COLOMBO JOURNAL OF MULTI-DISCIPLINARY RESEARCH

COLOMBO JOURNAL OF
MULTI-DISCIPLINARY RESEARCH

Volume 08 - No. 01 & 02 - March & November 2023

Faculty of Graduate Studies - University of Colombo

Diagnostic of COVID-19 Pneumonia through Convolutional Neural Networks Using Chest X-RAY

K. R. A. L. Kahatapitya [a], G. P. Liyanage [a] ✉, K. G. S. U. R. Bandara [a], K. B. A. Madushi [b]

[a] *Postgraduate Institute of Science, University of Peradeniya, Sri Lanka*
[b] *Department of Information Technology, SIBA Campus, Sri Lanka*

Abstract

This Study to emphasizes the need for improved diagnostic protocols and increased awareness to effectively manage COVID-19 and its complications, particularly pneumonia, to alleviate the burden on healthcare systems, underscores the critical importance of early identification of COVID-19 pneumonia as a strategic approach to mitigate devastating impact and fast detection of underlying symptoms. Introducing a novel model for detecting COVID-19 pneumonia, utilizing chest X-ray images available on open-source platform, and convolutional neural networks, enabling precise diagnosis in binary classification settings. Two steps followed to enhance classification accuracy and avoid Overfitting: (1) enlarging the data set while maintaining the balance of the classification scenarios; (2) incorporating regularization techniques and performing hyper-parameter optimization. The model is ideal for deployed locally with limited capacities and without an Internet access. Because of the network size, the model capacity reduced immensely. Comparison to literature, the final model performed better and required a disproportionately higher parameters while reaching a classification accuracy of 99.63% and model sensitivity of 93.75% for the binary cases. The models can be uploaded to a digital platform for quick diagnosis and make up for lack of professionals, and RT-PCR (reverse transcription polymerase chain reaction).

Keywords: Chest X-Ray, Convolution Neural Network, Covid-19, Deep Learning, Pneumonia, Radiography.

Received:	Accepted revised version:	Published:
26 May 2023	31 October 2023	30 November 2023

Suggested citation: Kahatapitya, K. R. A. L., Liyanage, G. P., Bandara, K. G. S. U. R., & Madushi, K. B. A. (2023). Diagnostic of COVID-19 pneumonia through convolutional neural networks using chest X-RAY. *Colombo Journal of Multi-Disciplinary Research, 08* (01 & 02), 1-20.

✉ Corresponding Author: pamuditha777@gmail.com

DOI: https://doi.org/10.4038/cjmr.v8i1-2.73
ISSN: 2362-0633

Introduction

On December 31, 2019, the World Health Organisation (Nour et al., 2020) issued a warning about several cases of respiratory illness emerging from the city of Wuhan, in the Hubei province of China, with clinical manifestations like viral pneumonia and symptoms such as coughing, fever, and dyspnea. This newly discovered virus was named COVID-19, and it is caused by the SARS-CoV-2 virus. Most people who are infected with COVID-19 experience mild to moderate respiratory illnesses and may recover without requiring special treatment. Older people and those with underlying medical problems such as cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious complications. An efficient way to prevent and reduce infections is by quickly detecting positive COVID-19 cases, isolating patients, and starting their treatment as soon as possible. COVID-19 spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes (Belman-López, 2022). There are currently no specific treatments for COVID-19 (Gunraj et al., 2019) (E. Capobianco, M. D. Dominietto, M. Aiello, H.Gunraj, L. Wang, and A. Wong, 2019) and, as the global pandemic progresses, scientists from a wide variety of specialties play pivotal roles in developing new diagnostic, forecasting, and modeling methods. For example, Sun and Wang (2020) developed a mathematical model to characterise imported and asymptomatic patients without strict 'quarantine', which was useful in modeling the COVID-19 epidemic. Mathematical models and deep learning approaches are used to present an evolutionary modeling of COVID-19 (Belman-López, 2022). However, regarding diagnostic and detection methods, much of the current literature has focused on radiology techniques such as portable chest radiography (X-rays) and computed tomography (CT) scans (Tahir et al., 2020; Taresh et al., 2021) for instance, used CT images and pre-trained deep learning models together with discriminant correlation analysis to classify COVID-19, pneumonia, tuberculosis, and healthy cases (Tahir et al., 2020; Zargari Khuzani et al., 2021). However, X-rays will probably be the most popular method for identifying and monitoring lung abnormalities, including COVID-19, due to the infection problems associated with patient transportation to CT suites, inefficiencies in CT room decontamination, and a shortage of CT equipment in some regions of the world. Additionally, for around an hour after the imaging of an infected patient, CT rooms may not be available for new examination, but it would depend on the air exchange rates on the CT suites. Therefore, early COVID-19 discovery by radiograph is important in regions of the globe where accessibility to accurate RT-PCR testing for COVID-19 was limited.

