

University of South Dakota

**USD RED**

---

Dissertations and Theses

Theses, Dissertations, and Student Projects

---

2023

# **MENTORING DEEP LEARNING MODELS FOR MASS SCREENING WITH LIMITED DATA**

Suprim Nakarmi

Follow this and additional works at: <https://red.library.usd.edu/diss-thesis>



Part of the [Computer Sciences Commons](#)

---

**MENTORING DEEP LEARNING MODELS FOR MASS SCREENING WITH LIMITED  
DATA**

By

Suprim Nakarmi

B.E., Tribhuvan University, 2018

A Thesis Submitted in Partial Fulfillment of  
the Requirements for the Degree of Master of Science

---

Department of Computer Science

Master Of Science Program  
In the Graduate School  
The University of South Dakota  
December 2023

Copyright By:  
SUPRIM NAKARMI  
2023  
All Rights Reserved

## Committee Signature Page

The members of the Committee appointed to examine the Thesis of Suprim Nakarmi find it satisfactory and recommend that it be accepted.

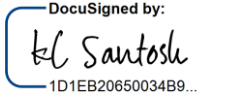
DocuSigned by:  
*KC Santosh*  
BBEF270EA578492...  
\_\_\_\_\_  
Chairperson

DocuSigned by:  
*Doug Goodman*  
3C0E646BD1404F3...  
\_\_\_\_\_

DocuSigned by:  
*Lu Bough*  
05454119D36C468...  
\_\_\_\_\_

## Abstract

Deep Learning (DL) has an extensively rich state-of-the-art literature in medical imaging analysis. However, it requires large amount of data to begin training. This limits its usage in tackling future epidemics, as one might need to wait for months and even years to collect fully annotated data, raising a fundamental question: is it possible to deploy AI-driven tool earlier in epidemics to mass screen the infected cases? For such a context, human/Expert in the loop Machine Learning (ML), or Active Learning (AL), becomes imperative enabling machines to commence learning from the first day with minimum available labeled dataset. In an unsupervised learning, we develop pretrained DL models that autonomously refine themselves through iterative learning, with human experts intervening only when the model misclassifies and for a limited amount of data. We introduce a new terminology for this process, calling it mentoring. We validated this concept in the context of Covid-19 in three distinct datasets: Chest X-rays, Computed Tomography (CT) scans, and cough sounds, each consisting of 1364, 4714, and 10,000 images, respectively. The framework classifies the deep features of the data into two clusters (0/1: Covid-19/non-Covid-19). Our main goal is to strongly emphasize the potential use of AL in predicting diseases during future epidemics. With this framework, we achieved the AUC scores of 0.76, 0.99, and 0.94 on cough sound, Chest X-rays, and CT scans dataset using only 40%, 33%, and 30% of the annotated dataset, respectively. For reproducibility, the link of implementation is provided: <https://github.com/2ailab/Active-Learning>.

Thesis Advisor: 1D1EB20650034B9...

KC Santosh, Ph.D.

## **Acknowledgements**

I express my deepest appreciation to Dr. KC Santosh, my thesis advisor, whose expert guidance, and steadfast support played a crucial role in the successful completion of this thesis. Without his invaluable mentorship, this work would not have come to fruition. I also acknowledge the contributions of my esteemed thesis committee members, namely Dr. KC Santosh, Dr. Doug Goodman, and Dr. Lee Baugh. Their active engagement and assistance throughout this process have been greatly valued. Additionally, I extend heartfelt thanks to the entire Applied Artificial Intelligence Research Lab (2AI) 2022 - 2023 team for their invaluable aid and support during the journey of writing this thesis. Your collective efforts have been essential, and I am genuinely grateful for your participation in this academic undertaking.

## **Dedication**

I would like to dedicate this thesis to my parents Kishor Nakarmi and Bhawani Nakarmi.

## Table of Contents

Committee Signature Page.....	i
Abstract.....	ii
Acknowledgements.....	iii
Dedication.....	iv
Table of Contents.....	v
List of Tables.....	vii
List of Figures.....	viii
1. Introduction.....	1
1.1 Context and problem.....	1
1.2 Goal.....	2
1.3 Methodology.....	2
1.4 Outlines.....	2
2. Active learning- theory.....	4
2.1 Background.....	4
2.2 Membership query synthesis.....	5
2.3 Stream-based selective sampling.....	5
2.4 Pool-based scenario.....	5
2.5 Query strategies.....	6
2.6 Why use AL?.....	7
2.7 Related works.....	7
3. Active learning – implementation.....	10
3.1 Background.....	10
3.2 Deep learning models.....	11
3.3 Convolutional neural networks.....	11
3.4 Mentoring to DL models.....	12
3.5 Unsupervised learning/clustering.....	14
3.6 K-way n-shot learning.....	14
4. Experimental setup.....	16
4.1 Overview.....	16
4.2 Dataset.....	17



4.3	Evaluation and validation.....	20
5.	Results and analysis.....	23
5.1	Cough sound.....	26
5.2	CT scans .....	31
5.3	Chest X-rays.....	36
5.4	Comparison .....	41
6.	Conclusion.....	43
	References.....	45

## List of Tables

Table 1: Comparison of time taken and computational complexity for training with and without formation of subclusters.....	24
Table 2: Table showing the clustering and classification results when using Euclidean distance as distance measure for cough sound dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.....	28
Table 3: Table showing the clustering and classification results when using Manhattan distance as distance measure for cough sound dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.....	29
Table 4: Table showing the clustering and classification results when using Cosine distance as distance measure for cough sound dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.....	30
Table 5: Table showing the clustering and classification results when using Euclidean distance as distance measure for CT scan dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data. ....	33
Table 6: Table showing the clustering and classification results when using Manhattan distance as distance measure for CT scan dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data. ....	34
Table 7: Table showing the clustering and classification results when using Cosine distance as distance measure for CT scan dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data. ....	35
Table 8: Table showing the clustering and classification results when using Euclidean distance as distance measure for Chest X-ray dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.....	38
Table 9: Table showing the clustering and classification results when using Manhattan distance as distance measure for Chest X-ray dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.....	39
Table 10: Table showing the clustering and classification results when using Cosine distance as distance measure for Chest X-ray dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.....	40
Table 11: Comparison of the classification results for cough sound among previous works that used DL models and our AL framework. ....	41
Table 12: Comparison of the classification results for CT scans among previous works that used DL and our AL framework.....	42
Table 13: Comparison of the classification results for CXRs among previous works that used DL and our AL framework.....	42

# List of Figures

Figure 1: A general example showing the working of AL algorithm.....	5
Figure 2: Illustration of three AL scenarios to query instances to oracle. Note that learner can be any ML/DL system. ....	6
Figure 3: Schema of the proposed human-in-the-loop (or AL) framework (used with permission). .....	13
Figure 4: Analogy of the proposed framework. A child is mentored until he/she reaches 18 years of age after which he/she is independent (used with permission). ....	14
Figure 5: Figure depicting the subclusters and formation of new subcluster when the model commits a mistake.....	17
Figure 6: Example images of Covid-19 positive, pneumonia, and normal CXRs used in this thesis. .....	18
Figure 7: Example images of Covid-19 positive, pneumonia, and normal CT scans used in this thesis. ....	19
Figure 8: Example images of covid-19 positive and non-covid-19 cases Mel-spectrograms used in this thesis.....	20
Figure 9: Illustration of cumulative average corrected counts on mentored cases made by the Expert when the model misclassifies the data sample. a) Record of the raw cumulative average count on four mentored cases on three datasets, and b) Rate of mistakes on four mentored cases on three datasets. Note: the figure was plotted when experimenting without forming subclusters...	24
Figure 10: Illustration of accuracy against the value of 'K' in K Nearest Neighbors to determine the optimum value of K. a) Accuracy against value of K in cough sound, b) Accuracy against value of K in CT scans, and c) Accuracy against value of K in X-rays. Note: the figure was plotted by experimenting without forming subclusters.....	25

# Chapter 1

## 1. Introduction

---

*Summary:* This chapter introduces the thesis, motivation of the work, and provides the outline of the thesis.

Key topics: Need, motivation, goal, and contribution.

---

### 1.1 Context and problem

Epidemics are a rapid or unexpected increase in disease cases in a certain community or geographical region [1], mostly a consequence of infectious disease outbreaks spreading from person to person through the air, contact, animal-to-person contact, environments, or other media. History has demonstrated enough evidence of occurrence of epidemics at different timelines. For instance, HIV/AIDS, discovered in 1981, spreader across central Africa and around the world infecting approximately 85.6 million people with over 40.4 million death cases; similarly, Ebola was discovered in 1976, Zika in 1950, E. coli in 1982, and Covid-19 in 2019. This induces an optimal statement “Future epidemics are inevitable”. Considering this situation, a pertinent question arises: are there readily available tools that can be promptly employed to monitor cases right from the onset of epidemics? Recently, with the advancement in the Deep Learning (DL) algorithms and myriad of available dataset, healthcare sectors have benefitted in drug discovery [2], genomics [3], personalized medicine[4], and medical imaging analysis[5] is not an exception. However, all the pre-existing DL models require a large amount of labeled dataset to work effectively [6][7]. With this, another question emerges: how much labeled data is enough to start training? A more critical question would be: what if we do not have dataset? Do we wait for months, or even years to have enough dataset to train our models. This is infeasible especially at the time of epidemics; we cannot wait for people to die until we have enough data to commence training.

Human-in-the-loop Machine Learning (ML) or Active Learning (AL) can assist public healthcare workers to begin training from day one with the limited amount of labeled dataset. When it comes

to scarcity of annotated dataset, which is always the case in emergencies such as epidemics, AL is the must [8][9]. The concept of implementing AL possibly mitigates the spreading of disease by having a system ready that can begin training from the first day of epidemic. AL is a subfield of ML, where the machine learns from a limited amount of labeled dataset by having some role in selecting the data instance it wants to learn from. Leveraging AL, in this thesis, we present an unsupervised clustering framework that can commence training with a minimum possible labeled dataset and provide a proof that it is applicable for any type of medical image dataset.

## 1.2 Goal

The primary goal of thesis is about development of an Artificial Intelligence (AI) driven tool that can be used for mass screening of any medical images to possibly mitigate the risk from upcoming epidemics. The risk is mitigated by having a system ready early so that it can assist public health workers to identify new patients curbing the spreading of disease in large populations. For this, a stream-based AL framework was developed and validated in three distinct datasets: Cough sound, Chest X-rays, and CT scans. All the datasets were based on Covid-19 as it is the most recent pandemic. Not to be confused, the framework developed in this thesis is only a proof-of-concept of how AL can possibly assist public health workers in mass screening at the time of epidemics.

## 1.3 Methodology

AL or human-in-the-loop ML is not a new concept and is ubiquitously used in the domain of DL having a rich literature. AL can be implemented in various scenarios depending on the ways the dataset is available, and selection of data samples to label can be applied using various strategies. Among the vast available methods, we selected stream-based AL scenario where labeling data sample occurs only when the model commits a mistake and for a limited amount of data. Stream-based AL scenario was selected with an assumption that continuous flow of unlabeled data occurs at the time of epidemic and data samples are selected (queried) when the model commits a mistake. In an unsupervised framework, the model then clusters the data into two clusters (0/1: non-Covid-19/ Covid-19 positive cases).

## 1.4 Outlines

Rest of the thesis is structured as follows:

Chapter 2: This chapter provides background of AL, its types, relevant related works, and discusses why to use AL.

Chapter 3: The chapter dives deep into the implementation of the proposed AL framework, explaining how mentoring occurs in DL models; how we integrate unsupervised clustering, and k ways n shot learning.

Chapter 4: Comprises experimental setup with description of dataset, evaluation and validation technique being used.

Chapter 5: This chapter deals with results on each dataset and comparison with the previous works that trained DL models using completely labeled dataset.

Chapter 6: This chapter concludes the thesis and mentions future work applicable for this thesis.

## Chapter 2

### 2. Active learning- theory

---

*Summary: This chapter provides the thorough theoretical explanation of AL, its types, querying strategies, and when to use it.*

Key topics: Active learning, active learning scenarios, and querying strategies.

---

#### 2.1 Background

AL, also known as Human/Expert-in-the-loop ML is a subfield of ML that operates under the assumption that the model can actively query (for example, unlabeled instance) an oracle (or a human/Expert annotator) for labels of carefully selected examples. Alternatively, ML techniques that aim to improve model performance by selecting the most informative examples from the unlabeled dataset for training. The technique is applicable to any kind of model such as neural network, Support Vector Machine (SVM), etc. Unlike passive learning, that refers to collecting copious of labeled dataset before training, AL focuses on actively acquiring labels for the most relevant instances, thereby reducing the labeling effort, cost, and computational burden involved. Figure 1 illustrate a general example of the working of AL. Any algorithm that implements AL iteratively trains the model, initially with the available labeled data, followed by model selecting the difficult examples or most informative data instance from unlabeled pool to query human Expert. After the oracle annotates the data, it is added to the existing labeled dataset to retrain the model. This process is repeated until certain convergence criteria is met. Usually, the convergence (also called stopping criterion) is pre-defined: the number of iterations, a target performance level, or other convergence indicator [10][11]. Based on the ways the AL techniques are applied, the branch of AL is divided into three scenarios (or setting): membership query synthesis, stream-based selective sampling, pool-based sample. Each scenario is discussed below.

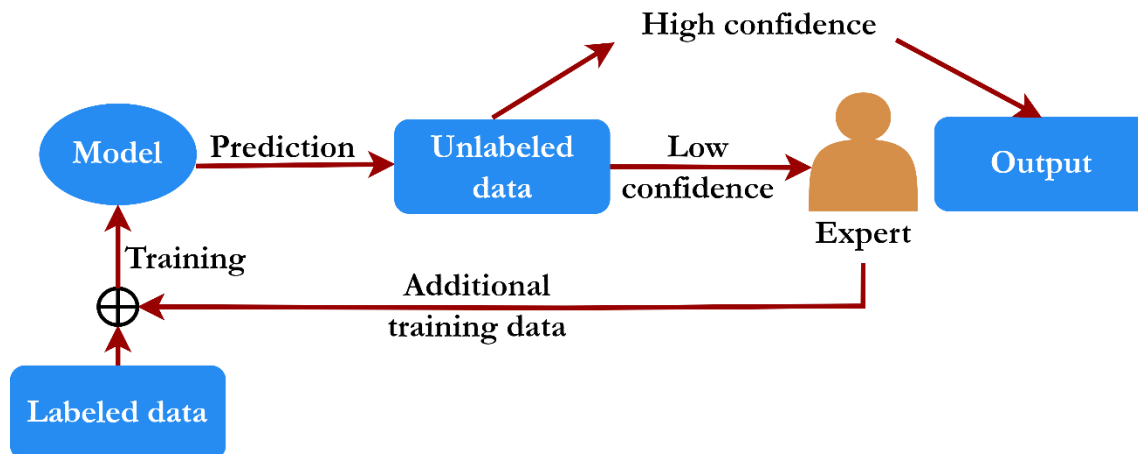


Figure 1: A general example showing the working of AL algorithm.

## 2.2 Membership query synthesis

Membership query synthesis [12] is an early active learning scenario where the learner actively seeks "label membership" (synthesis data) for unlabeled data instances within the defined input space. The assumption is made of learner having knowledge of the input space, including feature dimensions and ranges. While generating queries efficiently is practical for finite domains, labeling arbitrary instances, as in Natural Language Processing (NLP), can be challenging due to potential lack of coherence. Despite this challenge, this approach reduces the cost of experimental materials compared to running the least expensive experiment alone. Figure 2 illustrates the membership-based AL scenario.

## 2.3 Stream-based selective sampling

Stream-based selective sampling [13] is designed for handling streaming data, where instances arrive individually or in small batches. The learner selects unlabeled instances one by one or in batches from the input source and decides whether to query or discard each instance. The assumption is that obtaining an unlabeled instance is either cost-free or inexpensive. The stream-based scenario has been applied in real-world tasks like part-of-speech tagging, sensor scheduling, and learning ranking functions for information retrieval. Figure 2 illustrates the stream-based AL scenario.

## 2.4 Pool-based scenario

In real-world learning scenarios, obtaining extensive sets of unlabeled data can be challenging. To address this, pool-based [14] is adopted, combining a small, labeled dataset ('L') with a large pool of unlabeled data ('U'). Queries for labeling are chosen from this dataset, often assumed to be closed, indicating it is either static or doesn't change, although this is not always a prerequisite. The selection of queries typically follows a greedy approach based on a utility measure, especially when 'U' is extensive. Pool-based sampling has been extensively examined in various ML domains,



including text classification, information extraction, image and video classification and retrieval, speech recognition, and medical diagnosis. Figure 2 illustrates the pool-based AL scenario, which is more common in application papers, while the stream-based approach may be preferred in scenarios with memory or processing constraints, such as with mobile devices or large datasets.

