Cough Segmentation

remove previous data

add yamnet to serach path

net =

SeriesNetwork with properties:

Layers: [86×1 nnet.cnn.layer.Layer]

InputNames: {'input\_1'}

OutputNames: {'Sound'}

86×1 Layer array with layers:

1 'input\_1' Image Input 96×64×1 images

2 'conv2d' Convolution 32 3×3×1 convolutions with stride [2 2] and padding 'same'

3 'b' Batch Normalization Batch normalization with 32 channels

4 'activation' ReLU ReLU

5 'depthwise\_conv2d' Grouped Convolution 32 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

6 'L11' Batch Normalization Batch normalization with 32 channels

7 'activation\_1' ReLU ReLU

8 'conv2d\_1' Convolution 64 1×1×32 convolutions with stride [1 1] and padding 'same'

9 'L12' Batch Normalization Batch normalization with 64 channels

10 'activation\_2' ReLU ReLU

11 'depthwise\_conv2d\_1' Grouped Convolution 64 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

12 'L21' Batch Normalization Batch normalization with 64 channels

13 'activation\_3' ReLU ReLU

14 'conv2d\_2' Convolution 128 1×1×64 convolutions with stride [1 1] and padding 'same'

15 'L22' Batch Normalization Batch normalization with 128 channels

16 'activation\_4' ReLU ReLU

17 'depthwise\_conv2d\_2' Grouped Convolution 128 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

18 'L31' Batch Normalization Batch normalization with 128 channels

19 'activation\_5' ReLU ReLU

20 'conv2d\_3' Convolution 128 1×1×128 convolutions with stride [1 1] and padding 'same'

21 'L32' Batch Normalization Batch normalization with 128 channels

22 'activation\_6' ReLU ReLU

23 'depthwise\_conv2d\_3' Grouped Convolution 128 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

24 'L41' Batch Normalization Batch normalization with 128 channels

25 'activation\_7' ReLU ReLU

26 'conv2d\_4' Convolution 256 1×1×128 convolutions with stride [1 1] and padding 'same'

27 'L42' Batch Normalization Batch normalization with 256 channels

28 'activation\_8' ReLU ReLU

29 'depthwise\_conv2d\_4' Grouped Convolution 256 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

30 'L51' Batch Normalization Batch normalization with 256 channels

31 'activation\_9' ReLU ReLU

32 'conv2d\_5' Convolution 256 1×1×256 convolutions with stride [1 1] and padding 'same'

33 'L52' Batch Normalization Batch normalization with 256 channels

34 'activation\_10' ReLU ReLU

35 'depthwise\_conv2d\_5' Grouped Convolution 256 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

36 'L61' Batch Normalization Batch normalization with 256 channels

37 'activation\_11' ReLU ReLU

38 'conv2d\_6' Convolution 512 1×1×256 convolutions with stride [1 1] and padding 'same'

39 'L62' Batch Normalization Batch normalization with 512 channels

40 'activation\_12' ReLU ReLU

41 'depthwise\_conv2d\_6' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

42 'L71' Batch Normalization Batch normalization with 512 channels

43 'activation\_13' ReLU ReLU

44 'conv2d\_7' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

45 'L72' Batch Normalization Batch normalization with 512 channels

46 'activation\_14' ReLU ReLU

47 'depthwise\_conv2d\_7' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

48 'L81' Batch Normalization Batch normalization with 512 channels

49 'activation\_15' ReLU ReLU

50 'conv2d\_8' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

51 'L82' Batch Normalization Batch normalization with 512 channels

52 'activation\_16' ReLU ReLU

53 'depthwise\_conv2d\_8' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

54 'L91' Batch Normalization Batch normalization with 512 channels

55 'activation\_17' ReLU ReLU

56 'conv2d\_9' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

57 'L92' Batch Normalization Batch normalization with 512 channels

58 'activation\_18' ReLU ReLU

59 'depthwise\_conv2d\_9' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

60 'L101' Batch Normalization Batch normalization with 512 channels

61 'activation\_19' ReLU ReLU

62 'conv2d\_10' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

63 'L102' Batch Normalization Batch normalization with 512 channels

64 'activation\_20' ReLU ReLU

65 'depthwise\_conv2d\_10' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

66 'L111' Batch Normalization Batch normalization with 512 channels

67 'activation\_21' ReLU ReLU

68 'conv2d\_11' Convolution 512 1×1×512 convolutions with stride [1 1] and padding 'same'

