Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

Design Method of an Embedded System Equipped with AI for the COVID-19 Diagnosis from the Sound of Cough

Agostino Giorgio

Politecnico di Bari, Italy

agostino.giorgio@poliba.it

Abstract: The search for vaccines against the COVID-19 pandemic has been the most important action against this disaster. However, the prevention has a crucial role, also, and nowadays there is a lack of simple and reliable devices providing an early warning when it is suspected that there may be someone positive for the COVID-19 virus, especially in a closed environment. The aim of this paper is to describe the design of such a device. Screening by temperature measurement is not a very reliable practice because fever is not always associated with COVID-19 infection. Therefore, the assumed biomarker is the cough. In fact, the device can reliably distinguish among the COVID and the non-COVID cough. It is designed using a properly developed artificial intelligence algorithm on the "edge impulse" platform by using the TinyML model and is deployed in microcontroller-based Embedded Systems and in an Android smartphone.

***Keywords*:** Artificial Intelligence, Biological sounds, COVID-19, Digital Signal Processing, Embedded Systems, Smart Medical Devices.

I. INTRODUCTION

The COVID-19 is a well-known respiratory infection caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The disease has already infected millions of humans across the globe, has had a high fatality rate and, therefore, it has been also immediately demonstrated a challenge for the scientific community. One of the most interesting challenges is the prevention of contagions thanks to prompt diagnosis performed with reliable, consumer and low-cost devices. To this aim a very promising method is the recognition of the cough due to COVID-19 infection [1]. In fact, audio signals generated by the human body are used in diagnosis and monitoring of many diseases and recently scientists have started exploring how respiratory sounds, especially coughs, from patients tested positive for COVID-19 differ from sounds from healthy people [2 – 5].

Due to the ease of measurement, a temperature scan is currently the predominant massive screening method for COVID-19. However, between cough and fever, the number of non-COVID-19 medical conditions that can cause fever are much larger than the non-COVID-19 conditions that can cause cough. It has been also demonstrated that cough contains COVID-19 specific features even if it is non-spontaneous, i.e., when a COVID-19 patient is asked to cough [4]. This means cough can be used as a pre-screening method by asking the subject to simulate cough. Also, the author in a recent paper [1] has shown that there are technologies using AI (Artificial Intelligence) algorithms, which could help to perform a reliable diagnosis of the COVID-19 infection as well as of other respiratory and cardiac pathologies by analyzing cardiac sounds, respiratory sounds but also the sound of coughing and speech [2-21].

Particularly, it is possible to classify a pathological condition based on a cough sound because of the physical structure of the respiratory system gets altered with respiratory infections [21].

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

For examples, the authors in [2] presented an algorithm for automated diagnosis of pertussis using audio signals by analyzing cough and whoop sounds. Several other recent studies have attempted to identify by the cough sound chronic obstructive pulmonary disease, caused primarily due to smoking [22], and tuberculosis, disease usually caused by mycobacterium *tuberculosis bacteria* that affects the lungs [23]. In addition, for respiratory disorders like asthma and pneumonia, algorithms based on cough sounds recorded using a smartphone demonstrates a high level of accuracy [24]. Therefore, because of the cough is a symptom of several diseases, the diagnosis of a COVID-19 infection by cough alone is an extremely challenging multidisciplinary problem to address by investigating the distinctness of pathomorphological alterations in the respiratory system induced by COVID-19 and, therefore, suitably to be solved by using AI methods. For this reason, the research initiatives from University of Cambridge [25], Carnegie Mellon University [26], Wadhwani AI institute [27] and a project from EPFL [28] have already been launched. Also, in a recent work [4] the authors suggest a good accuracy for cough-based detection of COVID-19 using a preliminary investigation with a small number of subjects. Worth of notice is also the COSWARA project aiming at the same objectives [20].

In summary, using cough for the diagnosis of COVID-19 by using the AI, is due to the following key findings:

1) cough from different respiratory syndromes have distinct latent features [2, 29-35] that can be extracted by appropriate signal processing of the cough sounds. This is because of the pathomorphological alterations in the respiratory system are different from one respiratory disease to another. Therefore, cough caused by COVID-19 has distinct features from those associated with other respiratory infections.

2) Cough manifests as a symptom in the majority (e.g., 67.7% as per [36]) but not all COVID-19 carriers. However, studies show that coughing is one of the key mechanisms for the social spreading of COVID-19 [37]. Droplets containing the virus emitted through cough are the most prolific mechanism of spreading the COVID-19 [38].

Therefore, the use of the cough as a biomarker for COVID-19 infection offers a great potential for early diagnosis, which could be applied to the masses if embedded in commodity devices that could monitor individuals in a not uncomfortable way.

This paper presents the author's contribution along this direction by describing the development of a new AI algorithm and the design of a related consumer smart device, helping for the massive, real-time diagnosis of COVID-19. The new algorithm has been developed according to the author's paper [1] by using a highly optimized TinyML (Tiny Machine Learning) model [39] and has been trained with a significance dataset of cough sounds obtained bringing together the best-known and valuable public accessible datasets. The design was carried out using the "Edge Impulse" platform, very useful for quickly and easily developing and implementing AI algorithms on a wide variety of devices such as smartphones, microcontroller-base ES (Embedded Systems), minicomputers, personal computers. In this project they have been used a smartphone, the Portenta H7 board, both with an external microphone and with the vision shield (which has an embedded microphone), and the Arduino Nano 33 Ble Sense board, which has an embedded microphone. Obviously, the goal of the project is not to replace the existing clinical chemical testing and instrumental diagnosis methodologies but to supplement them with a cost effective, fast, and simpler tool, useful for a massive real-time first screening.

In Section II, it is proposed just a reminder about the AI method and algorithm used, deeply detailed in [1]. In Section III it is described the development environment, "Edge Impulse" or TinyML [39]. In Section IV it is described the procedure to obtain the AI model for COVID-19 cough detection, the performance of the model and the design of microcontroller-based ES implementing the developed AI model. Conclusions and final remarks are in Section V.

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

II. AI METHODS FOR BIOLOGICAL SOUNDS CLASSIFICATION

AI refers to a method of data analysis inspired by the way the human brain works and which takes the form of different computational models or algorithms, known as artificial neural networks (ANN) and convolutional neural networks (CNN) [40], which are extremely useful in many fields. An ANN can learn from data so it can be trained to recognize patterns, classify data, and calculate ("predict") future events based on what it has learned.

The operations performed by the AI are data analysis, identification and quantification of elements characterizing such data (process known as feature extraction) and data classification or assignment of a specific category to which they belong. To this aim it is fundamental the process known as model training, in which the AI model “learns” to assign a certain type and value of features to a certain category of membership (label or class). The training process requires a lot of data to improve the classification accuracy and AI performances. In fact, AI provides answers that are not true absolutely but with a level of "confidence", that is, with a certain probability.

This method is now in common use in speech recognition, implemented in the well-known voice assistants [41, 42], and applies very well to the classification of images or the automatic recognition of objects because each class of objects has very specific characteristics. The ANN has a layered structure resembling the structure of the network of neurons in the brain, with layers of connected nodes. ANNs that operate on two or three layers of connected neurons are known as surface neural networks. In contrast, deep networks, known as Deep Learning (DL) networks, can have many layers, even hundreds. Both are Machine Learning (ML) techniques that learn directly from the input [43-45]. The DL is particularly suitable for complex applications such as facial recognition, text translation, speech recognition, advanced driver assistance systems and so on [46-48].

A ML workflow to classify objects starts with a process of features extraction from images (that can be frames from a video in real time, also). The features to be extracted must be explicitly indicated. These are then organized in a dataset and used to create a model that categorizes the objects in the image.

With a DL workflow, on the contrary, the significant features are automatically identified and extracted from the images, with considerable advantage because a preliminary phase of identification and validation of the same by the operator is not required. In addition, the DL performs end-to-end learning, in which a network automatically learns how to process raw data and how to perform a classification activity.

