HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Computer Science Department

COVID-19 Chest X-Ray Image

Classification

A graduation project dissertation by:

- Emam Hussein Emam (20180132)
- Rana Tamer Mohamed (20180242)
- Al-Husseini Ahmed Abdelaleem (20180128)
- Omar Khaled Abdelfatah (20180368)
- Hussien Wagdy Mahmoud (20180208)

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Computer Science Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

Dr.Hala Abd Aljalil Al–Sayed

June 2022

جامعة حلوان
كلية الحاسبات والذكاء الإصطناعي
قسم علوم الحاسب

تصنيف كوفيد-19 بصور الأشعة السينية للصدر

رسالة مشروع تخرج مقدمة من:

- امام حسين امام محمد ٢٠١٨٠١٣٢
- رنا تامر محمد ابراهيم ٢٠١٨٠٢٤٢
- الحسيني احمد عبد العليم محمد ٢٠١٨٠١٢٨
- عمر خالد عبد الفتاح عبد اللطيف ٢٠١٨٠٣٦٨
- حسين وجدي محمود ٢٠١٨٠٢٠٨

رسالة مقدمة ضمن متطلبات الحصول على درجة البكالوريوس في الحاسبات والذكاء الإصطناعي ، بقسم علوم الحاسب، كلية الحاسبات والذكاء الإصطناعي، جامعة حلوان

تحت إشراف:

د/هالة عبد الجليل السيد

يونيو ٢٠٢٢

Abstract

The rise of the coronavirus disease 2019 (COVID-19) pandemic has made it necessary to improve existing medical screening and clinical management of this disease. While COVID-19 patients are known to exhibit a variety of symptoms, the major symptoms include fever, cough, and fatigue. Since these symptoms also appear in pneumonia patients, this creates complications in COVID-19 detection, especially during the flu season, so the value of quick, accurate, and confident diagnoses cannot be undermined to mitigate the effects of COVID-19 infection, particularly in severe cases. Early studies identified abnormalities in chest X-ray images of COVID-19 infected patients that could be beneficial for disease diagnosis. Therefore, chest X-ray image-based disease classification has emerged as an alternative to aid medical diagnosis. However, manual detection of COVID-19 from a set of chest X-ray images comprising both COVID-19 and pneumonia cases is cumbersome and prone to human error. Thus, artificial intelligence techniques powered by deep learning algorithms, which learn from radiography images and predict the presence of COVID-19 would help doctors provide a quick and confident diagnosis. As a result, patients could get the right treatment before the most severe effects of the virus to take hold. Towards this purpose, here we implemented a set of deep learning pre-trained models such as ResNet, VGG, and Inception in conjunction with trying to increase accuracy for a computer vision AI system based on the convolutional neural network (CNN) model: Deep Learning in Healthcare (DLH)-COVID by solving two problems in it. All these CNN models cater to image classification exercises. We used publicly available resources of images and further strengthened the model by tuning hyperparameters to provide better generalization during the model validation phase. Our final DLH-COVID model yielded the highest accuracy of 96.7% in the detection of COVID-19 from chest X-ray images when compared to images of both pneumonia-affected and healthy individuals.

Keywords

COVID-19, Coronavirus detection, CNN, Chest X-rays, Pneumonia, Classification

Acknowledgement

Here we are going to write our last document in our college life, we cannot say how excited we are for our last discussion. The journey was not that easy at all, but our family's support and trust make it much easier, and we cannot forget our support of each other and how the college community was very helpful and friendly.

We want to mention our supervisor **Dr.Hala Abd Aljail** for all her advices and her trust in us, we really owe her a lot.

We cannot forget our great professors and teacher assistants (**Dr,Salwa Osama**, **Dr.Amr Ghoneim** and **TA. Wael Aid**) at the faculty for their help and support.

Finally special thanks for the owners of DLH_COVID model for giving us the opportunity to work on this project, and all who make this journey easier and funnier**.**

Table of Contents

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CNN-Convolutional Neural Network
CXR-Chest X-Ray
TP-True Positive
TN-True Negative
FP-False Positive
FN-False Negative

List of Equations

Glossary

Accuracy: a parameter that evaluates the correctness of the model by measuring a ratio of accurately predicted cases out of total number of cases.

True Positive: number of correctly identified COVID19/pneumonia X-ray images.

False Negative: number of incorrectly classified COVID19/pneumonia X-ray images

True Negative: number of correctly identified healthy X-ray cases.

False Positive: number of incorrectly identified healthy X-ray cases.

Precision: the ratio of correctly predicted positive cases to the total predicted positive cases. High precision relates to a low false positive rate.

Recall: It is the ratio of correctly predicted positive observations to all observations in actual class.

F1-Score: F1 Score is measured in case of uneven class distribution especially with many true negative observations. It provides a balance between Precision and Recall.

Chapter 1: An Introduction

1.1 Motivation

Coronavirus disease (COVID-19) is an extremely contagious disease, and it has been declared a pandemic by the World Health Organization (WHO) on 11th March 2020 considering the extent of its spread throughout the world **[1]**. The pandemic declaration also stressed the deep concerns about the alarming rate of spread and severity of COVID-19. It is the first recorded pandemic caused by any coronavirus. It is defined as a global health crisis of its time, which has spread all over the world. Governments of different countries have imposed border restrictions, flight restrictions, social distancing, and increasing awareness of hygiene. However, the virus is still spreading at a very rapid rate. While most of the people infected with the COVID-19 experienced mild to moderate respiratory illness, some developed deadly pneumonia. There are assumptions that elderly people with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, renal or hepatic diseases, and cancer are more likely to develop serious illnesses **[2]**. Until now, no specific vaccine or treatment for COVID-19 has been invented. However, there are many ongoing clinical trials evaluating potential treatments. More than 7.5 million infected cases were found in more than 200 countries until 11th June 2020, among which around 421 thousand deaths, 3.8 million recovery, 3.2 million mild cases, and 54 thousand critical cases were reported **[3]**, **[4]**. To combat the spreading of COVID-19, effective screening and immediate medical response for the infected patients is a crying need. Reverse Transcription Polymerase chain reaction (RT-PCR) is the most used clinical screening method for the COVID-19 patients, which uses respiratory specimens for testing **[5]**. RT-PCR is used as a reference method for the detection of COVID-19 patients however, the technique is manual, complicated, laborious, and time-consuming with a positivity rate of only 63% **[5]**. Moreover, there is a significant shortage of its supply, which leads to delays in the disease prevention efforts **[6]**. Many countries are facing difficulties with the incorrect number of COVID-19 positive cases not only due to the lack of test kits but also due to the delay in the test results **[7]**. These delays can lead to infected patients interacting with healthy patients and infecting them in the process. It is reported that the RT-PCR kit costs about USD 120-130 and also requires a specialized biosafety lab to house the PCR machine, each of which may cost USD 15,000 to USD 90,000 **[8]**. Such an expensive screening tool with delayed test results is leading to the spread of the disease, making the scenario worst. This is not an issue for the low-income countries only, but certain developed countries are also struggling to tackle this **[9]**. The other diagnosis methods for the COVID-19 include clinical symptoms analysis, epidemiological history, and positive radiographic images (computed tomography (CT) /Chest radiograph (CXR)) as well as positive pathogenic testing. The clinical characteristics of severe

COVID-19 infection is that of bronchopneumonia causing fever, cough, dyspnea, and respiratory failure with acute respiratory distress syndrome (ARDS) **[10] – 13]**. Readily available radiological imaging is an important diagnostic tool for COVID-19. The majority of COVID-19 cases have similar features on radiographic images including bilateral, multi-focal, ground-glass opacities with a peripheral or posterior distribution, mainly in the lower lobes, in the early stage, and pulmonary consolidation in the late stage **[13 – 19]**. Although typical CXR images may help early screening of suspected cases, the images of various viral pneumonia are similar, and they overlap with other infectious and inflammatory lung diseases. Therefore, it is difficult for radiologists to distinguish COVID-19 from other viral pneumonia. The symptoms of COVID-19 being like that of viral pneumonia can sometimes lead to the wrong diagnosis in the current situation while hospitals are overloaded and working round the clock. Such an incorrect diagnosis can lead to a non-COVID viral Pneumonia being falsely labeled as highly suspicious of having COVID-19 and thus delaying treatment with consequent costs, effort, and risk of exposure to positive COVID-19 patients. Thus, computer-aided chest X-ray examination methods are required for the detection of COVID19 cases from chest X-ray images. Towards this purpose, deep learning methods have proven to be useful in delivering high-quality results in addition to other advantages such as (1) maximum utilization of unstructured data, (2) elimination of additional cost, (3) reduction of feature engineering, and (4) removal of explicit data labeling. Therefore, deep learning methods are often used to extract relevant features to classify image objects using their autonomous nature. Indeed, deep learning techniques have contributed significantly to the analysis of medical images and the achievement of excellent classification performance with less time-consuming simulated tasks **[20]**. In recent years, the use of deep learning methods in building convolutional neural networks (CNNs) has led to many breakthroughs in various computer vision-oriented research work such as image segmentation, image recognition, and object detection. Previous research related to COVID-19 detection used various pre-trained CNN models such as VGG19, MobileNet, ResNet, and others for multi-class and binary classification tasks. For example, a combination of VGG19 and MobileNet in a multi-class classification study gave 97.8% accuracy in COVID-19 detection in healthy and pneumonia patients **[21]**. Another similar multi-class classification study using DarkCovidNet achieved a lower accuracy of 87% **[22]**. Using a binary classification system involving different ResNet models to classify COVID-19 versus non COVID19 patients yielded higher accuracy of >98% **[20],[23]**.

1.2 Problem Statement

The rise of the coronavirus disease 2019 (COVID-19) pandemic has made it necessary to improve existing medical screening and clinical management of this disease. While COVID-19 patients are known to exhibit a variety of symptoms, the major symptoms include fever, cough, and fatigue. Since these symptoms also appear in pneumonia patients, this creates complications in COVID-19 detection especially during the flu season. Early studies identified abnormalities in chest X-ray images of COVID-19 infected patients that could be beneficial for disease diagnosis. Therefore, chest X-ray image-based disease classification has emerged as an alternative to aid medical diagnosis. However, manual detection of COVID-19 from a set of chest X-ray images comprising both COVID-19 and pneumonia cases is cumbersome and prone to human error. Thus, artificial intelligence techniques powered by deep learning algorithms, which learn from radiography images and predict presence of COVID-19 have potential to enhance current diagnosis process. We can say that given a set of CXR images and prior knowledge about the labels of the images find the correct Symantec label for the pixels in the images.

1.3 Scope and Objectives

We attempt to increase the accuracy of DLH_COVID model that already outperformed other models based on accuracy in COVID-19 detection from X-ray images **[24]**, evaluate the effectiveness of various pre-trained models.

To achieve this objective, we first try to solve the problems of an imbalanced dataset and the quality of images, then train multiple pre-trained models and DLH_COVID using an 80% train dataset, then we will validate each model using a 10% validation dataset.

Finally, we will select the best model and performed an accuracy check using a 10% test dataset.

Chapter 2: Related Work (Literature Review)

2.1 Background

2.1.1 CNN

In the past few decades, machine learning (ML) algorithms have gradually attracted researchers' attention. This type of algorithm could take full advantage of the giant computing power of calculators in images processing through given algorithms or specified steps. However, traditional ML methods in classification tasks need to manually design algorithms or manually set feature extraction layers to classify images. In response to the above situation, LeCun et al. **[25]** proposed a CNN method, which could automatically extract features through continuously stacked feature layers and output the possibility of which class the input images belonged to. The shallow networks mainly focus on low-level features of the image. As the number of network layers increases, CNN model gradually extracts high-level features. Combining and analysing these advanced features, CNN learns the differences between different images, and uses a back-propagation algorithm to update and record the learned parameters. The essence of CNN is to filter the previous image or feature maps through a specific convolution kernel to generate the feature map of the next layer and combine with operations such as pooling operations to reduce the feature map scale and reduce the computation. Then, a nonlinear activation function is added to the generated feature map to increase the characterization ability of the model. Common pooling operations include maximum pooling and average pooling. Maximum pooling means that the feature delivered into the pooling layer is split into a number of sub-regions and will output the maximum of each sub-region according to the strides in horizontal and vertical. The only difference between maximum pooling and average pooling is the output of the sub-region where the average pooling outputs the average of each sub-region. Common activation functions include ReLU (Rectified Linear Units) and Sigmoid.