Furthermore, among the most prominent tools in the fight against the COVID-19 pandemic are artificial intelligence (AI) methods because AI methods are cost-effective and good for third-world and developing countries. Castillo and Melin (2020)(O. Castillo and P. Melin, 2020) proposed a model for forecasting confirmed cases and deaths in different countries based on their corresponding time series using a hybrid fuzzy-fractal approach.

As of March 2021, 127 349, 248 confirmed cases of COVID-19, including 2 787, 593 deaths, had been reported worldwide (Belman-López, 2022), and the numbers keep rising. Therefore, it is critical to identify COVID-19 cases as soon as possible to prevent the pandemic from spreading further and to treat infected people immediately. Auxiliary diagnostic tools are now more important than before, and recent findings from radiology suggest that X-ray radiographs can provide valuable information about COVID-19 infections. AI methods coupled with radiology techniques can be helpful for the accurate detection of this disease and can also help overcome the shortage of radiologists and experts in remote places (Apostolopoulos & Mpesiana, 2020; Belman-Lopez, 2022; Mahmud et al., 2020; Taresh et al., 2021).

AI methods, especially machine learning, and deep learning have been used successfully in many areas, such as Internet of Thing (IoT) manufacturing, computer vision, autonomous vehicles, natural language processing, robotics, robotics, education, and healthcare (Belman-López, 2022). Machine learning and deep learning techniques like linear discriminant analysis, support vector machines, and feed-forward neural networks have been used in healthcare to solve a variety of classification problems, such as classifying lung diseases, separating cardiac sound signals efficiently, and classifying motor imagery electroencephalogram signals to improve brain-computer interfaces. Convolutional neural networks (CNNs), on the other hand, are the best way to classify images and do a better job with nonlinear problems in high-dimensional spaces. The word ‘simplicity’ refers to the fact that CNNs might not need feature engineering. Instead, they use simple models that can be trained from beginning to end. CNNs can be scaled up because they work well with parallelisation TPUs. CNNs can also be trained by going over large or small batches of data repeatedly. This means they can be trained on datasets of any size. Versatility and reusability are seen as features because, unlike many previous machine-learning methods, CNNs can be trained on more data without having to start from scratch. This makes them suitable for continuous learning. Also, trained CNNs can be used again and again. For example, CNNs that have been trained to classify images can be dropped into a pipeline for processing video.

This makes it possible to reinvest and reuse previous work to build models that are more complex and powerful. Also, new techniques and research are being done to improve the performance of CNNs. These include the novel convolutional block attention module, data augmentation techniques, new architectures and pre-trained models, optimisation algorithms, activation functions, and regularisation techniques such as dropout (Belman-Lopez, 2022; Elshennawy & Ibrahim, 2020; Mahmud et al., 2020; Tahir et al., 2020).

Lastly, CNNs can be put into small devices that are the right size so that people who are not trained can use them to diagnose many kinds of illnesses in places that aren't well-developed (Belman-López, 2022)

Literature Review

CNNs have been used for signal processing and biomedical image processing, for classifying multidimensional and thermal images, and for image segmentation. CNNs have several features that make them good tools for AI and image processing (Apostolopoulos & Mpesiana, 2020; Bar et al., 2015; Belman-López, 2022; Gunraj et al., 2019; Mahmud et al., 2020; Nour et al., 2020). Recently, radiology images (X-ray images) in conjunction with deep learning methods (such as CNNs) have been investigated for COVID-19 detection to eliminate disadvantages such as the insufficient number of RT-PCR test kits, testing costs, and the long wait for results.