A key advantage of the DL is the ability to improve performance as the amount of data provided for training increases [49-55]. Therefore, it is possible to carry out objects’ recognition through DL and/or ML within an image, performing the extraction of characteristics or features automatically (with the DL) or manually (with the ML), and then performing the recognition i.e., the classification. For this procedure to be applied to biological sounds, they must be transformed into images.

To produce a reliable diagnosis of COVID-19 from cough sounds transformed into images, as demonstrated in [1], the DL is the best approach because of the automatism with which the features are identified and extracted. The methods and the relevant processing steps to reliably transform biological sounds in images has been detailed in [1]. As just a reminder, in summary the methods are:

- Mel spectrogram [56]
- Image from Mel spectrogram coefficients (MFCC) [57]
- Gammatone spectrogram [58]
- Image from gammatone spectrogram coefficients (GTCC) [59]

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

- Scalogram by wavelet transform [60]
- Image from wavelets coefficients (CWT) [61].

These techniques for the transformation of audio signals into images are in some respects similar to each other, as they indicate different modes of time-frequency representation of an audio signal; however, they have differences that recommend a specific field of application for each [1].

The Mel spectrogram and its MFCC coefficients are among the most used techniques because they are sized on the basis of human hearing sensitivity. However, they have limitations related to the low efficiency of Mel filters in eliminating additive noise, especially that present in speech audio signals and therefore in speech recognition applications; the problem is less important when it comes to biological signals such as heart tones and the respiratory signal.

The problem of noise in speech processing applications can be solved by resorting to the Gammatone spectrogram and its GTCC coefficients, which better reproduce the behavior of the human cochlea membrane, including filtering the frequencies where the additive noise is mainly located.

The GTCC and the related Gammatone spectrogram are therefore more suitable for vocal identification and recognition, while the MFCC and its Mel spectrogram are preferred for classifying generic but low noisy signal (heartbeats, heart tones and biological signals in general), or any audio signal with a high SNR (Signal-to-Noise Ratio), which behavior as time varies is well defined and the sampling is easier. Moreover, for heart and respiratory sounds it must be considered that the additive noise is attenuated because they are acquired with dedicated low noise hardware. For the cough sound the SNR is particularly high due to the intensity of the sound.

The other technique considered is that relating to the scalogram and its wavelets coefficients, through which it is possible to have a good compromise between resolution over time and frequency. The wavelet analysis, in fact, allows to process information with better resolution than other techniques as the analysis time window is not fixed and allows to better capture audio signals with long time intervals at low frequencies and very short time intervals but with high frequencies. This technique, therefore, seems particularly suitable for the classification of respiratory signals acquired by a sthetoscope where, especially if pathological, sounds that overlap the vesicular murmur are detected, of short duration and at high or prolonged frequencies and at lower frequencies, such as whezees and crackles.

The images obtained with each of these 6 methods of transformation of audio signals into images previously described can be classified through DL algorithms. Therefore, the 6 methods described were implemented and compared by the author in [1] in order to establish which could be the most suitable for the classification of the signal produced by the cough from COVID-19 distinguishing it from the cough caused by other pathologies and that from the not pathological shot of cough and from the environmental strong noise.

In has been concluded in [1] that the Mel spectrogram and the MFCC coefficients are the most suitable methods. Therefore, the project described in this paper has been carried out according to these conclusions.

III. AI MODELS IMPLEMENTATION: EDGE IMPULSE PLATFORM

AI models can be implemented on PCs and/or smartphones but also on microcontrollers. The ML models suitable for implementation on microcontrollers are often called EML (Embedded ML) [62] or TinyML. "*Edge Impulse*" is a free (within certain limits), powerful, cloud-based platform for building AI models for microcontrollers that combines two popular techniques: AutoML and TinyML [39].

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

AutoML is an ML library for embedded applications. It provides a powerful model generator that can be run with minimal configuration and code. It works with any dataset and produces quality results in real time.

TinyML is a technique that integrates reduced and optimized ML applications that require "full-stack" solutions (hardware, system, software and applications). It can be implemented in low power consumption systems, such as sensors or microcontrollers, to perform automated tasks. Traditional ML models cannot be deployed on these devices. With TinyML, models are converted and optimized to work on the microcontroller. It allows to develop ML models and deploy them on various microcontroller-boards as Arduino Nano 33 BLE Sense [63], Portenta H7 [64], Raspberry Pi 4 [65], mobile phones [66].

Another powerful AI framework is TensorFlow Lite [67] powered by Google and often used with Google Colab to make quite easy the implementation of scripts in Python language. AutoML library is TensorFlow Lite-based. Also, Matlab environment provides a powerful tool (Matlab Coder) to deploy AI models directly from Matlab/Simulink to hardware [54, 55]. To this aim, it is also necessary to install the appropriate hardware support packages.

In this project, as development environment, it has been used the “Edge Impulse” platform that manages the entire pipeline from data acquisition to model deployment [39]. The development and the deployment of AI models require to acquire data, preprocess and organize them in a dataset. Finally, the dataset is used to train a neural network, an AI model. Once the model has been trained and evaluated for accuracy and precision (tested), is deployed in hardware.

To run in microcontroller-based ES, the model is also previously converted and compressed. “Edge Impulse” offers a complete workflow from the AI model development to its deployment in a wide variety of compatible hardware e does not require writing code, having a very efficient GUI (Graphycal User Interface). Therefore, it is possible to create a neural network without writing any code because of the GUI makes easy to navigate and edit model parameters. It allows also to generate templates from existing TensorFlow or Keras samples in the cloud.

“Edge Impulse” provides tools that can run in Windows or macOS to collect data from sensors and feed it into the cloud-based platform. Depending on the data format, such as time series, audio and video, it is possible to choose appropriate neural network architectures and optimization techniques to train the custom AI model, named the “Impulse”. “Edge Impulse” leverages the infrastructure, frameworks and tools typically used by experienced data scientists and ML engineers. It leverages parallelization for data processing and dataset sampling. It uses TensorFlow to build and train neural networks supported by powerful AI accelerators such as GPUs, and TensorFlow Lite to convert the model to be run on a microcontroller, building ES independent on external services.

Finally, “Edge Impulse” converts the model, optimizes it, and generates the necessary files to deploy it directly to a device selected in a wide range of compatible devices and development boards, as in the list on the platform web site [39]. The platform in the deployment step creates ready-to-use binary files including the AI model and provides libraries with examples code with the firmware customized. The whole workflow on the “Edge Impulse” platform is described in detail, step by step, in the following section.

IV. AI MODEL DEVELOPMENT AND DEPLOYMENT ON EDGE IMPULSE

First of all, it needs to create a free account on the “Edge Impulse” web site [39] and, after logging in, create a new project and give it a name by clicking on the title. Then, the data type to be processed have to be chosen among: accelerometer data, audio, images, something else.

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

The whole workflow is carried out by following the dashboard, shown in figure 1, located at the left side of the screen.

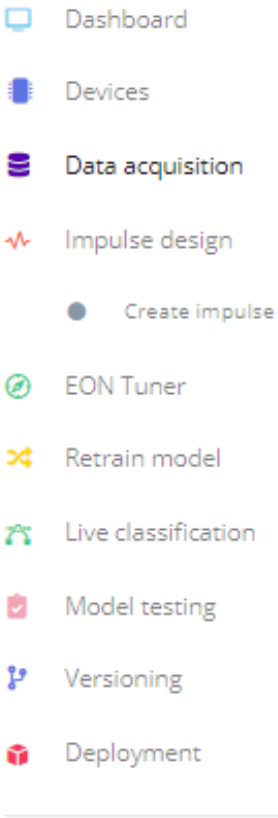


Fig. 1. Dashboard of the Edge Impulse workflow

The next step consists in collecting the data suitable for the AI model training and testing, whose classification must be well known. The "Edge Impulse" platform supports different modes for data acquisition, i.e.:

• By connecting a development board among those supported, equipped with sensors suitable for data acquisition

• Via a mobile phone and its embedded sensors

• Using a personal computer

• From a development board via the Data Fowarder

• Uploading (importing) an existing dataset

• Through the cloud

To properly organize the dataset, before starting the acquisition, it needs to create a label for each class to be distinguished. To classify between COVID and non-COVID cough it is necessary to create two labels, for example "positive" (label for COVID cough) and "negative" (label for non-COVID cough or simply environmental noise). The easiest and most immediate way to get started with data acquisition, is using a smartphone which collects audio samples and add them to the dataset. The steps are detailed in [66].