2.1.2 Image Enhancement Techniques

Image enhancement is an important image-processing technique, which highlights key information in an image and reduces or removes certain secondary information to improve the identification quality in the process **[26]**. We employ three different enhancement

strategies in this project. In the following sections, we will briefly introduce these image enhancement techniques:

2.1.2.1 Histogram Equalization (HE)

This method usually increases the global contrast of many images, especially when the image is represented by a narrow range of intensity values. Through this adjustment, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the highly populated intensity values which are used to degrade image contrast **[27]**.

2.1.2.2 Contrast limited adaptive histogram equalization (CLAHE)

Adaptive histogram equalization (AHE) is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image. However, AHE tends to overamplify noise in relatively homogeneous regions of an image. A variant of adaptive histogram equalization called contrast limited adaptive histogram equalization (CLAHE) prevents this by limiting the amplification **[28]**.

2.1.2.3 Image Invert/ Complement

The image inversion or complement is a technique where the zeros become ones and ones become zeros so black and white are reversed in a binary image. For an 8-bit gray scale image, the original pixel is subtracted from the highest intensity value, 255, the difference is considered as pixel values for the new image. For x-ray images, the dark spots turn into lighter and light spots become darker **[26]**.

2.1.3 Resampling Technique

A widely adopted technique for dealing with highly unbalanced datasets is called resampling. It consists of removing samples from the majority class (under-sampling) and/or adding more examples from the minority class (over-sampling). **Figure 1** shows how resampling technique work.

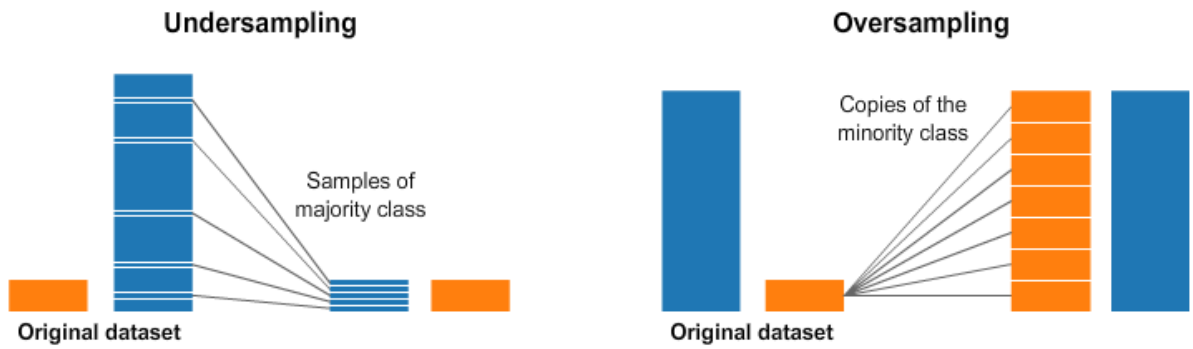


Figure 1. Representative image for how resampling technique works with imbalanced classes

Despite the advantage of balancing classes, these techniques also have their weaknesses (there is no free lunch).

Under-sampling can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge, but also it can discard potentially useful information which could be important for building rule classifiers. The sample chosen by random under-sampling may be a biased sample. And it will not be an accurate representation of the population. Thereby, resulting in inaccurate results with the actual test data set.

Over-sampling unlike under-sampling, this method leads to no information loss, so it outperforms under sampling, but if it replicates the minority class events it will increase the likelihood of overfitting.

2.2 Literature Survey

Mahmud et al. **[29]** proposed a deep learning-based technique for the classification of COVID-19 and pneumonia infection. Features are extracted using a deep CNN model named CovXNet. A public dataset is utilized for training containing 1493 samples of non-COVID-19 pneumonia and 305 samples of COVID-19 pneumonia. The model successfully classified non-COVID-19 pneumonia and COVID-19 pneumonia with an accuracy of 96.9%. Umair et al. **[30]** presented a technique for the binary classification of COVID-19. A publicly available dataset is used for training and evaluation of the technique, consisting of 7232 chest X-ray images. Four deep learning models are being compared in this study. Various evaluation parameters are utilized for the validation of results. Li et al. **[31]** proposed a technique for the detection of COVID-19 infection. The proposed technique successfully differentiates COVID-19 pneumonia and

community-acquired pneumonia (CAP). The deep learning model that is utilized for training is named COVNet; this is three-dimensional CNN architecture. A publicly available dataset is used which contains CT scan samples of COVID-19 and community acquired pneumonia (CAP). The COVNet model attained a rate of 90% sensitivity and 96% specificity. Abbas **[32]** proposes another convolution neural network-based technique for the classification of COVID-19 infection using chest X-ray images. The CNN model named decompose, transfer, and compose, and commonly known as DeTrac, was used. Multiple datasets from various hospitals throughout the world were used in this research. The DeTrac model attained an accuracy of 93.1% and a sensitivity of 100%. To classify COVID19 and typical pneumonia, Wang et al. **[33]** presented a technique based on deep learning which used the inception model **[34]**. Modifications in fully connected layers of inception are completed before training the network. In this study, 1053 images were used. The model gave an accuracy of 73.1% with a sensitivity of 74% and specificity of 67%. Sankar et al. **[35]** proposed a deep learning technique for the classification of COVID-19 infected chest X-rays. A Gaussian filter was used for preprocessing, while the local binary pattern was utilized to extract texture features. Later, the extracted LBP features are fused with the CNN model InceptionV3 to improve the performance. The classification is carried out using multi-layer perceptron. The model was validated on an X-ray dataset and attained an accuracy of 94.08%. Panwar **[36]** proposes a convolutional neural network-based technique where a 24-layer CNN model has been used for the classification of COVID-19 and normal images. The author named this model nCOVnet. The X-ray dataset was used for training nCOVnet. The model attains an accuracy of up to 97%. Zheng **[37]** presents the segmentation-based classification technique. The U-Net **[38]** is trained on CT images to generate lung masks. Two-dimensional U-Net is used for this purpose. Later, the mask generated by U-Net is fed to DeCoVNet for the classification of COVID-19. The architecture of DeCoVNet consists of three parts: (1) the stem network, consisting of 3-D vanilla, along with a batch norm and pooling layer; (2) two 3D ResBolcks are used in the second stage, where ResBolcks are used for feature map generation; (3) the third part of DeCoVNet is used for classification that is based on probabilities. A progressive classifier is used for the binary classification of COVID-19. Xu et al. **[39]** proposed a technique for the detection of COVID-19 infection using the deep learning-based model. Two ResNet **[40]** based models were used in this study: (1) ResNet18; (2) a modified ResNet18 with the mechanism of localization. The CT scan images were used for training the models. The final evaluation is performed using noisy-OR Bayesian. The overall accuracy of the proposed technique is cited as 88.7%. Hussain et al. **[41]** proposed a system that is called CoroDet and is based on

convolutional neural networks for the detection of COVID-19 infection. The proposed CNN model is comprised of 22 layers, and is trained on chest X-rays and CT scan images. The model is able to classify COVID-19 and non-COVID-19, Moreover, it can classify three different classes, including COVID-19, pneumonia, and normal. The 22 layered model shows good classification results. Khan et al. **[42]** presented a technique for the classification of COVID19 disease. The proposed technique used CNN for the classification; a known deep learning model Xception is modified for this purpose. The modified model is named CoroNet by the authors. The dataset used for training consists of four classes, including COVID-19, normal, viral pneumonia, and fungal pneumonia. Using the mentioned dataset, the model is trained using the different combinations of datasets. The model gave 89.6% accuracy. Choudary et al. **[43]** adopted a deep learning technique to classify COVID-19 and viral pneumonia. Various deep learning models have been used for training in this work. In addition, the transfer learning approach is exploited for training deep learning models. The public dataset is utilized for the training of models. The dataset contains samples of COVID-19, typical viral pneumonia, and chest X-rays of healthy and normal people. The models attained good classification accuracies. Ozturk et al. **[44]** presented a 17 layered Darknet model for the detection of COVID-19 infection. Different sizes of filters were employed at CNN layers. The presented technique classifies binary classes (COVID-19 and no finding) and multiple classes (COVID-19, pneumonia, and no findings). For model training, raw chest X-ray images were used. The model attained an accuracy of 98.08% for binary classification, while for multiple classes an accuracy of 87.02% is achieved.

2.3 Analysis of the Related Work

For detection of COVID-19, most of the research has been performed using chest X-rays, which show the importance of chest X-rays in diagnosing chest infections and, specifically, for diagnosing COVID-19. The chest X-rays were found to be the primary tool in medical image analysis. Traditional image processing-based feature extraction techniques are complex compared to deep learning techniques. Recently deep learning techniques surpassed traditional techniques in terms of performance. However, traditional techniques can be used, along with deep learning techniques, for aid **[35]**. Moreover, deep learning techniques require a large amount of data for training and testing. Deep learning models trained on the limited datasets are not generalized; thus, such models are not reliable. It has been found through the

literature, that data augmentation techniques can be used to resolve small dataset issues **[45]**. Furthermore, the already available research is more focused on the binary classification of COVID-19 **[29–33]** and limited research has been conducted for multiclass classification of COVID-19 **[39–44]**. The performance of multiclass classification is not yet adequate, and hence their performance needs to be improved.

Chapter 3: Methods

In this section, the two datasets used for the training and testing are discussed along with the deep learning models used in this project. The datasets used in this project acquired from Kaggle is composed of multiple datasets **[45 - 46]**.

3.1 Dataset

We used COVID-19 chest X-ray image repository that DLH_COVID model used which is publicly available **[45]**, and another public source of data Called COVID-19 Radiography Database **[46]**.

COVID-19 chest X-ray image dataset contain 6,432 images categorized into three groups: COVID-19, pneumonia, and normal/healthy. It is included X-ray images with confirmed COVID-19, confirmed common pneumonia, and normal\healthy individuals. This dataset comprised 80% train dataset and 20% test dataset. **Table 1** show the distribution of each class. **Figure 2** shows representative chest X-ray images of all the three conditions of COVID-19, pneumonia, and normal/healthy conditions.

Table 1. The number of images segregated in the train, and test folders of the original dataset. Distribution comprised 80% train, and20% test.

Classification	Train	Test	Total
COVID-19	460	116	576
Normal	1266	317	1583
Pneumonia	3418	855	4273
Total	5144	1288	6432

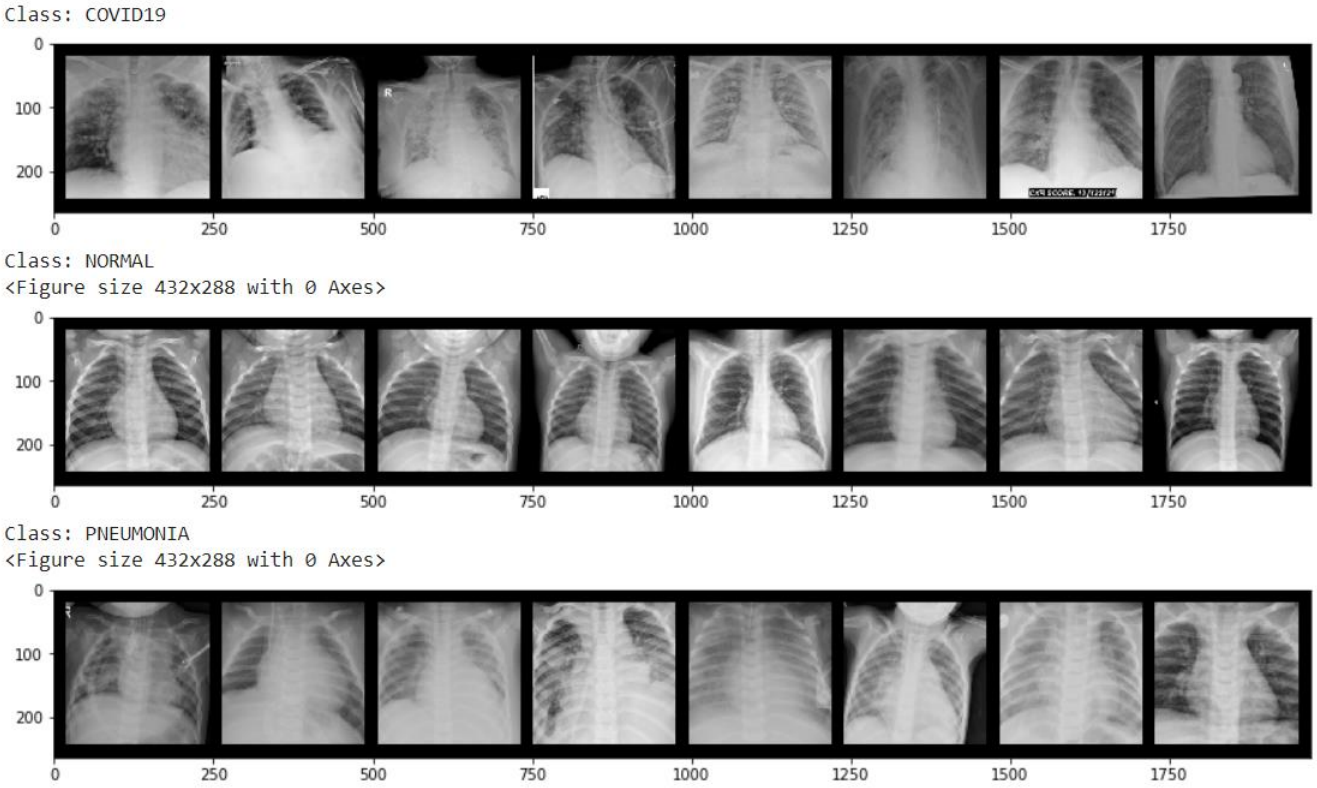


Figure 2. Representative chest X-ray images of COVID-19, pneumonia, and normal/healthy conditions. Note the increased ground glass opacity in COVID-19 X-ray images. Each image is of 224 × 224 resolutions.