69 'L112' Batch Normalization Batch normalization with 512 channels

70 'activation\_22' ReLU ReLU

71 'depthwise\_conv2d\_11' Grouped Convolution 512 groups of 1 3×3×1 convolutions with stride [2 2] and padding 'same'

72 'L121' Batch Normalization Batch normalization with 512 channels

73 'activation\_23' ReLU ReLU

74 'conv2d\_12' Convolution 1024 1×1×512 convolutions with stride [1 1] and padding 'same'

75 'L122' Batch Normalization Batch normalization with 1024 channels

76 'activation\_24' ReLU ReLU

77 'depthwise\_conv2d\_12' Grouped Convolution 1024 groups of 1 3×3×1 convolutions with stride [1 1] and padding 'same'

78 'L131' Batch Normalization Batch normalization with 1024 channels

79 'activation\_25' ReLU ReLU

80 'conv2d\_13' Convolution 1024 1×1×1024 convolutions with stride [1 1] and padding 'same'

81 'L132' Batch Normalization Batch normalization with 1024 channels

82 'activation\_26' ReLU ReLU

83 'global\_average\_pooling2d' Global Average Pooling Global average pooling

84 'dense' Fully Connected 521 fully connected layer

85 'softmax' Softmax softmax

86 'Sound' Classification Output crossentropyex with 'Speech' and 520 other classes

Create audioDatastore object of the data and split it into train and validation sets

Create audioFeatureExtractor object from audio signals

**win\_length** **overlap** **num\_spectrums** **segment\_length**

**\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**samples** 128 0.5 96 6208

**sec**  0.008 NaN NaN 0.388

Extract features from train and validation sets

100%

100%

cough segmentation requires only 2 classes (cough / non-cough).

Read in YAMNet, convert it to a [layerGraph](docid:nnet_ref#mw_399d64ed-680b-46f0-af77-0472ba1910d8), and then replace the final [fullyConnectedLayer](docid:nnet_ref#mw_1e7fbc56-4746-4f30-8cd9-7048ce806a0d) and the final [classificationLayer](docid:nnet_ref#bu5lho8) to reflect the new task.

Define training options

Train and save network

Training on single GPU.

|======================================================================================================================|

| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |

| | | (hh:mm:ss) | Accuracy | Accuracy | Loss | Loss | Rate |

|======================================================================================================================|