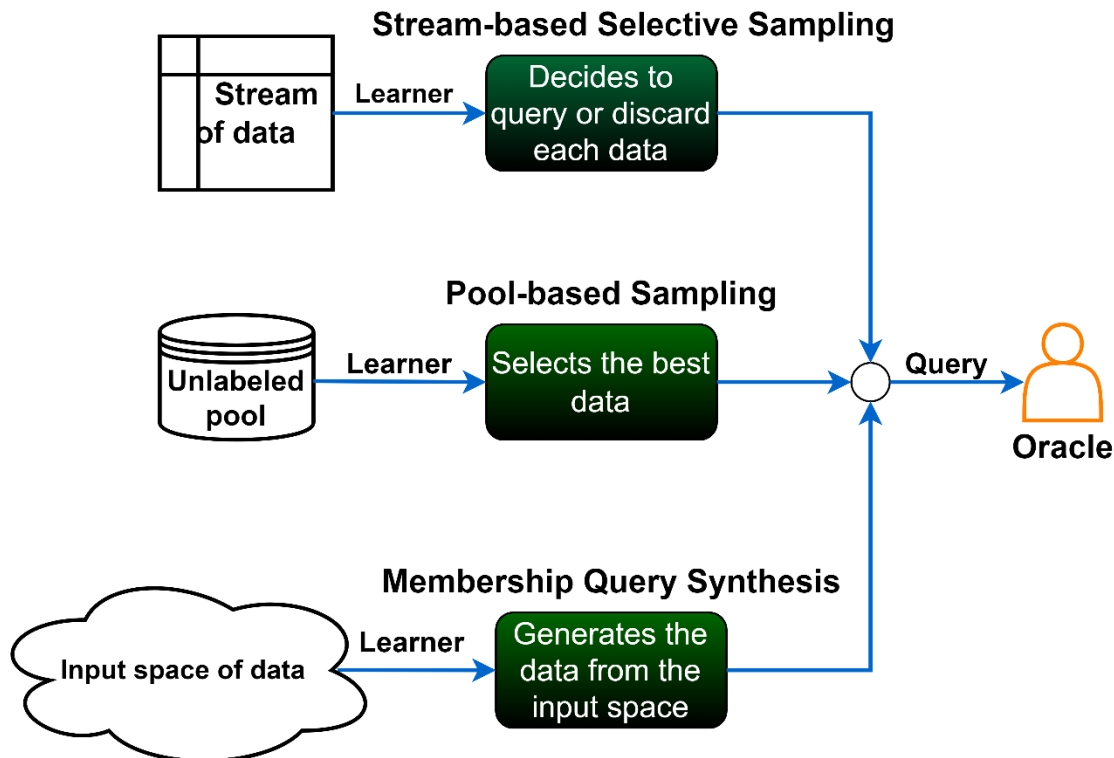


Figure 2: Illustration of three AL scenarios to query instances to oracle. Note that learner can be any ML/DL system.

## 2.5 Query strategies

In all AL scenarios, assessing the informativeness of unlabeled instances is a fundamental aspect, which can be accomplished through various methods discussed in the literature. Development of novel querying strategies has been an active research area in AL. In this section, classical querying strategies are listed along with some of the popular custom query strategies. Some examples of classical querying technique are uncertainty sampling, query by disagreement, query by committee, estimated error reduction, variance reduction and fisher information ratio, density-weighted methods, etc. One example of the querying techniques is provided below:

### Uncertainty sampling

Uncertainty sampling [14] entails selecting instances that the learner finds most uncertain. The selection is based on the idea that these instances are likely to offer the most valuable information for addressing the problem, aiming to prevent the querying of redundant instances. It further consists of three types, each described below.

Least Confident: Selects the instance that is predicted with lowest confidence. Given by the equation,  $a^* = \operatorname{argmax}_a P\left(\frac{\hat{b}}{a}\right)$ , where  $a^*$  is the selected instance, and  $\operatorname{argmax}_a P(b/a)$  is the instance prediction with the greatest posterior probability, ‘a’ is the available dataset.

Margin-based sampling: It relies on the margin of the output, given by  $a^* = \operatorname{argmin}_a \left[ P\left(\frac{\hat{b}_1}{a}\right) - P\left(\frac{\hat{b}_2}{a}\right) \right]$  where  $\hat{b}_1$  and  $\hat{b}_2$  represents the first and second most likely predictions under the model.

Entropy-based sampling: One of the most widely used and general uncertainty sampling strategies commonly employs entropy, typically represented as H, as the measure of utility:  $a^* = \operatorname{argmax}_a - \sum_{i=0}^n P\left(\frac{b_i}{a_i}\right) * \log\left(P\left(\frac{b_i}{a_i}\right)\right)$ , where b ranges over all labeling of a.

Other techniques developed either to optimize the querying and labeling cost or to work for specific task at hand (classification, prediction, etc.) are: Variational Adversarial Active Learning (VAAL), Query strategy for Convolutional neural networks (CNN), Bayesian Active Learning by Disagreements (BALD), Wasserstein Adversarial AL (WAAL), Generative Adversarial Active Learning (GAAL), Cost-Effective Active learning (CEAL), etc. Note that explaining various custom built querying techniques are out of the scope of this thesis.

## 2.6 Why use AL?

AL or human-in-the-loop ML is gaining popularity both in academic and industrial usages. It is especially suitable for cases characterized by an abundance of data but has high costs associated with labeling - both time and money - encountered frequently in complex supervised learning tasks or when the annotated dataset is scarce. In such cases, implementing AL will offer a solution by iteratively training models and selecting the most informative unlabeled dataset to label extending the available training set.

## 2.7 Related works

In this section, initially, we discuss AL techniques used in general medical imaging analysis task, followed by the DL models used for Covid-19 mass screening, after that we describe AL techniques used in Covid-19 mass screening.

Usage of AL have increased in the domain of medical imaging analysis as hiring Expert (or doctors) for annotations is expensive and time consuming. Hao et al. [15] used uncertainty sampling on Magnetic Resonance Imaging (MRI) for brain tumor classification achieving 82.89% AUC value. Similarly, Nguyen et al. [16] used AL, specifically uncertainty estimation for cancer cases classification using text-based reports achieving 98.25% sensitivity and 96.14% specificity. Gorriz et al. [17] implemented uncertainty sampling on microscopy images to segment Melanoma reporting the dice score of 74%. Likewise, Liu et al. [18] used uncertainty sampling for nodule detection on Low-Dose Computed Tomography (LDCT) images with a sensitivity of 92.1%. Jin et al. [19] segmented skin lesion using density-based querying strategy achieving 93.4% dice

score. Qureshi et al. [20] implemented uncertainty sampling on retinography images to classify retinal fundus image achieving 93% F-measure, and 98% accuracy. Shao et al. [21] classified nucleus using pair wise uncertainty sampling on colon pathology images with 79.2% F1-score. All the works were implemented in a pool-based scenario, which is popular in medical imaging analysis domain. In contrast, other researchers used AL on a custom scenario, for example, Park et al. [22] implemented semi-supervised based reinforced AL for nodule segmentation achieving 80.2% F1-score. Wu et al.[23] achieved 86.6% accuracy using sample diversity and predicted loss querying strategy for Covid-19 classification on CT scan images. Iglesias et al. [24] used query by disagreement using two models to recognize and segment CT scan images achieving 67% dice score. Hoa et al. [25] classified cough sound dataset using Fisher Linear Discriminant (FLD) achieving 91.60% accuracy.

Similarly, a rich literature is available for the classification of Covid-19 positive from other (healthy or general pneumonia) using DL models [26] [27] [28] [29] [30] [31]. Al-Waisy et al. used two pre-trained models: ResNet34 and high-resolution network to train chest x-rays recording the accuracy, sensitivity, specificity, precision, and F1-score of 99.99%, 99.98%, 100%, 100%, and 99.99%, respectively. The authors used histogram equalization and Butterworth bandpass filter were used to preprocess the X-ray images. Ismael et al. [32] used five CNN models and Support Vector Machines (SVM) classifier with four kernel functions to classify healthy and Covid-19 positive cases reporting the best result of 94.7% accuracy. Rajaraman et al. [33] employed iterative pruning of DL model that ensembles to detect pulmonary manifestations of Covid-19 in chest X-rays. By leveraging modality-specific knowledge transfer, iterative model pruning, and ensemble learning, their approach achieved a significant improvement, yielding an accuracy of 99.01% and an area under the curve of 0.9972 in identifying Covid-19 findings on chest radiographs, showcasing its potential for swift adoption in Covid-19 screening. Brunese et al. [34] developed evidence-based method that consisted of three phases: the initial phase detects the presence of pneumonia in a chest X-ray, followed by a second phase distinguishing between Covid-19 and pneumonia, and the final step is dedicated to localizing areas in the X-ray indicative of Covid-19 presence. This approach achieved an average Covid-19 detection time of approximately 2.5 seconds and an average accuracy of 0.97. Okolo et al. [35] assessed the performance of eleven CNN architectures in classifying chest X-ray images, distinguishing among healthy individuals, those with Covid-19, and those with viral pneumonia. The authors explored three distinct modifications to adapt the established architectures for this task by incorporating additional layers, achieving the highest classification accuracy of 98.04% and the highest F1-score of 98.22% for the most effective configuration across all examined architectures on a dataset comprising authentic chest X-ray images. Similarly, Khurana et al. [36] utilized four DL architectures identify Covid-19 in CT scan images, with ResNet50 achieving the highest accuracy of 98.9%. Canayaz et al. [37] introduced two novel methods employing data driven approach for Covid-19 diagnosis in CT scans using ResNet50 and MobileNetV2 for deep feature extraction, which was classified using SVM and K Nearest Neighbor algorithms. They reported the best accuracy of 99.06% using ResNet50 and SVM.

Similarly, Subhalakshmi et al. [38] proposed a model using Inception V4 and VGGNet16 for feature extraction, and Gaussian Naïve Bayes classifier as the final classifier for Covid-19 identification on CT scans, while Zouch et al. [39] applied ResNet50 and VGG19 for Covid-19 detection, achieving accuracies of 99.35% and 96.77%, respectively, for both models. Similarly, Pahar et al. [40] used Coswara dataset (cough sound) using ResNet50 achieving the AUC of 97.6% and sensitivity of 93%. Similarly, Meister et al. [41] used Recurrent Neural Networks (RNN) on cough sound and achieved 77.5% AUC and 77% accuracy. Arup et al. [42] implemented Support Vector Machines (SVM) and achieved 98% AUC, 96% accuracy, and 96% sensitivity.

However, to the best of our knowledge, a dearth of AL based solutions that specifically focuses on epidemics and Covid-19 exists in the literature. At the time of writing this thesis, only two work have contributed by implementing AL. Wu et al. [23] presents COVID-AL, a novel weakly-supervised deep AL framework that combines lung region segmentation with a 2D U-Net and Covid-19 diagnosis using a hybrid AL strategy that considers sample diversity and predicted loss, demonstrating superior performance compared to state-of-the-art approaches with over 95% accuracy using only 30% of the labeled data. Hussain et al. [43] proposed AL approach addresses the issue of unreliable machine-generated annotations by re-weighting samples based on the similarity of their gradient directions to expert-annotated data and the gradient magnitude of the deep model's last layer, demonstrating improved segmentation performance for pneumonia infection on clinical Covid-19 CT benchmark data achieving the dice score of 76.35%.

## Chapter 3

### 3. Active learning – implementation

---

*Summary: This chapter provides the detail explanation of each component of the proposed framework. This includes the DL models, specifically CNNs, introduction to mentoring the DL models, unsupervised clustering method, and k-ways n-shot learning.*

Key topics: Deep learning models, convolutional neural network, mentoring, unsupervised clustering, and k way n shot learning.

---

#### 3.1 Background

In the above section, we discussed AL methods implemented by other scholars in general medical imaging analysis task and to address the pressing issue of epidemic, specifically Covid-19. This section delineates our proposed framework and how we implemented such that training can commence from the first day of epidemic being declared. The framework was conceptualized by observing how doctors learn (or acknowledges) new diseases. No doctors in the world initially have much knowledge about an unknown disease; however, they can identify the reports as being abnormal and unforeseen. For instance, when Dr. Zhang Jixian, the first doctor in Wuhan, China to report Covid-19, received two patient’s Computed Tomography (CT) thorax images, she acknowledged they were different than common pneumonia – without knowing the disease, but was caused from single virus [44]. Later she confirmed that it was caused by infectious disease, after summoning the patients’ child with similar observation in the CT scan image. Likewise, our framework clusters the infected images (or abnormal cases) from normal ones so that the abnormal images are separated.

In essence, we introduce mentoring to the in-house DL models that enables training to begin early with a limited amount of labeled dataset. Mentoring simply means including a human Expert in the training loop while training (also called AL). More about mentoring is discussed in Section 3.4 after we describe DL models used in this study.

## 3.2 Deep learning models

DL is a subfield of ML that learns the representation of data using Artificial Neural Network (ANN) [45]. Inspired by the working of human brains, DL models can discover the complex hidden features from the data (raw or pre-processed) without needing the interference of humans. DL models accomplish this by having multiple hidden layers capable of representing non-linearity presented in the dataset. Each layer builds on top of the preceding layer creating a hierarchy (or stack) of features with initial layer representing simple features such as edges, and the next layer representing combinations of features from the previous layer to obtain more complex features, and so on. Higher the number of layers, the better the model is at capturing complex features; however, note that the abstractness of the model increases with layers [46]. Some of the popular DL models are CNNs, Recurrent Neural Networks (RNN), transformers, and Generative Adversarial Network (GAN). The primary advantage of DL models is that it does not need manual hand-crafted features for predictions, unlike Shallow Learning models that learn only from human crafted features (e.g., Support Vector Machines (SVM), K-Nearest Neighbors (KNN), etc.). It has been beating records in task related to object recognition, computer vision related task such as classification/detection, speech recognition, etc. However, one of the primary challenges of any DL models is the requirement of large amount dataset to learn the underlying features. Next, we'll discuss the DL models used in this thesis.

## 3.3 Convolutional neural networks

Convolutional Neural Networks (CNNs) are a special type of ANN designed to be used for analyzing visual data (e.g., images and videos). Composed of varieties of layers such as convolutional, pooling, and fully connected layers, CNNs have proven to be highly effective in various Computer Vision (CV) tasks including classification, object recognition/detection, and image segmentation tasks. Convolutional layers are considered a building block of CNN, where a small filter (or window) is defined to convolve over the input image – performing dot product with each set of pixel values (intensity) – to extract features such as edges, textures, etc. Similarly, a pooling layer down samples the output of convolutional layer to reduce the spatial dimension of the feature map. The feature map is obtained after an input data (images) is passed through the convolutional layer. After several convolutional and pooling layers, data is sent to fully connected layers – one or more – where each neuron is connected to all the succeeding neurons of the next layer in the network. Generally, convolutional layers capture local patterns such as edges, texture, etc., whereas fully connected layers capture global patterns such as arrangement of objects, etc. Having the combination of global and local features is advantageous in capturing features such that it is translation, rotation, and scaling invariant. Furthermore, CNNs have more advantages: local feature learning – learning low-level features, parameter sharing – reducing the total number of learnable parameters, transfer learning – suitable fine-tuning for a specific task with a smaller dataset, position invariant – can identify image regardless of its position in an image. Some of the popular CNN models are Residual Networks (ResNet), LeNet, Dense Convolutional Network

(DenseNet), InceptionNet, Visual Geometric Group (VGG), etc. We opted three popular CNNs model for this thesis: ResNet101, DenseNet169, and VGG16.

### **ResNet101**

ResNet101 is a CNN-based architecture belonging to the Residual Network family [47]. First introduced by Kaiming et al. [47], ResNet101 consists of 101 layers each having a residual learning block. As in general CNN, the blocks contain several convolutional layers, pooling layers, batch normalization, and Rectifier Linear Unit (ReLU). It also introduces skip connections (or identity networks) allowing the model to learn the difference between input and desired output, rather than learning to map the input directly to output. This allows training very DL networks by mitigating the vanishing gradient problem. The depth of the network enables learning more complex hidden features/representations of the data, setting impressive performance on various benchmarks. We selected this network because of its popularity in the domain of medical imaging analysis [48][11].

### **DenseNet169**

DenseNet169 is one of the many networks in Densely Connected Convolutional Networks family [50]. Huang et al. [50] Introduced this model in 2017, with distinctive densely connected blocks (or dense blocks), meaning each layer receives direct inputs from all other preceding layers. This enables feature reuse with efficient flow of information, mitigate gradient vanishing issues, and parameter efficiency, that directly improves the training efficiency and model performance. Like ResNet, this allows learning intricate features and train deeper network. DenseNet169 was selected for its capability to work with limited computational resources or relatively smaller dataset size.

### **VGG16**

VGG16 is a deep CNN architecture consisting of 16 layers developed by the Visual Geometry Group [51]. Characterized by its simplicity and uniformity, it gained widespread popularity for its outstanding performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. Simonyan et al. [51] introduced the network using convolutional, max-pooling, and fully connected layers. Note that all the convolutional layers in VGG network primarily use 3x3 filters with a stride of 1 that maintains a consistent field of receptive throughout the network. VGG16 contains a total of 16 layers, including convolutional and fully connected layers.

## **3.4 Mentoring to DL models**

The implementation of AL requires the integration of a data instance selection strategy, alternatively termed as an Expert's intervention within the training loop. The intervention can occur in two ways: a ML system requesting a query to an oracle (or Expert) for labeling complex data samples or a domain expert closely interacting with the system in the training loop by providing feedback [52][53]. In the thesis, we adopted the latter approach, where an expert continuously mentors the decisions made by the DL models and intervenes only when the model commits a mistake. For example, if the system misclassifies a Covid-19 positive case, the human expert corrects it. Mentoring involves including humans/experts in the training loop, where the

expert intervenes to improve performance until the model gains sufficient knowledge to make predictions independently. We present a special stream-based AL setting with a closed-loop Expert’s<sup>1</sup> feedback constantly mentoring DL models for a limited number of data samples. In a dynamic environment, e.g., epidemic, AL allows the exploitation of real-time data with expert mentoring until the system is ready to be independent, continuously updating the classifier and training data. In contrast to the conventional stream-based scenario, where the data are either queried or discarded, our approach incorporates even the dataset that does not necessitate querying into the training set (see Algorithm 1). This feasibility arises from the continuous guidance of an expert who mentors decisions for the dataset under consideration.