Table 1

Results Literature Review

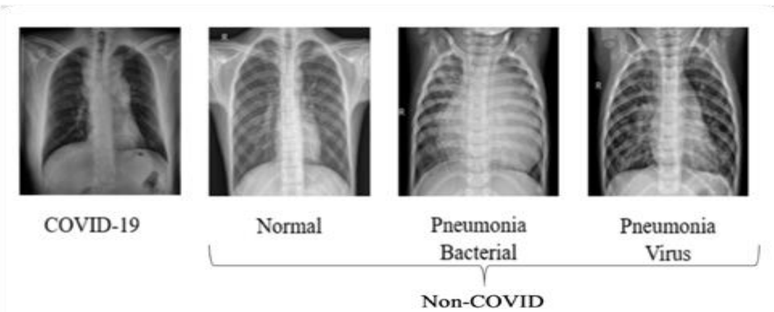
No	Article	Results
1	Used traditional neural networks with transfer learning from the ImageNet database (Liu and Deng, 2015)	93.3 % Classifying 3 class (COVID-19, normal, and pneumonia)
2	A deep learning model using 224 confirmed COVID-19 images (Ozturk et al., 2020)	98.75 % and 87.02 % in classifying 2 and 3 classes, respectively.
3	CovXNet: Using different data set (Mahmud et al., 2020)	97.4% for COVID/Normal, 96.9% for COVID/Viral pneumonia, 94.7% for COVID/Bacterial pneumonia, and 90.2% for multi-class COVID/normal/Viral/Bacterial types of pneumonia
4	Deep feature extraction, and pre-trained deep CNN models (ResNet18, ResNet50, ResNet101, VGG16, and VGG19) were used and various kernel functions were used (Ismael & Şengür, 2021).	ResNet50 model and support vector machine (SVM) classifier with the Linear kernel function produced a 94.7% accuracy score ResNet50 model was found to be 92.6% The developed CNN model produced a 91.6%
5	Presented two algorithms including deep neural network (DNN) on the fractal feature of images and convolutional neural network (CNN) methods with the use of the lung images, directly. Results classification shows that the presented (Hassantabar et al., 2020).	CNN architecture with a higher accuracy of 93.2% DNN method with an accuracy of 83.4%
6	The categorical scenario (4 classes) analysed a total of 3 829 images, 980 for each category (Belman-López, 2022).	Classification accuracy of 99.17 % and 94.03 % for the binary and categorical scenarios, respectively.

Table 1 summarises the results of several research articles on COVID-19 image classification using deep learning techniques. The first article uses traditional neural networks with transfer learning from the ImageNet database and achieves an accuracy of 93.3%. The second article develops a deep learning model using 224 confirmed COVID-19 images and achieves an accuracy of 98.75% and 87.02% in classifying 2 and 3 classes, respectively. The third article, CovXNet, uses different datasets and achieves accuracies ranging from 94.7% to 97.4% for various classifications. The fourth article uses deep feature extraction and pre-trained deep CNN models and achieves an accuracy of 94.7% with a ResNet50 model and support vector machine (SVM) classifier with a linear kernel function. The seventh article presents two algorithms, including a deep neural network (DNN) on the fractal feature of images and a convolutional neural network (CNN) method, and achieves an accuracy of 93.2% with the CNN architecture. The ninth article analyses a total of 3,829 images and achieves a classification accuracy of 99.17% and 94.03% for the binary and categorical scenarios, respectively.

According to the literature, results are biased and overfitted because the network capacity is so big compared to the small number of COVID-19 images. Also, there can only be two or three classes in each scenario. It is important to note that larger networks tend to overfit, while simpler models are less likely to do this. On the other hand, approaches based on transfer learning (Belman-López, 2022) used pre-trained models from datasets that had little to do with X-ray images and needed a lot of fine-tuning to change the abstract representations in the models to make them relevant for the problem at hand. But compared to the size of the dataset, they would end up with more capacity than they needed. Additionally, another drawback of transfer learning approaches is that they result in very big and heavy models, thus making it difficult to deploy them on devices with limited capacities or acquire internet access if these large models are deployed in the cloud, which is not always possible in remote or poor regions (Belman-López, 2022) or these reasons, this paper's contribution is to show new ways to find COVID-19 Pneumonia using chest X-ray images and CNNs. The proposed models were made to provide accurate diagnostics in more output classes than previous studies, covering binary classification (COVID-19 vs. non-COVID). Normal, bacterial pneumonia and viral pneumonia are taken as non-COVID x-ray images. Figure 1 illustrates examples of the images used in this study.

