Cough Detection using Arduino and ML

Vishal Shrivastav[1], Anas Shaikh[2], Mayur Chaudhary[3] , Mohsin Baig[4] ,Prof. R S Deshmukh[5]
[1]Student of Electronics Engineering, Rizvi College of Engineering, India
[2]Student of Electronics Engineering, Rizvi College of Engineering, India
[3]Student of Electronics Engineering, Rizvi College of Engineering, India
[4]Student of Electronics Engineering, Rizvi College of Engineering, India,
[5]Head of Department of Electronics and Telecommunication Engineering, Rizvi College of Engineering, India.

Abstract
There is a huge need for inexpensive, easily deployable solutions for COVID-19 and other flu-related early detection. In this project, we use Edge Impulse Machine learning on an Arduino Nano BLE Sense to detect the presence of coughing in real-time audio. We built a dataset of coughing and background noise samples, and applied a highly optimized TinyML model, to build a cough detection system that runs in real-time in under 20 KB of RAM on the Nano BLE Sense. This same approach applies to many other embedded audio pattern matching applications, for example, elderly care, safety, and machine monitoring.

Keywords: **Arduino Nano BLE Sense, Edge Impulse Studio, TinyML Model**

INTRODUCTION

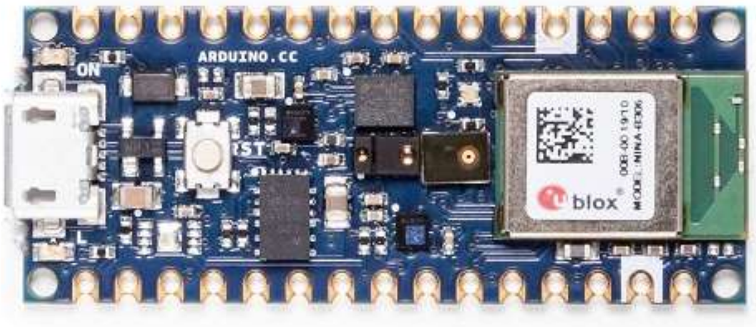
Respiratory symptoms can be caused by different underlying conditions, and often caused by viral infections, such as influenza like illness or other emerging viruses like coronavirus. These respiratory viruses, often, have common symptoms including coughing, high-temperature, congested nose, and difficulty breathing. However, early diagnosis of the type of the virus, can be crucial, especially in cases such as recent COVID-19 pandemic. One of the factors that contributed to the spread of the pandemic, was the late diagnosis of confusing it with regular flu- like symptoms. Science has proved that one of the possible differentiators of the underlying causes of these different respiratory diseases is coughing which comes in different types and forms. Therefore, a reliable lab-free tool for early and more accurate diagnosis that can differentiate between different respiratory diseases is very much needed. This project proposes an end-to-end portable system that can record data from patients with symptoms, including cough and translate them into health data for analysis. In our project we have included the software of Edge Impulse which has a certain number of datasheets which we upload in the software which helps to distinguish between a noise and a cough. We have also used Arduino BLE Sense which has a sensor included already in the board and does not require any other sensor. Thus, making it less bulky.

ABOUT THE PROJECT:

ML is a subset of AI that consists in the algorithmic modeling culture of statistical models, and only needs a small amount of knowledge to learn how to solve problems. Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), K-nearest Neighbor (KNN), Adaboost, K-means clustering (KC), Density clustering (DC), Hidden Markov Mod- els (HMM), Support vector machine (SVM), Naive Bayes (NB), Restricted Boltzmann Ma- chines (RBM), and Artificial Neural Network (ANN), such as Recurrent Neural Networks (RNN), including Long-short-term-memory (LSTM), Autoencoder (AE), and Generative Adversarial Network (GAN), are ML techniques. DL, on the other hand, is a subset of ML that focuses on building deep structural NN models that learn from data using algorithms of feedforward and backprobagation. Edge Impulse is a platform where you can perform such ML activities such as training and testing of a model.

COMPONENT SPECIFICATION:

Arduino Nano 33 BLE Sense:



The Nano 33 BLE Sense is Arduino's 3.3 V AI enabled board in the smallest available form factor.
It comes with a series of embedded sensors such as microphone to capture and analyse sound in real time, humidity and temperature sensor, gesture, proximity, light color and light intensity sensor.
The main feature of this board, besides the impressive selection of sensors, is the possibility of running the Edge Computing applications on it using TinyML. You can create your machine learning models using TensorFlow and upload them to your board using the Arduino IDE. The Arduino Nano 33 BLE Sense is based on the nRF52840 microcontroller operating at a voltage of 3.3V.
The Arduino Nano 33 BLE Sense is an evolution of the traditional Arduino Nano, but featuring a lot more powerful processor, the nRF52840 from Nordic Semiconductors, a 32-bit ARM Cortex running at 64 MHz. This will allow you to make larger programs than with the Arduino Uno (it has 1MB of program memory, 32 times bigger), and with a lot more variables (the RAM is 128 times bigger). The main processor includes other amazing features like Bluetooth® pairing via NFC and ultra low power consumption modes.

