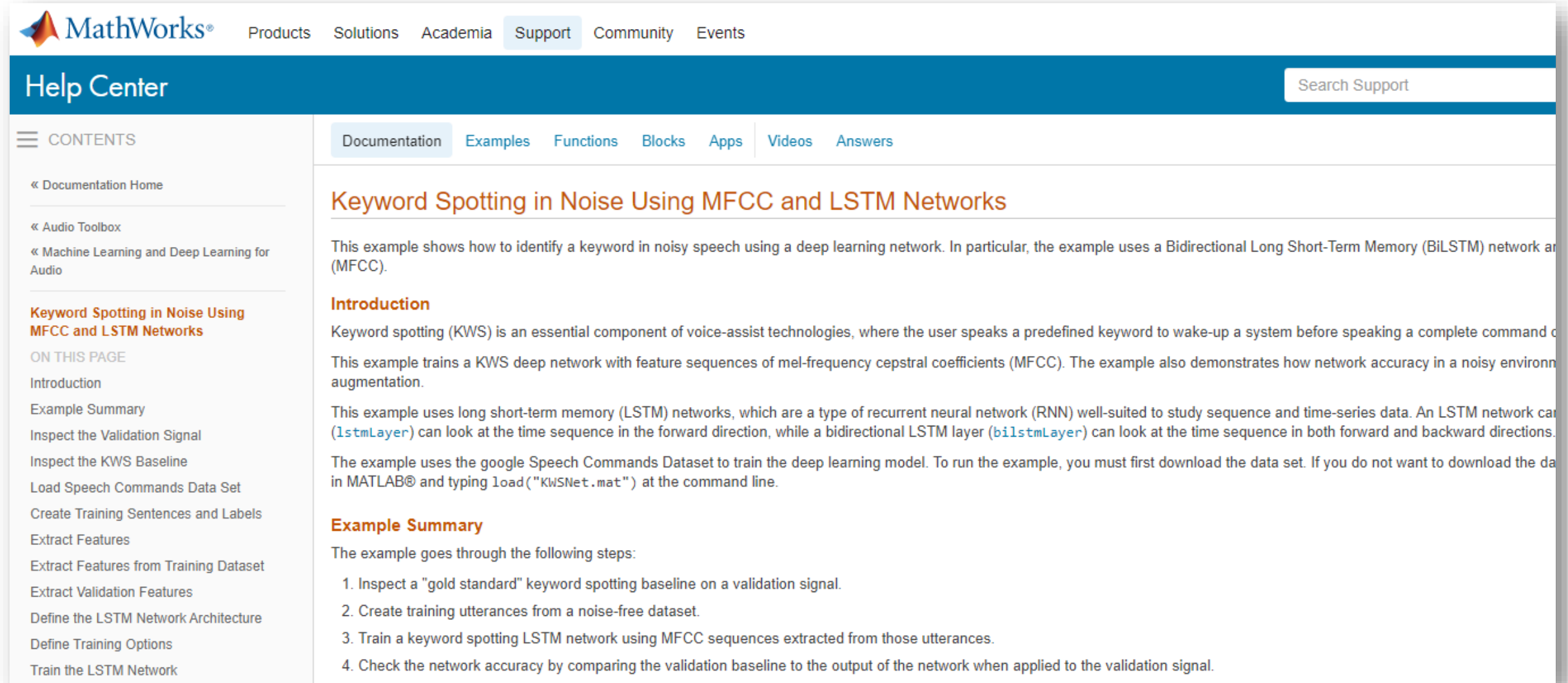


Ding!

Yes!



# Find most of the code for this example online



The screenshot shows the MathWorks Help Center interface. At the top, the MathWorks logo is on the left, and navigation links for Products, Solutions, Academia, Support (highlighted), Community, and Events are on the right. Below the logo is the 'Help Center' header with a search bar labeled 'Search Support'. A left sidebar contains a 'CONTENTS' menu with links like '« Documentation Home', '« Audio Toolbox', and '« Machine Learning and Deep Learning for Audio'. The main content area has tabs for Documentation, Examples, Functions, Blocks, Apps, Videos, and Answers. The 'Examples' tab is active, displaying the title 'Keyword Spotting in Noise Using MFCC and LSTM Networks'. The page content includes an introduction to keyword spotting, a detailed description of the LSTM network architecture used, and a list of four steps for running the example.

MathWorks® Products Solutions Academia Support Community Events

Help Center Search Support

CONTENTS

- « Documentation Home
- « Audio Toolbox
- « Machine Learning and Deep Learning for Audio

**Keyword Spotting in Noise Using MFCC and LSTM Networks**

ON THIS PAGE

- Introduction
- Example Summary
- Inspect the Validation Signal
- Inspect the KWS Baseline
- Load Speech Commands Data Set
- Create Training Sentences and Labels
- Extract Features
- Extract Features from Training Dataset
- Extract Validation Features
- Define the LSTM Network Architecture
- Define Training Options
- Train the LSTM Network

Documentation Examples Functions Blocks Apps Videos Answers

## Keyword Spotting in Noise Using MFCC and LSTM Networks

This example shows how to identify a keyword in noisy speech using a deep learning network. In particular, the example uses a Bidirectional Long Short-Term Memory (BiLSTM) network and Mel-Frequency Cepstral Coefficients (MFCC).

### Introduction

Keyword spotting (KWS) is an essential component of voice-assist technologies, where the user speaks a predefined keyword to wake-up a system before speaking a complete command or phrase.

This example trains a KWS deep network with feature sequences of mel-frequency cepstral coefficients (MFCC). The example also demonstrates how network accuracy in a noisy environment is improved using data augmentation.

This example uses long short-term memory (LSTM) networks, which are a type of recurrent neural network (RNN) well-suited to study sequence and time-series data. An LSTM network can look at the time sequence in the forward direction, while a bidirectional LSTM layer (`biLstmLayer`) can look at the time sequence in both forward and backward directions.

The example uses the google Speech Commands Dataset to train the deep learning model. To run the example, you must first download the data set. If you do not want to download the data set, you can load the data set in MATLAB® and typing `load("KWSNet.mat")` at the command line.

### Example Summary

The example goes through the following steps:

1. Inspect a "gold standard" keyword spotting baseline on a validation signal.
2. Create training utterances from a noise-free dataset.
3. Train a keyword spotting LSTM network using MFCC sequences extracted from those utterances.
4. Check the network accuracy by comparing the validation baseline to the output of the network when applied to the validation signal.

<https://www.mathworks.com/help/audio/examples/keyword-spotting-in-noise-using-mfcc-and-lstm-networks.html>

What does it take to develop an effective real-world deep learning system for signal processing applications?

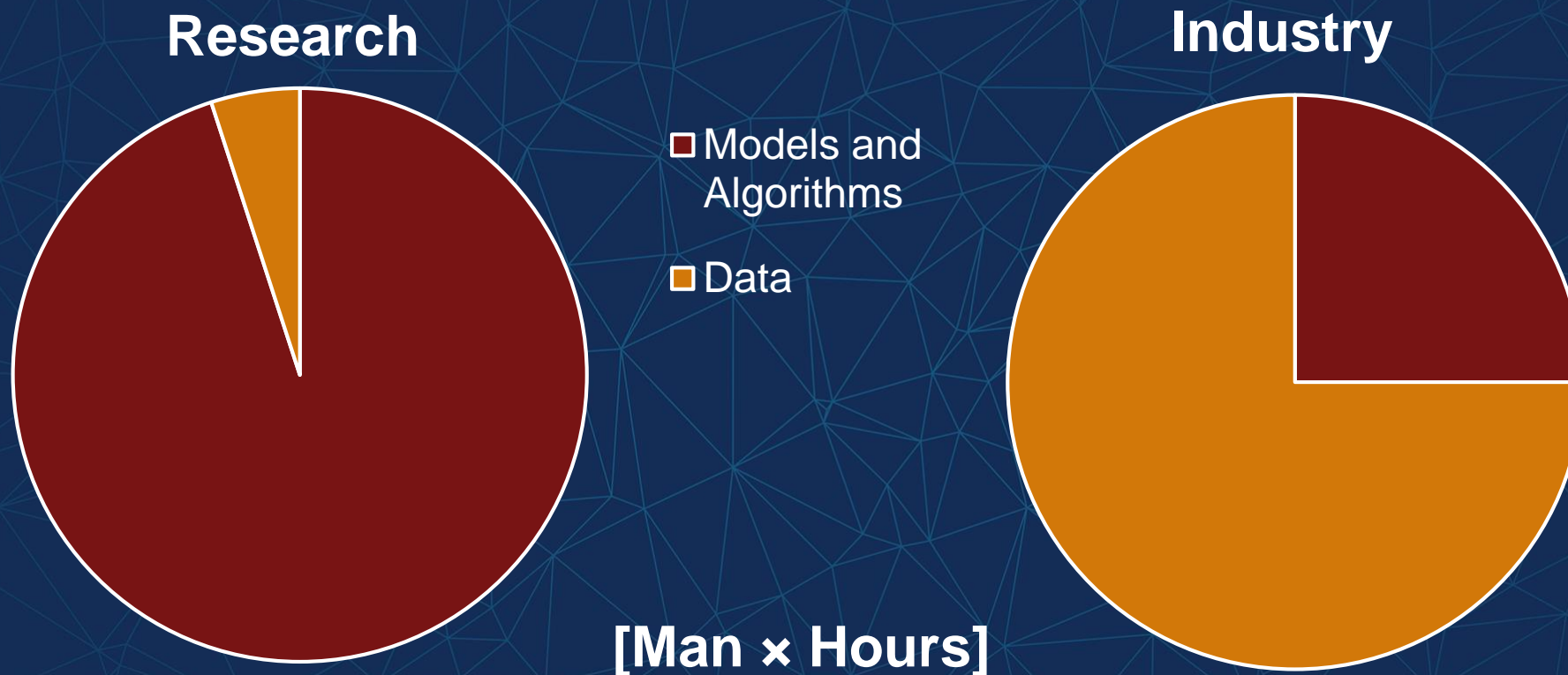


A: "The right deep network design"

"A BiLSTM network with layers of 150 hidden units each, followed by one fully-connected layer and a softmax layer"

A: "A lot of data, a good dose of signal processing expertise, and the right tools for the specific application in hand"

# Data Investments in Deep Learning Research vs. Industry





## CREATE AND ACCESS DATASETS

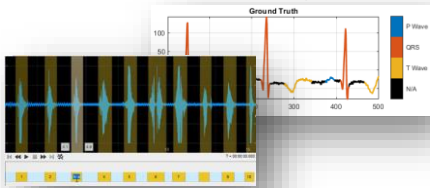
### Data sources



### Simulation and augmentation

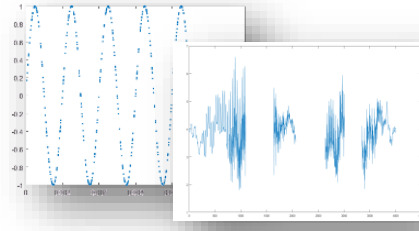


### Data Labeling

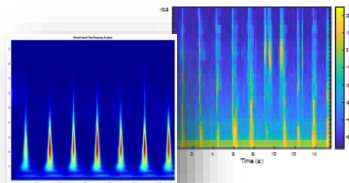


## PREPROCESS AND TRANSFORM DATA

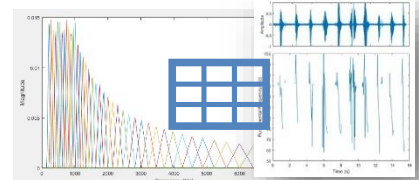
### Pre-Processing



### Transformation



### Feature extraction

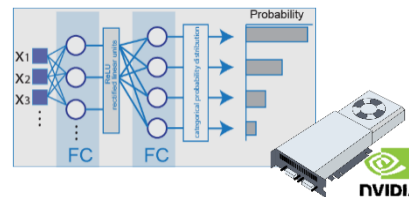


## DEVELOP PREDICTIVE MODELS

### Import Reference Models/ Design from scratch



### Hardware-Accelerated Training

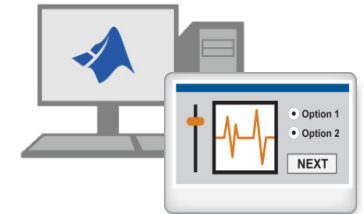


### Analyze and tune hyperparameters



## ACCELERATE AND DEPLOY

### Desktop Apps



### Enterprise Scale Systems

Java  
MATLAB  
C/C++  
Python

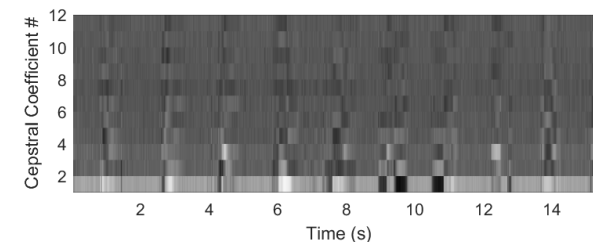
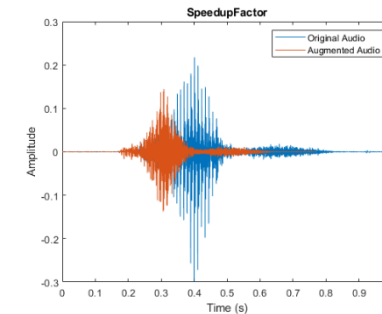
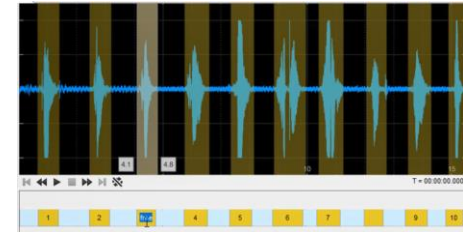
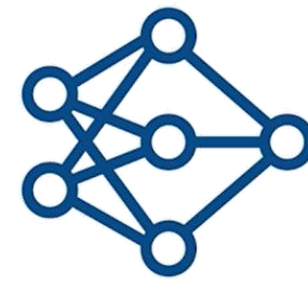
### Embedded Devices and Hardware





