VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-18, Karnataka, India.

A Mini Project Report on

“Novel preliminary detection system on Edge for COVID-19”

Mini Project Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

In

Telecommunication Engineering

By

MOHIT SENAPATI	**1DS17TE083**
AISHWARYA GOANKAR	**1DS17TE006**
DAVALASHRI PRASAD	**1DS16TE073**
PRIYA C	**1DS17TE042**

6th sem B.E

Under the guidance of

DR. GANASHREE T S

Assistant Professor

DEPARTMENT OF TCE, DSCE, BENGALURU

Department of Telecommunication Engineering

DAYANANDA SAGAR COLLEGE OF ENGINEERING
BENGALURU -560078.

2021-22

DAYANANDA SAGAR COLLEGE OF ENGINEERING

S M Hills, Kumaraswamy Layout, Bengaluru-560078

CERTIFICATE

This is to certify that the mini project work entitled **"Novel preliminary detection system on Edge for COVID-19"** is a bonafide work carried out by **MOHIT SENAPATI (1DS18TE083), AISHWARYA GOANKAR (1DS18TE006), DAVALASHRI PRASAD (1DS18TE073), PRIYA C (1DS18TE042),** students of 6th semester, Dept. of Telecommunication Engineering, **DSCE** in partial fulfillment for award of degree of **Bachelor of Engineering** in **Telecommunication Engineering** ,under the **Visvesvaraya Technological University**, **Belagavi** during the year 2021-22. The mini project has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the bachelor of engineering degree.

Signature of Guide	Signature of HOD	Signature of Principal
DR. GANASHREE T S	**Dr. A R ASWATHA**	**Dr. C.P.S. PRAKASH**
Assistant Professor	Professor & Head	Principal
Dept. of TCE	Dept. of TCE	DSCE
DSCE, Bangalore	DSCE, Bangalore	Bangalore

Name of Examiners	**Signature & Date**
1…………………………........	**……………………..**
2…………………………........	**……………………..**

ACKNOWLEDGEMENT

The success and outcome of this mini project require the guidance and assistance of many people. We would like to add a few words of appreciation for the people who have been part of this mini project right from its inception, without their support patience and guidance the task would not have been completed. It is to them we owe them our deepest gratitude.

We are grateful to **Dr. C.P.S PRAKASH,** Principal, Dayananda Sagar College of Engineering, for providing an opportunity to do this mini project as a part of our curriculum and for his kind cooperation for the mini project.

We are very much grateful to **Dr. A R ASWATHA,** Professor and Head, Department of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore for providing the encouragement for completion of our mini project.

We would like to express our deep gratitude to our guide **DR. GANASHREE T S**, **A**ssistant Professor of Telecommunication Engineering Department for his valuable guidance, patience, constant supervision and timely suggestions provided in making of this mini project.

I also thank our Mini-Project coordinator **Mr. VINOD KUMAR H**, **A**ssistant Professor of Telecommunication Engineering Department for his support throughout the Mini Project Phases.

We are also thankful to our parents and friends for their constant help and constructive suggestions throughout our mini project.

Mohit Senapati	**1DS17TE083**
Aishwarya Goankar	**1DS17TE006**
Davalashri Prasad	**1DS16TE073**
Priya C	**1DS17TE042**

ABSTRACT

With the unprecedented outbreak of SARS-CoV-2, life has come to a standstill. With the second wave of infections hitting India in more drastic way than the former, people across the country are turning towards aids that help trace, track or alleviate infections and related health problems. Initial symptoms resemble that of a normal seasonal flu, with cold, cough and fever, and gradually worsens to mainly affect the respiratory system (i.e. lungs). Due to the viral nature of the infection, transmissibility is high, and various precautionary methods like social distancing, wearing a mask/PPE kits etc. are encouraged. But immediate medical attention is paramount, once infected. Since the symptoms initially resemble a benign common cold, many people opt out from preliminary testing –which can prove to be dangerous later. To aid in a convenient way of testing for COVID-19 at an initial stage, a simple and cost effective way of detecting the early symptoms is proposed. The proposed project intends to use Machine Learning on Edge (Edge ML) and audio processing to detect presence of COVID-19 through the coughing patterns of a person. This uses an expandable dataset to create Neural Networks and deploy it over a smartphone to use existing hardware, minimizing cost and any additional peripherals. The trained models can also be deployed on standalone models (with microprocessors) to create dedicated devices for this purpose.

Keywords: **SARS-Cov-2, Neural Networks, Machine Learning, Edge ML, Audio processing.**

Contents

CHAPTER 1

INTRODUCTION

COVID-19 (Corona virus) is a contagious disease caused by severe acute respiratory syndrome. This disease has spread worldwide leading to an ongoing pandemic. Symptoms of COVID-19 are variable but usually include fever, cough, headache, fatigue, breathing difficulty, loss of smell and taste. Symptoms may begin within one to fourteen days after exposure to the Coronavirus. But a few of the infected people do not exhibit any symptoms at all. People remain contagious for up to 20 days, and can spread the virus even if they do not develop any symptoms. Several testing methods have been developed to diagnose the disease. But early detection of the virus in an infected person is very essential at the moment so that necessary steps can be taken against further spread of the disease. The demand for COVID-19 testing has increased, laboratory professionals have faced a growing list of challenges, uncertainties, and in some situations, controversy, as they have attempted to balance the need for increasing test capacity with maintaining a high-quality laboratory operation.

Our project proposes an easy method of self testing to determine early symptoms of COVID-19, whether or not the symptoms are noticeable, with it also being easily accessible to almost everyone. We shall be using cough sounds to determine if the user has COVID or not. The fluctuations in coughing sounds of infected and uninfected people will be taken, these audio samples will be pre-processed using digital signal processing methods and can be used as a database to train an Artificial Neural Network. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes. The nodes can take input data and perform simple operations on the data.

The final Machine Learning model will be deployed on a user friendly device such as a mobile phone or a standalone audio capturing equipment. We choose to deploy this model on Edge so that the latency requirements are reduced while processing the data and network costs are saved. It will also help in enhancing the accuracy and speed of the model. A user just needs to record his coughing, and the model should predict whether the person is affected or not.

CHAPTER 2

LITERATURE SURVEY

2.1 "Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods" by Mucahid Barstugan, Umut Ozkaya, Saban Ozturk.

In recent times The main objective of this paper is coronavirus classification based on CT images using machine learning algorithm. The CT images of the infected people shows that COVID-19 disease has own characteristics. Therefore, the clinical experts need lung CT images to diagnose the COVID-19 in early phase.

This study used 150 CT images for COVID-19 classification. The patch regions were cropped on 150 CT images and four different patch subsets were created. This study performs a coronavirus classification in two stages.

In the first stage, the classification process was implemented on four different subsets without feature extraction process. The subsets were transformed into vector and classified by SVM.

In the second stage, five different feature extraction methods such as Grey Level Co-occurrence Matrix (GLCM), Local Directional Patterns (LDP), Grey Level Run Length Matrix (GLRLM), Grey Level Size Zone Matrix (GLSZM), and Discrete Wavelet Transform (DWT) extracted the features and the features were classified by SVM. During the classification process, 2-fold, 5-fold, and 10-fold cross-validation methods were used. The mean classification results after cross-validations were obtained.

The feature extraction methods used in this study are as follows:

- **Grey Level Co-occurrence Matrix :** GLCM is used to obtain the second-degree statistical features on the images. GLCM consists of the relationships of different angles between the pixels of an image. GLCM method produces 1x19 feature vector for classifier input.

- **Local Directional Pattern :** LDP method uses Kirsch compass kernels to combine the directional elements. LDP method produces output matrix sized as input image. This matrix is transformed into a vector for classifier input.

Grey Level Run Length Matrix : GLRLM extracts texture features on a high level. Let L be the number of grey-levels, R is the longest run, and P is the number of pixels in the image. A GLRLM matrix is L×R, and each $p(i,j \mid \theta)$ element gives the number of occurrences in the θ direction with i grey level and j run length. GLRLM method produces 1x7 feature vector for classifier input.

