Classification of Respiratory Diseases and COVID-19 from Respiratory and Cough Sound Using Deep Learning Techniques

Prepared By

Md. Mubtasim Ahasan
18101195
Mohammad Fahim
18101487
Himadri Mazumder
18101041
Nur E Fatema
18101340
Sheikh Mustafizur Rahman
18101610

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis that we have presented is our unique individual research that we completed during studying at Brac University.

2. Except as properly acknowledged through proper and correct citation, the thesis does not contain information already published or written by a third party.

3. The thesis does not incorporate any content that has been approved or presented for another university or even other institution's diploma or degree.

4. All major sources of assistance have indeed been acknowledged.

Student's Full Name & Signature:

Mubtasim

Md. Mubtasim Ahasan
18101195

[illegible]

Mohammad Fahim
18101487

[illegible]

Himadri Mazumder
18101041

Fatema

Nur E Fatema
18101340

[illegible]

Sheikh Mustafizur Rahman
18101610

Approval

The thesis project titled "Classification of Respiratory Diseases and COVID-19 from Respiratory and Cough Sound Using Deep Learning Techniques" submitted by

1. Md. Mubtasim Ahasan(18101195)
2. Mohammad Fahim(18101487)
3. Himadri Mazumder(18101041)
4. Nur E Fatema(18101340)
5. Sheikh Mustafizur Rahman(18101610)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on January, 2022.

Examining Committee:

Supervisor: (Member)

[illegible]

Jannatun Noor
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department: (Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Infectious and non-infectious respiratory diseases are among the major reasons for deaths, financial and social crises around the world. However, medical personnel still find it very difficult to detect the diseases using conventional methods to combat this global crisis. We propose a respiratory disease identification method from respiratory auscultation sounds and COVID-19 infected and healthy patients from cough sound recordings. Our experiments demonstrate that artificial intelligence can be utilized as an alternative method to detect respiratory illnesses. We extract image representations of audio features such as Mel-frequency Cepstral Coefficients (MFCCs) and Mel-Spectrogram from each audio recording and use convolutional neural network models for our experiments. Also, we compare the two audio features and ten different convolutional neural network architecture's performance on disease classification. We conduct experiments with various model training procedures' such as transfer learning and 1cycle policy, and balanced mini-batch training. In our experiment, we classified respiratory diseases with 94.57 percent accuracy and 0.93 ROC-AUC scores and COVID-19 affected and healthy patients' cough recordings with 85.96 percent accuracy and 0.84 ROC-AUC scores.

Keywords: Deep Learning; Machine Learning; Respiratory Disease; Cough Sound; COVID-19; Mel-Spectrogram; MFCC; CNN

Acknowledgement

First and foremost, we express gratitude to the Almighty for allowing us to finish our thesis without any major setbacks. We would like to convey our profound gratitude to our esteemed thesis supervisor Ms. Jannatun Noor for her valuable time and guidance. Finally, without our parents' unwavering support, we may not be able to achieve our goals. We are currently on the verge of graduating thanks to their generous support and prayers.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	Artificial Intelligence
$ARDS$	Acute Respiratory Diseases Syndrome
CNN	Convolutional Neural Network
$COPD$	Chronic obstructive pulmonary disease
FFT	Fast Fourier Transform
$ICBHI$	International Conference on Biomedical and Health Informatics
$LSTM$	Long Short-Term Memory
$MFCC$	Mel Frequency Cepstral Coefficients
MLP	Multilayer Perceptron
$NAAT$	Nucleic Acid Amplification Tests
$NAPS$	National Asthma Prevalence Study
RLS	Recursive Least Squares
RNA	Ribonucleic acid
$ROC - AUC$	Area Under the Receiver Operating Characteristic Curve
$RT - PCR$	Reverse Transcription Polymerase Chain Reaction
$STFT$	Short-Time Fourier Transform
SVM	Support Vector Machine
$URTI$	Upper Respiratory Tract Infection
WHO	World Health Organization

Chapter 1

Introduction

The global prevalence of respiratory diseases, which affect both children and adults is rising rapidly. According to the World Health Organization (WHO), respiratory disorders are among the top reasons for mortality and disability worldwide [17]. Increased air pollution, industrialization, and rising living standards are wreaking havoc on the environment in cities worldwide. Consequences of which are Chronic respiratory diseases (CRDs) causing lung disorders that affect the airways and are major causes of mortality worldwide. Furthermore, some major life-threatening lung diseases are asthma, chronic bronchitis, acute respiratory distress syndrome (ARDS), pneumonia, chronic obstructive pulmonary disease (COPD), lung cancer, etc. Moreover, various factors, including heavy air pollution exposure, direct or indirect tobacco smoke exposure, and most commonly, virus exposure, such as the Coronavirus or the influenza virus, can aggravate respiratory conditions [2].

Countries concentrated with heavy populations have been battling air pollution for a long time. Moreover, around 150 million new cases are reported every year in the bulletin of The World Health Organization (WHO). Approximately 11-20 million (7-13%) of these instances are severe enough to require hospitalization. Mostly around 95% of the cases are seen in developing nations [59]. Pneumonia is responsible for about 28% of all deaths in children below five years old, and around 50,000 children die from pneumonia annually in Bangladesh [11]. Worldwide, pneumonia affects around one child in every 71 per year, with the highest rates of 2,500 cases in South Asia and 1,620 cases in West and Central Africa per 100,000 children [64]. According to WHO, asthma is a frequent condition among children that affects 235 million people worldwide [39]. In Bangladesh, about 5.2% or 7 million people have asthma, stated by the National Asthma Prevalence Study (NAPS). It also notes that more than 90% of the people cannot afford the treatment cost and do not take proper treatment [6].

The recent ongoing Coronavirus, an airborne disease, also known as COVID-19, caused by SARS-COV-2, is taking a deadly toll on the mass population. New progressive and fast-changing viral strains have lately been discovered worldwide, making herd immunity a thing soon. The latest variant Omicron is regarded as the most harmful variant compared to the previous ones as its full significance of the mutations is still unknown [56]. In no time, COVID-19 created massive damage in most countries as it affects the upper or lower respiratory tract leading to Acute

Respiratory Distress Syndrome (ARDS), which results in dangerously low levels of oxygen in the blood.

1.1 Problem Statement

Since the signs of respiratory infections are often quite similar, making a quick and appropriate diagnosis is critical for treatment when conducted by an incompetent or unpracticed individual, leading to misdiagnosis [5]. Moreover, the traditional method to detect a respiratory disease has several setbacks, as it needs intense attention and experienced doctors to comprehend the respiratory auscultation sound. Aside from the characteristics of the virus and the countries it has infected, this difficult hour calls for a cost-effective and efficient virus diagnosing method.

The irony of the situation is that, despite the number of deaths every minute due to incurable respiratory diseases, its detection techniques have not evolved much. Detecting the ultimate respiratory disease is a difficult task altogether as the doctors also go through some problems while performing the basic tests. A stethoscope is still the primary instrument for most doctors to hear the lung sound. Sometimes, it becomes difficult while determining the problem as it requires intense attention and experience. Using traditional handheld stethoscopes, even competent doctors may struggle to reach a high level of agreement on respiratory symptoms. Moreover, continuous lung sound monitoring over an extended period is nearly impossible [49]. In this case, an electronic stethoscope combined with an artificial intelligence system can be used to overcome the limitations of classical auscultation, resulting in a more efficient and reliable type of diagnosis via automated diagnostics [46].

The standard approach of detecting COVID-19 is not without flaws. WHO suggests employing Nucleic Acid Amplification Tests (NAAT), such as real-time Reverse Transcription Polymerase Chain Reaction (RT-PCR), to detect COVID-19 patients based on their unique RNA sequence. Although it is a solid technique, the testing arrangement is both time-consuming (2-48 hours) and costly, making it unsustainable for the large number of cases that arise every day. Also, it can expose more people to COVID-19 because the test needs an in-person visit. Furthermore, to perform tests properly, the medical personnel may have to violate the protection protocols [17], limiting the usage of auscultation on a patient with other respiratory disease.

Recognizing COVID-19 symptoms is difficult since infected patients can be asymptomatic [58]. However, because of its effect on the respiratory system, the coronavirus produces particular auditory characteristics that differ from those associated with other respiratory disorders and healthy people. Coughing and vocal sounds can also provide information about lung health that can be used to diagnose disease. Although the variances in coughing noises are minor and invisible to the human ear, their features are iterative and distinct for each patient [58]. As a result, computerized respiratory sound analysis is required to overcome the above limits [48]. Machine learning algorithms have also shown promise in differentiating cough sounds to diagnose respiratory illnesses such as pneumonia, asthma, and pertussis [15][8].

1.2 Motivation

Previous research has shown that deep learning models can detect cardiac disease [52], classify heart sounds [53], and identify Parkinson's disease using audio recording data [42]. Moreover, researchers from all around the world have been attempting to create the most effective techniques of detecting and controlling the coronavirus as early as possible. Similarly, AI researchers are working on developing an AI technology-based approach to identify COVID-19. Furthermore, the ongoing COVID-19 outbreak has opened the door to new deep learning-based research due to the availability of COVID-19 affected patients' data, such as the Coswara and COUGHVID datasets.

The goal of our research is to investigate the possibility of detecting respiratory disease and COVID-19 from respiratory and cough sound using deep learning techniques, as well as to contribute to AI-based research to develop an alternative method for respiratory-based disease detection that can be used as a primary warning tool for patients.

1.3 Research Objective

For our research, we utilize the publicly accessible lung sound data from the ICBHI 2017 challenge [21], as well as crowdsourced cough sound data from the COSWARA project [50] and the COUGHVID dataset [63].

We divide our classification task into two major tasks:

- Firstly, we use respiratory sound data to detect respiratory diseases. We utilize the ICBHI dataset for this task and further divide the task into two sub-tasks. The first sub-task is to classify the ICBHI dataset into six classes (COPD, Pneumonia, Healthy, URTI, Bronchiectasis, and Bronchiolitis). The second sub-task is to classify respiratory diseases into three classes based on disease severity (Chronic, Non-Chronic, and Healthy).
- Secondly, we use crowdsourced cough sound data to classify COVID-19 infected and healthy patients' cough sounds. We use the Coswara project and COUGVID crowdsourced cough sounds dataset. Since the two datasets contain two different audio recordings with different cough sound types, average cough count, and duration, we perform this separately for the two datasets. The first sub-task is to classify the cough sound using the Coswara data, whereas the second sub-task is to classify the cough sound using the COUGHVID dataset.

We conduct the following experiments on both of our tasks:

- To begin, we examine the performance of two audio characteristics, MFCC and Mel-Spectrogram, on disease classification.

- Next we compare ten Convolutional Neural Networks (ResNet-18, ResNet-34, ResNet-50, ResNet-101, VGG16, VGG19, AlexNet, SqueezeNet, DenseNet-121, XResNet-50) architectures performance on disease classification.
- Furthermore, we compare and contrast the effects of two distinct training strategies, transfer learning and 1cycle policy, on model performance.
- Lastly, we check model generalization ability using Balanced Mini-Batch Training with Mel-Spectrogram augmentation.

1.4 Contribution

The fundamental contribution of this study is that it presents a deep learning-based respiratory disease detection and COVID-19 patients' cough classification approach that can be utilized as a disease screening and warning system. In our research, we acquire a ROC-AUC of 0.84 in COVID-19 cough classification, which is comparable to several previous research studies. We also show that when using an imbalanced dataset in a respiratory disease classification task, a balanced mini-batch training strategy with feature augmentation can improve recall scores up to 5%. Moreover, we present a comparison of various audio features, CNN architectures, and training methodologies.

Chapter 2

Literature Review

2.1 Related Works

Previous research showed machine learning models to detect the crackle and wheeze sound from different lung sound recordings. The research work [30] presented a model for automated detection of crackle sounds using an electret subminiature and a mobile application inside a smartphone device. They developed a time-varying autoregressive modeling and Recursive Least Squares (RLS) algorithm and detected crackle sound with accuracy ranging from 91.2% to 94%. Another work [9] presented a model for detecting wheeze sounds from the lung sounds using a mobile application and an electronic stethoscope to collect lung sounds. They have collected data from 38 patients from different hospitals. Moreover, they have implemented two machine learning algorithms to extract features and identify abnormal sounds. Lastly, they trained a support vector machine to classify and achieve an accuracy of 86% to categorize the sound file either as wheeze or normal. A model for classifying lung sounds into wheeze and crackle using a deep learning algorithm was presented in the paper [12]. Their recorded lung sounds are 30 seconds long and taken from 11 different back, chest, and trachea locations using a mobile phone and a custom-made electronic stethoscope. Moreover, they used Short-Time Fourier Transform (STFT) to convert each sound to a spectrogram and used a denoising autoencoder for feature selection. Finally, by training two support vector machine models, one to detect crackles and another to identify wheezes, they achieved ROC curves with AUCs of 0.74 for the crackle and 0.86 for wheeze. The study [20] performed an automatic analysis of respiratory sound data from 60 patients by collecting the data using custom-made prototype equipment. Moreover, they used the Gaussian Mixture Model (GMM) with a two-stage pipeline to categorize lung sounds into wheeze, crackle, and normal and achieved an accuracy of 98.4%. The paper [35] examined the numerous types of unusual respiratory sounds like wheeze and crackle created a new classification algorithm, namely LungBRN. They have proposed STFT and wavelet analysis as the two essential input characteristics for the classifier. Furthermore, they suggested a network design that uses a bilinear biResNet model to distinguish between different forms of unexpected lung sounds, including crackle and wheeze. Moreover, the research work [25] presented a breathing analysis for detecting irregular patterns in respiratory cycles. To detect adventitious sounds present in breath sounds due to various disorders, they used a dataset collected from hospitals under the guidance of pulmonologists and physicians. In addition, they used wavelet denoising techniques to remove background noises from the respiratory sound and used Support Vector Machine (SVM) classifiers to classify the breath sound into normal and abnormal. They achieved 75% accuracy in detecting adventitious sounds from a complete respiratory sound cycle.

Several research works focused on detecting specific respiratory diseases, including paper [13], which presented a model for classifying COPD and Asthma. They used a two-stage logistic regression model to first differentiate between patients with COPD or Asthma from the ordinary people and then separated the patients with Asthma or COPD, achieving a ROC-AUC score of 0.97. The paper [4] presented a lung disease-diagnosing tool to record breath sound by using a highly sensitive microphone and analyzed the lung sound recording to check if the patient's lung is in good shape or not. Primarily, the sound is split and processed into upper respiratory and lung sounds and then compared to a database to determine the patient's illness.

Some research work has aimed at audio signal processing techniques for pre-processing the recorded sound to extract features from audio data. The research [34] selected Discrete Wavelet Transform coefficients (DWT), Mel-Frequency Spectral Coefficients (MFCC), and Time Domain Features to extract features. The algorithm they used for classification is a RUSBoost algorithm and a Decision Tree as a base classifier. To evaluate their model, they used an online testing dataset and achieved an accuracy of 87%. According to paper [18], signal processing algorithms were used to classify usual and unusual breath sounds using electronic stethoscopes. They retrieved Short-time Fourier transform-based features and used decomposition of singular values to reduce the number of features. The classifier was built on k-NN using Euclidean distance as a metric, and the suggested approach has an accuracy, specificity, and sensitivity of 91.55%, 92.20%, and 90.9%, respectively.

Researchers are becoming increasingly interested in innovating machine learning approaches as a means of solving problems for specific purposes. For instance, this paper [51] presented a framework integrating a random forest classifier with the Empirical Mode Decomposition approach for a multiple class classification task of recognizing respiratory diseases (COPD, Bronchiolitis, Bronchiectasis, URTI, and Pneumonia). Moreover, they evaluated 14 distinct lung sound variations and discovered that segmentation is crucial in categorizing different respiratory diseases. Their trained classifier obtained an accuracy of 88%, a precision of 91%, a recall of 87%, a specificity of 91%, and an F1-score of 81%. The study [12] proposed a classification model using a machine learning approach and extrapolated the results to decide on the patient level. Their strategy is separated into two parts: micro-level and macro-level. The micro-level is accountable for respiratory cycle classification, while the macro-level is responsible for extrapolating the recognition accuracy to account for patient categorization. They used a boosted decision tree model to identify each respiratory cycle based on whether unusual sounds detect at the micro-level. Their macro-level approach achieved accuracy up to 85%. Paper [49] claimed that though respiratory diseases are universal, their detection is challenging for the experts. The researchers used cough audio features to categorize the conditions in patients. They created and trained a "CoughGAN" generative adversarial network to replicate raw audio recordings of coughs. Their Random Forest and Support

Vector Machine models correctly identified participants' statuses between healthy and three common respiratory diseases, with a test accuracy of 76% and an F1 score of 83%. A recent work [47] proposed a model combining the CNN and Attention mechanism to classify the audio data to detect abnormal breath sounds from normal respiratory sounds. They implemented the CNN-Attention classification model, extracted the Mel Frequency Cepstral Coefficient (MFCC) feature from the data, and fed it to the model. Moreover, to prevent various dimensions of input vectors, they reduce the retrieved features into a single dimension.

Since 2019, more than 5 million people have perished from coronavirus disease, and the number is still counting. There will be many more deaths before a better way to cure or eliminate this disease is discovered. Researchers have been trying to find ways to eradicate this deadly disease from the world in recent times. In the paper [50], researchers have gathered nine different forms of sound data, such as deep and shallow breathing, heavy and shallow coughing, sustained vowel phonation of the letters a,e, and o, and normal and fast-paced digit counting from one to twenty. They have also collected data of 941 people, including their health status and pre-existing medical conditions. Moreover, they anonymized the data during storage and used 13 annotators to annotate it. The collection contains 6507 clean audio files, 1117 noisy audio files, and the remaining significantly degraded audio files, all of which correspond to respiratory sound samples from 941 persons. Furthermore, they have also used a random forest classifier to classify their dataset's nine different sounds categories using a 28-D feature, and the accuracy on test data was 66.74%. To enhance the detection of COVID-19, in the article [43], they gathered data from an extensive publicly supported collection of respiratory sounds using a web-based interface and an Android app. Specifically, their dataset is made out of 6613 unique participants. Among them, 235 announced having tested positive for COVID-19. Their SVM models accomplished an AUC score above 0.8 using coughs and breathing sounds using handcraft features and data augmentation. Moreover, they have utilized VGGish to extricate sound elements automatically and examined two unique features: handcrafted and transfer learning.