Figure 1

Examples of the images in the dataset, used in this study.



To make the results more accurate and stop overfitting, two steps can be taken:

1. Increase the size of the data set (by adding more data repositories and using data augmentation techniques) while keeping classification scenarios balanced.
2. Add regularisation techniques like dropout and automate hyperparameter optimisation.

The extensive exploration of crucial hyperparameters, including batch size, learning rate, number of epochs, image resolution, activation functions, the depth of convolutional layers, layer dimensions, and neuron configurations within the dense layers, formed the cornerstone of our research. We harnessed contemporary deep learning toolkits, namely Hyperas and Hyperopt (Belman-López, 2022), to systematically investigate these factors. Our meticulous hyperparameter tuning resulted in a noteworthy achievement. In direct comparison to existing studies within the scientific literature, our final model offerings exhibited remarkable efficiency and resource optimisation. These outcomes not only highlight the rigorous nature of our experimentation but also underscore the practical benefits of our model configurations in terms of computational economy and space utilisation.

Methods

The proposed method is illustrated as shown in Figure 2 in which chest x-ray images are used as input to the proposed covid 19 detection method.

Data Set

In this research, the used dataset is open-source, and it is taken from the Kaggle. There are 4099 COVID-19-infected x-ray images and 5361 non-COVID x-ray images, such as viral pneumonia, bacterial pneumonia, and a normal patient's chest radiograph.

In the literature, not many X-ray images were used to train the model. That could lead to a situation called ‘overfitting’. In the current study more X-ray images to train the model and avoid overfitting. When it came to the training and validation data sets, the current study depended on the training model because sometimes tried to train overfitting on purpose, and sometimes used a balanced data set for training. Though all the data sets were used, 10% of the datasets were taken as valid data sets, and sets were taken as training data sets. Below mentioned, six data sets were combined and created the dataset for the current study.

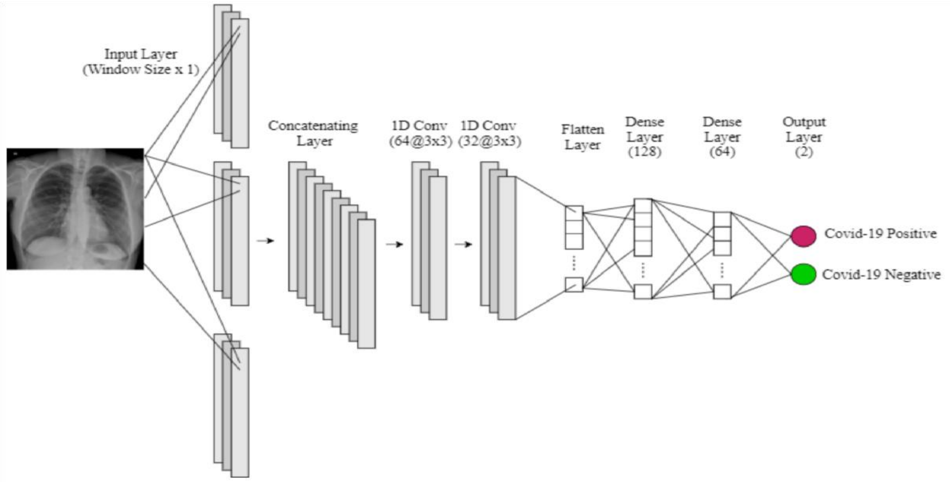
1. Covid Chest X-Ray Dataset (Anon., n.d.)
2. Pneumonia dataset by Praveen (Praveen, 2020)
3. COVID19 chest XRAY analysis by SAIMANASA_C (C, 2020)
4. COVID19 with Pneumonia and Normal Chest Xray(PA) Dataset by Amanullah Asraf (Asraf, 2020)
5. HASH_Directors - Covid19 by AMRUTH AMBRISHK (Ambrishk, 2020)
6. RICORD COVID-19 X-ray positive tests by RADDAR (RADDAR, 2020)

Experimental Setup

The CNNs were developed using Python as the main library as TensorFlow and Keras libraries. Training and experiments were conducted on a computer equipped with chip Apple M1 Pro, 16 GB of RAM, 500 GB HDD, and MacOS 13.0 and Kaggle repository with GPU.