Collecting audio data directly from a microcontroller-board is also a very useful option. The main steps are:

- connect the board to a personal computer by a USB cable

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

- update the firmware:

I. Download the latest "Edge Impulse" firmware from [68], and unzip the file;

II. Open the flash script for the used operating system (flash_windows.bat, flash_mac.command or flash_linux.sh) to flash the firmware;

III. Wait until flashing is complete, and press once the RESET button on the board to launch the new firmware.

- Set keys: from a command prompt run the command:

edge-impulse-daemon

This will start a wizard which will ask to log in, and choose an "Edge Impulse" project. To switch projects, if wanted, run the command with --clean. Alternatively, Google Chrome and Microsoft Edge can collect data directly from the development board, without the need for the Edge Impulse CLI [69].

Verify that the device is connected to "Edge Impulse" by clicking on "Devices" in the dashboard: the device will be listed.

Once the device (board) is connected, data samples are collected just like with the smartphone.

The samples of the dataset are visible on the "Data acquisition" page, as in figure 2, accessible always by clicking on the dashboard proper button. Moreover, by clicking on a sample, it appears what the sample looks like and is possible to hear the audio by clicking on the play button below each graph.

To accomplish the goals of this project, it has been built a wide database by merging together 3 important databases [20], [28] and [70]. Then, the dataset has been uploaded through the "Upload data" option in figure 4.

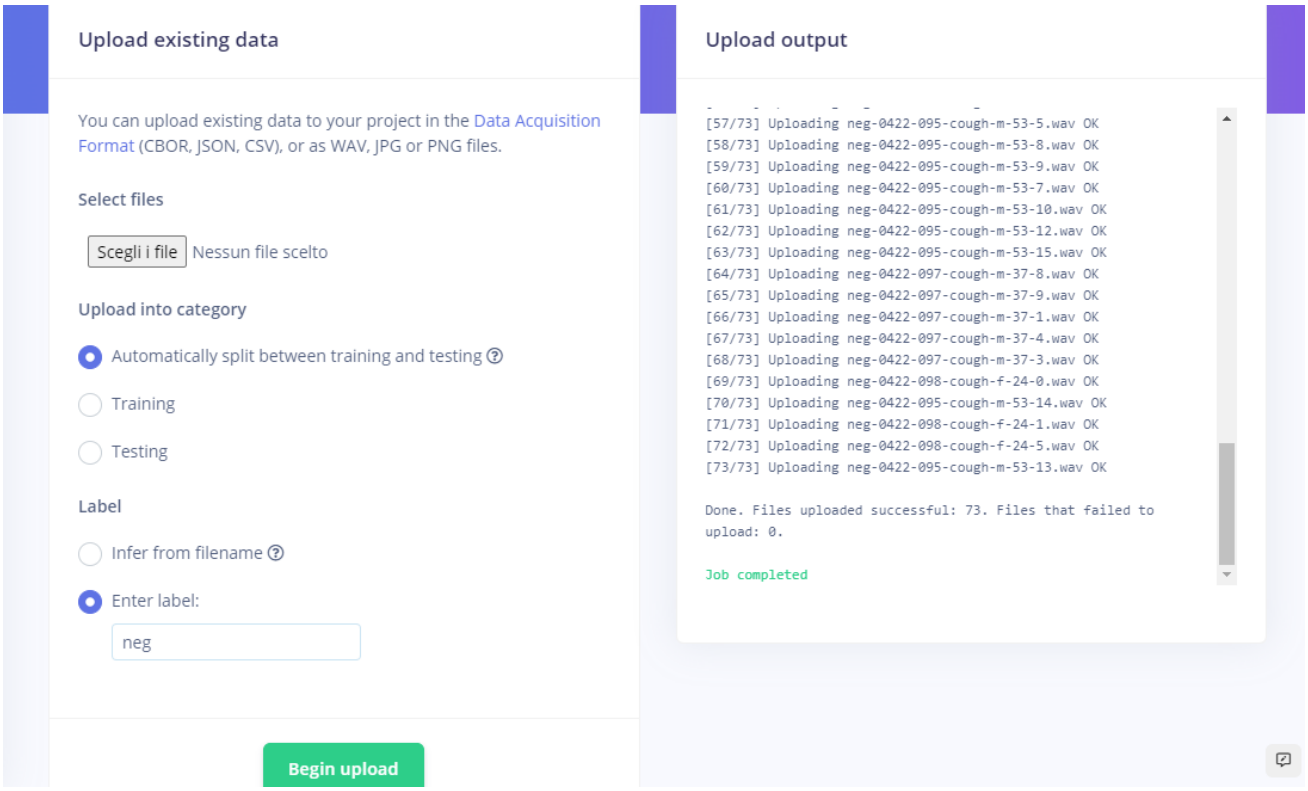


Fig. 2. Data acquisition section of the edge impulse platform

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

The dataset is automatically splitted into two subsets: the training set and the testing set (respectively 75% and 25% of the dataset but these percentages can be modified, if wanted).

The next step consists in creating the custom AI model. Therefore, needs to select signal processing and machine learning blocks on the "Create impulse" page. The model will start out blank, with Raw data and Output feature blocks. The default settings of a 1000 ms Window size and 500 ms Window increase can be leaved. This means the acquired audio data will be processed 1 s at a time, starting each 0.5 s. Using a small window saves memory on the embedded device, but needs sample data without large breaks in between them.

The impulse main settings are shown in figure 3

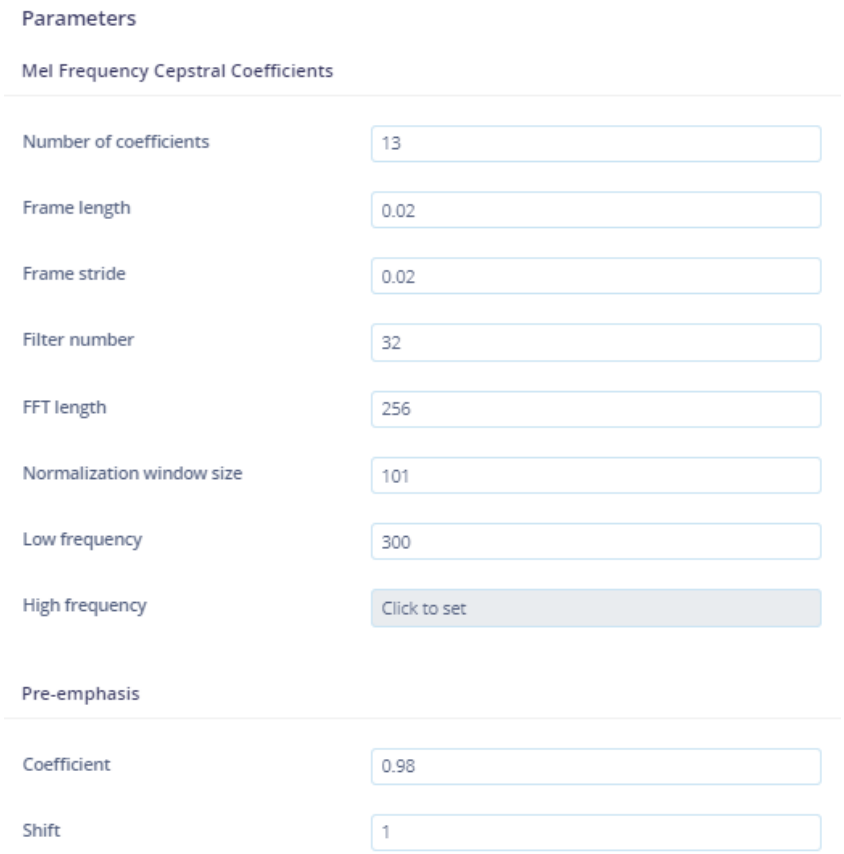


Fig. 3. Impulse (AI model) parameters

The next step is to create the signal processing chain, useful to prepare the data for the AI model training and testing, by clicking on 'Add a processing block' and, according to [1], selecting the Audio (MFCC) block. Moreover, needs clicking on 'Add a learning block' and selecting the Neural Network (Keras) block. This step ends by clicking on 'Save Impulse'.