In the work [44], they demonstrated that cough audio samples gathered from people worldwide via cellphones can be used to build an artificial intelligence-based approach that reliably predicts COVID-19 with a ROC-AUC score of 77.1 percent. They trained a deep neural network using the publicly available cough sound datasets COSWARA [50] and COUGHVID [63]. The paper [58] has used the MFCC features from the sound recordings as an input individually into LSTM Recurrent Neural Network, CNN, and Multilayer Perceptron (MLP) Network, respectively. Furthermore, they achieved an AUC score of 70.69 using Random Forest (baseline), 56.57 using MLP, 70.67 using LSTM, 72.33 using CNN, and 87.07 using CNN with Data Augmentation. The research [54] showed that a deep neural network could detect symptomatic and asymptomatic COVID-19 cases using breath and cough audio recordings. However, their dataset includes 517 crowdsourced breathing and coughing audio recordings from 355 individuals, and they achieved an AUC-ROC of 0.846 using this dataset. This study also used audio samples to generate spectrograms to create an end-to-end CNN design. Finally, the paper [57] has chosen 5,240 samples from 2,478 people from the collected dataset and divided them into separate participant-independent sets for model training and validation. Using a CNN model, they obtained an AUC-ROC of 0.71 using features from breathing, coughing, and voice data as predictors and predicted COVID-19.

Chapter 3

Data Description

The coronavirus disease (COVID-19) pandemic is unprecedented in most people's lifetimes. Due to the ongoing pandemic, the entire planet was brought to a standstill. For this reason, physically collecting the respiratory sound was not possible at that time. Among the various open-source datasets, we chose the ICBHI dataset for respiratory sound classification and COSWARA and COUGHVID dataset for COVID-19 classification. Moreover, before the data was ready to be trained, we applied multiple pre-processing techniques discussed in section 3.2.

3.1 Dataset

We used three separate datasets for our study, one lung sound dataset collected with an electronic stethoscope for respiratory disease classification and two crowdsourced cough sound data for covid-19 cough detection.

3.1.1 ICBHI Scientific Challenge Dataset

The first dataset we utilize in this study was revealed as part of an International Conference on Biomedical and Health Informatics (ICBHI) scientific challenge. The ICBHI challenge dataset was published in 2017 and used by several academics. Two study teams from Greece and Portugal accumulated the audio files in the ICBHI dataset. The two research teams took approximately 920 audio samples from 126 patients of different ages and analyzed them. There are 6898 respiratory cycles in the dataset, with 1864 containing crackles, 886 containing wheezes, and 506 comprising both wheezes and crackles. The average duration of these recordings is 21.49 seconds, and their median is 20 seconds. Adventitious sounds are not indicated in 259 of these annotated tracks. There is a text file against every audio file in the dataset, which includes information about the start and end time of a respiratory cycle and the presence or absence of wheeze and crackle sounds. Moreover, the diagnosis in the dataset include Healthy, Bronchiolitis, Pneumonia, Bronchiectasis, Lower Respiratory Tract Infection (LRTI), COPD, Asthma, and Upper Respiratory Tract Infection (URTI). The dataset contains people with COPD around 50.79%, which is the highest. On the other hand, healthy people with Upper Respiratory Tract Infection (URTI) are 20.63% and 11.11% respectively [21].

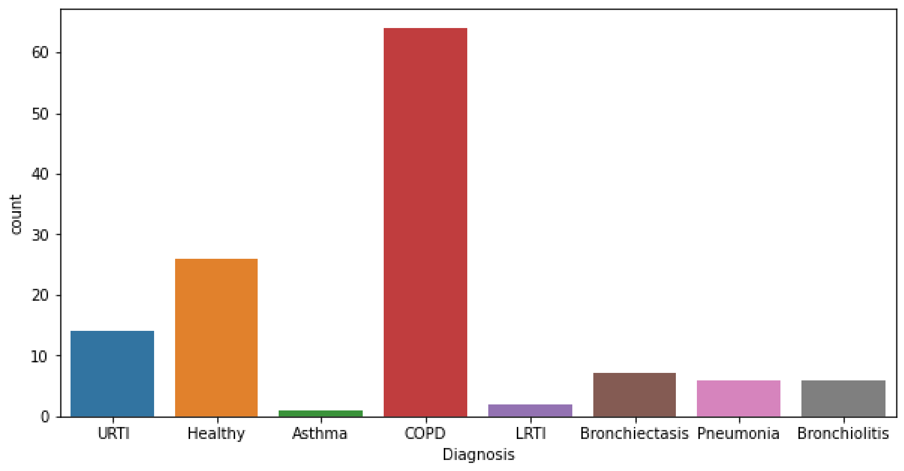

Figure 3.1: Number of subjects against their medical condition in ICBHI dataset

The ICBHI data is collected from two universities, Aristotle University of Thessaloniki (AUTH) and the University of Aveiro. The Researchers of Aristotle University of Thessaloniki (AUTH) recorded samples from six distinct places from the chest. The School of Health Sciences, University of Aveiro (ESSUA), on the other hand, recorded noises from the left and right anterior, as well as the trachea, lateral chest, and posterior. Normal and abnormal respiratory sounds like a wheeze, crackles were identified and annotated by expert physicians [21].

The majority of the spectrum power of a healthy vesicular breathing lung sound falls between 60 to 600 Hz, and the frequency ranges up to 1,000 Hz. On the other hand, wheeze and crackle have a frequency range of 1,000 Hz to 2,500 Hz. Because lung sound is weak and vulnerable to external disturbances like heartbeat and music, a 5th order Butterworth bandpass filter is utilized to keep the frequency of interest between 100 Hz and 2,000 Hz [35].

3.1.2 COSWARA

The second dataset we utilized is the Coswara dataset [50], a curated database of respiratory sounds, coughs, and voices which are categorized by gender, age, country, state, and health status. The audio samples of this dataset were obtained through global crowdsourcing using an online application where a user engages with the application for an average of 5-7 minutes. For each of the sound categories, each respondent contributes nine audio tracks. All of the audio samples were manually curated and recorded at a 48 kHz sampling rate. The annotator might listen to each sound recording and answer several questions by utilizing a web interface. These questions helped confirm the category label and the audio file's quality. For each audio track, the annotator was also given the option of adding any additional comments. The dataset contains 6507 clean audio files, 1117 noisy audio files, and the remaining significantly degraded audio files, all of which correspond to respiratory sound samples. Deep and shallow breathing, heavy and shallow coughing, sustained vowel phonation of the letters a,e, and o, and normal and fast-paced digit counting from one to twenty were among the nine types of sound data collected. During storage, they anonymized the data and utilized 13 annotators to annotate it.

3.1.3 COUGHVID

The third dataset we utilized in this paper is the COUGHVID dataset [63], one of the largest publicly available COVID-19 related cough sound datasets. This dataset has recordings of over 25,000 crowdsourced cough audios covering a broad range of participants. The COUGHVID crowdsourcing dataset is one of the massive cough datasets categorized by professionals, with 1,155 people claiming to have COVID-19 from around the world. In addition to making the majority of their cough corpus available to users, they have trained a cough identification machine learning model to extract non-cough recordings from the dataset. This automated cough identification tool enables developers to construct durable programs that automatically exclude non-cough noises from audios. An extra layer of validation was performed, in which four expert specialists evaluated a portion of the dataset to identify which crowdsourced samples are likely to come from COVID-19 patients. The COUGHVID dataset contains over 2,800 coughs labeled by experts, each with its severity level, diagnosis, and whether or not audible health problems such as nasal congestion, wheezing, dyspnea are present. Lastly, they validated that COVID-19 samples were collected in areas where the virus was active during the recording time, and they examined the quality of cough recordings.

3.2 Data Pre-processing

For our experiments, we preprocessed the three datasets separately which are discussed in this section.

3.2.1 Respiratory Disease Classification

The ICBHI dataset contains 920 audio recordings of eight different respiratory disease labels: COPD, Pneumonia, Healthy, URTI, Bronchiectasis, Bronchiolitis, Asthma, and LRTI. For respiratory disease classification, we first preprocess the dataset with two different approaches. In the first method, we exclude the Asthma and LRTI data due to insufficient data on Asthma and LRTI and perform a six-class classification with the rest of the data.

Disease	**Count**
COPD	793
Pneumonia	37
Healthy	35
URTI	23
Bronchiectasis	16
Bronchiolitis	13
LRTI (×)	2
Asthma (×)	1

Table 3.1: Disease distribution of audio samples from ICBHI datasets. Lowest two samples (marked with ×) were excluded from 6 class classification.

In the second method, we divide the dataset's eight diseases into three categories based on disease severity: Chronic, Non-Chronic, and Healthy. The Chronic category contains COPD, Asthma, and Bronchiectasis diseases, and The Non-Chronic category contains URTI, LRTI, Bronchiolitis, and Pneumonia diseases. We divided

Class	**Count**
Chronic	810
Non-Chronic	75
Healthy	35

Table 3.2: Disease distribution for 3 class classification

the dataset into three parts for this task: 60% for the training set, 20% for the validation set, and 20% for the test set.

3.2.2 COVID-19 Cough Classification

We pre-process the Coswara and COUGHVID datasets individually for our COVID-19 positive and negative cough classification task because the data formats and cough sounds differ significantly. In the Coswara data gathering method, participants were instructed to cough atleast three times. In contrast, the number of times participants should cough was not explicitly stated in the COUGHVID data collection process. Also, the number of coughs in a single audio file in the COUGHVID dataset is inconsistent, and some recordings contain only one cough sound. For this reason, we choose to pre-process and evaluate our models on these data separately, which also allows us to identify which type of cough recording is better for AI-based classification.

Coswara

The Coswara dataset contains seven different covid status information depicted on figure 3.2. And for one patient id, one status label is assigned. To separate the dataset into two classes, we label the healthy and no respiratory illness exposed status as COVID-19 Negative. And label the positive mild, positive asymptomatic, and positive moderate statues as COVID-19 Positive labels. After preprocessing the

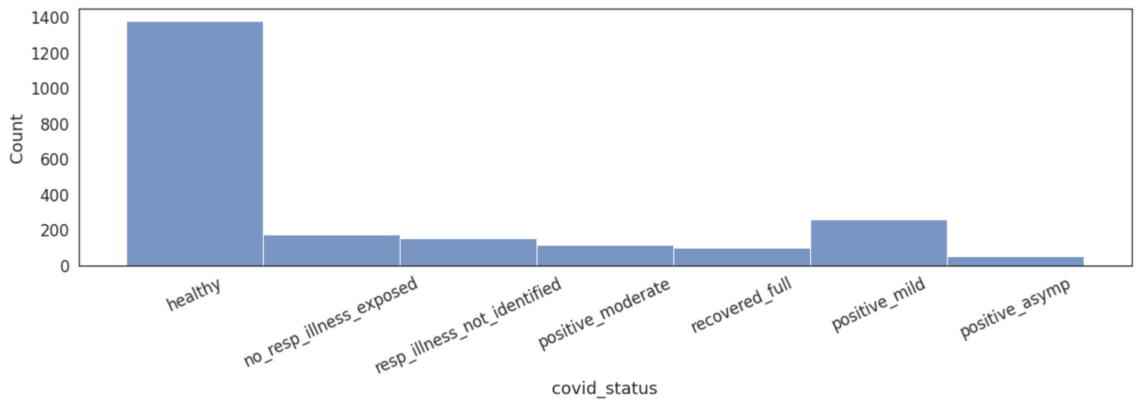

Figure 3.2: COVID-19 status label distribution in Coswara dataset

data, we get 429 COVID-19 positive and 1527 COVID-19 negative samples. Also, the Coswara data contains nine different audio recordings from each patient, such as counting, shallow and heavy coughing, breathing sounds, vowel pronunciations, etc. We only used the heavy cough recordings from each patient for our experiments. Lastly, we split the dataset into 75% training, 15% testing, and 15% validation set.

Class	**Count**
COVID-19 Negative	1527
COVID-19 Positive	429

Table 3.3: Sample count for Coswara Dataset

Class	**Count**
COVID-19 Negative	1788
COVID-19 Positive	596

Table 3.4: Sample count for COUGHVID Dataset

COUGHVID

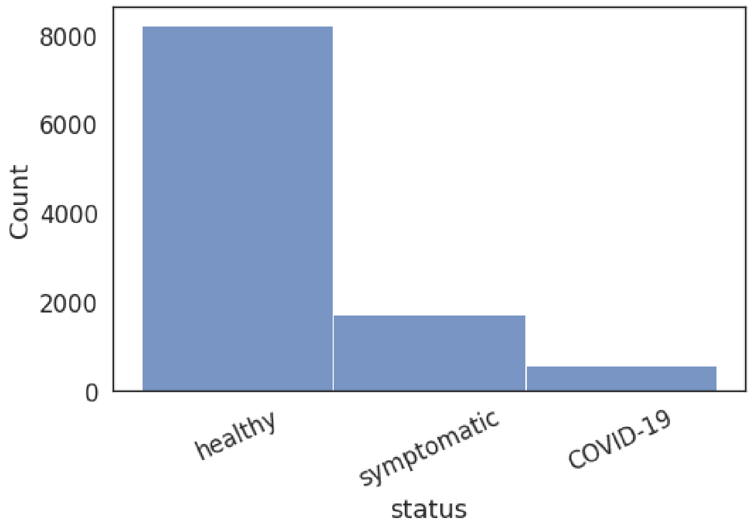

Figure 3.3: COVID-19 status label distribution in COUGHVID dataset

The COUGHVID dataset contains three different COVID-19 statuses collected from patient self-reports shown in Figure 3.3. The status ratio is highly imbalanced with the majority of the samples from the healthy patients, so we take all COVID-19 cough samples totaling 596 samples and only 1788 healthy samples from the dataset, which forms a 1:3 COVID-19 positive and negative ratio. We also utilize expert label annotations of diagnosis from the dataset to remove healthy patient recordings that contain an expert diagnosis of upper infection, lower infection, obstructive disease, and COVID-19. Finally, we split the data into 80% training, 10% test, and 10% validation sets.

Chapter 4

Proposed Methodologies

Our dataset contains just audio data, which is challenging to analyze for model training. So, we extract several audio features from the audio files before training the model. Since the data were imbalanced in class distribution, we also use a balanced mini batch training strategy to train our models. Finally, we employ several training strategies that improve our model's performance.

4.1 Feature Extraction

Classifier models find it difficult to identify a particular sound due to different complications in the raw sound data. For these reasons, to properly recognize audio data, it is required to extract relevant features from it. In our work, we extract two popular features Mel-Spectrogram and Mel Frequency Cepstral Coefficients (MFCC), to train and validate our models.

4.1.1 Mel-Spectrogram

The Mel-spectrogram is one of the most effective ways for extracting hidden elements from audio and visualizing them as an image [67]. To elaborate, a sound signal comprises many single-frequency sound waves, and while samples were taken over time, only the amplitudes of the signal are collected. Hence, the Fast Fourier Transform is used to extract meaningful information from an audio signal, and it creates a spectrum by decomposing the signal into its many frequencies and amplitudes, changing the frequency from the time domain to the frequency domain. To explain, a rapid way to perform the Fourier transform is using the Fast Fourier Transform (FFT), which allows us to evaluate the frequency content of a signal. However, because audio waves are non-periodic and change over time, the Short-Time Fourier Transform (STFT) is applied to overlapping window segments of the audio signal, which produces a spectrogram. A spectrogram is a visual representation of a signal's loudness or amplitude as it varies over time and at different frequencies. The Spectrogram plots Frequency (y-axis) against Time (x-axis), with different colors representing varying frequency amplitudes, and its brightness is proportional to the signal's energy. However, because people perceive frequency and loudness in a logarithmic pattern and are better at detecting lower frequencies than higher frequencies, spectrograms have little energy or color, and the frequencies and amplitudes are concentrated in a narrow range. As a result, the spectrogram is converted to the Mel Spectrogram to make it more intuitive for people to understand and more effective for CNN-based models. The Mel Spectrogram employs the Decibel scale instead of amplitude to show color or energy, and on the y-axis, mel scale is shown instead of Frequency [65].

The processes to create a mel spectrogram are as follows: first, air pressure samples are acquired over time to digitally represent an audio signal. The audio signal is then mapped from the time domain to the frequency domain using Fast Fourier Transform on overlapping windowed segments of the audio signal, which produces a Spectrogram. Finally, the Spectrogram is transformed into The Mel Spectrogram by converting the frequency to a mel scale, and the amplitude is changed to decibels.

For our experiments, we used 1024 as the length of the FFT window making with 512 number of samples between successive frames and took the whole frequency spectrum and divided it into 64 evenly spaced frequencies.

4.1.2 Mel Frequency Cepstral Coefficients (MFCC)

MFCC stands for Mel Frequency Cepstral Coefficients. We decided to use MFCC as it is a widely used feature in audio classification and speech recognition tasks. The potential of this feature is to represent the qualities and characteristics of the human auditory system accounts for its effectiveness.

To generate MFCC features from an audio signal the following steps are followed: As audio signals are always changing, they are initially framed into short frames of 20-40ms in length where it is assumed that the signals are stationary. Next, the discrete Fourier transform is then used to estimate the spectral density of the power spectrum of each frame, and the idea is influenced by the cochlea, a human organ in the ear. Based on the incoming sound's frequency the cochlea vibrates in various places, and different nerves send signals to the brain that particular frequencies are there. So, by determining the frequency present in the frame, the estimated spectral density of the power spectrum works similarly to the cochlea. Afterwards the mel filterbank is applied to the power spectrum, which sums the energy in each filter. The first filter in the Mel filterbank is quite narrow, and as the frequency rises, the filter becomes wider. Only the approximate amount of energy that occurs at each location of the mel filterbank is relevant. Here, the Mel scale tells us how wide and far apart the filterbanks should be. After collecting the filterbank energies, the logarithm of all energies is determined, which is inspired by the fact that people perceive loudness on a logarithmic scale. Since , the energies of the filterbanks are highly correlated and the filterbanks overlap, the discrete cosine transform is employed to decorrelate the energies, implying that diagonal covariance matrices can be used to represent the features. The resulting features are called Mel Frequency Cepstral Coefficients. [3][1].

For most of our experiments, we extracted 40 MFCC features from each audio file and used 2048 as the length of the FFT window, and used 512 numbers of samples between successive frames.