| 1 | 1 | 00:00:08 | 64.06% | 72.50% | 0.6833 | 0.6009 | 0.0003 |

| 1 | 28 | 00:00:30 | 97.27% | 94.64% | 0.1036 | 0.1636 | 0.0003 |

| 2 | 50 | 00:00:46 | 98.05% | | 0.0689 | | 3.0000e-05 |

| 2 | 56 | 00:00:52 | 96.88% | 95.41% | 0.0884 | 0.1616 | 3.0000e-05 |

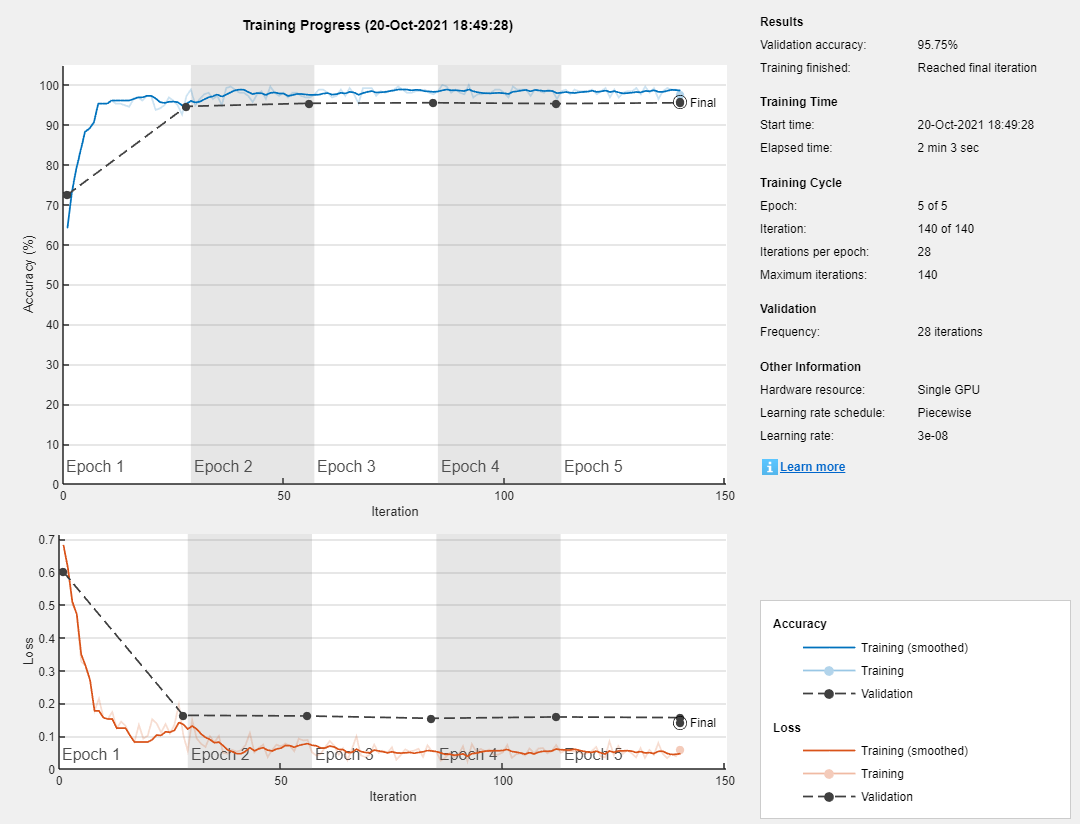
| 3 | 84 | 00:01:15 | 97.66% | 95.54% | 0.0545 | 0.1545 | 3.0000e-06 |

| 4 | 100 | 00:01:27 | 98.05% | | 0.0513 | | 3.0000e-07 |

| 4 | 112 | 00:01:38 | 96.48% | 95.32% | 0.0730 | 0.1587 | 3.0000e-07 |

| 5 | 140 | 00:02:01 | 97.66% | 95.54% | 0.0608 | 0.1567 | 3.0000e-08 |

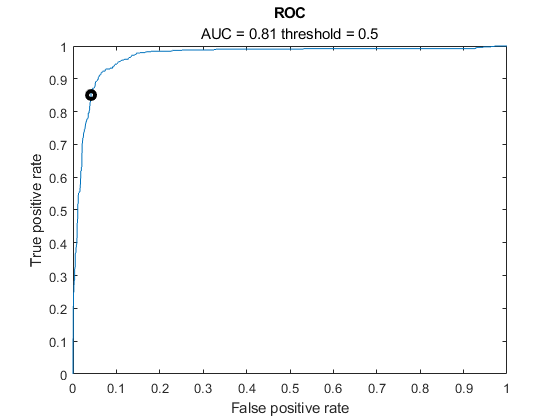
|======================================================================================================================|

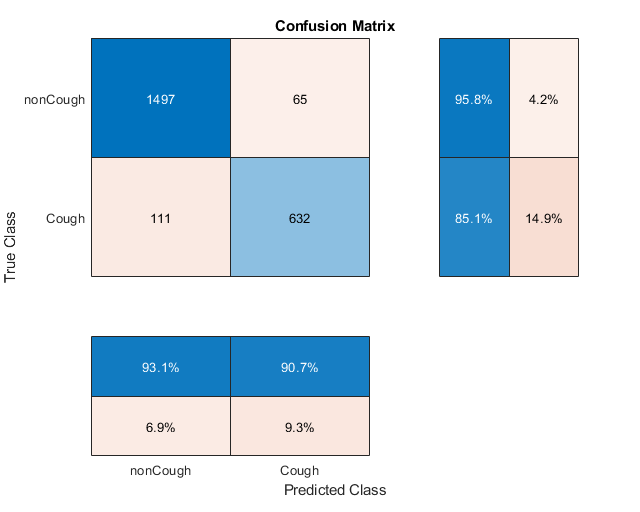


evaluate the performance of the network on the test set using:

