Cough Recognition Using Tiny ML

Shankar R
Electronics and Communication
Amrita School of Engineering, Chennai campus
ch.en.u4ece19023@ch.students.amrita.edu

Gautham Mythireyan K M
Electronics and Communication
Amrita School of Engineering, Chennai campus
ch.en.u4ece19007@ch.students.amrita.edu

Nikitha Reddy Nalla
Electronics and Communication
Amrita School of Engineering, Chennai campus
ch.en.u4ece19031@ch.students.amrita.edu

M Venkateshkumar
Electronics and Communication
Amrita School of Engineering, Chennai campus
m_venkateshkumar@ch.amrita.edu

2022 IEEE Industrial Electronics and Applications Conference (IEACon) | 978-1-6654-8006-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/IEACon55029.2022.9951763

Abstract— The whole world has been affected by the current COVID-19 epidemic. Still, the impact of the epidemic and its consequences are felt else reckoning on our standing as individualities and as crew of society. The Omicron and IHU variant probably may circulate more fluently than the other variants and the way fluently Omicron and IHU spreads compared to other variants endure unspecified. It is expected that anyone with the virus can spread to others, even if they're fully vaccinated or don't feel severe symptoms like shortness of breath and Lung infection. One of the perceptible common symptoms for a covid infected characteristic person is cough. Considering this COUGH RECOGNITION USING TINY ML is created. The given product would be suitable to separate coughs from covid infected person's cough and noise by using TinyML. Arduino Nano 33 BLE sense which has an inbuilt microphone will descry the cough of the person. Now by using TinyML the cough can be categorized into a covid infected person cough or a noise without covid background.

Keywords— TINYML, BLE

I. INTRODUCTION

The COVID-19 epidemic has directed to a prominent loss of public health worldwide and made us to face many challenges to healthiness, the education system, and the food sector. The social and economic disturbance due to the epidemic is destructive knockouts as billions of people are at the trouble of falling into extreme poverty, while the count of undernourished people, presently evaluated at closely 690 million, could expand by over to 132 million by the edge of the time. Billions of companies go through empirical trouble. Nearly half of the world's 3.5 billion across-the-board employees are in the trouble of losing their lives. Many economy laborers are especially exposed due to the majority of them lack social conservation and access to proper health care. Lots and lots of people are unable to feed themselves and their families without the way to earn revenue during lockdowns. For the utmost, no income means no food, or, at best, lower food and lower nutritive food.

The COVID-19's death rate is knee-high compared to some deadly diseases. But the main muddle for us with covid is with its ability to spread rapidly. India is one of the most affected countries, there are many reasons for it but the reason we have underlined is the lack of monitoring of the infected people. Now the rapid tests for covid have been contrived but still, it doesn't address the issue of lack of monitoring. Checking temperature is not a very coherent solution to the issue. One of the discernible common symptoms for a covid infected symptomatic person is cough. People across the globe are worried about the Omicron variant of COVID-19. "The new Covid variant Omicron was first found in a sample collected on November 11, 2021, in Botswana and on November 14, 2021, in South Africa". It is noted that the Omicron and IHU variant may spread more faster than the all variant and chances of spread are five times higher than other variants. "TinyML is a branch of study combining Machine Learning and Embedded Systems that explores the types of models you can run on small, low-powered devices like microcontrollers". For the process of data acquisition, Impulse design and Neural network classifier, and model training Edge impulse studio is used. The Live classification helps to detect the unusual cough of the covid-19 affected person. Considering this COUGH RECOGNITION USING TINY ML is developed and demonstrated in this paper.

II. FRAMEWORK

A. Data Acqusition

In the absence of a microcontroller board, utilising a cell phone to gather data is the most straightforward option. Cough and noise data are gathered for this research, and this specific data will be used to train the machine learning model. Data is gathered in accordance with the fact that the aim is to diagnose the cough. If the device is connected to the remote management API and the location may start sampling new data, the mobile phone can be used to acquire some data from the Data acquisition page, where all the raw data is stored.