- **Grey Level Size Zone Matrix :** GLSZM is a feature extraction method, which is developed version of GLRLM algorithm. GLSZM method produces 1x13 feature vector for classifier input.
- **Discrete Wavelet Transform :** DWT separates the image into frequency sub-bands by using an h low-pass filter and g high-pass filter. After DWT. The LL coefficients were obtained by db1 wavelet, and the coefficient matrix were transformed into a feature vector.
- **Support vector machine [SVM] :** SVM gives high classification accuracy in many applications. An SVM is based on two ideas. The first idea is to map feature vectors to a high dimensional space with a nonlinear method and to use linear classifiers in this new space.

DWT feature extraction method gives the best classification result that is 97.28% in Stage 2 with 10-fold cross validation. GLCM, GLSZM and DWT methods always had classification accuracy over 90% during 10-fold cross validation. The best classification performance was achieved by using GLSZM method with 5-fold cross-validation. GLSZM method extracts the features of the patches and form feature vector. The vector is classified by five different SVM structures, which were obtained during training phase. The mean classification performance is obtained by SVM classification.

2.2 "Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study" by Davide Brinati, Andrea Campagner, Davide Ferrari, Massimo Locatelli3, Giuseppe Banfi, Federico Cabitza

Intelligent The aim of this work is to develop a predictive model, based on Machine Learning techniques, to predict the positivity or negativity for COVID-19. The dataset used for this study was made available by the IRCCS Ospedale San Raffaele and it consisted of 279 cases, randomly extracted from patients admitted to that hospital from the end of February 2020 to mid of March 2020. Each case included the patient's age, gender, values from routine blood tests extracted as in, and the result of the RT-PCR test for COVID-19, performed by nasopharyngeal swab.

The dependent variable "Swab" is binary and it is equal to 0 in the absence of COVID-19 infection (negative swab test), and it is equal to 1 in the case of COVID-19 infection (positive to the swab test). The number of occurrences for the negative and positive class was respectively 102 (37%) and 177 (63%), thus the dataset was slightly imbalanced towards positive cases.

the categorical feature Gender has been transformed into two binary features by one-hot encoding. To address data incompleteness, missing data imputation is done by means of the Multivariate Imputation by Chained Equation (MICE) method. MICE is a multiple imputation method that works in an iterative fashion: in each imputation round, one feature with missing values is selected and is modeled as a function of all the other features, the estimated values are then used to impute the missing values and re-used in the subsequent imputation rounds.

Model training selection and evaluation is done using different models, this study considers the following classifier models:

- Decision Tree (DT)
- Extremely Randomized Trees (ET)
- K-nearest neighbor (KNN)
- Logistic Regression (LR)
- Na¨ıve Bayes (NB)
- Random Forest(RF)
- Support Vector Machines [41] (SVM)

This study also considered a modification of the Random Forest algorithm, called three-way Random Forest classifier (TWRF), which allows the model to abstain on instances, a TWFR achieves higher accuracy on the effectively classified instances at expense of coverage (i.e., the number of instances on which it makes a prediction). This class of model provides more reliable predictions for large data sets, while exposing the uncertainty regarding other cases so as to suggest further (and more expensive) tests on them.

From a technical point of view, Random Forest is an ensemble algorithm that relies on a collection of Decision Trees that are trained on mutually independent subsets of the original data in order to obtain a classifier with lower variance and lower bias. The independent datasets, on which the Decision Trees in the forest are trained, are obtained from an original dataset by both sampling with replacement the instance and selecting a random subset of the features. As Random Forest are a class of probability scoring classifiers, the abstention is performed on the basis of two thresholds $\alpha, \beta \in [0, 1]$: if we denote with 1 the positive class and 0 the negative class, then each instance is classified as positive if score (1)>α and score (1) > score (0), negative if score (0) > β and score (0) > score(1) and otherwise, the model abstains.

The models mentioned above have been trained, and evaluated, through a nested cross validation procedure. An inner cross-validation loop is executed to find the optimal hyperparameters via grid search and an outer loop evaluates the model performance on five folds.

To further validate the above findings, the entire dataset has been splitted into training and test/validation sets, respectively the 80% and the 20% of the total instances. Two models, Logistic Regression and Random Forest, exhibited comparable performance (difference less than 1%) in terms of AUC (LR = 85%, RF = 84 %) and sensitivity (LR = 93%, RF = 92%), but Random Forest reported higher performance in terms of accuracy (LR = 78%, RF = 82%) and much higher specificity (LR = 50%, RF = 65%): thus, Random Forest was selected as reference best performing model.

2.3 "Music Genre Classification using Machine Learning Techniques", by Hareesh Bahuleyan.

The main objective of this paper is to classify the music genre by using machine learning techniques. In this work, Audio Sets are used, which is a large-scale human annotated database of sounds. The dataset was created by extracting 10-second sound clips from a total of 2.1 million YouTube videos. The audio files have been annotated on the basis of an ontology which covers 527 classes of sounds including musical instruments, speech, vehicle sound animal. This study requires only the audio files that belong to the music category, specifically having one of the seven genre tags.

Data Pre-processing is one In order to improve the Signal-to-Noise Ratio (SNR) of the signal, a pre-emphasis filter, given by Equation is applied to the original audio signal.

$y(t) = x(t) - \alpha * x(t - 1)$

where, $x(t)$ refers to the original signal, and $y(t)$ refers to the filtered signal and α is set to 0.97. Such a pre-emphasis filter is useful to boost amplitudes at high frequencies

Using deep learning, the music genre classification can be done without the need for hand-crafted features. Convolutional neural networks (CNNs) have been widely used for the task of image classification. The 3-channel (RGB) matrix representation of an image is fed into a CNN which is trained to predict the image class. In this study, the sound wave can be represented as a spectrogram, which in turn can be treated as an image. The task of the CNN is to use the spectrogram to predict the genre label (one of seven classes).

A spectrogram is a 2D representation of a signal, having time on the x-axis and frequency on the y-axis. A colormap is used to quantify the magnitude of a given frequency within a given time window. In this study, each audio signal was converted into a MEL spectrogram (having MEL frequency bins on the y-axis).

The parameters used to generate the power spectrogram using STFT are listed below:

- Sampling rate (sr) = 22050
- Frame/Window size (n fft) = 2048
- Time advance between frames (hop size) = 512 (resulting in 75% overlap)
- Window Function: Hann Window
- Frequency Scale: MEL
- Number of MEL bins: 96
- Highest Frequency (f max) = sr/2

In this work, the task of music genre classification is studied using the Audioset data pose two different approaches to solving this problem. The first involves generating a spectrogram of the audio signal and treating it as an image.

An CNN based image classifier, namely VGG-16 is trained on these images to predict the music genre solely based on this spectrogram. The second approach consists of extracting time domain and frequency domain features from the audio signals, followed by training traditional machine learning classifiers based on these features. XGBoost was determined to be the best feature-based classifier; the most important features were also reported.

The CNN based deep learning models were shown to outperform the feature-engineered models. We also show that ensembling the CNN and XGBoost model proved to be beneficial. It is to be noted that the dataset used in this study was audio clips from YouTube videos, which are in general very noisy. Futures studies can identify ways to pre-process this noisy data before feeding it into a machine learning model, in order to achieve better performance.

2.4 "Two Convolutional Neural Networks for Bird Detection in Audio Signals" by Thomas Grill, Jan Schlüter

The aim of the paper is to classify bird audio signal based on based on two convolutional neural networks. For this study the provided training data comes from freefield1010 (7690 examples) and Warblr (8000 examples), the testing data mostly from Chernobyl and to a smaller extent from Warblr (8620 examples altogether).

The representation of the data which is used for machine learning consists of Mel-scaled log-magnitude spectrograms with 80 bands. In order to obtain a clearer picture of the data structure, we performed clustering on some simple features derived from those spectrograms.

