

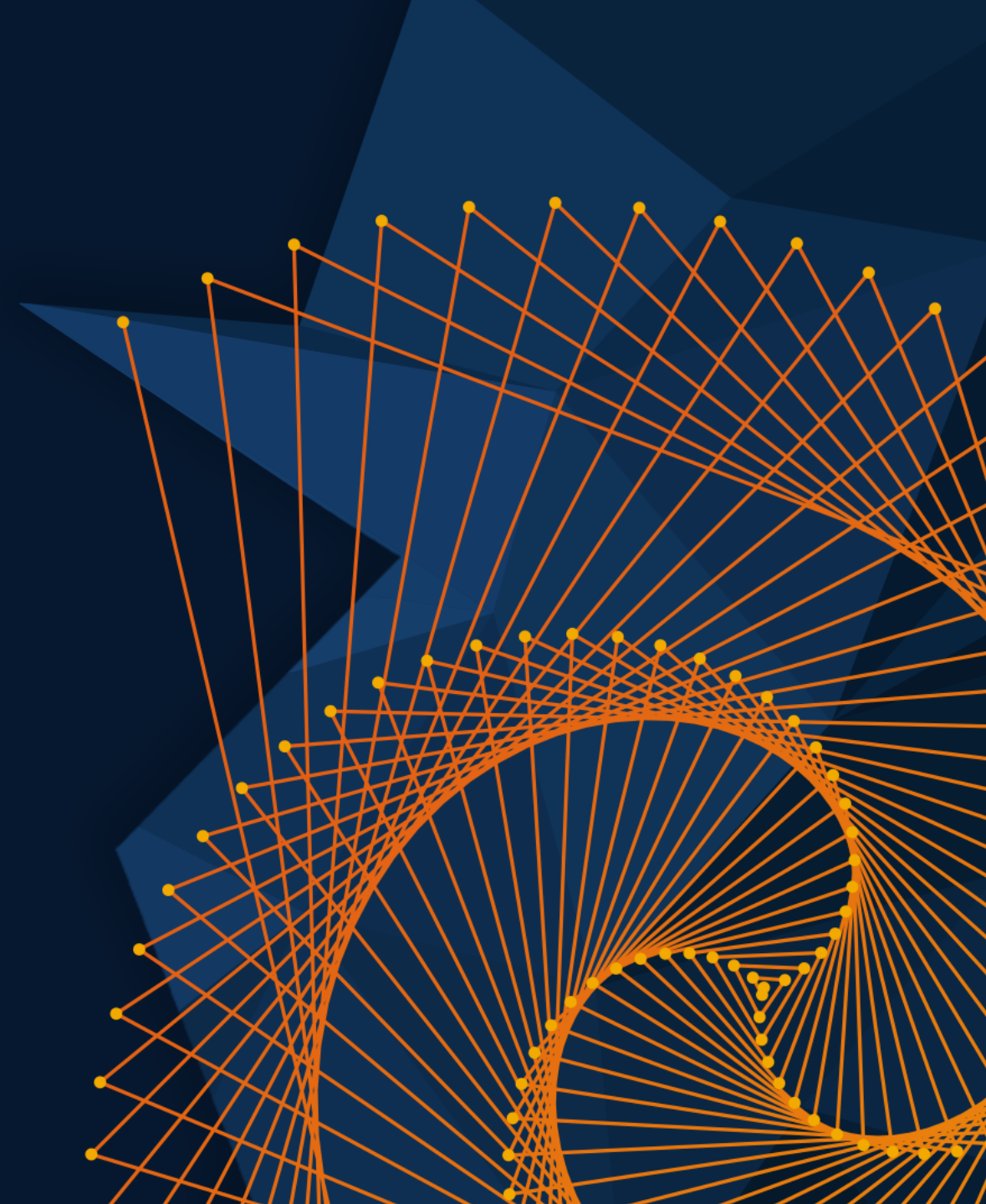
MATLAB EXPO

2024.06.11 | 그랜드 인터컨티넨탈 서울 파르나스

Audio Data Recognition

Using Deep Learning

엄준식, 매스웍스코리아



Contents

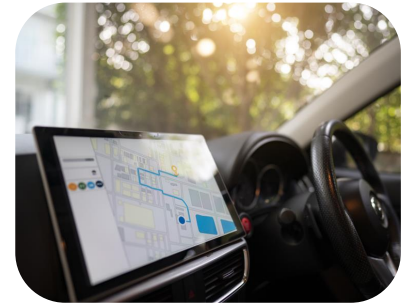
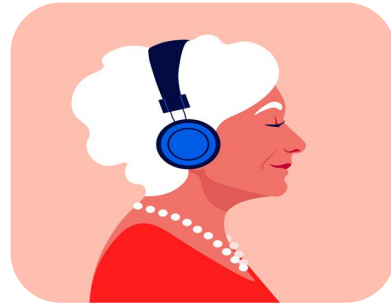
- Introduction
- Audio Data Pre-Processing
- Deep Learning Processing
- Additional Information
- Q & A

Introduction

Intelligent Assistant



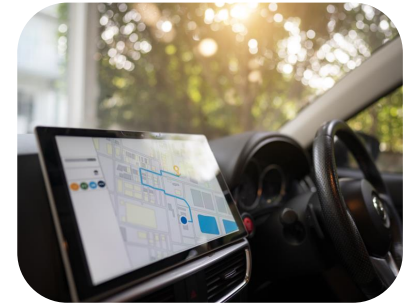
[Smart Home]



Intelligent Assistant



[Music]

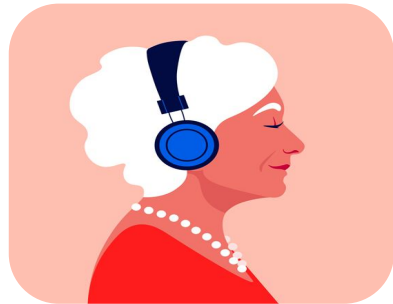


Intelligent Assistant



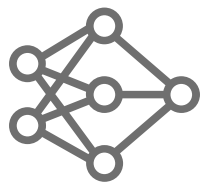
[Movie]

Intelligent Assistant



[Navigation]

Audio Toolbox



AI for Audio, Speech, and Acoustics



Audio processing and I/O



Acoustics measurements



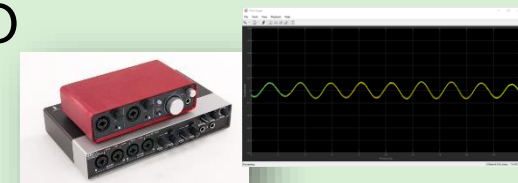
Deep learning



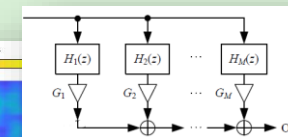
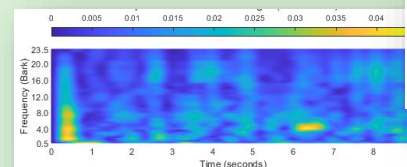
Audio plugins



Audio I/O



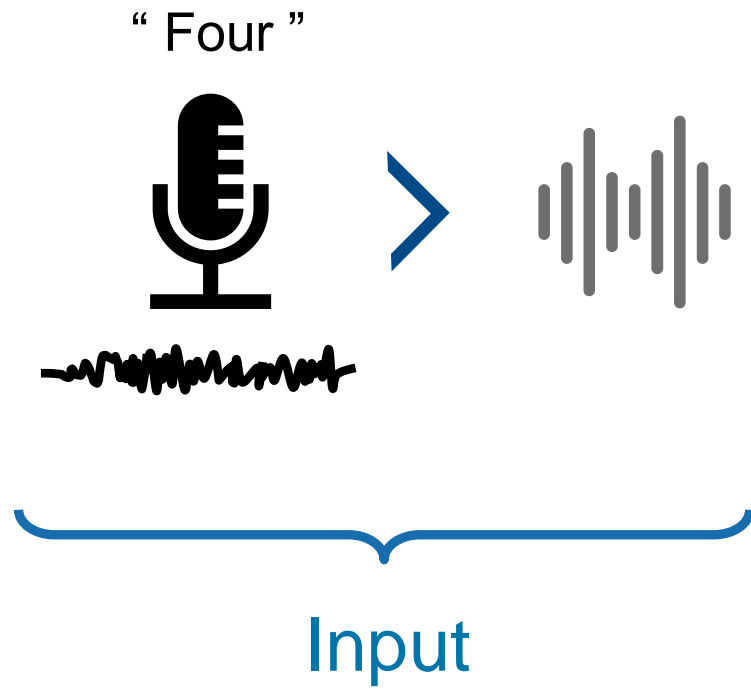
Signal processing



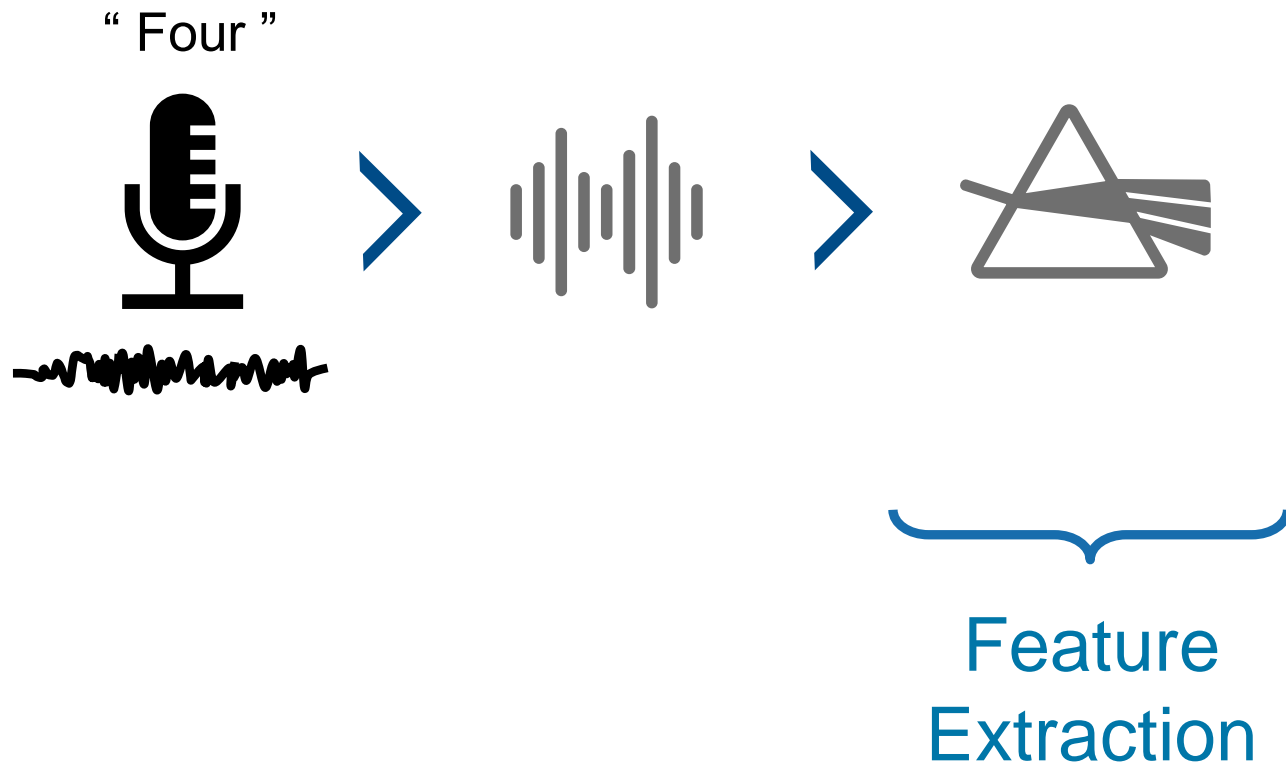
Audio Toolbox

- Speech-2-Text
- Speaker Recognition
- Voice Activity Detection
- Speech Enhancement
- Sound and Speech Synthesis
- Sound Classification
- Pitch Estimation
- Speech Separation