* ROC & AUC
* confusion matrix
* accuracy, sensetivity, PPV, F1-score

100%





**accuracy [%]** **sensitivity [%]** **PPV [%]** **F1\_score [%]**

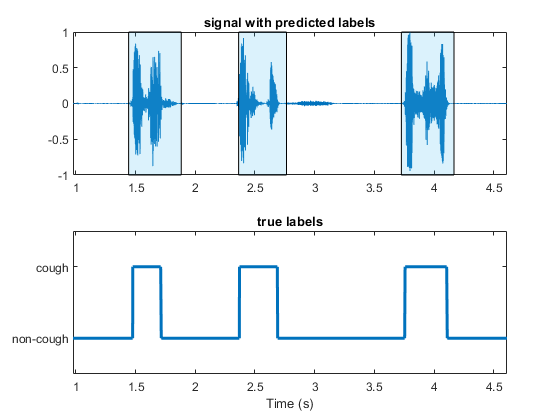
**\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_**

92.4 85.1 90.7 87.8

Save the trained network

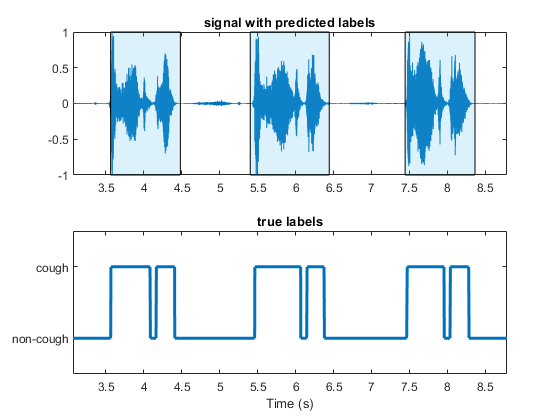
plot segmentation results on validation data set