# Agenda

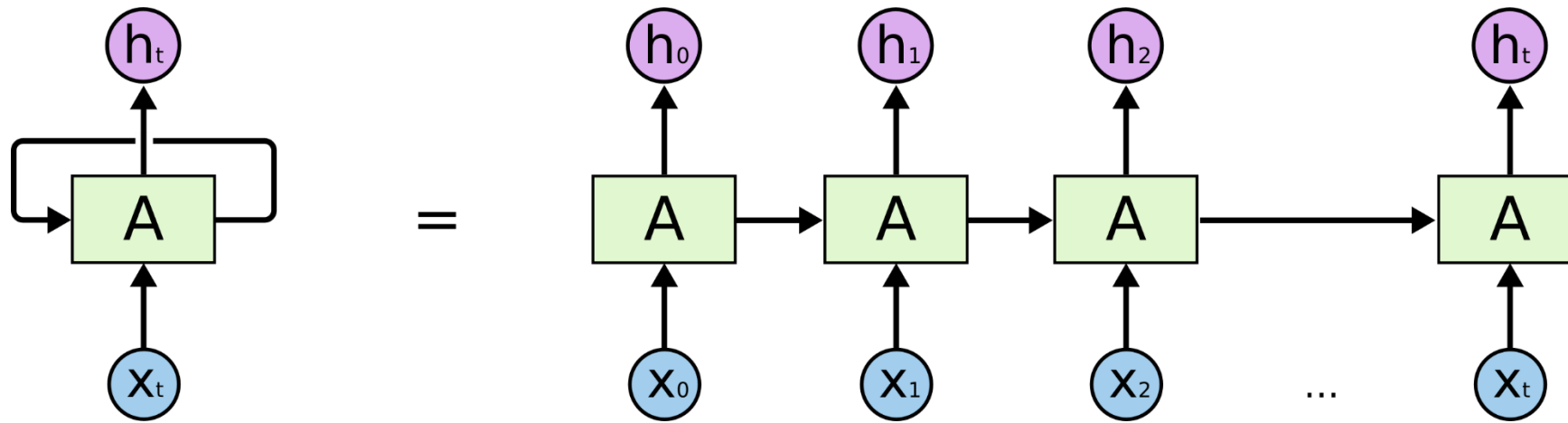
- Basics on training deep neural networks for signals
- Annotating data to train networks for practical applications
- Generating new data – synthesis and augmentation
- Creating inputs for deep networks
- From system models to real-time prototypes



# Defining a deep network architecture

```
layers = [ ...  
    sequenceInputLayer(numFeatures)  
    bilstmLayer(150, "OutputMode", "sequence")  
    bilstmLayer(150, "OutputMode", "sequence")  
    fullyConnectedLayer(2)  
    softmaxLayer  
    classificationLayer  
];
```





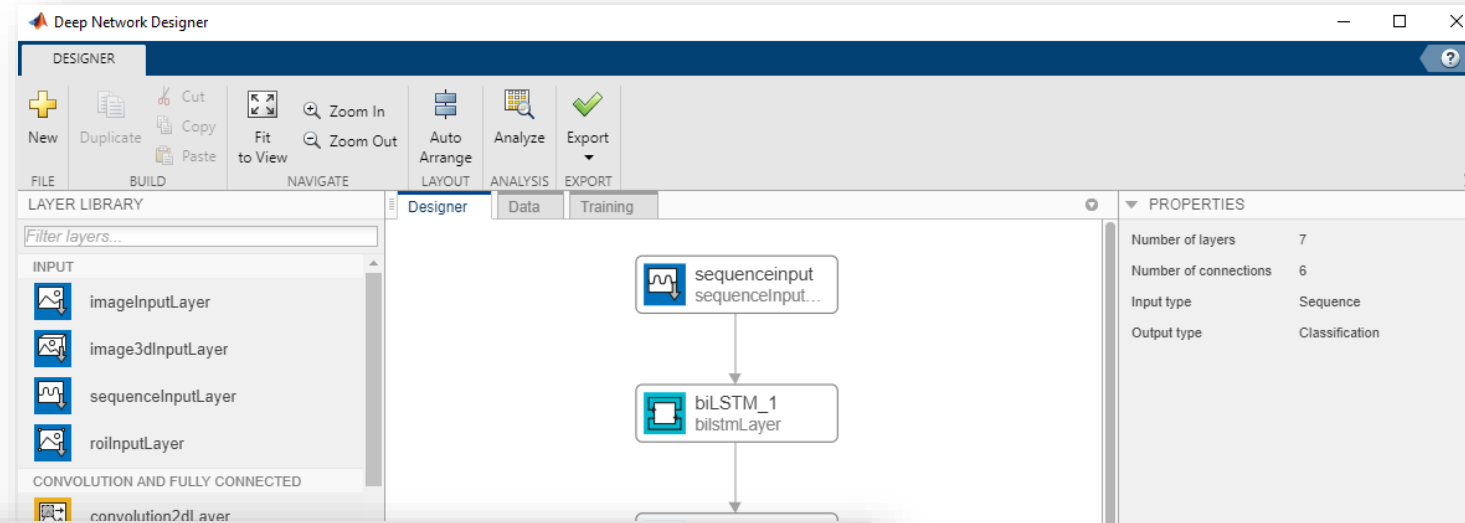
Long Short Term Memory (**LSTM**) Layer

(Recursive Neural Networks, **RNN**)



# Defining a deep network architecture

```
layers = [ ...  
    sequenceInputLayer(numFeatures)  
    biLSTMlayer(150,"OutputMode","sequence")  
    biLSTMlayer(150,"OutputMode","sequence")  
    fullyConnectedLayer(2)  
    softmaxLayer  
    classificationLayer  
];
```



## ANALYSIS RESULT

	Name	Type	Activations	Learnables	Total Learnables
1	sequenceinput Sequence input with 42 dimensions	Sequence Input	42	-	0
2	biLSTM_1 BiLSTM with 150 hidden units	BiLSTM	300	InputWeights 1200×42 RecurrentWeights 1200×150 Bias 1200×1	231600
3	biLSTM_2 BiLSTM with 150 hidden units	BiLSTM	300	InputWeights 1200×300 RecurrentWeights 1200×150 Bias 1200×1	541200
4	fc 2 fully connected layer	Fully Connected	2	Weights 2×300 Bias 2×1	602
5	softmax softmax	Softmax	2	-	0
6	classoutput crossentropyex	Classification Output	-	-	0

# Start from published recipes...

## Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling

Haşim Sak, Andrew Senior, Françoise Beaufays

Google, USA

## Long short-term memory for speaker generalization in supervised speech separation

Jitong Chen<sup>a)</sup> and DeLiang Wang<sup>b)</sup>

Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio 43210, USA

## An Improved Residual LSTM Architecture for Acoustic Modeling

Lu Huang

Department of Electronic Engineering  
Tsinghua University  
Beijing, China  
e-mail: huanglu.th@gmail.com

Jiasong Sun

Department of Electronic Engineering  
Tsinghua University  
Beijing, China  
e-mail: sunjiasong@tsinghua.edu.cn

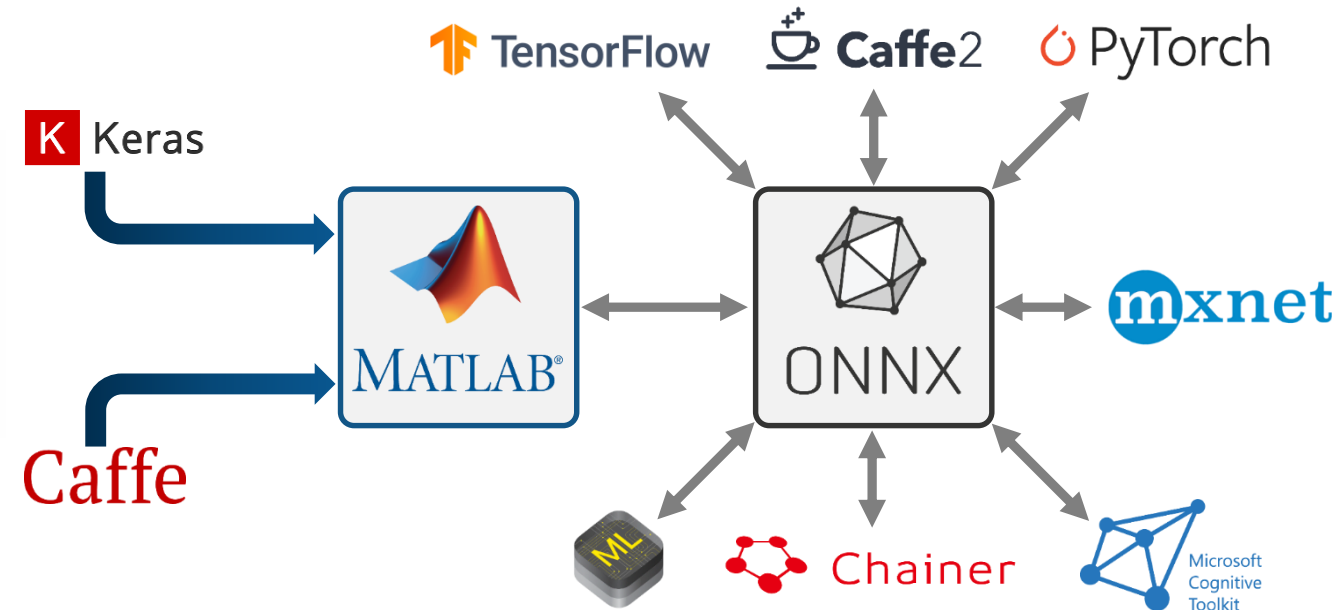
Ji Xu

Department of Speech Acoustics & Content Understanding  
Institute of Acoustics, Chinese Academy of Sciences

Yi Yang

Department of Electronic Engineering  
Tsinghua University

# ...or import models developed by others (including from different frameworks)



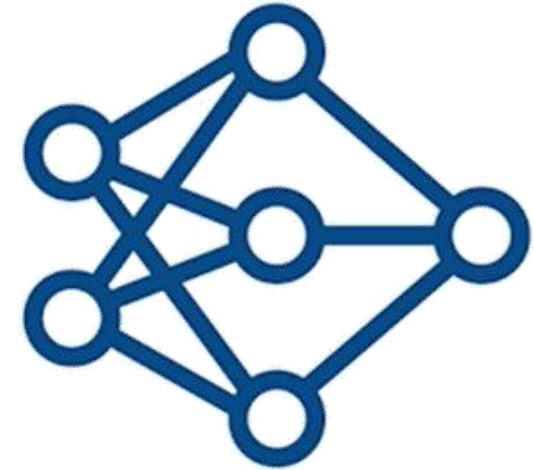
\* Random examples found via web search  
(No endorsement implied)

# Training a deep network

```
layers = [ ...
    sequenceInputLayer(numFeatures)
    bilstmLayer(150,"OutputMode","sequence")
    bilstmLayer(150,"OutputMode","sequence")
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer
];

maxEpochs      = 10;
miniBatchSize = 64;
options = trainingOptions("adam", ...
    "InitialLearnRate",1e-4,...
    "MaxEpochs",maxEpochs, ...
    "MiniBatchSize",miniBatchSize, ...
    "Shuffle","every-epoch",...
    "Verbose",false, ...
    "ValidationFrequency",floor(numel(TrainingFeatures)/miniBatchSize),...
    "ValidationData",{FeaturesValidationClean.',BaselineV},...
    "Plots","training-progress",...
    "LearnRateSchedule","piecewise",...
    "LearnRateDropFactor",0.1, ...
    "LearnRateDropPeriod",5);

[net,info] = trainNetwork(TrainingFeatures,TrainingMasks,layers,options);
```





MATLAB R2019b

HOME PLOTS APPS EDITOR PUBLISH VIEW

Search Documentation Andy

home matlab Documents AudioWebinar Code

Workspace

Name	Value
expectedNumPartitions	128
klstm	4
kovlp	4
loadFeatures	1
LSTMSize	150
LSTMSizes	[75,100,125,150]
LSTMSizes	[75,100,125,150]
M	1x4 cell
net	1x1 SeriesNetwork
netLayers	6x1 Layer