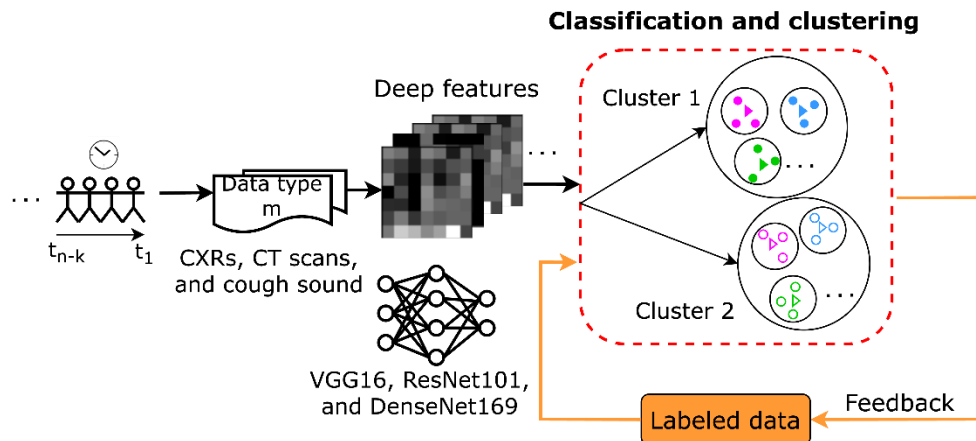


Figure 3: Schema of the proposed human-in-the-loop (or AL) framework (used with permission).

Figure 3 and Figure 4 illustrate the proposed framework and its analogy, respectively. The framework is akin to how children are mentored by their parents until the age of 18, after which they become independent. This does not mean that the child will never make mistake after 18 years, but they will be independent enough to take care of themselves [54]. Similarly, DL models are mentored until they are ready to make predictions on their own. The critical question is determining how many data samples require mentoring until the DL model can predict independently. Therefore, we conducted three case studies on three distinct datasets to investigate how much mentoring is necessary for the DL model’s decision-making to become independent. We used three pre-trained CNN models: VGG16 [51], ResNet101[47], and DenseNet169 [50], followed by unsupervised clustering using indices such as Dunn’s, Davies Bouldin, and Silhouette. The activation of the Expert’s intervention can be expressed as,

$$G(x) = \begin{cases} 1, & \text{if the model makes a mistake} \\ 0, & \text{otherwise,} \end{cases}$$

<sup>1</sup> Expert’s work is simulated in the code without employing a real Expert.



where  $G(x)$  gets activated only when the model makes a mistake. Details of the results on each data types i.e., Cough sound, CT scans, and X-rays, are described in Section 6, 7, and 8, respectively.



Figure 4: Analogy of the proposed framework. A child is mentored until he/she reaches 18 years of age after which he/she is independent (used with permission).

### 3.5 Unsupervised learning/clustering

We employed unsupervised clustering using an exclusive clustering algorithm- K-means clustering. Initially, this was applied to cluster healthy data from Covid-19 infected images, and subsequently, it was utilized to reduce the algorithmic complexity of the proposed framework through sub clustering. Euclidean, Manhattan, and Cosine distances were utilized to distinguish features. A supervised setting, involving mentoring, was employed to separate Covid-19 and healthy images, while an unsupervised setting was used to sub-cluster data instances within the formed cluster. For each class, the number of subclusters were first determined using an elbow method. The subcluster were extended for each test sample when the model misclassified the data, meaning a new subcluster consisting of test sample was created every time the model committed mistake at the time of mentoring.

### 3.6 K-way n-shot learning

Few Shots Learning (FSL) introduces the term "K-way n-shot learning" for classification tasks, where "K-way" denotes the number of classes, and "n-shot" represents the available labeled dataset instances for each class. In this context, a training set is comprised of pairs of data instances, making  $M$  equal to  $K$  multiplied by  $n$ . FSL, as described by Wang et al. [55], refers to a ML problem where experience 'E' is derived from a limited number of examples, providing supervised information for the target task 'T'. In simple terms, FSL involves learning from a few examples to generalize or learn new categories in a classification problem, utilizing only a limited number of labeled datasets without starting from scratch [55][56][57]. Model-Agnostic Meta-Learning (MAML) [58], Prototypical Networks [59], Matching Networks [60], and Relation Networks [56] are among the popular techniques for implementing FSL in classification problems.

We employ FSL to reduce algorithmic complexity in predicting test samples. The approach involves creating a single representative (support set) for each sub-cluster within the Covid-19 positive and negative clusters. The class of the test sample is determined by assigning it to the class

it most resembles. For this, we measured the distance from test data to all the representative of sub-clusters (or centroid) and assigned it to the class having the greatest number of similar sub-cluster - an approach like K-Nearest Neighbor. FSL is implemented using the concept of prototypical networks, where each support set represents the average of all data instances in the corresponding subcluster, and the test instance's class is set as the closest support set. If  $\hat{y}_i$  represents the predicted class of the  $i^{\text{th}}$  instance,  $x_i$  denotes the  $i^{\text{th}}$  test data, and  $\hat{y}$  represents the representatives of sub-clusters, the framework utilizes the following equation to define how FSL operates.

The algorithm of the framework is presented as follows:

---

**Algorithm 1: proposed feedback based Active learning algorithm**

---

Let  $M$ ,  $L$ ,  $TD$ ,  $x$ ,  $y$ ,  $G(x)$  and  $MD$  be the model, labeled dataset, training data, stream of unknown data instances, label for data instance, function for human intervention, and

```

1   For  $t=1, 2, \dots$  do
2       If instance  $x$  belongs to  $MD$ 
3           If  $M(x)$  is not correct
4               Activate  $G(x)$  (Feedback)
5                $TD = TD \cup \langle x, y \rangle$ 
6                $L = L \cup \langle x, y \rangle$ 
7           Else
8                $TD = TD \cup x$ 
9       Else
10           $TD = TD \cup x$ 
11      End
12  End

```

## Chapter 4

### 4. Experimental setup

---

*Summary:* This Chapter describes the overview of experimentations, the dataset, evaluation, and validation measure used, specifically clustering indices and classification metrics used.

**Key topics:** Dataset, classification metrics, clustering indices, validation, and experiments

---

#### 4.1 Overview

We selected three DL models that were pre-trained on ImageNet dataset. These models were mentored until they were ready to independently cluster Covid-19 positive cases from the non-Covid-19 cases. Pre-trained models were selected to illustrate that the framework works on any on-the-shelf models that are easily available/ accessible.

In all the three cases, experiments involved commencing training with a limited known sample, defining the portion of the dataset that is mentored ( $M_i$ ), and checking for every  $i$ , whether the machine commits a mistake. We used three pre-trained CNN models: VGG16, ResNet101, and DenseNet169, followed by unsupervised clustering using indices such as Dunn's, Davies Bouldin, and Silhouette. To measure the similarity between the test sample and train data, we used three distance measures: Euclidean, Manhattan and Cosine distance. Stopping criteria for each case was studied defining mentoring in four ascending data sample tiers–  $M_1$ ,  $M_2$ ,  $M_3$ , and  $M_4$ . Mentoring refers to including humans/Experts in the training loop where the Expert intervenes to improve performance until the model has sufficient knowledge to make predictions independently. The count of data samples ranged from the smallest in  $M_1$  to the largest in  $M_4$ . Due to the variation in the size of datasets, each modality has a different sized tier. For cough sound, the values of tiers are 200 ( $M_1$ ), 600 ( $M_2$ ), 800 ( $M_3$ ), 1210 ( $M_4$ ), whereas for CT scans they are 200 ( $M_1$ ), 600 ( $M_2$ ), 1400 ( $M_3$ ), 3000 ( $M_4$ ). Similarly, X-rays consist of tiers of 160 ( $M_1$ ), 360 ( $M_2$ ), 760 ( $M_3$ ), 1510 ( $M_4$ ).

Initially, we trained the framework without forming subclusters, meaning each deep feature was considered as an independent element and distance was measured to each sample before assigning the class of the test data. For instance, in the case of X-ray, a test data sample was compared with

all 40 images (train data) before assigning it to a specific class/cluster. This brute force approach increased the complexity of the framework with the computational complexity of  $O(n^2)$ , where  $n$  is the number of data samples. The time taken for training on each case is described in Chapter 5. To overcome this issue, we introduced the concept of FSL which reduces the complexity to  $O(n \times k)$ , where  $n$  is the number of data sample and  $k$  is the number of subcluster. Same experiments were conducted in both cases. Pictorial illustration of FSL is shown in Figure 5, where  $C_{k1}$  and  $C_{k2}$  are the two clusters consisting of ‘ $n$ ’ and ‘ $m$ ’ number of subcluster, respectively. Similarly,  $f_{k1i}^\mu$  and  $f_{k2j}^\mu$  represents the mean (or centroid) of the subcluster of  $C_{k1}$  and  $C_{k2}$ , respectively, where ‘ $i$ ’ denotes the number of subclusters in  $C_{k1}$  and ‘ $j$ ’ denotes the number of subclusters in  $C_{k2}$ .

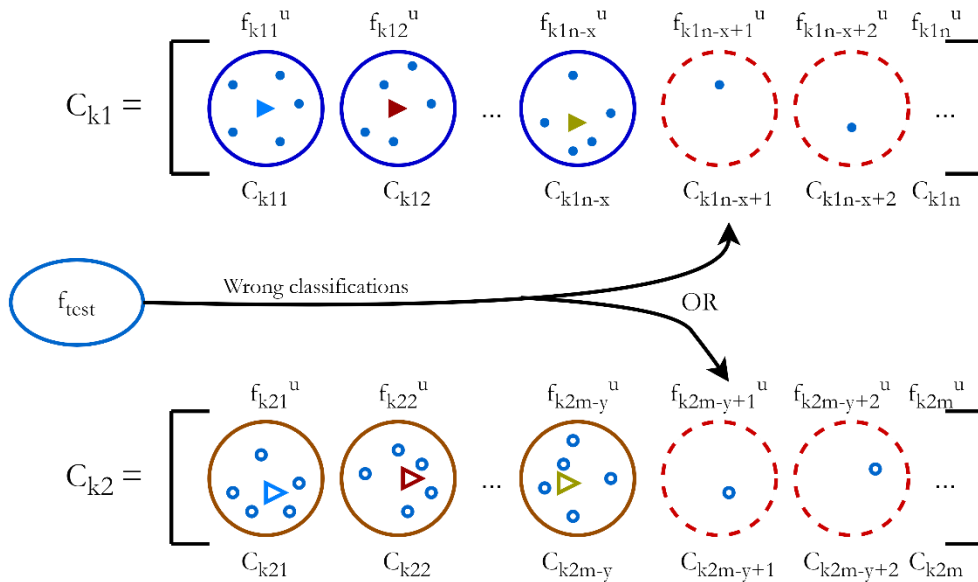


Figure 5: Figure depicting the subclusters and formation of new subcluster when the model commits a mistake.

## 4.2 Dataset

Not all infectious diseases affect the lungs [61], severe acute respiratory syndrome (SARS), Middle East Respiratory Syndrome (MERS), Covid-19, etc. are some of the major epidemics/pandemics that infect the lungs. Chest radiographs and cough sound are considered important modalities for the mass screening of Covid-19 lungs infection. For this reason, we selected three data modalities to validate the proposed framework: Chest X-rays, Chest Computer Tomography (CT) scans, and cough sounds. Each modality is described below:

### Chest X-rays (CXRs)

Generally, CXRs are globally popular due to their accessibility, painless imaging, low operating cost, and comprehensiveness in providing visual information on all relevant organs for diagnosing pulmonary diseases [62]. It is excessively used in the diagnosis of respiratory and pulmonary-related diseases such as pneumonia, tuberculosis [63], Chronic Obstructive Pulmonary Disease

(COPD) [64], and Covid-19 is not an exception [65]. At the time of experimenting, no single massive data consisting of both Covid-19 positive and non-Covid-19 were present, so we collected CXRs from various sources.

**Healthy and Pneumonia dataset** [66]: All healthy and pneumonia CXRs were collected from a retrospective cohort of children aged one to five years old at Guangzhou Women and Children’s Medical Center in Guangzhou. Three experts reviewed the radiographs for quality control; the first two experts rated the CXR images, approving them for use in the AI system, whereas the third Expert re-evaluated it ensuring no bias occurred when grading. The dataset consisted of 1,583 healthy images and 731 pneumonias. The pneumonia dataset was collected from patients’ regular medical treatment.

**Covid-19 dataset**<sup>2,3,4</sup>: The Covid-19 dataset was collected by combining data from three publicly available sources: the Radiological Society of North America, Qatar University, and the University of Dhaka. The dataset consists of 2,358 CXRs confirmed to be Covid-19 positive.

All combined, the dataset was composed of 4,714 images; however, to prepare a balanced dataset, we randomly selected 2,357 CXRs for both Covid-19 positive and non-Covid-19 cases. The healthy and pneumonia dataset was combined to create a non-Covid-19 cases. Figure 6 illustrates the example images of CXRs.

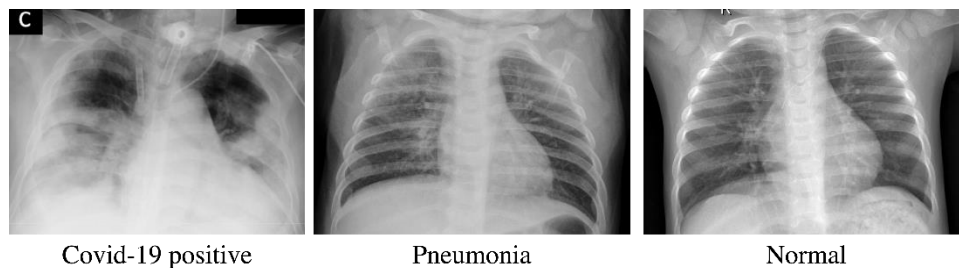


Figure 6: Example images of Covid-19 positive, pneumonia, and normal CXRs used in this thesis.

### Computed Tomography (CT) scans

Like CXRs, CT scans are one of the crucial modalities for studying/screening respiratory-related diseases. CT scans are obtained using specialized X-ray equipment and computer technology to produce detailed cross-sectional images of the body. The purpose of validating the framework on CT scans is to provide evidence that the proposed framework is versatile/applicable to any type of image dataset. We selected COVIDx CT-3 as it was considered a multinational benchmark CT scan dataset for Covid-19 detection/classification [67]. COVIDx CT-3 was created by collecting data from a cohort of patients from various locations. The organization and initiatives involved in collecting data are (1) China National Center for Bioinformation (CNCB), (2) the National

<sup>2</sup> <https://github.com/agchung/Figure1-COVID-chestxray-dataset>

<sup>3</sup> <https://github.com/agchung/Actualmed-COVID-chestxray-dataset>

<sup>4</sup> <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>

Institutes of Health Intramural Targeted Anti-Covid-19 (ITAC) Program, (3) the COVID-CTset, (4) Integrative CT Images and Clinical Features for COVID-19 (iCTCF), (5) COVID-19 CT Lung and Infection Segmentation initiative (COVID-19-CT-Seg), (6) Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI), (7) Radiopaedia collection, (8) MosMedData, (9) Stony Brook University, (10) Study of Thoracic CT in COVID-19 (STOIC), and (11) COVID-CT-MD. The dataset consists of patients with one of three medical conditions: (a) COVID-19, (b) Community-Acquired Pneumonia (CAP), or (c) normal controls. We considered CAP and normal controls non-COVID-19 for this case study, resulting in two class classification problems: 0/1 COVID-19/non-COVID-19. The dataset can be downloaded from Kaggle (<https://www.kaggle.com/datasets/hgunraj/covidxet>). It includes 431,605 axial CT slices from 6,068 patients across 17 different countries. Nevertheless, for numerous patients, it fails to offer the complete slices necessary to construct a CT scan volume. As a result, we treated each CT slice as a distinct data sample. We randomly selected 10,000 slices from the available pool due to the hardware restriction and to reduce the training time. Figure 7 illustrates the example of CT scans used in the thesis.

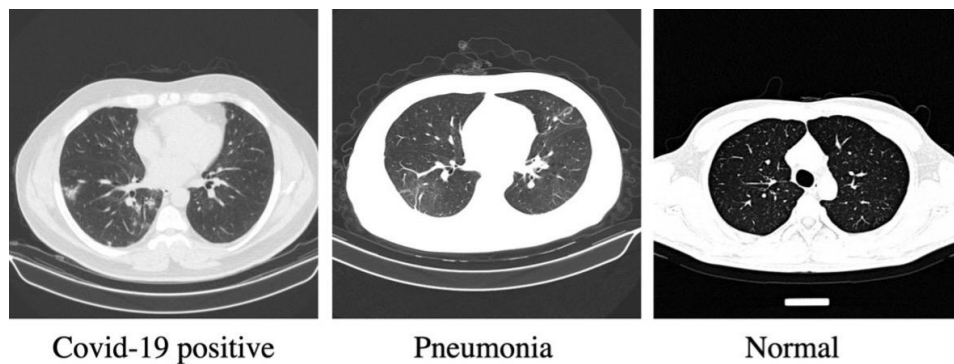


Figure 7: Example images of Covid-19 positive, pneumonia, and normal CT scans used in this thesis.

### Cough sound

Basically, a cough dataset is a collection of sound records of human cough collected primarily by crowdsourcing using a website or mobile application [14] [69]. Coughing is a shared symptom among various diseases, such as asthma, bronchitis, pertussis, and Covid-19. Notably, a dry cough is one of the most common symptoms associated with Covid-19 [70]. Out of all the available cough sound dataset, we selected the Coswara dataset as it has the quality and comprehensive labels for each instance. Coswara is a crowdsourced dataset of 2020 data samples, each corresponding to a unique subject. The dataset was collected through a web application, where participants - both infected and non-infected - recorded cough sounds (shallow and heavy), breathing sounds (respiratory), and voice sounds (vowel sounds). For the thesis, we selected heavy cough sound dataset. It consisted of seven classes: healthy, positive moderate, recovered full, positive mild, positive asymptomatic, no respiratory illness exposed, and respiratory illness not identified. We combined healthy, recovered full, and no respiratory illness as a non-Covid-19 class (1364 data

samples), whereas positive moderate, positive mild, and positive asymptomatic were considered Covid-19 positive class (656 data samples). We discarded all the data samples for which respiratory illness was not identified. Note that the audio waveforms were converted into Mel spectrograms (shown in Figure 8) before feeding them to the framework (using python library: Librosa [71]). The Mel spectrogram depicts the temporal evolution of a signal's frequency content, aligning the frequencies with the Mel scale to reflect the nuances of human auditory perception more accurately. This was a required pre-processing step as the framework only accept image data as input.