The audio block will calculate the MFCC for each windowed audio, and the neural network block will be trained to classify the spectrogram as either a 'covid cough (positive)' or 'non-covid cough (negative)' based on the training dataset.

The next step is the features extraction from the training dataset on the MFCC page, required to train the neural network and to create the AI classifier. The MFCC page (see figure 4) shows what the extracted Mel spectrogram looks like for each 1 second window from any of the dataset samples. We can leave the parameters to their defaults.

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

Next, by clicking on the 'Generate features' button the entire training dataset is processed with the designed processing chain, and is created the set of the features that will be used to train the Neural Network.

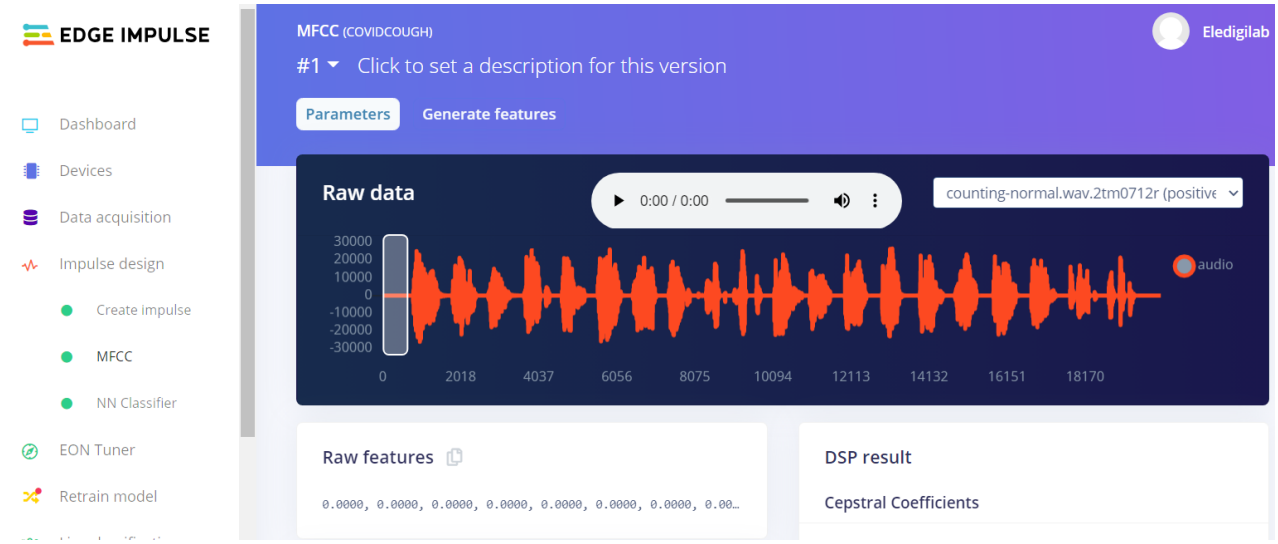


Fig. 4. Edge Impulse platform screen after features have been extracted

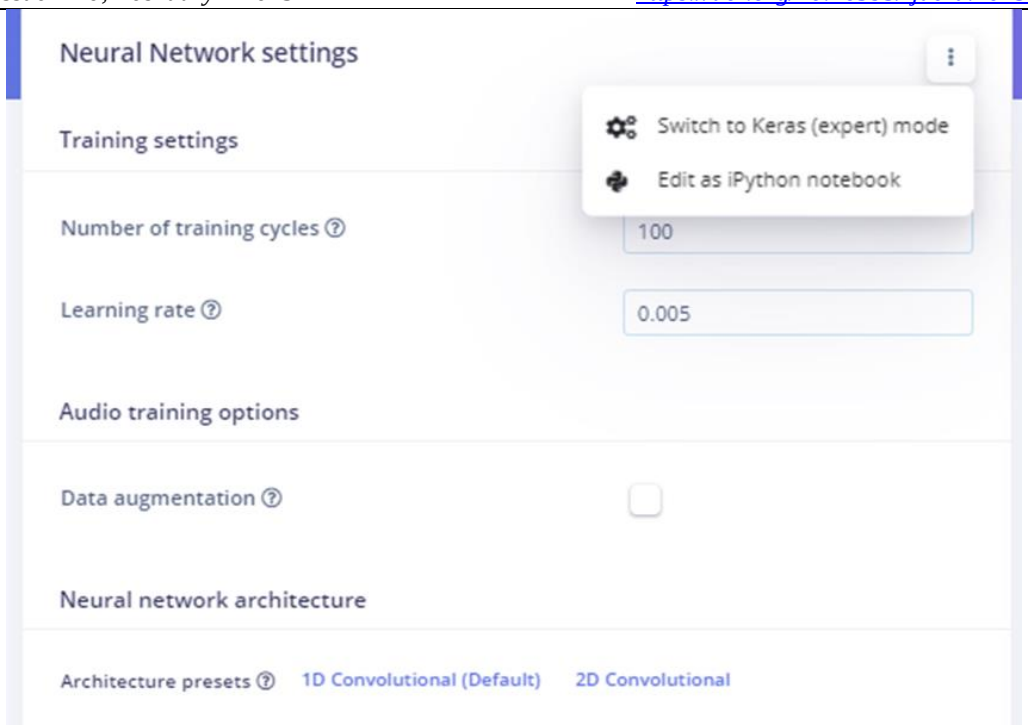
The next step is to setup and train the neural network, by clicking on to "NN Classifier" in the dashboard in figure 1.

The default neural network works well for continuous sounds like water running. Cough detection is a more complicated task, so it needs configure a richer network using 2D convolution across the spectrogram of each window. 2D convolution processes the audio spectrogram in a similar way to image classification. To customize the neural network, it needs to press the upper right corner of the 'Neural Network settings' section, and to select 'Switch to Keras (expert) mode'. In figure 5 are shown the settings and the code suitable for our purposes, according to author's previous results in [1].

Having the dataset and being the features and the neural network properly designed, next steps are to train and test the network, to build the AI model and to deploy it in hardware.

To train and test the model, it is used the dataset previously acquired and splitted into train and test partitions.

By clicking on to "Start training" button below the neural network architecture, the training is performed and, at the end of the process, the platform shows the results in terms of accuracy, as in figure 6. It is worth of notice that the accuracy in this project is very satisfactory, as expected from results in [1].