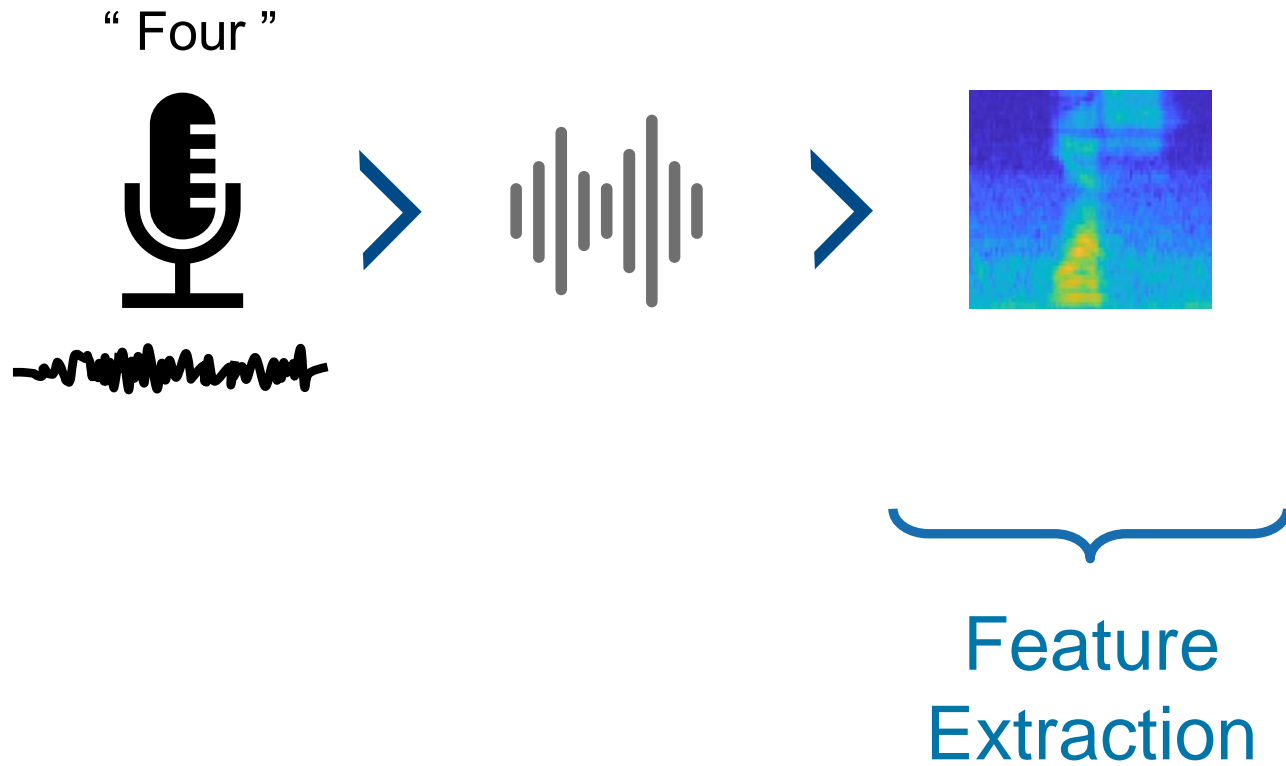
Demo Goal



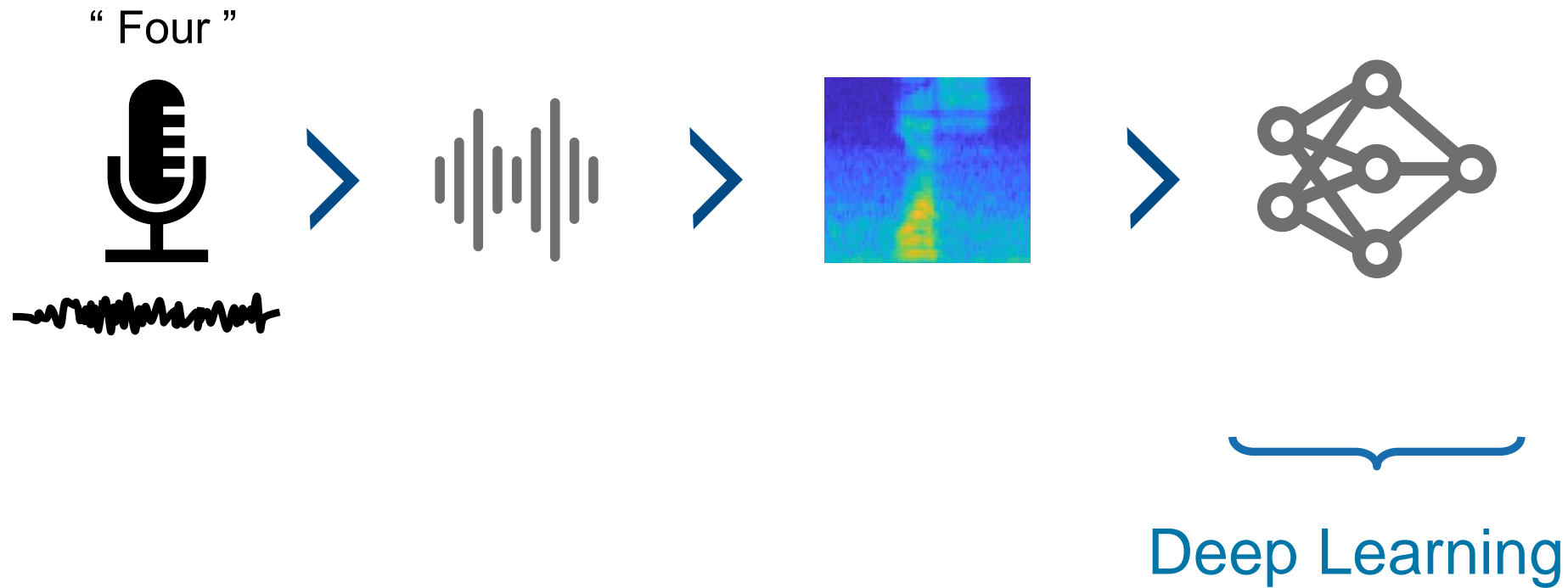
Demo Goal



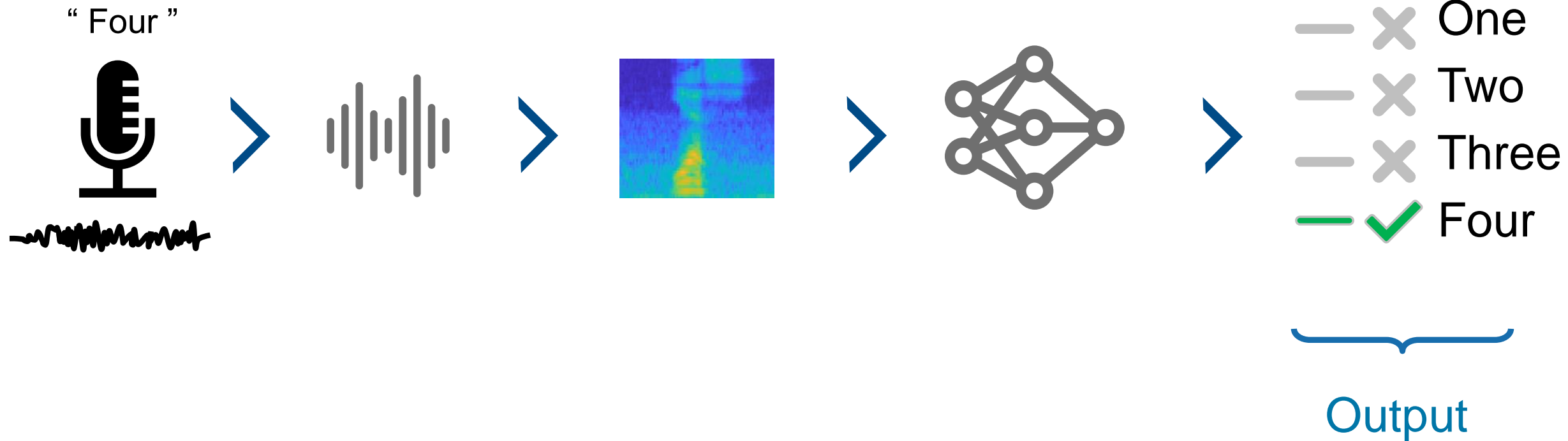
Demo Goal



Demo Goal



Demo Goal



Audio Data Pre-Processing

Audio Data Sources

- Speech Commands Data Set
 - A set of 64,727 audio files(wav)
 - A single spoken English word

tree marvin
eight house
sheila on four right
cat one yes zero left
wow no stop go bed
bird six seven five
three two up nine
down off dog
happy

Audio Data Sources

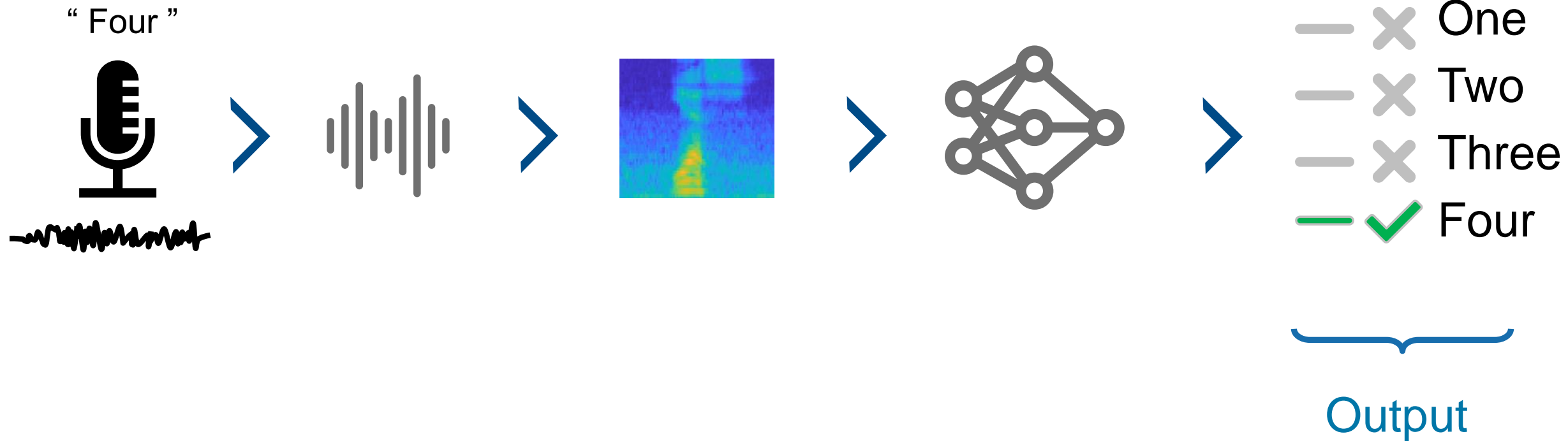
- Speech Commands Data Set
 - A set of 64,727 audio files(wav)
 - A single spoken English word

tree marvin
eight house
sheila on four right
catone **yes** **zero** left
wow **no** **stop** go bed
birdsix **seven** five
three two up nine
down off dog
happy

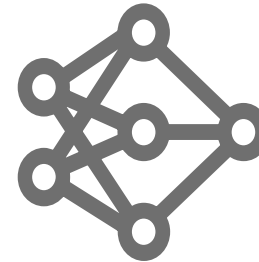
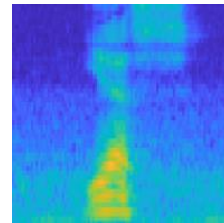


one two
Three four

Demo Goal



Speech Detection



- × One
- × Two
- × Three
- ✓ Four

Output

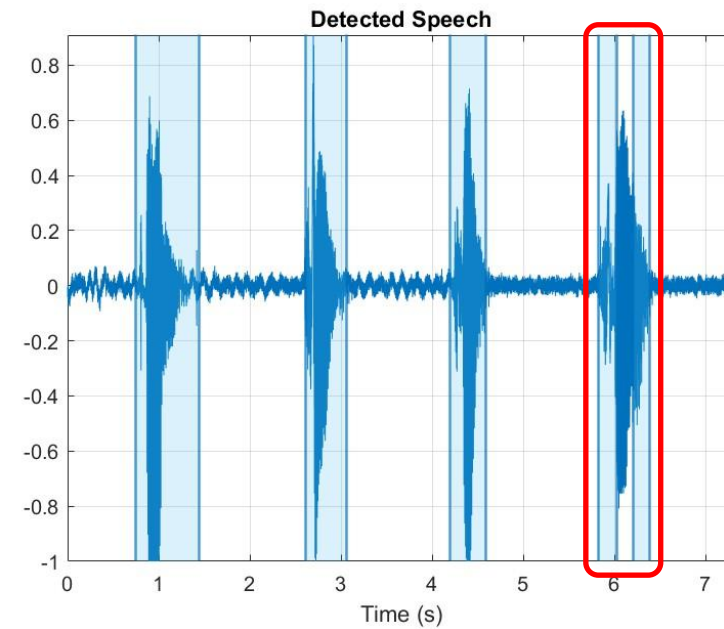
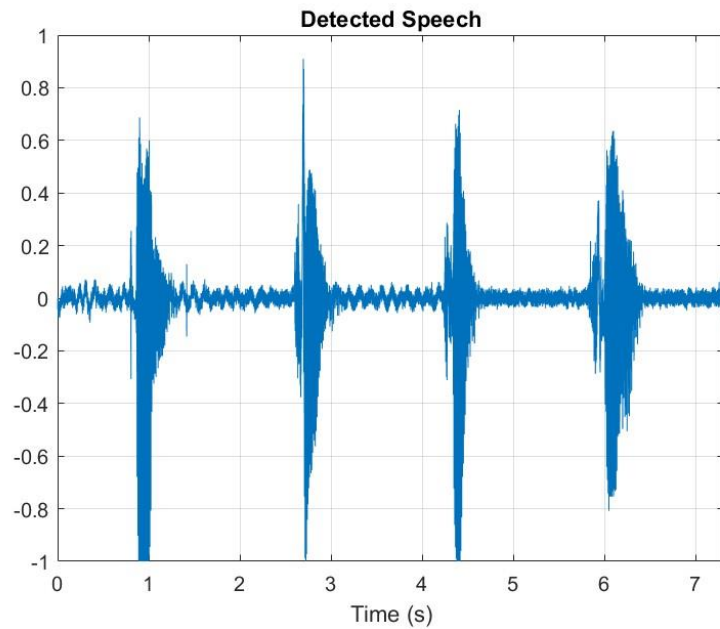
Speech Detection





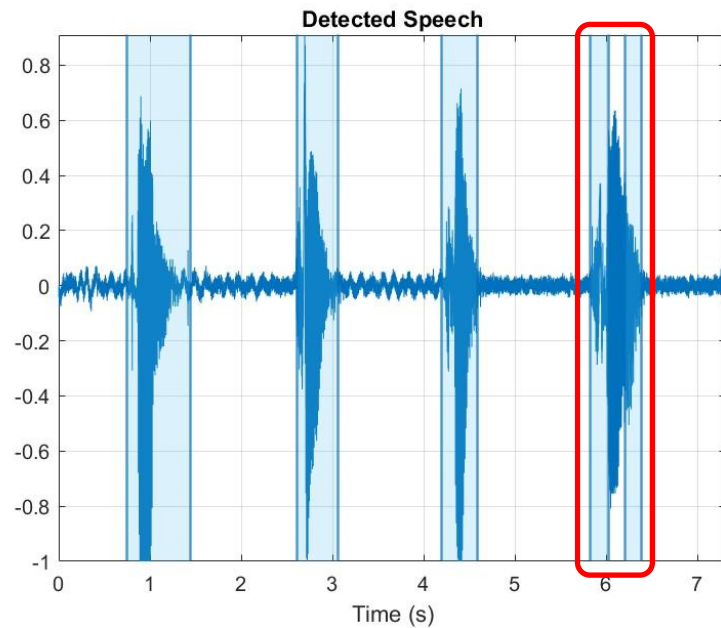
Speech Detection

```
>> detectSpeech(audioData, samplingFrequency)
```