The dataset can be downloaded from the following sources:

GitHub: <https://github.com/iiscleap/Coswara-Data>) and

Kaggle: <https://www.kaggle.com/datasets/janashreeanathan/coswara>.

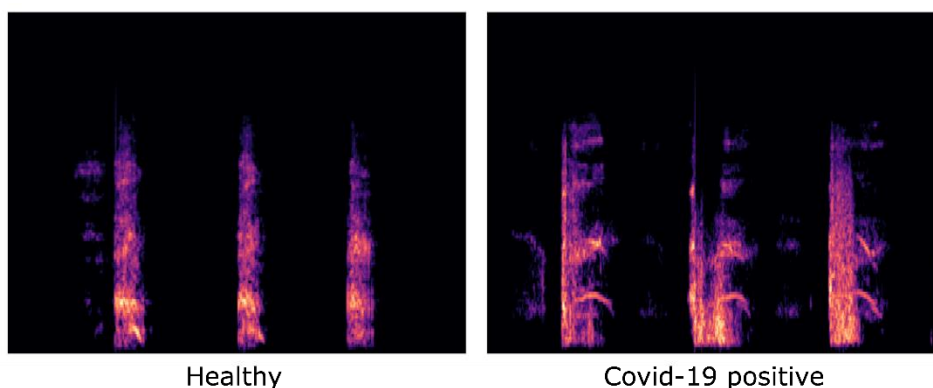


Figure 8: Example images of covid-19 positive and non-covid-19 cases Mel-spectrograms used in this thesis.

### 4.3 Evaluation and validation

We used two types of evaluation techniques to evaluate the performance of the framework: clustering indices and classification metrics. Clustering indices judged the quality of the clusters formed, whereas the classification metrics determined how well the framework classifies positive and negative cases (Covid-19 positive/non-Covid-19 cases).

#### Clustering indices

Clustering indices, also known as cluster validity indices, evaluate the quality of the clusters. The quality of clusters is based on the internal indices: cluster cohesion and separation. Cluster cohesion measures the compactness of the cluster, meaning how similar the objects are in the cluster formed. Cluster separation refers to how well the cluster are separated or how distinct are the elements of one cluster from another. We used three popular clustering indices: Dunn's index, Davies Bouldin Index, and Silhouette index. Each index is described below:

Dunn's index (DI) [72]: DI is the ratio of minimum inter-cluster distance to maximal intra-cluster distance. For the 'n' number of clusters, the DI is represented by as,

$$DI = \min_{i=1, \dots, n} \left( \min_{j=i+1, \dots, n} \left\{ \frac{\delta(s_i, s_j)}{\max_{k=1, \dots, n} (\delta(s_k))} \right\} \right),$$

where  $\delta(s_i, s_j) = \min_{c_a \in s_i, c_b \in s_j} \delta(c_a, c_b)$  and  $\delta(s_k) = \max_{c_a, c_b \in s} \delta(c_a, c_b)$ . DI tends to be maximum when clusters have large inter-cluster distances and small intra-cluster distances. In other words, an optimum number of clusters (or 'n') is the one that maximizes (best is 1) the DI.

Davies-Bouldin index (DB) [73]: It identifies the compactness of the clusters and the clusters far from each other, as shown in the following equation:

$$DB = \frac{1}{N} \sum_1^N \max_{j=1, \dots, n, i \neq j} \left( \frac{\text{dist.}(s_i) + \text{dist.}(s_j)}{\delta(s_i, s_j)} \right),$$

where  $\text{dist.}(s_i) = \frac{1}{k_i} \sum_{c_a \in s_i} \delta(c_a, c_i^{mean})$ ,  $k_i$  is the number of elements,  $c_i^{mean}$  is the centroid of cluster  $s_i$ , and N is the number of clusters. The minimal value of DB indicates the best number of clusters.

Silhouette index (SI) [74]: SI is based on the comparison of tightness and separation of clusters. The average Silhouette width provides information about how good the number of the selected clusters is. It is shown in the equation below:

$$SI = \frac{1}{N} \sum_{i=1}^N \frac{\text{dist}2_i - \text{dist}1_i}{\max(\text{dist}1_i, \text{dist}2_i)},$$

where  $\text{dist}2_i$  is the average distance between an element and all other elements in the same cluster,  $\text{dist}1_i$  is the minimum average distance between the element and all other elements in another cluster, and N is the total number of data points. The SI values range from -1 to 1, where 1 denotes the best value.

For all indices, the distance was calculated using three distance measures: Euclidean Distance (ED), Manhattan Distance (MD), and Cosine Distance (CS) given by  $ED(a, b) = \sqrt{\sum_{i=1}^n |a_i - b_i|^2}$ ,  $MD(a, b) = \sum_{i=1}^n |a_i - b_i|$ ,  $CD(a, b) = 1 - \frac{\sum_{i=1}^n a_i * b_i}{\sqrt{\sum_{i=1}^n a_i^2} * \sqrt{\sum_{i=1}^n b_i^2}}$ ,

respectively.

### Classification metrics

We opted for four widely used classification metrics: accuracy, specificity, sensitivity, and Area Under the ROC curve (AUC). In calculating these metrics, the model's predicted class falls into one of the four categories: True Positive (TP) when the framework correctly predicts Covid-19 positive cases, False Positive (FP) when it incorrectly predicts non-Covid-19 images as Covid-19, False Negative (FN) when the model incorrectly predicts Covid-19 images as non-Covid-19, and True Negative (TN) when it accurately predicts non-Covid-19 images. The following section elaborates on each of these metrics.



Accuracy (ACC): It measures the percentage of closeness of the model predicted classes with the Ground Truth (GT). For this study, it evaluates the capability of the model to predict TP and TN correctly, meaning how well the model predicts Covid-19 positive cases and non-Covid-19 cases into their corresponding classes. The formula for ACC is depicted below:

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Specificity (SPEC): It is simply known as a True Negative Rate (TNR). Specificity describes the ability of the model to predict a TN case. The formula is given by:

$$SPEC = \frac{TN}{(TN + FP)}$$

Sensitivity (SEN): Sensitivity (or Recall) refers to the capability of a model to predict the TP cases. It is often known as True Positive Rate (TPR) since it measures the percentage of correctly identified positive results among all the individuals or cases that are truly positive. High sensitivity imply that the model is predicting TP cases without missing the cases. The formula is depicted below:

$$SEN = \frac{TP}{(TP + FN)}$$

Area Under the ROC Curve (AUC): It determines how well a model distinguishes between the classes [75]. It is calculated by plotting the TPR (or sensitivity) against the False Positive Rate (FPR) (or 1-specificity) on different classification thresholds, and AUC is the area covered by the curve.

## Chapter 5

### 5. Results and analysis

---

**Summary:** *This chapter describes the results and observation in detail on all three datasets in both with and without subclusters formation. A separate section describing result for each dataset is provided with comparison table that records the result of our proposed method with traditional DL models.*

**Key topics:** Results and comparisons.

---

In this section, we thoroughly explain the observed results, first experimenting without forming subclusters, and then with subcluster. The DL models were iterative trained in four mentored cases ( $M_1$ ,  $M_2$ ,  $M_3$ , and  $M_4$ ), and the number of data samples requiring correction was recorded for each training session (see Figure 9). The figure provides an overview of the average cumulative counts of corrections for each case (Figure 9 (a)), and the mistake rate observed at each case (Figure 9 (b)). For instance, for cough sound in  $M_1$ , the models made an average of 77.93 mistakes, indicating that the expert's intervention or correction occurred approximately 77.93 times on average during the  $M_1$  session. The mistake rate as shown in Figure 9 (b) is the ratio of cumulative average mistake count and total number of data sample in the specific mentored case.

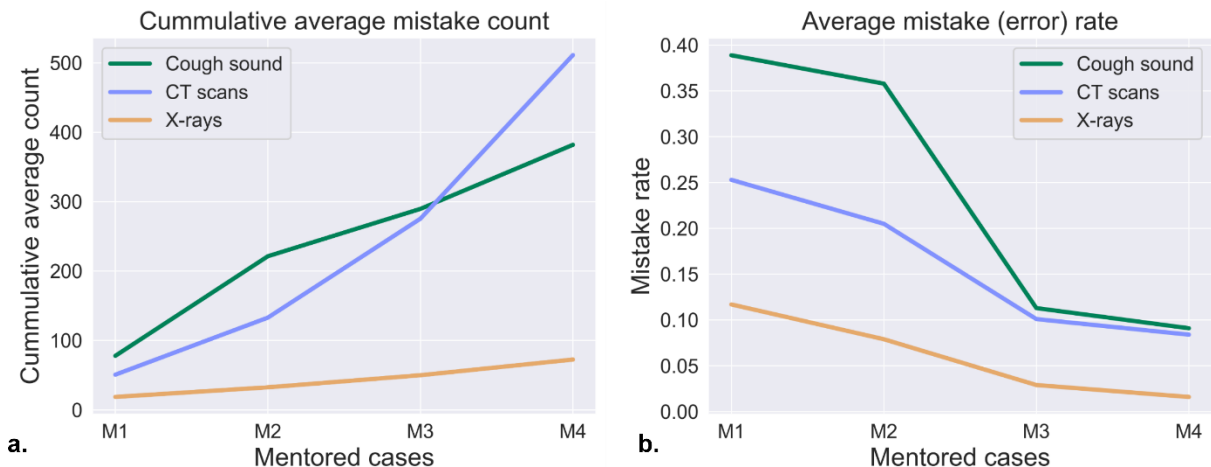


Figure 9: Illustration of cumulative average corrected counts on mentored cases made by the Expert when the model misclassifies the data sample. a) Record of the raw cumulative average count on four mentored cases on three datasets, and b) Rate of mistakes on four mentored cases on three datasets. Note: the figure was plotted when experimenting without forming subclusters.

Table 1 demonstrates the time taken for training on each modality when experimenting with and without forming subclusters. On both the experiments, we observed highest training time in CT scan dataset. This was expected as CT scan consisted highest amount of data samples. Creating subclusters remarkably reduced the required training time in all the cases as the computational complexity was better. Using the concept of FSL, the training time for X-rays, CT scans, and cough sound was approximately 8 times, 2.2 times, and 2 times faster, respectively.

Table 1: Comparison of time taken and computational complexity for training with and without formation of subclusters.

Dataset	Time taken for training	Complexity
Without subclusters		
X-rays	~ 8 minutes	$O(n^2)$
CT scans	~ 38 minutes	$O(n^2)$
Cough sound	~ 40 seconds	$O(n^2)$
With subclusters		
X-rays	~ 1 minute 5 seconds	$O(n \times k)$
CT scans	~ 15 minutes 50 seconds	$O(n \times k)$
Cough sound	~ 19 seconds	$O(n \times k)$

Since we measured the similarity between the clusters prior to assigning the class to test sample - like the K Nearest Neighbor algorithm - it is important to determine the optimum number of closest data to account before setting the class. To obtain such optimum number, we plotted a graph for  $M_1$  and  $M_4$  for each modalities recording the accuracy against the value of K (ranging from 5 to 39) as shown in Figure 10.  $M_1$  and  $M_4$  were selected as they consisted minimum and highest number of mentored data samples, respectively. For all the modalities, minimum (i.e.,  $K=5$ ) value of K yielded highest accuracy, except for cough sound at  $M_4$ , where the accuracy was highest at K equals 31. For the experiments, we selected the value of K as 31, 25, and 31 for cough sound, CT scans, and X-rays dataset, respectively.

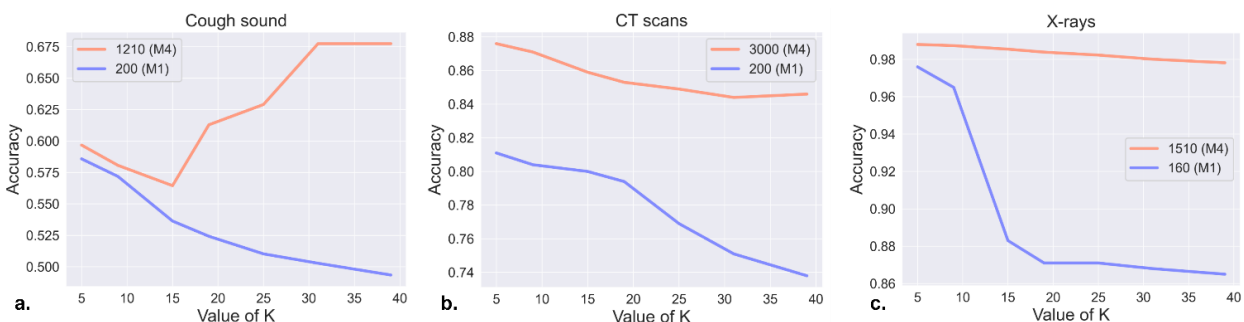


Figure 10: Illustration of accuracy against the value of ‘K’ in K Nearest Neighbors to determine the optimum value of K. a) Accuracy against value of K in cough sound, b) Accuracy against value of K in CT scans, and c) Accuracy against value of K in X-rays. Note: the figure was plotted by experimenting without forming subclusters.

For all modalities, the categorization of each unidentified data sample into either the Covid-19 or non-Covid-19 group was determined based on the similarity of the data to clusters, employing the distance measure  $\delta(x_i, x_j)$ . Here,  $\delta$  is defined as one of the distance measures: Euclidean, Manhattan, or Cosine distance, with  $x_i$  representing the unidentified data sample and  $x_j$  denoting the data within the cluster. We employed three clustering indices (DI, DBI, and SI) to assess the separation and cohesion between Covid-19 positive and non-Covid-19 cases within the clusters. Additionally, classification metrics such as accuracy, specificity, sensitivity, and area under the ROC curve were measured. For each iteration (or mentored tiers), the classification scores corresponding to the clustering indices were recorded to determine the point at which the mentoring could be concluded.

Our evaluation methodology is illustrated through the following example. For simplicity, we focus only on DI and AUC values for ResNet101 using Cosine distance on Cough sound dataset (shown in Table 4). In  $M_1$ , a DI of 0.349 and an AUC value of 0.642 were observed. Similarly,  $M_2$  showed DI and AUC values of 0.345 and 0.677.  $M_3$  exhibited a DI of 0.345 and an AUC value of 0.702, while  $M_4$  displayed DI and AUC values of 0.345 and 0.721, respectively. Despite the clustering index (DI) remaining relatively constant, there was an increasing trend in classification metrics from  $M_1$  to  $M_4$ . According to DI, the most effective cluster formation occurred in  $M_1$  (with 200 mentored data samples), indicating successful learning. However, the corresponding metrics did

not yield optimal results, suggesting subpar system performance. Consequently, attention shifted to  $M_2$ , where the DI was reduced, but the AUC value increased by 0.035. Similarly, in  $M_3$ , the AUC value increased by 0.025, and in  $M_4$ , it increased by 0.019. It's noteworthy that the most significant increase in the AUC value occurred from  $M_1$  to  $M_2$ , followed by  $M_2$  to  $M_3$  and  $M_3$  to  $M_4$ . Improvement, even if insignificant, in classification metrics was still observed from  $M_3$  to  $M_4$ . Therefore, we deduce that the Expert interference or mentoring is necessary until  $M_4$ , or mentoring is required only for 1,210 images. This example is applicable to all other classification metrics.

This was how we determined the stopping criteria to include Expert in the training loop in all the modalities. Following sections explains the results in detail on data modalities: cough sound, CT scans, and Chest X-rays.

## 5.1 Cough sound

The clustering and classification results on each distance measures for the cough sound dataset are described as follows:

**Euclidean Distance (ED):** Table 2 displays the results in ED. The most favorable DI, SI, and DBI were observed in  $M_1$ , registering at 0.349 (VGG16), 0.100 (ResNet101), and 2.664 (ResNet101), respectively. In  $M_2$ , the optimal DI was comparable to that of  $M_1$  (0.345, VGG16), while the best SI reduced to 0.050 (ResNet101), and the best DBI deteriorated to 4.082 (ResNet101). Similarly, in  $M_3$ , the optimum DI, DBI, and SI were 0.345 (VGG16), 4.802 (ResNet101), and 0.037 (ResNet101), respectively.  $M_4$  sustained the best DI at 0.345 (VGG16), with a DBI of 6.290 (ResNet101) and an SI of 0.021 (ResNet101). Similarly, the highest overall Accuracy (ACC) and area under the ROC curve (AUC) were noted in  $M_4$  (0.688, VGG16) and  $M_3$  (0.747, DenseNet169), respectively. For sensitivity and specificity, the optimum values occurred in  $M_4$  (0.762, ResNet101) and  $M_1$  (0.940, VGG16), respectively. Across all models, both accuracy and sensitivity exhibited a gradual increase from  $M_1$  through  $M_4$ . Interestingly, DenseNet169 and VGG16 demonstrated a fluctuating AUC, rising until  $M_3$  and declining in  $M_4$ . Strikingly, specificity was at its peak in  $M_1$ , following a descending trend from  $M_1$  through  $M_4$ . Following the above discussed example, for ED, mentoring is required until  $M_4$  for ResNet101 and until  $M_1$  for VGG16 and DenseNet169.