100%



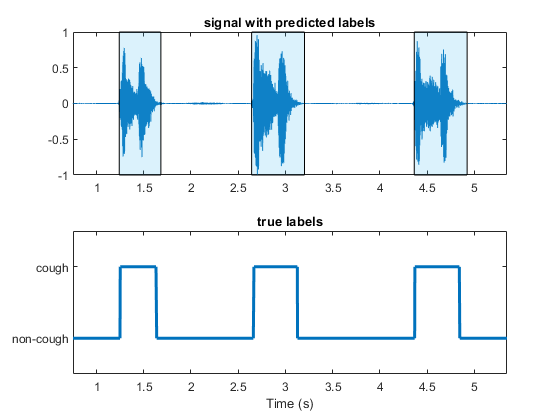
1

3



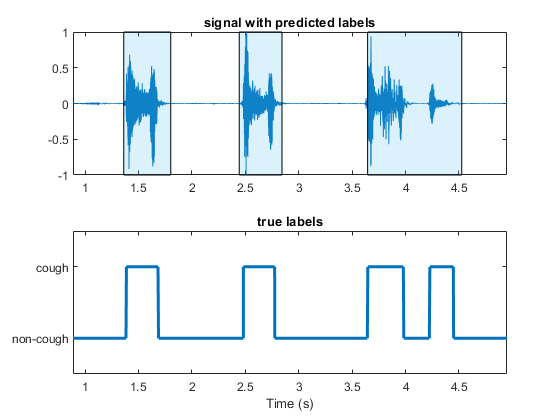
2

3



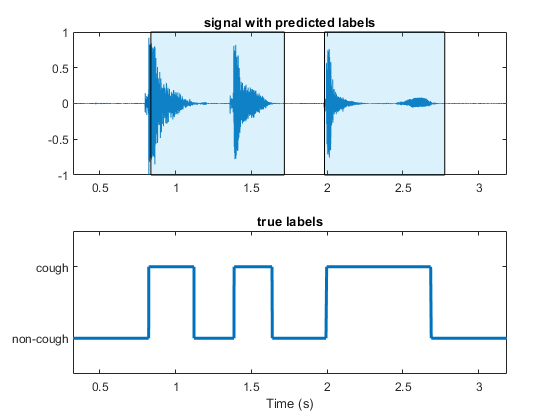
3

3



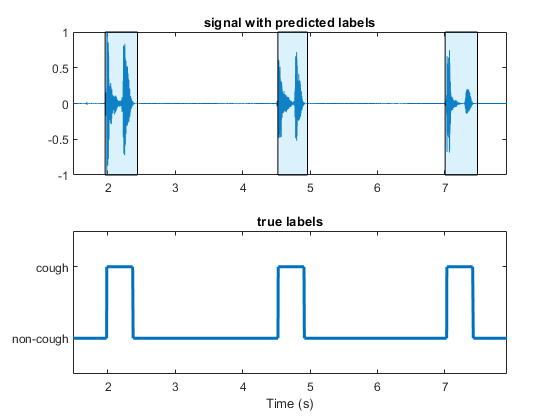
4

3



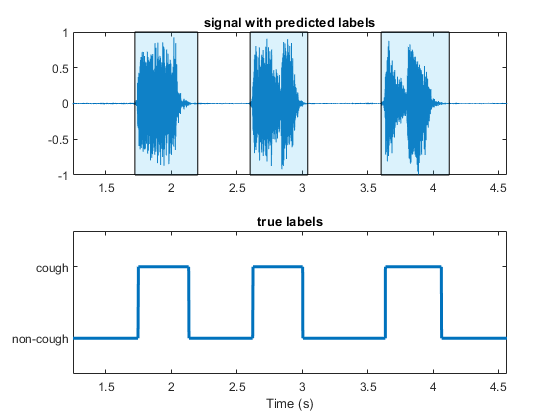
5

2



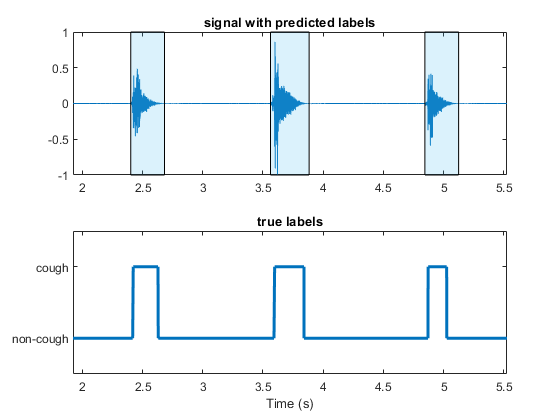
6

3