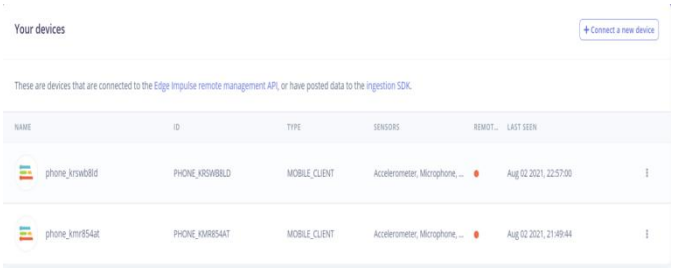

Fig 1. Linking Mobile phone to Edge impulse studio

The primary step in any machine learning project is to collect a dataset that represents known samples of data. To get started a small dataset of around 10 minutes of audio in two classes, "cough" and "noise" are accumulated.

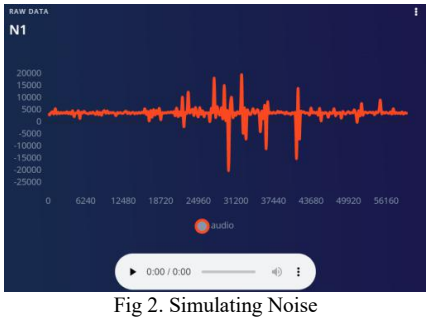

Fig 2. Simulating Noise

Fig 3. Simulating Cough

This dataset is compact and features a limited number of coughing samples and soft ground noise samples. Thus, the dataset is acceptable just for experimentation, and therefore the model produced during this paper is in a position to differentiate only between quiet ground noise and a little range of coughs.

B. Construct Dataset

Having collected enough data, it's then redeployed into a training set and a test set, usually during a ratio of 80% & 20%, however, own ratios are often made and separated accordingly.

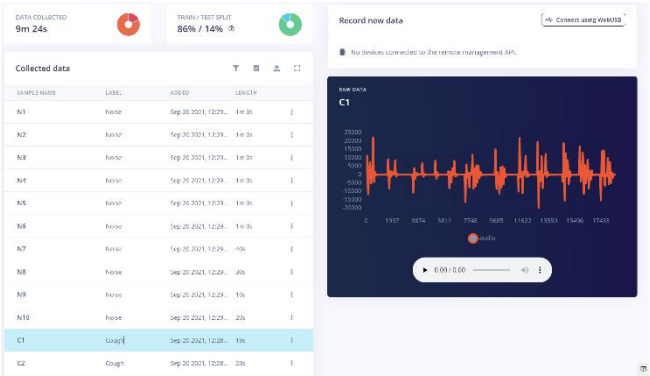

Fig 4. Collected Data

C. Impulse Design

Impulse designing is more trial and error than an exact piece of algorithm. The processing block and a learning block are added to the design parameters. Impulse settings guide how the data will be processed and used for predicting final outcome.

The data processed in impulse pieces it up in tiny windows and Digital signal processing (Dsp) blocks are used to extract features. Later for classifying the new data, learning blocks were added. An equivalent value for an equivalent input is returned by signal processing block and wasn't made the data easier to process, as learning blocks acquire a knowledge from past experiences. The MFCC signal processing block is used for this model. Mel Frequency Cepstral Coefficients (MFCC) which turns the raw audio containing an outsized amount of redundant information into a simplified form.

The collected data is passed into a Neural Network block, that will further classify the 2 classes of audio, i.e. Cough and Noise.

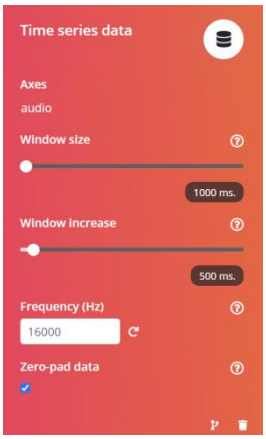

Fig 5. Time series data

Since each raw sample is divided into many windows, window increase field controls are used to regulate each window's offset.