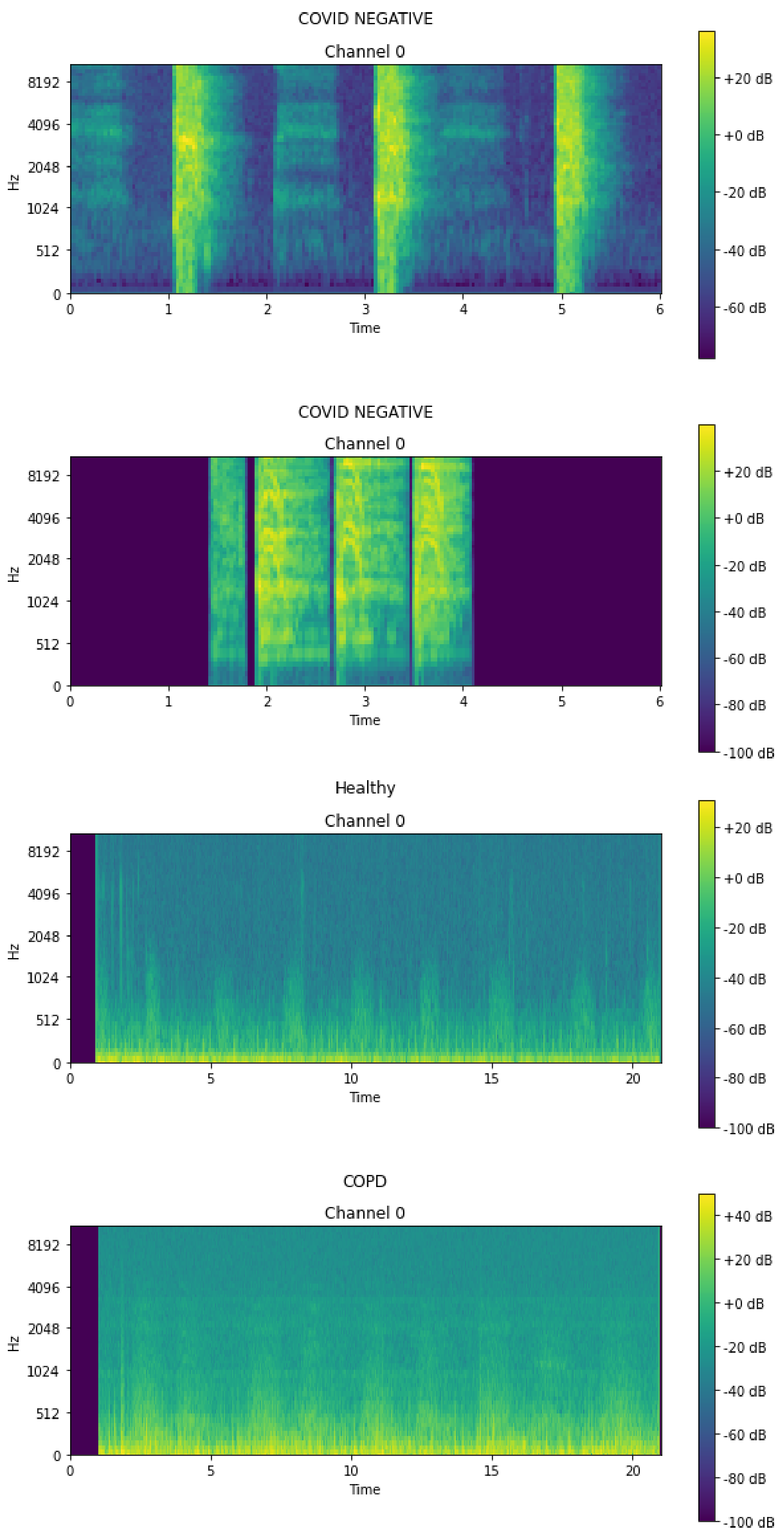

Figure 4.1: Mel-Spectrogram feature plot for different diagnosis

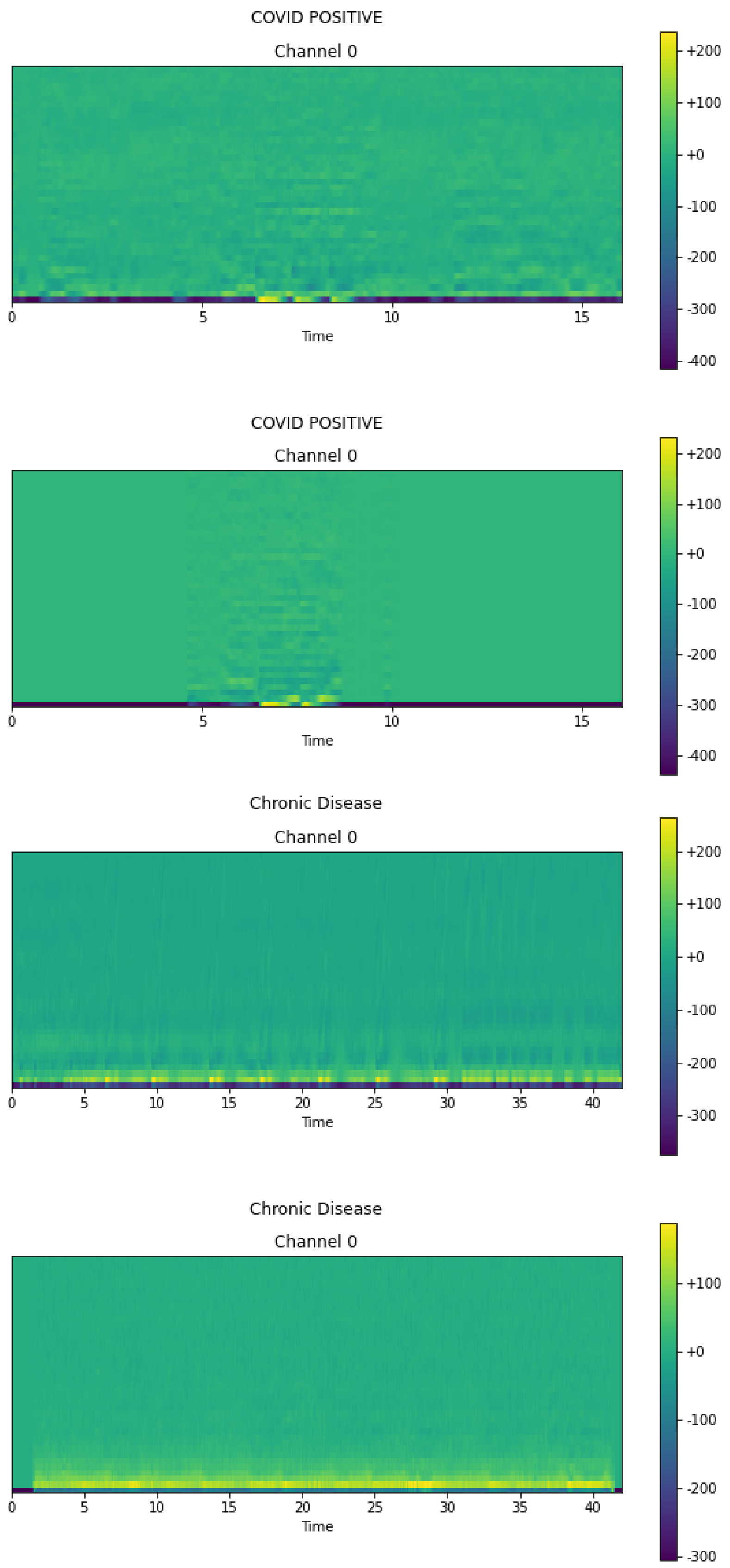

Figure 4.2: MFCC feature plot for different diagnosis

4.2 Feature Augmentation

One of the underlying issues with audio classification and speech recognition tasks is that the model overfits and fails to generalize when the training data is insufficient. The data augmentation technique is commonly applied to raw audio data in deep learning tasks to address the issue of data unavailability. However, using augmentation on raw audio data has certain disadvantages, such as increased processing time and cost. Furthermore, if the augmentation technique is unsuitable for the task, important audio information may be lost. Also, standard speech audio augmentation techniques such as pitch shift and altering the speed of the audio file can cause the audio to lose key audio characteristics in our respiratory sound classification task. Therefore, we employ feature augmentation techniques on the produced mel spectrogram features of our training audio data instead of using raw audio files. We utilize the Fastaudio library [45], which provides the mel spectrogram augmentation techniques Spectrogram Shifting (SGRoll), Time masking, and Frequency masking. The time and frequency masking augmentation technique was invented by Park et al. [37],which is applied directly to the mel spectrogram, and it is the input feature of the neural network. Since augmentation is done directly to the mel spectrogram features rather than raw audio data, no extra raw audio data is required for this augmentation. Furthermore, the augmentation is performed immediately during model training, which is also computationally efficient [38]. Besides, Park et al. [37] achieved state-of-the-art results using this augmentation technique and surpassed more complicated hybrid systems in automatic speech recognition tasks. Figure 4.3 shows the effect of three augmentations on Mel-Spectrogram features. We employ the following augmentation techniques in our experiments:

Time Masking

In time masking augmentation, [t0, t0 + t) time steps are masked, where t is the successive time steps along the x-axis of the mel spectrogram. Here, the value of t is chosen at random from a uniform distribution ranging from 0 to the time mask parameter T, while t0 is selected from $[0, \tau - t)$, where τ is the total number of time steps in the mel spectrogram. The purpose of this augmentation is to make spectrograms account for partial information loss in the time direction, allowing the model to learn more relevant features and distinguish audio features where the signal information is deformed in the time direction.

Frequency Masking

In frequency masking augmentation, Mel-Spectrogram frequency channels [f0, f0 + f) are masked, where f is the successive channels of the mel frequency. The choice of f depends on the uniform distribution from 0 to the frequency masking parameter F, while the choice of f0 is based on $[0, \nu - f)$, where ν is the mel frequency channels number. The motivation of this augmentation is to make the Mel-Spectrograms account for partial information loss in the frequency channels, allowing the model to acquire useful features even when partial frequency channel information is absent.

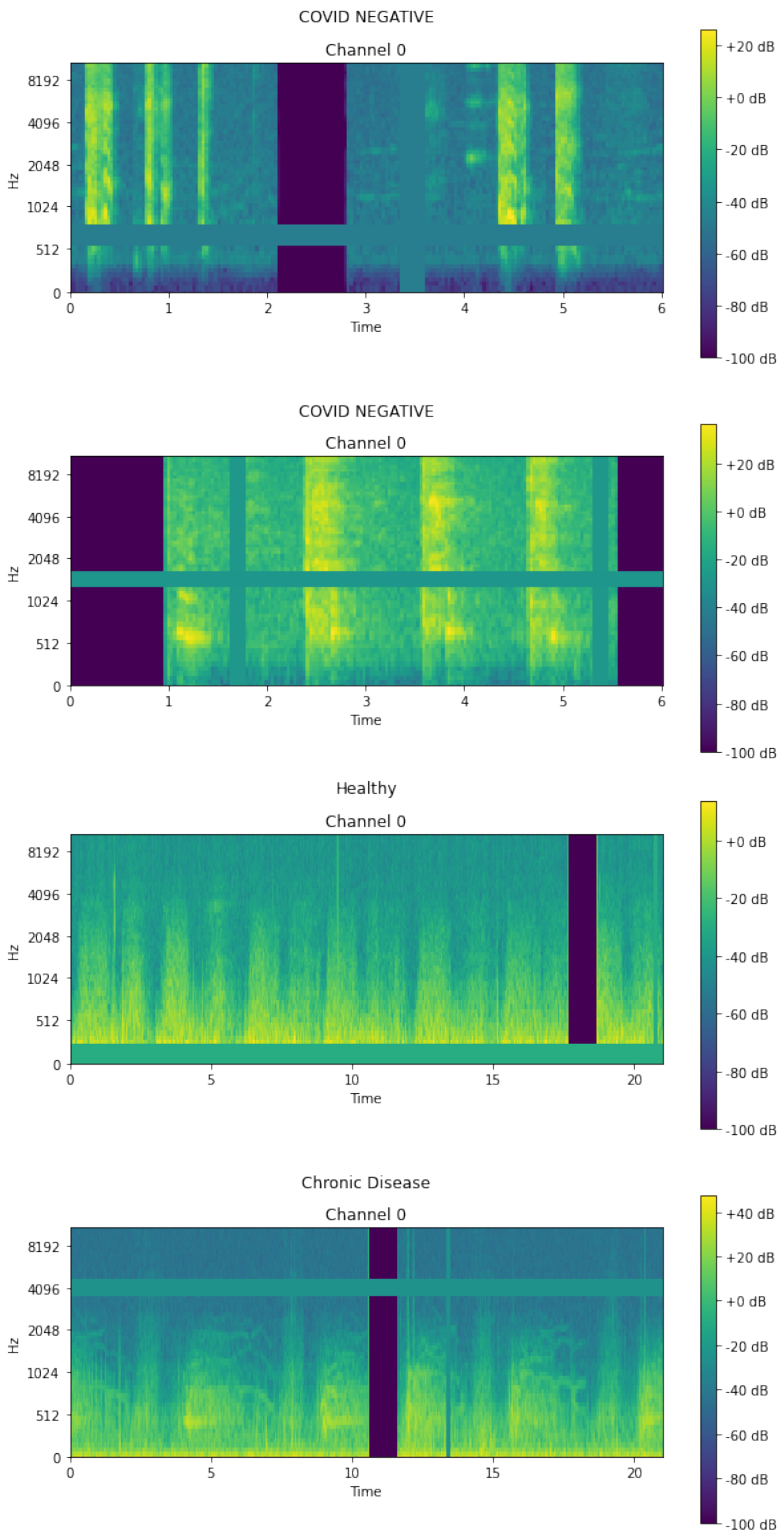

Figure 4.3: Augmented feature plot for different diagnosis

Spectrogram Shifting

This augmentation technique shifts the spectrogram along the x-axis and wraps around to the other side. The amount of shifting is picked at random between 0 and the maximum shifting parameter value, and the direction to shift the spectrogram is chosen at random or can be specified by the direction parameter value. The purpose of this augmentation is to reduce the model's bias toward positional values.

4.3 Training Strategies

For identifying a better model it is important to know what approach is being used to train it. Throughout this chapter, we'll briefly examine five training improvements that attempt to increase model accuracy even more. We utilize the fastai library [28] for most of our experiments.

4.3.1 Data Input

The class imbalance between data is one of the major obstacles in any deep learning task, as in most cases models trained on an imbalanced dataset fails to generalize and the gradient of the loss functions becomes flat. Learning from an imbalanced class distribution, the model performs well on the majority class, but the predictions of the minority classes are usually inconsistent. Data sampling approaches such as undersampling the majority class or oversampling the minority class are conventionally applied to overcome this problem. Class undersampling is the process of balancing the majority and minority classes by removing a certain number of samples from the majority class. However, there are certain drawbacks of undersampling, such as valuable data from the majority class getting lost and a lack of variety among data. On the other hand, the oversampling approach duplicates data in the minority class in order to equalize the number of samples in both minority and majority classes. The undersampling strategy also has certain drawbacks, such as overfitting the minority class since the model learns duplicate data [32].

During conventional deep learning model training, instead of using all of the samples at once, a subset of data samples called mini-batch is fed into the model one by one. The subset of samples is picked at random from all of the training samples and provided to the model until the original sample set is complete. The mini-batch training process then begins again, with the number of samples restored. When a mini-batch is generated from an imbalanced dataset, the mini-batches are likewise imbalanced, causing the model to learn classes in unequal proportions, overfitting it towards the majority class. It is necessary to balance data or the mini-batches to avoid the problem of model overfitting. Therefore we employ a balanced mini-batch training technique with feature augmentation to train our models. For this, we first count the number of samples in each class of training set, then assign a weight to each sample depending on the total sample count. Here, the weight refers to the probability that one sample would appear on a mini-batch. As a result, the majority class has a lower weight than the minority class. These weights assigned to each sample ensures that each mini-batch has an almost equal number of class samples which helps the model learn each class equally, reducing overfitting. When the minority class is assigned higher weights, samples from that class are oversampled by duplicate data. However, duplicate data can cause the model to overfit if the model learns the sample data multiple times. Hence, we only utilize the mini-batch oversampling approach with our mel-spectrogram feature that includes three augmentation procedures which ensures that an augmented version of the feature data is used rather than the duplicate data.

$$W_{class} = \frac{1}{\text{Total number of samples of that class}}$$

4.3.2 Learning Rate Finder

Learning rate is one of the most critical hyper-parameters to tune for deep learning. The choice of learning rate is particularly crucial, as a small learning rate can make model training exceedingly slow, and a large learning rate can cause the model to diverge. So, to train our model, we utilize the learning rate finder method invented by Smith (2017) [24], which provides a good approximation of the ideal learning rate value. The process includes monitoring the loss early while training, and the model trains for one epoch using a mini-batch by a small learning rate while the loss value is stored. Then the model trains for another epoch by increasing the learning rate by a specified percentage. Thus the method repeats until the loss value worsens and starts to diverge. Following that, a smoother version of the plot shows the recorded loss value of each iteration, therefore we use this plot to determine an optimal learning rate to train our model. The proposed optimum value is one magnitude order less than the plot's minimal loss value. Since the minimum loss value is not optimal since it is too high and at the edge where the loss value starts diverging. Additionally, the learning rate from the steepest slope, around 2/3 of the way through the longest valley, or the learning rate value found by the interval slide rule are excellent choices [61]. To train our model, we empirically tested several learning rates using the learning rate plot and found the learning rate from the longest valley to be optimum.

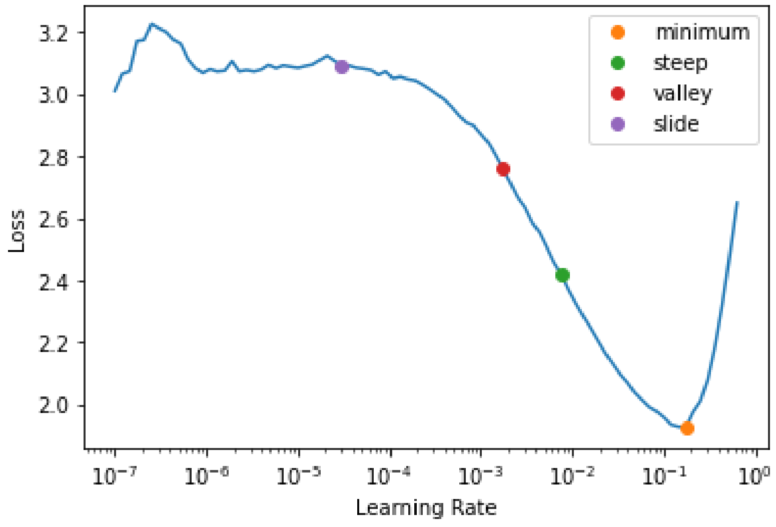

Figure 4.4: Learning Rate plot

4.3.3 1cycle Policy

Optimizing a deep learning model is a difficult task as several hyper-parameters can affect how the model performs for better or worse. For example, a low learning rate causes a model to learn slowly and takes many epochs to converge, whereas a high learning rate causes the loss value to fluctuate around the local minimum. Finding the sweet spot between small and large learning rate is critical to train the model quickly and efficiently. The basic technique to train the model is to test several learning rates at random and select the one that produces the best results; however, this strategy is both time intensive and impractical. So, we selected our applied learning rates using Smith's learning rate finder, which provides an excellent approximation of the optimum learning rate [24].