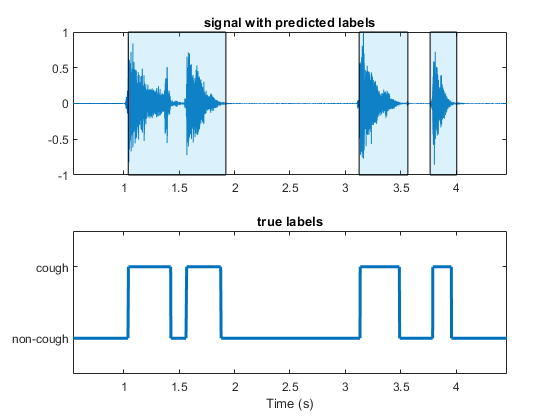
7

3



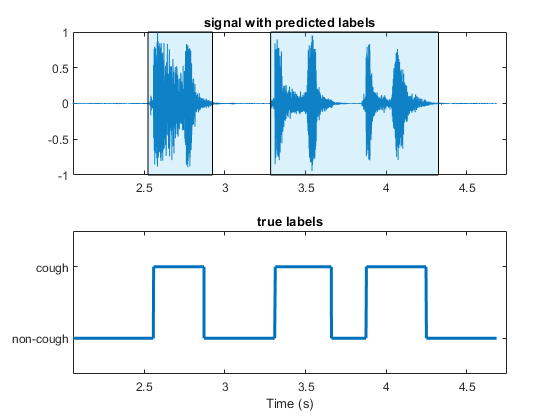
8

3



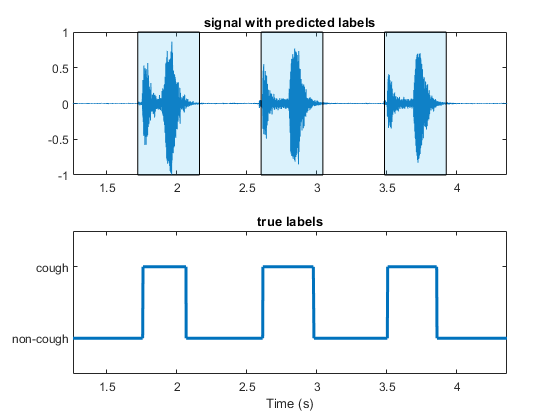
9

3



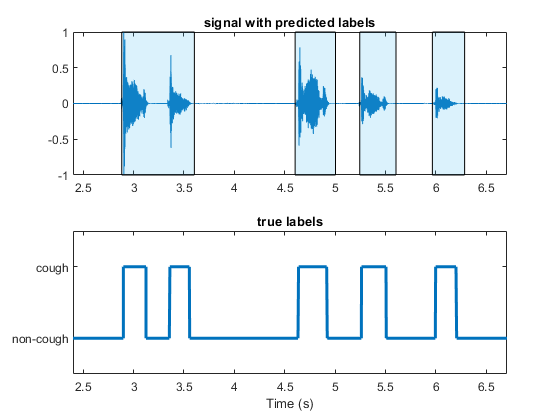
10

2



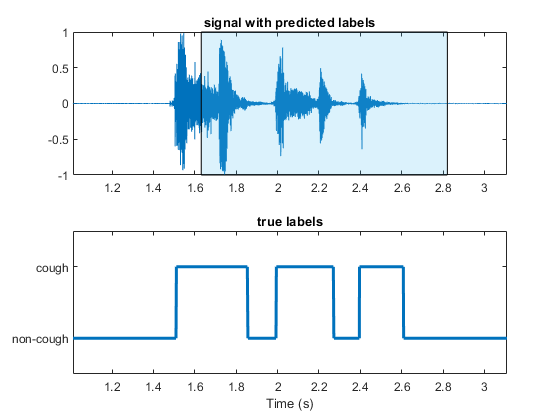
11

3



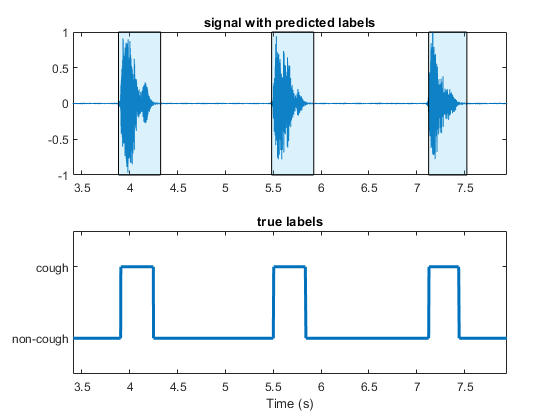
12

4