Figure 2

Proposed Model Architecture



Preprocessing

The goal of data preprocessing is to make raw data better suited for neural networks. So, the X-ray images that were taken were already processed to make the training process faster and easier. Before processing, the images were resized so that they were all the same shape (Belman-López, 2022). This was followed by value normalisation (VN), which involved dividing the values by 224 and casting them to float type so that the inputs were floating-point values in the 0-1 range.

Deep Convolutional Neural Networks (DCNN)

One of the most important ways to learn deeply is through convolutional neural networks (CNN), in which several layers are taught well. CNN uses many different perspectives with layers and layers to curtail preprocessing. A CNN has three main layers: the convolutional layer, the pooling layer, and the fully connected layer. Each layer has a different job to do.

There are two stages of training in each architecture: forward propagation and backpropagation. In the forward step, the information only moves forward from the hidden layer to the output layer. The value from the input layer is sent to the hidden layer, and then an output is made. When a neural network is set up with backpropagation, the input image is given to the network in the first stage. When the output is reached, the network error value is calculated (Hassantabar, 2020). This value and the cost function diagram are then sent back to the network to update the network weights (as illustrated in Figure 2). Convolutional Layer: The convolution layer is the most important part of a convolution network, and its output can be thought of as a pile of neurons in three dimensions. Simply put, this layer's result is a three-dimensional pile. In these layers, the CNN network convolves the input image and the central feature maps by using different kernels. There are three main benefits to convolution (Hassantabar, 2020)

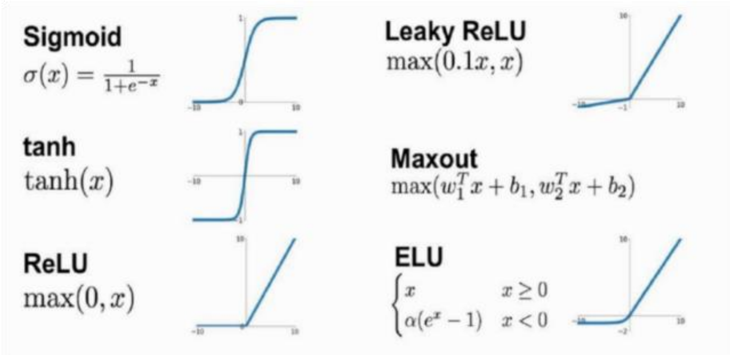
- The number of parameters is greatly reduced by the way weights are shared in each feature map.
- The local connection finds out how the neighbouring pixels are linked.
- It keeps the object stable

Kernels: Kernel helps separate data with a non-linear decision boundary using a linear classifier.

Activation function: The activation functions are the main mathematical logic functions of a neural layer. Each neural layer has a separate activation function given by the programmer. They are divided into two types, linear and non-linear functions.

Figure 3

Activation function



Source: Jayawardana (n.d.)

Since the required neural network is working based on the digital inputs and outputs, the Sigmoid and the Relu activation functions are more feasible to be included in the neural layers (Jayawardana & Bandaranayake, 2021). Max pooling: There are many effects of max pooling in neural networks. When max pooling is used, the network can figure out what the object is right away, even if only small changes are made to the image. Second, it gives the network more space to find features in the image. Pooling in the CNN is used to summarise the features during the subtracting sampling task so that we can get to deeper layers of the network. When we reach the end of each stage and want to cut back on sampling, the sampling reduces the amount of space we can use to store spatial information. So, if we want to keep this information, we need to start working together to collect what we have. Max and Average are the two most common ways to pooling (Belman-López, 2022).

Dense layer: A Dense Layer is a simple layer of neurons where each neuron gets input from all the neurons in the layer above it. Therefore, it is called dense. The output from convolutional layers is used by the Dense Layer to sort images into groups.

