

# ***Programmer's manual***

The thesis files and codes were processed using MATLAB 2020a/2021a and are located at the 'Corona' folder on zigel-server-5. The specific location is: <\\zigel-server-5.auth.ad.bgu.ac.il\\Corona>. This folder is divided into three subfolders: Database Codes, and Documents.

## ***Table of Contents***

Programmer's manual.....	1
1. Database.....	1
2. Codes.....	3
3. Documents .....	8
.4 Appendix A - Coswara dataset info and downloading manual.....	9
5. Appendix B – Audacity software.....	10

### ***1. Database***

Server Location: <\\zigel-server-5.auth.ad.bgu.ac.il\\Corona\\DataBase>.

PC Location: Zigel-Lab-25 (Nir's PC, room 103)

+ Yaniv's external disk

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

## Coswara

Contains the Coswara dataset. The raw data is in “Coswara-dataset-9.8.22-tar.gz” folder, and in tar.gz format which requires a specific extraction method. A manual for download and extraction of the raw data is provided in appendix A, and in folder: “coswara data information and downloading manual”. **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: [\\zigel-server-5.auth.ad.bgu.ac.il\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\coswara\load data](https://zigel-server-5.auth.ad.bgu.ac.il/Corona/Codes/Covid-19_diagnosis_using_cough_and_breathing_audio_signal_processing/coswara/load_data).

## COUGHVID

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: [\\zigel-server-5.auth.ad.bgu.ac.il\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\coughvid\load data](https://zigel-server-5.auth.ad.bgu.ac.il/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: [\\zigel-server-5.auth.ad.bgu.ac.il\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\voca\load data](https://zigel-server-5.auth.ad.bgu.ac.il/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: <\\zigel-server-5.auth.ad.bgu.ac.il\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. 3) SVM-based system.

### 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: “loadData`datasetName`”, located at: <\\zigel-server-5.auth.ad.bgu.ac.il\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: <\\zigel-server-5.auth.ad.bgu.ac.il\Corona\Codes\Covid-19 diagnosis using cough and breathing audio signal processing\datasetName>true seg labels>. For further information regarding the manual annotations, please refer to appendix B - Audacity software.

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

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

	File name (.m)	Description
<b>1</b>	CalculateEmpiricalAUC	Method to calculate the AUC. Method: DeLong’s Test.
<b>2</b>	calculateScores	Calculate the performance measurements: Accuracy, UAR, F1, Sensitivity, PPV, Specificity, AUC.
<b>3</b>	calculateScoresPerAgeGender	Same as <b>2</b> but divided per age and gender.
<b>4</b>	calculateScoresPerDataset	Same as <b>2</b> but divided per dataset.
<b>5</b>	calculateScoresPerSymptoms2classes	Same as <b>2</b> but divided per symptoms into 2 classes: Symptomatic / Asymptomatic.
<b>6</b>	calculateScoresPerSymptoms4classes	Same as <b>2</b> 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.
loadSegTrueLbIsYamnet	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.

## SVM-based system

The codes for COVID-19 classification using the SVM-based system are in folder: "SVM approach". There, the following sub-folders are:

Folder name	Description
auto-segmentation	The algorithm for the detection and segmentation of first and second phase of cough events.
feature selection + classification SVM	COVID-19 classification using an SVM classifier and feature selection during training.
feature-extraction	Extract 34 features from each phase of cough event.
figures	Figures presenting selected features.
manual-segmentation	Use manual annotations of phases 1&2 of cough events for the classification of COVID-19.
results	CSV files, containing the results for COVID-19.



### 3. Documents

Location: <\\zigel-server-5.auth.ad.bgu.ac.il\Corona\documents\Nir Iakoby>

This section contains all the documents of this work. They are all arranged by subfolder as follows:

Folder name	Description
ISF 2021	Application request to the ISF for a research grant in the subject of: "גילוי מרחוק של סימפטומים של קורונה בעזרת אותות שמע: דיבור, שיעול, וצלילי נשימה"
my article 2022-2023 - COVID-19 detection from Cough using RCNN	All information regarding my article.
network models	Figures. Implemented using 'draw io' software.
הצעת מחקר ודוחות התקדמות	Contains research proposal and two progress reports.
השוואות סטטיסטיות בין מסווגים	Information regarding how to perform statistical significance tests when comparing between classification models over several iterations.
כנסים	Information regarding 4 conferences I attended: 2 EMBCs (43th and 44th), IEEE EMBS-TAU 2022 and of the biomedical department at Ben Gurion university (2022).
כתיבת עבודת גמר לתואר 2	Contains all information regarding my thesis report.
מאמרים	All the papers I used for my research.
סמינר מחלקתי	Presentation about my research for the biomedical department.
שיחות עם יניב	All the content from my meetings with my supervisor, Dr. Yaniv Zigel.