**Manhattan Distance (MD):** Table 3 presents the results in MD. The most favorable DI, DBI, and SI were noted at  $M_1$ , recording values of 0.349 (VGG16), 0.096 (ResNet101), and 2.816 (ResNet101), respectively. In  $M_2$ , the optimum DI was comparable to  $M_1$  (0.345, VGG16), while the best SI decreased to 0.049 (ResNet101), and the best DBI reduced to 4.085 (ResNet101). Similarly, in  $M_3$ , the optimal DI, DBI, and SI were 0.345 (VGG16), 4.861 (ResNet101), and 0.036 (ResNet101), respectively.  $M_4$  sustained the best DI at 0.345 (VGG16), with a DBI of 6.380 (ResNet101) and an SI of 0.021 (ResNet101). The highest overall accuracy and area under the ROC curve were observed in  $M_4$  (0.688, VGG16) and  $M_3$  (0.747, DenseNet169), respectively. For sensitivity and specificity, the optimal values occurred in  $M_4$  (0.762, ResNet101) and  $M_1$  (0.940, VGG16), respectively. Across all models, both accuracy and sensitivity exhibited a gradual

increase from  $M_1$  through  $M_4$ . Interestingly, DenseNet169 and VGG16 demonstrated a fluctuating AUC, rising until  $M_3$  and declining at  $M_4$ . Strikingly, specificity was at its peak in  $M_1$ , following a descending trend from  $M_1$  through  $M_4$ . Applying the method as described in the example above, for MD, mentoring is required until  $M_4$  for ResNet101 and until  $M_3$  for VGG16 and DenseNet169.

**Cosine Distance (CD):** Table 4 presents the results in CD. The most favorable DI, SI, and DBI were observed at  $M_1$ , with values of 0.349 (VGG16), 0.095 (ResNet101), and 2.832 (ResNet101), respectively. In  $M_2$ , a comparable best DI to  $M_1$  was noted (0.345, VGG16), while the best SI decreased to 0.049 (ResNet101), and the best DBI degraded to 4.175 (ResNet101). Similarly, in  $M_3$ , the optimal DI, DBI, and SI were 0.345 (VGG16), 4.952 (ResNet101), and 0.036 (ResNet101), respectively.  $M_4$  sustained the best DI at 0.345 (VGG16), with a DBI of 6.392 (ResNet101) and an SI of 0.021 (ResNet101). The highest overall accuracy, area under the ROC curve, sensitivity, and specificity were observed in  $M_4$  (0.688, VGG16),  $M_3$  (0.762, DenseNet169),  $M_3$  (0.787, ResNet101), and  $M_1$  (0.957, VGG16), respectively. An increasing trend in ACC and SEN was observed in all models. Like MD, in all models, specificity was highest in  $M_1$ , followed by  $M_2$ , and so forth. In VGG16 and DenseNet169, AUC increased until  $M_3$  and decreased in  $M_4$ . For CD, mentoring is required up to  $M_3$  (800 data samples).

For cough sound, experimenting by creating subclusters, we observed the best Accuracy, Sensitivity, Specificity, and AUC of 0.620, 0.700, 0.500, and 0.680, respectively. However, the best DI, DBI and SI of 0.190, 2.805, and 0.059 was observed in  $M_1$ .

Table 2: Table showing the clustering and classification results when using Euclidean distance as distance measure for cough sound dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy (ACC)	AUC	Sensitivity (SEN)	Specificity (SPEC)
<b>200 (M<sub>1</sub>)</b>							
VGG16	0.349 ± 0.030	9.097± 2.355	0.010 ± 0.004	0.548± 0.034	0.646± 0.011	0.168± 0.109	0.940 ± 0.046
ResNet101	0.206 ± 0.004	2.664± 0.038	0.100 ± 0.003	0.591± 0.008	0.637 ± 0.012	0.541± 0.023	0.642± 0.026
DenseNet169	0.352 ± 0.024	9.391± 0.669	0.021 ± 0.002	0.526± 0.011	0.651± 0.010	0.089± 0.026	0.977± 0.008
<b>600 (M<sub>2</sub>)</b>							
VGG16	0.345 ± 0.021	9.461 ± 0.305	0.010 ± 0.001	0.660 ± 0.026	0.734 ± 0.002	0.494 ± 0.098	0.832 ± 0.066
ResNet101	0.173 ± 0.011	4.082 ± 0.285	0.050 ± 0.003	0.622 ± 0.003	0.671 ± 0.006	0.724 ± 0.107	0.521 ± 0.112
DenseNet169	0.315 ± 0.002	8.855 ± 0.340	0.017 ± 0.003	0.655 ± 0.019	0.728 ± 0.005	0.467 ± 0.083	0.850 ± 0.054
<b>800 (M<sub>3</sub>)</b>							
VGG16	0.345 ± 0.022	9.700 ± 0.135	0.009 ± 0.000	0.670 ± 0.014	0.740 ± 0.010	0.585 ± 0.067	0.760 ± 0.100
ResNet101	0.165 ± 0.000	4.802 ± 0.163	0.037 ± 0.001	0.624 ± 0.013	0.698 ± 0.010	0.744 ± 0.046	0.511 ± 0.064
DenseNet169	0.313 ± 0.000	9.407 ± 0.034	0.012 ± 0.002	0.680 ± 0.001	0.747 ± 0.007	0.589 ± 0.079	0.776 ± 0.067
<b>1210 (M<sub>4</sub>)</b>							
VGG16	0.345 ± 0.022	10.735±0.260	0.008 ± 0.000	0.688 ± 0.009	0.732 ± 0.048	0.659 ± 0.062	0.729 ± 0.061
ResNet101	0.165 ± 0.000	6.290 ± 0.412	0.021 ± 0.003	0.650 ± 0.008	0.725 ± 0.009	0.762 ± 0.035	0.546 ± 0.035
DenseNet169	0.342 ± 0.020	10.711 ± 0.323	0.008 ± 0.000	0.672 ± 0.030	0.740 ± 0.045	0.642 ± 0.090	0.714 ± 0.045

Table 3: Table showing the clustering and classification results when using Manhattan distance as distance measure for cough sound dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>200 (M<sub>1</sub>)</b>							
VGG16	0.349 ± 0.030	9.255 ± 2.699	0.011 ± 0.005	0.539 ± 0.030	0.629 ± 0.010	0.160 ± 0.110	0.930 ± 0.056
ResNet101	0.195 ± 0.016	2.816 ± 0.071	0.096 ± 0.005	0.605 ± 0.002	0.644 ± 0.011	0.660 ± 0.024	0.550 ± 0.020
DenseNet169	0.347 ± 0.021	10.948 ± 0.178	0.017 ± 0.003	0.514 ± 0.007	0.634 ± 0.011	0.057 ± 0.016	0.984 ± 0.007
<b>600 (M<sub>2</sub>)</b>							
VGG16	0.345 ± 0.021	9.278 ± 0.188	0.001 ± 0.001	0.650 ± 0.017	0.725 ± 0.005	0.502 ± 0.103	0.806 ± 0.080
ResNet101	0.173 ± 0.011	4.085 ± 0.188	0.049 ± 0.002	0.624 ± 0.011	0.668 ± 0.011	0.721 ± 0.088	0.528 ± 0.104
DenseNet169	0.315 ± 0.002	9.184 ± 0.392	0.018 ± 0.003	0.649 ± 0.014	0.732 ± 0.007	0.420 ± 0.053	0.886 ± 0.035
<b>800 (M<sub>3</sub>)</b>							
VGG16	0.345 ± 0.022	9.734 ± 0.054	0.009 ± 0.000	0.650 ± 0.006	0.724 ± 0.004	0.580 ± 0.067	0.726 ± 0.084
ResNet101	0.165 ± 0.000	4.861 ± 0.249	0.036 ± 0.002	0.631 ± 0.009	0.699 ± 0.012	0.764 ± 0.067	0.505 ± 0.079
DenseNet169	0.313 ± 0.000	9.696 ± 0.149	0.011 ± 0.002	0.669 ± 0.006	0.749 ± 0.004	0.593 ± 0.127	0.753 ± 0.137
<b>1210 (M<sub>4</sub>)</b>							
VGG16	0.345 ± 0.022	10.793 ± 0.192	0.008 ± 0.000	0.646 ± 0.026	0.719 ± 0.049	0.623 ± 0.056	0.676 ± 0.027
ResNet101	0.165 ± 0.000	6.380 ± 0.351	0.021 ± 0.003	0.648 ± 0.014	0.721 ± 0.012	0.756 ± 0.070	0.552 ± 0.029
DenseNet169	0.313 ± 0.000	10.884 ± 0.222	0.008 ± 0.000	0.668 ± 0.040	0.757 ± 0.032	0.668 ± 0.111	0.683 ± 0.054



Table 4: Table showing the clustering and classification results when using Cosine distance as distance measure for cough sound dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>200 (M<sub>1</sub>)</b>							
VGG16	0.349 ± 0.030	8.999 ± 1.712	0.009 ± 0.002	0.544 ± 0.030	0.643 ± 0.014	0.143 ± 0.085	0.957 ± 0.030
ResNet101	0.195 ± 0.016	2.832 ± 0.235	0.095 ± 0.005	0.590 ± 0.010	0.642 ± 0.016	0.631 ± 0.149	0.549 ± 0.138
DenseNet169	0.348 ± 0.022	8.219 ± 1.751	0.028 ± 0.005	0.578 ± 0.052	0.669 ± 0.022	0.225 ± 0.134	0.943 ± 0.035
<b>600 (M<sub>2</sub>)</b>							
VGG16	0.345 ± 0.021	9.593 ± 0.138	0.010 ± 0.000	0.671 ± 0.028	0.743 ± 0.004	0.493 ± 0.090	0.856 ± 0.060
ResNet101	0.173 ± 0.011	4.175 ± 0.198	0.049 ± 0.002	0.619 ± 0.006	0.677 ± 0.011	0.752 ± 0.079	0.485 ± 0.089
DenseNet169	0.313 ± 0.000	8.632 ± 0.283	0.016 ± 0.003	0.684 ± 0.003	0.749 ± 0.010	0.600 ± 0.059	0.771 ± 0.064
<b>800 (M<sub>3</sub>)</b>							
VGG16	0.345 ± 0.022	9.891 ± 0.138	0.009 ± 0.000	0.684 ± 0.017	0.747 ± 0.010	0.581 ± 0.067	0.792 ± 0.099
ResNet101	0.165 ± 0.000	4.952 ± 0.271	0.036 ± 0.002	0.633 ± 0.008	0.702 ± 0.016	0.787 ± 0.086	0.488 ± 0.093
DenseNet169	0.313 ± 0.000	9.489 ± 0.142	0.011 ± 0.002	0.677 ± 0.010	0.762 ± 0.008	0.691 ± 0.083	0.67 ± 0.099
<b>1210 (M<sub>4</sub>)</b>							
VGG16	0.345 ± 0.022	10.736 ± 0.256	0.008 ± 0.001	0.688 ± 0.016	0.727 ± 0.061	0.655 ± 0.008	0.736 ± 0.085
ResNet101	0.165 ± 0.000	6.392 ± 0.372	0.021 ± 0.003	0.656 ± 0.012	0.721 ± 0.014	0.777 ± 0.059	0.544 ± 0.037
DenseNet169	0.313 ± 0.000	10.828 ± 0.285	0.008 ± 0.000	0.675 ± 0.027	0.753 ± 0.046	0.702 ± 0.095	0.663 ± 0.056

## 5.2 CT scans

The clustering and classification results on each distance measures for the CT scan dataset are described as follows:

**Euclidean Distance (ED):** Table 5 provides a comprehensive overview of performance results assessed in ED. Specifically, in  $M_1$ , DI values were documented as 0.132 (for VGG16), 0.144 (for DenseNet169), and 2.989 (for DenseNet169). As the mentoring progressed to  $M_2$ , there was a reduction in DI, with the optimal DI in  $M_2$  being 0.082 (for VGG16). Concurrently, the best SI in  $M_2$  decreased to 0.086 (for DenseNet169), and the best DBI increased to 4.441 (for DenseNet169). In the subsequent stage,  $M_3$ , the best DI, DBI, and SI values were 0.037 (for ResNet101), 4.326 (for DenseNet169), and 0.067 (for DenseNet169), respectively. Finally, in the fourth mentored case,  $M_4$ , the highest DI was observed as 0.046 (for VGG16), the DBI reached 4.823 (for DenseNet169), and the SI attained a value of 0.049 (for DenseNet169). These findings collectively emphasize the variations in clustering performance across different mentored cases, with the initial stage,  $M_1$ , demonstrating the most favorable clustering outcomes. The highest overall accuracy and area under the ROC curve were noted at  $M_4$  with 0.854 (for VGG16) and 0.938 (for DenseNet169), respectively. For sensitivity and specificity, the optimal values occurred at  $M_2$  (0.912, DenseNet169) and  $M_3$  (0.937, VGG16), respectively. Both ACC and AUC exhibited a gradual increase from  $M_1$  through  $M_4$  in all models. Interestingly, DenseNet169 displayed a fluctuating sensitivity, increasing until  $M_2$  and reducing at  $M_3$ , followed by an increase at  $M_4$ . The trend of gradually increasing SEN remained consistent for VGG16 and ResNet101. Similarly, specificity decreased at  $M_4$  in VGG16, but the trend was consistent in other models. According to the method outlined in the example, for ED, mentoring is required until  $M_4$  (3,000 data samples).

**Manhattan Distance (MD):** Table 6 provides a comprehensive representation of the outcomes obtained in MD. Across the four distinct mentored cases ( $M_1$ ,  $M_2$ ,  $M_3$ , and  $M_4$ ), the most favorable values for DI, SI, and DBI were predominantly observed in the initial case,  $M_1$ . Specifically, within  $M_1$ , the DI achieved values of 0.129 (for VGG16), 0.116 (for DenseNet169), and 3.680 (for DenseNet169). As the mentoring process progressed to  $M_2$ , the DI values experienced a decrease, with the optimal DI in  $M_2$  recorded as 0.073 (for VGG16). Additionally, the optimal SI decreased to 0.081 (for DenseNet169), while the best DBI increased to 3.894 (for DenseNet169). In the subsequent case,  $M_3$ , the most favorable DI, DBI, and SI values were observed as 0.037 (for ResNet101), 4.313 (for DenseNet169), and 0.068 (for DenseNet169), respectively. Finally, in the fourth mentored case,  $M_4$ , the highest DI was noted as 0.046 (for VGG16), while the DBI reached 4.720 (for DenseNet169), and the SI attained a value of 0.051 (for DenseNet169). The overall highest accuracy, area under the ROC curve, and sensitivity were observed at  $M_4$  with values of 0.850 (DenseNet169), 0.935 (VGG16), and 0.894 (DenseNet169), respectively. For specificity, the optimal value was observed at  $M_3$  (0.960, VGG16). Both ACC and AUC exhibited a gradual increase from  $M_1$  through  $M_4$  in all the models. ResNet101 and DenseNet169 demonstrated a fluctuating SEN and SPEC. For MD, mentoring is required up to  $M_4$  (3,000 data samples) according to the outlined method.

**Cosine Distance (CD):** Table 7 provides a comprehensive overview of the results in CD. Across the four distinct mentored cases ( $M_1$ ,  $M_2$ ,  $M_3$ , and  $M_4$ ), the most favorable values for DI, SI, and DBI consistently appeared in the initial case,  $M_1$ . Specifically, within  $M_1$ , the DI yielded values of 0.147 (for ResNet101), 0.039 (for DenseNet169), and 3.725 (for DenseNet169). As the mentoring process progressed to  $M_2$ , the DI exhibited a reduction, with the optimal DI in  $M_2$  recorded as 0.086 (for ResNet101). Simultaneously, the optimal SI decreased to 0.026 (for DenseNet169), while the best DBI increased to 4.313 (for DenseNet169). In the subsequent case,  $M_3$ , the most favorable DI, DBI, and SI values were observed as 0.037 (for ResNet101), 4.619 (for DenseNet169), and 0.032 (for DenseNet169), respectively. Finally, in the fourth mentored case,  $M_4$ , the highest DI was noted as 0.053 (for ResNet101), while the DBI reached 5.105 (for DenseNet169), and the SI attained a value of 0.029 (for DenseNet169). The overall highest accuracy, area under the ROC curve, sensitivity, and specificity were observed at  $M_4$  with values of 0.875 (VGG16), 0.945 (VGG16), 0.855 (VGG16), and 0.905 (DenseNet169), respectively. All four metrics exhibited a gradual increase from  $M_1$  through  $M_4$  for all models, except for sensitivity in VGG16 and ResNet101, where the score decreased at  $M_2$ . For CD, mentoring is needed up to  $M_3$  (1,400 data samples), consistent with other distance measures.

For CT scans, experimenting by creating subclusters, we observed the best Accuracy, Sensitivity, Specificity, and AUC of 0.640, 0.955, 0.329, and 0.802, respectively. However, the best DI, DBI and SI of 0.140, 3.985, and 0.059 was observed in  $M_1$ .