Training Parameters

The CNNs were trained with a backpropagation algorithm to minimize the cross-entropy loss function (binary cross-entropy for the binary scenario), with the dropout set to 0.5 to prevent overfitting and weights updated with the Adam optimiser. After each convolution, there was an activation function for nonlinear activation, and then there was a max pooling layer with a 2x2 window and no stride. The number of steps per epoch was found by dividing the total number of training samples by the size of each batch. The number of validation steps was found by dividing the total number of testing samples by the size of each testing batch. During the training and testing phases, accuracy was used as a way to measure progress. In the binary scenario, relu activation and sigmoid activation were used to normalise the probability prediction in the output layer. A dense layer was used before the output layer.

Table 2

Propose Model Summary

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 109, 109, 64)	73792
activation_1 (Activation)	(None, 109, 109, 64)	0
max_pooling2d_1 (MaxPooling 2D)	(None, 54, 54, 64)	0
conv2d_5 (Conv2D)	(None, 52, 52, 32)	18464
activation_2 (Activation)	(None, 52, 52, 32)	0
max_pooling2d_2 (MaxPooling 2D)	(None, 26, 26, 32)	0
flatten (Flatten)	(None, 21632)	0
dropout (Dropout)	(None, 21632)	0
dense (Dense)	(None, 128)	2769024
dropout_1 (Dropout)	(None, 128)	0

dense_1 (Dense)	None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dropout_3 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 2)	66
Total Params		3,316,354

The provided TensorFlow model architecture consists of a series of layers, including Conv2D, Activation, MaxPooling2D, Flatten, Dropout, and Dense layers. The key characteristics of these layers are summarised below:

1. Conv2D Layer: This layer has 64 filters, produces an output shape of (109, 109, 64), and contains 73,792 parameters.
2. Activation Layer (Activation_1): Applied after the first Conv2D layer, it retains the same output shape of (109, 109, 64) and does not introduce any additional parameters.
3. MaxPooling2D Layer (MaxPooling2D_1): This layer reduces the output shape to (54, 54, 64) and does not have any parameters.
4. Conv2D Layer (Conv2D_5): The second Conv2D layer, with 32 filters, yields an output shape of (52, 52, 32) and consists of 18,464 parameters.
5. Activation Layer (Activation_2): Following the second Conv2D layer, this layer maintains the output shape at (52, 52, 32) and does not introduce additional parameters.
6. MaxPooling2D Layer (MaxPooling2D_2): Reduces the output shape to (26, 26, 32) and has no parameters.
7. Flatten Layer: Flattens the data to a shape of (21,632) and contains no parameters.
8. Dropout Layer: Applied after flattening, this layer has no parameters and keeps the same shape.
9. Dense Layer (Dense): Comprises 128 neurons and contains 2,769,024 parameters.
10. Dropout Layer (Dropout_1): Applied after the Dense layer, it retains the same output shape and contains no parameters.
11. Dense Layer (Dense_1): Consists of 64 neurons and has 8,256 parameters.
12. Dropout Layer (Dropout_2): Applied after Dense_1, it retains the same output shape and does not introduce additional parameters.
13. Dense Layer (Dense_2): Contains 32 neurons and 2,080 parameters.
14. Dropout Layer (Dropout_3): Applied after Dense_2, this layer retains the same shape and contains no parameters.
15. Dense Layer (Dense_3): The final layer with 2 neurons and 66 parameters.

In total, this model has approximately 3,316,354 parameters.

A. Method Type I

Dataset: A balanced data set was used to train and test the model that was proposed. Before beginning to train the model, all images are set same size and collected as a NumPy array. After splitting the data array into "train_data", "test_data", "train_target", and "test_target". Train_data and train_target are used to train the model and test_data and test_target are used to validate the model.

Platform: Kaggle and the Apple M1 Macbook Pro Computer

Training: The epochs count is 25 and used train data set and validate set to train the model

Models: Use model-changing trainable parameter size count and echo count on both platforms while training. Model Training: While training, collect data based on the below parameters and plot the training history over epochs.

- Training and validation accuracy model
- Training and validation Loss model
- Sparse categorical accuracy model
- Mean squared error in model.
- Mean squared absolute error in the model.
- Mean absolute percentage error in the model.
- Mean squared logarithmic error in model.
- Cosine similarity in model
- Log-Cosh Loss

Model Evaluation: After the model has been trained, evaluate it using the methods for model validation below, then plot the outcome.