COVID-19 Radiography Database contain in total 21165 samples divided into four main classes: Covid-19, Lung Opacity, Normal, and Viral Pneumonia. All the images are in Portable Network Graphics (PNG) file format and the resolution are 299x299 pixels. On this current update, the database currently holds 3,616 COVID-19 positive cases, 10,192 Normal, 6,012 Lung Opacity (Non-COVID lung infection), and 1,345 Viral Pneumonia images.

3.1.1 Data Preprocessing

Image pre-processing refers to all the transformations on the raw data before it is fed to the machine learning or deep learning algorithm. For instance, training a convolutional neural network on raw images will probably lead to bad classification performances **[47]**. **Figure 3** shows the percentage of each class in the dataset and illustrate that Class imbalance is a challenging aspect of COVID-19 chest X-ray image dataset.

The class imbalance problem typically occurs when there are many more instances of some classes than others. In such cases, the unbalanced classes create a problem due to two main reasons:

1. We don’t get optimized results for the class which is unbalanced in real time as the model/algorithm never gets sufficient look at the underlying class
2. It creates a problem of making a validation or test sample as its difficult to have representation across classes in case number of observation for few classes is extremely less.

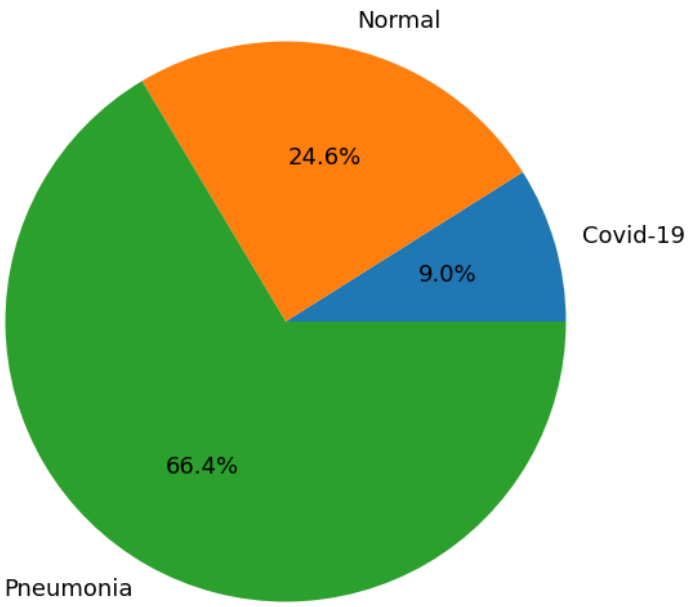


Figure 3: visualization of the percentage of each class in the COVID-19 chest X-ray image dataset.

We use resampling technique to try to solve the class imbalance problem that exists in the first dataset. We made over-sampling for COVID-19 class in the COVID-19 chest X-ray image repository by using COVID-19 Radiography Database as another source for data. We didn’t replicate the minority class to avoid increasing the likelihood of overfitting, so we load COVID-19 CXR images from specific unique reference **[48]** included in COVID-19 Radiography Database and based on that we didn’t load all COVID-19 CXR images from COVID-19 Radiography Database.

We did the same process for normal class in the COVID-19 chest X-ray image dataset. We loaded normal CXR images from specific unique reference **[49]** that they are existing in the COVID-19 Radiography Database, but because the normal class in the COVID-19 Radiography Database is large we selected defined number of images randomly to make the COVID-19 and normal classes equal in the number of data samples

The last step to solve the imbalance class problem is under-sampling the pneumonia class in the COVID-19 chest X-ray image dataset. We removed number of data samples randomly. Finally, all classes in the first dataset are equally and images are saved in jpeg format. **Figure 3** show the percentage of each class after applying resampling techniques on the first dataset. **Table 2** show the number of images distributed in the 80% train, and 20% test dataset after modifying the first dataset.

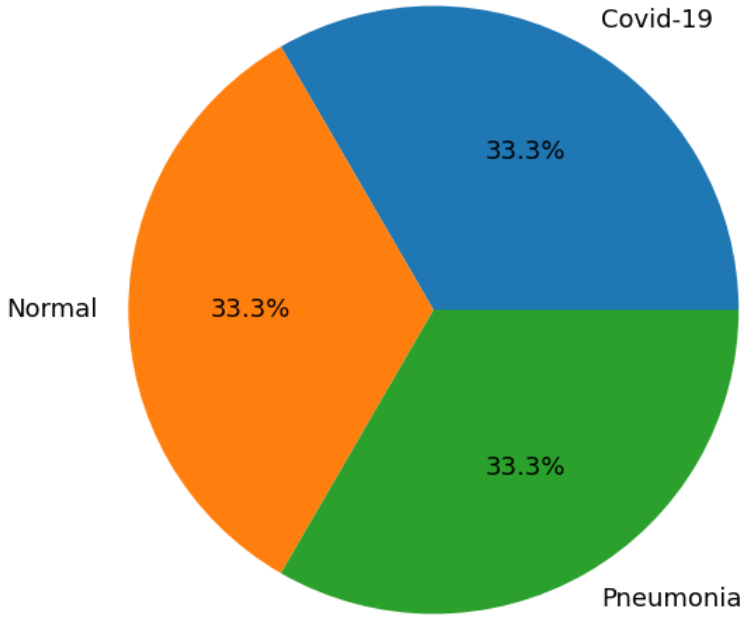


Figure 3: visualization of the percentage of each class in the modified dataset

After making dataset balanced, we applied the three different types of image enhancements techniques that we mentioned above. We applied all three techniques on one sample image to see the difference among them. **Figure 4** shows the original image before and after applied the different image enhancement techniques and their Histogram.

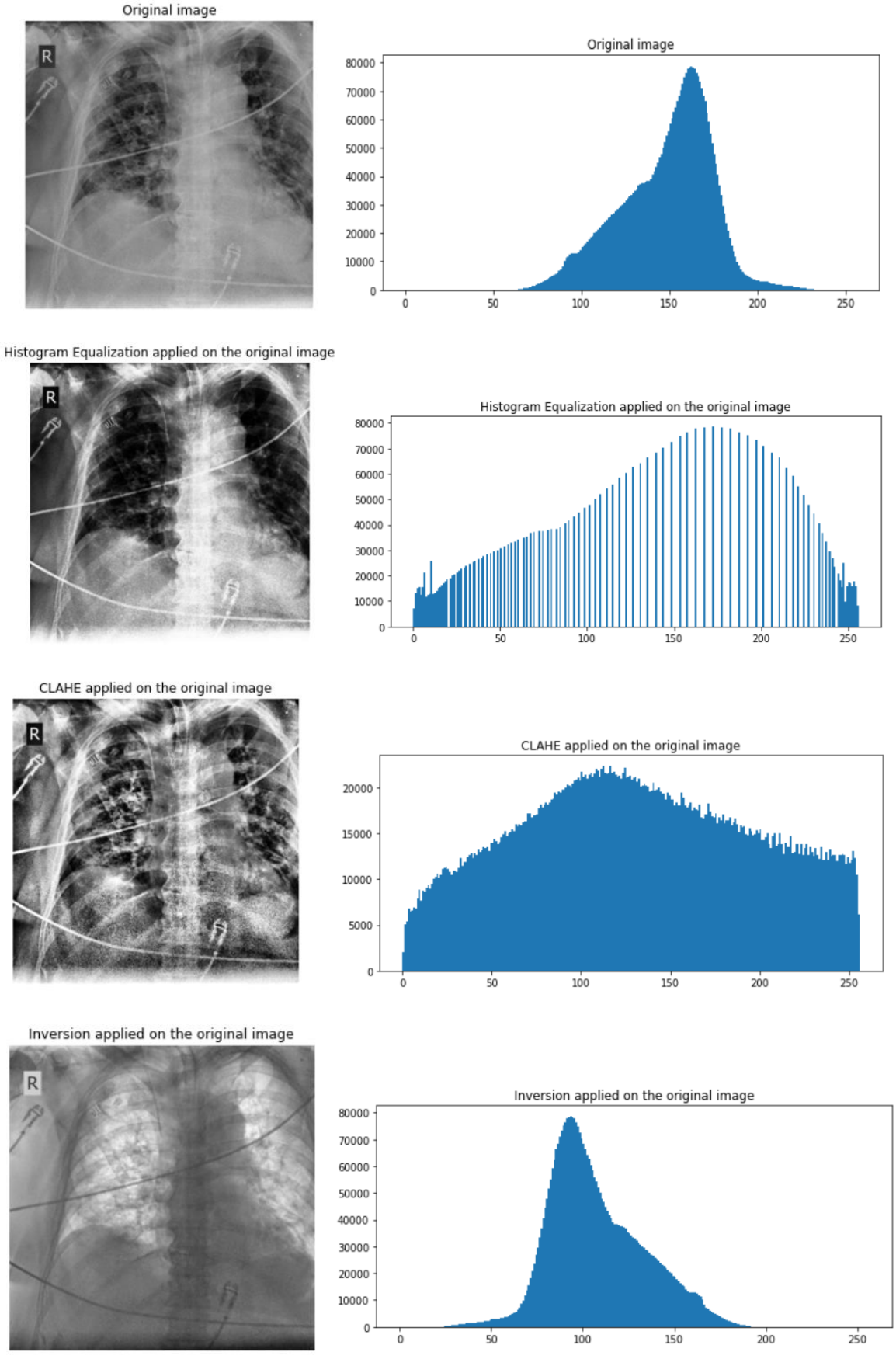


Figure 4. Shows the original image before and after applied the different image enhancement techniques and their Histogram

Previous research has shown that down sampling input images to a lower resolution increased the effectiveness of CNN classification models **[50]**. Therefore, we resized all train, and test images into standard size 224 x 224 to maintain uniformity in image resolution. Since one of the selected pre-trained models, Inception-V3, is only compatible with resolution 299 x 299 **[51]**. The pre-processing step also included a center crop mechanism, which was applied to all the X-ray images to reduce background noise and enhance focal length position.

3.1.2 Segregation of image dataset into train, validation, and test

To further prepare the test dataset for classification exercise, we applied stratified resampling method to split the test dataset into two subsets: 10% validation and 10% test subset. The 10% validation subset was used to prevent model overfitting and enhance model evaluation process. **Table 2** shows the final number of images distributed in the 80% train, 10% validation, and 10% test dataset used for the pre-trained model and DLH_COVID model described below.