Neural network architecture

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout,
    Flatten, Reshape, BatchNormalization, Conv2D, MaxPooling2D,
    AveragePooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.constraints import MaxNorm
# model architecture
model = Sequential()
model.add(InputLayer(input_shape=(X_train.shape[1], ), name
    ='x_input'))
model.add(Reshape((int(X_train.shape[1] / 13), 13, 1),
    input_shape=(X_train.shape[1], )))
model.add(Conv2D(10, kernel_size=5, activation='relu', padding
    ='same', kernel_constraint=MaxNorm(3)))
model.add(AveragePooling2D(pool_size=2, padding='same'))
model.add(Conv2D(5, kernel_size=5, activation='relu', padding
    ='same', kernel_constraint=MaxNorm(3)))
model.add(AveragePooling2D(pool_size=2, padding='same'))
model.add(Flatten())
model.add(Dense(classes, activation='softmax', name='y_pred',
    kernel_constraint=MaxNorm(3)))
# this controls the learning rate
opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
# train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt,
    metrics=['accuracy'])
model.fit(X_train, Y_train, batch_size=32, epochs=9,
    validation_data=(X_test, Y_test), verbose=2)
```

Fig. 5. Neural network settings and architecture

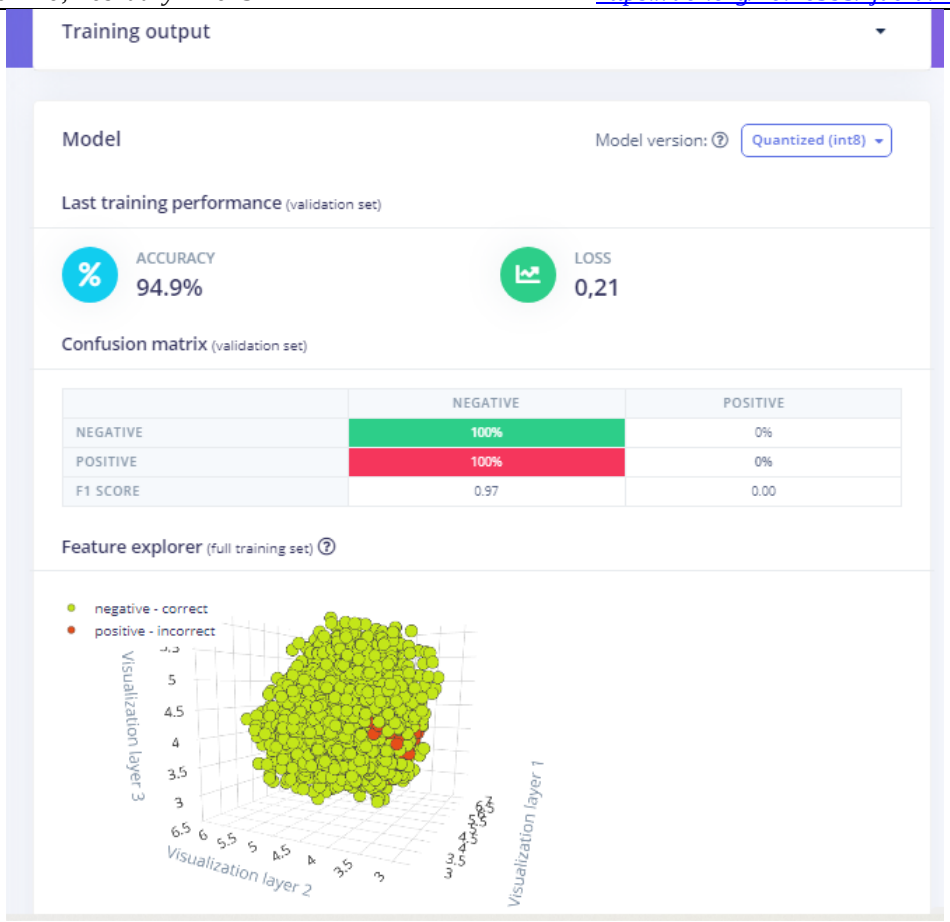


Fig. 6. Training output

Subsequently, is performed also the test of the trained model. Results are in figure 7. The testing accuracy is very good, also.

For further tests, in the "Edge Impulse" platform is available the "Live classification" option, as shown in the dashboard in figure 3, that allows to test the developed AI model also by streaming audio data from any supported board and smartphone.

In order to perform the live classification by a smartphone, it needs to open the "Edge Impulse" browser page (or to refresh that page if it has been opened earlier to acquire the dataset). Subsequently, by pressing the 'Switch to classification mode' button will automatically build the project into a WebAssembly package and execute it on the smartphone continuously.

The connection of the smartphone to the project is made easy by a QR code generated by the "Edge Impulse" platform, as previously explained.

Having properly tested and validated the AI model, the final step is the deployment in hardware, performed by clicking on to "Deployment" in the dashboard (see figure 1).

The platform shows a lot of possible choices. By selecting the 'Arduino Nano 33 BLE Sense' under 'Build firmware' and, then, by clicking 'Build', will build a complete firmware for the Nano BLE Sense including the developed AI model.

Following the instructions provided on the screen it is easy to flash the Arduino board with the binary programming file containing the AI model (classificator).

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

The same deployment can be performed on another board, for example on the very performing Portenta H7.

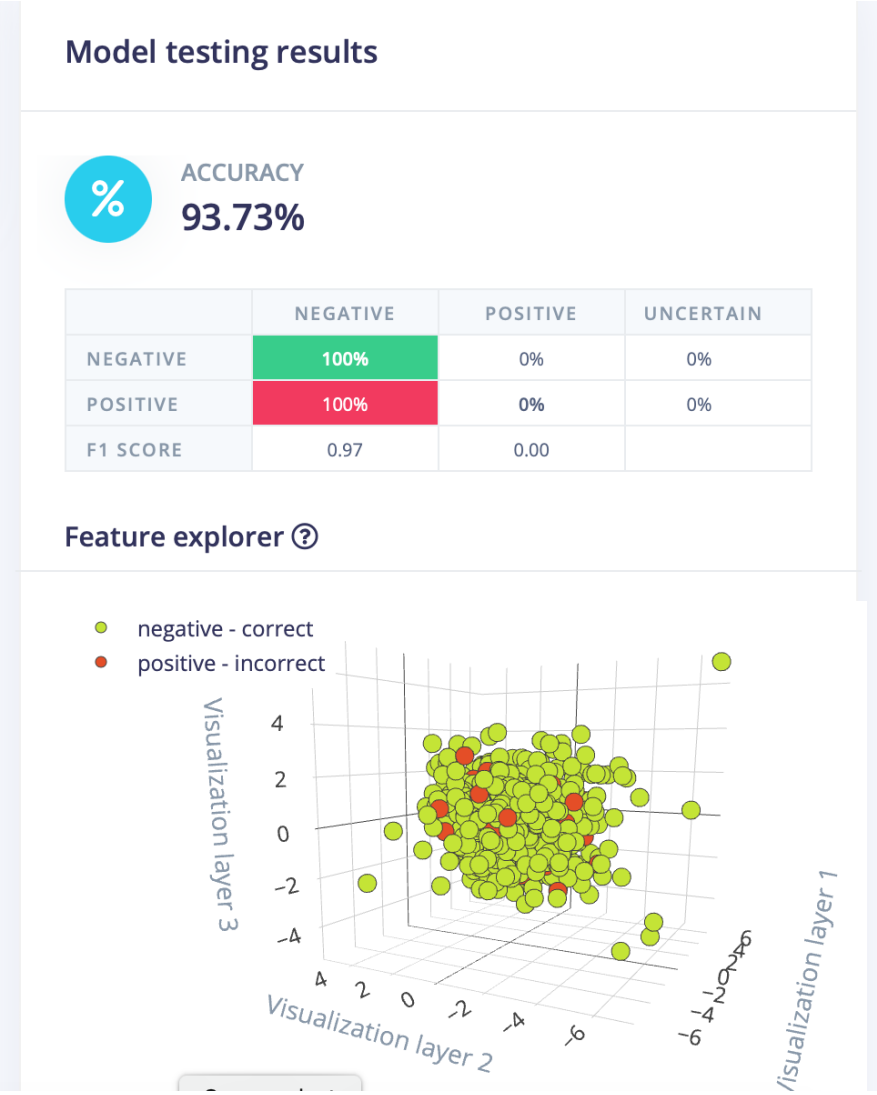


Fig. 7. Testing output

Alternatively, it is possible to choice the generic option "Arduino", and then the platform builds a library in a .zip format ready to be uploaded in the Arduino IDE.

The library returned contains the AI developed model and also the example sketches useful to implement the AI model into the arduino boards, specifically into the Nano 33 Ble Sense and into the Portenta H7 boards.

V. RESULTS

In this work, the model has been deployed in a smartphone, as already shown, and in Arduino boards, Nano 33 BLE Sense and Portenta H7. The Portenta H7 has been used both with an external microphone as using the vision shield with its integrated microphone.

Once the board is flashed, by the serial monitor of the Arduino IDE it is possible to see on the PC screen the results of the real time classification of the cough, as shown in figure 8.

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