Current Folder Command Window

fx >>

Editor - TrainSingleNetwork.m

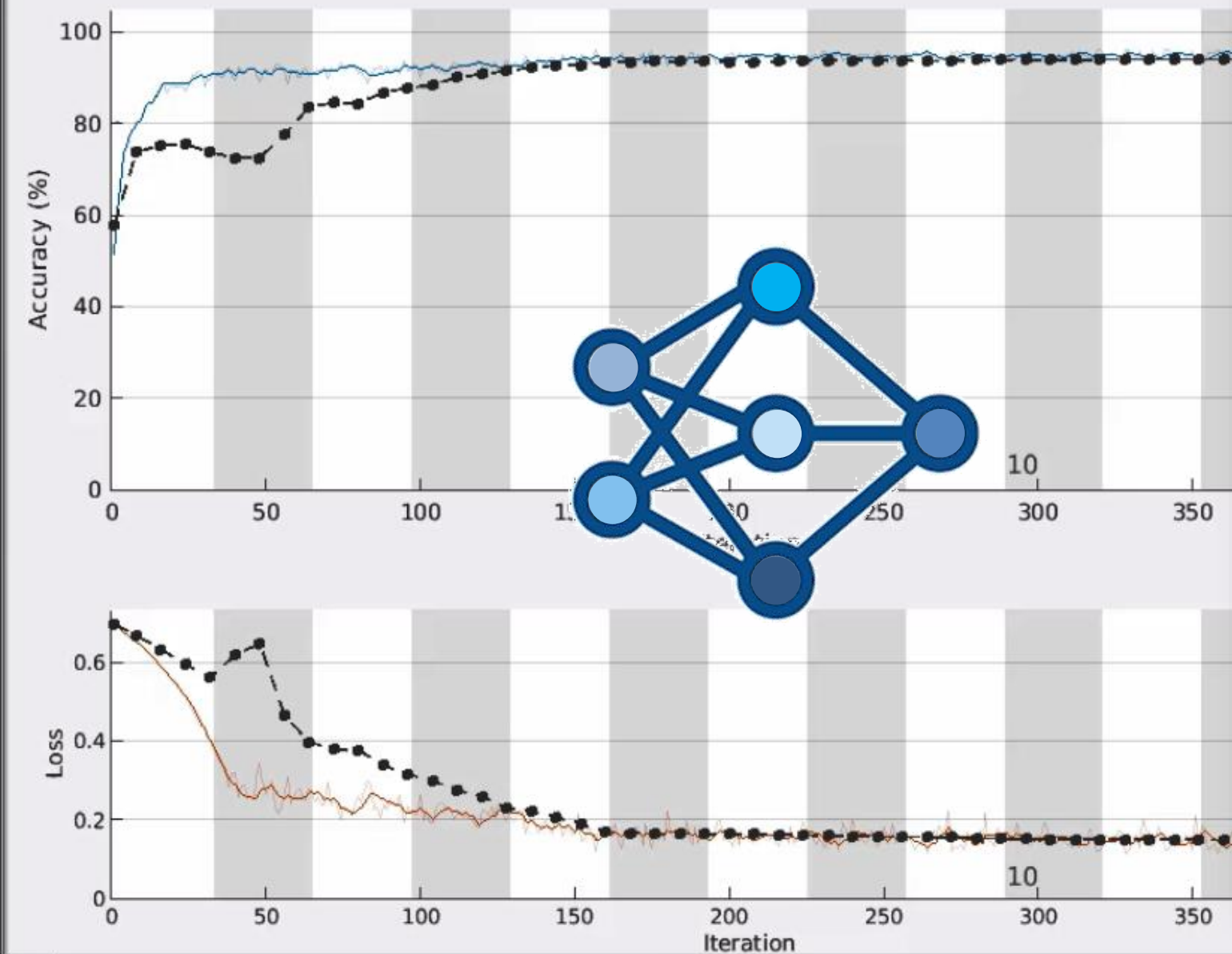
Variables - feat

```
TrainSingleNetwork.m
40 - netLayers = [ ...
41     sequenceInputLayer(numFeatures)
42     bilstmLayer(LSTMSizes(klstm),"OutputMode","sequence")
43     bilstmLayer(LSTMSizes(klstm),"OutputMode","sequence")
44     fullyConnectedLayer(2)
45     softmaxLayer
46     classificationLayer
47 ];
48
49 - trainOptions = trainingOptions("adam", .I.
50     "InitialLearnRate",1e-4, ...
51     "MaxEpochs",12, ...
52     "MiniBatchSize",4, ...
53     "Shuffle","every-epoch", ...
54     "Verbose",false, ...
55     "ValidationFrequency",8, ...
56     "ValidationData",{ValidationFeatures{kovlp},ValidationMasks{kovlp}}, ...
57     "Plots","training-progress", ...
58     "LearnRateSchedule","piecewise", ...
59     "LearnRateDropFactor",0.1, ...
60     "LearnRateDropPeriod",5,...
61     "SequenceLength","Shortest");
62
63 %% Network training
64
65 - tic;
66 - net = trainNetwork(trainingFeatures,trainingMasks,netLayers,trainOptions);
67 - fprintf('Training the network took %g s\n',toc);
68
69
```

script Ln 63 Col 1

Training Progress (20-Mar-2020 12:21:25)

## Training Progress (20-Mar-2020 12:21:25)



## Results

Validation accuracy: 93.96%

Training finished: Reached final iteration

## Training Time

Start time: 20-Mar-2020 12:21:25

Elapsed time: 8 min 15 sec

## Training Cycle

Epoch: 12 of 12

Iteration: 384 of 384

Iterations per epoch: 32

Maximum iterations: 384

## Validation

Frequency: 8 iterations

Patience: Inf

## Other Information

Hardware resource: Single GPU

Learning rate schedule: Piecewise

## Accuracy

— Training (smoothed)

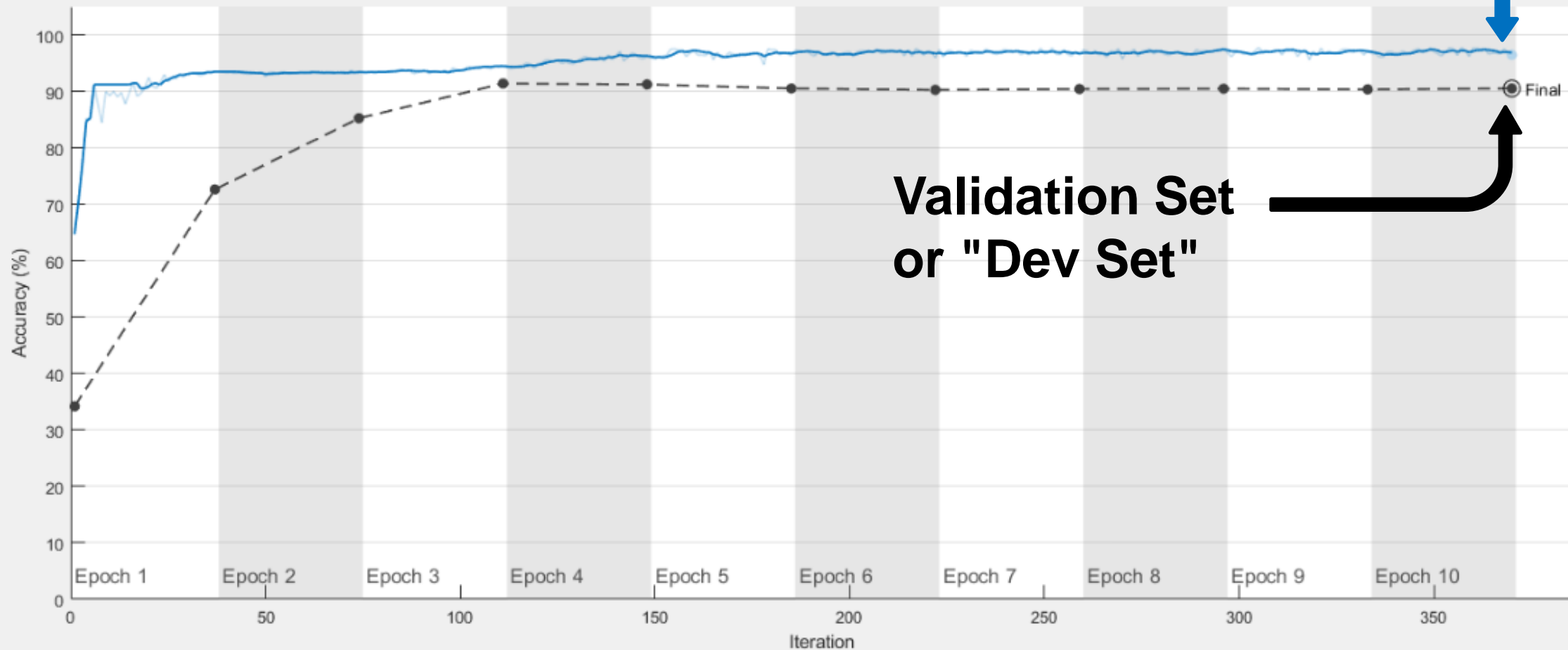
—•— Training

- -•- - Validation

400

- -•- - Validation

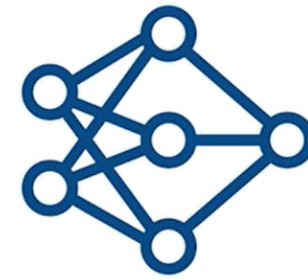
Training Set



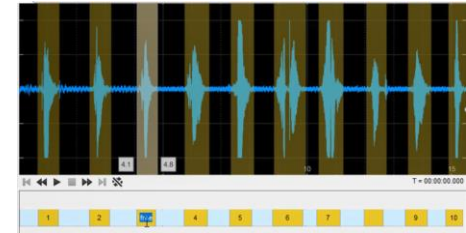


# Agenda

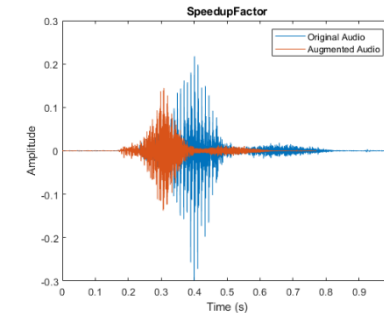
- Basics on training deep neural networks for signals



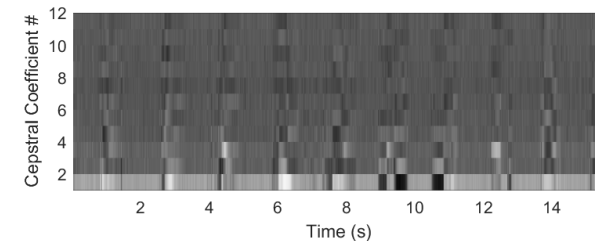
- Annotating data to train networks for practical applications



- Generating new data – synthesis and augmentation



- Creating inputs for deep networks



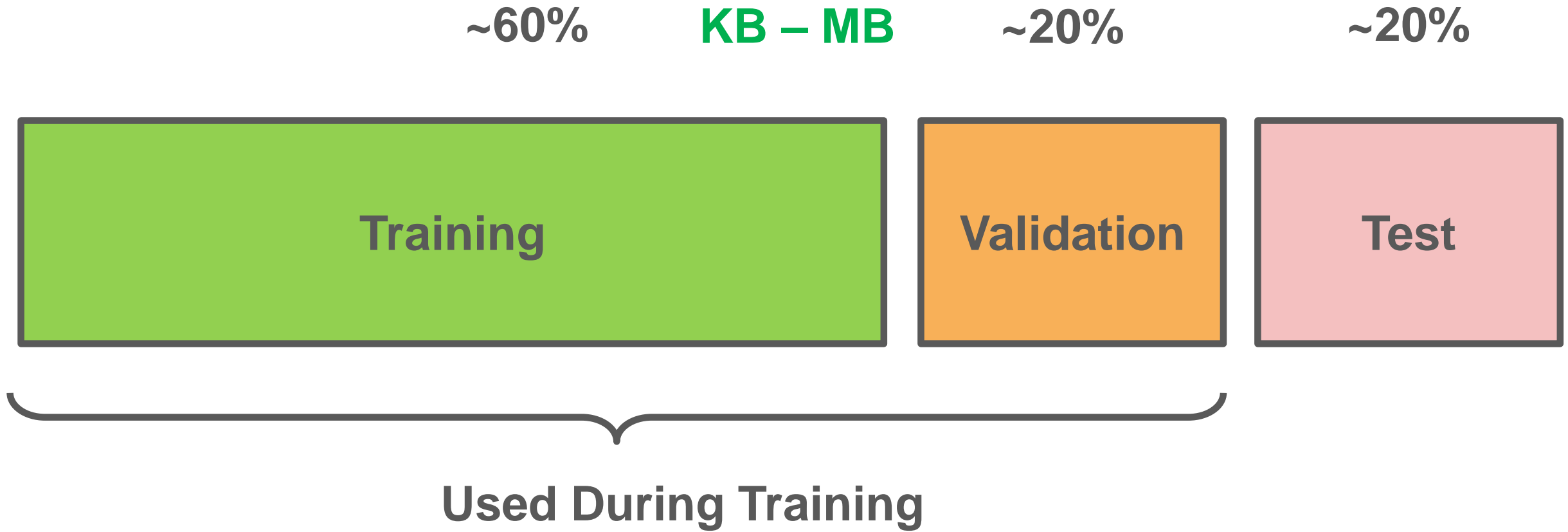
- From system models to real-time prototypes