In traditional deep learning model training, learning rate schedules and adaptive learning rates are conventionally used, as choosing one ideal static learning rate is challenging. Learning rate schedules consist of strategies like Step Decay, Time-Based Decay, and Exponential Decay to decrease the learning rate during model training. This technique remains constant throughout training since it is determined before the training process begins, which has the disadvantage of being unable to adjust to the specific characteristics of the dataset. Adaptive learning rate strategies, on the other hand, can lessen the problem which include employing adaptive gradient descent algorithms such as Adadelta, Adam, Adagrad, RMSprop; however, these strategies are computationally costly [36][22].

Thus we employ the "1cycle" policy invented by Smith [33]; a modified version of their Super-Convergence method to train the neural networks. Moreover, using super-convergence methods, models train in order of magnitude faster than standard training methods; and can increase the performance of models when the training data is limited. The "1cycle" policy ensures the accuracy to plateau before training ends, and it is a combination of Simulated annealing and Curriculum learning [33].

The procedure involves two phases of the learning rate, one of which raises the learning rate and the other of which reduces the learning rate; these two steps are called '1cycle'. The learning rate in the '1cycle' policy increases up to the maximum learning rate value that we selected using the learning rate finder plot, which can decrease up to ten times lower than the maximum value. The '1cycle' executes fewer times than the total number of epochs, and learning rates get reduced by many orders of magnitude for the remaining epochs. The motivation is to prevent the model from overfitting by using the high learning rate as a regularization method. The cyclic learning rate is also not computationally expensive, as it terminates the need to find the optimum learning rate because it will fall between the minimum and maximum bound [36].

4.3.4 Transfer Learning

Deep learning models typically perform well and learn to predict accurately when huge amounts of data are accessible; however, data scarcity remains a big issue in most deep learning tasks. Transfer learning is the usual solution to effectively use deep learning when large amounts of data are unavailable. The transfer learning procedure applies previously acquired features and weights from an existing model to a new task. The key idea is to reuse the learned properties of a model trained on a large dataset to a different problem with little data available. Using transfer learning, a model does not need to be trained from the beginning but instead can train on already existing features from pre-trained models, which resolves the resource and time constraints. Transfer learning also improves model efficiency and accelerates model's generalization [23].

In conventional image classification tasks, pre-trained models utilizing the ImageNet dataset are used for transfer learning. Here, models trained on the ImageNet dataset generalize on diverse image classification tasks because models can learn from a large dataset containing 1000 different image categories with 1.2 million training data. Some convolutional neural network models trained on the ImageNet dataset are ResNet, VGGNet, DenseNet, etc. Lower convolution layers of these models capture image features such as edges, diagonal patterns, and lines, and higher convolutional layers capture specific image features such as human body parts, eyes, faces, etc. The fully connected layers of these CNN models are task-specific and used to categorize images into different categories. So, to use these pre-trained models, the last fully connected layers are changed according to the specific task's number of classes [31].

In our classification task, the extracted audio features are image representations of the sound information, which allows us to use CNN-based architectures with transfer learning from image data. Although image data differs from the spectrogram or MFCC features, several fundamental operations of the first few convolutional layers are identical, such as identifying edges, patterns, diagonal lines, blobs of an area, etc. These common types of functionality, and the fact that our generated sounds features are images, encourage us to employ transfer learning techniques. Also, we use the fine-tuning strategy of transfer learning to change the weights of the higher convolutional layers as these are not specific to our task. For the fine-tuning procedure we first train the last fully connected output layer with all other layers frozen for one epoch, then we unfreeze all the layers and train our whole model for the remaining epochs.

4.3.5 Model Callbacks

Functions applied at a given stage of model training are named callbacks; the callbacks function has a significant role in deep learning model training as it can change the behavior of the model. To train our model, we applied following callbacks functions:

Early Stopping Callback

It terminates model training depending on the monitored parameter value. The callbacks prevent the model from overfitting when the model performance is not improving. We use early stopping callbacks to monitor the accuracy metric when using the transfer learning strategy, and we use early stopping callbacks to monitor the validation loss value while training with the 1cycle learning rate.

Save Model Callback

It saves the best model at the epochs where the monitored value is higher than prior epochs. This callback is essential to save the model with the best weights and performance. We utilize these callbacks to monitor the accuracy value and save the model when it is at its best.

Chapter 5

Model Architecture

Deep Convolutional Neural Networks are promising architecture that has been pioneered in recent years. There are numerous prominent deep learning models, and we employed six of them which are VGGNet, ResNet, XResNet, AlexNet, SqueezeNet, and DenseNet. The models have been trained and tested accordingly and the outcomes are quite convenient.

5.1 VGGNet

VGGNet is a convolutional neural network model that is commonly considered to be one of the best deep learning architectures currently available at the moment. The major distinguishing characteristic of VGGNet is that, instead of having a large number of hyper-parameters, they concentrated on generating a 3x3 convolution filter with stride 1. Moreover, they also used max pooling layers of 2x2 filters with stride 2 and same kind of padding. The convolution layers and max pooling layers are used throughout the structure frequently. VGGNet includes a variety of configurations that vary in depth of layers.

During training, the input to the first convolutional layer is a 224 x 224 RGB image of specific size. With the filters set to 3x3 receptive fields, the image was progressed via a sequence of convolutional layers. It also incorporates 1x1 convolution filters in one of the layouts. The stride of the convolution is fixed to 1 pixel by default. For 3x3 convolution layers the spatial padding of the input is 1 pixel such that the spatial resolution is kept after convolution. Max-pooling is performed with stride 2 over a 2x2 pixel window. However, it is not applied to all of the convolution layers. Max-pooling followed after every two or more convolutional layers, thus a total five max-pooling layers were observed that carried spatial pooling. After all convolutional layers of varied depths, three Fully-Connected (FC) layers are applied in all VGGNet configurations. The first two each have 4096 channels, whereas the latter has 1000 channels. In all networks, the fully connected layers are configured in the same way and followed by a soft-max layer. All the convolution and FC layers are equipped with an activation function which is Rectified Linear Unit (ReLU).

The only difference between the layouts is the depth, which follows the basic design found in architecture. Two CNN architectures that we used in our dataset are VGG16 and VGG19. The 16 in VGG16 refers to the number of weight layers (13

VGGNet Configuration		
VGG16	VGG16	VGG19
16 weight Layers	16 weight Layers	19 weight Layers
224×224 RGB image		
Conv-3 64	Conv-3 64	Conv-3 64
Conv-3 64	Conv-3 64	Conv-3 64
maxpool		
Conv-3 128	Conv-3 128	Conv-3 128
Conv-3 128	Conv-3 128	Conv-3 128
maxpool		
conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256
conv1-256	conv3-256	conv3-256
		conv3-256
maxpool		
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv1-512	conv3-512	conv3-512
		conv3-512
maxpool		
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv1-512	conv3-512	conv3-512
		conv3-512
maxpool		
FC-4096		
FC-4096		
FC-1000		
softmax		

Table 5.1: Layers arrangements of VGGNet architectures

conv. layers and 3 FC layers). This network is quite huge, with approximately 138 million parameters. Similarly, the 19 in VGG19 denotes the presence of 19 weight layers (16 conv. layers and 3 FC layers). This network is the largest of VGG configuration, with estimated 144 million parameters [7].

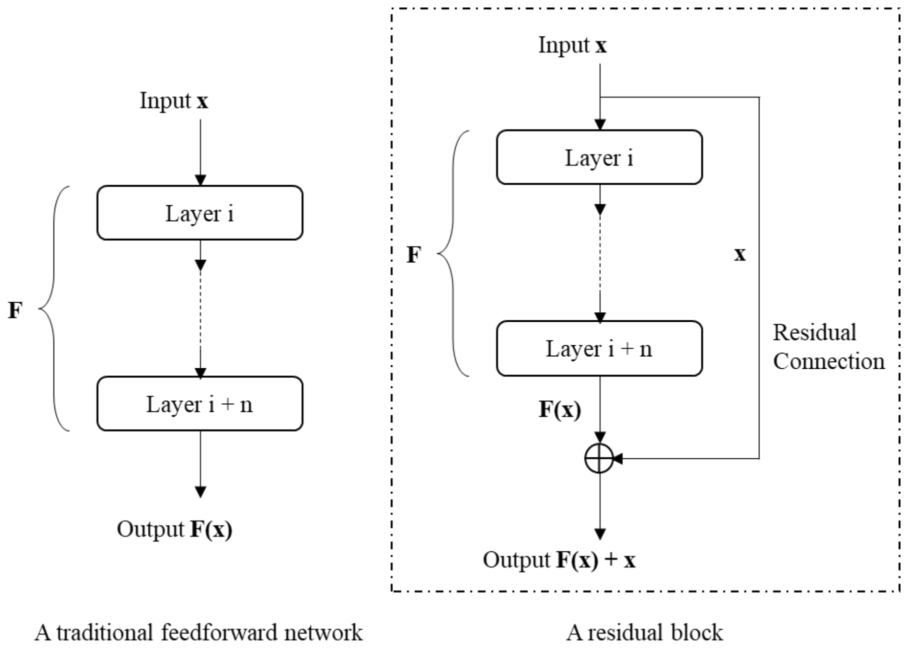

Figure 5.1: Difference between a traditional feedforward connection and a residual block

5.2 ResNet

ResNet is referred to as residual network which is a simple yet efficient method to ease the complexity of deep neural network training. It has been widely embraced by computer vision, natural language processing and reinforcement learning.

When the network grows deeper, the accuracy becomes saturated and diminishes quickly, causing a degradation problem. This degradation is neither originated by overfitting nor by additional layers to a deep network. Furthermore, adding more layers to networks causes the vanishing gradient problem, which affects convergence. This implies that when the route for information from the input to the output layers lengthens, certain data may ‘vanish’ or be lost, reducing the network’s capacity to train effectively.

Microsoft proposed a deep residual learning architecture to address this issue. By skipping some layers, residual connection provides an alternative way for data to reach later regions of the neural network. They allowed these layers to fit a residual mapping precisely rather than assuming every few stacked layers to match a specified underlying mapping directly.
In Figure 5.1, a series of layer i to layer i+n is considered and the input of the layer i is denoted by x.The result of the layer i+n is F(x) where x will simply run through these layers one after another in a classic feedforward setup. The residual connection conducts element-wise addition F(x) + x after applying identity mapping to x. A residual block is the term used in literature to describe the entire architecture that takes an input x and creates an output F(x) + x. An activation function ReLU is also applied in a residual block.

The usual plain deep network has the same amount of filters for the similar feature maps. To maintain the time complexity of each layer, the amount of filters is doubled when the size of the features map is halved. It is indeed important to note that ResNet networks have less filters and are easier than VGG networks.

Generally several residual blocks, of the same or distinct architectures, are employed throughout the neural network. When the input x and the output F(x) are of the same size, then the element-wise addition F(x) + x can be utilized directly. If their dimensions change, we can implement identity mapping with additional zero padding for higher dimensions. Moreover, by using a linear transformation (i.e. multiplication by a matrix W) we can use F(x) + Wx instead of identity mapping to match the dimensions.

The input of a ResNet is a 7x7 convolution layer with stride 2, followed by max pooling layer and four identical blocks which have different configuration and iteration. It conducts 3x3 convolution for ResNet-18 and ResNet-34 and bypasses the input after every 2 convolutions. However, in the case of ResNet-50 and ResNet-101 it conducts 3x3 and 1x1 convolution and bypasses the input after every 3 convolutions. Every layer is incorporated with an activation function ReLU. An average pooling layer as well as a 1000-way fully-connected layer with softmax concludes the network [10].

Layer Name	**output size**	**18-layer**	**34-layer**	**50-layer**	**101-layer**	**152-layer**
conv1	112x112	7x7, 64, stride 2				
conv2_x	56x56	3x3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28x28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14x14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7x7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1x1	average pool, 1000-d fc,softmax				
Flops		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 5.2: Layer arrangement of ResNet architectures

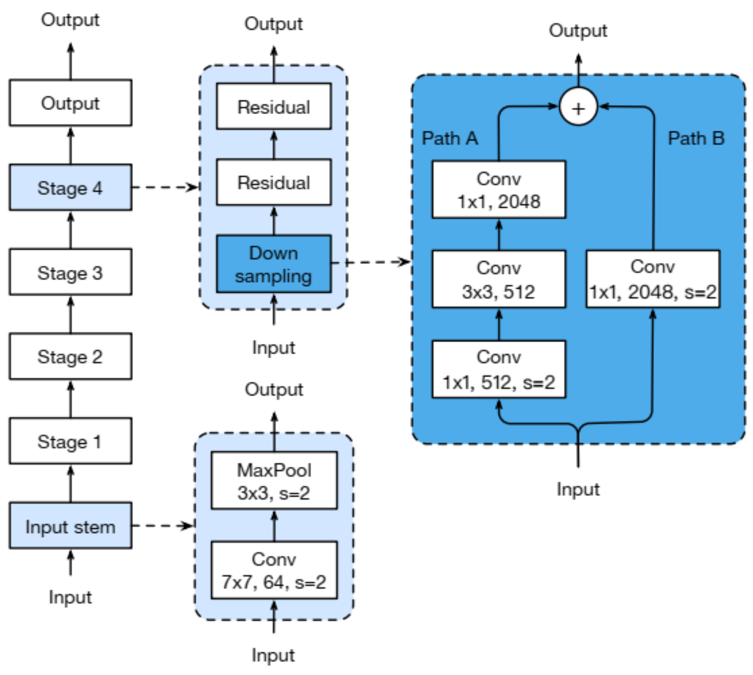

Figure 5.2: Traditional ResNet architecture

5.3 XResNet

XResNet is an enhanced version of ResNet that is used consistently to increase model performance which was first introduced in paper [27] further mentions that higher accuracy can be achieved by making slight changes to the data preprocessing and model architecture as well as learning rate schedule and loss function.

An input stem, four succeeding stages, and a final output layer make up a ResNet network. A 7x7 convolution with an output of 64 channels is used in the input stem followed by a 3x3 max pooling layer with stride 2 for each layer. The input stem decreases the height and width of the input by four times while increasing the size of the channel to 64. Beginning with stage 2, each stage starts with a downsampling block, followed by multiple residual blocks. There are two paths in the downsampling block: path A and path B. Path A is arranged in a bottleneck structure and has three convolutions, each with a kernel size of 1x1, 3x3, and 1x1 respectively. In order to halve the input width and height the first convolution has a stride of 2, while the output channel of the last convolution is 4 times larger than the preceding two. Path B transforms the input shape into the output shape of path A using a 1x1 convolution with a stride of 2, so that the outputs of both paths can be added to get the downsampling block's output. The only difference between a residual block and a downsampling block is that a residual block only uses convolutions with a stride of one.

An improvement of ResNet was introduced in the paper [16] which changed the downsampling block of ResNet. Since the path A of the downsampling block utilizes a kernel size of 1x1 along with a stride of 2, the convolution overlooks three-quarters of the input feature map. The work [16] alters the stride size of the first two convolutional layers in path A, ensuring that no data is lost. Path A's output structure remains unaffected as the second convolution has a filter size of 3x3. The

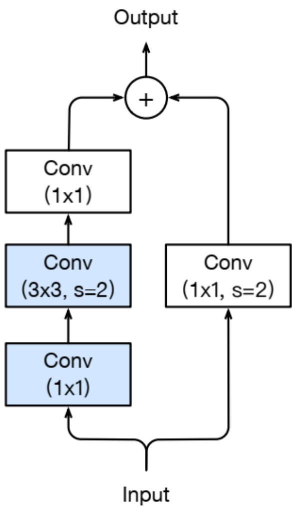

Figure 5.3: Modified downsampling block of ResNet

paper [27] shows that XResNet is influenced by the above modification of ResNet, it is seen that in path B of the downsampling block, the 1x1 convolution also excludes three quarter of the input feature maps, thus it is altered so that no data is discarded. Also it is deduced that altering the convolution layer's stride from 1 to 2 and placing a 2x2 average pooling layer with stride 2, works better practically and has a little effect on the computational cost. In total XResNet-50 enhances ResNet-50 by 1% and in terms of training throughput, it is only 3% slower than ResNet-50.

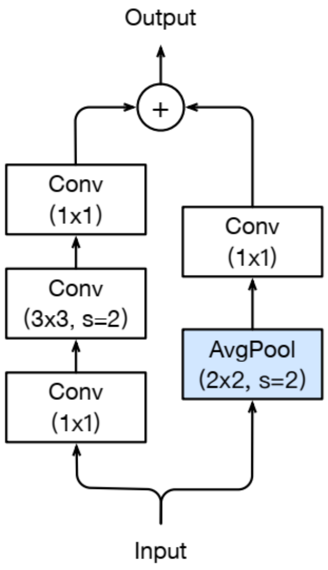

Figure 5.4: Modification proposed by XResNet

5.4 Alexnet

AlexNet is a convolutional neural network which has a major effect on machine learning, especially when it comes to computer vision and deep learning after [19] was published. There are now far more complicated CNNs that can operate very quickly on faster GPUs, even on very huge datasets.

AlexNet comprises eight weighted layers, with the first five being convolutional layers while the rest are fully connected layers. In a convolutional layer, there are usually several filters of similar size. The initial convolution layer includes 96 11x11x3 kernels. The height and width of the filters are generally equal, and the number of channels determine the depth.

The first two convolutional layers are followed by overlapping max pooling layers. Max Pooling layers are commonly used to reduce the layers' height and width while keeping the depth constant. Overlapping max pooling layers are comparable to regular max pooling layers, except that the neighboring windows over which the maximum is determined, overlap one another.

The third, fourth, and fifth convolutional layers include direct connections between them. An overlapping max pooling layer feeds the output of the fifth convolutional layer into a series of two fully connected layers. Dropout was utilized before the two fully connected layers to prevent significant overfitting. The concluding fully connected layer's output is sent to the 1000-way softmax classifier. Every convolutional and fully-connected layer's output is equipped with ReLU non-linearity.