## 4. Appendix A - Coswara dataset info and downloading manual

אתר מקור: כל הנתונים נלקחו מאתר GitHub ("https://github.com/iiscleap/Coswara-Data") כחלק מפרויקט Coswara. מומלץ להתעדכן באתר כי כל הזמן מעלים עוד נתונים.

סוגי קבצים: כל תיקייה מכילה אותות שמע מנבדק בודד. אותות אלו מחולקים בקבצים נפרדים ל:

1. נשימה עמוקה.
2. נשימה רדודה.
3. שיעול כבד.
4. שיעול רדוד.
5. ספירה בקצב מהיר.
6. ספירה בקצב רגיל.
7. "aaaa"
8. "eeee"
9. "oooo"

Metadata: מכיל פירוט על כל נבדק באופן הבא:

1. Id.
2. מדינה.
3. גיל.
4. מצב רפואי: בריא, ללא בעיות נשימתיות – נחשף, מאובחן חיובי – מצב קל, מאובחן חיובי – מצב בינוני, נרפא, בעיות נשימתיות – ללא זיהוי.
5. ?
6. מגדר.
7. ?
8. עיר.
9. בעיות רפואיות: סכרת, אסטמה, עישון, ht, חום, cld, ihd, rU, קור, שיעול, איבוד חוש ריח, דלקת ריאות, um, .ftg, bd, test, mp, st

קריאת קבצים: כל הקבצים בפורמט – tar.gz.aa/ab/ac..

פתיחת קבצים אלו מתבצעת ב2 שלבים:

1. איחוד של כל קבוצת קבצים בתוך תיקייה אחת המסודרים לפי aa, ab, ac.. לקובץ מכוון מסוג WinRAR או .zip.
2. פתיחת הקובץ המכוון לפי תוכנת כיוון סטנדרטית כגון WinRAR.

איחוד של הקבצים:

1. בחיפוש ב-toolbar רשום ופתח command prompt.
2. שנה מיקום לתיקייה הרלוונטית בה נמצאים קבוצת קבצים ספציפיים אותם צריך לאחד. זאת ע"י רישום cd ואחריו רווח, ולאחר מכן רישום כל התיקיות עד לרלוונטיות עם קו נטוי ביניהם. לדוגמה: cd project/CosWara/unorganized data
3. רשום: type \*.tar.gz.\* > file\_name.tar.gz. במקום "file name" יש לרשום את שם הקבצים (צריך להיות מספר).
4. בסיום הפעולה צריך להתקבל קובץ מכוון סטנדרטי.

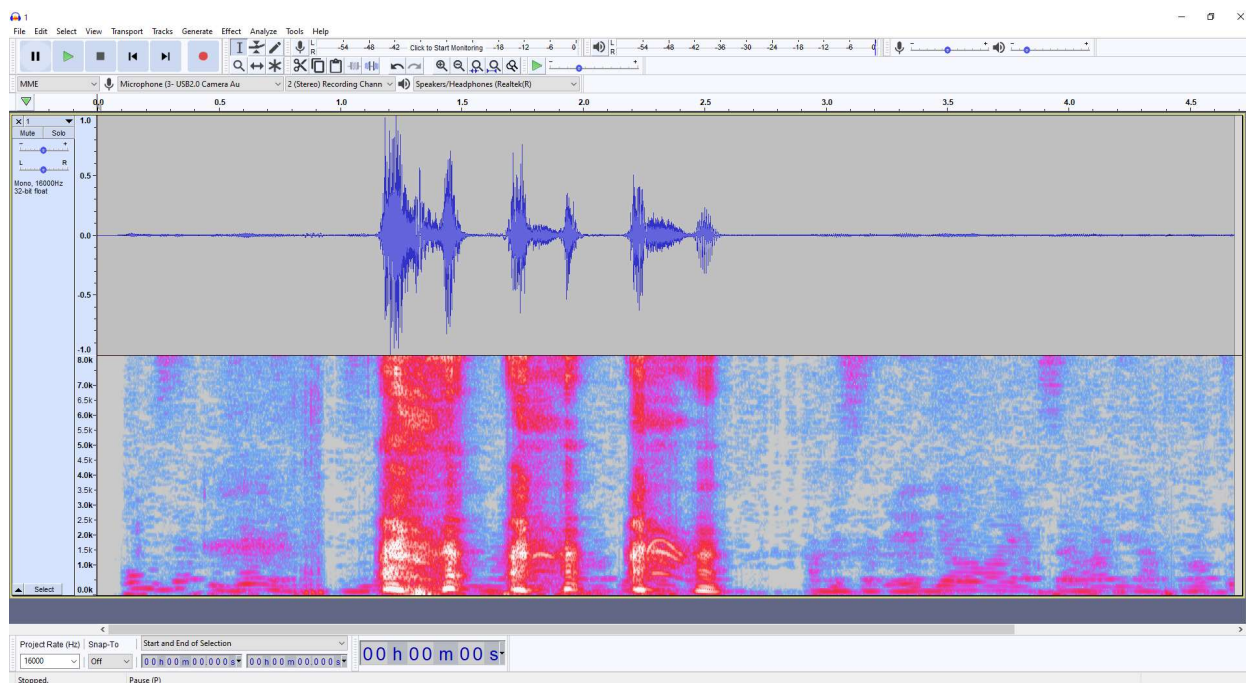
## 5. Appendix B – Audacity software

הסבר על הליך ביצוע סגמנטציה ידנית של אותות שיעול ונשימה

שם תוכנה: Audacity.