# Training, Validation, and Test Data

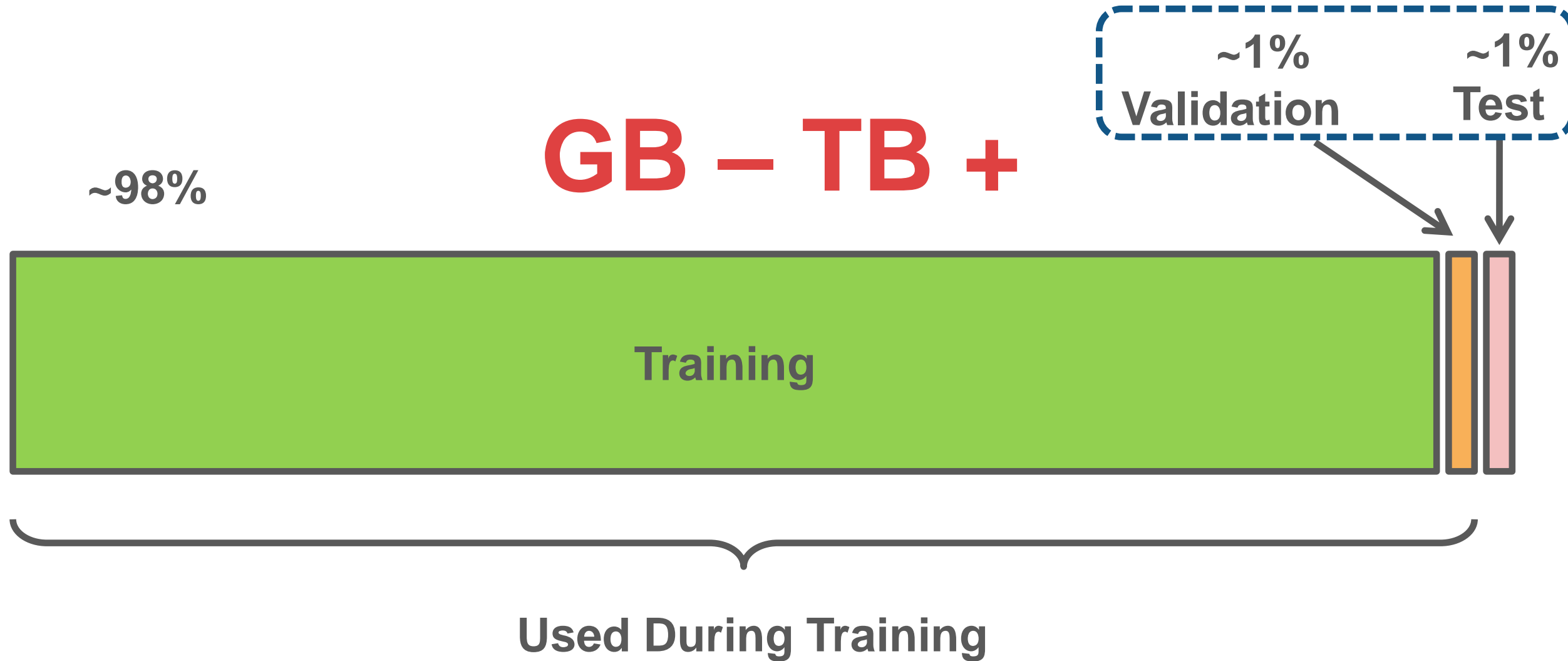
Your full dataset (All of your **data + labels**)

# Training, Validation, and Test Data





# Training, Validation, and Test Data



# A good validation data sample – Realistic recording, accurately labeled



# How to label new non-annotated data?

Use an intelligent system trained to carry out a similar tasks with proven accuracy!

For example:

- Humans



Audio Labeler - ExplainingDetectionRequirements-16-mono.wav

LABELRECORD

LoadSaveImport

FILE

Audio Player:  
Primary Sou...

Settings

DEVICE

Default LayoutLegend

VIEW

Speech DetectorSpeech to Text

AUTOMATION

Export

EXPORT

Cleanup

Data Browser

▼ Audio Files

KeywordSpeech-16-16-mono-34secs.flac

ExplainingDetectionRequirements-16-mono.wav

▼ Audio File Info

ExplainingDetectionRequirements-16-

Channels: 1

Sample Rate: 16000 Hz

Duration: 39.260 s

Compression: Uncompressed

Bit Depth: 16 bits/sample

Location: C:\Docs\Material\Proj

ExplainingDetectionRequirements-16-mono.wav

File Labels

To label an audio file, you must first import or add a file label definition.

ROI Labels

SpeechContent

7.27.47.52177.67.888.28.48.68.877

7.5217

Yes

Ready

Samples Underrun = 0

# How to label new non-annotated data?

Use an intelligent system trained to carry out a similar tasks with proven accuracy!

For example:

- Humans
- Pre-trained machine learning models

LABEL

RECORD

SPEECH TO TEXT

Service Name Google

Options

Segment Words ☒


PARAMETERS


Label SpeechContent


Type ROI - String

Data All audio files

SELECTION

  
Run

  
Undo

  
Close

AUTOMATE

CLOSE

Help

Speech to Text

Data Browser

▼ Audio Files

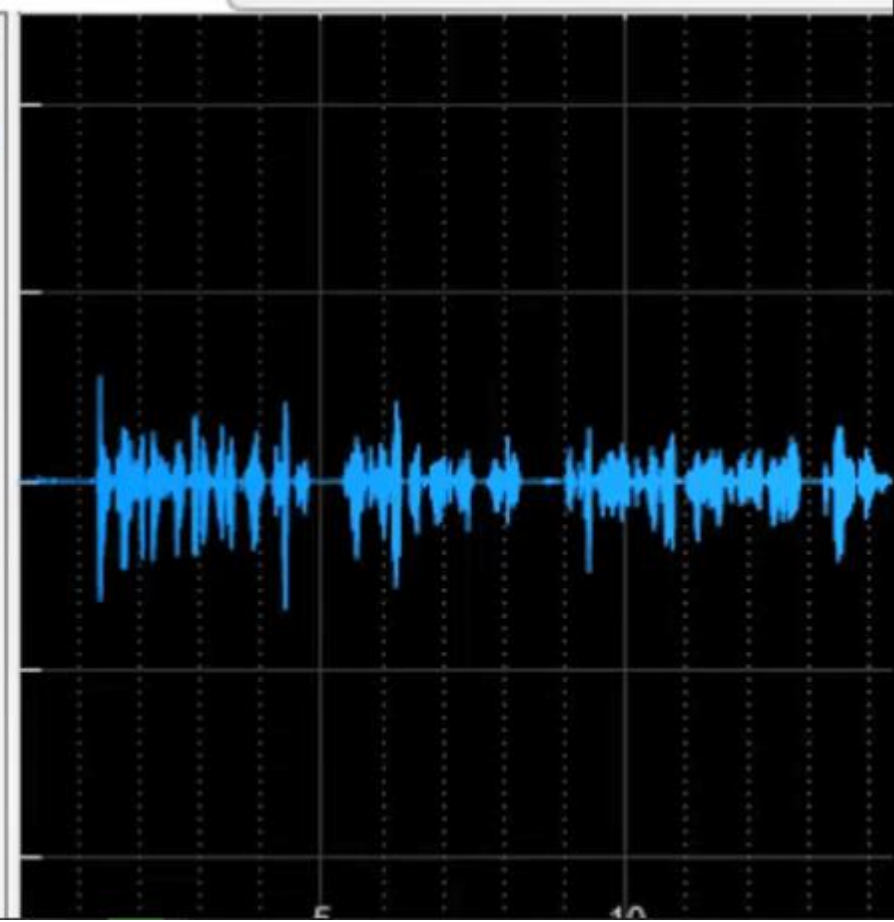
KeywordSpeech-16-16-mono-34secs.flac

ExplainingDetectionRequirements-16-mono.wav

ExplainingDetectionRequirements-16-mono.wav

File Labels

To label an audio file, you must first import or add a file label definition.





LABEL

RECORD

SPEECH TO TEXT

Service Name Google

Options

Segment Words ☒


PARAMETERS


Label SpeechContent


Type ROI - String

Data All audio files

SELECTION

 Run


 Undo Run

 Close Speech to Text

AUTOMATE

CLOSE

Cleanup



Data Browser

ExplainingDetectionRequirements-16-mono.wav

▼ Audio Files

KeywordSpeech-16-16-mono-34secs.flac

ExplainingDetectionRequirements-16-mono.wav

▼ Audio File Info

ExplainingDetectionRequirements-16-

Channels: 1

Sample Rate: 16000 Hz

Duration: 39.260 s

Compression: Uncompressed

Bit Depth: 16 bits/sample

Location: C:\Docs\Material\Proj


File Labels

To label an audio file, you must first import or add a file label definition.


ROI Labels

SpeechContent

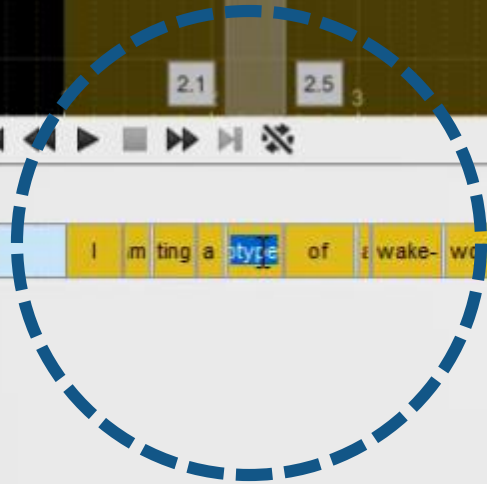
I m ting a type of wake- word detector train t w up w i hea th woi yes fd lik m system t wa