5.5 SqueezeNet

SqueezeNet is a deep convolutional neural network (CNN) and has a compressed architecture design. The model generates an architecture so the amount of parameters can be reduced, particularly with the help of fire modules that use 1x1 convolutions to "squeeze" the parameters. SqueezeNet achieved the same level of accuracy as AlexNet on the ImageNet dataset but with 50 times less parameters. The model size of AlexNet is 240MB whereas the model size of SqueezeNet is only 4.8MB. The SqueezeNet fire module is said to encompass a squeeze convolution layer that

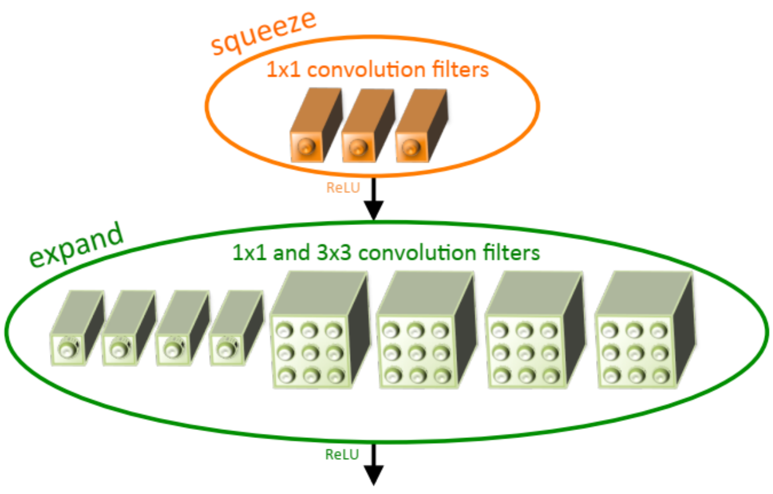

Figure 5.5: Microarchitectural view of Fire Module used in SqueezeNet

has solely 1x1 filters, flowing into an expand layer with a combination of 3x3 and

1x1 convolution filters. In the Fire module there are three adjustable hyperparameters which are the number of 1x1 convolution filters in the squeeze layer as well as in the expand layer and the number of 3x3 convolution filters in the expand layer.

In order to design this CNN architecture three strategies have been adapted. The first strategy it follows is to replace the 3x3 filters with 1x1 filters. This modification will have 9x less parameters and computational cost. Next, it declines the input channels to 3x3 filters utilizing the squeeze layers to maintain a smaller number of parameters. Lastly, in order to keep the activation map large as it might lead to higher classification accuracy it delays the downsample in the network by late allocation of the pooling layers. SqueezeNet on figure 5.6 starts with a soli-

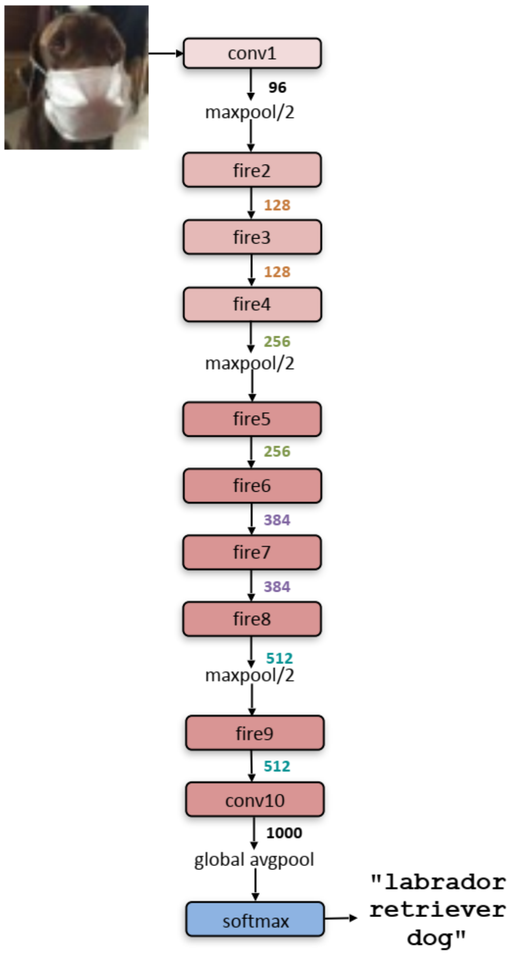

Figure 5.6: Architecture of SqueezeNet

tary convolution layer, then 8 Fire modules, and finally a convolution layer. From the beginning to the end of the network, the number of filters per fire module is steadily increased. Moreover, the conv1, fire4, fire8, and conv10 layers are followed by max-pooling layers with stride of 2. Some more configurations of SqueezeNet like SqueezeNet with simple and complex bypass connections are inspired from the ResNet architecture [14].

5.6 DenseNet

DenseNet is a convolutional neural network in which each layer is connected to all of the levels below it. In a conventional feed-forward Convolutional Neural Network (CNN), apart from the first one (which takes in the input) every convolutional layer draws the output of the preceding convolutional layer and employs an output feature map that is then passed on to the next convolutional layer. The 'vanishing gradient' problem emerges when a CNN network gets deeper. DenseNets solves this problem by easing the connectivity pattern between layers and altering the usual CNN architecture. Since the DenseNet architecture is connected to every other layer directly, its name is Densely Connected Convolutional Network.

DenseNets are separated into dense blocks which are formed by multiple convolutional layers. Inside a dense block, inputs of each layer are concatenated with the feature maps of all the previous layers, rather than being added element-wise. The feature maps have been considered as the network's global state. After passing through each convolutional layer, the feature map expands in size, with each layer contributing 'k' features on top of the global state. This parameter 'k' is called the network's growth rate which controls the amount of data added to each layer of the network. Even though each layer only provides a 'k' number of output feature-maps,

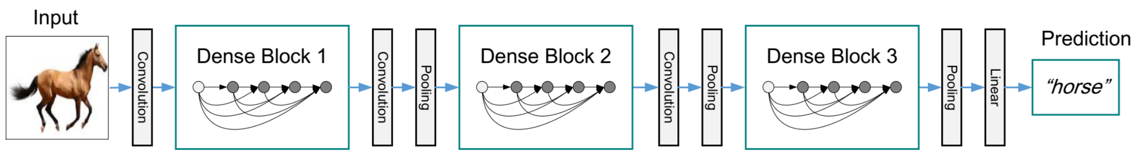

Figure 5.7: Three dense blocks are shown in a deep DenseNet

the number of inputs can be fairly high, especially when more layers are added. To increase the efficiency and speed of computations, a 1x1 convolution layer has been included as a bottleneck layer before each 3x3 convolution.

The dimensions of the feature maps remain constant inside a dense block however the amount of filters between them keeps changing. When the size of feature maps varies, use of the concatenation method is not reasonable. However, down-sampling of layers, which minimizes the size of feature-maps through dimensionality reduction to achieve higher computation speeds, is an important aspect of CNNs. Thus, to get rid of this dilemma Transition Layers are added between the blocks that lower the number of channels to half of what they were before. A 1x1 convolutional layer and a 2x2 average pooling layer with stride 2 are included in each transition layer. DenseNets require fewer parameters than a conventional CNN because of the bottleneck structure and transition layers.

For our classification problem, we are using DenseNet-121, among the various DenseNet configurations. All the configurations have an input of a basic convolution layer with 64 filters of size 7x7, followed by a 3x3 max pooling layer both having a stride of 2. Next the feature map advances through four dense blocks with transition layers between them. The final output of the dense blocks goes through a 7x7 global average pooling before being attached by a softmax classifier [29].

Layers	Output Size	DenseNet-121
Convolution	112 x 112	7 x 7 conv, stride 2
Pooling	56 x 56	3 x 3 max pool, stride 2
Dense Block (1)	56 x 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 x 56	1×1 conv
	28 x 28	2 x 2 average pool, stride 2
Dense Block (2)	28 x 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 x 28	1×1 conv
	14 x 14	2 x 2 average pool, stride 2
Dense Block (3)	14 x 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Transition Layer (3)	14 x 14	1×1 conv
	7 x 7	2 x 2 average pool, stride 2
Dense Block (4)	14 x 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$
Classification Layer	1 x 1	7×7 global average pool
		1000D fully-connected, softmax

Table 5.3: Architecture of DenseNet-121

Chapter 6

Evaluation

We utilize several metrics to correctly evaluate the performance of our models, including Accuracy, Area Under the Receiver Operating Characteristic Curve (ROC AUC), F1 scores, Recall score, Sensitivity score and Precision score.

Accuracy is the most widely used threshold metric for evaluating classifiers which shows the ratio of accurately predicted samples to total samples.

$$Accuracy = \frac{\text{True Negative} + \text{True Positive}}{\text{True Negative} + \text{True Positive} + \text{False Positive} + \text{False Negative}}$$

Here, True Positive (TP) refers to the outcomes where the positively predicted classes are actually positive, and True Negative (TN) are those outcomes where the Negatively predicted classes are actually negative. In contrast, False Positive (FP) are those positive predictions where the actual class is negative and False Negative (FN) are those negative predictions where the actual class is positive. In short, TP and TN are the correct predictions, whereas FP and FN are the wrong predictions.

Recall score, also known as the Sensitivity score, is the percentage of all positive samples accurately predicted by the model. In our task, the recall metric indicates what percentage of total patients with diseases were predicted by the model to have that disease. The high recall score is significant when we need to identify more sick patients.

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Specificity score is the opposite of the Recall score and is the percentage of all negative samples accurately predicted by the model. The specificity metric in our task reveals what percentage of total healthy patients the model predicted to be healthy.

$$Specificity = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

Precision score assesses a model's ability to predict positive labels correctly and displays the percentage of relevant findings from all positive predictions. The precision metric in our task indicates the percentage of patients who had the disease and is diagnosed by the model. A high precision score is essential if we focus on lowering False positives rates.

$$Precision = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

The F1 score, obtained by determining the harmonic mean of precision and recall, gives a balanced combination of the recall and precision scores.

$$F1score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The ROC-AUC metric measures a model's ability to differentiate across classes. We utilize The ROC-AUC score for our Covid-19 infected and healthy patients' cough detection task to assess the model's ability to differentiate between positive and negative cough recordings. Also, we utilize the ROC-AUC plot to display the model's predictions at different threshold values. For our classification tasks, we display the macro-average scores for Recall, Precision, F1, and Sensitive. The macro-average computes the score for each class independently and then averages them, assigning equal weight to each class. The macro-average score is particularly useful because we aim to classify all diseases and healthy individuals separately.

We show our results for each experiment separately in the tables below. We use the feature column to show the results for three types of feature inputs (i.e MFCC, Mel-Spectrogram, and Mel-Spectrogram with data augmentation). Additionally, we show the results from the two types of the training process: fine-tuning and the 1cycle policy in separate tables.

6.1 Respiratory Diseases Classification

The tables below show the experimental result of respiratory disease classification where we use the ICBHI Challenge Dataset. Table 6.1, 6.2 represent the results for our six class respiratory disease classification with both fine-tune and 1cycle procedure respectively. Similarly, Table 6.3, 6.4 depict the experimental finding of our 3 class classification with both training procedures as well.

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	Mel-Spectrogram	90.22	68.16	44.72	42.01	94.3	54.38
	Mel-Spectrogram(Augmented)	85.87	78.43	51.28	59.86	96.99	47.31
	MFCC	86.41	60.75	28.64	29	92.5	28.6
ResNet-34	Mel-Spectrogram	87.5	75.69	47.62	54.67	96.71	47.76
	Mel-Spectrogram(Augmented)	86.41	86.25	64.03	75.41	97.08	58.12
	MFCC	86.41	60.75	28.64	29	92.5	28.6
ResNet-50	Mel-Spectrogram	88.04	75.97	47.1	55.72	96.23	44.12
	Mel-Spectrogram(Augmented)	88.59	82.79	57.18	68.11	97.47	54.1
	MFCC	89.13	68.25	41.43	41.8	94.69	41.51
ResNet-101	Mel-Spectrogram	90.76	79.51	56.92	61.75	97.27	56.45
	Mel-Spectrogram(Augmented)	83.7	75.66	47.24	55.84	95.48	44.37
	MFCC	86.96	70.3	41.75	44.56	96.04	41.9
VGG16	Mel-Spectrogram	89.67	80.37	49.14	63.66	97.09	47.25
	Mel-Spectrogram(Augmented)	91.3	86.32	66.74	74.14	98.5	64.24
	MFCC	88.04	75.19	47.36	54.72	95.65	43.22
VGG19	Mel-Spectrogram	92.39	82.98	60.18	67.83	98.12	60.45
	Mel-Spectrogram(Augmented)	91.3	90.18	72.82	81.86	98.5	67.56
	MFCC	90.22	72.19	51.47	50.64	93.74	56.13
AlexNet	Mel-Spectrogram	88.59	64.99	33.88	37.09	92.88	33.86
	Mel-Spectrogram(Augmented)	82.07	76.43	43.85	55.95	96.92	39.08
	MFCC	86.39	68.74	36.16	43.39	94.08	32.66
SqueezeNet	Mel-Spectrogram	89.13	72.3	49.77	50.48	94.12	54.07
	Mel-Spectrogram(Augmented)	81.52	75.2	41.97	53.57	96.83	37.78
	MFCC	85.33	59.13	25.86	26.51	91.74	28.14
DenseNet-121	Mel-Spectrogram	91.85	80.53	64.26	64.18	96.88	72.9
	Mel-Spectrogram(Augmented)	82.07	74.84	41.91	52.77	96.92	39
	MFCC	85.33	74.84	45.8	53.35	96.33	52.32
XResNet-50	Mel-Spectrogram	89.67	85.26	62.32	72.87	97.65	56.2
	Mel-Spectrogram(Augmented)	86.41	84.3	55.51	70.92	97.68	50.24
	MFCC	87.5	81.26	54.71	64.67	97.85	50.23

Table 6.1: Experimental results for 6 class classification using fine-tune approach on ICBHI dataset

6.2 COVID-19 Cough Classification

In this section, the tables illustrate the COVID-19 infected and Healthy patients' cough classification result where we use the Coswara and COUGHVID Dataset. The experimental results based on the Coswara dataset are shown in Table 6.5, 6.6 whereas experimental results based on the COUGHVID dataset are depicted in Table 6.7, 6.8. All the evaluation metrics are included in the tables including both of the training procedures.

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	Mel-Spectrogram	92.93	86.52	72.61	75.4	97.64	83.12
	Mel-Spectrogram(Augmented)	85.87	83.21	50.41	68.85	97.57	42.41
	MFCC	91.85	80.43	64.78	65.14	95.73	65.03
ResNet-34	Mel-Spectrogram	88.59	69.5	35.21	42.12	96.88	31.5
	Mel-Spectrogram(Augmented)	90.76	82.38	60.35	66.36	98.41	58.05
	MFCC	87.5	70.4	40.25	44.66	96.14	40.14
ResNet-50	Mel-Spectrogram	90.22	86.12	65.16	73.93	98.32	64.83
	Mel-Spectrogram(Augmented)	86.96	80.9	54.05	64.62	97.19	50.4
	MFCC	86.41	72.47	46.4	49.01	95.94	51.61
ResNet-101	Mel-Spectrogram	90.76	75.77	56.65	57.15	94.4	67.62
	Mel-Spectrogram(Augmented)	85.33	79.65	55.15	62.98	96.33	51.99
	MFCC	85.87	78.43	51.28	59.86	96.99	47.31
VGG16	Mel-Spectrogram	90.76	85.81	68.42	74.93	96.69	64.55
	Mel-Spectrogram(Augmented)	87.5	83.73	57.69	70.17	97.29	52.34
	MFCC	89.67	77.22	54.64	57.36	97.07	63.18
VGG19	Mel-Spectrogram	84.24	76.51	46.28	56.31	96.71	42.21
	Mel-Spectrogram(Augmented)	86.96	77.84	48.6	59.06	96.61	43.19
	MFCC	86.41	67.66	35.54	39.96	95.37	33.92
AlexNet	Mel-Spectrogram	86.96	61.71	34.15	31.38	92.03	46.13
	Mel-Spectrogram(Augmented)	81.52	81.78	50.47	66.74	96.82	48.25
	MFCC	86.39	65.37	33.83	38.1	92.64	31.79
SqueezeNet	Mel-Spectrogram	90.76	74.5	52.34	54.03	94.97	51.05
	Mel-Spectrogram(Augmented)	77.72	78.56	45.04	61.51	95.6	40.03
	MFCC	79.89	66.29	33.45	37.74	94.83	31.41
DenseNet-121	Mel-Spectrogram	93.48	79.72	63.83	62.28	97.16	67.74
	Mel-Spectrogram(Augmented)	88.59	81.08	57.63	65.83	96.32	53.69
	MFCC	88.04	68.73	37.46	42.97	94.5	35.39
XResNet-50	Mel-Spectrogram	92.93	87.63	72.02	77.63	97.64	68.95
	Mel-Spectrogram(Augmented)	83.15	80.21	51.09	63.88	96.54	47.1
	MFCC	87.5	75.11	50.32	54.67	95.56	50.03

Table 6.2: Experimental results for 6 class classification using 1cycle approach on ICBHI dataset

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	Mel-Spectrogram	91.85	73.59	60.08	58	89.19	63.8
	Mel-Spectrogram(Augmented)	91.22	85.04	67.39	73.21	96.87	63.48
	MFCC	93.48	88.63	73.98	80.88	96.39	72.41
ResNet-34	Mel-Spectrogram	94.57	83.15	72.67	72.18	94.12	74.81
	Mel-Spectrogram(Augmented)	89.19	88.07	68.72	80	96.13	63.22
	MFCC	92.39	75.18	60.02	59.7	90.67	63.04
ResNet-50	Mel-Spectrogram	93.48	79.58	62.45	64.14	95.01	68.63
	Mel-Spectrogram(Augmented)	91.22	88.13	75.07	81.03	95.23	71.95
	MFCC	92.39	76.4	68.89	64.78	88.02	75.98
ResNet-101	Mel-Spectrogram	94.57	90.42	77.89	82.78	98.06	74.07
	Mel-Spectrogram(Augmented)	91.89	84.13	72.11	75.98	92.28	69.05
	MFCC	91.3	85.56	68.23	75.5	95.62	65.66
VGG16	Mel-Spectrogram	92.93	84.87	70.21	73.57	96.16	67.55
	Mel-Spectrogram(Augmented)	90.54	85.91	66.98	75.21	96.6	64.06
	MFCC	92.93	83.82	70.67	74.1	93.54	70.84
VGG19	Mel-Spectrogram	94.02	84.55	77.11	76.53	92.58	77.75
	Mel-Spectrogram(Augmented)	92.57	81.85	71.13	71.2	92.5	71.61
	MFCC	92.93	73.05	60.62	57.89	88.22	70.04
AlexNet	Mel-Spectrogram	91.3	79.39	63.5	65.86	92.93	62.14
	Mel-Spectrogram(Augmented)	90.54	92.61	74.98	88.59	96.63	67.78
	MFCC	89.67	71.85	56.44	56.65	87.04	63.56
SqueezeNet	Mel-Spectrogram	89.67	78.14	61.77	65.24	91.04	59.18
	Mel-Spectrogram(Augmented)	78.38	74.77	50.54	60.51	89.04	47.97
	MFCC	91.3	79.01	64.8	66.38	91.63	63.61
DenseNet-121	Mel-Spectrogram	91.3	80	66.68	68.4	91.61	65.3
	Mel-Spectrogram(Augmented)	82.43	73.55	52.67	59.79	87.3	49.7
	MFCC	92.93	84.55	73.67	75.59	93.51	72.12
XResNet-50	Mel-Spectrogram	88.59	83.54	63.11	72.45	94.64	59.22
	Mel-Spectrogram(Augmented)	90.54	84.65	65.47	72.69	96.61	61.9
	MFCC	92.39	83.56	73.12	76.43	90.68	71.75