This approach to the Bird audio detection challenge deploys feed-forward CNNs trained on Mel-scaled log-magnitude spectrograms

For each audio file under analysis, need to compute an STFT magnitude spectrogram with a window size of 1024 samples at 22.05 kHz sample rate with 70 frames per second (hop size 315 frames), apply a mel-scaled filter bank of n = 80 triangular filters from 50 Hz to 11 kHz (bulbul) or 10 kHz (sparrow) and scale magnitudes logarithmically.

Two types of CNN architecture is used here,

- **Global architecture (Submission bulbul) :** This highest-scoring submission to the challenge uses a network with a wide receptive field of 1000 frames processed into a single binary output., a sequence of four combinations of convolution and pooling condenses the input of 1000×80 into 16 feature maps of 11×8 units.

- **.Local architecture (Submission sparrow) :** this method is used for prediction of file with no labels. Since there is no label of short excerpts, only for a full recording, this is a multiple-instance learning problem. It follows the standard MI assumption, a recording is labeled positively if and only if at least one of its excerpts is positive.

Training is done by stochastic gradient descent on mini batches of 64 (bulbul) or 32 (sparrow) examples, using the ADAM update rule with an initial learning rate of 0.001, reduced by a factor of 10 two times during training. sparrow uses a fixed scheme, training for 80,000 updates with learning rate drops after 40,000 and 60,000 updates. bulbul uses a variable scheme dropping the learning rate whenever the training error does not improve over three consecutive episodes of 1500 updates, resulting in about the same number of updates. sparrow is trained on excerpts of 701 frames, bulbul on 1000 frames. Files shorter than required are looped up to the length needed. After training, to obtain a prediction for a recording, loop it as needed to fill the network's receptive field.

To improve results, for both submissions, average the file-wise predictions of five networks trained on each of five cross-validation splits of the training data.

The Bird audio detection challenge featured a submission site where contestants could upload their predictions for the test set, at most once every 24 hours. A 'preview score' was then computed giving the AUC (area under ROC curve) for a subset of 1293 files from the test set. Scores for the full test set were published after the contest deadline, deviating from the preview scores by some tenths of a percent for the top submissions. For development, AUC is computed using five-fold cross-validation on the training set

This paper presented two deep learning based approaches for detecting bird calls in audio recordings. Despite using different network architectures, they perform very similarly. Moreover, they perform on par with other top submissions to the QMUL bird audio detection challenge (AUC 88.7% for our bulbul system, and 88.5%, 88.2%, 88.1%, 88.1% for the next four contestants).

2.5 "Machine Learning Algorithms for Environmental Environmental Sound Recognition: Towards Soundscape Semantics" by Vasileios Bountourakis, Lazaros Vrysis, George Papanikolaou

This paper investigates methods aiming at the automatic recognition and classification of discrete environmental sounds, for the purpose of subsequently applying these methods to the recognition of soundscapes. The main objective of this paper is to perform a comparison between the most commonly used methods in the ESR field. Six machine learning algorithms were tested in combination with three different feature sets, which resulted from a feature selection process.

Feature extraction is a process of transforming audio data from a high to a low dimensional representation. In practice, the abstracted features are supposed to describe some useful aspects of the original data. For this reason, features are sometimes referred to as 'descriptors'.

High dimensional feature sets do not necessarily lead to good performance. As the feature dimension increases, data points become sparser and there are potentially irrelevant features that could negatively impact the classification result.

A smaller feature set containing only the most significant features is better, since it reduces the computational complexity and, consequently, running time, while it can possibly lead to a higher performance. For this reason, it is common practice to select an optimal subset of features from the larger set of the extracted ones, by discarding those features that contribute less to the discrimination between the classes and those that are linearly correlated with other features. The selection of the salient audio features was performed in the software environment of Weka.

To perform the classification process, we used inbuilt implementations of the following algorithms in the environment of Weka:

- k-Nearest Neighbors (k-NN)
- Naive Bayes
- Support Vector Machines (SVM)
- C4.5 algorithm (decision tree)
- Logistic Regression
- Artificial Neural Networks (ANN)

The choice of the particular classifiers was based on their performance in similar classification tasks. Every algorithm is tested with each of the three aforementioned feature sets as an input.

The results of the aforementioned parameter analysis were:

- for k = 8, k-NN achieved recognition rate of 87.52%
- for LR = 0.5, ANN achieved recognition rate of 87.3%
- for C = 2, SVM achieved recognition rate of 87.04%

In evaluation, the detailed results of the final classification scheme is presented in terms of precision, recall and F-measure. The algorithms are also compared with respect to the time taken for their model to be built in our experiments, so as to evaluate their potential use in real-time applications. precision and recall are the basic metrics used in evaluating classifier output quality.

Regarding the classifiers, k-NN, SVM and ANN provided the highest recognition rates among the algorithms used in this study. However, although the three algorithms showed comparable performance, k-NN was proved significantly faster in constructing the training model, with SVM following and ANN being the slowest. It should be also mentioned that PCA, apart from minimizing the computational complexity, it also improved the performance of the SVM and ANN algorithms.

CHAPTER 3

COVID-19 DETECTION PARAMETERS AND TESTING

Symptoms of COVID-19 are ranging from mild symptoms to severe illness. Common symptoms include headache, loss of smell and taste, nasal congestion and runny nose, cough, muscle pain, sore throught, fever, diarrhea, and breathing difficulties. People with the same infection may have different symptoms and their symptoms may change over time. Three common clusters of symptoms have been identified: one respiratory symptom cluster with cough, sputum, shortness of breath, and fever; a musculoskeletal symptom cluster with muscle and joint pain, headache, and fatigue; a cluster of digestive symptoms with abdominal pain, vomiting, and diarrhea. In people without prior ear, nose, and throat disorders, loss of taste combined with loss of smell is associated with COVID-19.

COVID-19 can provisionally be diagnosed on the basis of symptoms and confirmed using reverse transcription polymerase chain reaction (RT-PCR) or other nucleic acid testing of infected secretions. Along with laboratory testing, chest CT scans may be helpful to diagnose COVID-19 in individuals with a high clinical suspicion of infection. Detection of a past infection is possible with serological tests, which detect antibodies produced by the body in response to the infection.

3.1 Viral testing : The standard methods of testing for presence of SARS-CoV-2 are nucleic acid tests, which detects the presence of viral RNA fragments. As these tests detect RNA but not infectious virus, its ability to determine duration of infectivity of patients is limited. The test is typically done on respiratory samples obtained by a nasopharyngeal swab. A nasal swab or sputum sample may also be used. Results are generally available within hours. Molecular tests detect genetic material – the RNA – of the coronavirus and are sensitive enough to need only a very tiny amount of it. Until now, the best PCR tests generally required trained personnel, specific reagents and expensive machines. The sample is collected with a nasal or throat swab and they tend to take hours to provide results. Good PCR tests like the ones used over the past eight months at UC Davis Health’s lab are close to 100% accurate. However, not all molecular tests, including PCR methods, are perfect. Some lesser testing platforms have reported false negative rates as high as 15% to 20%. Both of UC Davis Health’s tests, the rapid COVID-19/flu test and the lab test for COVID-19, are highly sensitive, highly specific PCR tests. The sensitivity of molecular methods can be a double-edged sword. In some cases, it can still detect the virus' genetic material after a patient has recovered from a COVID-19 infection and is no longer contagious.