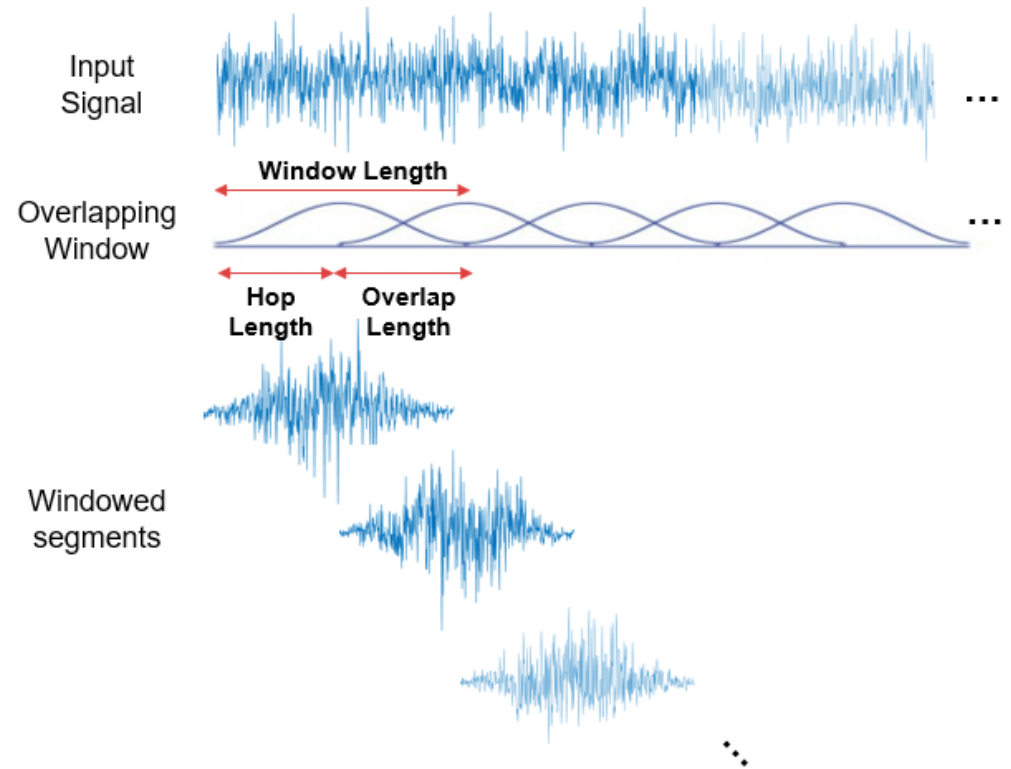
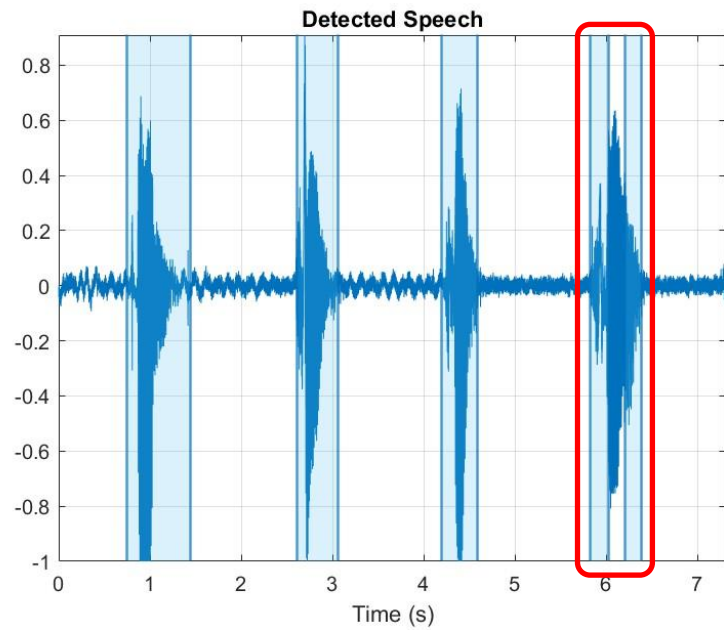


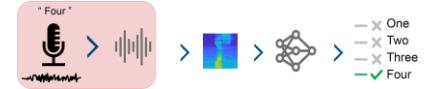
Speech Detection



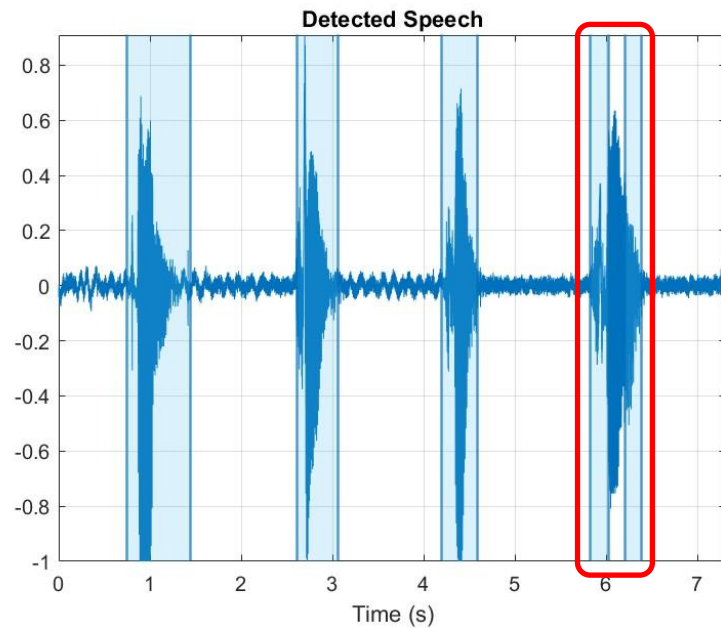
- Window
 - Time-frequency representation
- OverlapLength
 - Decrease noise
- MergeDistance
 - Regions declared are merged

Speech Detection





Speech Detection



```
Fs = 16000; % Sampling Frequency
```

```
windowDuration = 0.074;
```

```
numWindowSamples = round(windowDuration*Fs);
```

```
win = hamming(numWindowSamples, "periodic");
```

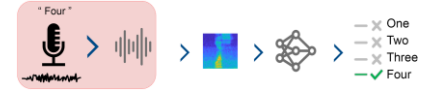
```
percentOverlap = 0.20;
```

```
overlap = round(numWindowSamples*percentOverlap);
```

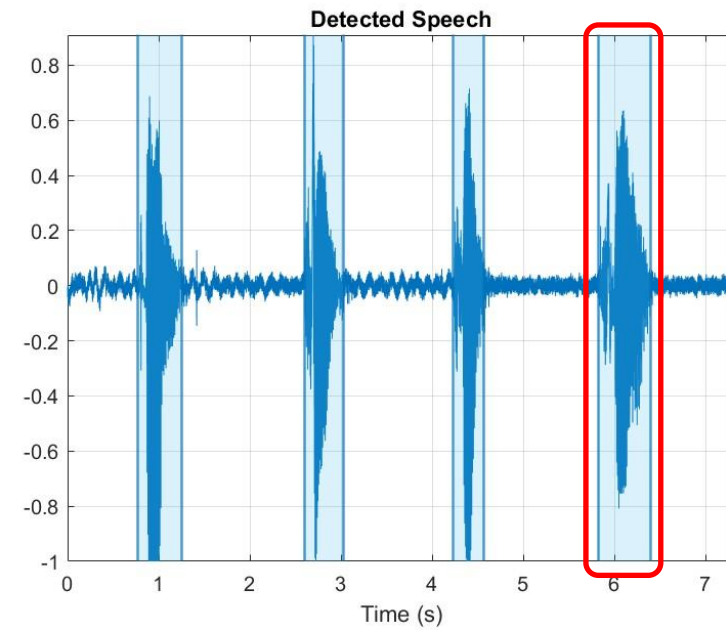
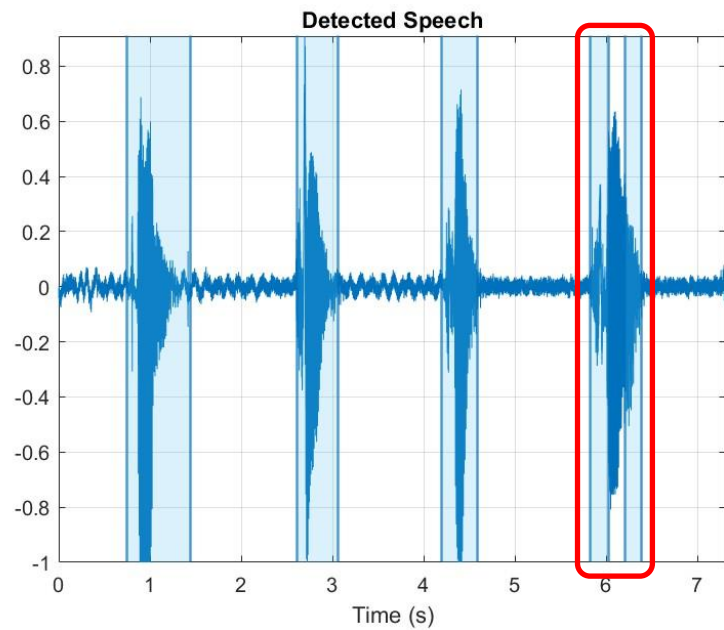
```
mergeDuration = 0.2;
```

```
mergeDist = round(mergeDuration*Fs);
```

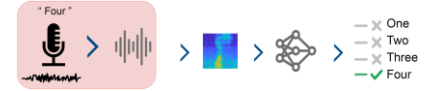
```
idx = detectSpeech(audioData, Fs, ...
    "Window", win, ...
    "OverlapLength", overlap, ...
    "MergeDistance", mergeDist);
```

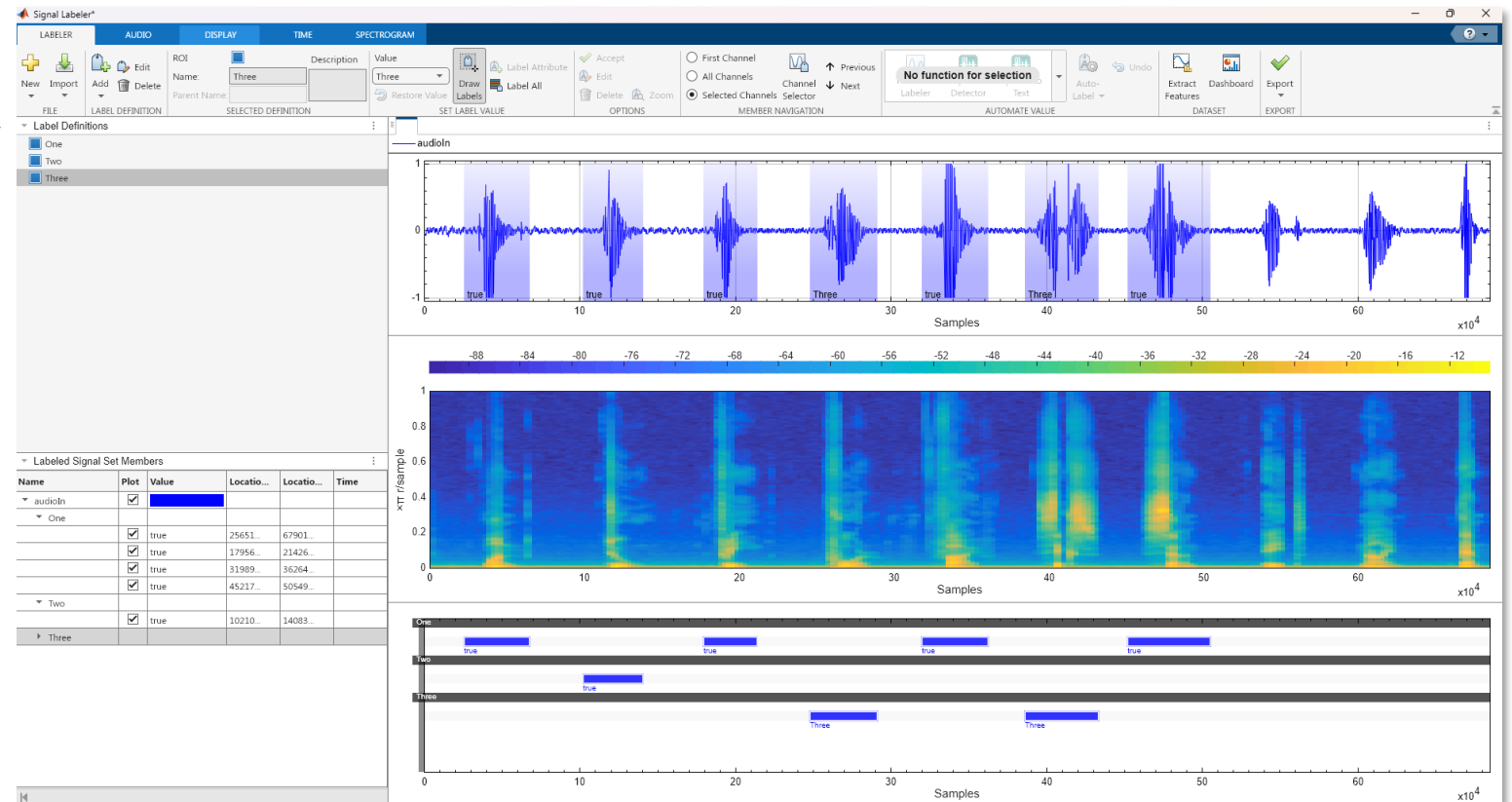
Speech Detection



Speech Labeling



- Signal Labeler App
 - Manually and Automatically
 - Audio Data Labeling
 - Spectrogram
 - Spectrum
 - Signal Feature Extraction