```
COM7

Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 10 ms., Classification: 217 ms., Anomaly: 0 ms.):
[0.00068, 0.99932]
    neg: 0.00068
    pos: 0.99932
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 10 ms., Classification: 217 ms., Anomaly: 0 ms.):
[0.94349, 0.05651]
    neg: 0.94349
    pos: 0.05651
```

Fig. 8. Real time classification of a positive (covid) and negative (non-covid) cough performed by the Arduino Nano 33 BLE Sense board and printed on the screen of a PC by the serial monitor tool of the Arduino IDE

In figure 9 is shown the PortentaH7 board with an external microphone and in figure 10 the same board equipped with the vision shield, which embedded microphone is used to acquire the cough sounds to be classified.

The classification results printed on the Arduino IDE serial monitor are in figure 11. They are almost the same with both microphones.

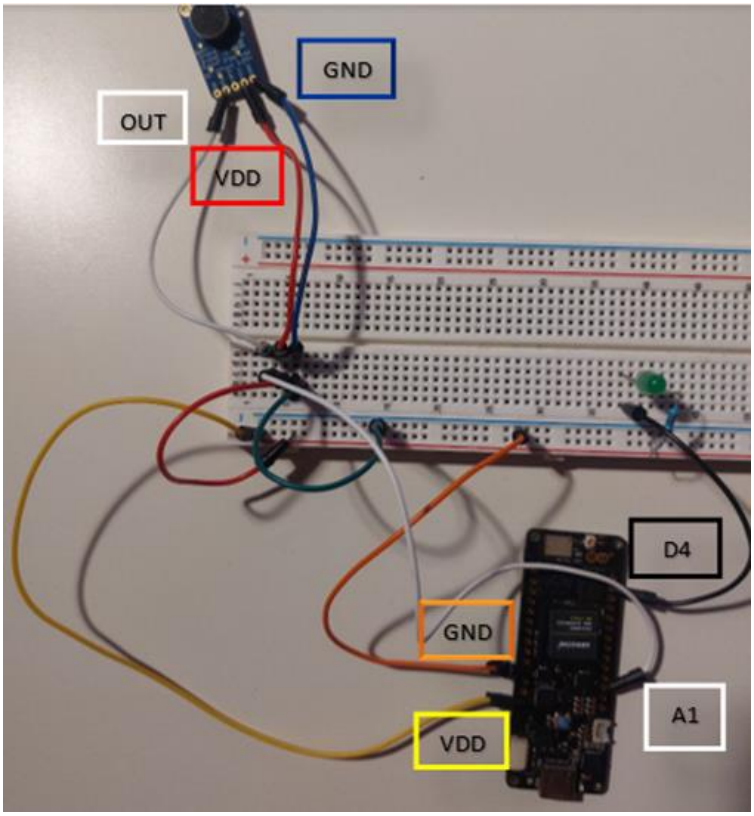


Fig. 9. Portenta H7 board with the external microphone

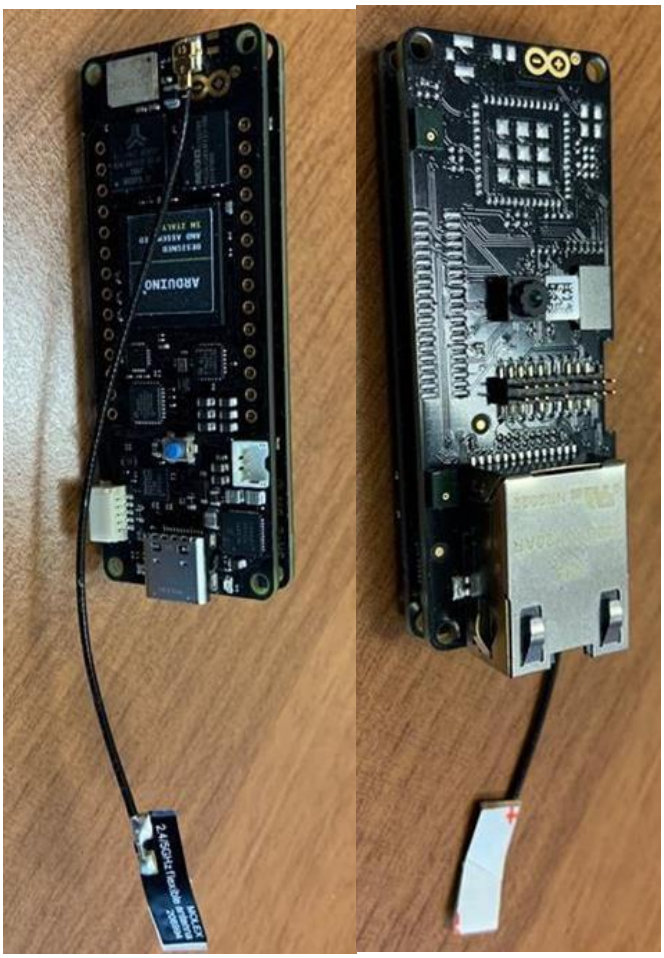


Fig. 10. PortentaH7 board with the vision shield