Table 2. The number of images segregated in the train, validation, and test folders of the modified dataset. Final distribution comprised 80% train, 10% validation and 10% test

Classification	Train	Validation	Test	Total
COVID-19	2399	300	301	3000
Normal	2399	300	301	3000
Pneumonia	2399	300	301	3000
Total	7197	900	903	9000

3.2 Pre-trained Model Selection

Previous research helped us to identify pre-trained models with high accuracy of COVID-19 detection from chest X-ray images [3,12,13,14]. These are the following models we used:

1- ResNet: These architectures were proposed by He et al. from Microsoft [16]. ResNet architectures introduced the use of residual layers and skip connections to solve the problem of vanishing gradient that may impact the weightage change in a neural network.

2- VGG: These architectures were introduced by Oxford University's Visual Geometry Group [17], where they demonstrated that using small filters of size 3 x 3 in each convolutional layer throughout the network may result in better performance. The main idea behind VGG architecture is that multiple small filters can make the design simpler and reproduce similar results compared to that of larger filters.

3- GoogleNet: The main feature of GoogleNet/Inception architecture [18] is the innovation of the inception module, which is a series of 1-by-1 convolutional layers/blocks used for dimensionality reduction and feature aggregation. This model comprised a total of 22 layers with 9 inception modules.

3.3 DLH_COVID Model Architecture

DLH_COVID is a new convolutional neural network (CNN) for image classification exercise. It consists of three convolutional layers followed by two fully connected linear layers **[52]**. **Figure 5** shows the detailed architecture of DLH_COVID:

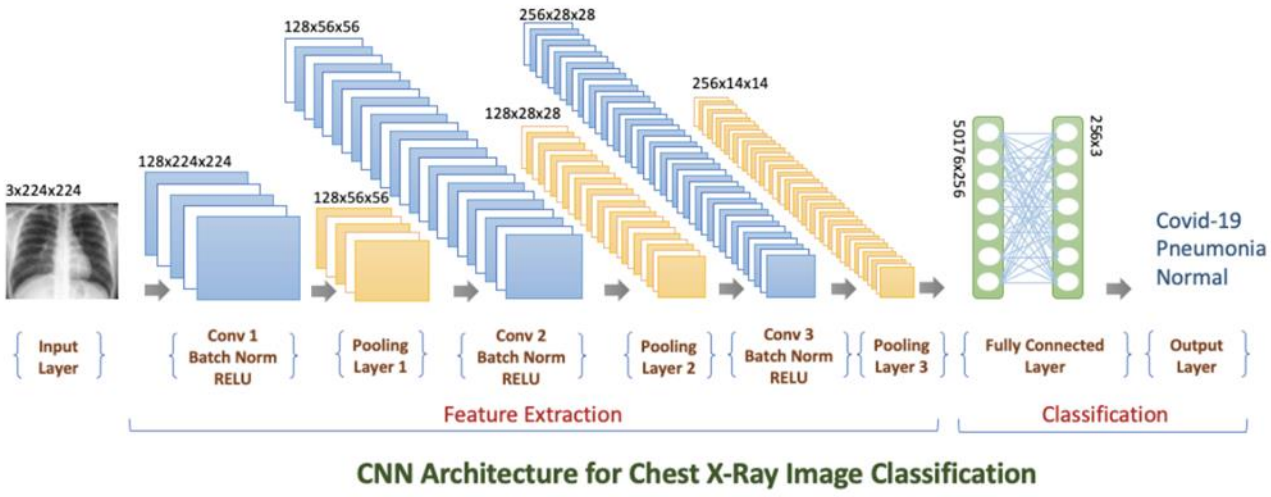


Figure 5. Schematic of DLH_COVID model architecture. It consists of three convolutional layers, two fully connected linear layers, and additional intermediate max pool layers. Input dimension of the CNN network: 3 x 224 x 224 and output dimension of the CNN network: 256 x 3.

3.4 Classification Evaluation Metrics

In this subsection, several evaluation metrics, accuracy, precision, recall, F1 score and so on, are described. According to the outputs of model, four indices, True Positive, True Negative, False Positive, False Negative, are used to analyse and identify the performance of model. The True Positive means that the chest X-ray images, which suffer from COVID-19, are signed as COVID-19 as well by the model. The True Negative means if the chest X-ray images do not show COVID-19 as well as the model predicts. The remaining matrices have a similar definition.

The four metrics are given as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

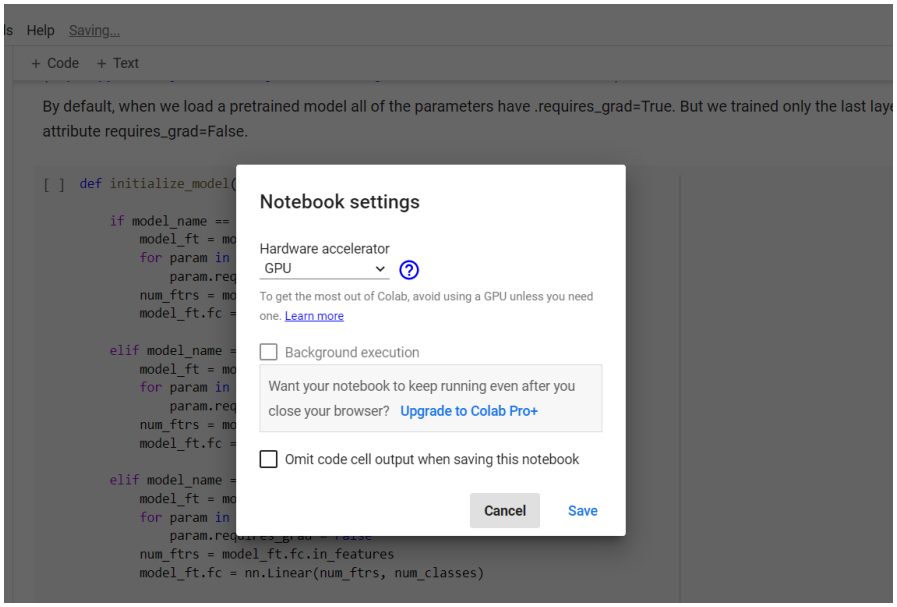
$$\text{F1} - \text{score} = 2 \text{ x } \frac{\text{precision x recall}}{\text{precision} + \text{recall}} \quad (4)$$

Chapter 4: Implementation, Experimental Setup, & Results

4.1 Implementation Details

The implementation process can be split into the following steps:

1. Setting up Google Colab



2. Importing Libraries

```
import os
import time
import cv2
import random
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import torch
import torch.optim as optim
import torch.nn as nn
import torch.nn.functional as F
import torchvision
import torchvision.datasets as datasets
import torchvision.transforms as transforms
from torchvision import models
from torch import optim
from sklearn.metrics import *
from PIL import Image
import shutil
import zipfile
import math
from pathlib import Path
# import splitfolders
# which denotes the number of processes that generate batches in parallel.
num_workers = 2 #change this parameter based on your system configuration
# which denotes the number of samples contained in each generated batch.
batch_size = 32 #change this parameter based on your system configuration
seed = 24
random.seed(seed)
metrics = ['Accuracy', 'Precision', 'Recall', 'F1-score']
categories = ["COVID19", "NORMAL", "PNEUMONIA"]
num_classes = len(categories)
splits = ['train', 'val', 'test']
```

3. Loading datasets from google drive

```
[ ] # make copy of chest-xray-covid19-pneumonia dataset
    # then unzip the file
    !cp '/content/drive/MyDrive/chest-xray-covid19-pneumonia.zip' '/content'
    local_zip = '/content/chest-xray-covid19-pneumonia.zip'
    zip_ref = zipfile.ZipFile(local_zip, 'r')
    zip_ref.extractall('/content')
    zip_ref.close()
```

```
# make copy of covid19-radiography-database database
# then unzip the file
!cp '/content/drive/MyDrive/covid19-radiography-database.zip' '/content'
local_zip = '/content/covid19-radiography-database.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/content')
zip_ref.close()
```

```
[ ] DATA_PATH = '/content/Data'
    model_path = '/content'
```

4. Resampling technique

```
if not os.path.isdir('/content/covid_data'):
  os.mkdir('/content/covid_data')
if not os.path.isdir('/content/covid_data/COVID-19'):
  os.mkdir('/content/covid_data/COVID-19')
files = []
for i in range(1143, 3567):
  files.append('COVID-'+ str(i) + '.png')

def load_covid_data(file_source, file_destination,files_name=files):
  """ this function move specific files
      from COVID folder that included in database
      to COVID-19 folder"""
  get_files = os.listdir(file_source)
  for file in get_files:
    if file in files:
      shutil.move(file_source + file, file_destination)
def png_to_jpeg_converter(filePath='/content/covid_data/COVID-19'):
    inputPath = Path(filePath)
    inputFiles = inputPath.glob("**/*.png")
    outputPath = Path(filePath)
    for f in inputFiles:
        outputFile = outputPath / Path(f.stem + ".jpg")
        im = Image.open(f)
        im.save(outputFile)
        try:
            f.unlink()
        except OSError as e:
            print(f"Error:{ e.strerror}")
def move_data():

  train_path = '/content/output/train/COVID-19/'
  test_path = '/content/output/val/COVID-19/'
  for file in os.listdir(train_path):
     shutil.move(train_path + file, '/content/Data/train/COVID19')
  for file in os.listdir(test_path):
    shutil.move(test_path + file, '/content/Data/test/COVID19')
  try:
    shutil.rmtree('/content/output')
  except OSError as e:
      print("Error: %s : %s" % ('/content/output', e.strerror))

def remove_pne_data():

  train_files = os.listdir('/content/Data/train/PNEUMONIA')
  test_files = os.listdir('/content/Data/test/PNEUMONIA')
  train_files = random.sample(train_files, 1019)  # Pick 1019 random files
  test_files = random.sample(test_files, 254)  # Pick 254 random files

  for train_file in train_files:
    os.remove(os.path.join('/content/Data/train/PNEUMONIA', train_file))
  for test_file in test_files:
    os.remove(os.path.join('/content/Data/test/PNEUMONIA', test_file))
```

5. Creating a validation set

```
[ ]  # Split original test folder randomly and equaly between test and validation (new) folders
     if not os.path.isdir(os.path.join(DATA_PATH,'val')):
       os.mkdir(os.path.join(DATA_PATH,'val'))
       for subdir in categories:
         for root, dirs, files in os.walk(os.path.join(DATA_PATH, 'test/' +subdir)):
           size = len(files)
           migrate_index = random.sample(range(size), int(size/2))
           files_to_migrate = [files[i] for i in migrate_index]
           for file in files_to_migrate:
             file_path = os.path.join(DATA_PATH, 'test/' + subdir + '/' + file)
             new_dir = os.path.join(DATA_PATH, 'val/' + subdir)
             if not os.path.exists(new_dir):
               os.mkdir(new_dir)
             if not os.path.exists(os.path.join(new_dir, file)):
               a = shutil.move(file_path, new_dir)
     else: print("Folder 'val' is already available and the split didn't happen")
```

6. Apply HE technique

```
# Histogram equalization
HIST_DIR_PATH = '/content/Hist'

# create COVID-19 folder to store covid data from database
if not os.path.isdir('/content/Hist'):
  os.mkdir('/content/Hist')
  os.mkdir('/content/Hist/train')
  os.mkdir('/content/Hist/test')
  os.mkdir('/content/Hist/val')
  os.mkdir('/content/Hist/test/COVID19')
  os.mkdir('/content/Hist/test/NORMAL')
  os.mkdir('/content/Hist/test/PNEUMONIA')
  os.mkdir('/content/Hist/train/COVID19')
  os.mkdir('/content/Hist/train/NORMAL')
  os.mkdir('/content/Hist/train/PNEUMONIA')
  os.mkdir('/content/Hist/val/COVID19')
  os.mkdir('/content/Hist/val/NORMAL')
  os.mkdir('/content/Hist/val/PNEUMONIA')


def hist_equ(file_name):
    # Read image file
    img = cv2.imread(file_name, 0)
    equ = cv2.equalizeHist(img)
    return equ

def traverse_and_apply_hist(data=DATA_PATH, new_dis=HIST_DIR_PATH):
    # Iterate through each image file
    for split in os.listdir(data):
        for category in os.listdir(os.path.join(data, split)):
            for file_name in os.listdir(os.path.join(data, split + '/' +category)):
                equ = hist_equ(os.path.join(data, split+ '/' + category + '/' +file_name))
                # Save file to new directory
                cv2.imwrite(os.path.join(new_dis, split+ '/' + category + '/' +file_name), equ)
```