Table 5: Table showing the clustering and classification results when using Euclidean distance as distance measure for CT scan dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy (ACC)	AUC	Sensitivity (SEN)	Specificity (SPEC)
<b>200 (M<sub>1</sub>)</b>							
VGG16	0.132 ± 0.012	6.284 ± 0.335	0.023 ± 0.003	0.775 ± 0.022	0.863 ± 0.666	0.666 ± 0.049	0.883 ± 0.063
ResNet101	0.112 ± 0.004	5.612 ± 0.190	0.030 ± 0.002	0.708 ± 0.014	0.804 ± 0.003	0.591 ± 0.085	0.824 ± 0.056
DenseNet169	0.039 ± 0.007	2.988 ± 0.141	0.144 ± 0.058	0.614 ± 0.031	0.776 ± 0.04	0.853 ± 0.138	0.376 ± 0.197
<b>600 (M<sub>2</sub>)</b>							
VGG16	0.082 ± 0.059	6.838 ± 0.098	0.021 ± 0.001	0.794 ± 0.022	0.898 ± 0.008	0.676 ± 0.079	0.912 ± 0.048
ResNet101	0.076 ± 0.052	6.007 ± 0.063	0.026 ± 0.001	0.747 ± 0.003	0.841 ± 0.003	0.689 ± 0.027	0.804 ± 0.023
DenseNet169	0.030 ± 0.001	4.441 ± 0.329	0.086 ± 0.015	0.697 ± 0.034	0.841 ± 0.019	0.912 ± 0.016	0.482 ± 0.079
<b>1400 (M<sub>3</sub>)</b>							
VGG16	0.035 ± 0.048	7.265 ± 0.016	0.019 ± 0.000	0.829 ± 0.002	0.925 ± 0.002	0.722 ± 0.004	0.937 ± 0.002
ResNet101	0.037 ± 0.049	6.474 ± 0.043	0.022 ± 0.001	0.788 ± 0.009	0.879 ± 0.007	0.732 ± 0.018	0.844 ± 0.036
DenseNet169	0.022 ± 0.015	4.326 ± 0.093	0.067 ± 0.006	0.794 ± 0.012	0.894 ± 0.007	0.874 ± 0.008	0.714 ± 0.028
<b>3000 (M<sub>4</sub>)</b>							
VGG16	0.046 ± 0.064	7.670 ± 0.012	0.017 ± 0.000	0.849 ± 0.003	0.938 ± 0.001	0.764 ± 0.010	0.934 ± 0.003
ResNet101	0.054 ± 0.072	7.071 ± 0.092	0.019 ± 0.000	0.831 ± 0.010	0.913 ± 0.004	0.785 ± 0.017	0.877 ± 0.003
DenseNet169	0.001 ± 0.000	4.823 ± 0.030	0.049 ± 0.002	0.854 ± 0.004	0.936 ± 0.001	0.895 ± 0.002	0.813 ± 0.009

Table 6: Table showing the clustering and classification results when using Manhattan distance as distance measure for CT scan dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>200 (M<sub>1</sub>)</b>							
VGG16	0.129 ± 0.010	5.801 ± 0.588	0.024 ± 0.002	0.700 ± 0.036	0.853 ± 0.011	0.458 ± 0.105	0.941 ± 0.039
ResNet101	0.109 ± 0.000	5.591 ± 0.139	0.027 ± 0.001	0.683 ± 0.007	0.805 ± 0.005	0.470 ± 0.036	0.890 ± 0.024
DenseNet169	0.035 ± 0.007	3.680 ± 0.628	0.116 ± 0.063	0.618 ± 0.036	0.779 ± 0.030	0.927 ± 0.050	0.310 ± 0.120
<b>600 (M<sub>2</sub>)</b>							
VGG16	0.073 ± 0.051	6.708 ± 0.171	0.021 ± 0.002	0.744 ± 0.040	0.890 ± 0.003	0.538 ± 0.098	0.949 ± 0.022
ResNet101	0.072 ± 0.049	5.883 ± 0.116	0.024 ± 0.001	0.719 ± 0.010	0.842 ± 0.002	0.529 ± 0.049	0.909 ± 0.028
DenseNet169	0.020 ± 0.014	3.894 ± 0.161	0.081 ± 0.011	0.696 ± 0.028	0.822 ± 0.007	0.855 ± 0.053	0.538 ± 0.106
<b>1400 (M<sub>3</sub>)</b>							
VGG16	0.035 ± 0.047	7.300 ± 0.04	0.018 ± 0.000	0.777 ± 0.007	0.917 ± 0.001	0.594 ± 0.01	0.960 ± 0.001
ResNet101	0.037 ± 0.048	6.451 ± 0.012	0.021 ± 0.001	0.761 ± 0.003	0.873 ± 0.005	0.761 ± 0.003	0.891 ± 0.025
DenseNet169	0.011 ± 0.015	4.313 ± 0.144	0.068 ± 0.005	0.785 ± 0.007	0.889 ± 0.007	0.874 ± 0.025	0.695 ± 0.011
<b>3000 (M<sub>4</sub>)</b>							
VGG16	0.046 ± 0.063	7.753 ± 0.032	0.016 ± 0.000	0.823 ± 0.004	0.935 ± 0.001	0.694 ± 0.010	0.953 ± 0.001
ResNet101	0.038 ± 0.050	7.040 ± 0.087	0.018 ± 0.000	0.810 ± 0.009	0.909 ± 0.004	0.709 ± 0.023	0.911 ± 0.011
DenseNet169	0.010 ± 0.014	4.720 ± 0.164	0.051 ± 0.001	0.850 ± 0.009	0.933 ± 0.007	0.894 ± 0.022	0.806 ± 0.006

Table 7: Table showing the clustering and classification results when using Cosine distance as distance measure for CT scan dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>200 (M<sub>1</sub>)</b>							
VGG16	0.120 ± 0.011	6.754 ± 0.443	0.021 ± 0.001	0.740 ± 0.030	0.837 ± 0.019	0.831 ± 0.012	0.649 ± 0.072
ResNet101	0.147 ± 0.002	5.690 ± 0.122	0.028 ± 0.001	0.707 ± 0.014	0.812 ± 0.009	0.847 ± 0.026	0.568 ± 0.046
DenseNet169	0.009 ± 0.012	3.725 ± 0.210	0.039 ± 0.015	0.737 ± 0.019	0.818 ± 0.019	0.649 ± 0.021	0.825 ± 0.059
<b>600 (M<sub>2</sub>)</b>							
VGG16	0.063 ± 0.044	7.240 ± 0.227	0.019 ± 0.001	0.819 ± 0.011	0.893 ± 0.008	0.817 ± 0.016	0.821 ± 0.028
ResNet101	0.086 ± 0.062	6.268 ± 0.086	0.024 ± 0.001	0.787 ± 0.011	0.871 ± 0.005	0.806 ± 0.010	0.768 ± 0.028
DenseNet169	0.009 ± 0.012	4.313 ± 0.046	0.026 ± 0.004	0.784 ± 0.006	0.875 ± 0.004	0.686 ± 0.018	0.882 ± 0.009
<b>1400 (M<sub>3</sub>)</b>							
VGG16	0.035 ± 0.048	7.660 ± 0.067	0.017 ± 0.000	0.859 ± 0.005	0.929 ± 0.004	0.837 ± 0.013	0.880 ± 0.005
ResNet101	0.037 ± 0.049	6.682 ± 0.042	0.021 ± 0.000	0.828 ± 0.006	0.905 ± 0.004	0.823 ± 0.012	0.833 ± 0.020
DenseNet169	0.012 ± 0.016	4.619 ± 0.024	0.032 ± 0.002	0.824 ± 0.002	0.911 ± 0.001	0.767 ± 0.008	0.882 ± 0.006
<b>3000 (M<sub>4</sub>)</b>							
VGG16	0.047 ± 0.065	7.974 ± 0.031	0.016 ± 0.000	0.875 ± 0.004	0.945 ± 0.003	0.855 ± 0.013	0.894 ± 0.004
ResNet101	0.053 ± 0.072	7.205 ± 0.079	0.019 ± 0.000	0.859 ± 0.006	0.931 ± 0.005	0.834 ± 0.005	0.884 ± 0.007
DenseNet169	0.010 ± 0.013	5.105 ± 0.139	0.029 ± 0.001	0.848 ± 0.003	0.934 ± 0.001	0.791 ± 0.006	0.905 ± 0.003

### 5.3 Chest X-rays

The clustering and classification results on each distance measures for the Chest X-ray dataset are described as follows:

**Euclidean Distance (ED):** Table 8 presents a summary of our findings in ED. Notably, the most favorable results for DI, SI, and DBI were recorded at different stages of the mentoring process and with different neural network architectures. In  $M_3$ , the highest DI of 0.212 was achieved using ResNet101, while  $M_1$  with DenseNet169 yielded the best SI at 1.607. As for DBI,  $M_1$  with DenseNet169 also performed best with a score of 0.286. In the subsequent phase,  $M_2$  saw a decline in performance, with DI dropping to 0.040 with VGG16, SI decreasing to 0.233 with DenseNet169, and DBI increasing to 2.065 with DenseNet169. Returning to  $M_4$ , we witnessed a DI of 0.212 (ResNet101), DBI of 2.409 (DenseNet169), and SI of 0.166 (DenseNet169). Finally, in  $M_4$ , we observed a DI of 0.190 with ResNet101, DBI of 2.502 with DenseNet169, and SI of 0.141 with DenseNet169. The overall highest accuracy, area under the ROC curve, sensitivity, and specificity were observed at  $M_4$  (0.990, ResNet101),  $M_2$  (0.998, ResNet101),  $M_1$  (0.997, VGG16), and  $M_1$  (0.998, DenseNet169), respectively. ResNet101 had comparable scores in all four mentored cases. In VGG16 and DenseNet169, all four metrics gradually increased from  $M_1$  through  $M_4$ , except for sensitivity in VGG16 and specificity in DenseNet169, which remained constant. According to the method explained in the example, for ED, mentoring is needed up to  $M_3$  (760 data samples).

**Manhattan Distance (MD):** Table 9 presents a comprehensive overview of our results using MD. Notably, the most favorable values for DI, SI, and DBI were identified at different stages of the mentoring process, often associated with specific neural network architectures. In  $M_4$ , we attained the highest DI, reaching a value of 0.300, utilizing ResNet101. For SI, the best result was achieved in  $M_1$  using DenseNet169, with a score of 0.282. Regarding DBI,  $M_1$ , also employing DenseNet169, exhibited the best performance with a DBI of 1.622. Moving to  $M_2$ , there was a decline in performance across the board, with DI decreasing to 0.143 with ResNet101, SI dropping to 0.222 using DenseNet169, and DBI increasing to 2.108 with the same architecture. In  $M_3$ , the DI, DBI, and SI were recorded at 0.151 (ResNet101), 2.481 (DenseNet169), and 0.141 (DenseNet169), respectively. Finally, in  $M_4$ , we observed the highest DI of 0.300 with ResNet101, a DBI of 2.504 using DenseNet169, and an SI of 0.134 with the same DenseNet169 architecture. The overall highest accuracy, area under the ROC curve, sensitivity, and specificity were observed at  $M_4$  (0.988, ResNet101),  $M_3$  (0.999, ResNet101),  $M_1$  (0.998, VGG16), and  $M_1$  (0.998, DenseNet169), respectively. From  $M_1$  through  $M_4$ , AUC in ResNet101, SEN in VGG16 and ResNet101, and SPEC in DenseNet169 remained constant. Except for these cases, all other scores gradually increased, with the highest in  $M_4$ . For MD, mentoring is required up to  $M_3$  (760 data samples).

**Cosine Distance (CD):** Table 9 presents the results obtained using CD. Interestingly, the most favorable outcomes for DI, SI, and DBI were distributed across different mentoring phases and with varying neural network architectures. In  $M_4$ , the highest DI, measuring 0.190, was achieved

using ResNet101. The best SI was recorded in  $M_1$ , where DenseNet169 was utilized, with a score of 0.175. For DBI, once again in  $M_1$  and with the use of DenseNet169, the best performance was observed with a value of 2.160. In  $M_2$ , there was a noticeable decline in performance across these metrics. DI decreased to 0.073 with ResNet101, SI dropped to 0.147 using DenseNet169, and DBI increased to 2.391 with the same architecture. In  $M_3$ , the values were recorded at 0.082 (ResNet101) for DI, 2.424 (DenseNet169) for DBI, and 0.141 (DenseNet169) for SI. Finally, in  $M_4$ , we observed the highest DI of 0.190, utilizing ResNet101, a DBI of 2.456 with DenseNet169, and an SI of 0.136, again with DenseNet169. The overall highest accuracy, area under the ROC curve, sensitivity, and specificity were observed at  $M_4$  (0.985, VGG16/ResNet101),  $M_2$  (0.998, VGG16),  $M_4$  (0.972, VGG16/ResNet101), and  $M_2$  (0.999, ResNet101). A consistent performance from  $M_1$  through  $M_4$  was observed in all metrics for VGG16, AUC, and SPEC for ResNet101, and SPEC for DenseNet169. For CD, mentoring is required up to  $M_2$  (360 data samples).

For chest X-rays, experimenting by creating subclusters, we observed that the best DI (0.1902) in  $M_4$  with the corresponding Accuracy, Sensitivity, Specificity, and AUC of 0.984, 0.985, 0.982, and 0.986, respectively. However, the best DBI and SI of 3.119, and 0.089 was observed in  $M_2$ .



Table 8: Table showing the clustering and classification results when using Euclidean distance as distance measure for Chest X-ray dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>160 (M<sub>1</sub>)</b>							
VGG16	0.099 ± 0.140	2.673 ± 0.032	0.121 ± 0.005	0.926 ± 0.041	0.992 ± 0.007	0.997 ± 0.001	0.854 ± 0.082
ResNet101	0.135 ± 0.123	3.095 ± 0.001	0.090 ± 0.000	0.987 ± 0.001	0.998 ± 0.001	0.987 ± 0.001	0.987 ± 0.003
DenseNet169	0.005 ± 0.004	1.607 ± 0.041	0.286 ± 0.008	0.662 ± 0.025	0.946 ± 0.014	0.326 ± 0.049	0.998 ± 0.001
<b>360 (M<sub>2</sub>)</b>							
VGG16	0.040 ± 0.057	2.692 ± 0.026	0.118 ± 0.005	0.941 ± 0.039	0.994 ± 0.004	0.996 ± 0.001	0.885 ± 0.079
ResNet101	0.000 ± 0.000	3.096 ± 0.001	0.090 ± 0.000	0.987 ± 0.001	0.998 ± 0.001	0.988 ± 0.001	0.987 ± 0.002
DenseNet169	0.005 ± 0.004	2.065 ± 0.150	0.233 ± 0.015	0.772 ± 0.016	0.979 ± 0.006	0.548 ± 0.031	0.997 ± 0.000
<b>760 (M<sub>3</sub>)</b>							
VGG16	0.100 ± 0.142	2.714 ± 0.002	0.114 ± 0.000	0.975 ± 0.003	0.998 ± 0.000	0.995 ± 0.001	0.954 ± 0.006
ResNet101	0.212 ± 0.078	3.099 ± 0.001	0.090 ± 0.000	0.989 ± 0.001	0.998 ± 0.001	0.987 ± 0.002	0.991 ± 0.001
DenseNet169	0.016 ± 0.013	2.409 ± 0.097	0.166 ± 0.018	0.899 ± 0.029	0.992 ± 0.003	0.804 ± 0.061	0.994 ± 0.002
<b>1510 (M<sub>4</sub>)</b>							
VGG16	0.142 ± 0.122	2.724 ± 0.004	0.112 ± 0.001	0.980 ± 0.003	0.998 ± 0.001	0.995 ± 0.000	0.965 ± 0.001
ResNet101	0.190 ± 0.058	3.103 ± 0.000	0.090 ± 0.000	0.990 ± 0.001	0.998 ± 0.001	0.987 ± 0.002	0.993 ± 0.002
DenseNet169	0.010 ± 0.000	2.502 ± 0.006	0.141 ± 0.003	0.942 ± 0.008	0.995 ± 0.001	0.892 ± 0.017	0.993 ± 0.001

Table 9: Table showing the clustering and classification results when using Manhattan distance as distance measure for Chest X-ray dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>160 (M<sub>1</sub>)</b>							
VGG16	0.040 ± 0.056	2.652 ± 0.026	0.124 ± 0.004	0.900 ± 0.031	0.990 ± 0.000	0.998 ± 0.001	0.802 ± 0.062
ResNet101	0.294 ± 0.208	3.092 ± 0.000	0.090 ± 0.000	0.981 ± 0.001	0.998 ± 0.000	0.994 ± 0.000	0.968 ± 0.001
DenseNet169	0.005 ± 0.004	1.622 ± 0.032	0.282 ± 0.006	0.671 ± 0.019	0.951 ± 0.012	0.345 ± 0.036	0.998 ± 0.001
<b>360 (M<sub>2</sub>)</b>							
VGG16	0.040 ± 0.060	2.689 ± 0.027	0.119 ± 0.004	0.930 ± 0.033	0.994 ± 0.003	0.997 ± 0.001	0.863 ± 0.068
ResNet101	0.143 ± 0.202	3.093 ± 0.002	0.090 ± 0.000	0.983 ± 0.001	0.998 ± 0.000	0.994 ± 0.001	0.972 ± 0.002
DenseNet169	0.026 ± 0.026	2.108 ± 0.090	0.222 ± 0.004	0.796 ± 0.024	0.983 ± 0.002	0.596 ± 0.049	0.997 ± 0.001
<b>760 (M<sub>3</sub>)</b>							
VGG16	0.000 ± 0.000	2.712 ± 0.004	0.115 ± 0.001	0.965 ± 0.006	0.998 ± 0.000	0.997 ± 0.001	0.933 ± 0.012
ResNet101	0.151 ± 0.213	3.095 ± 0.001	0.090 ± 0.000	0.986 ± 0.000	0.999 ± 0.000	0.994 ± 0.001	0.978 ± 0.000
DenseNet169	0.009 ± 0.002	2.481 ± 0.017	0.141 ± 0.005	0.954 ± 0.009	0.997 ± 0.000	0.916 ± 0.019	0.993 ± 0.001
<b>1510 (M<sub>4</sub>)</b>							
VGG16	0.100 ± 0.141	2.722 ± 0.005	0.113 ± 0.000	0.972 ± 0.001	0.998 ± 0.001	0.996 ± 0.001	0.948 ± 0.001
ResNet101	0.300 ± 0.212	3.099 ± 0.002	0.090 ± 0.000	0.988 ± 0.002	0.999 ± 0.000	0.993 ± 0.000	0.983 ± 0.003
DenseNet169	0.011 ± 0.000	2.504 ± 0.002	0.134 ± 0.001	0.966 ± 0.002	0.998 ± 0.000	0.939 ± 0.006	0.993 ± 0.001

Table 10: Table showing the clustering and classification results when using Cosine distance as distance measure for Chest X-ray dataset. The standard deviation denotes the deviation of results when experimenting on three independent subsets of the data.