```
    pos: 0.01450
Predictions (DSP: 4 ms., Classification: 216 ms., Anomaly: 0 ms.):
    neg: 0.97700
    pos: 0.02300
Predictions (DSP: 4 ms., Classification: 215 ms., Anomaly: 0 ms.):
    neg: 0.98181
    pos: 0.01819
Predictions (DSP: 4 ms., Classification: 215 ms., Anomaly: 0 ms.):
    neg: 0.99754
    pos: 0.00247
Predictions (DSP: 4 ms., Classification: 216 ms., Anomaly: 0 ms.):
    neg: 0.98056
    pos: 0.01944
Predictions (DSP: 4 ms., Classification: 216 ms., Anomaly: 0 ms.):
    neg: 0.97817
    pos: 0.02183
Predictions (DSP: 4 ms., Classification: 216 ms., Anomaly: 0 ms.):
    neg: 0.97286
    pos: 0.02714
```

Fig. 11. Cough classification example by Portenta H7 board

Volume –8, Issue – 10, February – 2023 https://doi.org/10.46593/ijaera.2023.v08i10.001

The classifications performed are all accurate, according to the model testing results.

VI. CONCLUSIONS AND FUTURE DEVELOPMENTS

The project proposed in this work aims to demonstrate how is it possible to design a consumer device implementing an AI model for cough sound classification, suitable as an auxiliary tool for one of the most important goals of the COVID-19 treatment and prevention, namely the early diagnosis of the infection to avoid a massive infection.

The author has taken into account other recent studies that demonstrate the usefulness of cough sound in diagnosing COVID infection. The author also refers to its previous paper, which shows that AI methods are especially suitable for cough classification. Additionally, the author used valuable public domain COVID-cough databases and the efficient "Edge Impulse" platform to develop an AI model that accurately classifies cough sounds. The author, therefore, has successfully deployed this model on smartphones and microcontroller-based embedded systems, which makes it possible to quickly and easily develop consumer devices with potential applications.

The AI model reliability is demonstrated but it is very important to frequently update the model by training it using larger and larger database of cough to obtain classification confidence levels that are as high as possible, tending to 100%. Moreover, continuously updating the database to train the AI model, allows to take into account the variants of the COVID-19 that are frequently detected.

In order to improve the AI model, it will be very useful also to collect also and correlate each other voice sounds and breath sounds, together with cough sounds. In fact, vocal patterns, alongside breathing and cough, could give useful additional features for classification.

The obtained results are encouraging but surely they are not as solid as would be necessary to constitute a standalone screening tool even if, perhaps, in a near future, may be. Therefore, the results obtained in this paper only scratch the surface of the potential of the usefulness of the consumer devices that can be designed using the described workflow. Morevoer, if the classifier is further updated to more cough sound classes, may be possible reliably and accurately to distinguish among cough sounds in COVID-19 and cough sounds from asthma and other non-covid cough deseases.

Conflict of interest: The author declares that he has no conflict of interest.

Ethical statement: The author declares that he has followed ethical responsibilities.

REFERENCES

[1] A. Giorgio, “Artificial Intelligence Methods for Image Classification applied to Biological Sounds for the early diagnosis of cardiorespiratory pathologies and COVID-19 infection”, *Intern. Jour. of Biomed. Engin. and Tech.*,(In Press).

[2] R.X.A. Pramono, S.A. Imtiaz, E. Rodriguez-Villegas “A Cough-Based Algorithm for Automatic Diagnosis of Pertussis*” PLoS ONE* vol. 11 no. 9: e0162128. Sept 2016, doi:10.1371/journal.pone.0162128

[3] B. W. Schuller, D. M. Schuller, K. Qian, J. Liu, H. Zheng, and X. Li, “Covid-19 and computer audition: An overview on what speech & sound analysis could contribute in the sars-cov-2 corona crisis,” *Front. Digit. Health*, March 2021, doi: 10.3389/fdgth.2021.564906

[4] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, S. Riaz, K. Ali, C. N. John, and M. Nabeel, “Ai4covid-19: Ai enabled preliminary diagnosis for covid-19 from cough samples via an app”, *Informatics in medicine Unlocked*, vol. 20, June 2020, doi: 10.1016/j.imu.2020.100378

[5] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data." *KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, August 2020, pp. 3474–3484, doi: 10.1145/3394486.3412865

[6] J. Han, K. Qian, M. Song, Z. Yang, Z. Ren, S. Liu, J. Liu, H. Zheng, W. Ji, T. Koike, X. Li, Z. Zhang, Y. Yamamoto, and B. W. Schuller, “An Early Study on Intelligent Analysis of Speech under COVID-19: Severity, Sleep Quality,

Fatigue, and Anxiety", *Proc. Interspeech 2020, 4946-4950*, doi: 10.21437/Interspeech.2020-2223 [Online] Available: https://arxiv.org/pdf/2005.00096.pdf

[7] C. Bales, M. Nabeel, C. N. John, U. Masood, H. N. Qureshi, H. Farooq, I. Posokhova, and A. Imran, "Can machine learning be used to recognize and diagnose coughs?", *Proc. Int. Conf. e-Health Bioeng. (EHB), Oct. 2020, pp. 1–4.*

[8] L. Brabenec, J. Mekyska, Z. Galaz, and I. Rektorova, "Speech disorders in Parkinson's disease: Early diagnostics and effects of medication and brain stimulation" *Journal of Neural Transmission*, vol. 124, no. 3, pp. 303–334, March 2017, doi: 10.1007/s00702-017-1676-0

[9] G. Deshpande and B. Schuller, "An overview on audio, signal, speech, & language processing for COVID-19". May 2020 *arXiv:2005.08579.* [Pre-print online] Available: https://arxiv.org/pdf/2005.08579.pdf .

[10] B. E. Sakar, G. Serbes, and C. O. Sakar, "Analyzing the effectiveness of vocal features in early telediagnosis of Parkinson's disease" *PloS One*, vol. 12, n. 8, pp. 1–18, Aug. 2017. doi: 10.1371/journal.pone.0182428

[11] M. Faurholt-Jepsen, J. Busk, M. Frost, M.V.E.M. Christensen, O. Winther, J. E. Bardram, and L. V. Kessing, "Voice analysis as an objective state marker in bipolar disorder". *Translational Psychiatry*, vol. 6, no. 7:e856, July 2016, doi: 10.1038/tp.2016.123

[12] J. Han, K. Qian, M. Song, Z. Yang, Z. Ren, S. Liu, J. Liu, H. Zheng, W. Ji, T. Koike, X. Li, Z. Zhang, Y. Yamamoto, and B. Schuller, "An early study on intelligent analysis of speech under COVID-19: Severity, sleep Quality, fatigue, and anxiety", May 2020 [Preprint online] Available: https://arxiv.org/abs/2005.00096., doi: https://arxiv.org/abs/2005.00096

[13] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification". *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jan 2017, pp. 131–135. 10.48550/arXiv.1609.09430

[14] Y. Huang, S. Meng, Y. Zhang, et al., "The respiratory sound features of COVID-19 patients fill gaps between clinical data and screening methods." *J. Health Sci Dev*, vol: 4, no. 1, pp. 8-15, March 2021, doi: 10.1101/2020.04.07.20051060

[15] S.H. Li, B.S. Lin, C.H. Tsai, C.T. Yang, and B.S. Lin, "Design of wearable breathing sound monitoring system for real-time wheeze detection". *Sensors*, vol. 17, no. 1, pp. 1–15, Jan 2017, doi.org/10.3390/s17010171 .

[16] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless sleep apnea detection on smartphones", *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys).*, Florence, Italy, May 2015, pp. 45–57, doi: 10.1145/2742647.2742674 .

[17] D. Oletic and V. Bilas, "Energy-efficient respiratory sounds sensing for personal mobile asthma monitoring", *IEEE Sensors Journal* vol. 16, no. 23, pp. 8295–8303, Dec. 2016, doi: 10.1109/JSEN.2016.2585039.

[18] R. Xaviero A. Pramono, S. Bowyer, and E. Rodriguez-Villegas, "Automatic adventitious respiratory sound analysis: A systematic review". *PloS One*, vol. 12, no. 5, May 2017, doi: 10.1371/journal.pone.0177926 43

[19] K. S. Alqudaihi, N. Aslam, I. U. Khan, A. M. Almuhaideb, S- J. Alsunaidi, N.M.A.R. Ibrahim, F.A. Alhaidari, F.S. Shaikh, Y.M. Alsenbel, D.M. Alalharith, H.M. Alharthi, W.M. Alghamdi, M.S. Alshahrani. "Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities". *IEEE Access*, Vol. 15, no. 9, pp. 102327-102344. July 2021, doi: 10.1109/ACCESS.2021.3097559 .