7. Load both the training and validation images into memory, pre-processing them as described in the previous section.

```
def load_data(data_path=DATA_PATH, num_workers=num_workers):
    transform_dict = {
        'model': transforms.Compose(
                                    [transforms.Resize(224),
      .                              transforms.CenterCrop(224),
                                     transforms.ToTensor(),
                                     ])}
    train_data = datasets.ImageFolder(root=data_path + '/train', transform=transform_dict['model'])
    train_loader = torch.utils.data.DataLoader(train_data, batch_size=batch_size, shuffle=True, num_workers=num_workers)
    val_data = datasets.ImageFolder(root=data_path + '/val', transform=transform_dict['model'])
    val_loader = torch.utils.data.DataLoader(val_data, batch_size=batch_size, shuffle=False, num_workers=num_workers)
    test_data = datasets.ImageFolder(root=data_path + '/test', transform=transform_dict['model'])
    test_loader = torch.utils.data.DataLoader(test_data, batch_size=batch_size, shuffle=False, num_workers=num_workers)
    return train_data, train_loader, val_data, val_loader, test_data, test_loader

train_data, train_loader, val_data, val_loader, test_data, test_loader = load_data()
dataset = torch.utils.data.ConcatDataset([train_data, val_data, test_data])
```

8. Define the loss function, accuracy

```
def train_val_model(model):
    t_start = time.time()
    global best_val_model
    global best_val_loss
    best_val_loss = 1
    global best_val_epoch
    best_val_epoch = 0
    df = pd.DataFrame(columns = ['model_name', 'epoch', 'train', 'val'])
    print(f"Training model {model_name} with {df_dataset.loc['TOTAL', 'train']} samples and max of {n_epochs} epochs, and validating with {df_dataset.loc['TOTAL', 'val']} samples\n")
    train_size, val_size = len(train_loader), len(val_loader)
    for epoch in range(1, n_epochs+1):
        # Beginning of training step
        t0 = time.time()
        model.train()
        train_loss, val_loss = 0.0, 0.0
        for i, (data, target) in enumerate(train_loader):
            target = target.to(device)
            data = data.to(device)
            optimizer.zero_grad()
            outputs = model(data)
            loss = criterion(outputs, target)
            loss.backward()
            optimizer.step()
            train_loss += loss.detach().cpu().numpy()
        # Beginning of evaluation step
        model.eval()
        for j, (data, target) in enumerate(val_loader):
            target = target.to(device)
            data = data.to(device)
            outputs = model(data)
            loss = criterion(outputs, target)
            val_loss += loss.detach().cpu().numpy()
        print(f"Epoch {epoch}:\t train loss={train_loss/train_size:.5f} \t val loss={val_loss/val_size:.5f} \t time={(time.time() - t0):.2f}s")
        df.loc[len(df)] = [model_name, epoch, train_loss/train_size, val_loss/val_size]
        # if epoch >= best_val_epoch + 15:
        #     break
        if use_scheduler: scheduler.step(val_loss/val_size)   # Optional to use scheduler for dynamic learning rate
    print(f"Best model has val loss={best_val_loss:.5f} for {best_val_epoch} epochs")
    print(f"Total time training and evaluating: {(time.time()-t_start):.2f}s")
    return model, df
```

9. Define the network architecture and training parameters

```
layer = [128, 128, 256, 256, 3]

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(3, layer[0], 3, padding=1)
        self.bn1 = nn.BatchNorm2d(layer[0])
        self.pool1 = nn.MaxPool2d(kernel_size=4, stride=4)
        self.conv2 = nn.Conv2d(layer[0], layer[1], 3, padding=1)
        self.bn2 = nn.BatchNorm2d(layer[1])
        self.conv3 = nn.Conv2d(layer[1], layer[2], 3, padding=1)
        self.bn3 = nn.BatchNorm2d(layer[2])
        self.pool2 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.linear1 = nn.Linear(14 * 14 * layer[2], layer[3])
        self.linear2 = nn.Linear(layer[3], layer[4])
        self.relu = nn.ReLU()
        self.dropout = nn.Dropout()
    def forward(self, x):
        x = self.pool1(self.relu(self.bn1(self.conv1(x))))
        x = self.pool2(self.relu(self.bn2(self.conv2(x))))
        x = self.pool2(self.relu(self.bn3(self.conv3(x))))
        x = x.reshape(x.size(0), -1)
        x = self.relu(self.linear1(x))
        x = self.dropout(x)
        x = self.linear2(x)
        return x
```

10. Train the network, logging the validation/training loss and the validation accuracy

```
np.random.seed(seed)
torch.manual_seed(seed)
net = Net().to(device)
model_name = 'DLH_COVID'
n_epochs = 25
learning_rate = 1e-3
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(net.parameters(), lr=learning_rate)

# Optional: use_scheduler = True will use dynamic values of learning_rate
use_scheduler = True  # Set True if using scheduler
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, factor=0.8, patience=3, threshold = 0.001,
                                                 verbose=True, min_lr = 1e-5, threshold_mode = 'abs')
our_model, df_epochs = train_val_model(net)
```

11. Making predictions

```
def accuracy_model(model, loader):
    model.eval()
    print(f"Testing the model {model_name} with {df_dataset.loc['TOTAL', 'test']} samples \n")
    predictions, actuals = calc_pred_actuals(model, loader)
    conf_matrix = create_confusion_matrix(predictions, actuals)
    df_test = calc_metrics(predictions, actuals, 'Test Results').astype(float)
    print(df_test)
    return df_test, conf_matrix
```

12. Plot the logged values

```
def plot_train_val_losses(df):
    df2 = pd.melt(df, id_vars=['epoch'], value_vars=['train', 'val'], var_name='process', value_name='loss')
    sns.lineplot(x = "epoch", y = "loss", data = df2, hue = "process",
                 style = "process", palette = "hot", dashes = False,
                 markers = ["o", "<"],  legend="brief").set_title("Train and Validation Losses by Epoch")
    plt.show()
```

13. Save and freeze the trained network

```
torch.save(our_model.state_dict(), os.path.join(model_path, 'model_21e_1e-2_96.7_hist_v2.pth'))
```

4.2 Experimental / Simulations Setup

To accomplish the task of COVID-19 CXR Image Classification, a CNN model called DLH_COVID is used for tuning its hyperparameters (learning rate, epochs, batch size, optimizer) and optimize recognition accuracy, and we proposed to investigate different image enhancement techniques.

The recognition process of the COVID-19 CXR Image Classification consists of the following steps:

1. To load the COVID-19 CXR image data.
2. To apply the pre-processing techniques.
3. To divide the input images into training, test, validation images.
4. To divide the training dataset into batches of a suitable size.
5. To train DLH_COVID model and tune its hyperparameters.
6. To use a trained model for the classification.
7. To save and freeze the trained network.
8. If the accuracy is not high enough, return to step 5.
9. To analyse the recognition accuracy.

In summary, the present work for COVID-19 CXR Image Classification investigates the role of image enhancement techniques and tuning hyperparameters. All the experiments were done in Pytorch and Keras on google colab. As mentioned earlier, the chest X-ray datasets are taken from Kaggle. The first **[45]** dataset went through the pre-processing stage, which address the problem of class imbalance. This is an important step, as an imbalanced dataset adversely affects the model training by showing bias towards one or more classes. Later, the dataset is split into three subsets—training, validation, and testing—with a ratio of 80:10:10, respectively. The testing dataset is unseen and is used for the evaluation of the model after training the models.

4.3 Conducted Results

In this section, the classification performance of the DLH_COVID model on a multiclass CXR image dataset applied different image enhancement techniques on it. The dataset and classification results are presented in this section.

4.3.1 Dataset

As mentioned earlier, there are different image enhancement techniques that applied on the dataset in the pre-processing stage. **Figure 6** shows representative chest X-ray images of all the three conditions of COVID-19, pneumonia, and normal/healthy conditions after applying HE. **Figure 7** shows representative chest X-ray images of all the three conditions of COVID-19, pneumonia, and normal/healthy conditions after applying CLAHE. **Figure 8** shows representative chest X-ray images of all the three conditions of COVID-19, pneumonia, and normal/healthy conditions after applying Invert image.

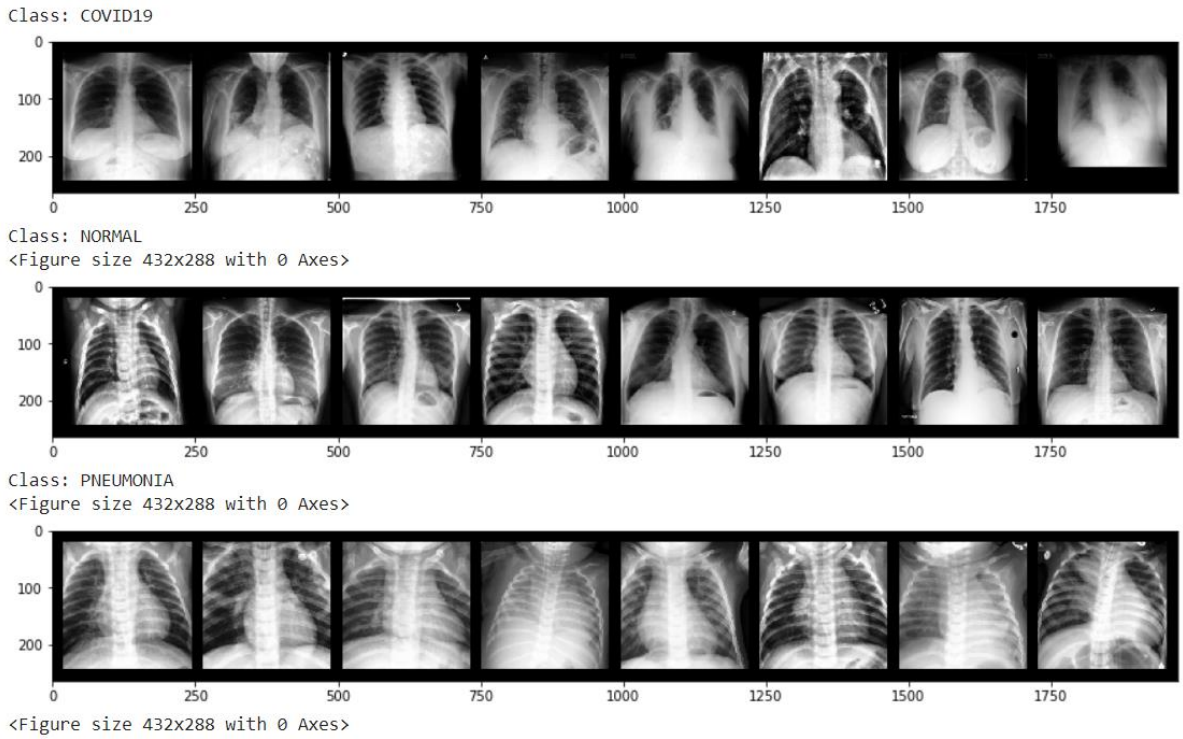


Figure 6. Representative chest X-ray images of COVID-19, pneumonia, and normal/healthy conditions after applying HE.