3.2 Antigen tests : An antigen is a substance recognized by the body's immune system, which can then respond by generating proteins called antibodies that specifically recognize that antigen. The point of an antigen test is to detect the presence of a protein—the nucleocapsid protein—which is part of the SARS-CoV-2 virus that is the cause of COVID-19. According to the FDA, antigen tests are collected via nasal cavity swabs, which are then placed into a special solution for virus detection. In addition to quick results, antigen tests are also cheaper and easier to use, compared to other tests available. PCR tests, per the FDA, detect the genetic material from the virus—or the virus' RNA—which can help diagnose an active COVID-19 infection. That's different than antigen tests, which, again, test for the virus' proteins. Up until now, the tests used to detect active infections of the virus detect the genetic material of the virus, not proteins. These PCR tests, however, are still done through nasal or throat swabs. Those antibody tests are also done through testing a person's blood serum or plasma. While antibodies and antigens are both typically proteins, a positive antigen test reflects active infection, while a positive antibody test reflects recent or past infection. While antigen tests are notably quicker than PCR tests, the downside to increased testing speed may be decreased accuracy: The FDA says antigen tests aren't as specific at PCR tests, and may provide false negatives, which then need to be confirmed through a PCR test. Luckily, per the FDA, positive results from antigen tests are highly accurate. Antigen tests also aren't designed for home use and requires a specialized instrument to be run in certified laboratories. Depending on the quality of the antigen test and the test takers, false negatives could be as high as 20%

3.3 Imaging : Computer Tomography also known as CT Scan or CAT scan is a diagnostic tool that uses a combination of x-rays and computer imaging allowing doctors to see the condition of organs, bones, blood vessels and tissues. A CT Scan can be done on any body part, it is a non-invasive, painless procedure which gives detailed images. Chest CT scans may be helpful to diagnose COVID-19 in individuals with a high clinical suspicion of infection but are not recommended for routine screening. In the patients with coronavirus, the radiologist looks for Ground glass Opacity (GGO). If the findings reveal patchy, bilateral, peripheral and subpleural markings, a scoring is given based on the severity for the doctor can decide on the further course of treatment. Bilateral multilobar ground-glass opacities with a peripheral, asymmetric, and posterior distribution are common in early infection. Subpleural dominance, crazy paving (lobular septal thickening with variable alveolar filling), and consolidation may appear as the disease progresses. Characteristic imaging features on chest radiographs and computed tomography (CT) of people who are symptomatic include asymmetric peripheral ground-glass opacities without pleural effusions.

CHAPTER 4

PROPOSED SOLUTION

4.1 DESIGN :

The proposed solution for detection of COVID-19 requires a method of deployment that streamlines the process further. To do that, the following parameters were taken for consideration:

i) Ease of use
ii) Fast data processing for better results
iii) Lesser hardware redundancy

Ease of use ensures that every user has a seamless deployment experience, despite their level of technical competence. This was specially taken into consideration given the fact that the most susceptible demographic to the disease is the elderly, who may not have the technical know-how to operate a more complex user interface.

Another consideration for making the proposed solution into a deployable model was faster data processing. As discussed earlier, current COVID-19 detection tests may take a couple of hours to days to determine whether a user is COVID positive or negative. A major goal for the proposed solution was faster and reliable test results –which eventually not only saved time, but also may ease the pressure on hospital resources like testing equipment, consultation time etc.

And finally, upgradability of the device, or lesser hardware redundancy, was considered. This was taken into account provided the fact that technology is constantly evolving, and hardware upgradability is important to get up-to-date peripherals that can increase efficiency of the device. Taking hardware redundancy into account also brings multiple options to choose from, catering to different people for different prices.

With all these considerations, two methods were chosen to be best for deployment –

i) Local deployment to create a testing device
ii) Using existing devices like smartphones and its peripherals

i) Local deployment to create a testing device

This method uses the proposed model to be deployed locally, on a device like an Arduino Nano 33 BLE Sense, or an AVR IOT Development Board. The idea is to make a standalone device that a user can always have on themselves, ready to be used anytime. The features proposed for such model are:

- Powered by rechargeable battery pack to maximize reusability.
- Portable 'on-board' model provides a small footprint and ample mobility.
- Local processing for faster speeds.
- Convenience to choose hardware as per user's needs, and upgrade them if required.
- Inexpensive to build and deploy.

Although this method provides great compatibility with build and deployment, another better way for this task was to use existing hardware and retrofit them, as shown in the next method.

ii) Using existing devices like smartphones and its peripherals

In this process, the created model is deployed over an existing hardware, like a mobile phone, via an application. This has many advantages like:

- Saving time and money that otherwise might have been used unnecessarily to create a new peripheral from scratch.
- Most people own smartphones, making the proposed solution available widely, and making it easily distributable to users.
- Modern smartphones contain high quality sensors, fast processors and fast network availability. All of this can aid in yielding a highly accurate result for the proposed COVID detection test.
- App based user interface is more intuitive, easier to learn, and easier to use for first time users.

The proposed solution for COVID testing is deployed over a smartphone either via a native application on the phone itself, or via a web application (as demonstrated in the project). All a user has to do is to use the app near the patient while they cough. The in-built microphone. The sound, once captured, is run through the ML algorithm (discussed in the next section), which compares it to the pre-trained model, and based on the various methodologies used, determines it as positive or negative.

4.2 DATA CAPTURE AND PROCESSING (ML MODEL) :

As mentioned earlier, the in-built microphones of the smartphone is used as an input to capture the cough sounds from the patient.

To process this data, a Machine Learning (ML) model was created to determine whether the recorded cough was positive or negative. This Machine Learning model was built using Edge Impulse, and the code for this was edited on Python. Tensorflow Lite was used as the framework to build the proposed solution.

Some of the key steps in making the ML model for the solution were:

- Data collection: Various cough sound samples were collected from different sources and databases (like Kaggle), along with custom cough inputs. This data is pre-processed to some extent by clipping it into segments for short lengths, and eliminating 'noise' by clipping the redundant (non-distorted) waveform as seen on the visualizer.
- Once data is filtered, it is categorized as negative and positive by assigning tags to each group.
- Since different people have different voice modulations, frequencies of speech and amplitude, we pass the pre-processed data through MFC (Mel Frequency Cepstrum) filter, to obtain MFCC (Mel Frequency Cepstrum Coefficients). This converts raw audio format to a format suitable for computation for Machine Learning algorithms. This makes data readability higher.
- Finally, a neural network is built on the predecessor.

The block diagram in Figure 1 shows a flowchart on how the data is processed through various stages till deployment.

4.3 BUILDING THE MODEL AND DEPLOYMENT :

As discussed previously, the trained N-N classifier (i.e, the algorithm to classify the cough sounds to determine COVID) can be either deployed locally via an application, or a web based application.

To ensure that the proposed model runs smoothly on all kinds of devices, it is deployed on a web based application.

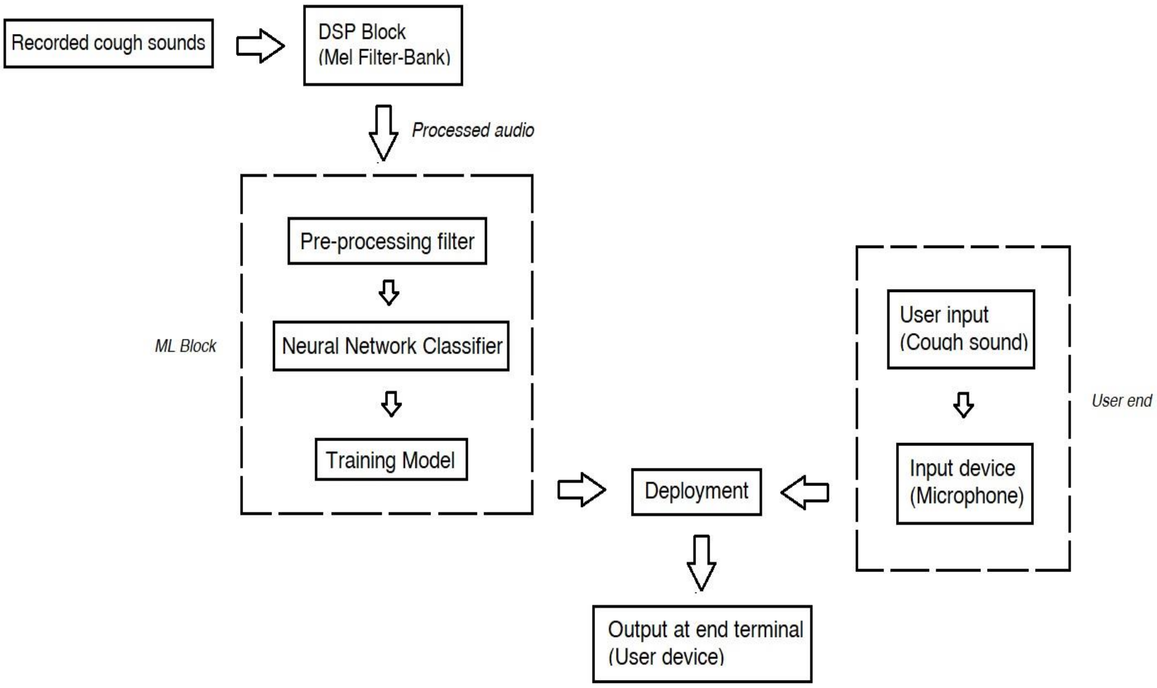

Figure 1 Operational flowchart of the project

To demonstrate this, the remote deployment server on Edge Impulse is used. A secure two-way connection is made between the end-device and the cloud via secured encryption. Once this is set up, the algorithm is locally uploaded onto the phone temporarily.