2.1 2.5 3 4 5 6 7 8 9 10



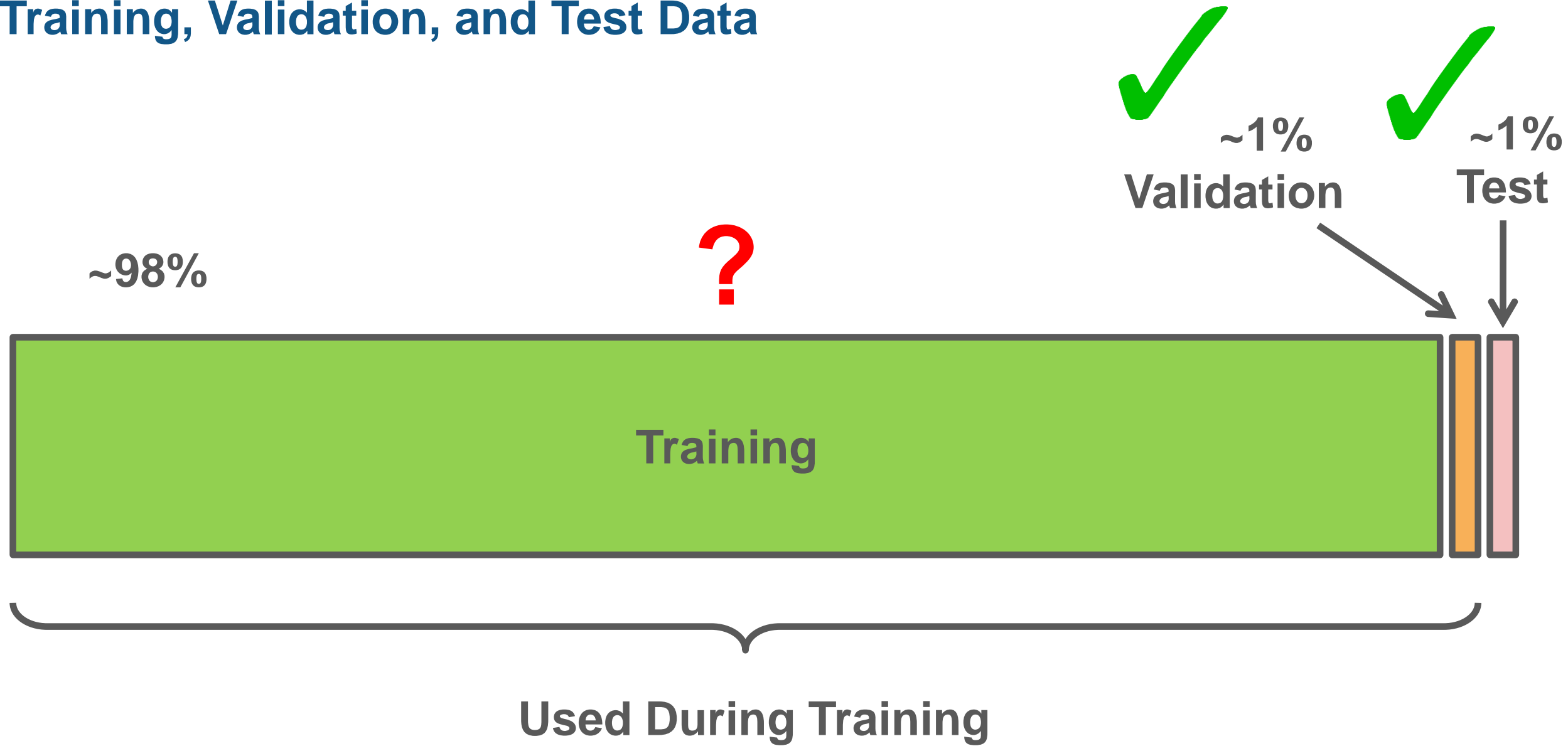
T = 00:00:00.000



Samples Underrun = 0

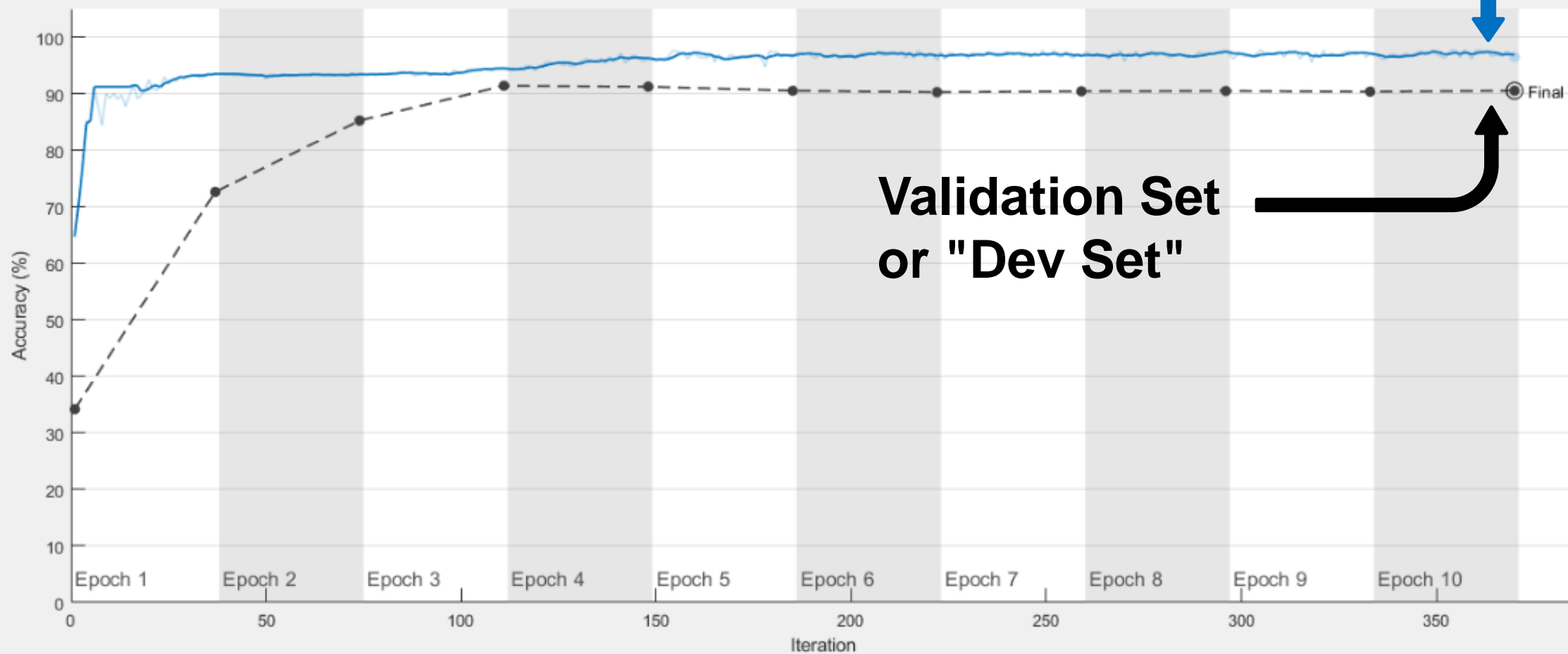


# Training, Validation, and Test Data



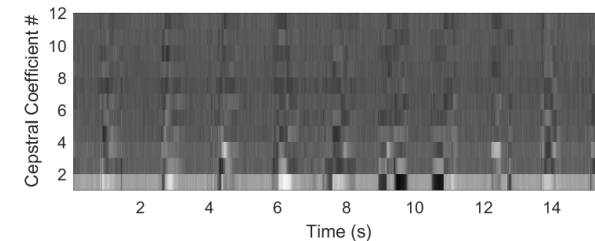
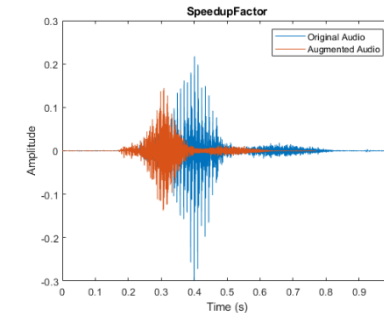
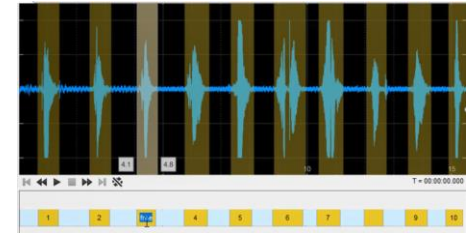
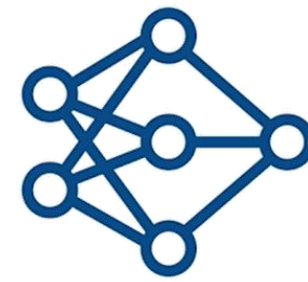


Training Set



# Agenda

- Basics on training deep neural networks for signals
- Annotating data to train networks for practical applications
- Generating new data – synthesis and augmentation
- Creating inputs for deep networks
- From system models to real-time prototypes

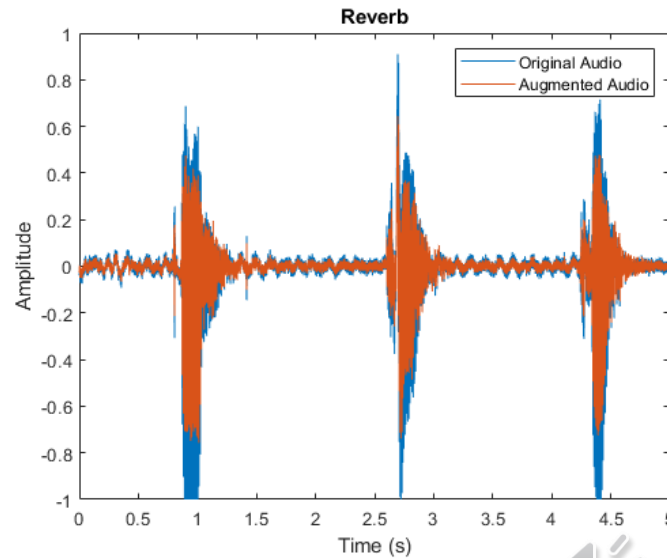


# Augmentation – Application-Specific Effects

```
>> auAugm.AugmentationInfo  
ans =  
    struct with fields:
```



Reverb: 1

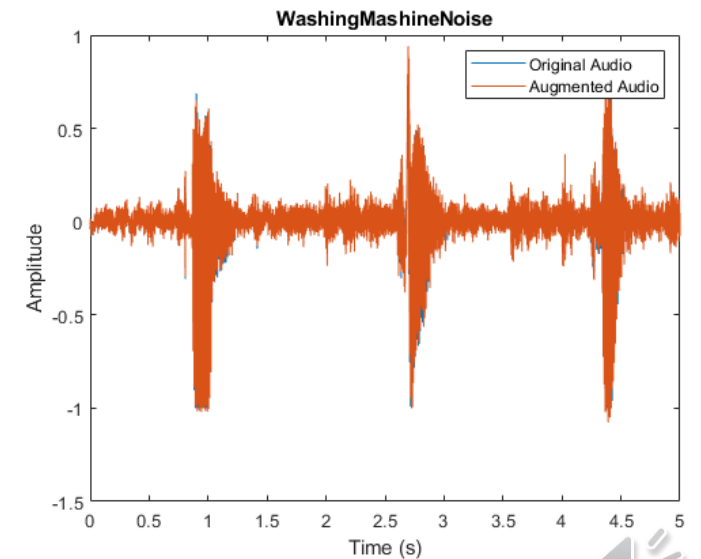


Add Kitchen  
Reverberation



```
>> auAugm.AugmentationInfo  
ans =  
    struct with fields:
```

WashingMachineNoise: 9



Add Washing  
Machine Noise

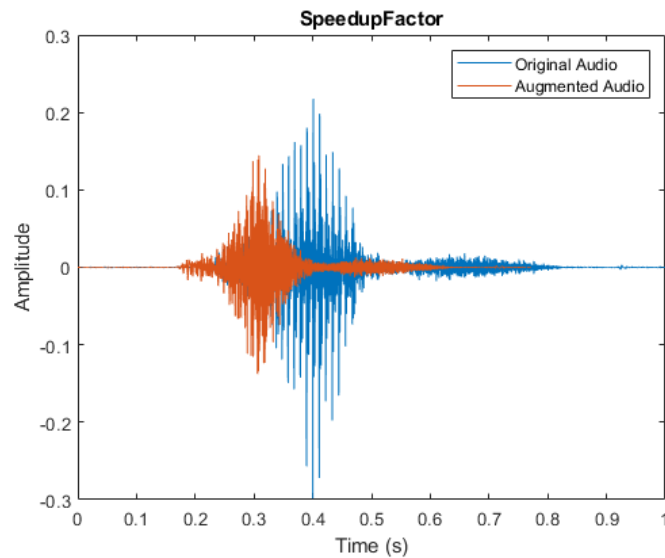


# Augmentation – Common Effective Speech Effects

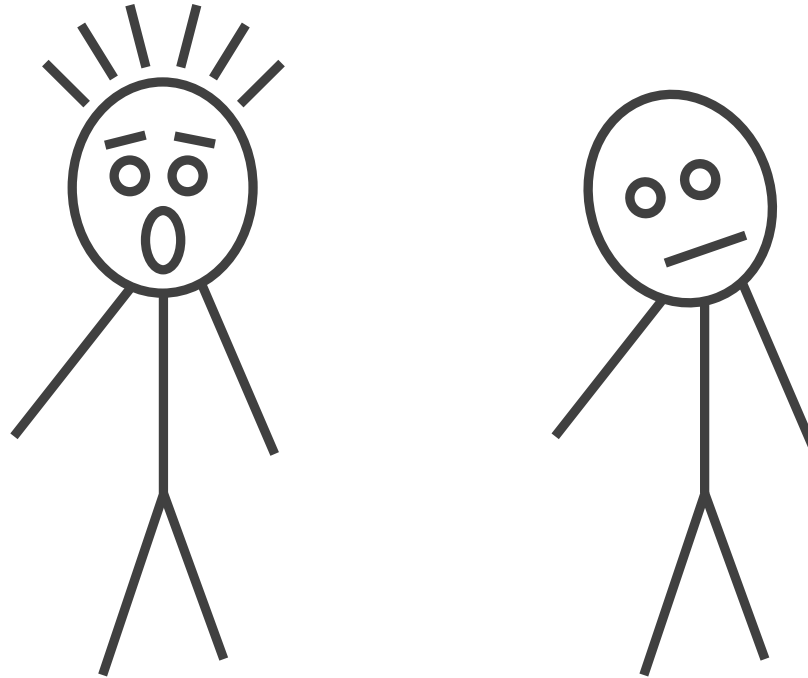


```
>> data.AugmentationInfo(1)
```

```
ans = struct with fields:  
SpeedupFactor: 1.3
```



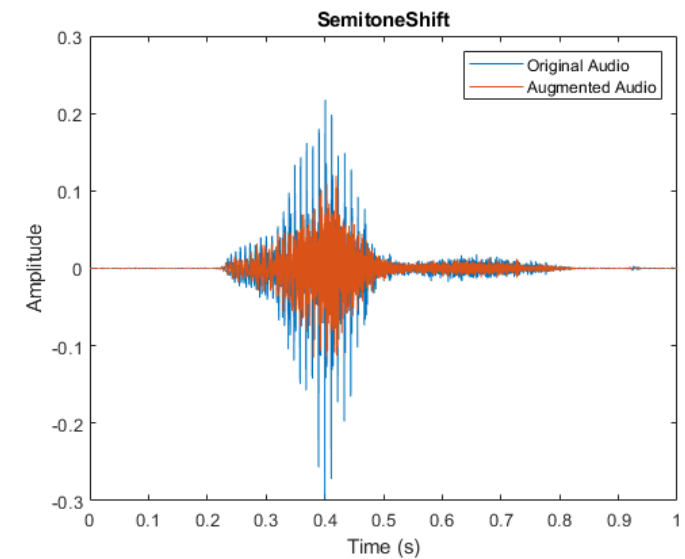
Time  
Stretching



Learn more on [audioDataAugmenter](#)

```
>> data.AugmentationInfo(2)
```

```
ans = struct with fields:  
SemitoneShift: -2
```