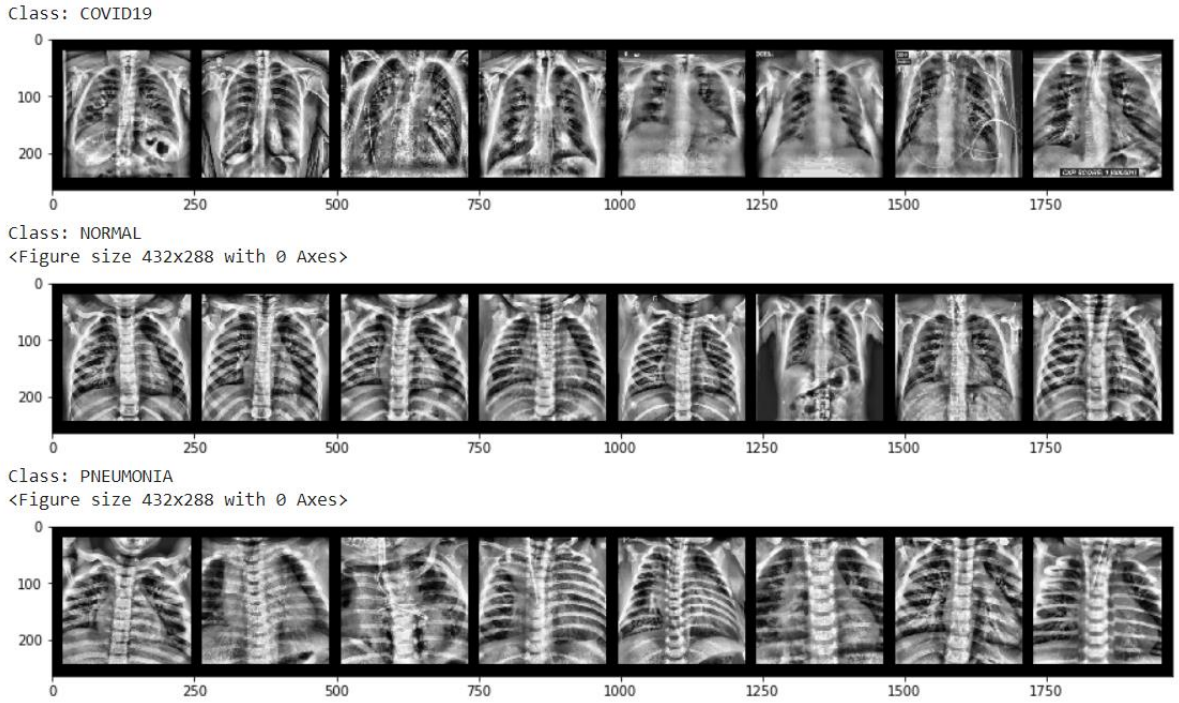


Figure 7. Representative chest X-ray images of COVID-19, pneumonia, and normal/healthy conditions after applying CLAHE with default parameters. Note the noise in COVID19 class, it will be amplified.

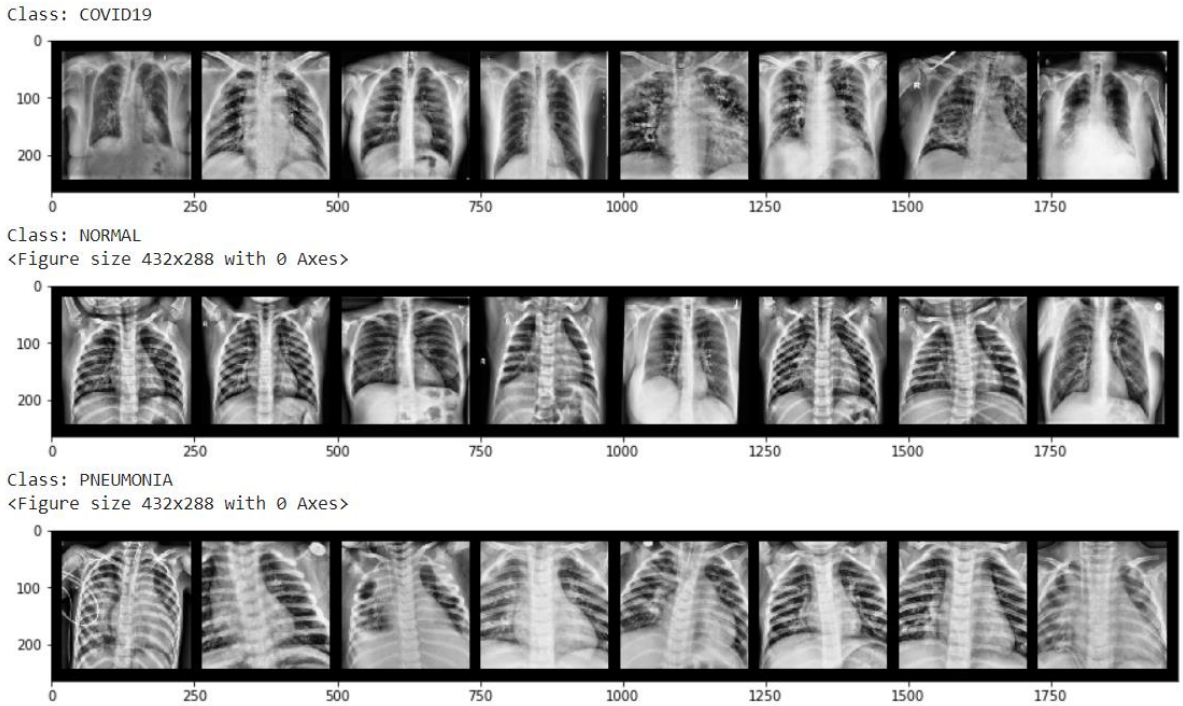


Figure 8. Representative chest X-ray images of COVID-19, pneumonia, and normal/healthy conditions after applying CLAHE with cliplimt parameter = 3.0.

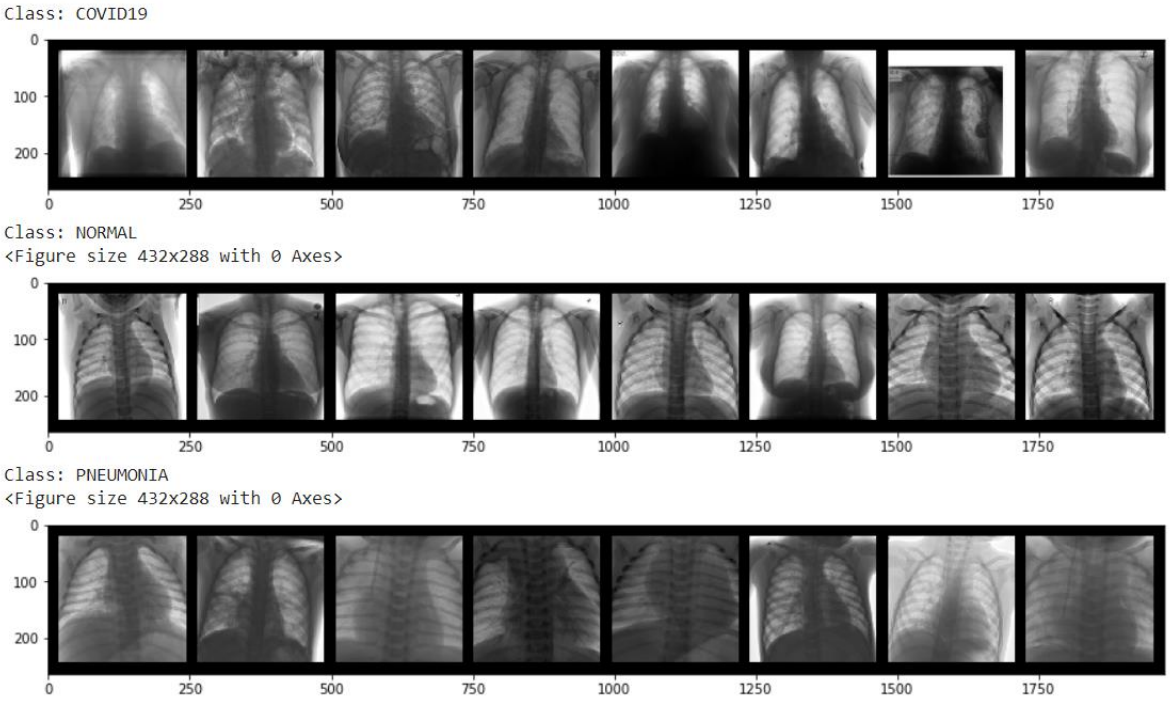


Figure 9. Representative chest X-ray images of COVID-19, pneumonia, and normal/healthy conditions after applying invert image technique.

4.3.2 Classification Results

The results of the DLH_COVID model are presented in this section. The results obtained using Histogram equalization technique and the learning rate is 1.00E-03 are presented in **Table 3**. Shows that among these different number of epochs, when a model with 21 epochs outperforms the others with the highest test accuracy of 96.78%. It attains a precision of 96.78%, while the recall and F1 scores are 96.78%, and 96.78%, respectively. On the other hand, when a model with 20 epochs provided an accuracy of 96.34% and a precision of 96.34%. The recall is recorded as 96.34% and it gave an F1 score of 96.3%. When a model with 22 epochs provided an accuracy of 95.57% and a precision of 95.64%. The recall is recorded as 95.57% and it gave an F1 score of 95.57%. When a model with 23 epochs provided an accuracy of 95.34% and a precision of 95.40%. The recall is recorded as 95.34% and it gave an F1 score of 95.35%. When a model with 24 epochs provided an accuracy of 95.57% and a precision of 95.57%. The recall is recorded as 95.57% and it gave an F1 score of 95.55%. When a model with 25 epochs provided an accuracy of 95.68% 1 and a precision of 95.71%. The recall is recorded as 95.68% and it gave an F1 score of 95.64%.

Table 3. Classification results using Histogram equalization technique and the learning rate is 1.00E-03.

Enhancement technique	Learning_rate	Epochs	Accuracy	Precision	Recall	F1-score
Histogram equalization	1.00E-03	20	0.963455	0.963452	0.963455	0.96327
		21	0.967885	0.967806	0.967885	0.967812
		22	0.955703	0.95648	0.955703	0.955789
		23	0.953488	0.954098	0.953488	0.953517
		24	0.955703	0.955797	0.955703	0.955568
		25	0.956811	0.957106	0.956811	0.956462

The results obtained using Histogram equalization technique and the learning rate is 1.00E-02 are presented in **Table 4**. Shows that among these different number of epochs, when a model with 22 epochs outperforms the others with the highest test accuracy of 0.967885. It attains a precision of 0.967806, while the recall and F1 scores are 0.967885, and 0.967812, respectively. On the other hand, when a model with 20 epochs provided an accuracy of 0.957918 and a precision of 0.958. The recall is recorded as 0.957918 and it gave an F1 score of 0.957609. When a model with 21 epochs provided an accuracy of 0.945736 and a precision of 0.946753. The recall is recorded as 0.945736 and it gave an F1 score of 0.945293. When a model with 23 epochs provided an accuracy of 0.953488 and a precision of 0.954248. The recall is recorded as 0.953488 and it gave an F1 score of 0.952988. When a model with 24 epochs provided an accuracy of 0.937984 and a precision of 0.939875. The recall is recorded as 0.937984 and it gave an F1 score of 0.937849. When a model with 25 epochs provided an accuracy of 0.956811 and a precision of 0.957403. The recall is recorded as 0.956811 and it gave an F1 score of 0.956629.

Table 4. Classification results using Histogram equalization technique and the learning rate is 1.00E-02.

Enhancement technique	Learning_rate	Epochs	Accuracy	Precision	Recall	F1-score
Histogram equalization						
	1.00E-02	20	0.957918	0.958	0.957918	0.957609
		21	0.945736	0.946753	0.945736	0.945293
		22	0.967885	0.967806	0.967885	0.967812
		23	0.953488	0.954248	0.953488	0.952988
		24	0.937984	0.939875	0.937984	0.937849
		25	0.956811	0.957403	0.956811	0.956629

The results of the proposed technique are presented in this section. The results obtained using Adaptive histogram equalization technique and the learning rate is 1.00E-03 are presented in **Table 5.** Shows that among these different number of epochs, when a model with 20 epochs outperforms the others with the highest test accuracy of 0.950166. It attains a precision of 0.950197, while the recall and F1 scores are 0.950166, and 0.949957, respectively. On the other hand, when a model with 21 epochs provided an accuracy of 0.946844 and a precision of 0.946817. The recall is recorded as 0.946844 and it gave an F1 score of 0.94681. When a model with 22 epochs provided an accuracy of 0.942414 and a precision of 0.942802. The recall is recorded as 0.942414 and it gave an F1 score of 0.942524. When a model with 23 epochs provided an accuracy of 0.944629 and a precision of 0.944929. The recall is recorded as 0.944629 and it gave an F1 score of 0.944519. When a model with 24 epochs provided an accuracy of 0.945736 and a precision of 0.946545. The recall is recorded as 0.945736 and it gave an F1 score of 0.945909. When a model with 25 epochs provided an accuracy of 0.945736 and a precision of 0.94575. The recall is recorded as 0.945736 and it gave an F1 score of 0.945736.

Table 5. Classification results using Adaptive histogram equalization technique and the learning rate is 1.00E-03.