Windows that overlap are formed by choosing a Window Increase less than the Window Size. Each overlapping window continues to be a single instance of audio that serves as the sample's label, even when the data it contains overall is similar.

The first of our training data is then utilised by employing overlapping windows. For instance, with a Window size of 1000 ms and an increase in Window of 100 ms, 10 distinct windows are retrieved from just 2 seconds of knowledge. Verify that the Window increase field does not go over 250 milliseconds. Neural network (Keras) has been selected as the learning block for the MFCC block for our model.

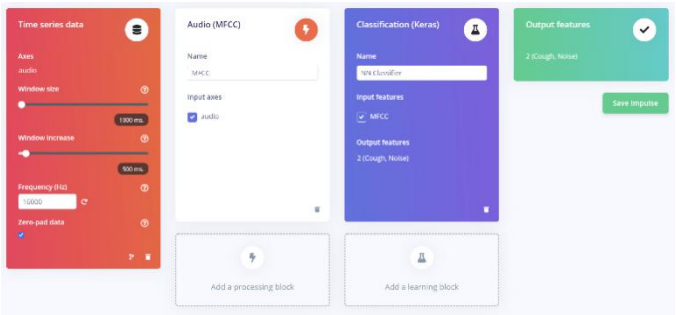

Fig 6. Impulse design

D. MFCC Block Configuration

This model permits to design of the MFCC block and permits to review of how the information will be changed. This model shows a perception of just a little noise, which is known as a spectrogram.

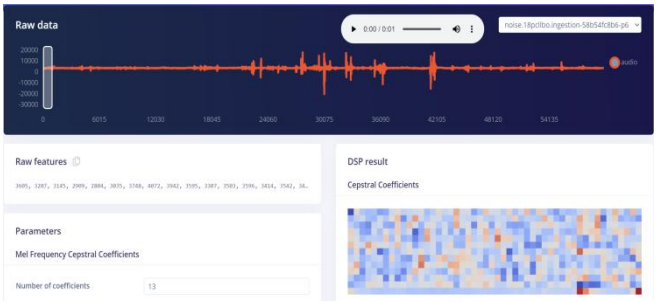

Fig 7. Spectral features

The MFCC block converts a window of sound into an informational grid where each line deals with a different set of frequencies and each segment deals with a different period of time. Each unit's value reflects the amplitude of the range of frequencies it is associated to during that time period. Each cell is represented by a shaded square on the spectrogram, and the amplitude determines how much the intensity changes.

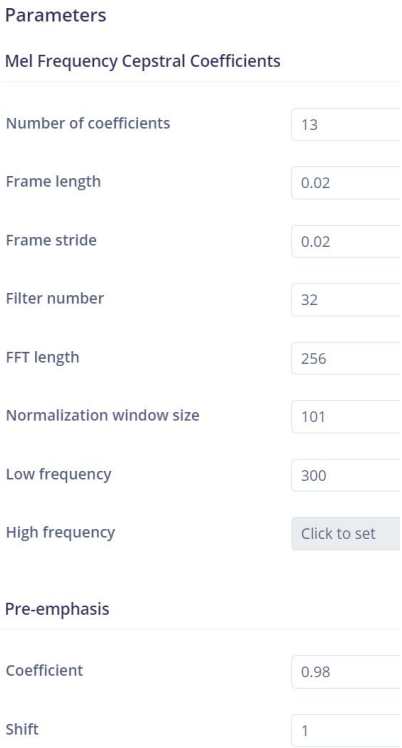

Fig 8. Spectral features parameters

The spectrograms produced by the MFCC block will be sent into a neural network specification that excels at recognising patterns in this kind of tabular data. The generation of MFCC blocks for various audio windows occurs prior to training a neural network. For this process, feature generation is carried out in order to produce the various features.