Data Augmentation

Audio Data Augmentation

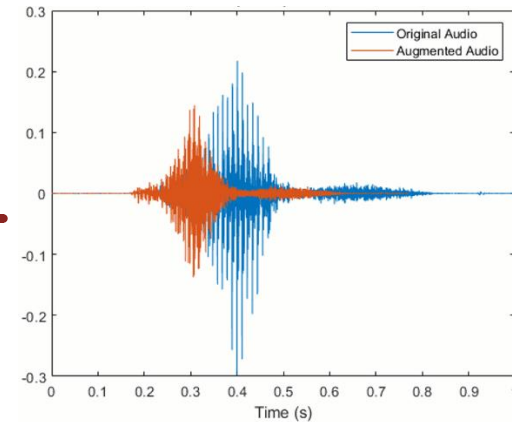
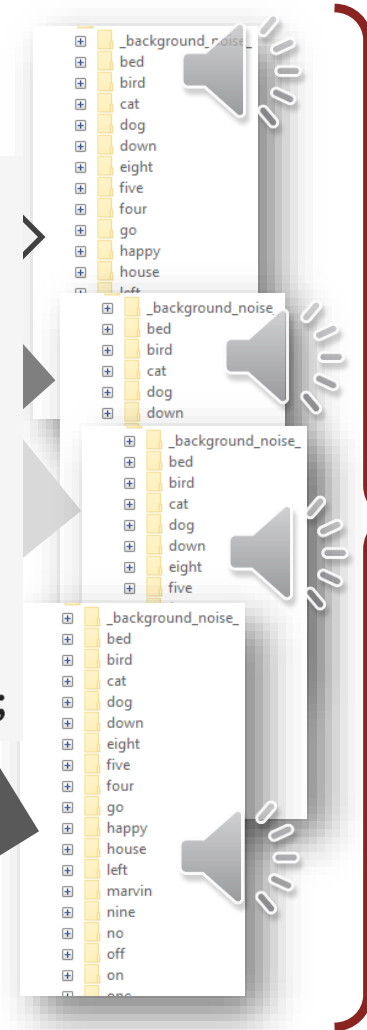
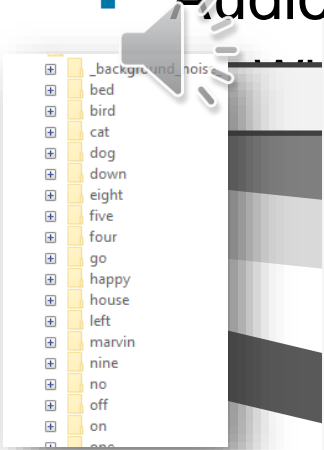
```

augmenter = audioDataAugmenter( ...
    "AugmentationMode", "independent", ...
    "AugmentationParameterSource", "specify", ...
    "ApplyTimeStretch", false, ...
    "ApplyPitchShift", false, ...
    "ApplyVolumeControl", false, ...
    "ApplyAddNoise", false, ...
    "ApplyTimeShift", true, ...
    "TimeShift", 25e-3);
  
```

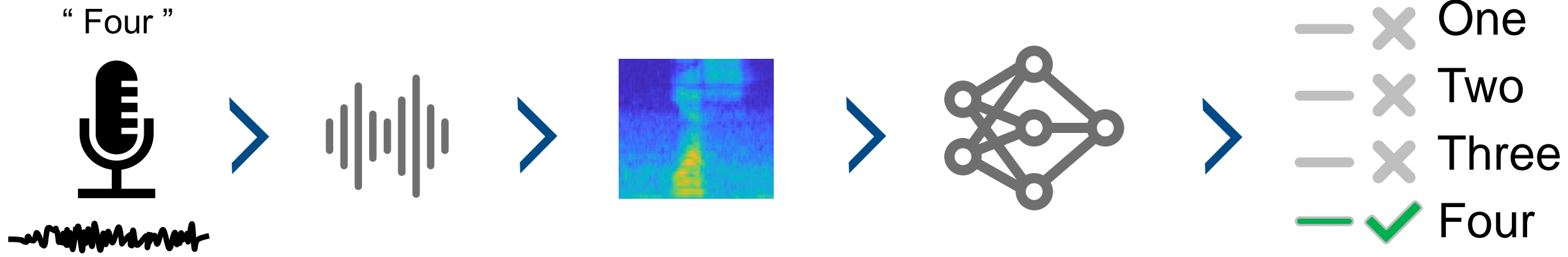
```

data = augment(augmenter, audioData, samplingFrequency);
  
```

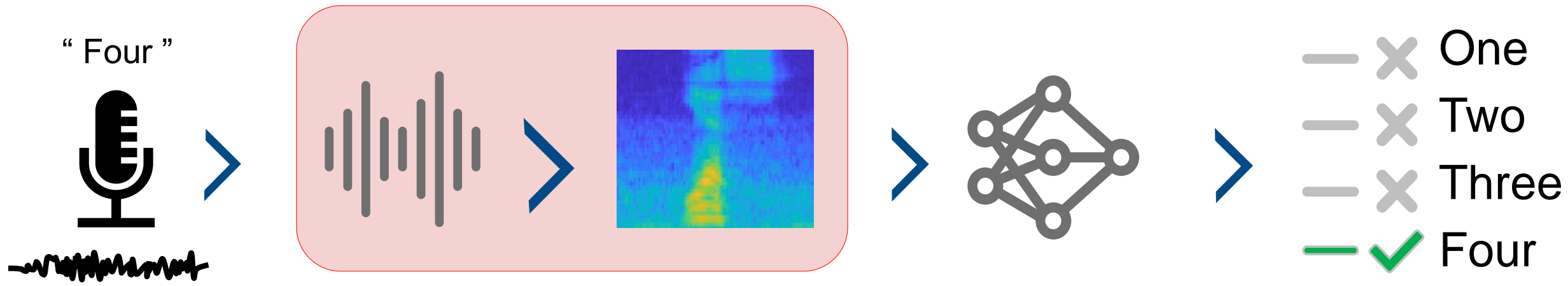
Original
Dataset



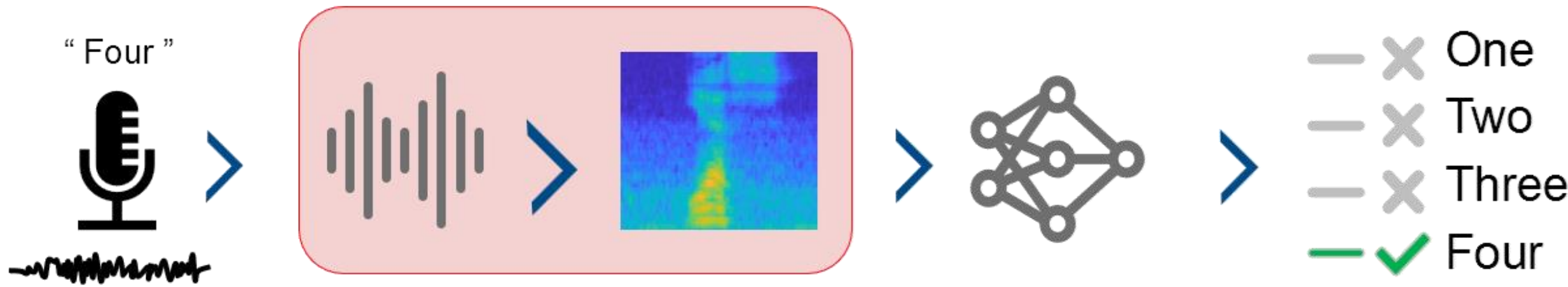
Demo Goal



Feature Extraction

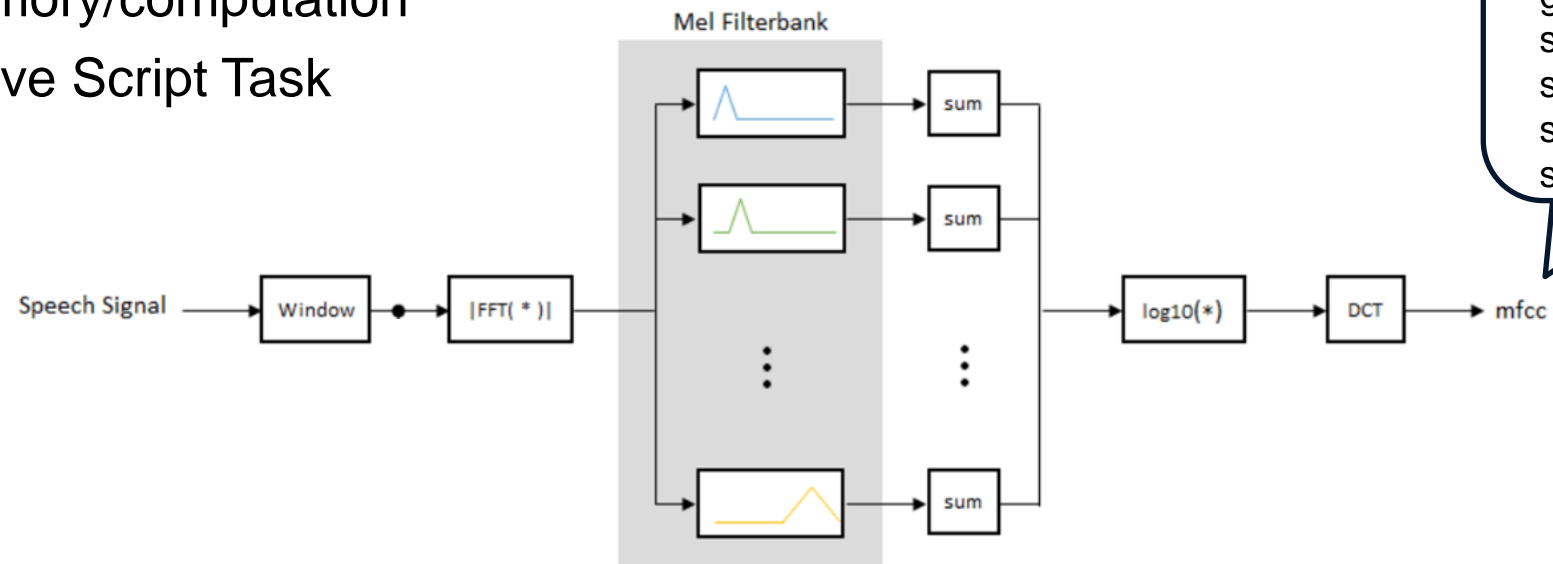


Feature Extraction

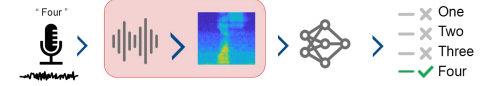


Feature Extraction

- >> audioFeatureExtractor
 - Extract multiple features at once
 - Optimize memory/computation
 - Possible in Live Script Task



mfcc
mfccDelta
gtcc
gtccDelta
spectralCentroids
spectralCrest
spectralEntropy
spectralFlatness



Feature Extraction

- >> audioFeatureExtractor
 - Extract multiple features at once
 - Optimize memory/computation
 - Possible in Live Script Task

Create object of feature extractor

```
afe = audioFeatureExtractor(...  
    "SampleRate", samplingFrequency, ...  
    "Window", hann(frameSamples, "periodic"), ...  
    "OverlapLength", overlapSamples, ...  
    "barkSpectrum", true, ...  
    "mfcc", true, ...  
    "mfccDelta", false, ...  
    "mfccDeltaDelta", false, ...  
    "pitch", false, ...  
    "spectralCentroid", false, ...  
    "zerocrossrate", false, ...  
    "shortTimeEnergy", false)
```




Feature Extraction

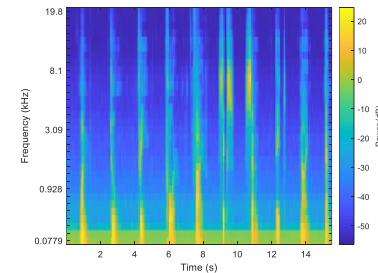
- >> audioFeatureExtractor
 - Extract multiple features at once
 - Optimize memory/computation
 - Possible in Live Script Task

Set parameters

```
% Set parameters of the Bark spectrum extraction
setExtractorParameters(afe, "barkSpectrum", ...
    "NumBands", numBands, ...
    "WindowNormalization", false);
```

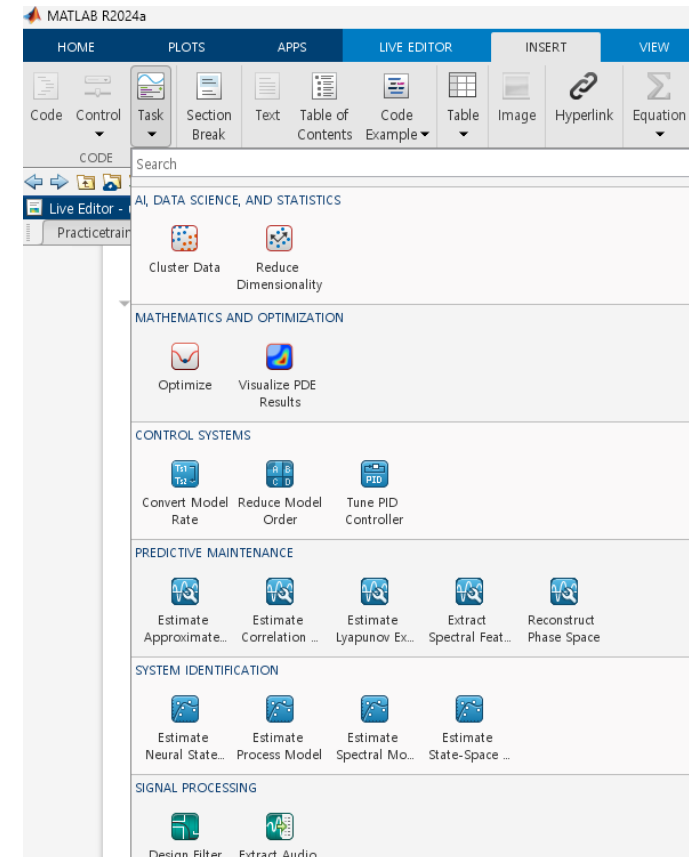
Extract & Scale the features

```
features = extract(afe, AudioData);
```