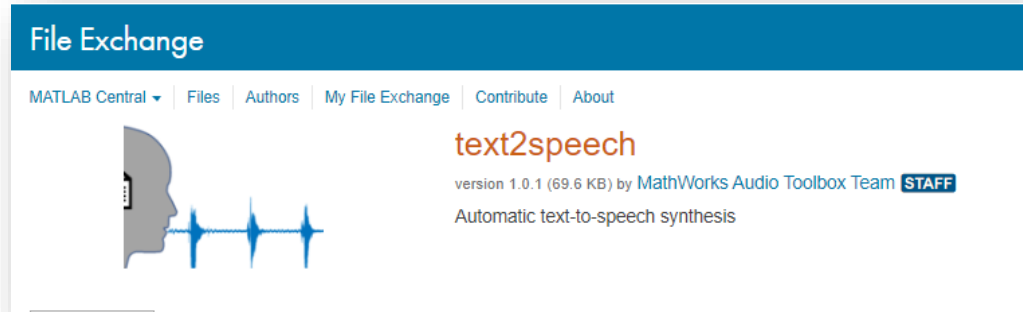
Pitch  
Shifting





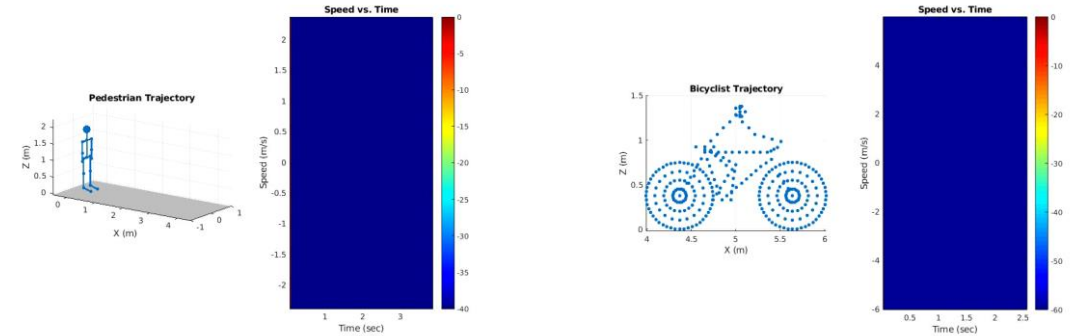
# Synthesis – Generative AI models or domain-specific simulations

## New text2speech function



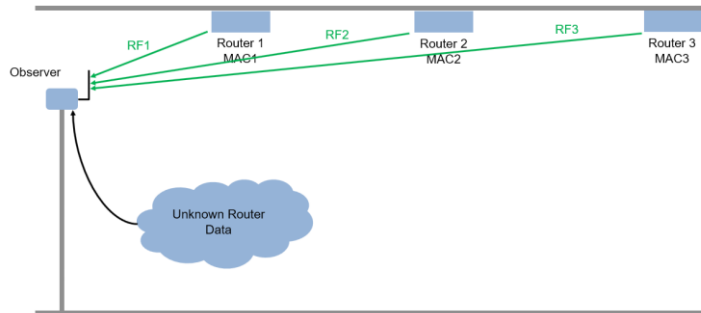
<https://www.mathworks.com/matlabcentral/fileexchange/73326-text2speech>

## Pedestrian and Bicyclist (Radar) Classification



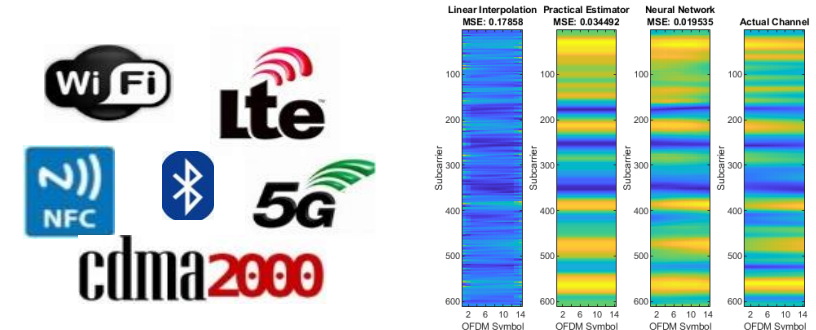
<https://www.mathworks.com/help/phased/examples/pedestrian-and-bicyclist-classification-using-deep-learning.html>

## WLAN Router Impersonation Detection



<https://www.mathworks.com/help/comm/examples/design-a-deep-neural-network-with-simulated-data-to-detect-wlan-router-impersonation.html>

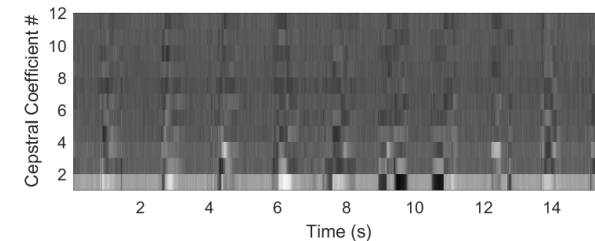
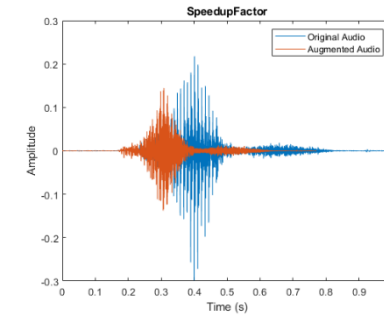
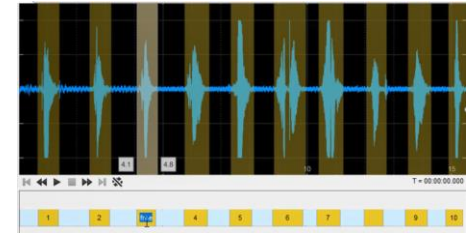
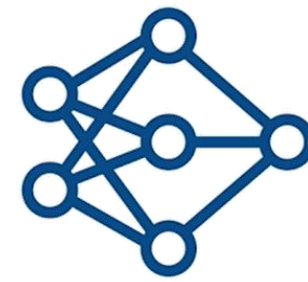
## 5G Channel Estimation



<https://www.mathworks.com/help/5g/examples/deep-learning-data-synthesis-for-5g-channel-estimation.html>

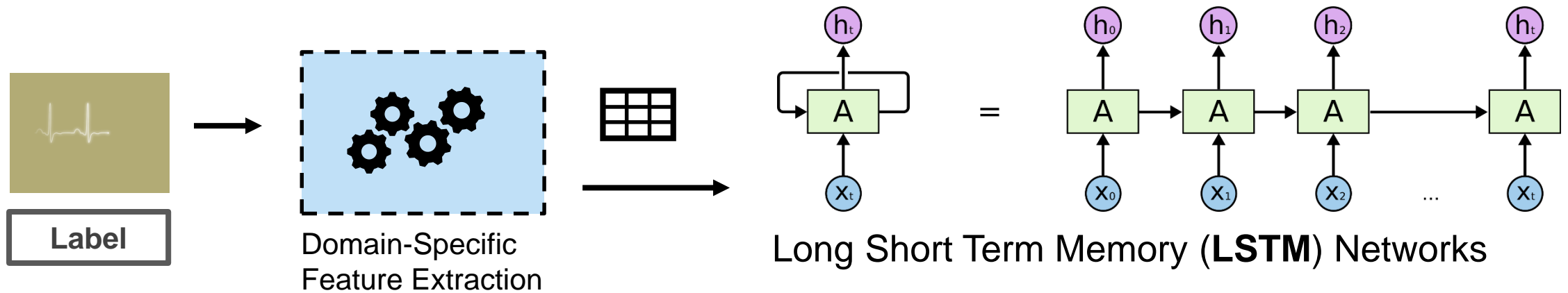
# Agenda

- Basics on training deep neural networks for signals
- Annotating data to train networks for practical applications
- Generating new data – synthesis and augmentation
  - Creating inputs for deep networks
- From system models to real-time prototypes

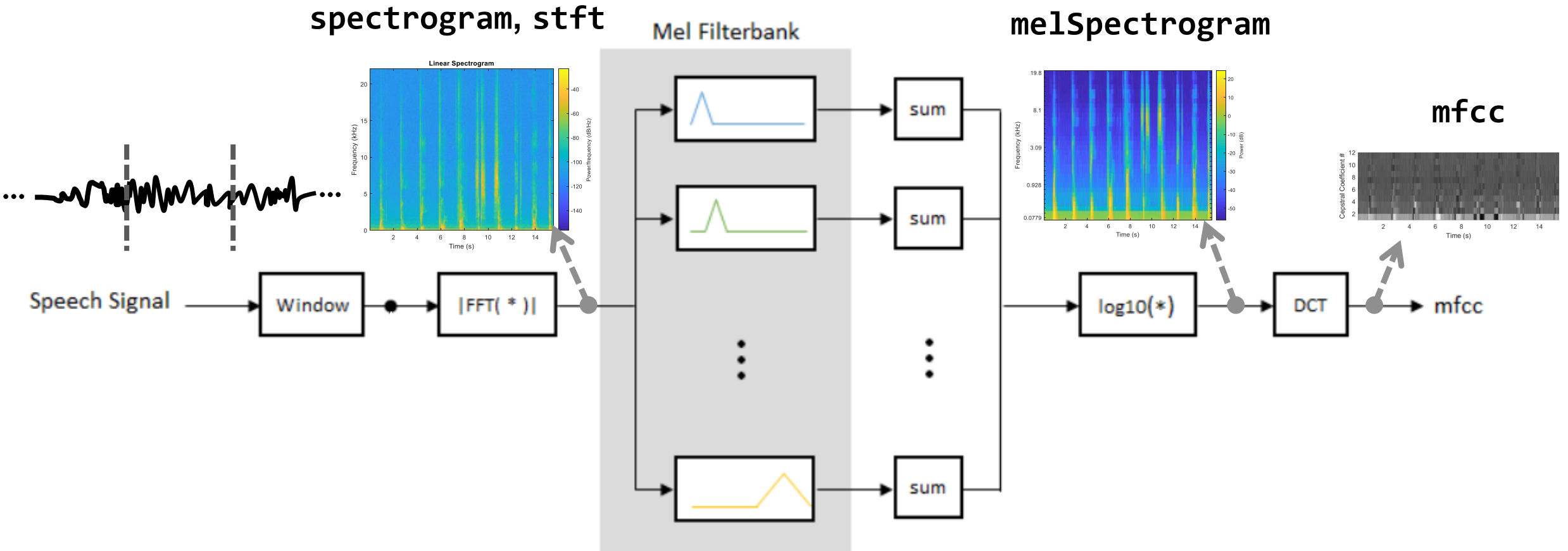


# Training deep networks with time-domain signals most often requires extracting features

Deep learning  $\neq$  End-to-end learning



# Different applications require different feature extraction techniques

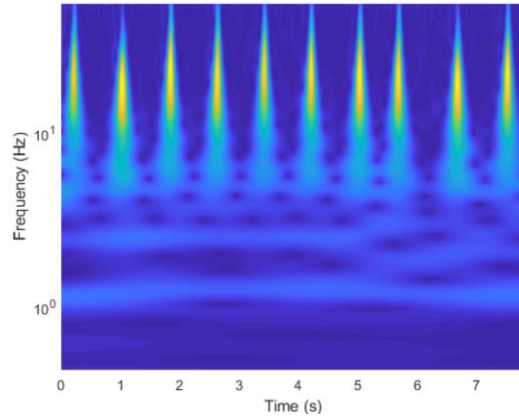




# Many other time-frequency transforms and signal features

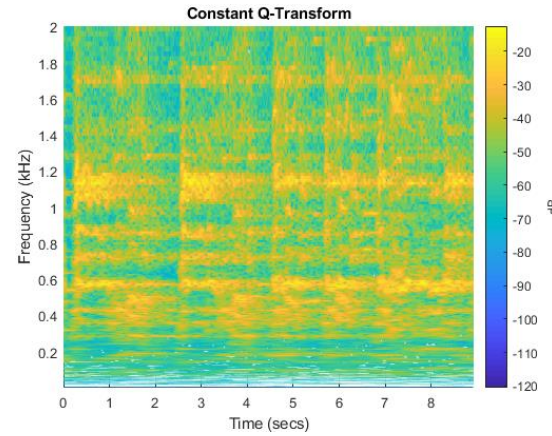
**cwt**

(Continuous wavelet transform)