Fig 9. Spectral features Training set

Fig 10. Spectral features Output

The feature explorer displays a visualisation of the dataset once this operation is finished. Here, dimensionality reduction is utilised to map the features into a 3D space and the feature explorer is used to detect mislabeled data, determine how well different classes separate, and find more information about displaying complex datasets.

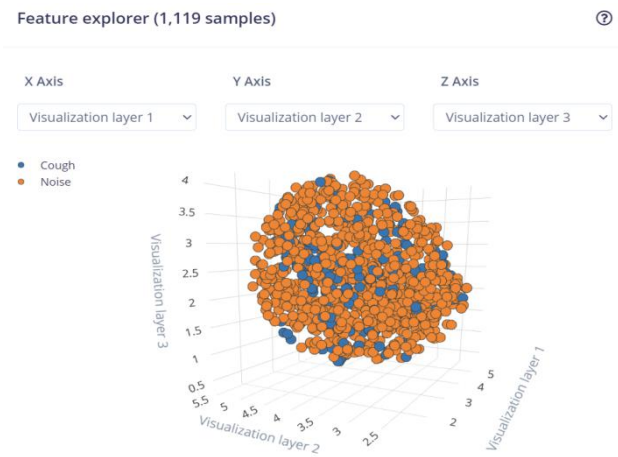

Fig 11. Generated feature explorer

E. Netural Network configuration

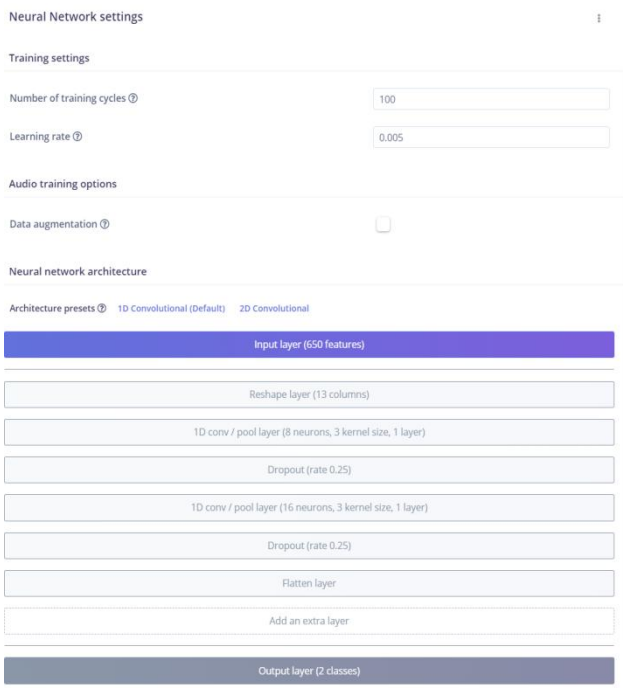

Fig 12. Neural network parameters

Once all the data has been processed, the neural network training has been started. Neural networks are algorithms that roughly model the human brain and learn to recognize patterns that appear in training data. The network you are training here takes a spectral analysis as input, checks it, and assigns it to at least one of the two classes. To train your model, NN Classifier is used. After a while, the model will start displaying information about the accuracy of model. The model displays data about the accuracy of the model.