A brief summary of the methodologies used in the project have been elucidated in the next chapter.

CHAPTER 5

METHODOLOGY

5.1 ARTIFICIAL NEURAL NETWORK (ANN) :

Artificial Neural Networks are used in various classification task like image, audio, words. Different types of Neural Networks are used for different purposes.

In Neural Network there are three types of layers:

1. Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data.
2. Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into probability score of each class.
3. Hidden Layer: The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layers can have different numbers of neurons which are generally greater than the number of features.

The data is then fed into the model and output from each layer is obtained this step is called feed forward, we then calculate the error using an error function, some common error functions are cross entropy, square loss error etc. After that, we back propagate into the model by calculating the derivatives. This step is called Back propagation which basically is used to minimize the loss.

5.2 CONVOLUTIONAL NEURAL NETWORK (CNN) :

CNN is a class of artificial neural network, most commonly applied to analyze visual imagery. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer.

Convolution layers consist of a set of learnable filters . Every filter has small width and height and the same depth as that of input volume .

Types of layers:

1. **Input Layer:** This layer holds the raw input of image with width, height and depth.
2. **Convolution Layer:** This layer computes the output volume by computing dot product between all filters and image patch.
3. **Activation Function Layer:** This layer will apply element wise activation function to the output of convolution layer.
4. Pool Layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents from overfitting.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer.CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered.The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution.Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

5.2.1 ARCHITECTURE OF CNN

A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. This product is usually the Frobenius inner product, and its activation function is commonly ReLU. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers.CNNs are often used in image recognition systems. In 2012 an error rate of 0.23% on the MNIST database was reported. CNN called AlexNet won the ImageNet Large Scale Visual Recognition Challenge 2012.

5.2.2 BLOCK DIAGRAM OF CNN :

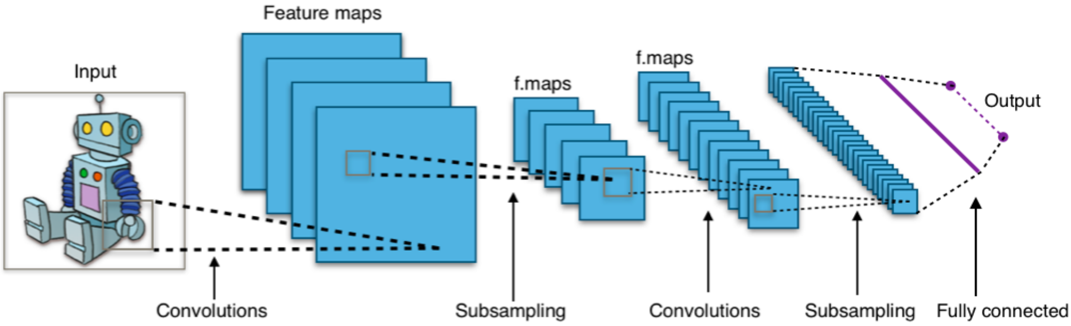

Figure 2 Block Diagram of a CNN

5.3 DIGITAL SIGNAL PROCESSING (DSP)

Digital signal processing (DSP) is the use of digital processing, such as by computers or more specialized digital signal processors, to perform a wide variety of signal processing operations.DSP is applicable to both streaming data and static (stored) data. The digital signals processed in this manner are a sequence of numbers that represent samples of a continuous variable in a domain such as time, space, or frequency.

Figure 3 Block Diagram of Digital Signal Processing

Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repeatedly on a series of data samples. Sampling is usually carried out in two stages, discretization and quantization. Discretization means that the signal is divided into equal intervals of time, and each interval is represented by a single measurement of amplitude. Quantization means each amplitude measurement is approximated by a value from a finite set. Signals (perhaps from audio or video sensors) are constantly converted from analog to digital, manipulated digitally, and then converted back to analog form. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time, and deferred (or batch) processing is not viable.

Theoretical DSP analyses and derivations are typically performed on discrete-time signal models with no amplitude inaccuracies (quantization error), "created" by the abstract process of sampling. Numerical methods require a quantized signal, such as those produced by an ADC. The processed result might be a frequency spectrum or a set of statistics. But often it is another quantized signal that is converted back to analog form by a digital-to-analog converter (DAC).

Signals are converted from time or space domain to the frequency domain usually through use of the Fourier transform. The Fourier transform converts the time or space information to a magnitude and phase component of each frequency.

5.3.1 SOFTWARE ARCHITECTURE OF DSP:

By the standards of general-purpose processors, DSP instruction sets are often highly irregular; while traditional instruction sets are made up of more general instructions that allow them to perform a wider variety of operations, instruction sets optimized for digital signal processing contain instructions for common mathematical operations that occur frequently in DSP calculations. Both traditional and DSP-optimized instruction sets are able to compute any arbitrary operation but an operation that might require multiple ARM or x86 instructions to compute might require only one instruction in a DSP optimized instruction set.

One implication for software architecture is that hand-optimized assembly-code routines (assembly programs) are commonly packaged into libraries for re-use, instead of relying on advanced compiler technologies to handle essential algorithms. Even with modern compiler optimizations hand-optimized assembly code is more efficient and many common algorithms involved in DSP calculations are hand-written in order to take full advantage of the architectural optimizations.

DSP can involve linear or nonlinear operations. Nonlinear signal processing is closely related to nonlinear system identification and can be implemented in the time, frequency, and spatio-temporal domains.

5.3.2 APPLICATIONS

Applications of DSP lie in various fields, such as:

- Audio signal processing
- Audio data compression e.g. MP3

- Video data compression
- Computer graphics
- Digital image processing
- Photo manipulation
- Speech processing
- Speech recognition
- Data transmission
- Radar
- Sonar
- Economic forecasting
- Seismology
- Weather forecasting

5.4 Mel Frequency Cepstral Coefficients (MFCC) :

Cepstrum is the information of rate of change in spectral bands. In the conventional analysis of time signals, any periodic component shows up as sharp peaks in the corresponding frequency spectrum (ie, Fourier spectrum. This is obtained by applying a Fourier transform on the time signal).On taking the log of the magnitude of Fourier spectrum, and then again taking the spectrum of this log by a cosine transformation observe a peak wherever there is a periodic element in the original time signal. spectrum of the log of the spectrum of the time signal was named cepstrum.

Pitch is one of the characteristics of a speech signal and is measured as the frequency of the signal. Mel scale is a scale that relates the perceived frequency of a tone to the actual measured frequency. It scales the frequency in order to match more closely what the human ear can hear.

This scale has been derived from sets of experiments on human subjects. The range of human hearing is 20Hz to 20kHz.

Mel frequency can be mathematically represented as:

$$\mathrm{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Any sound generated by humans is determined by the shape of their vocal tract. If this shape can be determined correctly, any sound produced can be accurately represented. The envelope of the time power spectrum of the speech signal is representative of the vocal tract and MFCC , accurately represents this envelope.