תוכנה חינמית. קישור להורדה: [/https://www.audacityteam.org/download](https://www.audacityteam.org/download)

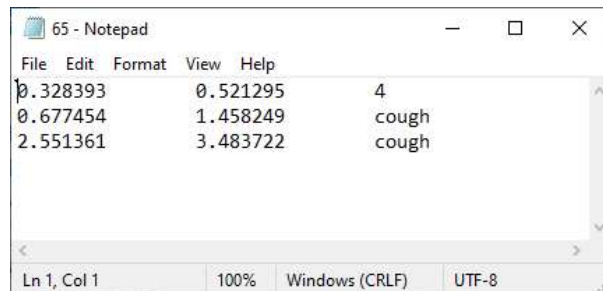
דוגמה לתצוגה עבור אות שיעול:



### הליך ביצוע סימון ידני בתוכנת Audacity

1. פותחים הקלטה בעזרת תוכנת Audacity (ניתן לגרור את קובץ ההקלטה לקובץ התוכנה או לחילופין דרך התוכנה לפתוח את הקובץ הנדרש). אתם צריכים לראות מסך בדומה לדוגמה למעלה.
2. מומלץ להציג גם ספקטרוגרמה של האות. לצורך כך, לחצו על הסימון בו מופיע שם הקובץ (מספר 1 בדוגמה) יחד עם חץ כלפי מטה בצד שמאלי עליון של המסך. בחלון שנפתח, בחרו Multi-view לתצוגה של האות גם בזמן וגם בתצורת ספקטרוגרמה.
3. לצורך סימון תחום מסוים באות הזמני, לחצו עם לחצן שמאלי בעכבר באזור תחילת הזמן הרצוי וגררו עד סיום הזמן הרצוי. לאחר מכן לחצו Ctrl+B במקלדת. ייפתח אזור חדש מתחת עם האזור הנבחר ואופציה למתן שם לאותו סימון.
4. מתן שם לסימון:  
a. עבור שיעול יש לרשום 123 או cough.

- b. עבור נשימה יש לרשום 4 או breathing או breath.
- הערה: יש לרשום את השם במדויק ולוודא כי אין טעויות. כל טעות הכי מינורית בשם תגרום למצב שאותו סימון לא ייקרא כלל. לרעיונות לשמות חלופיים, יש לתאם קודם עם ניר.
5. יש לוודא שהסימון מציג את זמן התחלה וסיום הסימון (בחירה בחלון למטה: Start and End of Selection).
6. חזרו על שלבים 3-4 עד לסיום סימון כל החלקים הרלוונטיים בהקלטה.
7. בסיום הסימון עבור כל ההקלטה, יש לשמור את הסימונים בקובץ txt באופן הבא:
- בצד שמאלי עליון של המסך, בחרו File > Export > Export Labels..
  - בחרו תיקייה רלוונטית (היכן שכל קבצי ה txt נמצאים).
  - בחרו שם קובץ txt מתאים (שם זהה לשם ההקלטה).
  - לחצו save.
8. וודאו כי ההקלטה קיימת וכי בתוכה נבחרו הסימונים הנכונים.
9. דוגמה לקובץ txt שנשמר:



- כאשר עמודה שמאלית מציגה זמן התחלה, עמודה אמצעית מציגה זמן סיום ועמודה ימנית מציגה את שם הסימון.
10. לאחר בדיקה כי הכל תקין עם קובץ ה txt, צאו מההקלטה הנוכחית בתוכנה ועברו להקלטה הבאה (ניתן לצאת מהתוכנה ולגרור קובץ שמע חדש לתוכנה).

### הערות/המלצות נוספות במהלך שימוש בתוכנה

- ניתן לבצע zoom in בציר הזמן.
- ניתן לבצע השמעה של ההקלטה בזמנים ספציפיים ובמהירויות שונות.
- ניתן לבצע scaling לאות בציר ה y וגם להציג ביחידות של dB (נוח עבור אותות מאוד חלשים).
- ניתן לשלוט על מאפייני התמרת STFT (בחלון בו בחרתם Spectrogram > Multi-view Settings..). לדוגמה: שינוי גודל החלון ישפיע על הרזולוציה בזמן ובתדר של הספקטרוגרמה. מומלץ לשחק עם מאפיינים אלו ולראות את ההשפעה.



אם מצאתם פיצ'רים נוספים מעניינים > תעדכנו אותנו