METHODOLOGY:

We are going to use Edge Impulse, an online development platform for machine learning on edge devices.
Collecting the dataset: The first step in any machine learning project is to collect a dataset that represents known samples of data that we would like to be able to match on our Arduino device. There are various platforms where you can find datasets, in our case we used Kaggle where we obtained various forms of data.
For Machine Learning we have to follow the following steps:
Goal defining, Data Gathering, Data Parsing, Model Creation and Accuracy Testing.
Data Gathering is the process of gathering and measuring information from countless different sources. Data Collection is one of the most important and difficult step of the process. It requires a lot of survey and tracking.
Data Parsing is a method where one string of data gets converted into a different type of data. A parser will take the file and transform it into a more readable data format that can be easily read and understood.
Modeling in machine learning is an iterative phase where we have to continually train and test machine learning models to discover the best one for the given task.
Accuracy testing is one metric for evaluating classification models. Accuracy is the fraction of predictions our model got right.
After gathering the data, we have to create two classes "cough" and "noise". And then we have to load the samples by using either mobile phones or through Edge Impulse CLI Uploader. For uploading the data through your mobile phone you have to scan a OR code which will appear when you click on" Use your Mobile Phone". This will connect your phone with Edge Impulse Studio.
With your phone connected with Edge Impulse Studio, you can now load your samples. To load the samples, click on 'Data Acquisition'. Now on the Data acquisition page, enter the label name, select the microphone as a sensor, and enter the sample length. And then click on "Start Sampling", to start sampling a 40 sec sample. After uploading the cough samples, now set the label to "noise" and collect another samples. These samples are for Training the module. Now we have to collect for the Test Data, which should be at least 30% of the training data.
Now, we have to start Training the model, we have to create impulse and add processing block which is Audio (MFCC) block. And add a learning block and select the Neural Network block. The default neural network works well for continuous sounds like water running. Cough detection is more complicated, so we will 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

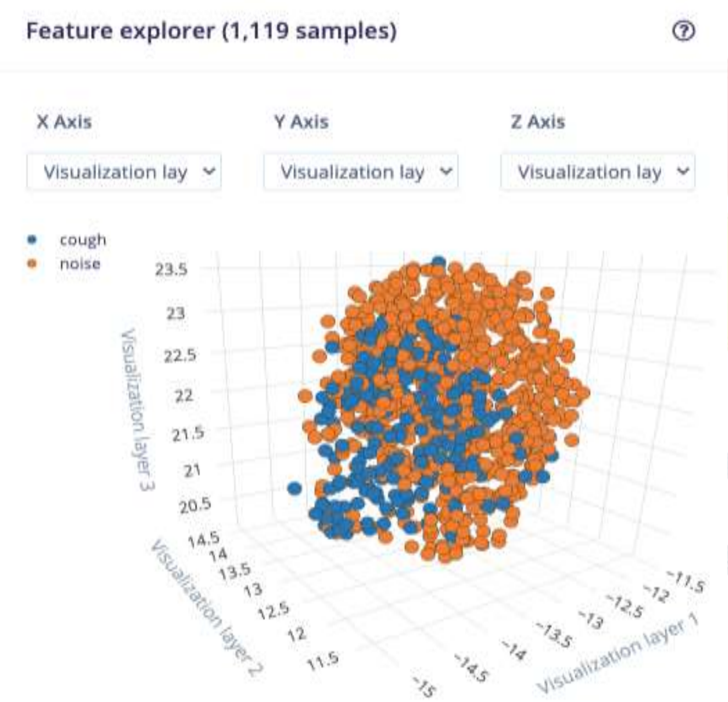
classification. Therefore we are going to switch to expert mode. For programming tensorflow is used which is an open source for machine learning programming.
Then we generate features, which generate MFCC blocks for all of our windows of audio. After that we go to Neural Network Settings and Switch to Keras mode.
After training the model, it will show the training performance.

Deployment to Arduino library:
We can easily deploy our cough detection algorithm. By just going to the classification mode it will automatically build the project into a WebAssembly package and execute it. There is no cloud storage required for this purpose.
Next we can deploy the algorithm to the Nano BLE Sense by goin to the Deployment page. And selecting "Arduino Nano 33 BLE Sense" under 'Build Firmware" and the build.
This will build a complete firmware for the Nano BLE Sense including the latest algorithm.
Once the Arduino is flashed, we can open a serial port to device while it is plugged into USB at 115, 200 baud.