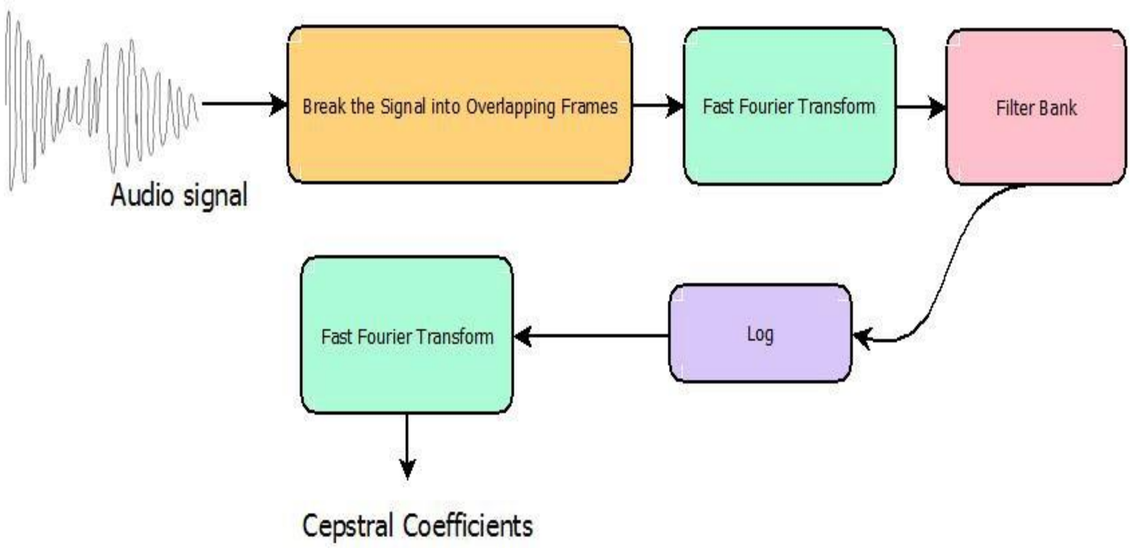

Figure 4 Block diagram representing MFCC

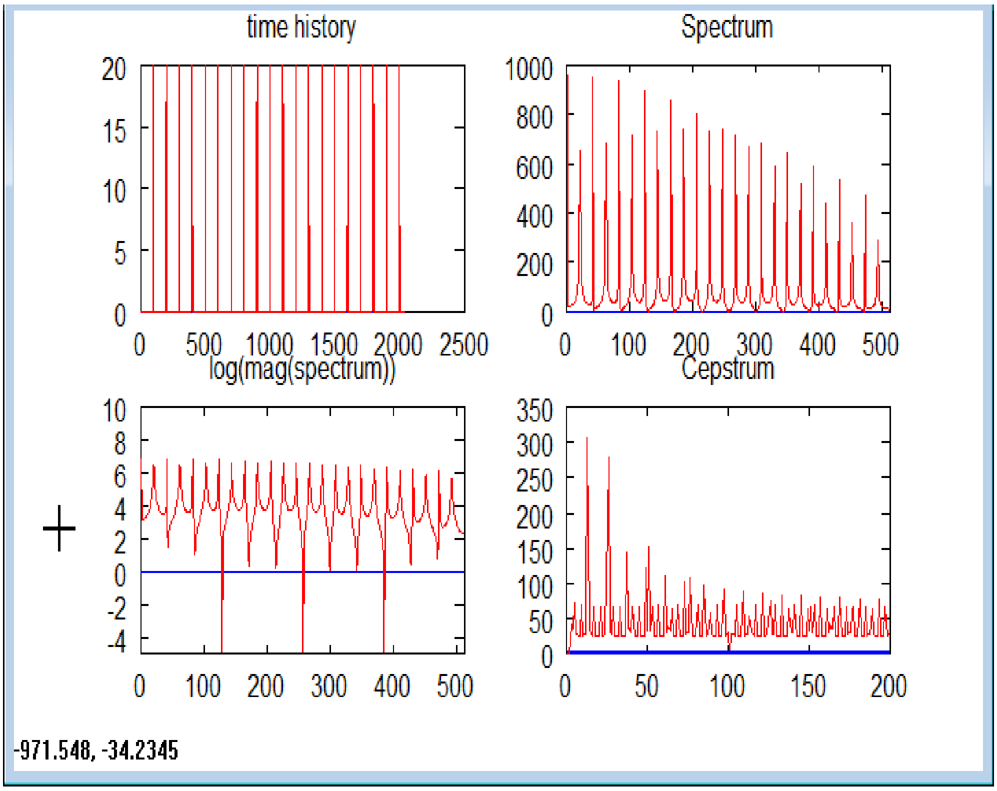

Figure 5 Conversion of a waveform to MFCC

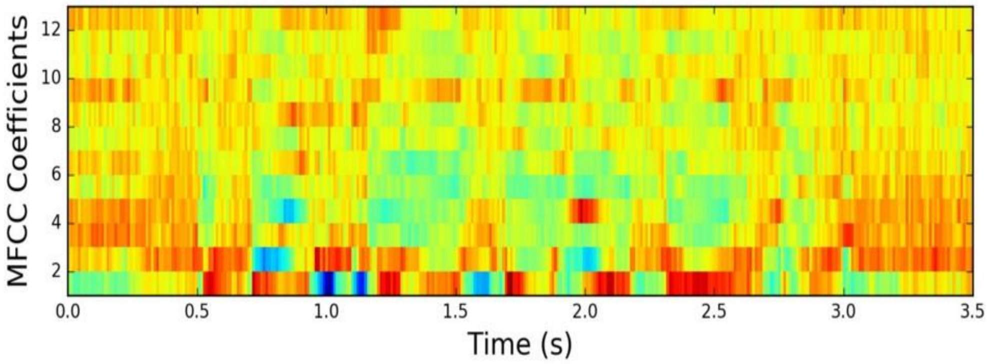

Figure 6 Converted wafeform spectral analysis

5.5 MACHINE LEARNING :

Machine learning is the science of getting computers to act without being explicitly programmed or Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data.

Machine Learning is a subfield of Artificial Intelligence which evolved from Pattern Recognition and Computational Learning theory.

Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

Gathering past data in any form is not always suitable for processing. The better the quality of data, the more suitable it will be for modeling.

In Data Processing, sometimes, the data collected is in the raw form and it needs to be pre-processed.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

Splitting data for Machine Learning:

Training Data: This is the part of data we use to train our model. This is the data which your model actually sees (both input and output) and learn from.

Validation Data: The part of data which is used to do a frequent evaluation of model, fit on training dataset along with improving involved hyperparameters (initially set parameters before the model begins learning). This data plays it’s part when the model is actually training.

Testing Data: Once our model is completely trained, testing data provides the unbiased evaluation. When we feed in the inputs of Testing data, our model will predict some values(without seeing actual output). After prediction, we evaluate our model by comparing it with actual output present in the testing data. This is how we evaluate and see how much our model has learned from the experiences feed in as training data, set at the time of training.

Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

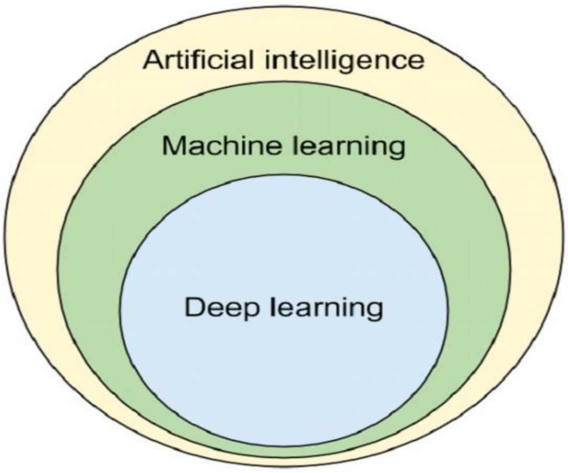

Figure 7 Various layers of Machine Learning

5.5.1 TENSORFLOW :

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances.

while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples. Characterizing the generalization of various learning algorithms is an active topic of current research, especially for deep learning algorithms.