Enhancement technique	Learning_rate	Epochs	Accuracy	Precision	Recall	F1-score
	1.00E-03	20	0.950166	0.950197	0.950166	0.949957
		21	0.946844	0.946817	0.946844	0.94681
		22	0.942414	0.942802	0.942414	0.942524
		23	0.944629	0.944929	0.944629	0.944519
		24	0.945736	0.946545	0.945736	0.945909
		25	0.945736	0.94575	0.945736	0.945736
daptive histogram equalizatio	cliplimt = 3.0					

The results of the proposed technique are presented in this section. The results obtained using Adaptive histogram equalization technique and the learning rate is 1.00E-02 are presented in **Table 6**. Shows that among these different number of epochs, when a model with 22 epochs outperforms the others with the highest test accuracy of 96.12%. It attains a precision of 96.20%, while the recall and F1 scores are 96.12%, and 96.13%, respectively. On the other hand, when a model with 20 epochs provided an accuracy of 95.79% and a precision of 95.79%. The recall is recorded as 95.79% and it gave an F1 score of 95.79%. When a model with 21 epochs provided an accuracy of 96.01% and a precision of 96.03%. The recall is recorded as 96.01% and it gave an F1 score of 96.02%. When a model with 23

epochs provided an accuracy of 93.90% and a precision of 94.02%. The recall is recorded as 93.90% and it gave an F1 score of 93.90%. When a model with 24 epochs provided an accuracy of 95.90% and a precision of 95.97%. The recall is recorded as 95.90% and it gave an F1 score of 95.91%. When a model with 25 epochs provided an accuracy of 93.57% and a precision of 93.70%. The recall is recorded as 93.57% and it gave an F1 score of 93.60%.

Table 6. Classification results using Adaptive histogram equalization technique and the learning rate is 1.00E-02.

Enhancement technique	Learning_rate	Epochs	Accuracy	Precision	Recall	F1-score
Adaptive histogram equalizatio	cliplimt = 3.0					
	1.00E-02	20	0.957918	0.957973	0.957918	0.957917
		21	0.960133	0.960382	0.960133	0.960202
		22	0.96124	0.962033	0.96124	0.961381
		23	0.939092	0.940288	0.939092	0.939022
		24	0.959025	0.959764	0.959025	0.95918
		25	0.93577	0.937039	0.93577	0.936006

The results obtained using Inverted equalization technique and the learning rate is 1.00E-03 are presented in **Table 7**. Shows that among these different number of epochs, when a model with 25 epochs outperforms the others with the highest test accuracy of 95.33%. It attains a precision of 95.33%, while the recall and F1 scores are 95.34%, and 95.33%, respectively. On the other hand, when a model with 20 epochs provided an accuracy of 93.13% and a precision of 93.39%. The recall is recorded as 93.13% and it gave an F1 score of 93.02%. When a model with 21 epochs provided an accuracy of 95.23% and a precision of 95.24%. The recall is recorded as 95.23% and it gave an F1 score of 95.23%. When a model with 22 epochs provided an accuracy of 95.12% and a precision of 95.11%. The recall is recorded as 95.12% and it gave an F1 score of 95.12%. When a model with 23 epochs provided an accuracy of 94.46% and a precision of 94.49%. The recall is recorded as 94.46% and it gave an F1 score of 94.42%. When a model with 24 epochs provided an accuracy of 94.68% and a precision of 94.71%. The recall is recorded as 94.68% and it gave an F1 score of 94.65%.

Table 7. Classification results using Inverted equalization technique and the learning rate is 1.00E-03.

Enhancement technique	Learning_rate	Epochs	Accuracy	Precision	Recall	F1-score
Inverted equalization	1.00E-03	20	0.93134	0.933923	0.93134	0.930219
		21	0.952381	0.952411	0.952381	0.952363
		22	0.951274	0.951182	0.951274	0.951212
		23	0.944629	0.94492	0.944629	0.944273
		24	0.946844	0.947162	0.946844	0.946587
		25	0.953488	0.953396	0.953488	0.953367

The results obtained using Inverted equalization technique and the learning rate is 1.00E-02 are presented in **Table 8**. Shows that among these different number of epochs, when a model with 21 epochs outperforms the others with the highest test accuracy of 95.12%. It attains a precision of 95.17%, while the recall and F1 scores are 95.12%, and 95.08%, respectively. On the other hand, when a model with 20 epochs provided an accuracy of 95.01% and a precision of 95.13%. The recall is recorded as 95.01% and it gave an F1 score of 94.95%. When a model with 22 epochs provided an accuracy of 89.03% and a precision of 91.16%. The recall is recorded as 89.03% and it gave an F1 score of 89.14%. When a model with 23 epochs provided an accuracy of 93.90% and a precision of 94.40%. The recall is recorded as 93.90% and it gave an F1 score of 93.83%. When a model with 24 epochs provided an accuracy of 95.12% and a precision of 95.12%. The recall is recorded as 95.12% and it gave an F1 score of 95.10%. When a model with 25 epochs provided an accuracy of 94.90% and a precision of 94.92%. The recall is recorded as 94.90% and it gave an F1 score of 94.87%.

Table 8. Classification results using Inverted equalization technique and the learning rate is 1.00E-02.

Enhancement technique	Learning_rate	Epochs	Accuracy	Precision	Recall	F1-score
Inverted equalization						
	1.00E-02	20	0.950166	0.951306	0.950166	0.949508
		21	0.951274	0.951714	0.951274	0.95084
		22	0.890365	0.911691	0.890365	0.891421
		23	0.939092	0.940421	0.939092	0.938391
		24	0.951274	0.951291	0.951274	0.951094
		25	0.949059	0.949203	0.949059	0.948765

Figure 10 shows the confusion matrix of the highest accuracy

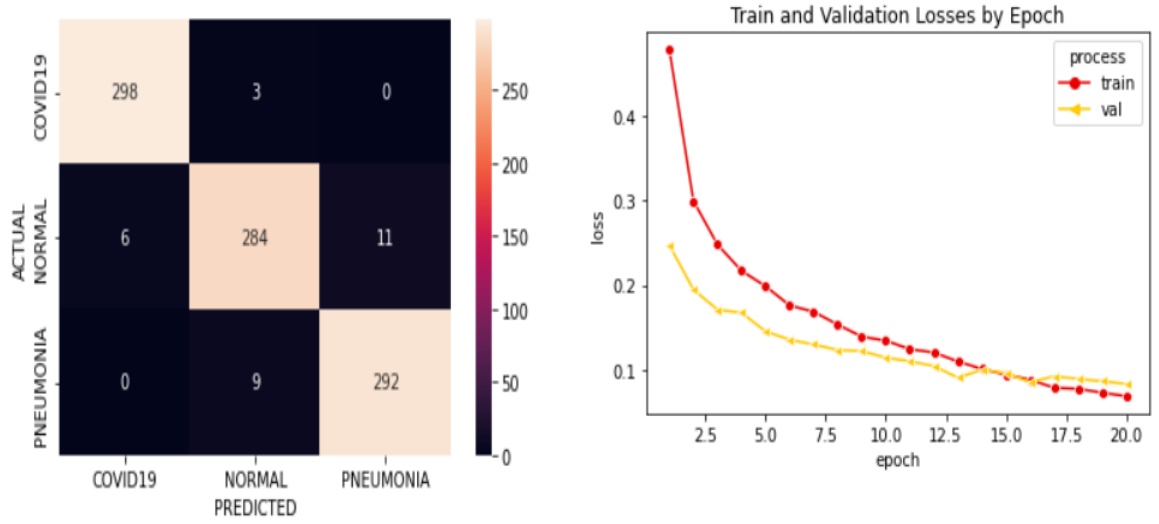


Figure 10. Confusion matrix of best value and plotting of training and validation loss

4.3.3 COVID-19 detection user interface

Lastly, we integrated the DLH_COVID model with **Gradio** library which makes simple interface to detect COVID-19. This AI system is an application that takes chest X-ray image as input data, processes image data through the DLH_COVID classification model and generates probability score of the image class as output. The image class corresponds to COVID-19, pneumonia, and normal/healthy condition. Since the probability score is a measure to determine the class of the uploaded chest X-ray image, the image class with the highest score is predicted as the final output of the AI model. For example, **Figure 11** shows that DLH_COVID was able to accurately predict the presence of pneumonia from an uploaded image. It estimated a high probability score >95% in the image class of pneumonia. Similarly, AI model enables users to detect COVID-19 from chest X-ray images as well, demonstrating that the model can execute multi-class image classification in real time.

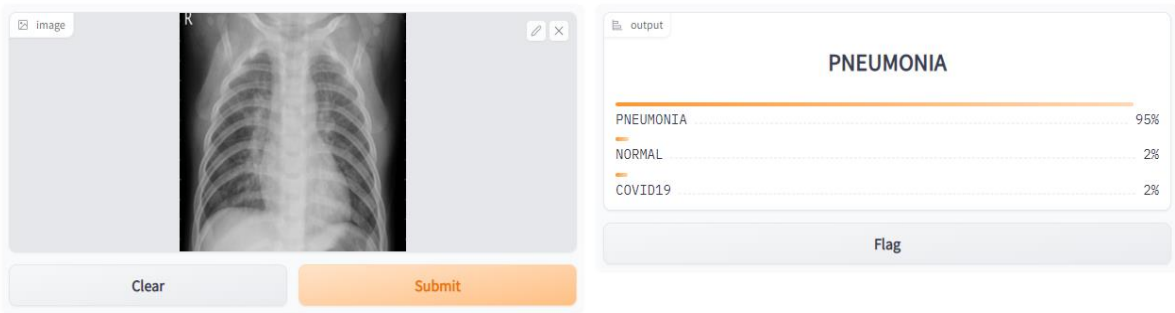


Figure 11. Simple user interface using gradio library

4.4 Testing & Evaluation

Model performance was evaluated using the common statistical measure confusion matrix from which we obtained various metrics like accuracy, precision, recall and f1-score. In **Table 9**, we show the evaluation metrics of both pre-trained models and DLH_COVID based on 10% validation dataset. We considered accuracy score as the best statistical measure to compare performance of the pre-trained models with that of the DLH_COVID model. Initial analysis of validation data revealed the following: (1) none of the pre-trained models achieved more than 89% accuracy score. **Table 9** (2) DLH_COVID model with a simpler architectural design was able to outperform these pre-trained models in terms of detection of COVID-19 from 10% image validation dataset.

Table 9. depicts the preliminary evaluation metrics of different models acquired for this image classification task. We used 'accuracy' column values to measure performance of each model and selected DLH_COVID, since it exhibited the highest accuracy.

Model	Accuracy	Precision	Recall	F1-score
VGG16	0.86711	0.86748	0.86711	0.867026
VGG19	0.887043	0.887508	0.887043	0.887198
ResNet18	0.893688	0.900239	0.893688	0.895066
ResNet50	0.872647	0.872494	0.872647	0.87231
ResNet101	0.873754	0.874185	0.873754	0.873352
ResNet34	0.86711	0.867559	0.86711	0.867234
Inception	0.881506	0.881677	0.881506	0.881587
DLH_COVID	0.967885	0.967806	0.967885	0.967812

Chapter 5: Discussion, Conclusions

5.1 Discussion

As mentioned earlier, for detection of COVID-19, most of the research has been performed using chest X-rays, which shows the importance of chest X-rays in diagnosing chest infections and, specifically, for diagnosing COVID-19, how researches for multiclass classification of COVID-19 is limited, and how it is very important to improve their performance. We used DLH_COVID model which can predict the presence of COVID-19 in a chest X-ray image and distinguish it from pneumonia or normal condition.

We attempted to handle imbalance class challenge that the first dataset was faced **[45]**, fine-tune the hyperparameters (learning rate, epochs) of model through trial and-error approach, as theoretically it is impossible to determine the optimal hyper parameters without going through a comprehensive series of training cycles, and we also attempted to enhance the quality of the CXR images using image enhancement techniques such as histogram equalization, contrast limited adaptive histogram equalization, and invert image . The experimental results during prospective validation phase suggests that the histogram equalization technique at epoch 21 and learning rate 1e-3 got the highest accuracy 96.7% in classifying COVID-19, pneumonia, and normal/healthy cases from the image dataset.