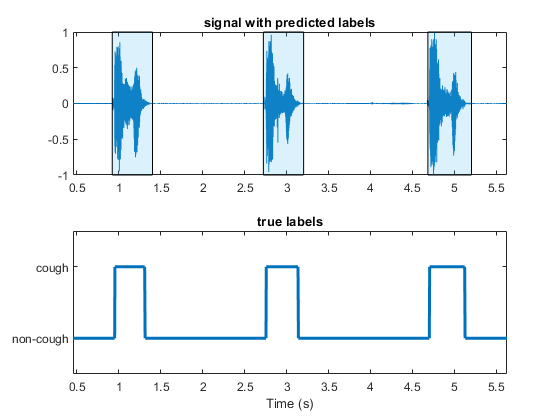
13

1



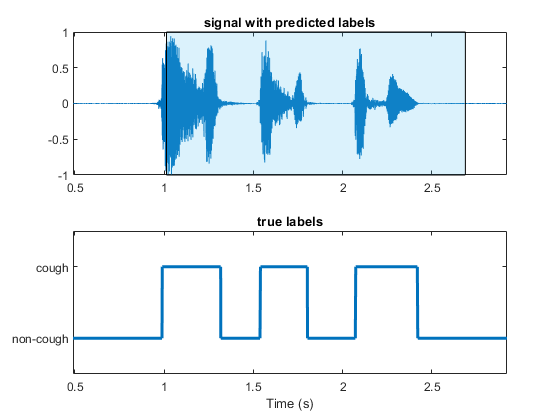
14

3



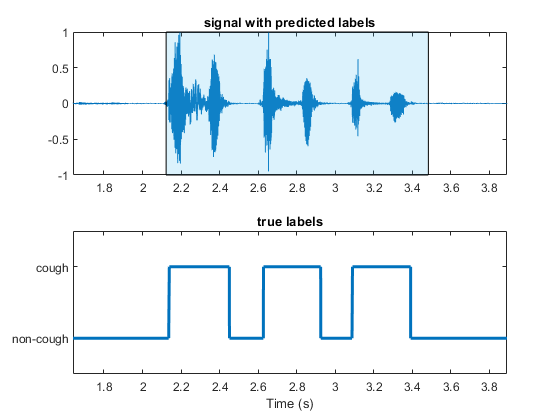
15

3



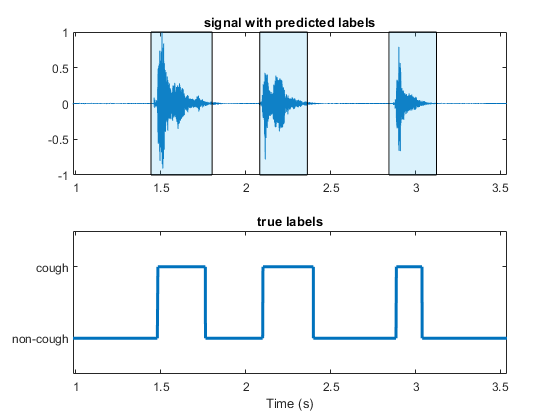
16

1



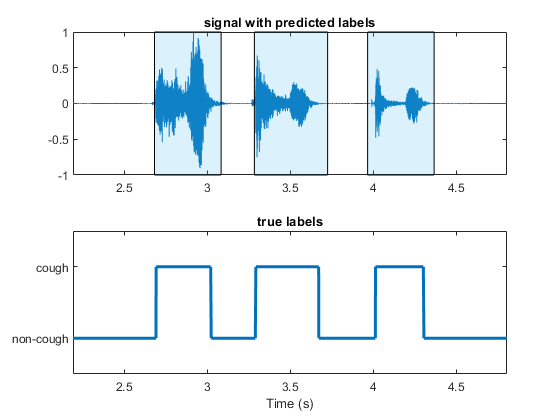
17

1



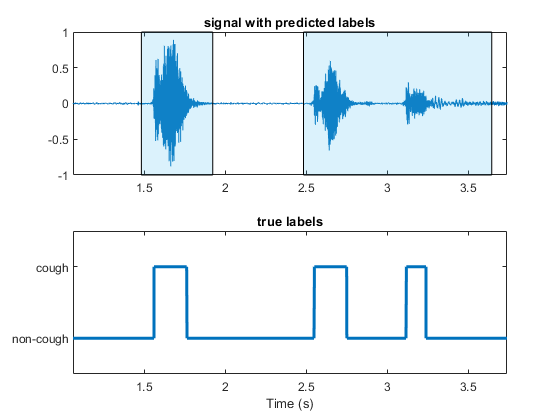
18

3



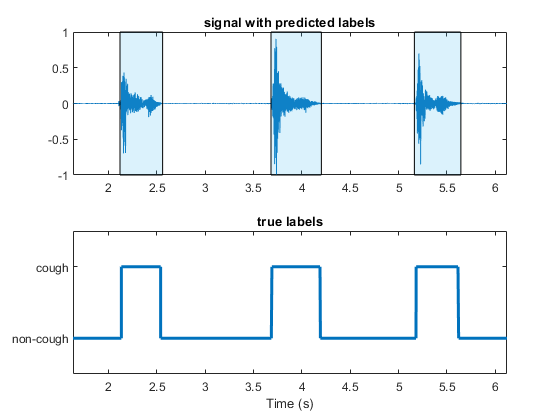