CHAPTER 6

RESULTS

With the collected sample data, digital signal processing (MFCC) was applied to make the data easier to compute, as shown in Figure 8 :

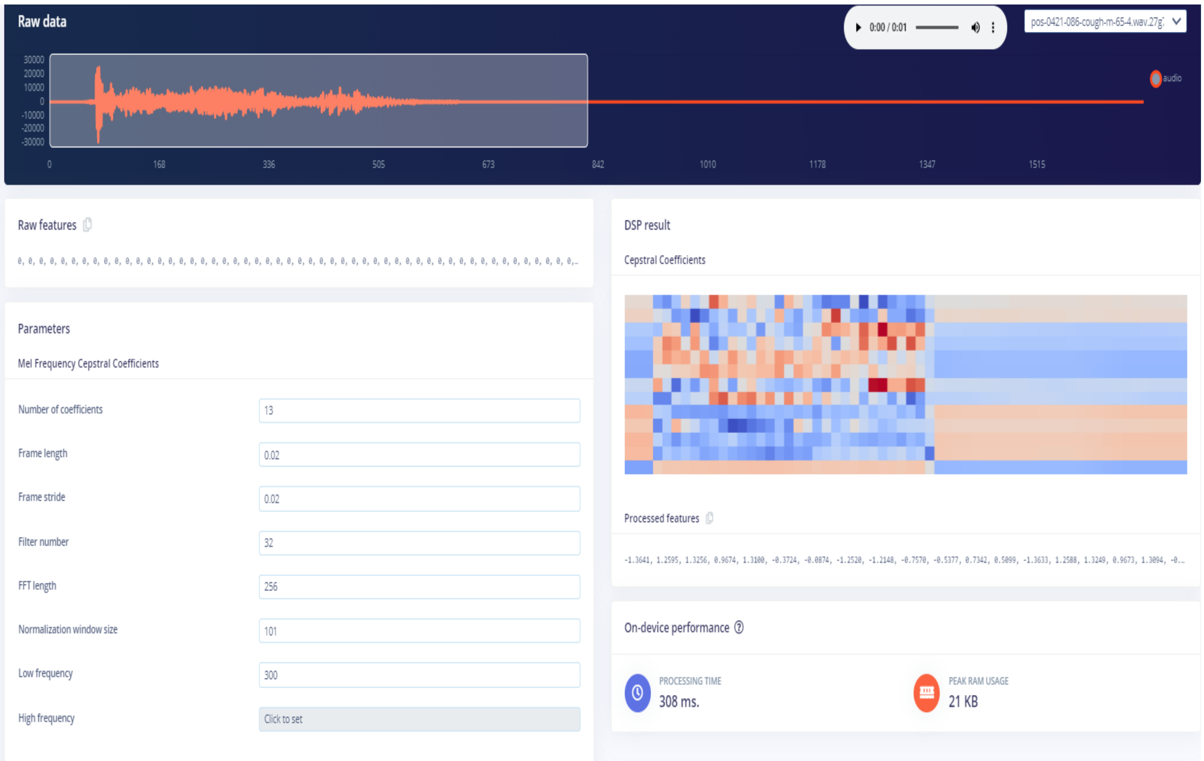

Figure 8 Digital Signal Processing results

Various parameters were set for the clipped audio data, and the DSP results were successfully obtained as shown.

With all the data samples loaded, a feature explorer of the entire training set was obtained (Figure 9) through the Neural Network. It can be seen that the distribution is biased due to insufficient data. This can be corrected easily by taking a larger dataset for training the Neural Network. A target waveform and its position in the feature explorer can also be seen in Figure 9.

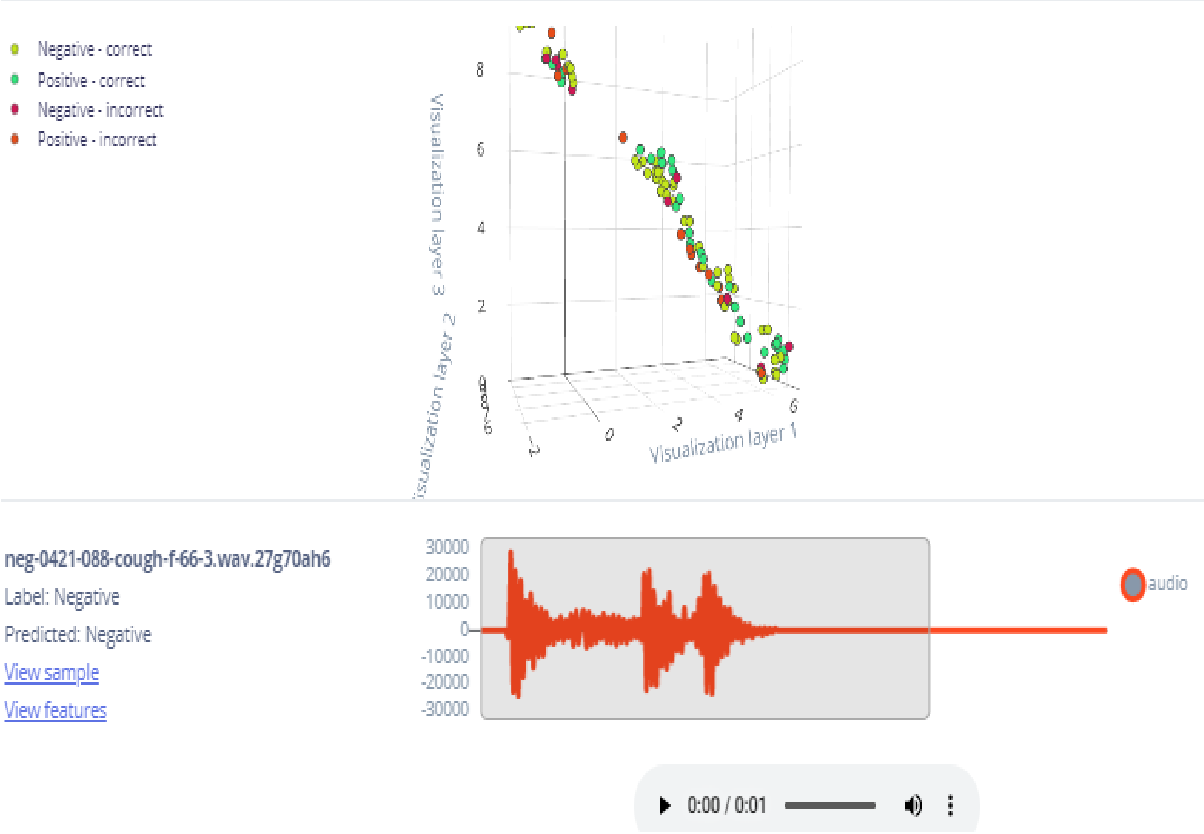

Figure 9 Feature explorer plotted for the trained NN Classifier.

After training the model, a Confusion Matrix (as shown in Figure 10) is obtained. Note that due to the imbalance in the dataset, the matrix is biased. This can be corrected with a larger dataset, as mentioned previously.

Despite this, initial training yields fairly accurate results, as seen in the figure.

Last training performance (validation set)

ACCURACY 73.9%

LOSS 0.61

Confusion matrix (validation set)

	NEGATIVE	POSITIVE
NEGATIVE	90.9%	9.1%
POSITIVE	41.7%	58.3%
F1 SCORE	0.77	0.70

Figure 10 Confusion Matrix

A code snippet used to train the model can be seen in Figure 11.

```
import sys, os, random
import tensorflow as tf
from sklearn.model_selection import train_test_split

import logging
tf.get_logger().setLevel(logging.ERROR)
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

# Set random seeds for repeatable results
RANDOM_SEED = 3
random.seed(RANDOM_SEED)
np.random.seed(RANDOM_SEED)
tf.random.set_seed(RANDOM_SEED)

classes_values = [ "Negative", "Positive" ]
classes = len(classes_values)

Y = tf.keras.utils.to_categorical(Y - 1, classes)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=1)

input_length = X_train[0].shape[0]

train_dataset = tf.data.Dataset.from_tensor_slices((X_train, Y_train))
validation_dataset = tf.data.Dataset.from_tensor_slices((X_test, Y_test))

def get_reshape_function(reshape_to):
    def reshape(image, label):
        return tf.reshape(image, reshape_to), label
    return reshape

callbacks = []

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D, Flatten, Reshape, MaxPooling1D, MaxPooling2D,
from tensorflow.keras.optimizers import Adam

# model architecture
model = Sequential()
model.add(Reshape((int(input_length / 13), 13), input_shape=(input_length, )))
model.add(Conv1D(8, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Conv1D(16, kernel_size=3, activation='relu', padding='same'))
model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(classes, activation='softmax', name='y_pred'))

# this controls the learning rate
opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
# this controls the batch size, or you can manipulate the tf.data.Dataset objects yourself
BATCH_SIZE = 32
train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)
validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)

# train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
model.fit(train_dataset, epochs=100, validation_data=validation_dataset, verbose=2, callbacks=callbacks)
```

Figure 11 Code snippet for the ML Model

Finally, the model is deployed (based on the web based API of Edge Implse), and can live classify cough samples with a fair amount of accuracy, as seen in Figure 12.