However, there were 2٩ images misclassified by the DLH_COVID model during the test stage. it maybe appears to you as slight difference with the accuracy that the model's owners achieved, but it's better of them because they got in their research result 21 images misclassified from 58 samples of test dataset and this tell us that there is bias in their results, but after we made the dataset balanced the samples of test dataset was 301 as it showed in **Table 2.**

The results show to us how different image enhancement techniques with different hyperparameters got some bad accuracy, but because our limited resources in training stage we couldn't try other techniques like gamma correction, balance contrast enhancement technique and of course tuning hyperparameters with different values to be more confident of our results. We also have to mention that we tried different values for cliplimit parameter of contrast limited adaptive histogram equalization such as 2.0, 3.0 **Figure 8**, and 40.0 **Figure 7** but there is still noise in CXR images.

The recent success of an AI system in carrying out similar X-ray image classification task supplements our objective of developing a user interface system driven by DLH_COVID model. On the contrary, another recent study revealed that applicability of deep learning models in real hospital management ecosystem is still unclear. Thus, it is imperative for more assessments to be made to assert the reliability of AI systems as an important tool for COVID 19 diagnosis. Therefore, we have ensured that the DLH_COVID based desktop application is publicly accessible so that doctors and radiologists can easily test the underlying AI model in clinical settings and capture results accordingly. Critical feedback from medical professional will provide additional guidance to improve the DLH_COVID AI model. This will eventually benefit COVID-19 clinical management settings in pandemic hotspots with an accurate and fast diagnosis process in the foreseeable future.

5.2 Summary & Conclusion

A deep learning-based technique is proposed for the classification of different chest infections. The proposed automated system can differentiate chest infections after the evaluation of chest X-ray images. Histogram equalization is used as a preprocessing tool to enhance the images as the data are gathered from different sources. We trained our model with different epochs and learning rate, the best case was when we applied histogram equalization technique with the learning rate is 1.00E-03 at the number of epochs 21, the test accuracy was 96.78%. We passed our data and compared our model with different types of pre trained models, they were resnet18, resnet34, resnet50, resnet101, vgg16, vgg19 and inception.

References (or Bibliography)

[1] (2020). WHO Director-General's Opening Remarks at the Media Briefing on COVID-19–11 March 2020. [Online]. Available: https://www.who. int/dg/speeches/detail/who-director-general-s-opening-remarks-at-themedia-briefing-on-covid-19—11-march-2020

[2] (2020). Coronavirus Disease 2019 (COVID-19). [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/ people-at-higher-risk.html

[3] Global COVID-19 Report, World Health Organization, Geneva, Switzerland, Mar. 2020.

[4] J. H. U. Medicine. (2020). Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU). [Online]. Available: https://coronavirus.jhu.edu/map.html

[5] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, and W. Tan, ''Detection of SARS-CoV-2 in different types of clinical specimens,'' Jama, vol. 323, no. 18, pp. 1843–1844, 2020.

[6] T. Yang, Y.-C. Wang, C.-F. Shen, and C.-M. Cheng, Point-of-Care RNABased Diagnostic Device for COVID-19. Basel, Switzerland: Multidisciplinary Digital Publishing Institute, 2020.

[7] A. J. News. (2020). India's Poor Testing Rate May Have Masked Coronavirus Cases. [Online]. Available: https://www. aljazeera.com/news/2020/03/india-poor-testing-rate-masked-coronaviruscases-200318040314568.html

[8] A. J. News. (2020). Bangladesh Scientists Create 3 Kit. Can it Help Detect COVID-19. [Online]. Available: https://www.aljazeera. com/news/2020/03/bangladesh-scientists-create-3-kit-detect-covid-19- 200323035631025.html

[9] N. Wetsman. (2020). Coronavirus Testing Shouldn'T Be This Complicated. [Online]. Available: https://www.theverge.com/2020/3/17/21184015/ coronavirus-testing-pcr-diagnostic-point-of-care-cdc-techonology

[10] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, Y. Zhao, Y. Li, X. Wang, and Z. Peng, ''Clinical characteristics of 138 hospitalized patients with 2019 novel Coronavirus— Infected pneumonia in Wuhan, China,'' JAMA, vol. 323, no. 11, p. 1061, Mar. 2020.

[11] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei, J. Xia, T. Yu, X. Zhang, and L. Zhang, ''Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, China: A descriptive study,'' Lancet, vol. 395, no. 10223, pp. 507–513, Feb. 2020.

[12] Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K. S. Leung, E. H. Lau, J. Y. Wong, and X. Xing, ''Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia,'' New England J. Med., vol. 382, pp. 1199–1207, Jan. 2020.

[13] C. Huang et al., ‘‘Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China,’’ Lancet, vol. 395, pp. 497–506, Feb. 2020.

[14] V. M. Corman et al., ‘‘Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR,’’ Eurosurveillance, vol. 25, no. 3, Jan. 2020.

[15] D. K. W. Chu, Y. Pan, S. M. S. Cheng, K. P. Y. Hui, P. Krishnan, Y. Liu, D. Y. M. Ng, C. K. C. Wan, P. Yang, Q. Wang, M. Peiris, and L. L. M. Poon, ‘‘Molecular diagnosis of a novel coronavirus (2019- nCoV) causing an outbreak of pneumonia,’’ Clin. Chem., vol. 66, no. 4, pp. 549–555, Apr. 2020.

[16] N. Zhang, L. Wang, X. Deng, R. Liang, M. Su, C. He, L. Hu, Y. Su, J. Ren, F. Yu, L. Du, and S. Jiang, ‘‘Recent advances in the detection of respiratory virus infection in humans,’’ J. Med. Virology, vol. 92, no. 4, pp. 408–417, Apr. 2020.

[17] M. Chung, A. Bernheim, X. Mei, N. Zhang, M. Huang, X. Zeng, J. Cui, W. Xu, Y. Yang, Z. A. Fayad, A. Jacobi, K. Li, S. Li, and H. Shan, ‘‘CT imaging features of 2019 novel coronavirus (2019-nCoV),’’ Radiology, vol. 295, no. 1, pp. 202–207, Apr. 2020.

[18] M. Hosseiny, S. Kooraki, A. Gholamrezanezhad, S. Reddy, and L. Myers, ‘‘Radiology perspective of coronavirus disease 2019 (COVID-19): Lessons from severe acute respiratory syndrome and middle east respiratory syndrome,’’ Amer. J. Roentgenology, vol. 214, no. 5, pp. 1078–1082, May 2020.

[19] S. Salehi, A. Abedi, S. Balakrishnan, and A. Gholamrezanezhad, ‘‘Coronavirus disease 2019 (COVID-19): A systematic review of imaging findings in 919 patients,’’ Amer. J. Roentgenology, vol. 215, pp. 1–7, Mar. 2020.

[20] Jain G, Mittal D, Thakur D, Mittal MK. A deep learning approach to detect COVID-19 coronavirus with X-Ray images. Biocybern. Biomed. Eng. (2020); 40(4):1391–405

[21] Apostolopoulos ID, Mpesiana TA. COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys. Eng. Sci. Med. (2020); 43:635–40

[22] Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput. Biol. Med. (2020); 121:103792

[23] Ahuja AS, Reddy VP, Marques O. Artificial Intelligence and COVID-19: A Multidisciplinary Approach. Integr. Med. Res. (2020); 9(3):100434

[24] Gunther Correia Bacellar, Mallikarjuna Chandrappa, Rajlakshman Kulkarni, Soumava Dey. COVID-19 Chest X-Ray Image Classification Using Deep Learning. (2021) https://www.medrxiv.org/content/10.1101/2021.07.15.21260605v1

[25] LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1989, 1, 541–551. [CrossRef]

[26] Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir, Serkan Kiranyaz, Saad Bin Abul Kashem, Mohammad Tariqul Islam, Somaya Al Maadeed, Susu M Zughaier, Muhammad Salman Khan, Muhammad E. H. Chowdhury. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-rays Images. (2020) https://arxiv.org/abs/2012.02238

[27] Wikipedia web page, https://en.wikipedia.org/wiki/Histogram_equalization

[28] Wikipedia web page, https://en.wikipedia.org/wiki/Adaptive_histogram_equalization

[29] Mahmud, T.; Rahman, M.A.; Fattah, S.A. 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. 2020, 122, 103869.

[30] Umair, M.; Khan, M.S.; Ahmed, F.; Baothman, F.; Alqahtani, F.; Alian, M.; Ahmad, J. Detection of COVID-19 Using Transfer Learning and Grad-CAM Visualization on Indigenously Collected X-ray Dataset. Sensors 2021, 21, 5813.

[31] Li, L.; Qin, L.; Xu, Z.; Yin, Y.; Wang, X.; Kong, B.; Bai, J.; Lu, Y.; Fang, Z.; Song, Q. Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology 2020, 296, 200905.

[32] Abbas, A.; Abdelsamea, M.M.; Gaber, M.M. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. Appl. Intell. 2021, 51, 854–864. [CrossRef]

[33] Wang, S.; Kang, B.; Ma, J.; Zeng, X.; Xiao, M.; Guo, J.; Cai, M.; Yang, J.; Li, Y.; Meng, X.; et al. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). Eur. Radiol. 2021, 31, 6096–6104.

[34] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.

[35] Shankar, K.; Perumal, E. A novel hand-crafted with deep learning features based fusion model for COVID-19 diagnosis and classification using chest X-ray images. Complex Intell. Syst. 2021, 7, 1277–1293.

[36] Panwar, H.; Gupta, P.K.; Siddiqui, M.K.; Morales-Menendez, R.; Singh, V. Application of deep learning for fast detection of COVID-19 in X-rays using nCOVnet. Chaos Solitons Fractals 2020, 138, 109944.

[37] Zheng, C.; Deng, X.; Fu, Q.; Zhou, Q.; Feng, J.; Ma, H.; Liu, W.; Wang, X. Deep Learning-Based Detection for COVID-19 from Chest CT Using Weak Label. medRxiv 2020.

[38] Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.

[39] Xu, X.; Jiang, X.; Ma, C.; Du, P.; Li, X.; Lv, S.; Yu, L.; Ni, Q.; Chen, Y.; Su, J.; et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering 2020, 6, 1122–1129.

[40] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 770–778.

[41] Hussain, E.; Hasan, M.; Rahman, M.A.; Lee, I.; Tamanna, T.; Parvez, M.Z. CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. Chaos Solitons Fractals 2021, 142, 110495.

[42] Khan, A.I.; Shah, J.L.; Bhat, M.M. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. Comput. Methods Programs Biomed. 2020, 196, 105581.

[43] Chowdhury, M.E.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Mahbub, Z.B.; Islam, K.R.; Khan, M.S.; Iqbal, A.; al Emadi, N.; et al. Can AI help in screening viral and COVID-19 pneumonia? IEEE Access 2020, 8, 132665–132676.

[44] Ozturk, T.; Talo, M.; Yildirim, E.A.; Baloglu, U.B.; Yildirim, O.; Acharya, U.R. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput. Biol. Med. 2020, 121, 103792.

[45] Patel P. Chest X-ray (COVID-19 & Pneumonia). Kaggle. (2020); https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia

[46] COVID-19 Radiography Database-Kaggle. Available online: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database(accessed on 20 November 2021).

[47] Shamout FE, Shen Y, Wu N, Kaku A, Park J, Makino T, et al. An artificial intelligence system for predicting the deterioration of COVID-19 patients in the emergency department. npj Digit. Med. (2020); 4, 80.

[48] BIMCV-COVID-19, Datasets Related to COVID-19's Pathology Course. 2020. Available online: https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711(accessed on 20 November 2021).

[49] Kaggle. RSNA Pneumonia Detection Challenge. Available online: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data(accessed on 20 November 2021).

[50] Jain G, Mittal D, Thakur D, Mittal MK. A deep learning approach to detect COVID-19 coronavirus with X-Ray images. Biocybern. Biomed. Eng. (2020); 40(4):1391–405

[51] Guan Q, Wan X, Lu H, Ping B, Li D, Wang L, et al. Deep convolutional neural network Inception-v3 model for differential diagnosing of lymph node in cytological images: a pilot study. Ann. Transl. Med. (2019); 14:307

[52] Gunther Correia Bacellar, Mallikarjuna Chandrappa, Rajlakshman Kulkarni, Soumava Dey. COVID-19 Chest X-Ray Image Classification Using Deep Learning.
doi: https://doi.org/10.1101/2021.07.15.21260605