19

3



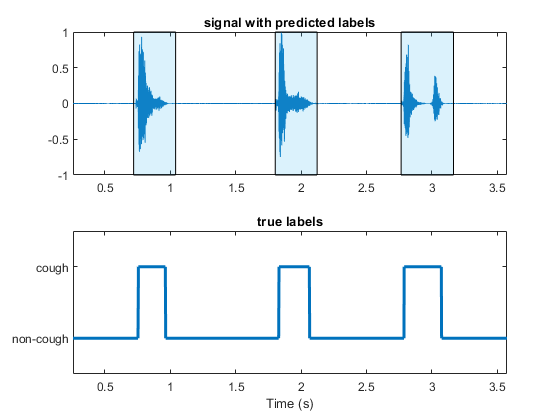
20

3



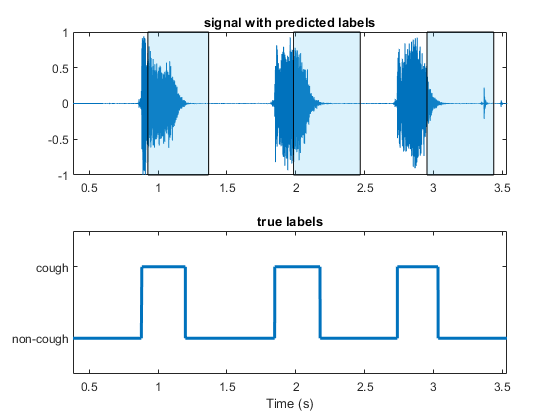
21

3



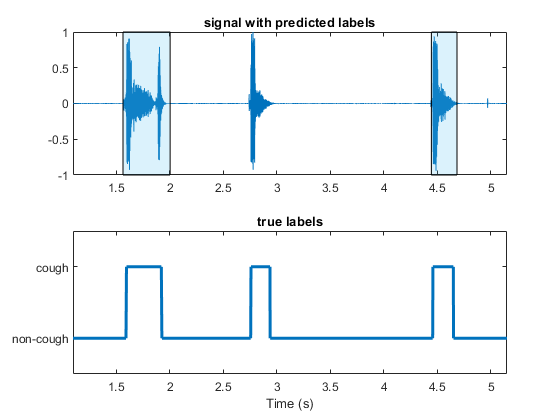
22

3



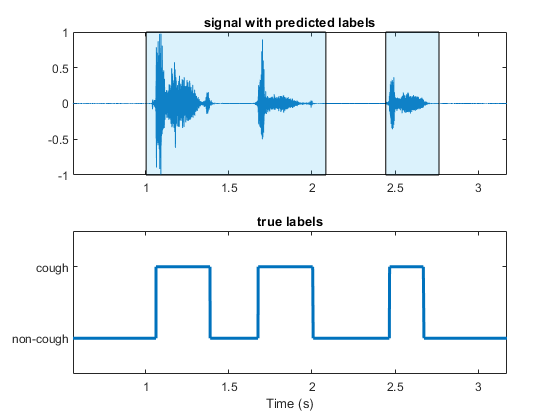
23

3



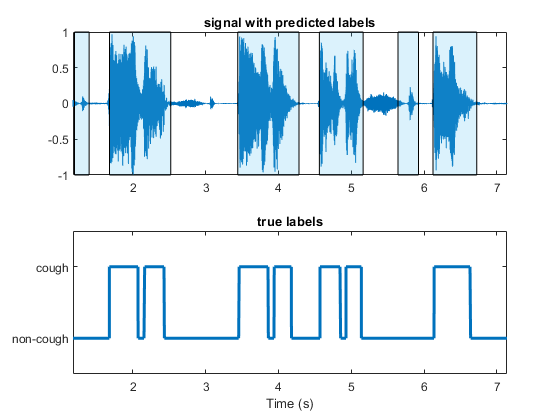
24

2



25

2



26

6

No. coughs = 72