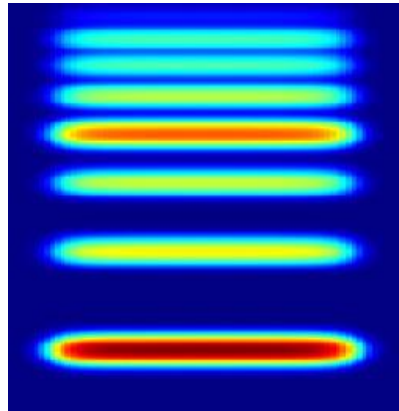
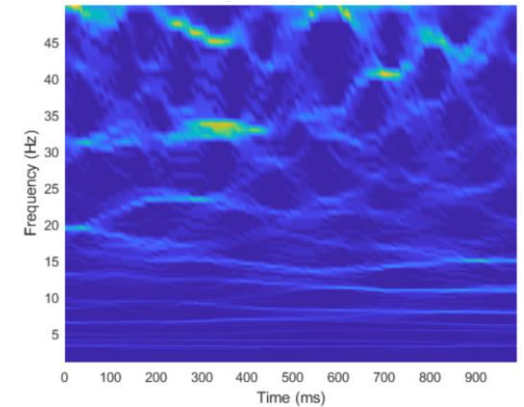
**cqt**

(Constant Q transform)

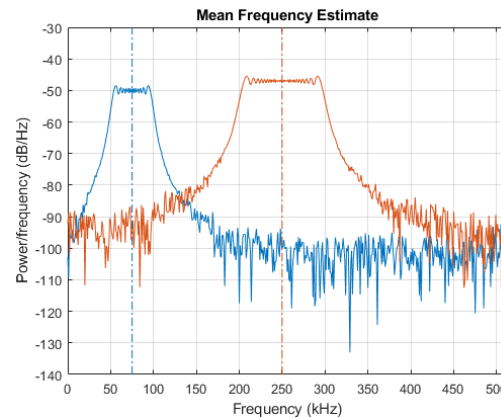


**wsstridge**

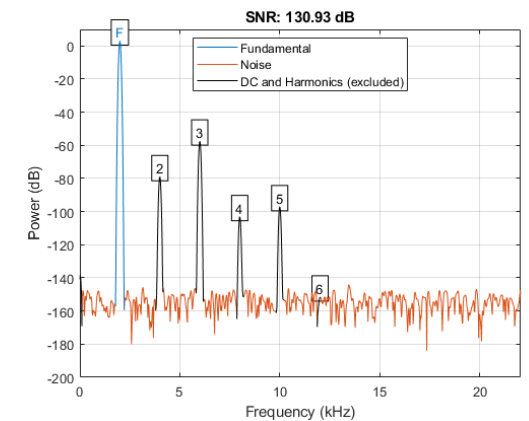
(Synchrosqueezing)



**waveletScattering**

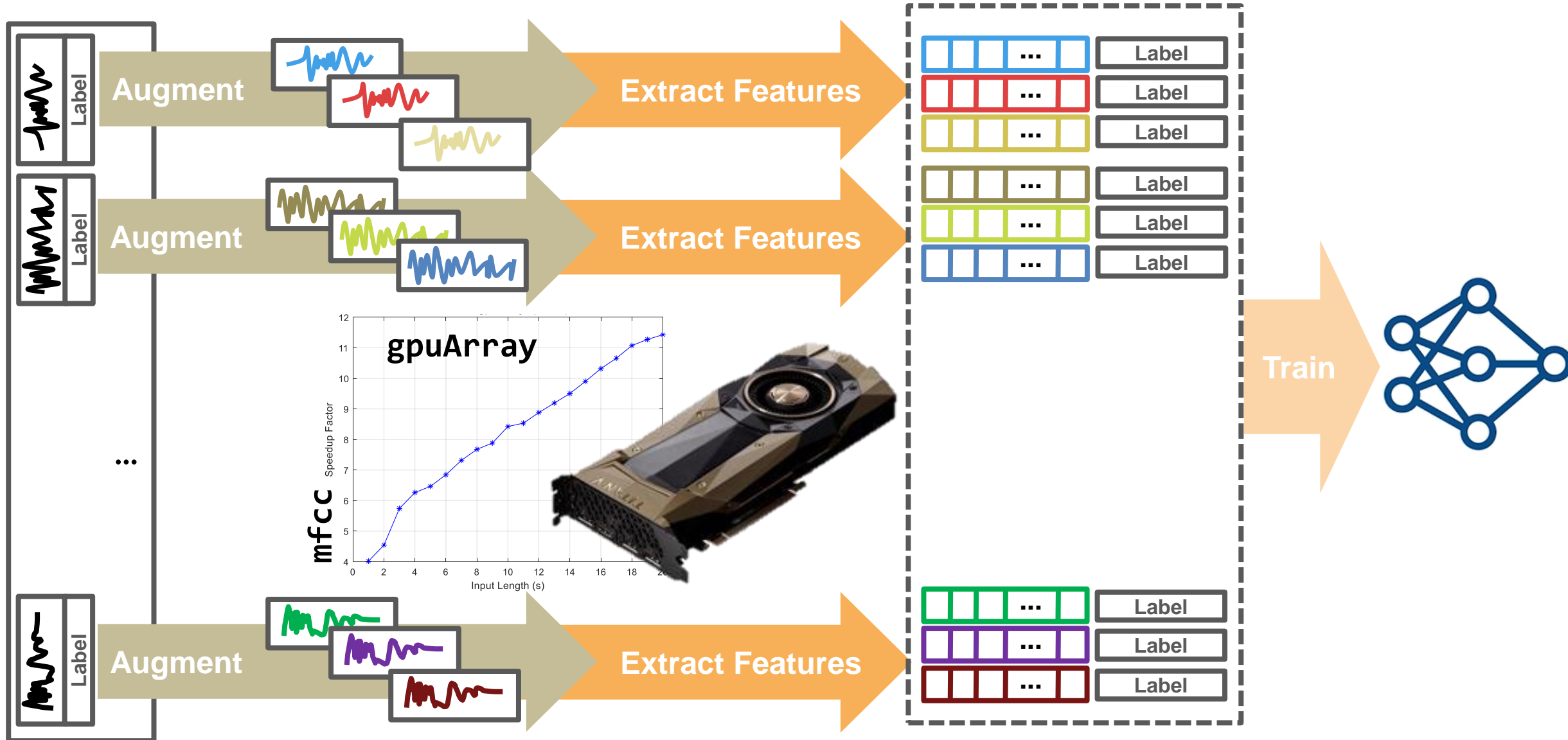


(Spectral statistics)

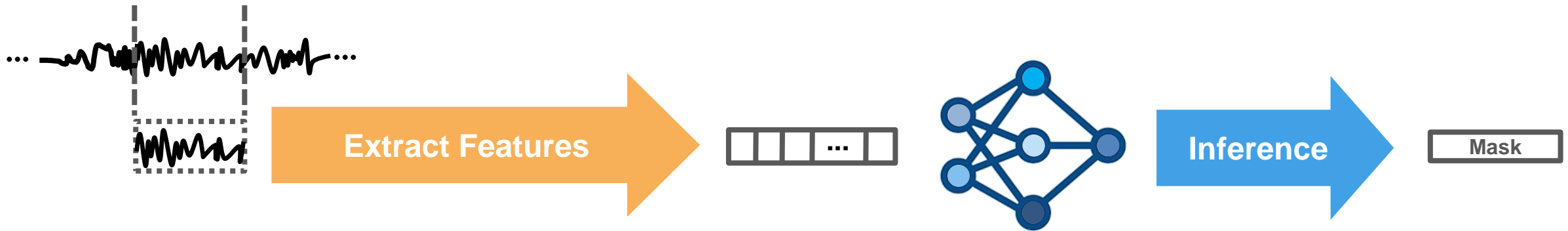


(Harmonic analysis)

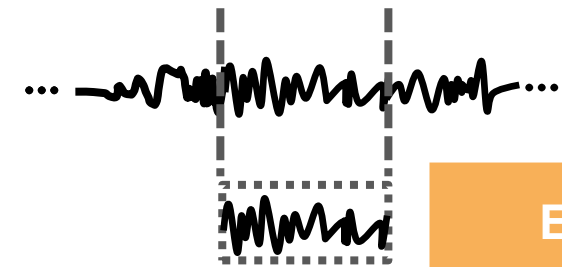
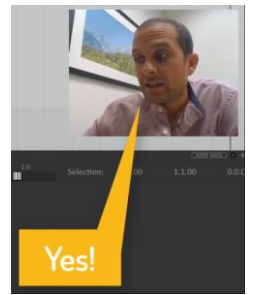
# Providing Input Data for Network Training



# Using Network for Prediction (aka Inference)



# Using Network for Prediction (aka Inference)

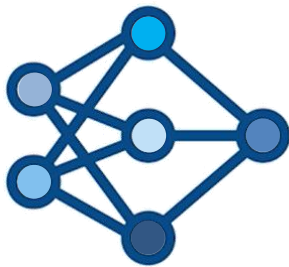


Extract Features

[...]

```
% Extract MFCC from whole analysis buffer
[coeffs,delta,deltaDelta] = mfcc(buf,SampleRate,...
    'WindowLength',winLength,...
    'OverlapLength',ovlpLength);
```

```
% Concatenate and normalize features
featureMatrix = [coeffs,delta,deltaDelta];
featureMatrix = (featureMatrix - M)./S;
```



Inference

```
% Detect keyword with LSTM network (Mask around speech keyword)
featMask = classify(net,featureMatrix.');
```

```
% Debounce and re-align detections in time domain
[timeMask, chimePosition] = debounceAnalyzeDetectionMask(featMask);

% Generate chimes for detection events
chime = generateChimeAtSample(chimePosition,...
```

[...]

Mask

Trigger

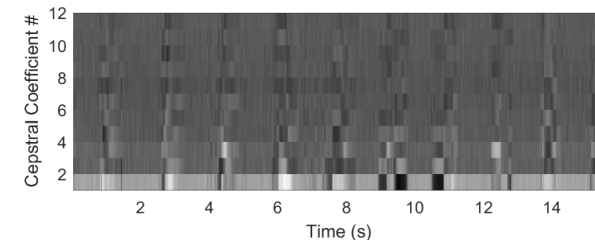
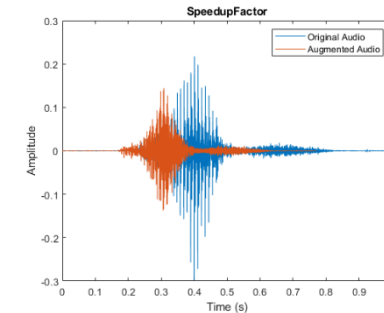
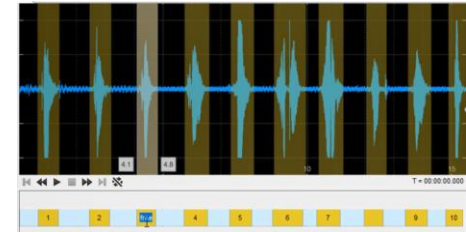
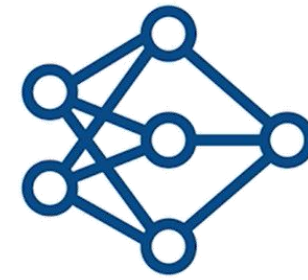




# Agenda

- Basics on training deep neural networks for signals
- Annotating data to train networks for practical applications
- Generating new data – synthesis and augmentation
- Creating inputs for deep networks

▪ From system models to real-time prototypes



## CREATE AND ACCESS DATASETS

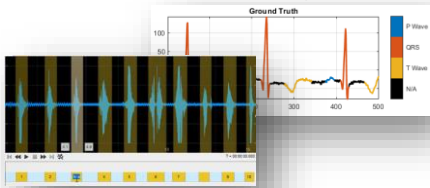
### Data sources



### Simulation and augmentation

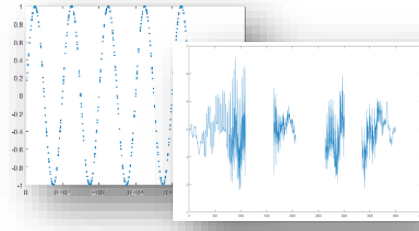


### Data Labeling

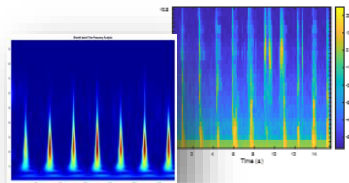


## PREPROCESS AND TRANSFORM DATA

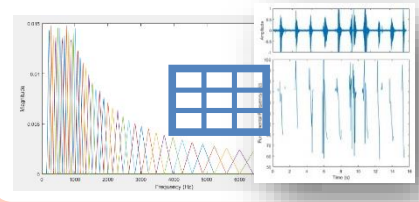
### Pre-Processing



### Transformation



### Feature extraction

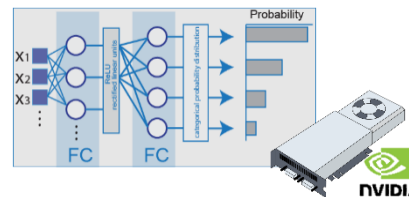


## DEVELOP PREDICTIVE MODELS

### Import Reference Models/ Design from scratch



### Hardware-Accelerated Training

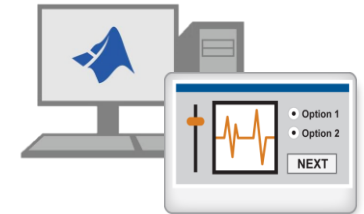


### Analyze and tune hyperparameters



## ACCELERATE AND DEPLOY

### Desktop Apps



### Enterprise Scale Systems

Java  
MATLAB  
C/C++  
Python

### Embedded Devices and Hardware



>> generateAudioPlugin triggerWordDetector

..