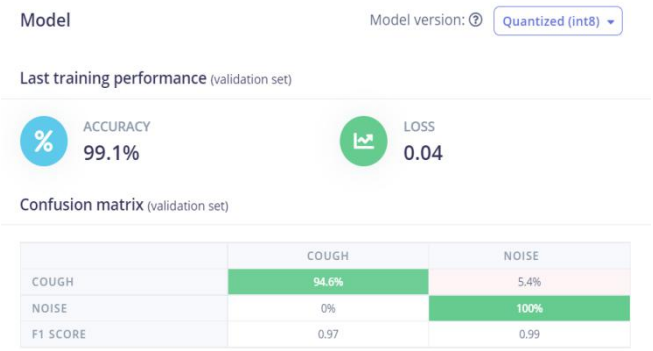

Fig 13. Model Accuracy and results

Neural networks consist of layers of virtual neurons that appear on the left side of the NN classifier page. The input spectral analysis is fed to the primary neuron layer, which filters and transforms to support the unique internal state of each neuron. The second level receives the first level's output, and so on until the first input has progressively undergone a complete transformation. In this instance, the middle layer transforms the DSP filter input to two numbers. That is the likelihood that various forms of detection are represented by the input. During training, the neuron's state is gradually optimised and enhanced, and the network modifies the input in the right way to produce the right output. This frequently entails entering coaching data into one sample, understanding and modifying the neuron's internal state to obtain the precise response for the following round. It is done by increasing the likelihood of being generated. After running 1000 times, the network will be trained. The specific placement of layers is called architecture, and various architectures serve various tasks. The custom neural network code can also be imported from tools such as TensorFlow and PyTorch.

Before training. There will be 30 coaching cycles total. According to this, the neural network will be trained 30 times using the entire body of knowledge. The network is not learning everything from the training data if there are too few cycles conducted. Overfitting, or the network starting to store training data, might decrease performance with data that has never been seen before if there are too many cycles.

A number of text flashes by within the Training output panel. At the start of coaching, about 20% of the training data is set aside for validation. This implies that rather than becoming accustomed to coaching the model, it is preferable to evaluate how the model is performing. The Last Training Performance window displays the validation results and provides useful information about the model and its behaviour. The accuracy percentage is the proportion of accurately identified detection windows. Although an accuracy close to 100% is improbable, the higher the number, which typically indicates that your model has overfitted the training data. In the following stage, during model testing, it is determined whether this is frequently true. For many applications, accuracy of more than 80% is frequently regarded as outstanding. The balance of windows that were correctly vs wrongly identified is displayed in the confusion matrix, which may be a table. Compare the values in each row to determine it. For instance, 99.5% of the tap detection windows in the screenshot above were mistakenly classed as Other Activities, while only 0.5% of them were Unusual Cough.

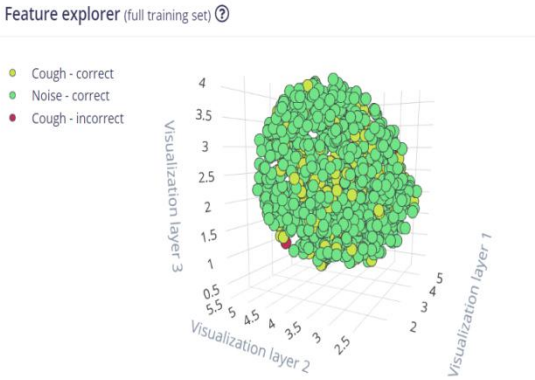

Fig 14. Data Visualization

The on-device performance section describes how the model can be processed on the device. Inference time is an estimate of how long it takes a model to analyse a single second of data from a typical microcontroller. Peak memory usage indicates how much RAM will be required to run the model on the device.

III. ARDUINO DEPLOYMENT

The model can be deployed back to the device once the impulse is designed, trained and verified. "This allows the model to run without an internet connection, with minimal latency and power consumption." Edge Impulse includes the entire model, including the Spectral analysis algorithm, classification code, and neural network weights, in a single Arduino library that can be used in any software. Once the model has been trained, Edge impulse provides a Zip library containing all of the features that can be compiled into an Arduino nano 33 Ble sense for further processing.

The working of the model can be tested using Arduino software using its serial monitor. Furthermore, the Arduino code has been modified to keep sending either Noise or Cough via BLE to the Arduino, depending on whether the audio is classified as a cough or not. Serial monitor with baud rate 125000 shows the desired output based on the audio input which will be observed using the microphone present in the Nano 33 Ble sense. As Arduino is open-source which can be further be interlinked with all other resources for a more specific way of producing the output.

IV. Results and Discussion

A. *Classification of existing data*

Edge Impulse gives some supportive instruments to testing the model, including how to catch live information from the device and quickly plan to order it. Classification of existing data can be made for those data which are already collected from the user on the studio.