Table 6.3: Experimental results for 3 class classification using fine-tune training on ICBHI dataset

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	Mel-Spectrogram	93.48	84.67	71.47	74.3	95.04	70.1
	Mel-Spectrogram(Augmented)	88.51	90.59	72.71	85.3	95.87	66.34
	MFCC	90.76	72.85	56.23	55.57	90.12	57.91
ResNet-34	Mel-Spectrogram	93.48	77.86	66.96	64.67	91.06	71.21
	Mel-Spectrogram(Augmented)	90.54	81.93	66.14	70.43	93.44	64.11
	MFCC	91.3	80	66.68	68.4	91.61	65.3
ResNet-50	Mel-Spectrogram	94.57	83.49	75.58	74.19	92.78	77.21
	Mel-Spectrogram(Augmented)	91.89	81.01	66.45	68.16	93.87	67.12
	MFCC	94.02	80.61	72.99	69.95	91.26	76.77
ResNet-101	Mel-Spectrogram	94.57	84.93	72.19	73.15	96.71	86.75
	Mel-Spectrogram(Augmented)	87.84	82.76	66.9	74.7	90.82	62.14
	MFCC	90.76	85.46	67.59	74.24	96.68	64.29
VGG16	Mel-Spectrogram	92.39	82.62	67.93	71.88	93.37	70.54
	Mel-Spectrogram(Augmented)	91.22	86.74	72.41	78.25	95.23	70.52
	MFCC	88.04	77.67	59.54	66.2	89.15	57
VGG19	Mel-Spectrogram	92.39	76.99	63.28	63.29	90.69	65.66
	Mel-Spectrogram(Augmented)	90.54	88.7	70.89	80.77	96.63	65.19
	MFCC	88.04	69.63	47.99	51.48	87.78	45.71
AlexNet	Mel-Spectrogram	88.04	71.18	53.68	54.54	87.81	52.92
	Mel-Spectrogram(Augmented)	82.43	83.05	58.84	72.39	93.71	54.33
	MFCC	91.85	76.81	69.91	67.11	86.52	73.39
SqueezeNet	Mel-Spectrogram	91.3	77.68	66.12	66.38	88.98	66.03
	Mel-Spectrogram(Augmented)	72.97	72.46	45.96	56.2	88.71	44.88
	MFCC	91.85	82.08	72.2	73.69	90.48	70.91
DenseNet-121	Mel-Spectrogram	91.85	78.46	63.43	65.1	91.83	64.64
	Mel-Spectrogram(Augmented)	89.19	80.02	62.35	67.14	92.91	59.69
	MFCC	92.39	78.73	68.43	66.8	90.67	71.06
XResNet-50	Mel-Spectrogram	91.3	75.76	55.91	57.27	94.24	55.37
	Mel-Spectrogram(Augmented)	92.57	92.09	78.31	86.84	97.33	72.69
	MFCC	89.67	86.56	69.08	79.44	93.69	63.85

Table 6.4: Experimental results for 3 class classification using 1cycle approach on ICBHI dataset

Model	**Feature Task**	**Accuracy (%)**	**ROC-AUC (%)**	**F1 score (%)**	**Recall(%)**	**Specificity (%)**	**Precision(%)**
ResNet-18	MFCC	84.23	76.53	73.61	71.06	71.06	78.53
	Mel-Spectrogram	81.85	82.03	68.76	66.57	66.57	73.71
	Mel-Spectrogram (Augmented)	74.66	75.46	65.57	67.16	67.16	64.71
ResNet-34	MFCC	82.89	75.04	67.51	64.65	64.65	79.35
	Mel-Spectrogram	84.59	81.08	73.06	70.04	70.04	79.95
	Mel-Spectrogram (Augmented)	76.37	79.29	69.18	72.28	72.28	67.9
ResNet-50	MFCC	78.19	70.58	65.88	64.98	64.98	67.21
	Mel-Spectrogram	80.48	74.14	70.29	69.72	69.72	70.96
	Mel-Spectrogram (Augmented)	71.23	70.04	62.96	65.55	65.55	62.27
ResNet-101	MFCC	80.54	76.44	66.72	64.81	64.81	71.29
	Mel-Spectrogram	85.96	81.51	75.84	72.64	72.64	82.4
	Mel-Spectrogram (Augmented)	82.19	80.73	74.27	74.84	74.84	73.77
VGG16	MFCC	83.22	79.83	67.87	64.87	64.87	80.77
	Mel-Spectrogram	84.59	80.73	72.18	68.89	68.89	81.06
	Mel-Spectrogram (Augmented)	75	77.71	66.82	69.1	69.1	65.77
VGG19	MFCC	81.88	78.57	69.02	66.78	66.78	74.12
	Mel-Spectrogram	82.88	79	69.35	66.65	66.65	76.7
	Mel-Spectrogram (Augmented)	79.45	80.21	71.53	73.09	73.09	70.47
AlexNet	MFCC	81.05	77.85	67.24	65.28	65.28	71.7
	Mel-Spectrogram	82.19	76.46	70.88	69.09	69.09	73.91
	Mel-Spectrogram (Augmented)	55.48	51.71	48.42	50.9	50.9	50.63
SqueezeNet	MFCC	73.83	68.05	61.2	61.08	61.08	61.33
	Mel-Spectrogram	83.22	80.72	70.2	67.44	67.44	77.3
	Mel-Spectrogram(Augmented)	66.78	66.16	59.89	63.86	63.86	60.06
DenseNet-121	MFCC	81.88	77.67	69.9	67.89	67.89	73.8
	Mel-Spectrogram	82.19	78.91	70.03	67.94	67.94	74.17
	Mel-Spectrogram (Augmented)	77.05	78.34	70.08	73.29	73.29	68.72
XResNet-50	MFCC	79.87	71.34	57.58	57.17	57.17	72.74
	Mel-Spectrogram	82.88	81.04	73.78	72.97	72.97	74.74
	Mel-Spectrogram (Augmented)	69.18	74.71	62.66	67.12	67.12	62.48

Table 6.5: Experimental results of fine-tune training with Coswara dataset

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	MFCC	77.05	73.32	64.19	63.51	63.51	65.16
	Mel-Spectrogram	84.59	83.72	74.62	72.34	72.34	78.48
	Mel-Spectrogram (Augmented)	66.44	61.61	57.94	60.24	60.24	57.87
ResNet-34	MFCC	75.34	73.02	62.7	62.42	62.42	63.03
	Mel-Spectrogram	83.22	82.54	73.81	72.62	72.62	75.39
	Mel-Spectrogram (Augmented)	72.82	74.97	65.26	68.2	68.2	64.37
ResNet-50	MFCC	81.16	76.9	66.55	64.4	64.4	72.55
	Mel-Spectrogram	83.56	74.06	71.49	68.81	68.81	77.5
	Mel-Spectrogram (Augmented)	70.81	71.68	64.07	68.02	68.02	63.53
ResNet-101	MFCC	85.62	78.27	74.66	71.27	71.27	82.52
	Mel-Spectrogram	82.88	75.45	71.18	68.95	68.95	75.53
	Mel-Spectrogram (Augmented)	75.5	73.59	66.09	67.14	67.14	65.39
VGG16	MFCC	80.82	78.18	67.73	65.91	65.91	71.44
	Mel-Spectrogram	80.48	75.56	69.92	69.14	69.14	70.9
	Mel-Spectrogram (Augmented)	74.5	76.7	65.53	67.05	67.05	64.69
VGG19	MFCC	81.51	77.54	70.95	69.8	69.8	72.55
	Mel-Spectrogram	80.48	71.87	66.91	65.12	65.12	70.77
	Mel-Spectrogram (Augmented)	81.88	77.21	72.83	72.33	72.33	73.39
AlexNet	MFCC	75.4	70.32	60.29	59.63	59.63	61.5
	Mel-Spectrogram	79.11	70.56	60.3	59.07	59.07	67.69
	Mel-Spectrogram (Augmented)	66.11	66.91	58.59	61.69	61.69	58.7
SqueezeNet	MFCC	80.82	72.34	66.74	64.76	64.76	71.58
	Mel-Spectrogram	80.45	71.38	68.93	67.17	67.17	72.39
	Mel-Spectrogram (Augmented)	73.49	71.46	66.12	69.18	69.18	65.15
DenseNet-121	MFCC	77.05	78.41	67.72	68.69	68.69	67.02
	Mel-Spectrogram	79.7	72.82	68.63	67.25	67.25	70.96
	Mel-Spectrogram (Augmented)	76.17	80.74	69.3	72.56	72.56	68.03
XResNet-50	MFCC	80.82	75.89	70.26	69.36	69.36	71.43
	Mel-Spectrogram	78.95	73.59	69.51	69.07	69.07	70.02
	Mel-Spectrogram (Augmented)	80.2	78.62	69.78	69.04	69.04	70.71

Table 6.6: Experimental results of 1cycle training with Coswara dataset

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	MFCC	74.42	58.11	58.17	57.69	57.69	63.17
	Mel-Spectrogram	75.31	73.08	60.62	59.7	59.7	65.32
	Mel-Spectrogram (Augmented)	59.83	62.69	54.91	57.67	57.67	55.93
ResNet-34	MFCC	74.88	65.46	61.42	60.46	60.46	64.78
	Mel-Spectrogram	71.13	64.69	60.02	59.67	59.67	60.54
	Mel-Spectrogram (Augmented)	68.2	67.28	59.46	59.93	59.93	59.18
ResNet-50	MFCC	76.28	69.62	63.87	62.62	62.62	67.42
	Mel-Spectrogram	68.62	60.7	57.06	56.89	56.89	57.31
	Mel-Spectrogram (Augmented)	65.27	65.46	56.77	57.43	57.43	56.58
ResNet-101	MFCC	71.16	68.92	63	63.52	63.52	62.63
	Mel-Spectrogram	74.48	62.37	59.28	58.58	58.58	63.56
	Mel-Spectrogram (Augmented)	65.27	60	57.54	58.53	58.53	57.35
VGG16	MFCC	73.49	63.54	58.21	57.68	57.68	61.71
	Mel-Spectrogram	76.15	64.23	59.22	58.59	58.59	67.25
	Mel-Spectrogram (Augmented)	72.38	64.7	58.87	58.3	58.3	60.77
VGG19	MFCC	72.56	68.94	61.83	61.37	61.37	62.52
	Mel-Spectrogram	74.9	64.67	62.06	61.08	61.08	64.95
	Mel-Spectrogram (Augmented)	65.69	68.35	60.45	63.24	63.24	60.52
AlexNet	MFCC	74.61	68.1	62.4	61.5	61.5	64.48
	Mel-Spectrogram	72.8	68.44	54.33	54.7	54.7	58.81
	Mel-Spectrogram (Augmented)	67.78	63.07	59.52	60.21	60.21	59.19
SqueezeNet	MFCC	79.53	72.44	65.54	63.57	63.57	75.73
	Mel-Spectrogram	75.73	68.03	58.88	58.31	58.31	66.12
	Mel-Spectrogram (Augmented)	65.27	61.45	57.16	57.98	57.98	56.97
DenseNet-121	MFCC	75.81	75.17	63.46	62.31	62.31	66.61
	Mel-Spectrogram	72.38	68.09	61.03	60.51	60.51	61.95
	Mel-Spectrogram (Augmented)	67.36	65.35	61.24	63.25	63.25	60.93
XResNet-50	MFCC	76.28	73.06	63.87	62.62	62.62	67.42
	Mel-Spectrogram	73.64	68.17	60.46	59.69	59.69	62.82
	Mel-Spectrogram (Augmented)	75.31	71.68	59.27	58.59	58.59	65.15

Table 6.7: Experimental results of fine-tune training with COUGHVID dataset

Model	Feature Task	Accuracy (%)	ROC-AUC (%)	F1 score (%)	Recall(%)	Specificity (%)	Precision(%)
ResNet-18	MFCC	67.91	60.25	57.08	57.03	57.03	57.12
	Mel-Spectrogram	73.22	71.72	61.21	60.52	60.52	62.72
	Mel-Spectrogram (Augmented)	67.78	70.1	59.12	59.66	59.66	58.84
ResNet-34	MFCC	71.16	64.37	57.13	56.75	56.75	58.72
	Mel-Spectrogram	74.9	63.22	61.49	60.53	60.53	64.8
	Mel-Spectrogram (Augmented)	63.6	61.4	57.27	59.08	59.08	57.37
ResNet-50	MFCC	73.95	67.1	65.38	65.38	65.38	65.38
	Mel-Spectrogram	75.73	68.1	63.8	62.75	62.75	66.55
	Mel-Spectrogram (Augmented)	51.46	48.7	47.05	49.32	49.32	49.48
ResNet-101	MFCC	69.77	61.97	56.09	55.81	55.81	57.09
	Mel-Spectrogram	74.48	67.96	63.27	62.46	62.46	64.83
	Mel-Spectrogram (Augmented)	62.76	61.55	57.51	60.18	60.18	57.99
VGG16	MFCC	78.14	63.87	60.36	59.56	59.56	74.69
	Mel-Spectrogram	72.8	63.96	55.89	55.81	55.81	59.67
	Mel-Spectrogram (Augmented)	59	60.69	54.52	57.67	57.67	55.87
VGG19	MFCC	74.88	57.52	60.05	59.23	59.23	64.46
	Mel-Spectrogram	71.13	58.18	57.89	57.46	57.46	59.18
	Mel-Spectrogram (Augmented)	62.76	63.85	57.22	59.63	59.63	57.62
AlexNet	MFCC	74.61	62.93	58.61	58.02	58.02	63.29
	Mel-Spectrogram	73.22	61.2	52.89	53.87	53.87	58.7
	Mel-Spectrogram (Augmented)	66.11	63.51	58.92	60.2	60.2	58.67
SqueezeNet	MFCC	75.81	69.2	63.46	62.31	62.31	66.61
	Mel-Spectrogram	74.9	64.55	58.93	58.31	58.31	64.23
	Mel-Spectrogram (Augmented)	67.36	65.46	59.18	59.93	59.93	58.87
DenseNet-121	MFCC	73.49	65.33	59.62	58.91	58.91	62.3
	Mel-Spectrogram	70.71	62.18	57.57	57.18	57.18	58.68
	Mel-Spectrogram (Augmented)	56.9	60.04	52.33	55.16	55.16	53.94
XResNet-50	MFCC	80	69.36	70.03	68.18	68.18	74.01
	Mel-Spectrogram	76.57	67.36	63.5	62.2	62.2	67.93
	Mel-Spectrogram (Augmented)	69.04	67.89	62.31	63.82	63.82	61.82

Table 6.8: Experimental results of 1cycle training with COUGHVID dataset

Chapter 7

Discussion

For respiratory disease classification, we attain the maximum 93.48% accuracy using the DenseNet-121 model in classifying respiratory diseases into six different disease categories; the model was trained with Mel-Spectrograms features using 1cycle training policy. We achieve the highest macro-average ROC-AUC and specificity score of 0.90 and 98.50% using the VGG19 model. On the other hand, the highest macro F1 value of 72.82% and recall score of 81.86% are slightly lower while employing the VGG19 model. The VGG19 model was trained using the balanced mini-batches and augmented Mel-Spectrograms along with fine-tuning training procedures. Moreover, the ResNet-18 model has the highest macro average precision score of 83.12%; the model was trained using the Mel-Spectrograms feature and 1cycle training policy. The low macro-average f1 score, recall score, and precision scores are due to the fact that the ICBHI dataset contains some classes with very little data, such as Bronchiolitis have only 13 samples which affects the overall macro average scores. As a result, the model's predictions are comparatively lower on those classes which restricts us from evaluating the model's performance appropriately. The bar plot in Figure 7.1 shows the individual Recall scores of six separate disease classes for the VGG19 model.

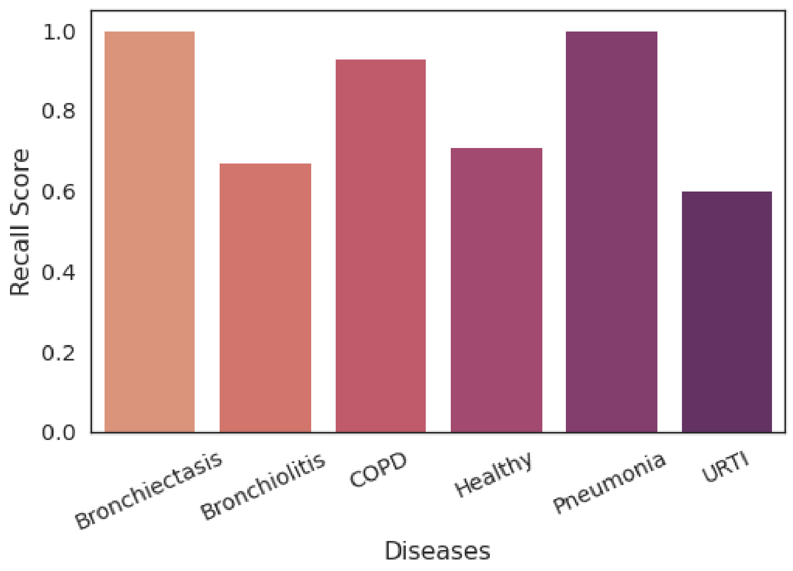

Figure 7.1: Recall scores of six separate disease classes

For respiratory disease categorization, based on the severity of the disease, we acquire the highest accuracy of 94.57% using the ResNet-34 and ResNet-101 which were trained with fine-tune training procedure. Moreover, the same accuracy was acquired using ResNet-50 and ResNet-101 architectures which were trained with 1cycle policy. All of these models were trained with Mel-Spectrogram features. Also, we achieved the highest ROC-AUC and macro average recall score of 0.92 and 88.59%, respectively, using the AlexNet architecture trained with augmented Mel-spectrogram in balanced mini-batches with the fine-tuning training procedure. Additionally, we achieve the highest macro average f1 scores of 78.31% using the XResNet-50 architecture; the model was trained with augmented Mel-Spectrograms using the 1cycle training policy. However, the ResNet-101 architecture has the highest macro average specificity and precision scores of 98.06% and 86.75% trained using Mel-Spectrograms with fine-tuning and 1cycle training policy, respectively. Our result for categorizing respiratory disease into three categories is better than the first sub-task of classifying the disease into separate classes, indicating that the model can categorize the disease more effectively than it can individually identify it.