DL models	Dunn's index	Davies Bouldin Index	Silhouette Index	Accuracy	AUC	Sensitivity	Specificity
<b>160 (M<sub>1</sub>)</b>							
VGG16	0.042 ± 0.000	2.753 ± 0.002	0.107 ± 0.000	0.982 ± 0.001	0.995 ± 0.001	0.967 ± 0.002	0.998 ± 0.000
ResNet101	0.081 ± 0.016	3.150 ± 0.022	0.088 ± 0.001	0.968 ± 0.009	0.997 ± 0.000	0.936 ± 0.019	0.999 ± 0.001
DenseNet169	0.033 ± 0.023	2.160 ± 0.075	0.175 ± 0.008	0.903 ± 0.011	0.986 ± 0.003	0.813 ± 0.022	0.994 ± 0.000
<b>360 (M<sub>2</sub>)</b>							
VGG16	0.046 ± 0.007	2.751 ± 0.002	0.108 ± 0.000	0.983 ± 0.001	0.998 ± 0.000	0.969 ± 0.001	0.998 ± 0.000
ResNet101	0.073 ± 0.008	3.129 ± 0.004	0.090 ± 0.000	0.978 ± 0.002	0.997 ± 0.000	0.956 ± 0.004	0.999 ± 0.000
DenseNet169	0.032 ± 0.023	2.391 ± 0.020	0.147 ± 0.003	0.960 ± 0.005	0.995 ± 0.002	0.927 ± 0.009	0.994 ± 0.000
<b>760 (M<sub>3</sub>)</b>							
VGG16	0.061 ± 0.018	2.750 ± 0.001	0.108 ± 0.000	0.984 ± 0.001	0.997 ± 0.000	0.970 ± 0.002	0.997 ± 0.000
ResNet101	0.082 ± 0.019	3.118 ± 0.001	0.089 ± 0.000	0.982 ± 0.002	0.997 ± 0.000	0.966 ± 0.004	0.998 ± 0.000
DenseNet169	0.038 ± 0.029	2.424 ± 0.012	0.141 ± 0.002	0.966 ± 0.002	0.996 ± 0.001	0.940 ± 0.005	0.993 ± 0.001
<b>1510 (M<sub>4</sub>)</b>							
VGG16	0.102 ± 0.033	2.748 ± 0.002	0.108 ± 0.000	0.985 ± 0.001	0.998 ± 0.001	0.972 ± 0.003	0.998 ± 0.001
ResNet101	0.190 ± 0.058	3.115 ± 0.002	0.089 ± 0.000	0.985 ± 0.001	0.998 ± 0.001	0.972 ± 0.002	0.998 ± 0.001
DenseNet169	0.053 ± 0.014	2.456 ± 0.015	0.136 ± 0.002	0.970 ± 0.003	0.997 ± 0.001	0.946 ± 0.006	0.994 ± 0.001

## 5.4 Comparison

Rich literature implementing state-of-the-art DL model exists [76] [26] [27] [77] [78] [79] [80]. These are our previous works (both reviews and original research), where we show/implement the use cases of DL techniques for various medical imaging analysis tasks. However, to the best of our knowledge, literature lacks the utilization of AL methodologies specifically focused for epidemics – for all three data modalities. Therefore, we conduct a comparative evaluation of our framework by juxtaposing it with prior studies that heavily relied on fully labeled datasets. Table 11, Table 12, and Table 13 lists the classification performance of previous work that use similar dataset as ours for cough sound, CT scans, and Chest X-ray dataset, respectively. Our results demonstrated comparable performance with other DL models in the literature that utilized a 100% labeled dataset, employing only 40%, 30%, 33% of the total labeled dataset for training on cough sound, CT scans, and X-rays, respectively. A direct and fair comparison is challenging due to the selected works not using the same dataset as ours, although they included at least one type of dataset like ours. Our future work will incorporate a fair comparison technique, as discussed in Chapter 6.

Table 11: Comparison of the classification results for cough sound among previous works that used DL models and our AL framework.

Authors	Methods	AUC	Accuracy	Specificity	Sensitivity
Pahar [40]	ResNet50	0.976	0.953	0.980	0.930
Chowdhury [81]	Recursive Feature Elimination with Cross-Validation + Extremely Randomized Trees	0.640	-	-	0.580
Meister [41]	Ceprtal features (Random Forest)	0.775	0.770	-	-
Feng [82]	Recurrent Neural Networks (RNN)	-	0.995	-	-
Sezer [83]	CSCCov19Net (Custom Network)	0.795	0.748	-	-
Wall [84]	Ensemble (Attention, Neural Networks)	-	0.975	0.942	1.000
Arup [42]	Support Vector Machines (SVM)	0.980	0.969	0.975	0.967
Kumar [85]	Convolutional Neural Networks	-	0.846	0.812	0.834
Ours	Active Learning	0.760	0.680	0.980	0.770

Table 12: Comparison of the classification results for CT scans among previous works that used DL and our AL framework.

Authors	Methods	AUC	Accuracy	Specificity	Sensitivity
Gunraj [67]	COVID-NET CT-2	-	0.980	0.990	0.990
Loddo [86]	VGG19	-	0.980	0.990	0.970
Zhao [87]	ResNet V2	-	0.980	0.990	-
Zhang [88]	Swin-B Transformer	-	0.943	-	0.938
Hammam [89]	CovidResNet	-	0.824	0.911	-
Zhang [90]	ResNet50 + Attention + SSL	0.932	0.901	0.889	0.914
Hasija [91]	Custom CNN	-	-	0.991	0.981
Garg [92]	Multi-scale ResNet50	-	0.888	-	0.890
Ours	Active Learning	0.940	0.870	0.900	0.850

Table 13: Comparison of the classification results for CXRs among previous works that used DL and our AL framework.

Authors	Methods	AUC	Accuracy	Specificity	Sensitivity
Santosh [65]	ChexNet	0.99	0.98	0.98	0.99
Mahmud [93]	CovXNet	0.96	0.97	0.94	0.97
Asnaoui [94]	Inception ResNet V2	-	0.92	0.96	0.92
Ucar [95]	Bayes-SqueezeNet	-	0.98	0.99	-
Panwar [96]	nCOVnet	0.880	0.881	0.785	0.976
Brunese [34]	VGG16	-	0.980	0.940	0.870
Samira [97]	CoviNet	0.990	0.986	0.987	0.985
Shelke [98]	DenseNet161	-	0.989	0.990	0.980
Ours	Active Learning	0.990	0.960	0.990	0.990

## Chapter 6

### 6. Conclusion

---

*Summary:* This chapter concludes the thesis by briefly describing the proposed AL framework, observed results, and future works.

**Key topics:** Conclusion and future works

---

Future epidemics are inevitable, and Covid-19 is an example. Although the predictive modeling offers a promising approach to anticipate future epidemics, it faces challenges for robust predictions due to the presence of numerous unforeseen events and factors that resist parameterization within mathematical frameworks [9]. Also, the traditional DL models are not practical to use at epidemics as we cannot wait for months and even years to amass data for training our models. In such cases, there is a crucial need for specialized data-driven tools tailored to address challenges related to data scarcity, particularly in situations like epidemics.

In this thesis, we have developed an AI guided tool/framework that can commence training from the first day of epidemics using the concept of AL. The framework has been validated in three distinct Covid-19 imaging datasets: Cough sound, CT scans, and Chest X-rays. With this work, we have demonstrated the proof-of-concept that leveraging AL allows training to begin early and can possibly mitigate the risk from any upcoming epidemic by having tool for quick screening of the infected cases. Despite having DL tools readily available for mass screening, these models are of no use at the time of epidemic due to the requirement of completely labeled dataset. Therefore, AL is the must at the emergencies such as epidemics [8] [9].

In the proposed framework, the training initiates with a limited set of labeled data, and mentoring for the DL models commences, with Expert intervention occurring only when the model makes errors. Like parents guiding their children until they reach adulthood (i.e., 18 years), the expert's mentor the DL models for a specified number of data samples within a specific timeframe. This early initiation of the learning process allows the model to swiftly acquire knowledge to define the required hyperplane. To streamline the computational complexity of the framework, we employed k-way n-shot learning within an unsupervised learning technique. The framework was validated

on cough sound, CT scans, and X-ray dataset consisting of 1364, 10,000, and 4712 images, respectively. To evaluate the framework, we have used three clustering indices: DI, DBI, and SI, and four classification metrics: ACC, SEN, SPEC, and AUC value.

The DL models underwent iterative training in four mentoring cases ( $M_1$ ,  $M_2$ ,  $M_3$ , and  $M_4$ ) to assess their performance using various classification metrics. The goal was to determine the optimal point for concluding mentoring. In the case of cough sound, we have achieved the highest values for AUC, ACC, SEN, and SPEC at 0.760, 0.680, 0.770, and 0.980, respectively, utilizing only 40% of the total labeled dataset. For CT scans, VGG16 have achieved an AUC of 0.940, ACC of 0.870, SPEC of 0.900, and SEN of 0.850 in  $M_4$ . Despite lower metrics compared to some studies due to less mentoring, this approach, using only 30% of the labeled dataset, demonstrated favorable comparisons to models trained with 100% labeled data. Similarly, in the mentoring process for chest X-rays, high classification metrics were achieved, including AUC of 0.999, ACC of 0.998, SPEC of 0.998, and SEN of 0.998, with just 33% of the total labeled dataset. Previously, we have implemented AL to querying in noisy dataset [99], and published extended version of this work as a book [100].

This is the initial work that validates the proposed framework, providing an assertion that AL can be used for training DL models for screening medical images. This lays the groundwork to add varieties of work on top of this framework. Currently, mentoring occurs for individual test dataset, making algorithm slower; however, in future work, one might implement mentoring in batches to swiften the training process. Querying in batches is one of the major challenges of AL [101]. In this work, each new data sample are assigned to the nearest subcluster using distance measure; however, defining a distance threshold and creating a new subcluster for data sample exceeding this threshold might improve the performance of the framework. Also, the work can directly be extended to multimodal learning by combining the three datasets into a multimodal representation to classify the infected cases from normal ones [102]. Furthermore, it is crucial to recognize that, despite the mentor-guided approach, the model's performance may not reach perfection in the future. The data distribution could shift with the influx of more data, introducing new challenges. Consequently, our forthcoming efforts involve integrating an agent trained to emulate the behavior of an expert or experts when intervention is necessary, a concept known as Imitation Learning (IL) [103]. This integration aims to enable agent-based mentoring once the mentoring phase by the human expert is completed. To achieve this, we have established a policy network designed to discern the most suitable action (intervention or inaction) for each data sample by observing the actions taken by the human expert during the mentoring phase. This ongoing learning capability, even post-expert mentoring, will empower the model to refine its knowledge and adapt to evolving circumstances even when human intervention concludes. It should be noted that deploying this framework in the real-world application requires it to be explainable to doctors and general population [104]. Not only should it be explainable, but also secured so that the classified information of patients remains secured – using Federated Learning [105]. Additionally, one can explore developing a lightweight custom CNN that swiftly adapts to new samples as the dataset expands over time [80].

## References

- [1] S. Nakarmi and KC Santosh, “Active Learning to Minimize the Risk from Future Epidemics,” in *2023 IEEE Conference on Artificial Intelligence (CAI)*, IEEE, Jun. 2023, pp. 329–330. doi: 10.1109/CAI54212.2023.00145.
- [2] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, “The rise of deep learning in drug discovery,” *Drug Discovery Today*, vol. 23, no. 6. 2018. doi: 10.1016/j.drudis.2018.01.039.
- [3] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani, and A. Telenti, “A primer on deep learning in genomics,” *Nat Genet*, vol. 51, no. 1, 2019, doi: 10.1038/s41588-018-0295-5.
- [4] S. J. Maceachern and N. D. Forkert, “Machine learning for precision medicine,” *Genome*, vol. 64, no. 4. 2021. doi: 10.1139/gen-2020-0131.
- [5] G. Litjens Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafourian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez., “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42. 2017. doi: 10.1016/j.media.2017.07.005.
- [6] KC Santosh and S. Ghosh, “Covid-19 Imaging Tools: How Big Data is Big?,” *J Med Syst*, vol. 45, no. 7, 2021, doi: 10.1007/s10916-021-01747-2.
- [7] D. Das, KC Santosh, and U. Pal, “Cross-population train/test deep learning model: Abnormality screening in chest x-rays,” in *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, 2020. doi: 10.1109/CBMS49503.2020.00103.
- [8] KC Santosh, “AI-Driven Tools for Coronavirus Outbreak: Need of Active Learning and Cross-Population Train/Test Models on Multitudinal/Multimodal Data,” *J Med Syst*, vol. 44, no. 5, 2020, doi: 10.1007/s10916-020-01562-1.
- [9] KC Santosh, “COVID-19 Prediction Models and Unexploited Data,” *J Med Syst*, vol. 44, no. 9, 2020, doi: 10.1007/s10916-020-01645-z.
- [10] A. Vlachos, “A stopping criterion for active learning,” *Comput Speech Lang*, vol. 22, no. 3, 2008, doi: 10.1016/j.csl.2007.12.001.
- [11] J. Zhu, H. Wang, E. Hovy, and M. Ma, “Confidence-based stopping criteria for active learning for data annotation,” *ACM Transactions on Speech and Language Processing*, vol. 6, no. 3, 2010, doi: 10.1145/1753783.1753784.
- [12] D. Angluin, “Queries and Concept Learning,” *Mach Learn*, vol. 2, no. 4, 1988, doi: 10.1023/A:1022821128753.
- [13] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Mach Learn*, vol. 15, no. 2, 1994, doi: 10.1007/bf00993277.



- [14] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 1994. doi: 10.1007/978-1-4471-2099-5\_1.
- [15] R. Hao, K. Namdar, L. Liu, and F. Khalvati, "A Transfer Learning–Based Active Learning Framework for Brain Tumor Classification," *Front Artif Intell*, vol. 4, 2021, doi: 10.3389/frai.2021.635766.
- [16] D. H. M. Nguyen and J. D. Patrick, "Supervised machine learning and active learning in classification of radiology reports," *Journal of the American Medical Informatics Association*, vol. 21, no. 5, 2014, doi: 10.1136/amiajnl-2013-002516.
- [17] L. G. Batista, P. H. Bugatti, and P. T. M. Saito, "Classification of Skin Lesion through Active Learning Strategies," *Comput Methods Programs Biomed*, vol. 226, 2022, doi: 10.1016/j.cmpb.2022.107122.
- [18] J. Liu, L. Cao, and Y. Tian, "Deep Active Learning for Effective Pulmonary Nodule Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020. doi: 10.1007/978-3-030-59725-2\_59.
- [19] Q. Jin, S. Li, X. Du, M. Yuan, M. Wang, and Z. Song, "Density-based one-shot active learning for image segmentation," *Eng Appl Artif Intell*, vol. 126, 2023, doi: 10.1016/j.engappai.2023.106805.
- [20] I. Qureshi, J. Ma, and Q. Abbas, "Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning," *Multimed Tools Appl*, vol. 80, no. 8, 2021, doi: 10.1007/s11042-020-10238-4.
- [21] W. Shao, L. Sun, and D. Zhang, "Deep active learning for nucleus classification in pathology images," in *Proceedings - International Symposium on Biomedical Imaging*, 2018. doi: 10.1109/ISBI.2018.8363554.
- [22] W. H. and K.-H. J. Sejin Park, "Semi-supervised reinforced active learning for pulmonary nodule detection in chest X-rays.," 2022.
- [23] X. Wu, C. Chen, M. Zhong, J. Wang, and J. Shi, "COVID-AL: The diagnosis of COVID-19 with deep active learning," *Med Image Anal*, vol. 68, 2021, doi: 10.1016/j.media.2020.101913.
- [24] J. E. Iglesias, E. Konukoglu, A. Montillo, Z. Tu, and A. Criminisi, "Combining generative and discriminative models for semantic segmentation of CT scans via active learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011. doi: 10.1007/978-3-642-22092-0\_3.

- [25] H. T. Hoa, T. V. An, and T. H. Dat, "Semi-Supervised Tree Support Vector Machine for online cough recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2011. doi: 10.21437/interspeech.2011-489.
- [26] KC Santosh, S. Ghosh, and D. Ghoshroy, "Deep Learning for Covid-19 Screening Using Chest X-Rays in 2020: A Systematic Review," *Intern J Pattern Recognit Artif Intell*, vol. 36, no. 5, 2022, doi: 10.1142/S0218001422520103.
- [27] KC Santosh, D. GhoshRoy, and S. Nakarmi, "A Systematic Review on Deep Structured Learning for COVID-19 Screening Using Chest CT from 2020 to 2022," *Healthcare (Switzerland)*, vol. 11, no. 17. 2023. doi: 10.3390/healthcare11172388.
- [28] M. M. Islam, F. Karray, R. Alhajj, and J. Zeng, "A Review on Deep Learning Techniques for the Diagnosis of Novel Coronavirus (COVID-19)," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3058537.
- [29] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and M. Chowdhury, "A review of deep learning-based detection methods for COVID-19," *Computers in Biology and Medicine*, vol. 143. 2022. doi: 10.1016/j.compbiomed.2022.105233.
- [30] Y. H. Bhosale and K. S. Patnaik, "Application of Deep Learning Techniques in Diagnosis of Covid-19 (Coronavirus): A Systematic Review," *Neural Processing Letters*, vol. 55, no. 3. 2023. doi: 10.1007/s11063-022-11023-0.
- [31] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep Learning applications for COVID-19," *J Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-020-00392-9.
- [32] A. M. Ismael and A. Şengür, "Deep learning approaches for COVID-19 detection based on chest X-ray images," *Expert Syst Appl*, vol. 164, 2021, doi: 10.1016/j.eswa.2020.114054.
- [33] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-Rays," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3003810.
- [34] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, "Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays," *Comput Methods Programs Biomed*, vol. 196, 2020, doi: 10.1016/j.cmpb.2020.105608.
- [35] G. I. Okolo, S. Katsigiannis, T. Althobaiti, and N. Ramzan, "On the use of deep learning for imaging-based COVID-19 detection using chest X-rays," *Sensors*, vol. 21, no. 17, 2021, doi: 10.3390/s21175702.
- [36] Y. Khurana and U. Soni, "Leveraging deep learning for COVID-19 diagnosis through chest imaging," *Neural Comput Appl*, vol. 34, no. 16, 2022, doi: 10.1007/s00521-022-07250-0.