- Model Evaluation Results
- Confusion Matrix
- Sensitivity / True Positive Rate
- Specificity / True Negative Rate
- ROC AUC Score
- ROC Curve
- Classification Report
- Accuracy Score
- Balance Accuracy Score
- Average precision score
- Precision Score
- Recall Score
- F1 Score
- Binary accuracy
- Categorical accuracy

B. Method Type II

As with method I, this part also runs, but the only change is that an identical data set was used to train it.

Result and Discussion

These results were taken after training 60 models in the Kaggle platform and training 5 models in the M1 MacBook computer. Considering high accuracy two models to discuss in this section. The selected model belongs to the same trainable parameters count group and it is showing the best result on the Apple M1 MacBook machine. There is only a difference in used imbalance and balanced data set to train the model. The balance data set model gives 99.63% accuracy. But it compares with another model it increments is 0.16%.

Table 3

Result Table on the selected model.

Model	Accuracy	Loss	Data Description			Model Details
			Covid Positive	Covid Negative	Total Images	Total Param
1	99.63%	0.80%	4,099	4,099	8,198	3,316,354
2	99.47%	1.70%	4,099	5,361	9,460	3,316,354

In the context of two different models, the table presents their performance metrics. Model 1 demonstrates a high accuracy of 99.63% with a relatively low loss of 0.80%. It is trained on a dataset consisting of 4,099 Covid-positive and 4,099 Covid-negative images, making a total of 8,198 images. The model has a parameter count of 3,316,354. On the other hand, Model 2 achieves a slightly lower accuracy of 99.47% and a somewhat higher loss of 1.70%. Its training dataset is also composed of 4,099 Covid-positive images, but a higher number of Covid-negative images, totaling 9,460 images. The parameter counts for Model 2 matches that of Model 1 at 3,316,354.

Selected model training parameter evaluations plot and some model evaluation results are shown on figure 4 -12 below.