Figure 7.2 and 7.3 contrasts fine-tuning and 1cycle training techniques. Both training approaches produce nearly equal results when employed with the MFCC feature. Also, using the balanced mini-batch training with augmented Mel-Spectrogram in-

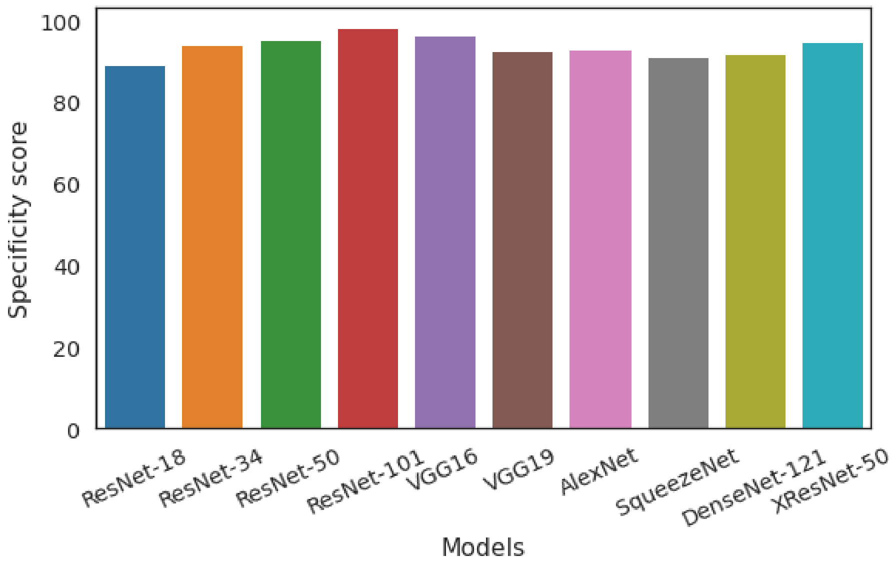

Figure 7.2: Specificity score with fine-tune training procedure

creased the average recall for all models up to 5% in our experiments. The Figure 7.4 and 7.5 shows the comparison of two types of Mel-Spectrogram feature input.

For COVID-19 infected and healthy patients' cough classification using the Coswara dataset, we acquired the maximum accuracy and F1 score of 85.96% and 75.83%, respectively, using the ResNet-101 model trained using Mel-Spectrograms with the fine-tune training procedure. However, using the ResNet-101 model trained with augmented spectrograms with fine-tuning, we attained the highest recall and specificity score of 74.84%. The highest precision score is 82.52%, which was similarly

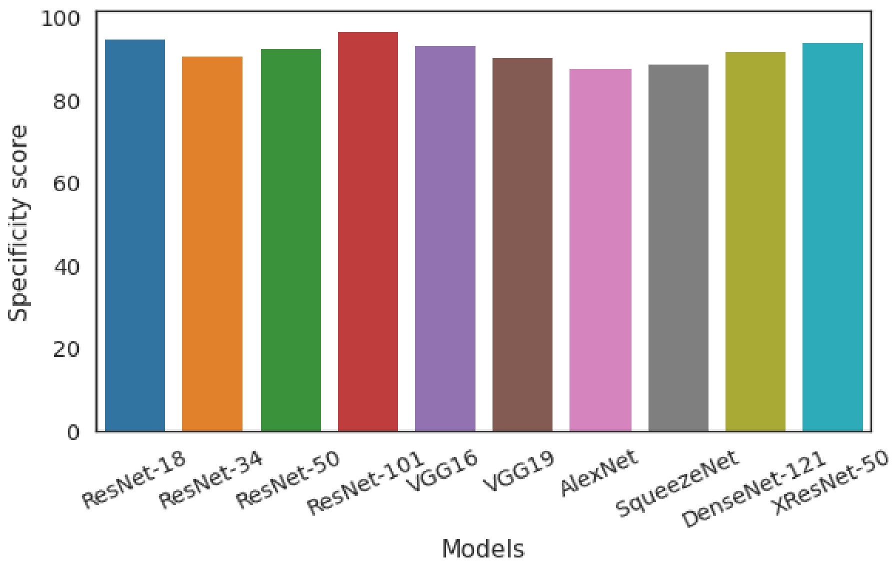

Figure 7.3: Specificity score with 1cycle training procedure

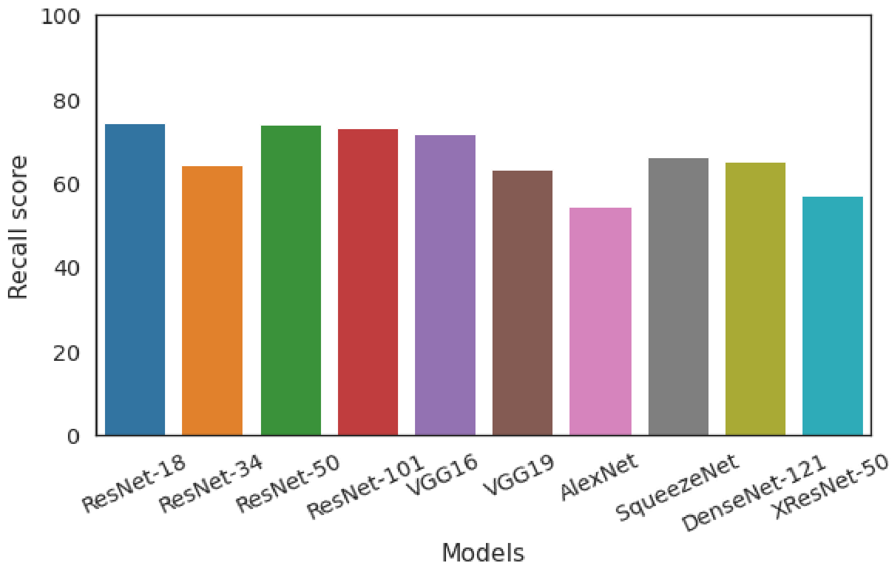

Figure 7.4: Recall score for Mel-Spectrogram input without augmentation

achieved with the ResNet-101 model, but this time employing MFCC features with the 1cycle policy. In addition, utilizing a ResNet-18 model trained with 1cycle policy using Mel-Spectrograms, we find the highest ROC-AUC score of 0.84. For this task, we are using a crowdsourced cough dataset with a significant chance of incorrect disease categorization during collection since the labels are self-provided by the participants. Such mislabeling could be one of the explanations for the model's slightly lower performance compared to our respiratory disease classification task, where we employed the ICBHI dataset with expert-diagnosed disease labels. Figure 7.6 shows the confusion matrix of the ResNet-101 model trained with augmented Mel-Spectrograms. Here, the macro average sensitivity and specificity score is 74.84%.

Utilizing the COUGVID dataset for COVID-19 infected and healthy patients' cough classification, we achieve the highest accuracy and F1 scores of 80% and 70.03% re-

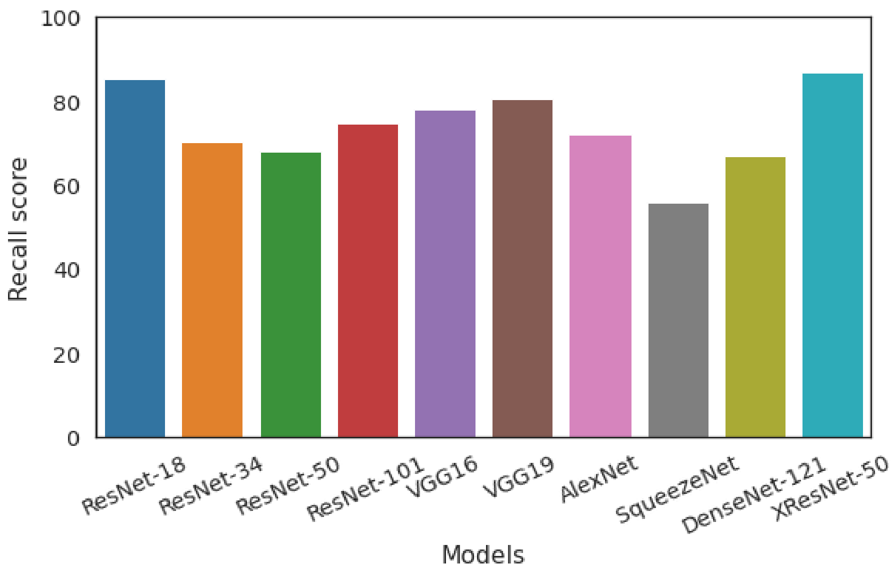

Figure 7.5: Recall score for Mel-Spectrogram input with augmentation

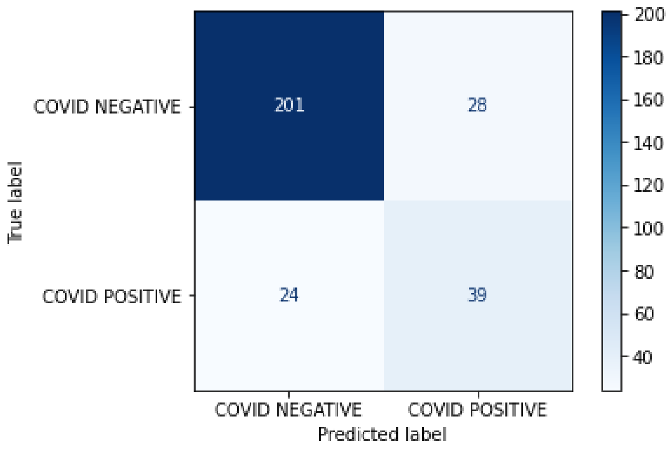

Figure 7.6: Confusion Matrix of ResNet-101 model

spectively using the XResNet-50 model on the MFCC feature with 1cycle training policy. We also find the highest Recall and Specificity score of 68.19% using the same model. However, we achieve the highest Roc-Auc score of 0.75 using the DenseNet-121 model and the highest precision score of 75.73% with the SqueezeNet model, employing MFCC features with fine-tuning training in both models. The score is lower than the Coswara datasets score, which might be due to a variation in cough sounds in the two datasets. Some recordings in the COUGHVID dataset, for example, contain just one cough since participants were not instructed how many times they should cough during the data collection method; whereas, participants in the Coswara data collection procedure were asked to cough at least three times. Furthermore, because the COUGHVID dataset contains a variety of cough sounds and the files are not explicitly labeled by the cough type, we could not choose only the heavy-cough files, as we did with the Coswara dataset.

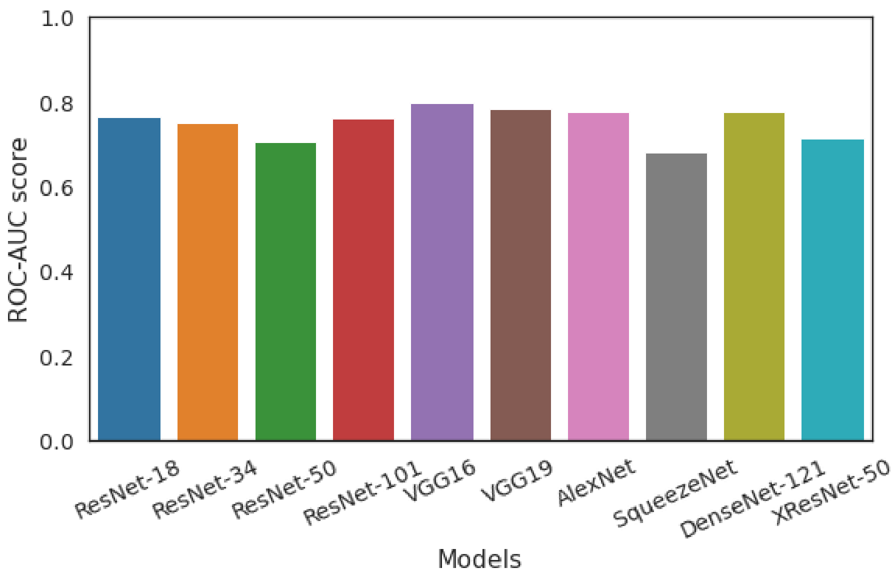

Figure 7.7: ROC-AUC score for Coswara dataset

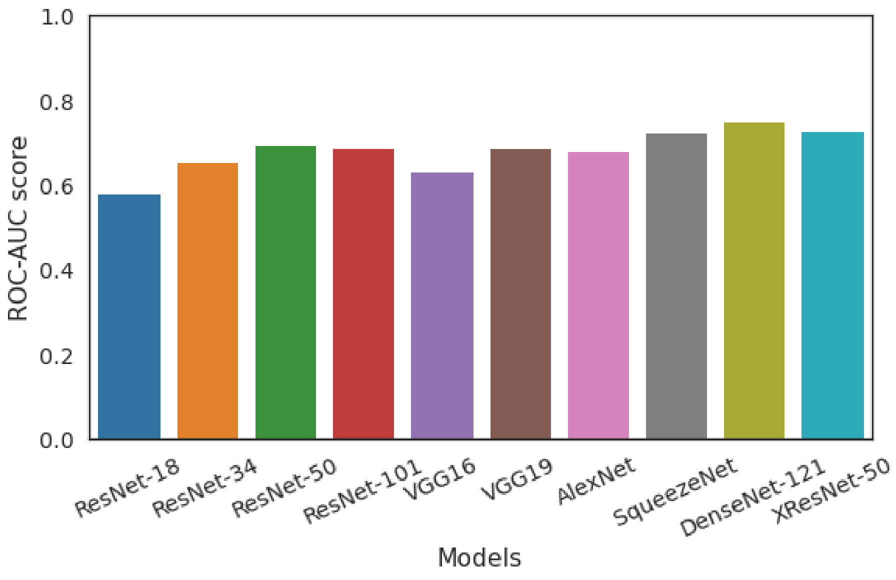

Figure 7.8: ROC-AUC score for COUGHVID dataset

Here the above Figure 7.7 and 7.8 contrast the ROC-AUC score for MFCC feature models for the two datasets, Coswara and COUGHVID.

Chapter 8

Comparison

In this paper, we have adapted numerous models and training methods to get the best outcomes. We chose the model with the highest performance to compare it to earlier studies. However, because many implementation details are unavailable, this comparison is not considered a head-to-head comparison.

8.1 Respiratory Disease Classification

The comparison of Respiratory Disease Classification is shown in tables 8.1 and 8.2, where we have compared our best results with previous research work. It shows that most previous studies employed machine learning techniques, but we used a variety of deep CNN models. Despite the data being imbalanced, most previous research has avoided using augmentation techniques. In contrast, we used a variety of augmentation strategies to balance the data in our study. Most of the studies used six or more class classification procedures; however, we also used a three-class classification technique along with six-class classification.

Previous work	Data	Augmentation	Task	Model	Feature used
Chambres,G. et al [26]	ICBHI	No	Wheeze, Crackle based 2 class	machine learning	Low-level features
Fraiwan,L. et al[55]	King Abdullah University Hospital+ICBHI	No	6 class	Boosted Decision Tree	Shannon entropy, logarithmic energy entropy, and spectrogram-based spectral entropy
Kok, X. H. et al[34]	ICBHI	No	6 class	RUSBoost	Feature from Wilcoxon Rank Sum test and mRMR algorithm
Wu,L. et al[51]	ICBHI	No	6 class	combining random forest classifier and Empirical Mode Decomposition (EMD)	Mean of multiple features
Basu, V. et al[41]	ICBHI	Yes	6 class	Custom Deep CNN	MFCC
Nguyen, T. et al[62]	ICBHI	Yes	3 class	Pre-Trained ResNet-101 with Stochastic Normalization	Logmel Spectrogram
Nguyen, T. et al[62]	ICBHI	Yes	2 class	Pre-Trained ResNet-101 with Stochastic Normalization	Logmel Spectrogram
Ours	ICBHI	Yes	6 class	CNN	Mel-Spectrogram, MFCC
Ours	ICBHI	Yes	3 class	CNN	Mel-Spectrogram, MFCC

Table 8.1: Theoretical comparison of Respiratory Disease Classification with previous works

Previous work	Task	Accuracy (%)	Sensitivity (%)	Specificity (%)
Chambres,G. et al [26]	Wheeze, Crackle based 2 class	85	-	-
Fraiwan,L. et al[55]	6 class	98.27	95.28	98.90
Kok, X. H. et al[34]	6 class	87.1	86.8	93.6
Wu,L. et al[51]	6 class	88	87	97
Basu, V. et al[41]	6 class	95.67	95.65	-
Nguyen, T. et al[62]	3 class	-	91.47	100
Nguyen, T. et al[62]	2 class	-	96.41	100
Ours (VGG19)	6 class	91.30	81.86	98.50
Ours (ResNet-101)	3 class	94.57	82.78	98.06

Table 8.2: Experimental result comparison of Respiratory Disease Classification with previous works

8.2 COVID-19 Cough Classification

The result of cough classification in COVID-19 patients is shown in tables 8.3 and 8.4. Although we have used both Coswara and Coughvid dataset, Coswara has produced better results. While employing the models, most of the previous researchers have used various amount of data with class imbalance.