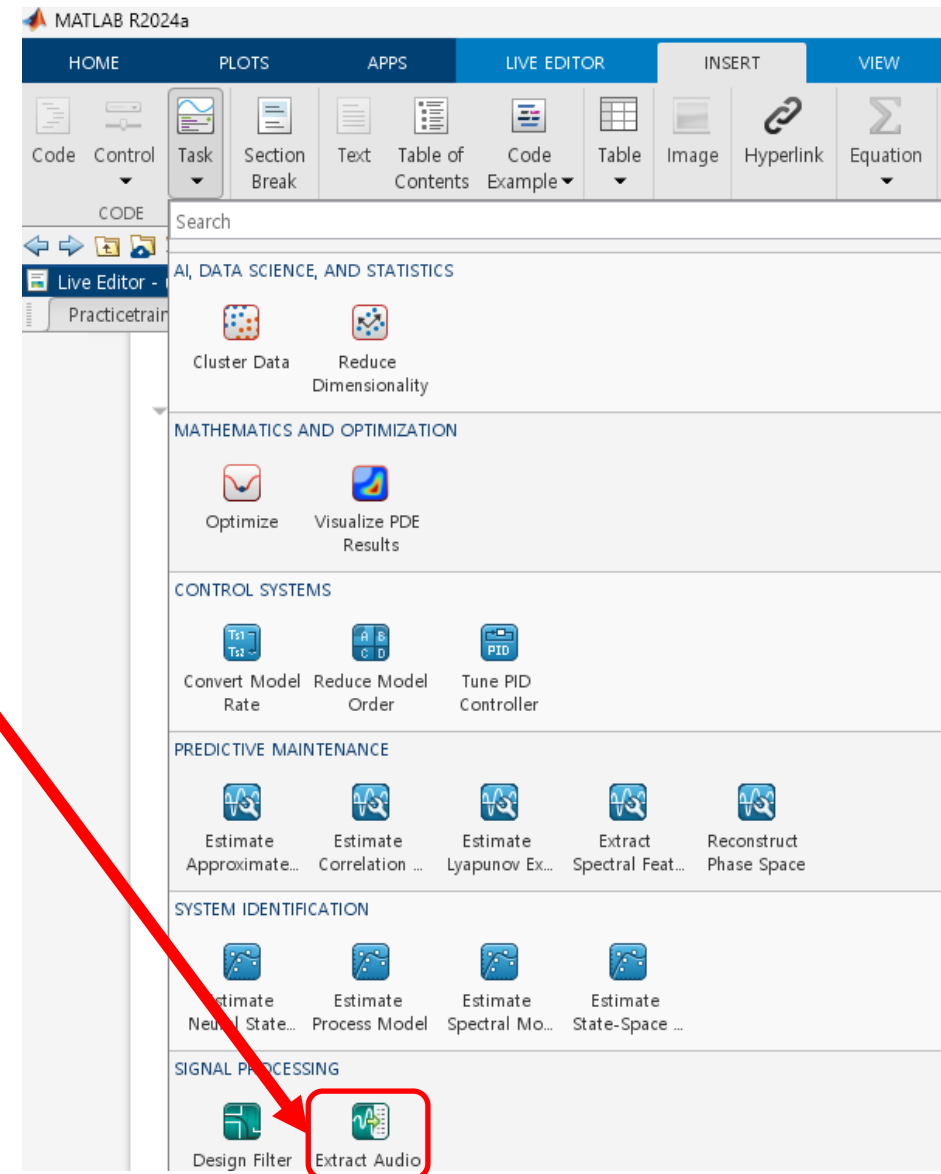
Feature Extraction – Live Script (GUI)

- >> audioFeatureExtractor
 - Extract multiple features at once
 - Optimize memory/computation
 - Possible in Live Script Task




Feature Extraction – Live Script (GUI)

- `>> audioFeatureExtractor`
 - Extract multiple features at once
 - Optimize memory/computation
 - Possible in Live Script Task



Feature Extraction – Live Script (GUI)



Extract Audio Features

Extract Audio Features

`features2`, `extractor` = MFCC and spectral centroid extracted from `audioIn`

▼ Select data

Input audio data `audioIn` Sample rate (Hz) `Fs`

▼ Specify window properties

Window `Hamming` `1024` `samples`

Overlap length `50` `%` FFT length `Auto`

▼ Select features to extract

Spectral features

Linear spectrum Mel spectrum Bark spectrum ERB spectrum

Cepstral features

MFCC MFCC delta MFCC delta delta

GTCC GTCC delta GTCC delta delta

Spectral descriptors

Centroid Crest Decrease Entropy

Flatness Flux Kurtosis Rolloff point

Skewness Slope Spread

Periodicity features

Pitch Harmonic ratio

Energy features

Zero-crossing rate Short-time energy

► Specify feature extractor parameters

Feature Extraction – Live Script (GUI)



Extract Audio
Features

▼ Select features to extract

Spectral features

Linear spectrum Mel spectrum Bark spectrum ERB spectrum

Cepstral features

MFCC MFCC delta MFCC delta delta

GTCC GTCC delta GTCC delta delta

Spectral descriptors

Centroid Crest Decrease Entropy

Flatness Flux Kurtosis Rolloff point

Skewness Slope Spread

Periodicity features

Pitch Harmonic ratio

Energy features

Zero-crossing rate Short-time energy

▶ Specify feature extractor parameters

▼ Display results

Output summary Plot features Plot audio

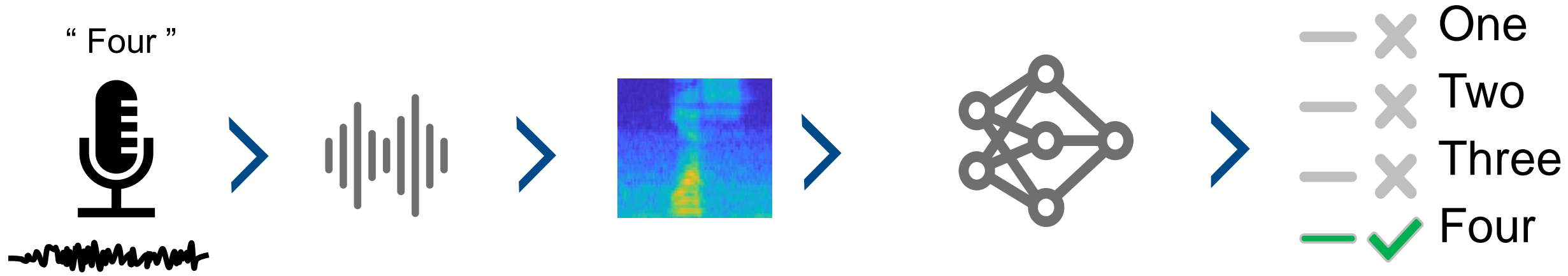
▼ Hide code

```
% Create and set up an audioFeatureExtractor object  
extractor = audioFeatureExtractor(SampleRate=Fs, ...  
    mfcc=true,spectralCentroid=true);
```

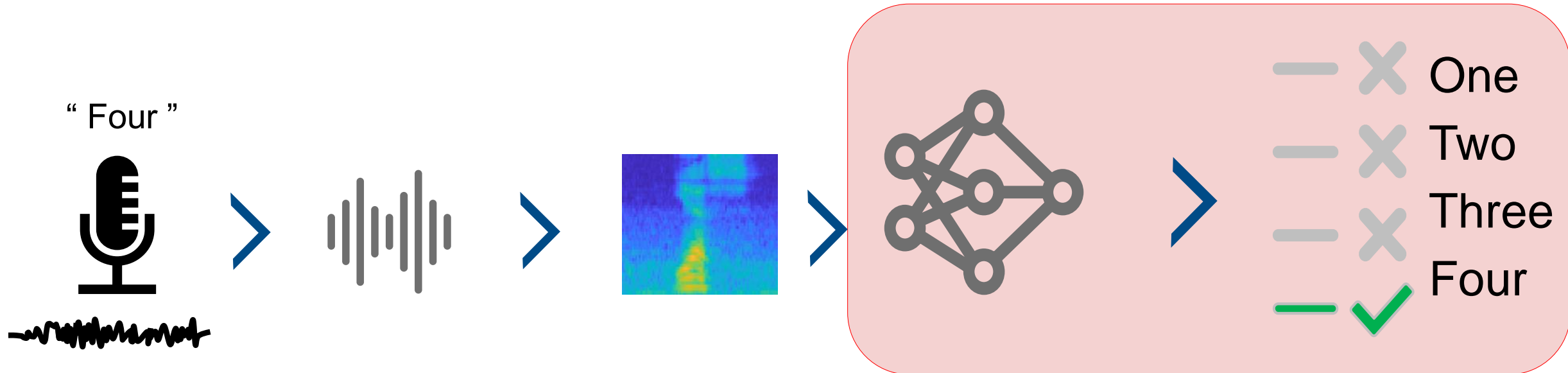
```
% Extract features from audio data  
features2 = extract(extractor, audioIn);
```

Deep Learning Processing

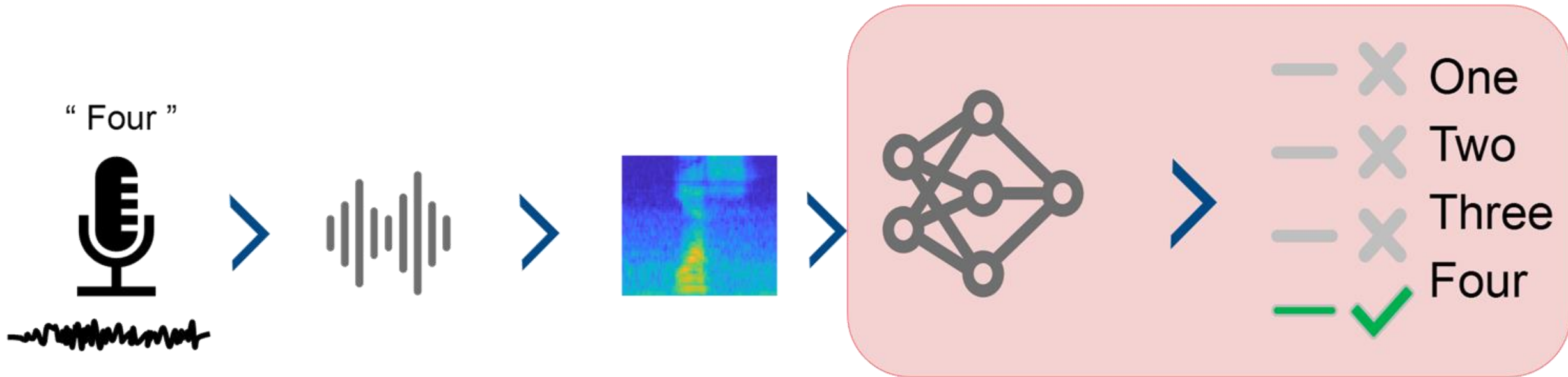
Demo Goal



Demo Goal

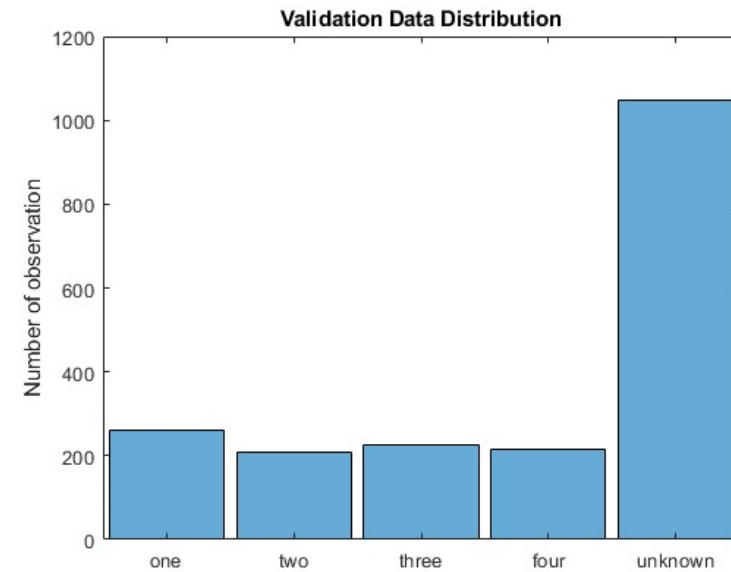
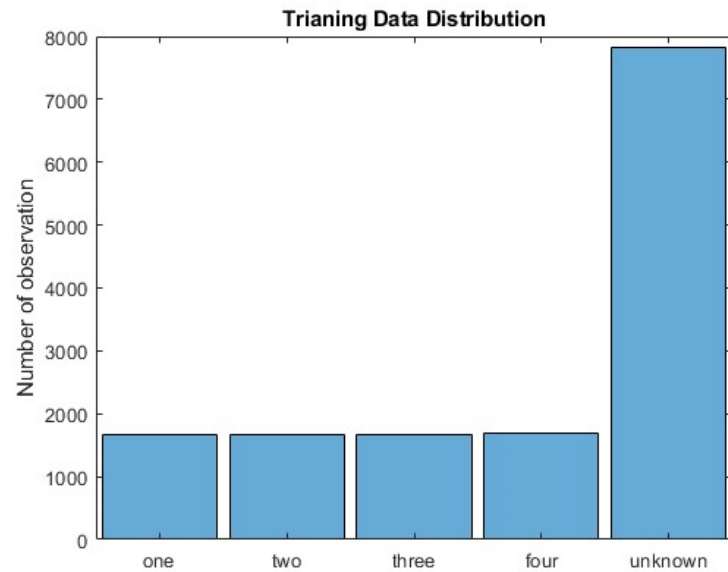
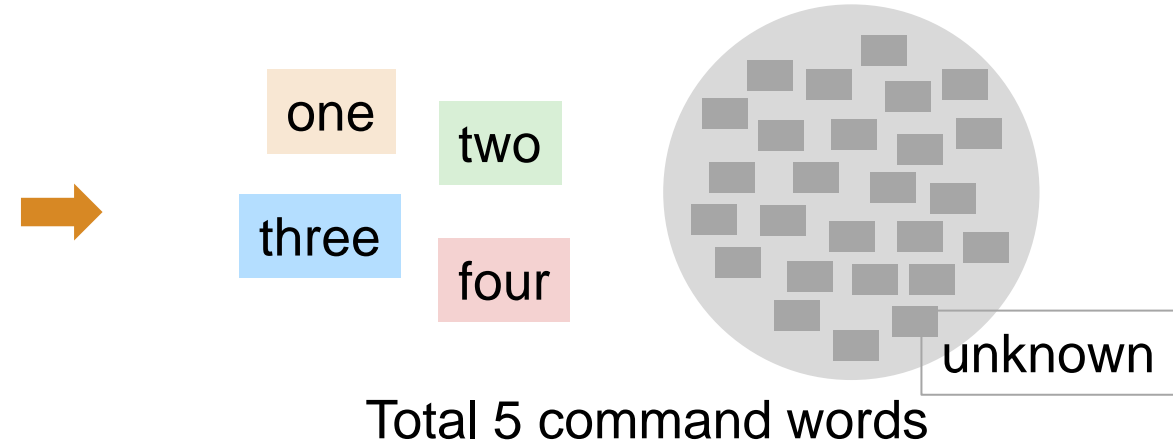
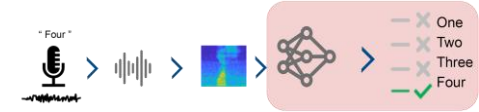


