Programmer’s manual

The thesis files and codes were processed using MATLAB 2020a/2021a and are located at the ‘Corona’ folder on our lab-server. The specific location is: \\lab-server\Corona. This folder is divided into 2 subfolders: Database and Codes.

Table of Contents

1. Database

Database Location: \\lab-server\Corona\DataBase.

This section contains all the datasets that were used in this work. They are all arranged by subfolder as follows:

Coswara

Link to download the dataset: https://github.com/iiscleap/Coswara-Data

Contains the Coswara dataset. The raw data is in “Coswara-dataset-9.8.22-tar.gz” folder, and in tar.gz format which requires extraction. **The extracted WAV files** are in folder: “Coswara-dataset-9.8.22”. manual annotations of coughing and breathing events are located in “true seg labels”. The folder: “audioDataFolder” contains the data after pre-processing and arranged in separated subfolders based on the COVID-19 status (positive/negative).

The code for pre-processing and rearrangement of the audio files is in: “loadDataCoswara.m”, located at: \\lab-server\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\coswara\load data.

COUGHVID

Link to download the dataset: https://zenodo.org/records/7024894

Contains the COUGHVID dataset. The raw data is located at: “coughvid-dataset-03-02-21”. The raw data consist of OGG and WEBM files. MATLAB software cannot load WEBM files so those files needed to be transformed into WAV files. A code to transform the audio files from WEBM to WAV is in: “ConvertWebm2Wav”, and all audio files in WAV/OGG formats are located in: “coughvid-dataset-03-02-21-WAV+OGG”. The folder: “audioDataFolderVer1” contains the data after pre-processing and arranged in separated subfolders based on the COVID-19 status (positive/negative).

The code for pre-processing and rearrangement of the audio files is in: “loadDataCoughvid.m”, located at: \\lab-server\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\coughvid\load data.

Voca

Contains the Voca dataset. The raw data is located at: “voca-dataset-09-09-20”. The folder: “audioDataFolder” contains the data after pre-processing and arranged in separated subfolders based on the COVID-19 status (positive/negative).

The code for pre-processing and rearrangement of the audio files is in: “loadDataVoca.m”, located at: \\lab-server\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\voca\load data.

2. Codes

All the codes that were used in this research are located at: \\lab-server\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing. They are organized in this manual based on the following: 1) loading and pre-processing the raw audio files. 2) RCNN-based system comprises of cough event detection and COVID-19 classification.

loading and pre-processing the raw audio files

Each dataset was loaded and pre-processed at a different folder, each named by the corresponding dataset (datasetName = “COUGHVID” | “Coswara” | “Voca”). For each datasetName, the name of the main function is: “loadDatadatasetName”, located at: \\lab-server\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\datasetName\load data. This function performed the following:

- Load the data from the corresponding original folder.
- Remove subjects based on the subjects’ exclusion criteria.
- Copy all the remaining audio files to: “audioDataFolder” (same dataset folder) in two separated subfolders (COVID-19 positive/negative).
- Create a CSV file that contains all the metadata. This file is used for further processing to find the locations of all the audio files.

For each dataset, the following sub-functions were used:

File name (.m)	Description
pre_process_audio_sig	Pre-process the audio signal: down-sampling, DC removal, amplitude normalization and spectral subtraction.
SSBOLL79	The algorithm for spectral subtraction, based on the paper of boll.

Coughing and breathing manual annotations are located at: \\lab-server\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\datasetName\true seg labels.

RCNN-based system: cough detection and COVID-19 classification

The codes for **cough event detection** and segmentation are in folder: “coughEventDetectionYamnet”. The codes for **COVID-19 classification using the CNN-VGGish** model are in folder: “classify-CNN”. The codes for **COVID-19 classification using the RCNN** model are in folder: “classify-CRNN”.

Each folder contains the following main functions:

File name	Description

1	tuneHyperparametersYamnet tuneHyperparametersCNN tuneHyperparametersCRNN	Main code to find optimal hyper-parameters. Method: repeated random sub-sampling validation.
2	mainCoughDetectYamnet mainClassifyCnn mainClassifyCrnn	Main code to train and test the model over 1 iteration and observe training progress and results.
3	mainCoughDetectYamnetLoop mainClassifyCnnLoop mainClassifyCrnnLoop	Main code to train and test the model over several iterations (instead of **2**), and observe mean results.

For COVID-19 classification using RCNN and CNN models, the following sub-functions were used:

	File name (.m)	Description
1	CalculateEmpiricalAUC	Method to calculate the AUC. Method: DeLong's Test.
2	calculateScores	Calculate the performance measurements: Accuracy, UAR, F1, Sensitivity, PPV, Specificity, AUC.
3	calculateScoresPerAgeGender	Same as **2** but divided per age and gender.
4	calculateScoresPerDataset	Same as **2** but divided per dataset.
5	calculateScoresPerSymptoms2classes	Same as **2** but divided per symptoms into 2 classes: Symptomatic / Asymptomatic.
6	calculateScoresPerSymptoms4classes	Same as **2** but divided per symptoms into 4 classes: 1) Positive with symptoms. 2) Positive without symptoms. 3) Negative with symptoms. 4) Negative without symptoms.
7	dispProgressAndTime	Display the elapsed and remaining time to finish running the code.
8	extractClassifyScores	Connects the main function to findBestThresholdAndPlotROC and calculateScores functions and display the results.
9	extractNetFromCheckpoint	After model training, find the best epoch

		(Lowest cost on val set).
10	extrctWinInfo	Extract information regarding the frames and spectrograms (length, overlaps, zero padding for FFT etc.).
11	findBestThresholdAndPlotROC	Find the optimal threshold on the ROC curve.
12	loadCnnLgraph	Load the pre-trained CNN model (YAMNet, VGGish, OpenL3).
13	loadSplitShuffleDatasets	Load the CSV metadata file of each dataset. Combine all into 1 unified metadata. Separate and shuffle the subjects in the combined metadata.
14	perAudio2perSubject	Transform the results from per audio signal to per subject.
15	perSegment2perAudio	Transform the results from per segment to per audio signal.
16	preprocessClassify	Transform the audios signals into spectrograms segments.
17	selectGpu	Check if any GPU in server 5 is available and select it.
18	setAudioFeatureExtractor	Set parameters based on extrctWinInfo for transformation of each audio signal into spectrogram.

For cough event detection and segmentation, the following sub-functions were used:

File name (.m)	**Description**
erosion_dilation_noise_removal	Remove short intervals (presumably non-cough) using erosion and dilation operations.
extractPredictions	Extract the trained model prediction from a new dataset.
extractScoresCoughDetectYamnet	Using the predictions, calculate the performance measurements.
loadNewAds	Given a dataset’s name, load the data.
loadSegTrueLblsYamnet	Load the manual annotations of cough events timestamps.
plotBar	Plot a bar that contains the segmentation results segment-by-segment and per overlap between the detected and true cough events locations.

plotSigDec	Plot an audio signal with segmentation (detected and true labelled).
predictOnNewAds	Given a dataset’s name, make cough detection on the dataset.
preprocessSegYamnet	Pre-processing: transformation into spectrogram segments.
seg_comp_per_event	Compare between the detected and true labelled cough events.

The following sub-functions are the same as in the RCNN and CNN classification folders: CalculateEmpiricalAUC, dispProgressAndTime, extractNetFromCheckpoint, extractScores, findBestThresholdAndPlotROC, loadSplitShuffleDatasets, selectGpu, setAudioFeatureExtractor.