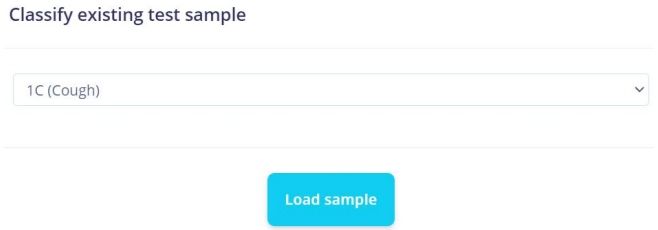

Fig 15. Existing data classification

When the example is transferred, it's parted into windows. These windows are then classified. The model grouped all windows of the captured detection. This is regularly a good outcome. The model has accurately recognized that the detection that caught. Though this is regularly new information that wasn't a piece of its training set.

Fig 16. Cough detected sample

Fig 17. Cough detected sample with Timestamp

It is also possible that some of the windows were incorrectly classified. Because the model was 99.5% accurate with its validation data, a minimum of 0.5% of windows will be classified incorrectly, and most likely far more, because validation data does not represent every possible type of background detection.

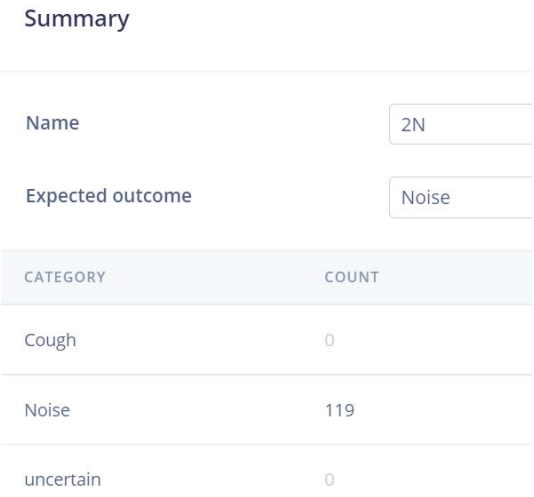

Fig 18. Noise detected sample

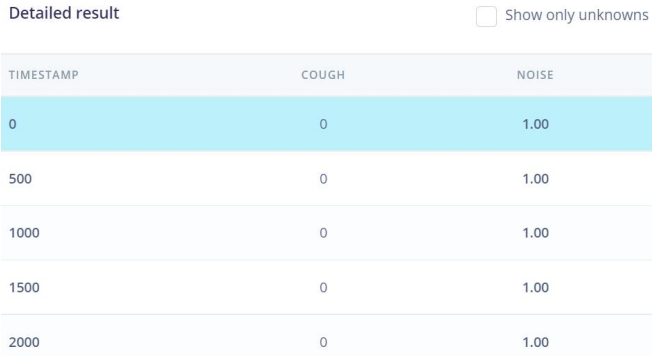

Fig 19. Noise detected sample with Timestamp

B. *Live Classification using PC/Mobile phone*

Data can be classified live using the microphone facility from PC/Laptop or mobile phone. Classified data is displayed during the live session along with the Time strap for better understanding.