- [37] M. Canayaz, S. Şehribanoğlu, R. Özdağ, and M. Demir, “COVID-19 diagnosis on CT images with Bayes optimization-based deep neural networks and machine learning algorithms,” *Neural Comput Appl*, vol. 34, no. 7, 2022, doi: 10.1007/s00521-022-07052-4.
- [38] R. T. Subhalakshmi, S. A. alias Balamurugan, and S. Sasikala, “Deep learning based fusion model for COVID-19 diagnosis and classification using computed tomography images,” *Concurr Eng Res Appl*, vol. 30, no. 1, 2022, doi: 10.1177/1063293X211021435.
- [39] W. Zouch, Wassim Zouch, Dhousha Sagga, Amira Echtioui, Rafik Khemakhem, Mohamed Ghorbel, Chokri Mhiri, and Ahmed Ben Hamida, “Detection of COVID-19 from CT and Chest X-ray Images Using Deep Learning Models,” *Ann Biomed Eng*, vol. 50, no. 7, 2022, doi: 10.1007/s10439-022-02958-5.
- [40] M. Pahar, M. Klopper, R. Warren, and T. Niesler, “COVID-19 cough classification using machine learning and global smartphone recordings,” *Comput Biol Med*, vol. 135, 2021, doi: 10.1016/j.combiomed.2021.104572.
- [41] J. A. Meister, K. A. Nguyen, and Z. Luo, “Audio Feature Ranking for Sound-Based COVID-19 Patient Detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022. doi: 10.1007/978-3-031-16474-3\_13.
- [42] A. Anupam, N. J. Mohan, S. Sahoo, and S. Chakraborty, “Preliminary diagnosis of COVID-19 based on cough sounds using machine learning algorithms,” in *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, 2021. doi: 10.1109/ICICCS51141.2021.9432324.
- [43] M. A. Hussain, Mohammad Arafat Hussain a, Zahra Mirikharaji a, Mohammad Momeny b, Mahmoud Marhamati c, Ali Asghar Neshat c, Rafeef Garbi d, and Ghassan Hamarneh., “Active deep learning from a noisy teacher for semi-supervised 3D image segmentation: Application to COVID-19 pneumonia infection in CT,” *Computerized Medical Imaging and Graphics*, vol. 102, 2022, doi: 10.1016/j.compmedimag.2022.102127.
- [44] X. Li, W. Cui, and F. Zhang, “Who was the first doctor to report the COVID-19 outbreak in Wuhan, China?,” *Journal of Nuclear Medicine*, vol. 61, no. 6. 2020. doi: 10.2967/jnumed.120.247262.
- [45] Y. Bengio, Y. Lecun, and G. Hinton, “Deep learning for AI,” *Commun ACM*, vol. 64, no. 7, 2021, doi: 10.1145/3448250.
- [46] D. Learning, “Deep Learning - Goodfellow,” *Nature*, vol. 26, no. 7553, 2016.
- [47] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.90.

- [48] W. Xu, Y. L. Fu, and D. Zhu, “ResNet and its application to medical image processing: Research progress and challenges,” *Comput Methods Programs Biomed*, vol. 240, 2023, doi: 10.1016/j.cmpb.2023.107660.
- [49] S. Serte, A. Serener, and F. Al-Turjman, “Deep learning in medical imaging: A brief review,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 10, 2022, doi: 10.1002/ett.4080.
- [50] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.243.
- [51] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [52] A. Holzinger, “Interactive machine learning for health informatics: when do we need the human-in-the-loop?,” *Brain Inform*, vol. 3, no. 2, 2016, doi: 10.1007/s40708-016-0042-6.
- [53] Robert (Munro) Monarch, *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning, 2021.
- [54] D. E. Szwedo, E. T. Hessel, E. L. Loeb, C. A. Hafen, and J. P. Allen, “Adolescent support seeking as a path to adult functional independence,” *Dev Psychol*, vol. 53, no. 5, 2017, doi: 10.1037/dev0000277.
- [55] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a Few Examples: A Survey on Few-shot Learning,” *ACM Comput Surv*, vol. 53, no. 3, 2020, doi: 10.1145/3386252.
- [56] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, “Learning to Compare: Relation Network for Few-Shot Learning,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00131.
- [57] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Trans Pattern Anal Mach Intell*, vol. 28, no. 4, 2006, doi: 10.1109/TPAMI.2006.79.
- [58] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.
- [59] Z. Ji, X. Chai, Y. Yu, Y. Pang, and Z. Zhang, “Improved prototypical networks for few-Shot learning,” *Pattern Recognit Lett*, vol. 140, 2020, doi: 10.1016/j.patrec.2020.07.015.
- [60] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in *Advances in Neural Information Processing Systems*, 2016.
- [61] C. W. M. Ong, Catherine Wei Min Ong, Giovanni Battista Migliori, Mario Raviglione, Gavin MacGregor-Skinner, Giovanni Sotgiu, Jan-Willem Alffenaar, Simon Tiberi, Cornelia Adlhoch, Tonino Alonzi, Sophia Archuleta, Sergio Brusin, Emmanuelle Cambau, Maria

- Rosaria Capobianchi, Concetta Castilletti, Rosella Centis, Daniela M. Cirillo, Lia D'Ambrosio, Giovanni Delogu, Susanna M.R. Esposito, Jose Figueroa, Jon S. Friedland, Benjamin Choon Heng Ho, Giuseppe Ippolito, Mateja Jankovic, Hannah Yejin Kim, Senia Rosales Klintz, Csaba Ködmön, Eleonora Lalle, Yee Sin Leo, Chi-Chiu Leung, Anne-Grete Märtson, Mario Giovanni Melazzini, Saeid Najafi Fard, Pasi Penttinen, Linda Petrone, Elisa Petruccioli, Emanuele Pontali, Laura Saderi, Miguel Santin, Antonio Spanevello, Reinout van Crevel, Marieke J. van der Werf, Dina Visca, Miguel Viveiros, Jean-Pierre Zellweger, Alimuddin Zumla, and Delia Goletti ., “Epidemic and pandemic viral infections: Impact on tuberculosis and the lung,” *European Respiratory Journal*, vol. 56, no. 4. 2020. doi: 10.1183/13993003.01727-2020.
- [62] B. Kelly, “The chest radiograph,” *Ulster Medical Journal*, vol. 81, no. 3, 2012, doi: 10.1016/b978-0-323-39952-4.00004-4.
- [63] Miriam Harris ,Amy Qi,Luke Jeagal,Nazi Torabi,Dick Menzies,Alexei Korobitsyn, Madhukar Pai, and Ruvandhi R. Nathavitharana, Faiz Ahmad Khan, “A systematic review of the diagnostic accuracy of artificial intelligence-based computer programs to analyze chest x-rays for pulmonary tuberculosis,” *PLoS One*, vol. 14, no. 9, 2019, doi: 10.1371/journal.pone.0221339.
- [64] G. M. F. Wallace, J. H. Winter, J. E. Winter, A. Taylor, T. W. Taylor, and R. C. Cameron, “Chest X-rays in COPD screening: Are they worthwhile?,” *Respir Med*, vol. 103, no. 12, 2009, doi: 10.1016/j.rmed.2009.07.001.
- [65] KC Santosh and S. Ghosh, “CheXNet for the Evidence of Covid-19 Using 2.3K Positive Chest X-rays,” in *Communications in Computer and Information Science*, 2022. doi: 10.1007/978-3-031-07005-1\_4.
- [66] D. Kermany, K. Zhang, M. Goldbaum, and others, “Labeled optical coherence tomography (oct) and chest x-ray images for classification,” *Mendeley data*, vol. 2, no. 2, 2018.
- [67] H. Gunraj, A. Sabri, D. Koff, and A. Wong, “COVID-Net CT-2: Enhanced Deep Neural Networks for Detection of COVID-19 From Chest CT Images Through Bigger, More Diverse Learning,” *Front Med (Lausanne)*, vol. 8, 2022, doi: 10.3389/fmed.2021.729287.
- [68] L. Orlandic, T. Teijeiro, and D. Atienza, “The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms,” *Sci Data*, vol. 8, no. 1, 2021, doi: 10.1038/s41597-021-00937-4.
- [69] Neeraj Sharma, Prashant Krishnan, Rohit Kumar, Shreyas Ramoji, Srikanth Raj Chetupalli, Nirmala R., Prasanta Kumar Ghosh, and Sriram Ganapathy., “Coswara - A database of breathing, cough, and voice sounds for COVID-19 diagnosis,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020. doi: 10.21437/Interspeech.2020-2768.

- [70] CDC, “Symptoms of Coronavirus Disease 2019 (COVID-19) | CDC,” *Centers for Disease Control and Prevention*, 2020.
- [71] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nietok., “librosa: Audio and Music Signal Analysis in Python,” in *Proceedings of the 14th Python in Science Conference*, 2015. doi: 10.25080/majora-7b98e3ed-003.
- [72] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, vol. 3, no. 3, 1973, doi: 10.1080/01969727308546046.
- [73] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Trans Pattern Anal Mach Intell*, vol. PAMI-1, no. 2, 1979, doi: 10.1109/TPAMI.1979.4766909.
- [74] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J Comput Appl Math*, vol. 20, no. C, 1987, doi: 10.1016/0377-0427(87)90125-7.
- [75] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognit*, vol. 30, no. 7, 1997, doi: 10.1016/S0031-3203(96)00142-2.
- [76] KC Santosh, N. Rasmussen, M. Mamun, and S. Aryal, “A systematic review on cough sound analysis for Covid-19 diagnosis and screening: is my cough sound COVID-19?,” *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/peerj-cs.958.
- [77] A. Khanal, R. Rizk, and KC Santosh, “Ensemble Deep Convolutional Neural Network to Identify Fractured Limbs using CT Scans,” in *Proceedings - 2023 IEEE Conference on Artificial Intelligence, CAI 2023*, 2023. doi: 10.1109/CAI54212.2023.00075.
- [78] KC Santosh, S. Allu, S. Rajaraman, and S. Antani, “Advances in Deep Learning for Tuberculosis Screening using Chest X-rays: The Last 5 Years Review,” *J Med Syst*, vol. 46, no. 11, 2022, doi: 10.1007/s10916-022-01870-8.
- [79] D. Das, KC Santosh, and U. Pal, “Inception-based deep learning architecture for tuberculosis screening using chest x-rays,” in *Proceedings - International Conference on Pattern Recognition*, 2020. doi: 10.1109/ICPR48806.2021.9412748.
- [80] D. Das, KC Santosh, and U. Pal, “Truncated inception net: COVID-19 outbreak screening using chest X-rays,” *Phys Eng Sci Med*, vol. 43, no. 3, 2020, doi: 10.1007/s13246-020-00888-x.
- [81] N. K. Chowdhury, M. A. Kabir, M. M. Rahman, and S. M. S. Islam, “Machine learning for detecting COVID-19 from cough sounds: An ensemble-based MCDM method,” *Comput Biol Med*, vol. 145, 2022, doi: 10.1016/j.compbimed.2022.105405.
- [82] K. Feng, F. He, J. Steinmann, and I. Demirkiran, “Deep-learning based approach to identify covid-19,” in *Conference Proceedings - IEEE SOUTHEASTCON*, 2021. doi: 10.1109/SoutheastCon45413.2021.9401826.

- [83] S. Ulukaya, A. A. Sarıca, O. Erdem, and A. Karaali, "MSCCov19Net: multi-branch deep learning model for COVID-19 detection from cough sounds," *Med Biol Eng Comput*, vol. 61, no. 7, 2023, doi: 10.1007/s11517-023-02803-4.
- [84] C. Wall, L. Zhang, Y. Yu, A. Kumar, and R. Gao, "A Deep Ensemble Neural Network with Attention Mechanisms for Lung Abnormality Classification Using Audio Inputs," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155566.
- [85] S. Kumar, S. K. Gupta, V. Kumar, M. Kumar, M. K. Chaube, and N. S. Naik, "Ensemble multimodal deep learning for early diagnosis and accurate classification of COVID-19," *Computers and Electrical Engineering*, vol. 103, 2022, doi: 10.1016/j.compeleceng.2022.108396.
- [86] A. Loddo, F. Pili, and C. Di Ruberto, "Deep learning for covid-19 diagnosis from ct images," *Applied Sciences (Switzerland)*, vol. 11, no. 17, 2021, doi: 10.3390/app11178227.
- [87] W. Zhao, W. Jiang, and X. Qiu, "Deep learning for COVID-19 detection based on CT images," *Sci Rep*, vol. 11, no. 1, 2021, doi: 10.1038/s41598-021-93832-2.
- [88] L. Zhang and Y. Wen, "A transformer-based framework for automatic COVID19 diagnosis in chest CTs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021. doi: 10.1109/ICCVW54120.2021.00063.
- [89] H. Alshazly, C. Linse, M. Abdalla, E. Barth, and T. Martinetz, "COVID-Nets: deep CNN architectures for detecting COVID-19 using chest CT scans," *PeerJ Comput Sci*, vol. 7, 2021, doi: 10.7717/peerj-cs.655.
- [90] Y. Zhang, L. Su, Z. Liu, W. Tan, Y. Jiang, and C. Cheng, "A semi-supervised learning approach for COVID-19 detection from chest CT scans," *Neurocomputing*, vol. 503, 2022, doi: 10.1016/j.neucom.2022.06.076.
- [91] S. Hasija, P. Akash, M. Bhargav Hemanth, A. Kumar, and S. Sharma, "A novel approach for detection of COVID-19 and Pneumonia using only binary classification from chest CT-scans," *Neuroscience Informatics*, vol. 2, no. 4, 2022, doi: 10.1016/j.neuri.2022.100069.
- [92] P. Garg, R. Ranjan, K. Upadhyay, M. Agrawal, and D. Deepak, "Multi-scale residual network for covid-19 diagnosis using CT-scans," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021. doi: 10.1109/ICASSP39728.2021.9414426.
- [93] T. Mahmud, M. A. Rahman, and S. A. Fattah, "CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization," *Comput Biol Med*, vol. 122, 2020, doi: 10.1016/j.compbimed.2020.103869.

- [94] K. El Asnaoui and Y. Chawki, "Using X-ray images and deep learning for automated detection of coronavirus disease," *J Biomol Struct Dyn*, 2020, doi: 10.1080/07391102.2020.1767212.
- [95] F. Ucar and D. Korkmaz, "COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images," *Med Hypotheses*, vol. 140, 2020, doi: 10.1016/j.mehy.2020.109761.
- [96] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, and V. Singh, "Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet," *Chaos Solitons Fractals*, vol. 138, 2020, doi: 10.1016/j.chaos.2020.109944.
- [97] S. Lafraxo and M. El Ansari, "CoviNet: Automated COVID-19 detection from X-rays using deep learning techniques," in *Colloquium in Information Science and Technology, CIST*, 2020. doi: 10.1109/CiSt49399.2021.9357250.
- [98] Ankita Shelke, Madhura Inamdar, Vruddhi Shah, Amanshu Tiwari, Aafiya Hussain, Talha Chafekar, and Ninad Mehendale, "Chest X-ray Classification Using Deep Learning for Automated COVID-19 Screening," *SN Comput Sci*, vol. 2, no. 4, 2021, doi: 10.1007/s42979-021-00695-5.
- [99] M. R. Bouguelia, S. Nowaczyk, K. Santosh, and A. Verikas, "Agreeing to disagree: active learning with noisy labels without crowdsourcing," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, 2018, doi: 10.1007/s13042-017-0645-0.
- [100] KC Santosh and S. Nakarmi, *Active Learning to Minimize the Possible Risk of Future Epidemics*. Singapore: Springer Nature Singapore, 2023. doi: 10.1007/978-981-99-7442-9.
- [101] B. Settles, "From Theories to Queries: Active Learning in Practice," *Proceedings of the Workshop on Active Learning and Experimental Design*, vol. 16, 2011.
- [102] KC Santosh and S. Antani, "Guest Editorial Multimodal Learning in Medical Imaging Informatics," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 3. 2023. doi: 10.1109/JBHI.2023.3241369.
- [103] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys*, vol. 50, no. 2. 2017. doi: 10.1145/3054912.
- [104] M. S. Kamal, L. Chowdhury, N. Dey, S. J. Fong, and K. Santosh, "Explainable AI to Analyze Outcomes of Spike Neural Network in Covid-19 Chest X-rays," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2021. doi: 10.1109/SMC52423.2021.9658745.
- [105] A. Makkar and KC Santosh, "SecureFed: federated learning empowered medical imaging technique to analyze lung abnormalities in chest X-rays," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 8, 2023, doi: 10.1007/s13042-023-01789-7.