triggerWordDetector.m

```
[...]

% Extract MFCC from whole analysis buffer
[coeffs,delta,deltaDelta] = mfcc(buf,SampleRate,...
    'WindowLength',winLength,...
    'OverlapLength',ovlpLength);

% Concatenate and normalize features
featureMatrix = [coeffs,delta,deltaDelta];
featureMatrix = (featureMatrix - M)./S;

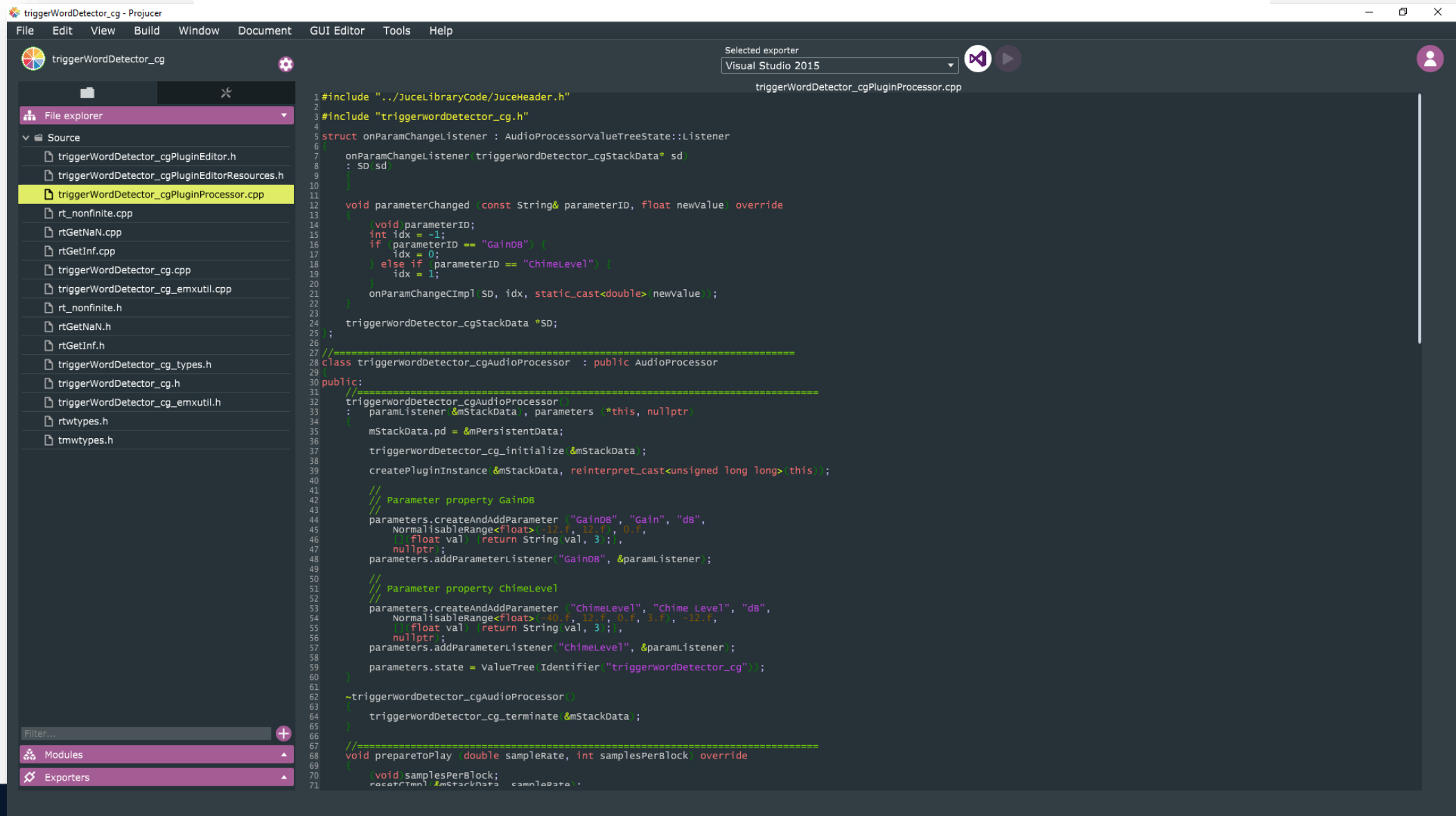
% Detect keyword with LSTM network (Mask around speech keyword)
featMask = classify(net,featureMatrix.);

% Debounce and re-align detections in time domain
[timeMask, chimePosition] = debounceAnalyzeDetectionMask(featMask);

% Generate chimes for detection events
chime = generateChimeAtSample(chimePosition,...

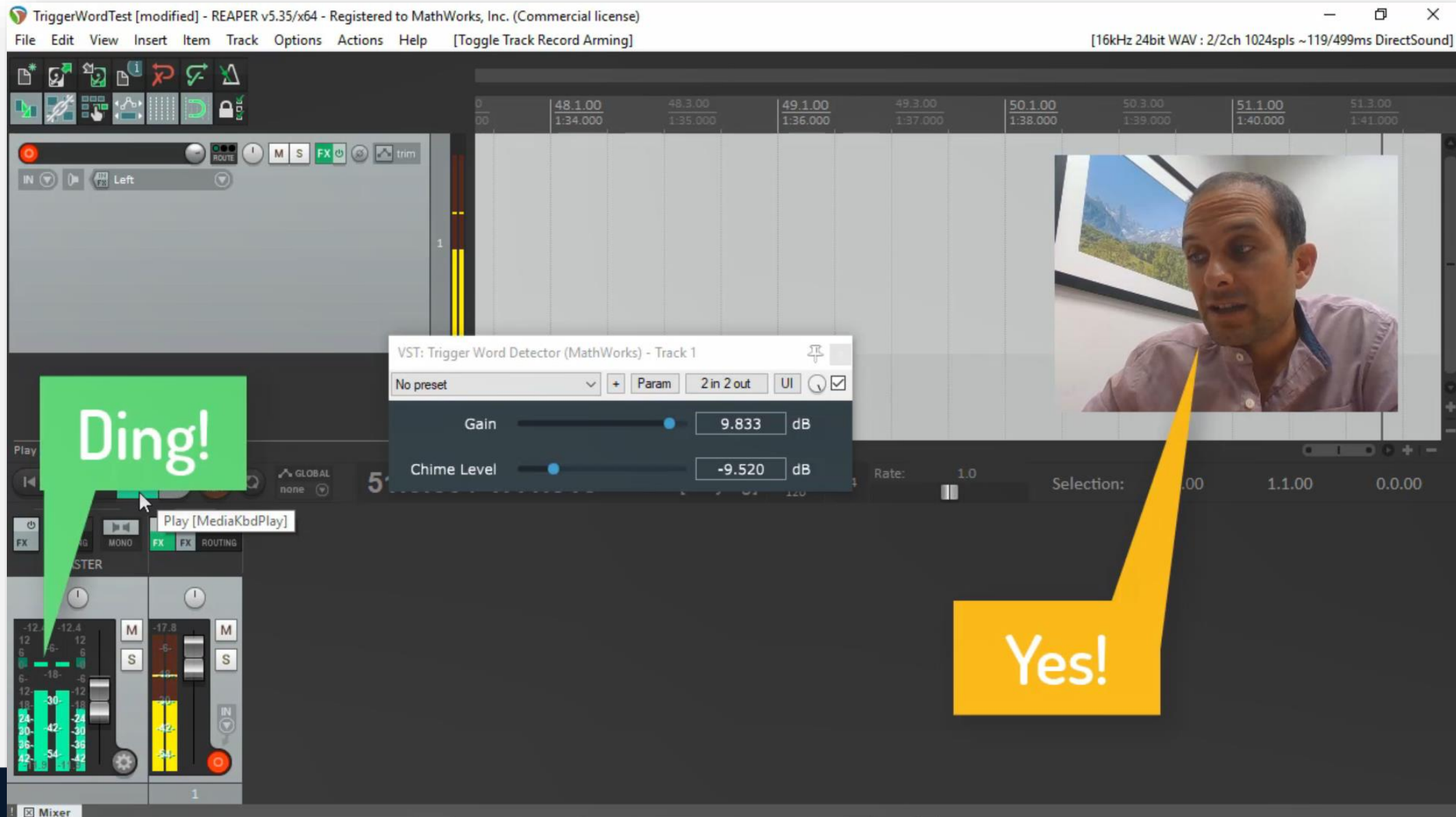
[...]
```

## >> generateAudioPlugin triggerWordDetector



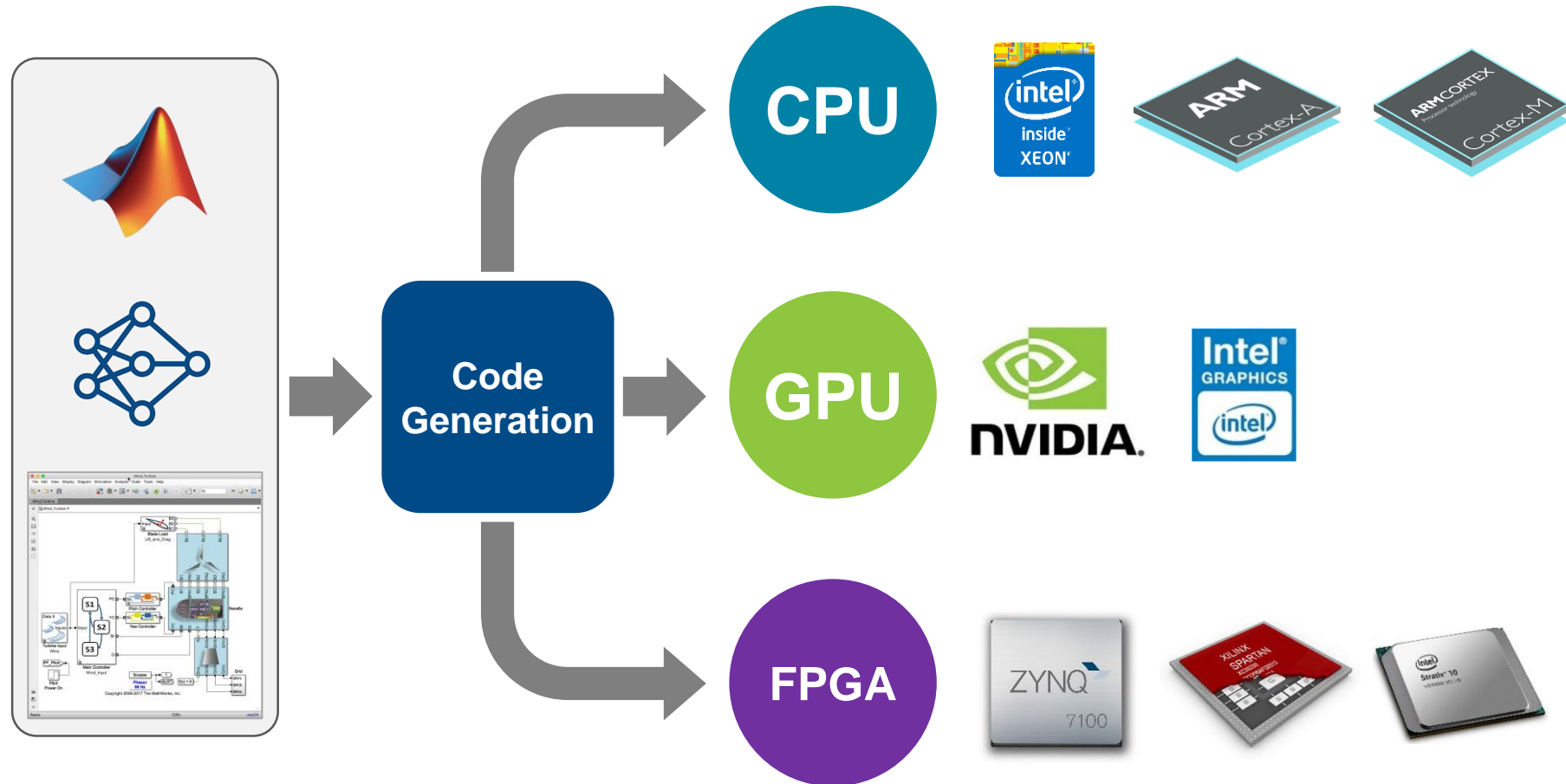


## >> generateAudioPlugin triggerWordDetector



# Deploy to any processor with best-in-class performance

AI models in MATLAB and Simulink can be deployed on embedded devices, edge devices, enterprise systems, the cloud, or the desktop.



Q: "What do I need to develop such a system?"

A: "A simple and proven deep learning model"

A: "A lot of data, a good dose of signal processing expertise, and the right tools for the specific application in hand"

**Deep learning systems can only be as good as the data used to train them**