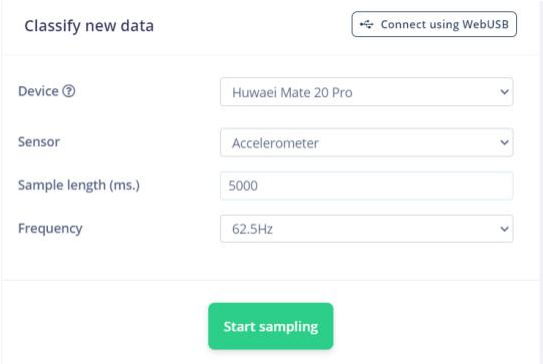

Fig 20. Live data classification

Fig 21. Live data classification in Mobile phone

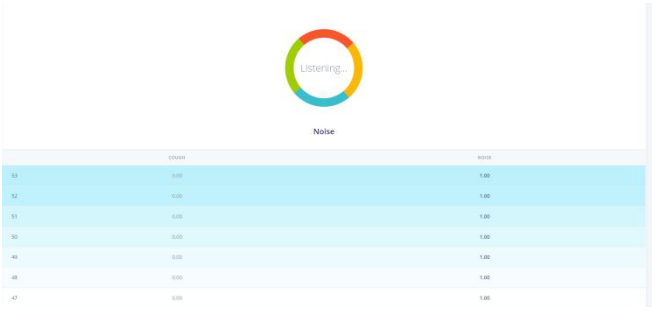

Fig 22. Live data classification in PC/Laptop

C. *Arduino classification*

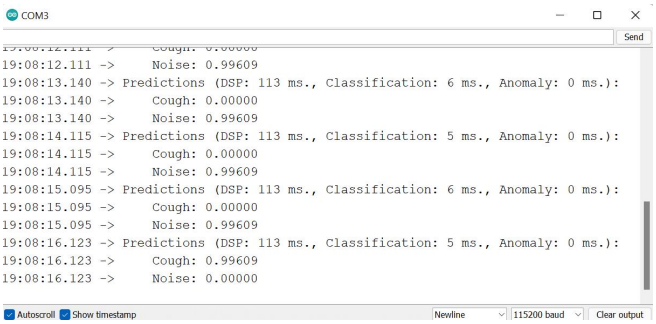

Fig 23. Arduino Serial monitor classification

V. CONLUSION

This paper reported on the deduction of covid using cough, as covid is an immediate threat humanity is facing. The paper briefly described all the stages (training and testing) which covered cough from covid, Arduino deployment, impulse design, model training, evaluation and graphical user interface using EDGE IMPULSE. Edge Impulse to coach a specify to detect the particular disease from the collected data. Our model has attained an accuracy of 99.5% which can run on the cloud to classify the data smoothly. There are endless applications for this sort of model, from monitoring the human body to recognizing various methods. The model can be also equipped with more innovative additions because its simple design. Now that the model is trained, the impulse is integrated within the firmware of the embedded device and running the impulse locally on the studio as well as on the Live classification.

VI. FUTURE SCOPE

As the device is of a simple design, so a similar framework can be implemented in various other infections. With new variants of covid emerging like omicron the detection using cough will be very efficient. We are currently working on the monitoring process. If the cough is detected to be covid infected the camera attached to device will capture the picture of the person and send it the administrator along with the location. The device can be set in organization in multiple locations in order to have more efficient monitoring. The design of the product is kept very simply make it compatible to any additions. Along with that camera-to-camera connection is added to constantly monitor the covid infected person and if any contact is made by the person with others.

REFERENCES

[1] T.Alafif, A. M.Tehame, S. Bajaba, A. Barnawi and Z. J. Saad, C. Maxwell, “Machine and Deep Learning towards COVID-19 Diagnosis and Treatment” *A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon,* 1892, pp.68–73.

[2] S. A. Mahmoudi, P. D. C. Possa, T. Ravet, T. Drugman, R. Chessini, T. Dutoit and C. V. K. Elissa, “Sensor-based System for Automatic Cough Detection and Classification” , *ICT Innovations 2015*

[3] G. Signoretti, M. Silva, P. Andrade, I. Silva, E. Sisinni, and P. Ferrari, “An Evolving TinyML Compression Algorithm for IoT Environments Based on Data Eccentricity,” *Sensors, vol. 21, no. 12, p. 4153, Jun.* 2021, doi: 10.3390/s21124153.

[4] S. S. Birring, T. Fleming, S. Matos, A. A. Raj, D. H. Evans, I. D. Pavord “Leicester Cough Monitor: a novel automated cough detection system”, *European Respiratory Journal, 2008*

[5] W. T. Goldsmith, A. M. Mahmoud, J. S. Reynolds et al., “A system for recording high fidelity cough sound and airflow characteristics,” *Annals of Biomedical Engineering,* vol. 38, no. 2, pp. 469–477, 2010.