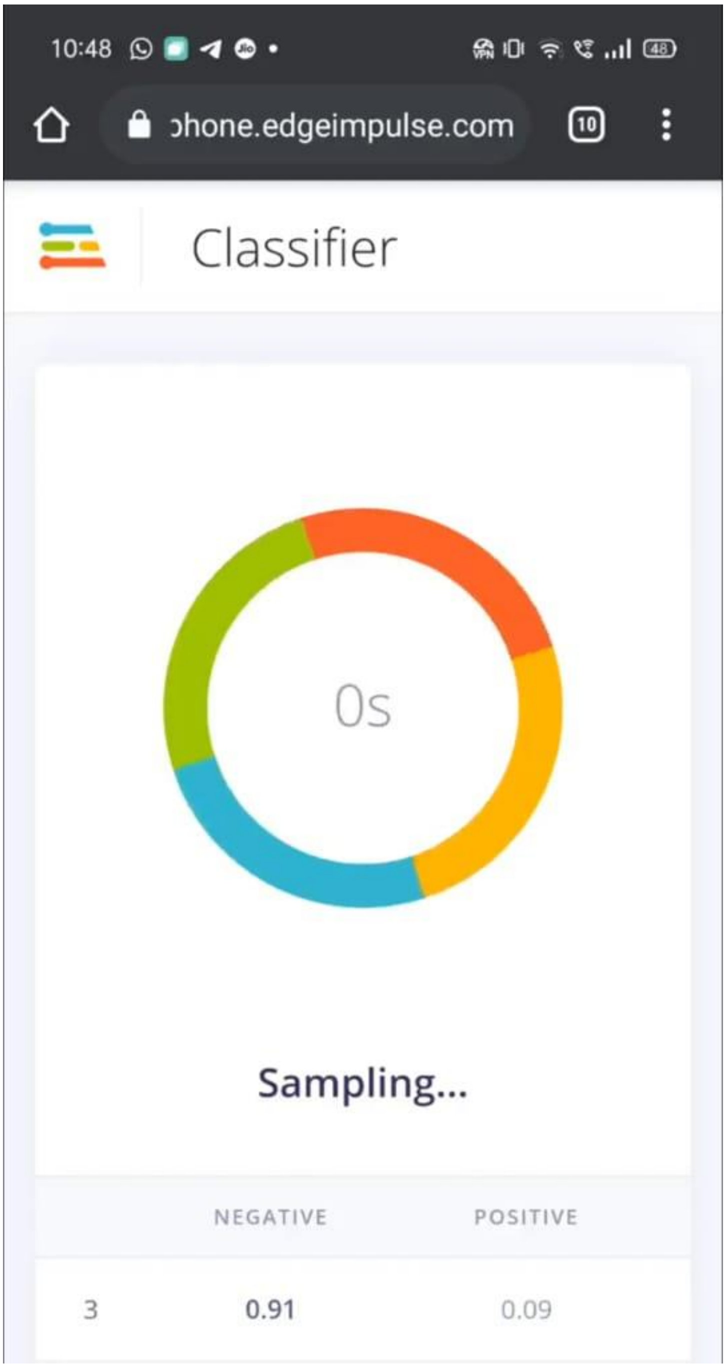

Figure 12 Live Classification of cough sample; data collected through inbuilt microphone of the phone.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLSION

The proposed project shows promising applications in the field of COVID-19 detection. It shows fairly accurate results as a preliminary test to detect COVID-19, and provides a quick and non-invasive way to get fast results. And with a diversified dataset, more and more range of the human voice (or cough sound) can be captured, making the model more accurate.

The proposed project, albeit provides fairly accurate data, can be improved further by training the Machine Learning model with a balanced dataset. This requires access to COVID-19 positive patients through medically supervised and safe methods and get recorded voice samples of cough sounds.

Flexibility of the project proposed, and the open source nature of the tools used also ensures that the code used to build the ML model can be streamlined and improved upon.

7.2 FUTURE SCOPE

Although the methodology and algorithms used in this project serve well in achieving the goal, it may be used for a wide variety of applications. Machine Learning models like the one demonstrated here can be valuable in fault analysis, pattern recognition (regression models), low power applications in standalone devices, etc. There are various sectors that can use these in real life applications, such as:

- Aerospace and Defence
- Manufacturing
- Medicine
- Traffic detection
- Mathematical modelling

CHAPTER 8

REFERENCES

[1] Mrs. Mucahid Barstugan, Umut Ozkaya, Saban Ozturk, Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods, named by WHO, March 2020.

[2] Davide Brinati, Andrea Campagner, Davide Ferrari, Massimo Locatelli3, Giuseppe Banfi, Federico Cabitza, Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study, Journal of Medical Systems, July 2020.

[3] Thomas Grill, Jan Schlüter, Two Convolutional Neural Networks for Bird Detection in Audio Signals, 25th European Signal Processing Conference (EUSIPCO), September 2017.

[4] Poonam Mahana, Gurbhej Singh, Comparative Analysis of Machine Learning Algorithms for Audio Signals Classification, IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015.

[5] Vasileios Bountourakis, Lazaros Vrysis, George Papanikolaou, Machine Learning Algorithms for Environmental Sound Recognition: Towards Soundscape Semantics, Audio Mostly 2015 on Interaction with Sound, October 2015.

[6] Feng Rong, Audio Classification Method Based on Machine Learning, International Conference on Intelligent Transportation, Big Data and Smart City, December 2016.

[7] Hareesh Bahuleyan , Music Genre Classification using Machine Learning Techniques, April 2018.

[8] Abigail Copiaco, Christian Ritz, Stefano Fasciani , Nidhal Abdulaziz, Scalogram Neural Network Activations with Machine Learning for Domestic Multi-channel Audio Classification, 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), decemper 2019.

[9] Mursal Dawodi, Jawid Ahamd Baktash, Tomohisa Wada, Najwa Alam, Mohammad zarif Joya, Dari Speech Classification Using Deep Convolution Neural Network, 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), September 2020.

[10] Prabira Kumar Sethy, Santi Kumari Behera, Pradyumna Kumar Ratha, Preesat Biswas, Detection of Coronavirus Disease (COVID-19) Based on Deep Features and Support Vector Machine, International Journal of Mathematical, Engineering and Management Sciences, April 2020.

[11] Irina Valeryevna Pustokhina, Denis AlexandrovichPustokhin, K.Shankar, A novel

machine learning–based detection and diagnosis model for coronavirus disease (COVID-19) using discrete wavelet transform with rough neural network, Data Science For Covid-19,2021 .

[12] Chenglong Liu, Xiaoyang Wang, Chenbin Liu, Qingfeng Sun & Wenxian Peng, Differentiating novel coronavirus pneumonia from general pneumonia based on machine learning, august 2020.

[13] Ali Narin, Ceren Kaya & Ziynet Pamuk, Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, pattern application and analysis, may 2021.

[14] Davide Brinati, Andrea Campagner, Davide Ferrari, Massimo Locatelli, Giuseppe Banfi & Federico Cabitza, Detection of COVID-19 Infection from Routine Blood Exams with Machine Learning: A Feasibility Study, journal of medical systems, July 2020.