Previous Work	**Task**	**Model**	**Feature**	**Augmentation**
Bagad, P. et al[40]	COVID-19 cough classification	CNN: ResNet18	Mel-Spectrogram	Yes
Sharma, M. et al[66]	COVID-19 cough classification	CNN: ResNet18	Mel-Spectrogram	Yes
Mohammed, E. A. et al[60]	COVID-19 cough classification	CNN Ensemble	Multiple features	Yes
Chaudhari,G. et al[44]	COVID-19 cough classification	CNN Ensemble	Multiple features	No
Brown, C. et al[43]	COVID-19 cough classification	CNN	Multiple features	Yes
Mahanta, H. K.[58]	COVID-19 cough classification	CNN	MFCC	Yes
Coppock, H. et al[54]	COVID-19 cough classification	CNN	Log Spectrogram	No
Han, J. et al[57]	COVID-19 cough classification	CNN	Mel-Spectrogram	No
OUR	COVID-19 cough classification	CNN	Mel-Spectrogram, MFCC	Yes

Table 8.3: Theoretical comparison of COVID-19 cough Classification with previous works

Previous Work	**Data**	**Accuracy(%)**	**ROC-AUC**
Bagad, P. et al[40]	3,117 total cough samples	-	0.72
Sharma, M. et al[66]	1394 Positive and 2866 Negative samples	-	0.79
Mohammed, E. A. et al[60]	638 Positive and 8248 Negative samples	77	0.77
Chaudhari,G. et al[44]	539 Positive and 2810 Negative samples	-	0.77
Brown, C. et al[43]	54 Positive and 34 Negative samples	-	0.82
Mahanta, H. K.[58]	75 Positive and 965 Negative samples	-	0.87
Coppock, H. et al[54]	54 Positive and 32 Negative samples	-	0.846
Han, J. et al[57]	9 Positive and 1964 Negative samples	-	0.71
Ours (ResNet-18)	429 Positive and 1527 Negative samples	84.59	0.84

Table 8.4: Experimental result comparison of COVID-19 Cough Classification with previous works

Chapter 9

Conclusion

In this research, we present deep-learning based techniques for detecting respiratory diseases and distinguishing COVID-19 infected and healthy patients' cough from audio recordings. We achieve 93.48% accuracy and 0.90 ROC-AUC scores to classify six respiratory conditions from respiratory sound recordings, and 94.57% accuracy and 0.93 ROC-AUC scores to classify respiratory diseases in three categories. Moreover, we achieve 85.95% accuracy and a ROC-AUC score of 0.84 in classifying cough recordings of COVID-19 infected and healthy patients; our result is comparable to several previous research works. Also, we compare the performance of two audio features, MFCC and Mel-Spectrogram performances with ten different convolutional neural network architectures. Furthermore, we experiment with various training procedures such as transfer learning, 1cycle policy, and balanced mini-batch training and compare the performance of each method. Our work shows that AI can be utilized as an alternative screening and preliminary warning tool for potential patients. A key challenge is the limited availability of human respiratory sounds data, as the ICBHI dataset contains only 920 audio recordings. Also, the lack of ground truth in cough sound data limits the scope because we had to rely on patients' self-reports with potential misdiagnosis to train and evaluate the models. To improve our research we will generate our own dataset under the supervision of a medical professional and train it in a modified model architecture to get more efficient results. In addition, we will aim to acquire larger and more representative data with clinical ground truth in the future, to improve our models' performance.

Bibliography

[1] S. . Davis and P. . Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. DOI: 10.1109/tassp.1980.1163420.

[2] K. R. Smith, “Fuel combustion, air pollution exposure, and health: The situation in developing countries,” *Annual Review of Energy and the Environment*, vol. 18, no. 1, pp. 529–566, 1993. DOI: 10.1146/annurev.eg.18.110193.002525.

[3] X. Huang, A. Acero, and H.-W. Hon, “Spoken language processing: A guide to theory, algorithm, and system development,” Jan. 2001.

[4] R. L. Moedomo, M. S. Mardiyanto, M. Ahmad, B. Alisjahbana, and T. Djatmiko, “The breath sound analysis for diseases diagnosis and stress measurement,” in *2012 International Conference on System Engineering and Technology (ICSET)*, 2012, pp. 1–6. DOI: 10.1109/ICSEngT.2012.6339358.

[5] A. . Badnjevic, M. . Cifrek, and D. . Koruga, “Integrated software suite for diagnosis of respiratory diseases,” *Eurocon 2013*, 2013. DOI: 10.1109/eurocon.2013.6625037.

[6] U. K. Barua, S. K. Saha, D. K. Ghosh, and M. M. K. Ruble, “Epidemiological study on bronchial asthma at shaheed suhrawardy medical college hospital, dhaka,” *Journal of Shaheed Suhrawardy Medical College*, vol. 5, no. 2, pp. 77–80, 2013. DOI: 10.3329/jssmc.v5i2.20759.

[7] K. . Simonyan. (Sep. 4, 2014). “Very deep convolutional networks for large-scale image recognition,” [Online]. Available: https://arxiv.org/abs/1409.1556.

[8] Y. . Amrulloh, U. . Abeyratne, V. . Swarnkar, and R. . Triasih, “Cough sound analysis for pneumonia and asthma classification in pediatric population,” *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*, 2015. DOI: 10.1109/isms.2015.41.

[9] D. Chamberlain, J. Mofor, R. Fletcher, and R. Kodgule, “Mobile stethoscope and signal processing algorithms for pulmonary screening and diagnostics,” in *2015 IEEE Global Humanitarian Technology Conference (GHTC)*, 2015, pp. 385–392. DOI: 10.1109/GHTC.2015.7344001.

[10] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[11] ICDDRB, *Pneumonia and other respiratory diseases*, https://www.icddrb.org/news-and-events/press-corner/media-resources/pneumonia-and-other-respiratory-diseases, Accessed: 2022-01-15, 2015.

[12] D. . Chamberlain, R. . Kodgule, D. . Ganelin, V. . Miglani, and R. R. Fletcher, "Application of semi-supervised deep learning to lung sound analysis," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016. DOI: 10.1109/embc.2016.7590823.

[13] D. B. Chamberlain, R. . Kodgule, and R. R. Fletcher, "A mobile platform for automated screening of asthma and chronic obstructive pulmonary disease," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016. DOI: 10.1109/embc.2016.7591897.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size*, 2016. arXiv: 1602.07360 [cs.CV].

[15] R. X. A. Pramono, S. A. Imtiaz, and E. . Rodriguez-Villegas, "A cough-based algorithm for automatic diagnosis of pertussis," *PLOS ONE*, vol. 11, no. 9, D. F. Hozbor, Ed., e0162128, 2016. DOI: 10.1371/journal.pone.0162128.

[16] (Feb. 4, 2016). "Torch | training and investigating residual nets," [Online]. Available: http://torch.ch/blog/2016/02/04/resnets.html.

[17] Forum of International Respiratory Societies, Forum of International Respiratory Societies, Forum of International Respiratory Societies Staff, European Respiratory Society, and European Respiratory Society Staff, *The Global Impact of Respiratory Disease.* European Respiratory Society, 2017.

[18] S. I. Khan, V. Ahmed, and N. P. Jawarkar, "Application of signal processing techniques for preliminary detection of adventitious lung sounds in paediatric population using electronic stethoscope," in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, 2017, pp. 335–338. DOI: 10.1109/ICBDACI.2017.8070859.

[19] A. . Krizhevsky, I. . Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. DOI: 10.1145/3065386.

[20] Y. . Liu, Y. . Lin, X. . Zhang, Z. . Wang, Y. . Gao, G. . Chen, and H. . Xiong, "Classifying respiratory sounds using electronic stethoscope," *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017. DOI: 10.1109/uic-atc.2017.8397496.

[21] B. M. Rocha, D. . Filos, L. . Mendes, I. . Vogiatzis, E. . Perantoni, E. . Kaimakamis, P. . Natsiavas, A. . Oliveira, C. . Jácome, A. . Marques, R. P. Paiva, I. . Chouvarda, P. . Carvalho, and N. . Maglaveras, "A respiratory sound database for the development of automated classification," *Precision Medicine Powered by pHealth and Connected Health*, pp. 33–37, 2017. DOI: 10.1007/978-981-10-7419-6_6.

[22] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017. arXiv: 1609.04747 [cs.LG].

[23] ——, *Transfer Learning - Machine Learning's Next Frontier*, http://ruder.io/transfer-learning/, 2017.

[24] L. . N. . Smith. (Jun. 3, 2017). "Cyclical learning rates for training neural networks," [Online]. Available: https://arxiv.org/abs/1506.01186.

[25] M. A. Azam, A. Shahzadi, A. Khalid, S. M. Anwar, and U. Naeem, "Smartphone based human breath analysis from respiratory sounds," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 445–448. DOI: 10.1109/EMBC.2018.8512452.

[26] G. . Chambres, P. . Hanna, and M. . Desainte-Catherine, "Automatic detection of patient with respiratory diseases using lung sound analysis," *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, 2018. DOI: 10.1109/cbmi.2018.8516489.

[27] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, *Bag of tricks for image classification with convolutional neural networks*, 2018. arXiv: 1812.01187 [cs.CV].

[28] J. Howard *et al.*, *Fastai*, https://github.com/fastai/fastai, 2018.

[29] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, 2018. arXiv: 1608.06993 [cs.CV].

[30] N. . Olvera-Montes, B. . Reyes, S. . Charleston-Villalobos, R. . Gonzalez-Camarena, M. . MejiaAvila, G. . Dorantes-Mendez, S. . Reulecke, and T. A. Aljama-Corrales, "Detection of respiratory crackle sounds via an android smartphone-based system," *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018. DOI: 10.1109/embc.2018.8512672.

[31] D. . Sarkar. (Nov. 17, 2018). "A comprehensive hands-on guide to transfer learning with real-world applications in deep learning," [Online]. Available: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.

[32] R. Shimizu, K. Asako, H. Ojima, S. Morinaga, M. Hamada, and T. Kuroda, "Balanced mini-batch training for imbalanced image data classification with neural network," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, 2018, pp. 27–30. DOI: 10.1109/AI4I.2018.8665709.

[33] L. N. Smith and N. Topin, *Super-convergence: Very fast training of neural networks using large learning rates*, 2018. arXiv: 1708.07120 [cs.LG].

[34] X. H. Kok, S. . Anas Imtiaz, and E. . Rodriguez-Villegas, "A novel method for automatic identification of respiratory disease from acoustic recordings," *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019. DOI: 10.1109/embc.2019.8857154.

[35] Y. Ma, X. Xu, Q. Yu, Y. Zhang, Y. Li, J. Zhao, and G. Wang, "Lungbrn: A smart digital stethoscope for detecting respiratory disease using bi-resnet deep learning algorithm," Oct. 2019. DOI: 10.1109/BIOCAS.2019.8919021.

[36] K. . Mavropalias. (Feb. 19, 2019). "Understanding fastai's fit$_one_cyclemethod$," [Online]. Available: https://iconof.com/1cycle-learning-rate-policy/ (visited on 01/15/2022).

[37] D. . S. . Park. (Apr. 18, 2019). "Specaugment: A simple data augmentation method for automatic...," [Online]. Available: https://arxiv.org/abs/1904.08779.

[38] (Apr. 22, 2019). "Specaugment: A new data augmentation method for automatic speech recognition," [Online]. Available: https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html.

[39] M. . Trivedi and E. . Denton, "Asthma in children and adults—what are the differences and what can they tell us about asthma?" *Frontiers in Pediatrics*, vol. 7, 2019. DOI: 10.3389/fped.2019.00256.

[40] P. . Bagad. (Sep. 17, 2020). "Cough against covid: Evidence of covid-19 signature in cough sounds," [Online]. Available: https://arxiv.org/abs/2009.08790.

[41] V. . Basu and S. . Rana, "Respiratory diseases recognition through respiratory sound with the help of deep neural network," *2020 4th International Conference on Computational Intelligence and Networks (CINE)*, 2020. DOI: 10.1109/cine48825.2020.234388.

[42] J. . Bielby, S. . Kuhn, S. . Colreavy-Donnelly, F. . Caraffini, S. . O'Connor, and Z. A. Anastassi, "Identifying parkinson's disease through the classification of audio recording data," *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020. DOI: 10.1109/cec48606.2020.9185915.

[43] C. . Brown, J. . Chauhan, A. . Grammenos, J. . Han, A. . Hasthanasombat, D. . Spathis, T. . Xia, P. . Cicuta, and C. . Mascolo, "Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2020. DOI: 10.1145/3394486.3412865.

[44] G. . Chaudhari. (Nov. 26, 2020). "Virufy: Global applicability of crowdsourced and clinical datasets...," [Online]. Available: https://arxiv.org/abs/2011.13320#:%5C%7E:text=This%5C%20study%5C%20demonstrates%5C%20that%5C%20crowdsourced,%5C%25%5C%20(75.2%5C%25%5C%2D78.3%5C%25)..

[45] H. A. Coultas Blum, L. G. Scart, and R. Bracco, *Fastaudio*, Aug. 2020. [Online]. Available: https://github.com/fastaudio/fastaudio.

[46] F. . Demir, A. M. Ismael, and A. . Sengur, "Classification of lung sounds with cnn model using parallel pooling structure," *IEEE Access*, vol. 8, pp. 105 376–105 383, 2020. DOI: 10.1109/access.2020.3000111.

[47] C. Li, H. Du, and B. Zhu, *Classification of lung sounds using cnn-attention*, EasyChair Preprint no. 4356, 2020.

[48] P. D. Muthusamy, K. Sundaraj, and N. Abd Manap, "Computerized acoustical techniques for respiratory flow-sound analysis: A systematic review," *Artificial Intelligence Review*, vol. 53, Jun. 2020. DOI: 10.1007/s10462-019-09769-6.

[49] V. . Ramesh, K. . Vatanparvar, E. . Nemati, V. . Nathan, M. M. Rahman, and J. . Kuang, "Coughgan: Generating synthetic coughs that improve respiratory disease classification*," *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020. DOI: 10.1109/embc44109.2020.9175597.

[50] N. Sharma, P. Krishnan, R. Kumar, S. Ramoji, S. R. Chetupalli, N. R., P. K. Ghosh, and S. Ganapathy, "Coswara — a database of breathing, cough, and voice sounds for covid-19 diagnosis," *Interspeech 2020*, Aug. 2020. DOI: 10.21437/interspeech.2020-2768. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2768.

[51] L. Wu and L. Li, "Investigating into segmentation methods for diagnosis of respiratory diseases using adventitious respiratory sounds," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 768–771. DOI: 10.1109/EMBC44109.2020.9175783.

[52] W. . Chen, Q. . Sun, X. . Chen, G. . Xie, H. . Wu, and C. . Xu, "Deep learning methods for heart sounds classification: A systematic review," *Entropy*, vol. 23, no. 6, p. 667, 2021. DOI: 10.3390/e23060667.

[53] J. S. Chorba, A. M. Shapiro, L. . Le, J. . Maidens, J. . Prince, S. . Pham, M. M. Kanzawa, D. N. Barbosa, C. . Currie, C. . Brooks, B. E. White, A. . Huskin, J. . Paek, J. . Geocaris, D. . Elnathan, R. . Ronquillo, R. . Kim, Z. H. Alam, V. S. Mahadevan, S. G. Fuller, G. W. Stalker, S. A. Bravo, D. . Jean, J. J. Lee, M. . Gjergjindreaj, C. G. Mihos, S. T. Forman, S. . Venkatraman, P. M. McCarthy, and J. D. Thomas, "Deep learning algorithm for automated cardiac murmur detection via a digital stethoscope platform," *Journal of the American Heart Association*, 2021. DOI: 10.1161/jaha.120.019905.

[54] H. . Coppock, A. . Gaskell, P. . Tzirakis, A. . Baird, L. . Jones, and B. . Schuller, "End-to-end convolutional neural network enables covid-19 detection from breath and cough audio: A pilot study," *BMJ Innovations*, vol. 7, no. 2, pp. 356–362, 2021. DOI: 10.1136/bmjinnov-2021-000668.

[55] L. . Fraiwan, O. . Hassanin, M. . Fraiwan, B. . Khassawneh, A. M. Ibnian, and M. . Alkhodari, "Automatic identification of respiratory diseases from stethoscopic lung sound signals using ensemble classifiers," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 1, pp. 1–14, 2021. DOI: 10.1016/j.bbe.2020.11.003.

[56] Geddes. (Nov. 26, 2021). "What do we know about the new b.1.1.529 coronavirus variant and should we be worried?" [Online]. Available: https://www.gavi.org/vaccineswork/what-we-know-about-new-b11529-coronavirus-variant-so-far?gclid=CjwKCAiAtdGNBhAmEiwAWxGcUkxG5P23X1g5TuA33gIjdyqoBFaV1 NxoCQN4QAvD_BwE (visited on 01/15/2022).

[57] J. Han, T. Xia, D. Spathis, E. Bondareva, C. Brown, J. Chauhan, T. Dang, A. Grammenos, A. Hasthanasombat, A. Floto, P. Cicuta, and C. Mascolo, *Sounds of covid-19: Exploring realistic performance of audio-based digital testing*, 2021. arXiv: 2106.15523 [cs.SD].

[58] S. . K. . Mahanta. (Oct. 12, 2021). "Covid-19 diagnosis from cough acoustics using convnets and data...," [Online]. Available: https://arxiv.org/abs/2110.06123.

[59] Maraqa, DeNicola, Udeani, and Custodio. (Jun. 26, 2021). "What is the global prevalence of bronchiolitis?" [Online]. Available: https://www.medscape.com/answers/961963-36369/what-is-the-global-prevalence-of-bronchiolitis.

[60] E. . A. . Mohammed. (Jul. 28, 2021). “An ensemble learning approach to digital corona virus preliminary screening from cough sounds,” [Online]. Available: https://www.nature.com/articles/s41598-021-95042-2?error=cookies_not_supported&code=e09cf70f-8ecf-422f-939e-adc0b3152e8d.

[61] Z. . Mueller. (Mar. 16, 2021). “Methods for automating learning rate finders,” [Online]. Available: https://www.novetta.com/2021/03/learning-rate/.

[62] T. . Nguyen. (Aug. 4, 2021). “Lung sound classification using co-tuning and stochastic normalization,” [Online]. Available: https://arxiv.org/abs/2108.01991.

[63] L. Orlandic, T. Teijeiro, and D. Atienza, “The coughvid crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms,” *Scientific Data*, vol. 8, no. 1, May 2021, ISSN: 2052-4463. DOI: 10.1038/s41597-021-00937-4. [Online]. Available: http://dx.doi.org/10.1038/s41597-021-00937-4.

[64] (Jul. 20, 2021). “Pneumonia in children statistics,” [Online]. Available: https://data.unicef.org/topic/child-health/pneumonia/.

[65] L. . Roberts. (Dec. 13, 2021). “Understanding the mel spectrogram - analytics vidhya,” [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

[66] M. . Sharma. (Jun. 5, 2021). “Impact of data-splits on generalization: Identifying covid-19 from...,” [Online]. Available: https://arxiv.org/abs/2106.03851.

[67] Q. . Zhou, J. . Shan, W. . Ding, C. . Wang, S. . Yuan, F. . Sun, H. . Li, and B. . Fang, “Cough recognition based on mel-spectrogram and convolutional neural network,” *Frontiers in Robotics and AI*, vol. 8, 2021. DOI: 10.3389/frobt.2021.580080.