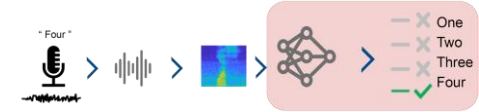
Demo Goal



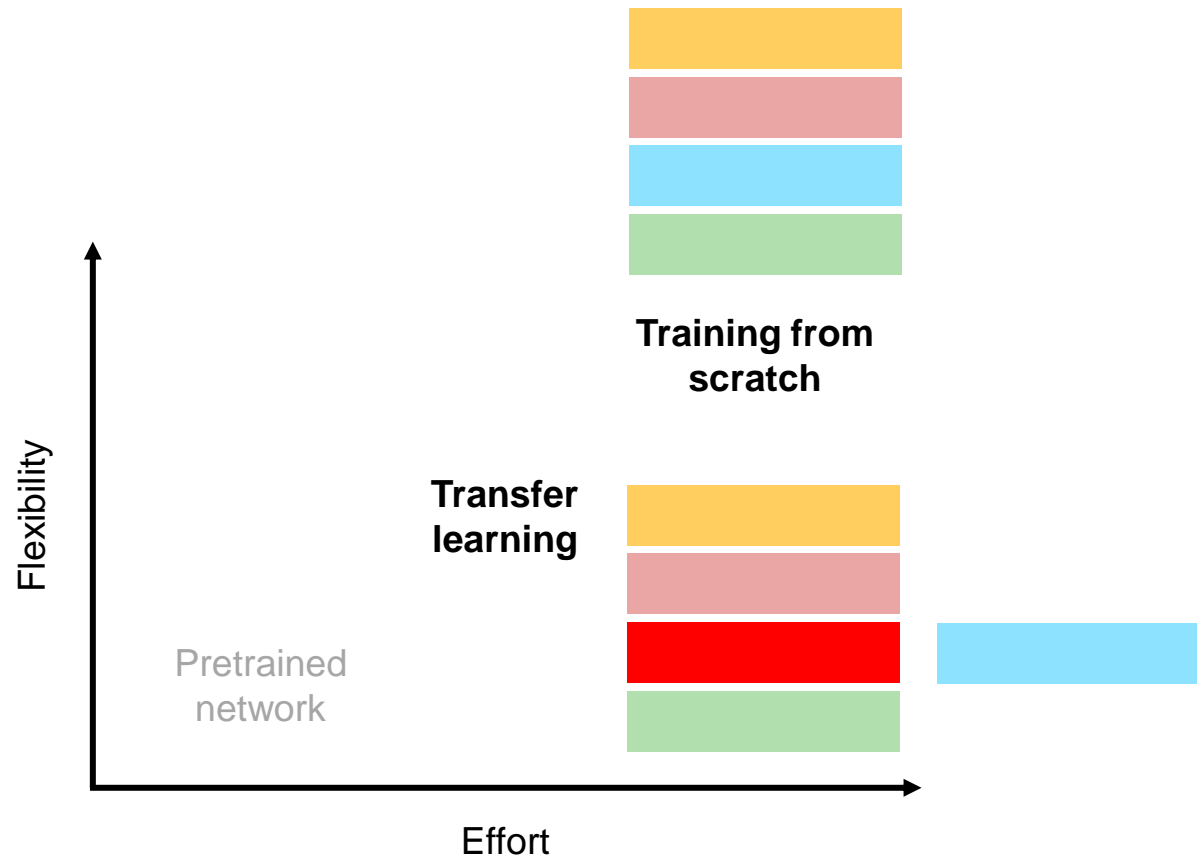
Training Data

- Speech Commands Data Set
 - 30 command words



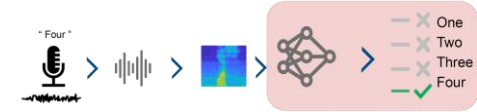


Deep Learning Network



- Training from Scratch
 - Build a network myself

- Transfer learning
 - Modify pretrained network



Training from Scratch VS Transfer Learning

- Training from Scratch

- Pros

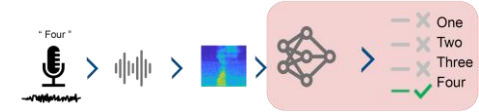
- We can train with **various dimension dataset**
 - We can change our model architecture.
 - Many possibilities to enhance performance

- Cons

- It **takes long time to train** from the beginning
 - It is hard to find proper and high performance model.

```
layers = [  
    imageInputLayer([numHops, afe.FeatureVectorLength])  
  
    convolution2dLayer(3, numF, "Padding", "same")  
    batchNormalizationLayer  
    reluLayer  
    maxPooling2dLayer(3, "Stride", 2, "Padding", "same")  
  
    convolution2dLayer(3, 2*numF, "Padding", "same")  
    batchNormalizationLayer  
    reluLayer  
    maxPooling2dLayer(3, "Stride", 2, "Padding", "same")  
  
    convolution2dLayer(3, 4*numF, "Padding", "same")  
    batchNormalizationLayer  
    reluLayer  
    maxPooling2dLayer(3, "Stride", 2, "Padding", "same")  
  
    convolution2dLayer(3, 4*numF, "Padding", "same")  
    batchNormalizationLayer  
    reluLayer  
  
    convolution2dLayer(3, 4*numF, "Padding", "same")  
    batchNormalizationLayer
```

Training from Scratch VS Transfer Learning



Transfer Learning

Pros

- It takes **less training time** than training from scratch
- The **Normal performance is guaranteed.**
- It doesn't take time to consider the architecture of deep learning model

Cons

- Deep learning **input data dimension is limited**
- It is hard to change deep learning model architecture as we want.

Deep Network Designer Start Page

MATLAB Deep Network Designer

Getting Started | Compare Pretrained Networks | Transfer Learning

> General

Image Networks (Pretrained)

SqueezeNet, GoogLeNet, ResNet-50, EfficientNet-b0, DarkNet-53, DarkNet-19, ShuffleNet

Inception-v3, ResNet-101, VGG-19, VGG-16, AlexNet

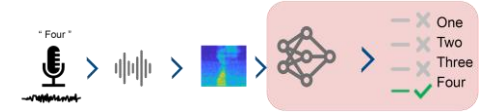
Sequence Networks

Audio Networks (Pretrained)

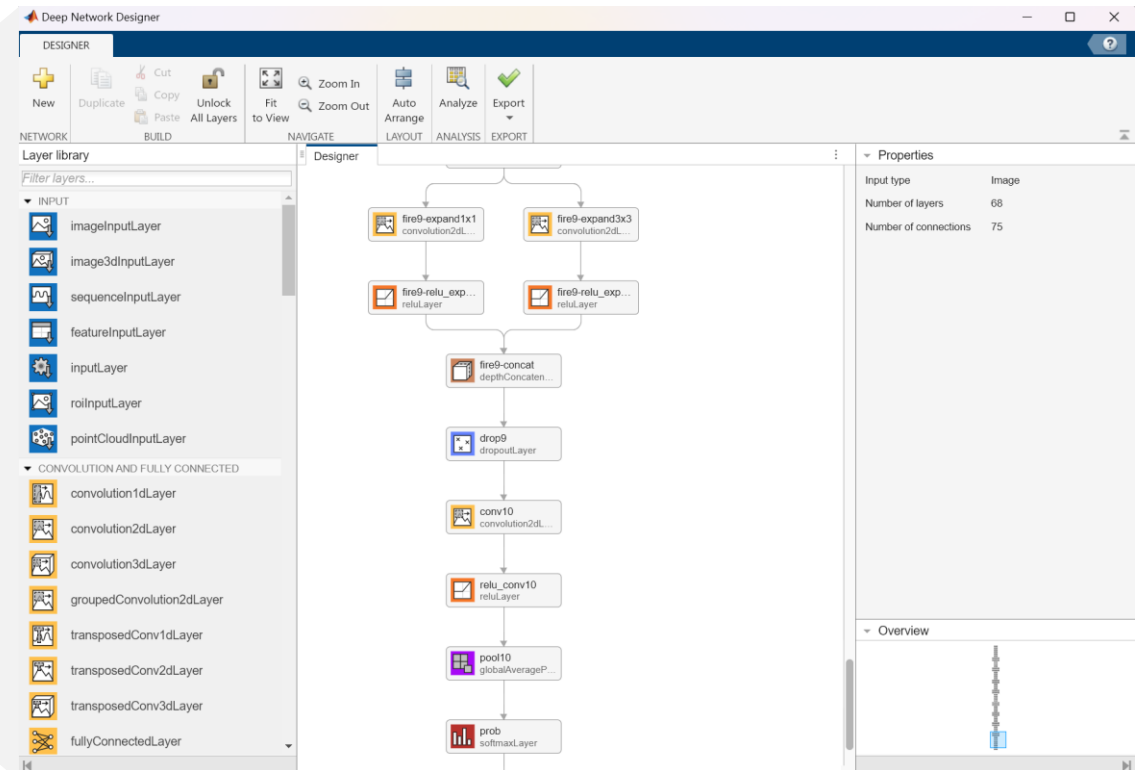
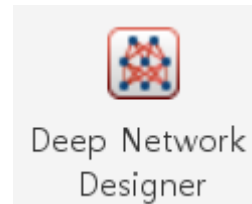
CREPE, OpenL3, VADNet, VGGish, YAMNet

Deep Network Designer

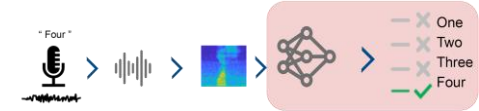
Deep Network Designer



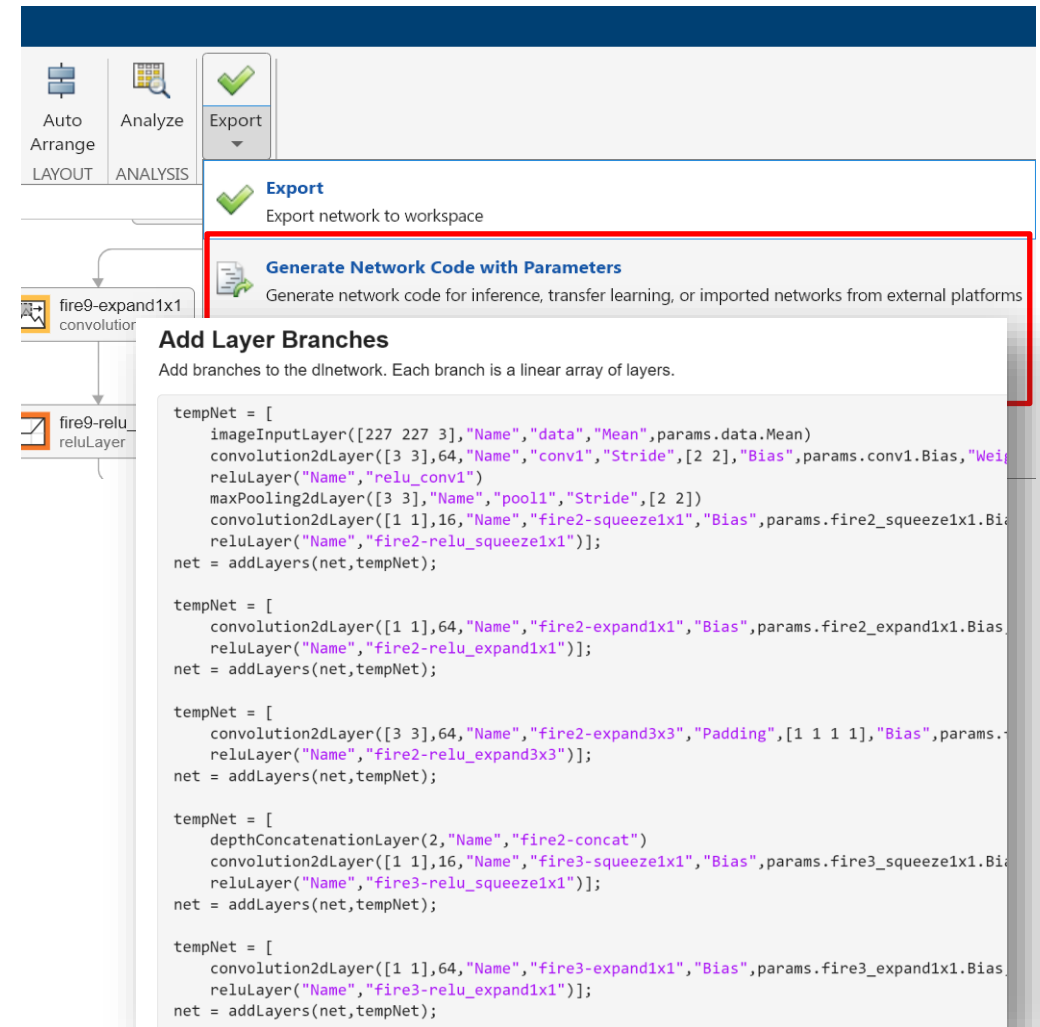
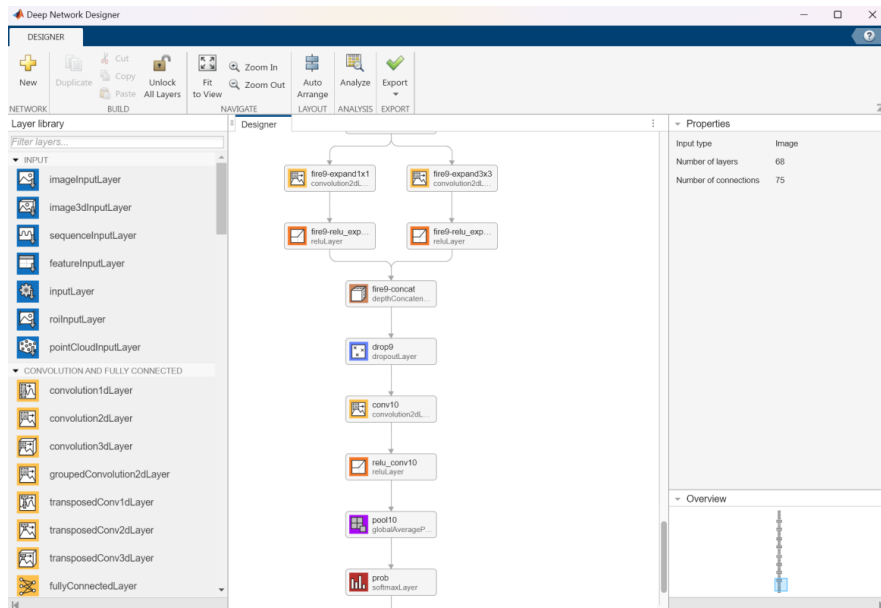
- >> deepNetworkDesigner
 - Deep Learning Network Design with App
 - GUI