TESTS:
We performed various tests where we tried to increase the accuracy of the model and decrease the loss. We researched through certain websites such as Kaggle where certain datasets where obtained. We tried getting data from normal microphone but it was not that helpful as certain level of distortion was present in the sample. After getting the sufficient data we were able to train the model and get a decent accuracy and loss.

RESULTS:
The above-mentioned tests were been conducted and the accuracy and loss were being identified. But if the quality of dataset used in training and testing is improvised they results might differ.

MFCC Block



MEL Frequency Cepstral Coefficients

Number of Coefficients: 13

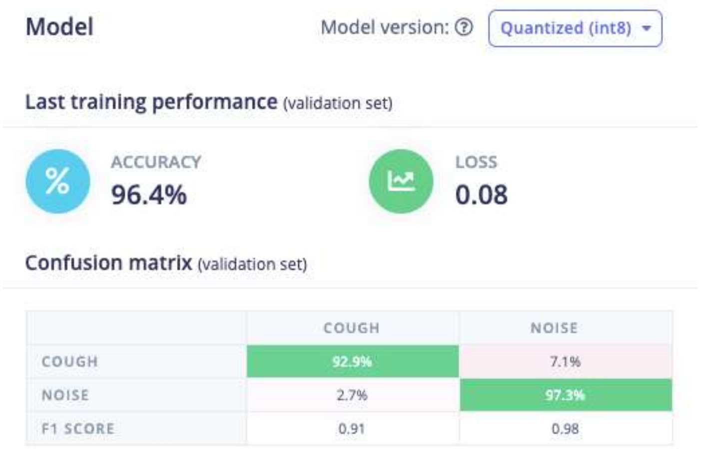
Processing Time: 217 ms

Peak RAM usage: 17 KB

Data in Training set: 9m 32s

Classes: 2(cough, noise)

Final Accuracy and Loss results:



FUTURE DIRECTIONS:

AI-based ML and DL applications in COVID-19 and cough detection can be further developed in the following ways:
Detection of non-contact disease: By using automated image classification. Further, digital image processing can be helpful in processing X-ray and CT images.
Video diagnosis: In rural areas where there are no healthcare facilities use of AI and ML can be helpful in building remote video diagnosis in order to provide treatments for patients.
Drug Development: By using AI and ML, we can mimic drug protein and receptors, thus predicting future drug and vaccine reactions.

CONCLUSIONS

This paper proposes a technique of Cough Detection on Arduino using ML. The poposed approach relies on alrge variety of datasets used for proper differentiating between the audio. The sky is the limit with TinyML, sensors and Edge Impulse on Arduino. In future, due to improvisation in the audio technology the datasets can be without background noise. We can conclude that we were successfully able to build a working model that would play a major role in cough detection which is cheap and portable at the same time thus being available to more people.

ACKNOWLEDGMENTS

We the authors of this project paper acknowledge the information used to prepare the paper is true to our knowledge. We also acknowledge that all the references used to prepare this paper are mentioned in the References section and they hold the copyright to that information.

REFERENCES

1]HindawiJournal of SensorsVolume 2018, Article ID 9845321, https://doi.org/10.1155/2018/9845321
2]International Journal of Research and Reviews in Applied Sciences And Engineering (IJRRASE) Vol 8. No.1 – 2016 Pp.81-84 ©gopalax Journals, Singapore.
3]Thomas Drugman, Jerome Urbain, Nathalie Bauwens,Ricardo Chessini, Anne-Sophie Aubriot, Patrick Lebecque, Thierry Dutoit TCTS Lab - University of Mons, Belgium2Pediatric Pulmonology & Cystic Fibrosis Unit, Cliniques St Luc, University of Louvain, Belgium.
4]Birring S., Matos, S., Evans, D., Pavord, I., “Leicester Cough Monitor: a novel automated cough detection system”, Am J Resp Crit Care Med, 175:A379, 2007.
5]Sensor-based System for Automatic Cough Detection and Classification Sidi Ahmed Mahmoudi, Paulo Da Cunha Possa, Thierry Ravet, Thomas Drugman, Ricardo Chessini, Thierry Dutoit and Carlos Valderrama.
6]Cough Detection with TinyML on Arduino Zach Shelby
7] Automatic cough detection based on airflow signals for portable spirometry system.
8] 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.
9] Machine and Deep Learning towards COVID-19 Diagnosis and Treatment by Tarik Alafif, Abdul Muneeim Tehame, Saleh Bajaba, Ahmed Barnawi and Saad Zia