[20] N. Sharma, P. Krishnan, R. Kumar, S. Ramoji, S. R. Chetupalli, R.Nirmala, P. Kumar Ghosh, and S. Ganapathy, "Coswara – A database of breathing, cough, and voice sounds for COVID-19 diagnosis", Proc. Interspeech 2020, Oct. 2020 Shangai, [Preprint online] Available: https://arxiv.org/abs/2005.10548 doi: 10.21437/Interspeech.2020-2768.

[21] M. W. Tobias, "AI And Medical Diagnostics: Can A Smartphone App Detect Covid-19 From Speech Or A Cough?", *Forbes*, May 2020

[22] A. Windmon, M. Minakshi, P. Bharti, S. Chellappan, M. Johansson, B. A. Jenkins, and P. R. Athilingam, "Tussiswatch: A smartphone system to identify cough episodes as early symptoms of chronic obstructive pulmonary disease and congestive heart failure," *IEEE Jour. of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1566–1573, July 2019, doi: 10.1109/JBHI.2018.2872038

[23] G. Botha, G. Theron, R. Warren, M. Klopper, K. Dheda, P. Van Helden, and T. Niesler, "Detection of tuberculosis by automatic cough sound analysis" *Physiological Measurement*, vol. 39, no. 4, April 2018, doi: 10.1088/1361-6579/aab6d0

[24] P. Porter, U. Abeyratne, V. Swarnkar, J. Tan, T.-w. Ng, J. M. Brisbane, D. Speldewinde, J. Choveaux, R. Sharan, K. Kosasih, P. Della, "A prospective multicentre study testing the diagnostic accuracy of an automated cough sound centred analytic system for the identification of common respiratory disorders in children" *Respiratory Research*, vol. 20, no. 1, p. 81, June 2019, doi: 10.1186/s12931-019-1046-6

[25] "Cambridge university, UK - COVID-19 sounds app," Weblink https://covid-19-sounds.org/en/ .

[26] "CMU sounds for covid project," Weblink https://node.dev.cvd.lti.cmu.edu/.

[27] "Cough against covid - wadhwani AI institute," Weblink https://coughagainstcovid.org/ .

[28] "Cough for COVID-19 detection," Weblink https://coughvid.epfl.ch/ .

[29] W. Thorpe, M. Kurver, G. King, and C. Salome, "Acoustic analysis of cough" *Proc. of the IEEE Seventh Australian and New Zealand Intelligent Information Systems Conference*, pp. 391–394, doi: 10.1109/ANZIIS.2001.974110

[30] H. Chatrzarrin, A. Arcelus, R. Goubran, and F. Knoefel, "Feature extraction for the differentiation of dry and wet cough sounds" *Proc. IEEE International Symposium on Medical Measurements and Applications*, May 2011, pp. 162-166, doi: 10.1109/MeMeA.2011.5966670.

[31] I. Song, "Diagnosis of Pneumonia from Sounds Collected using Low Cost Cell Phones", *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1-8, doi: 10.1109/IJCNN.2015.7280317.

[32] C. Infante, D. Chamberlain, R. Fletcher, Y. Thorat, and R. Kodgule, "Use of cough sounds for diagnosis and screening of pulmonary disease," in *Proc. 2017 IEEE Global Humanitarian Technology Conference (GHTC), 2017*, pp. 1-10, doi: 10.1109/GHTC.2017.8239338

[33] M. You, H. Wang, Z. Liu, C. Chen, J. Liu, X.-H. Xu, and Z.-M. Qiu, "Novel Feature Extraction Method for Cough Detection Using NMF," *IET Signal Processing*, vol. 11, no. 5, pp. 515–520 July 2017, doi: 10.1049/iet-spr.2016.0341

[34] I. D. Miranda, A. H. Diacon, and T. R. Niesler, "A comparative study of features for acoustic cough detection using deep architectures" *Proc. 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2601–2605, Oct. 2019, doi: 10.1109/EMBC.2019.8856412

[35] M. Soli´nski, M. Łepek, and Ł. Kołtowski, "Automatic cough detection based on airflow signals for portable spirometry system" *Informatics in Medicine Unlocked*, vol. 18, no. 20, doi: 10.1016/j.imu.2020.100313

[36] World Health Organization, "Report of the WHO-China Joint Mission on Coronavirus Disease 2019 (COVID-19)", 2020

[37] G. Gao, "Not wearing masks to protect against coronavirus is a 'big mistake' top Chinese scientist says", *Science*, Mar 2020, [Online]. Available: https://www.science.org/content/article/not-wearing-masks-protect-against-coronavirus-big-mistake-top-chinese-scientist-says

[38] National Institute of Health, "New coronavirus stable for hours on surfaces", 2020, [Online]. Available: https://www:nih:gov/news-events/news-releases/new-coronavirus-stable-hours-surfaces

[39] www.edgeimpulse.com

[40] https://it.mathworks.com/discovery/neural-network.html

[41] Speech Command Recognition using Deep Learning, Mathworks: https://www.mathworks.com/help/deeplearning/ug/deep-learning-speech-recognition.html

[42] Introduction to Deep Learning for Audio and Speech Applications, Webinar by Gabriele Bunkheila, MathWorks: https://www.mathworks.com/videos/introduction-to-deep-learning-for-audio-and-speech-applications-1560448385032.html

[43] Machine Learning and Deep learning for Audio, MathWorks: https://www.mathworks.com/help/audio/feature-extraction-and-deep-learning.html

[44] https://it.mathworks.com/discovery/deep-learning.html

[45] https://it.mathworks.com/discovery/machine-learning.html

[46] Deep Learning Onramp and Deep Learning with Matlab, https://it.mathworks.com/learn/tutorials/deep-learning-onramp.html

[47] Deep Learning for Signals and Sound, Webinar by J. Pingel and E. Andersson, MathWorks: https://www.mathworks.com/videos/deep-learning-for-signals-and-sound-1544467789023.html

[48] Deep Learning for Speech and Audio Processing with NVIDIA GPUs, Webinar by Gabriele Bunkheila, MathWorks: https://www.mathworks.com/videos/deep-learning-for-speech-and-audio-processing-with-nvidia-gpus-1586524417560.html

[49] Deep Learning Toolbox, Mathworks: https://www.mathworks.com/products/deep-learning.html

[50] Get Started with Transfer Learning, Mathworks: https://www.mathworks.com/help/deeplearning/gs/get-started-with-transfer-learning.html

[51] Transfer Learning with Deep Network Designer, Mathworks: https://www.mathworks.com/help/deeplearning/ug/transfer-learning-with-deep-network-designer.html

[52] https://it.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html#mw_45a8c0b2-26fa-48e9-905a-a7ed7b87bfc8

[53] M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification" In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135, 2017, [Preprint online] Available: https://arxiv.org/abs/1609.09430 .

[54] Deep Learning in Simulink using Deep Neural Networks library, Mathworks: https://www.mathworks.com/help/gpucoder/ug/deep-learning-in-simulink-using-deep-neural-networks-library.html

[55] Getting Started with Android Devices, Mathwork: https://www.mathworks.com/help/supportpkg/android/examples/getting-started-with-android-devices.html

[56] https://www.mathworks.com/help/audio/ref/melspectrogram.html

[57] https://www.mathworks.com/help/audio/ref/mfcc.html

[58] https://www.mathworks.com/help/audio/ref/gammatonefilterbank-system-object.html

[59] https://www.mathworks.com/help/audio/ref/gtcc.html
[60] https://www.mathworks.com/help/wavelet/ref/cwtfilterbank.wavelets.html?searchHighlight=wavelet&s_tid=srchtitle
[61] https://www.mathworks.com/help/wavelet/ref/cwt.html
[62] https://docs.edgeimpulse.com/docs/what-is-embedded-machine-learning-anyway
[63] https://docs.edgeimpulse.com/docs/arduino-nano-33-ble-sense
[64] https://docs.edgeimpulse.com/docs/arduino-portenta-h7
[65] https://docs.edgeimpulse.com/docs/raspberry-pi-4
[66] https://docs.edgeimpulse.com/docs/using-your-mobile-phone
[67] https://www.tensorflow.org/lite
[68] https://cdn.edgeimpulse.com/firmware/arduino-nano-33-ble-sense.zip
[69] Collect Sensor Data Straight From Your Web Browser (edgeimpulse.com)
[70] https://github.com/virufy/virufy-data
[71] Gunvant Chaudhari, Xinyi Jiang, Ahmed Fakhry, Asriel Han, Jaclyn Xiao, Sabrina Shen, Amil Khanzada, "Virufy: Global Applicability of Crowdsourced and Clinical Datasets for AI Detection of COVID-19 from Cough", Jan 2021, https://doi.org/10.48550/arXiv.2011.13320
[72] A. Khanzada, S. Hegde, S. Sreeram, G. Bower, W. Wang, R.P. Mediratta, K.D. Meister, A. Rameau, "Challenges and Opportunities in Deploying COVID-19 Cough AI Systems". *J Voice*, Nov. 2021; vol. 35, no. 6, pp. 811-812. doi: 10.1016/j.jvoice.2021.08.009. Epub 2021 Sep 7. PMID: 34610883; PMCID: PMC8421112.
[73] E. Darici, N. Rasmussen, J. Ranjani J., J. Xiao, G. Chaudhari, A. Rajput, P. Govindan, M. Yamaura, L. Gomezjurado, A. Khanzada, M. Pilanci, "Using Deep Learning with Large Aggregated Datasets for COVID-19 Classification from Cough", *arXiv preprint arXiv:2201.01669*, Jan. 2022 doi: https://doi.org/10.48550/arXiv.2201.01669