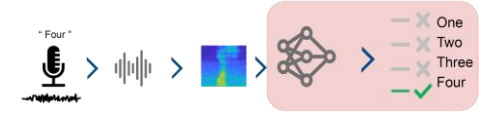
Deep Network Designer



- >> deepNetworkDesigner
 - Deep Learning Network Design with App
 - GUI



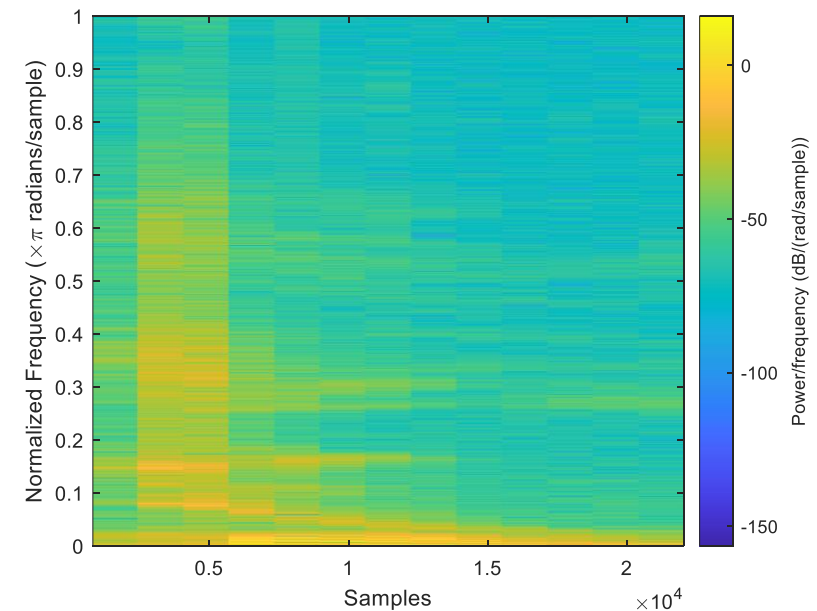
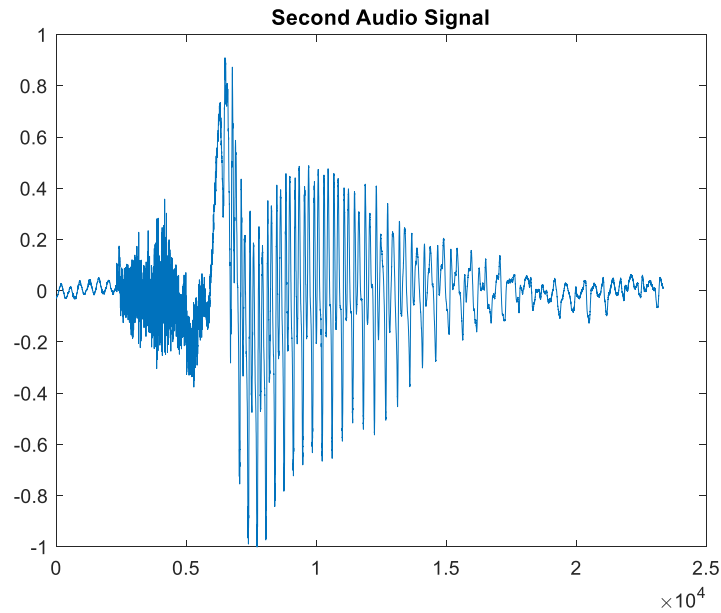
Feature Extraction for Scratch

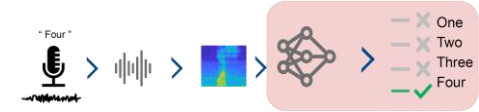


- Available features to Extract
 - 25 features (In the Demo, only bark spectrum is selected)

```
>> audioFeatureExtractor
```

```
>> extract
```

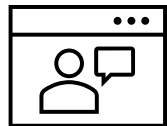
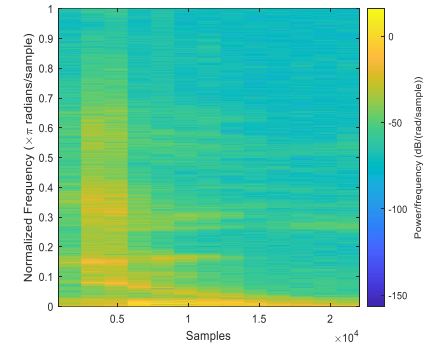
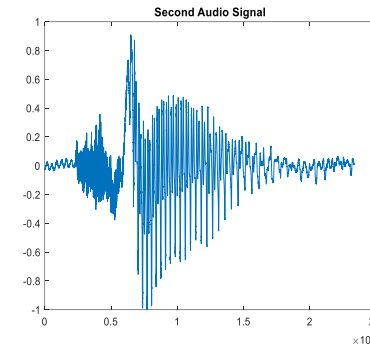




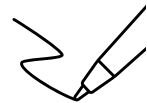
Feature Extraction for Transfer Learning

- VGGish Model(Pre-trained)
 - Audio classification model
 - A set of 2,000,000 audio data from Google
 - 527 classes
 - Feature(Mel Spectrogram)

```
>> vggishPreprocess
```



Youtube Video Clips

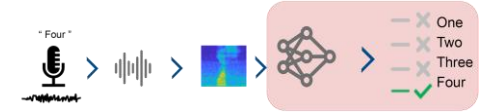


Labeling Manually



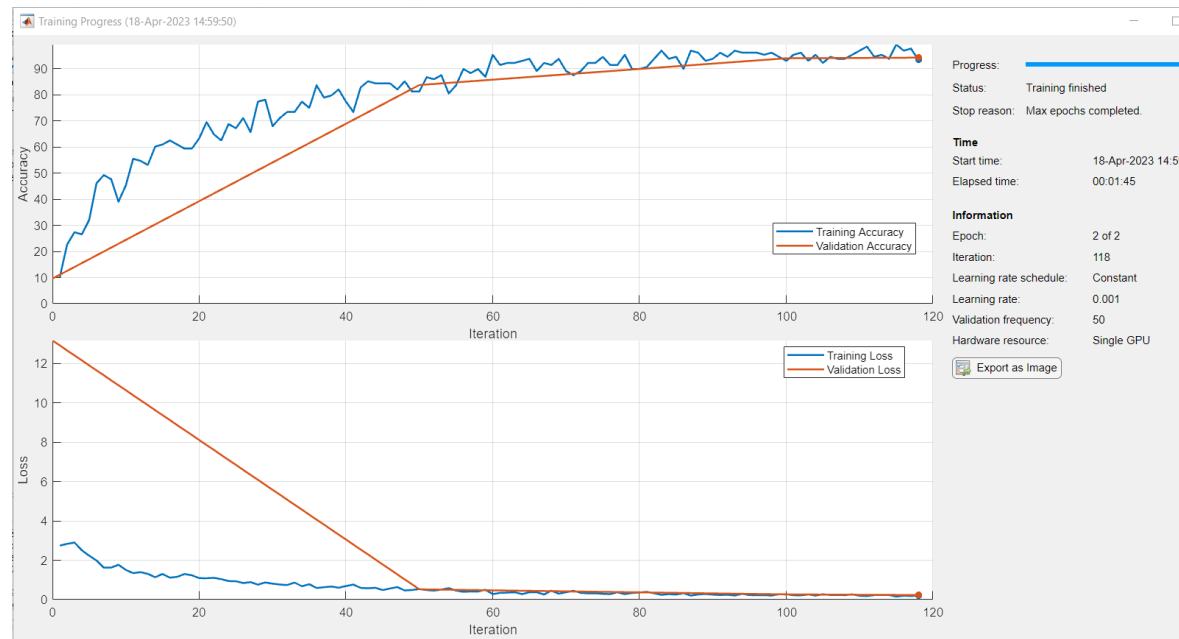
527 classes

Training Network



- trainnet function

```
>> options = trainingOptions('adam','Plots','training-progress');
>> net = trainnet(trainData, trainLabel, Model, LossFunction, options);
```

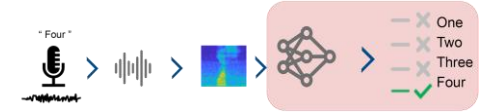


```
opts = trainingOptions('adam',...
    'InitialLearnRate',0.0001,...
    'Plot','training-progress',...
    'LossFunction','crossentropy',...
    'ValidationFrequency','Validation',...
    'ValidationFrequencyInEpochs',10,...
    'ValidationFrequencyInIteration',50);
```

"binary-crossentropy"

- "mse"
- "mean-squared-error"
- "l2loss"
- "mae"
- "mean-absolute-error"
- "l1loss"
- "huber"

Network Performance



True Class	one	222			2	6	96.5%	3.5%
	two		225		3	8	95.3%	4.7%
	three		1	235	1	11	94.8%	5.2%
	four	2			273	5	97.5%	2.5%
	unknown	19	10	12	15	1105	95.2%	4.8%

91.4%	95.3%	95.1%	92.9%	97.4%
8.6%	4.7%	4.9%	7.1%	2.6%

one	two	three	four	unknown
-----	-----	-------	------	---------

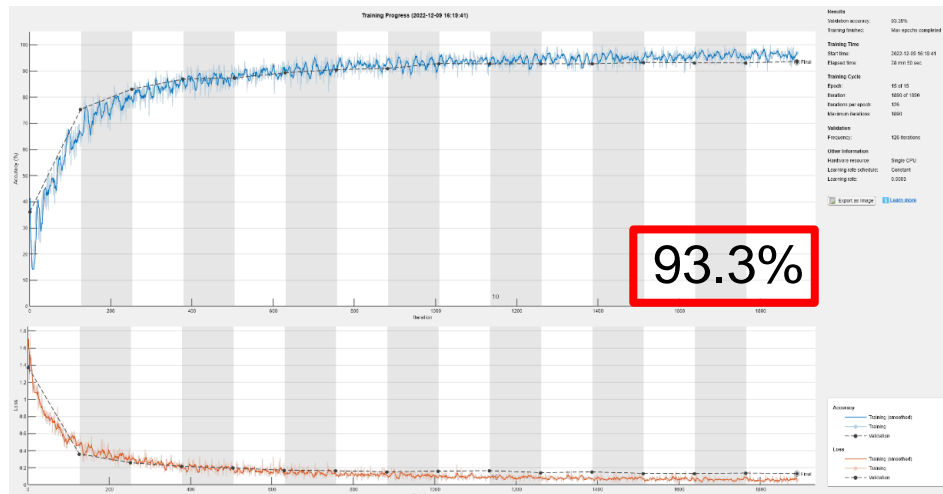
Predicted Class

True Class	one	198			2	9	94.7%	5.3%
	two		200	1	2	11	93.5%	6.5%
	three		2	214		9	95.1%	4.9%
	four	4			248	9	95.0%	5.0%
	unknown	3	3	5	2	1034	98.8%	1.2%

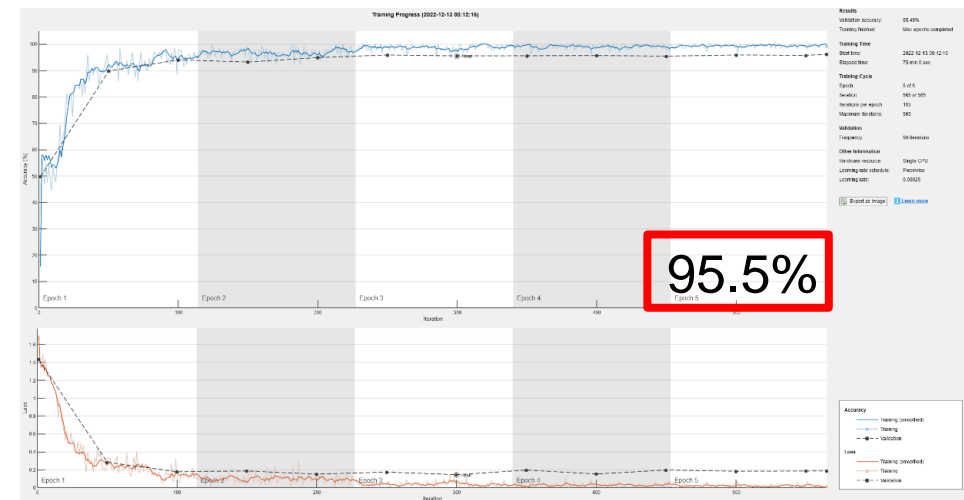
96.6%	97.6%	97.3%	97.6%	96.5%
3.4%	2.4%	2.7%	2.4%	3.5%

one	two	three	four	unknown
-----	-----	-------	------	---------

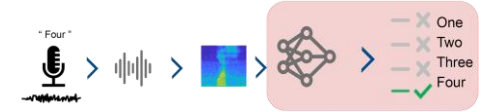
Predicted Class



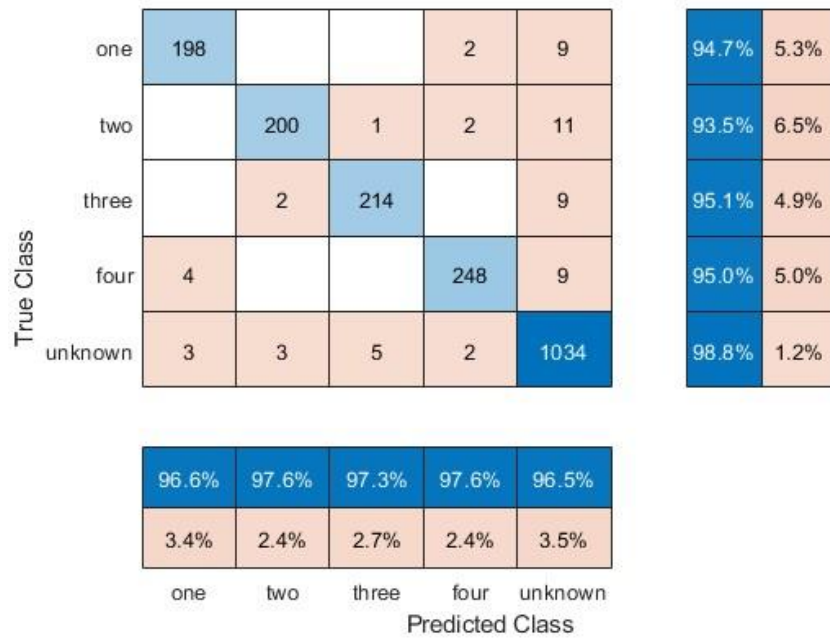
Training from scratch



Transfer Learning(VGGish)



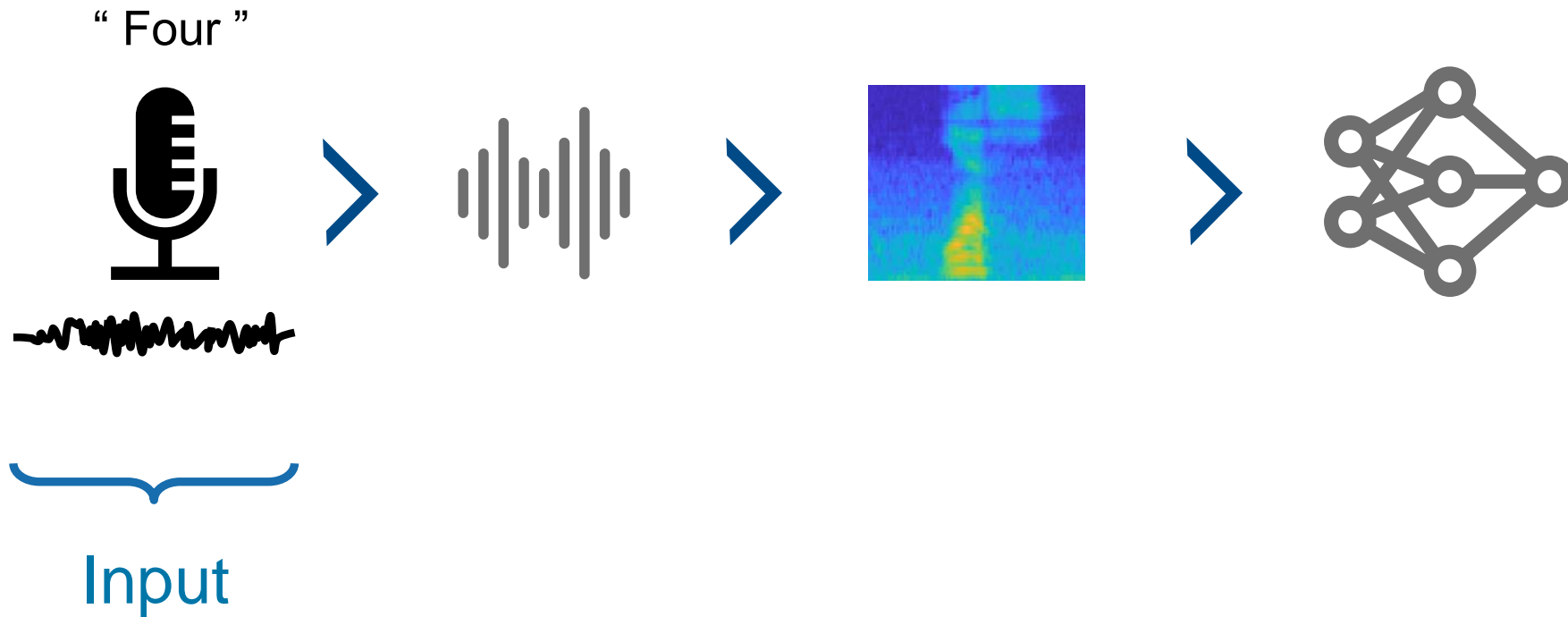
Confusion Chart



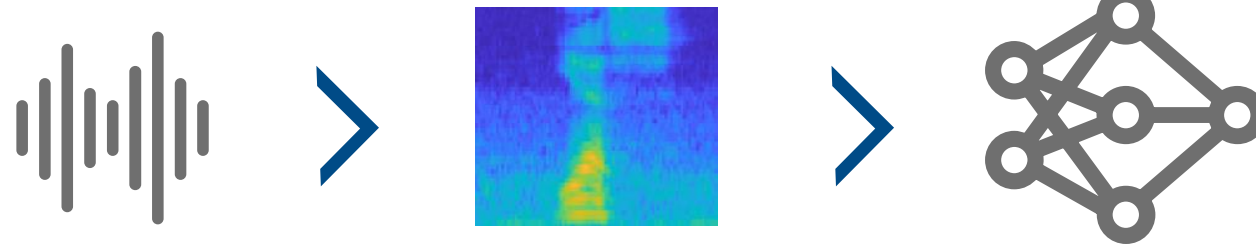
Performance : Confusion chart

```
cm = confusionchart(validLabel, predictedData, ...
    "ColumnSummary", "column-normalized", ...
    "RowSummary", "row-normalized");
```

Classification with Real Time Recording



Classification with Real Time Recording

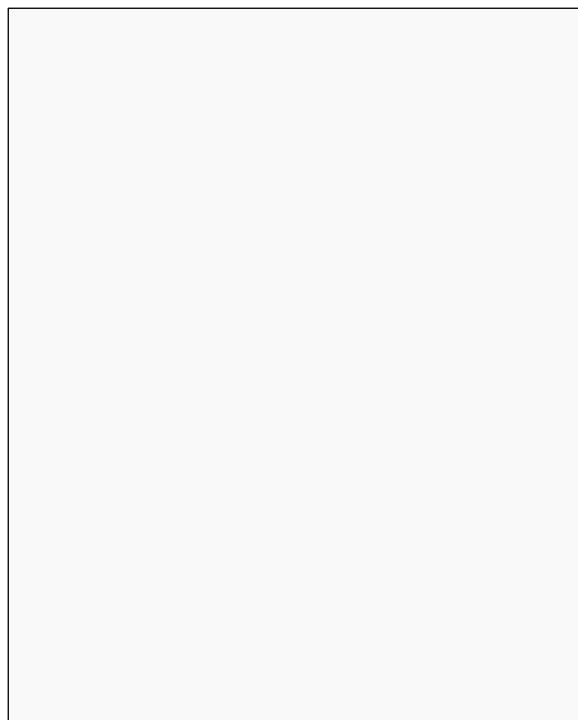


- × One
- × Two
- × Three
- ✓ Four

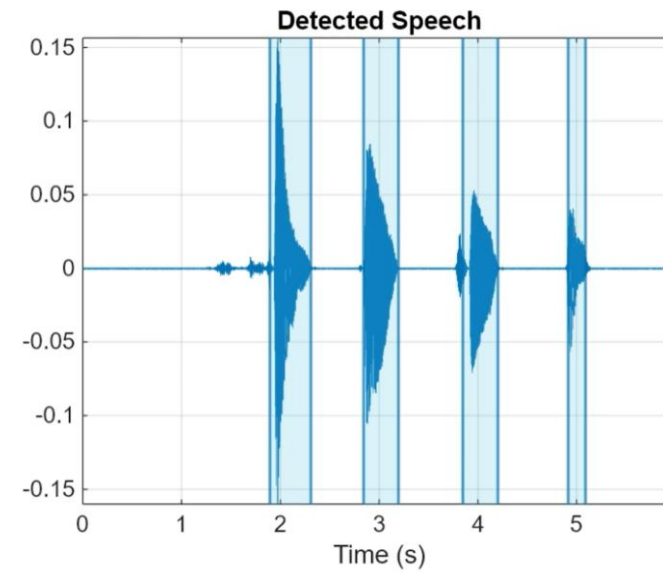
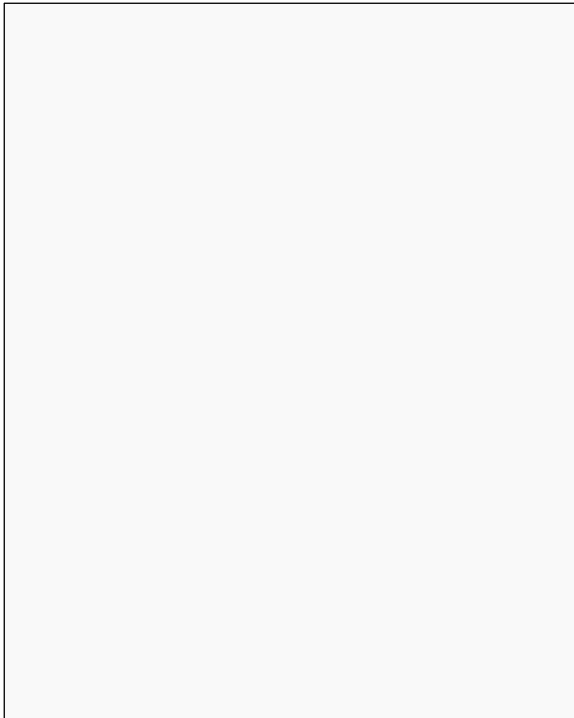


Classify

Demo



Demo

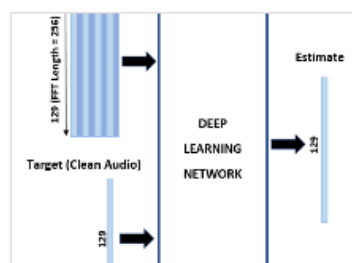


`predictedWord = "four three two one"`

Additional Information

Audio Toolbox Examples

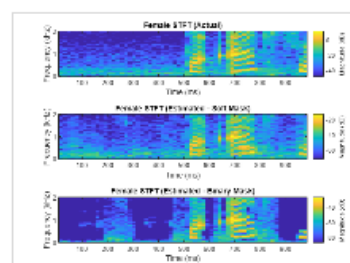
- 119 examples (R2024a)
 - AI for Audio
 - Audio Processing Algorithm Design
 - Measurements and Spatial Audio
 - ...



Denoise Speech Using Deep Learning Networks

Denoise speech signals using deep learning networks. The example compares two types of networks applied to the

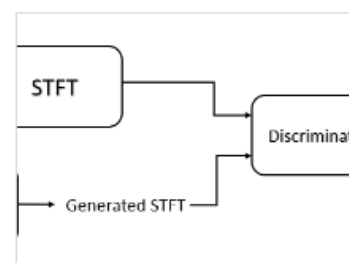
[Open Live Script](#)



Cocktail Party Source Separation Using Deep Learning Networks

Isolate a speech signal using a deep learning network.

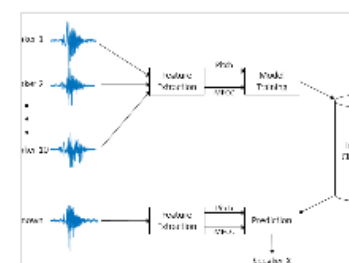
[Open Live Script](#)



Train Generative Adversarial Network (GAN) for Sound Synthesis

Train and use a generative adversarial network (GAN) to generate sounds.

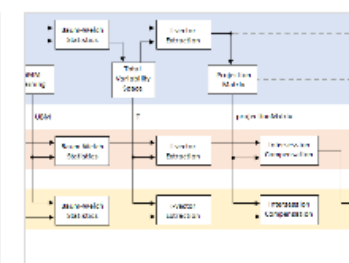
[Open Live Script](#)



Speaker Identification Using Pitch and MFCC

Use machine learning to identify people based on features extracted from recorded speech.

[Open Live Script](#)

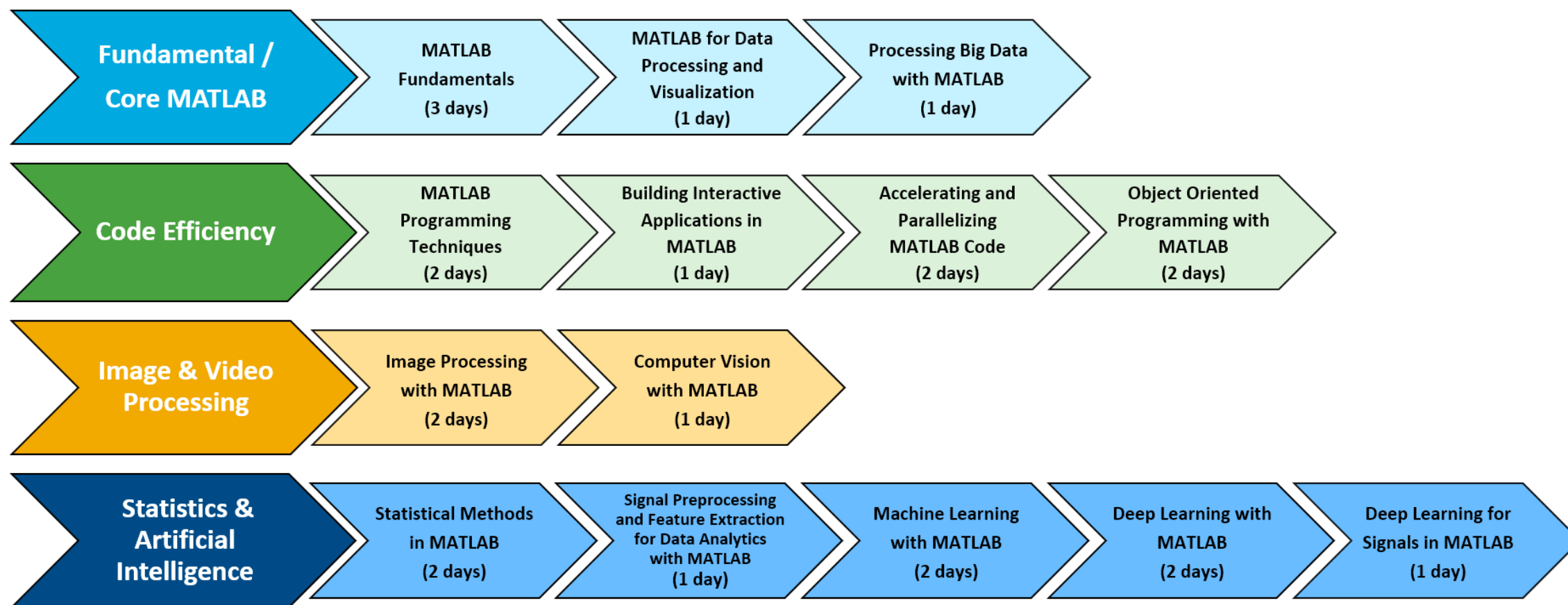


Speaker Verification Using i-Vectors

Speaker verification, or authentication, is the task of confirming that the identity of a speaker is who they purport

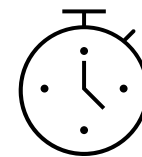
[Open Live Script](#)

AI and Data Science Training Curriculum



Training Courses for AI

- Machine Learning with MATLAB (2 days)
 - General data files(no time series)
- Deep Learning with MATLAB (2 days)
 - Image data files
- Deep Learning for Signals in MATLAB (1 day)
 - Time series data files



More Information



- 02-6006-5100
- training@mathworks.co.kr

MATLAB EXPO

Thank you



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