Training parameter evaluations plots

Figure 4

Training and Validation Accuracy Model

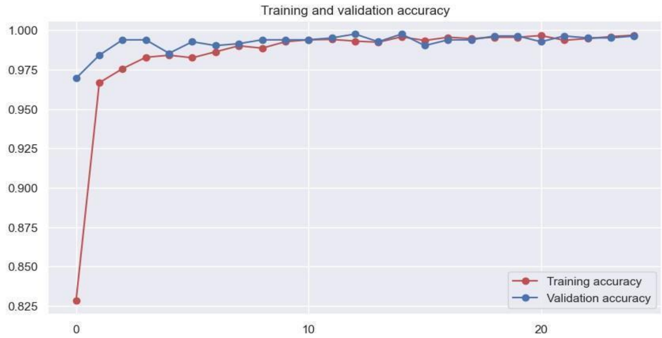


Figure 5

Training and Validation Loss Model

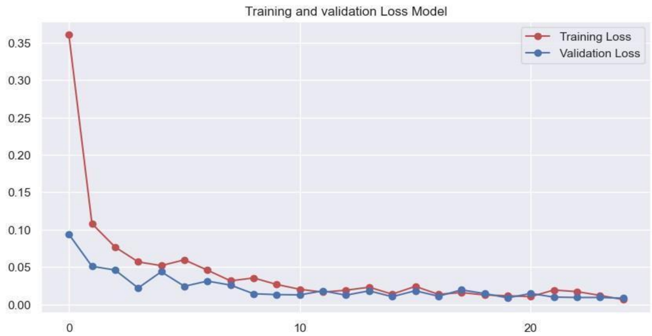


Figure 6

Sparse categorical accuracy Model

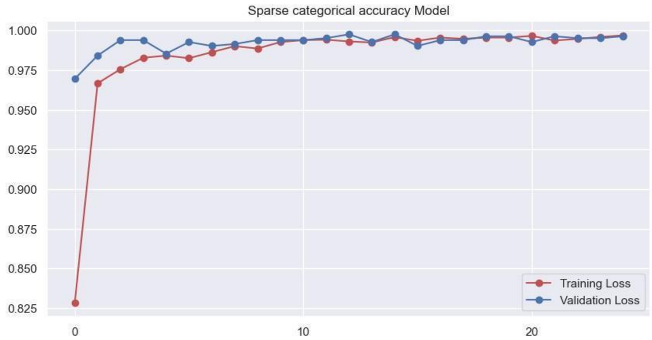


Figure 7

Mean squared error in the model

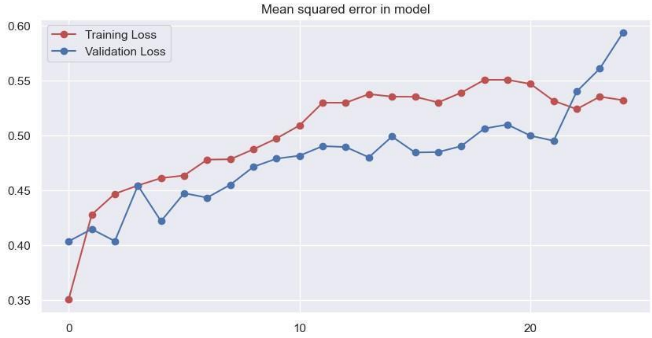


Figure 8

Mean squared absolute error in the model

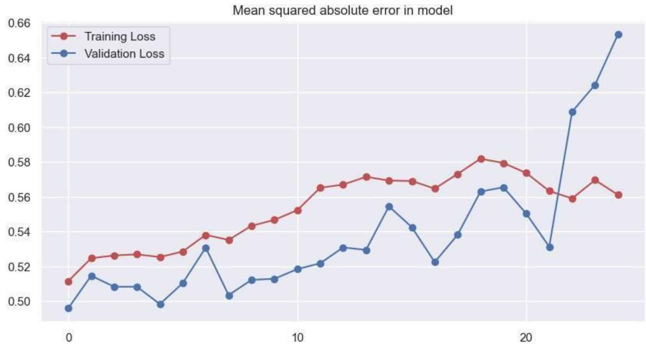


Figure 9

Mean absolute percentage error in the model

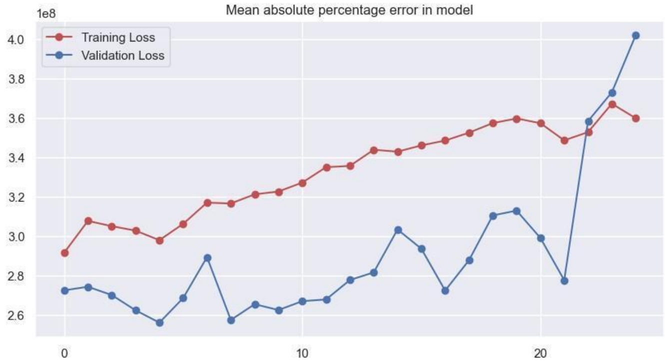


Figure 10

Mean squared logarithmic error in the model

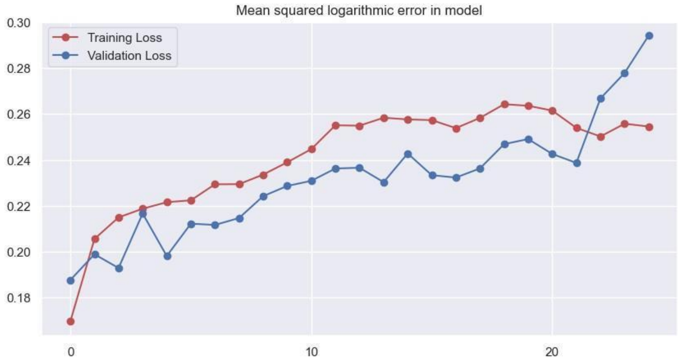


Figure 11

Cosine similarity in model

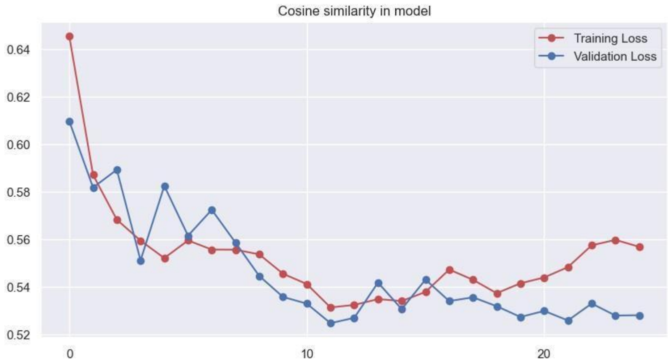
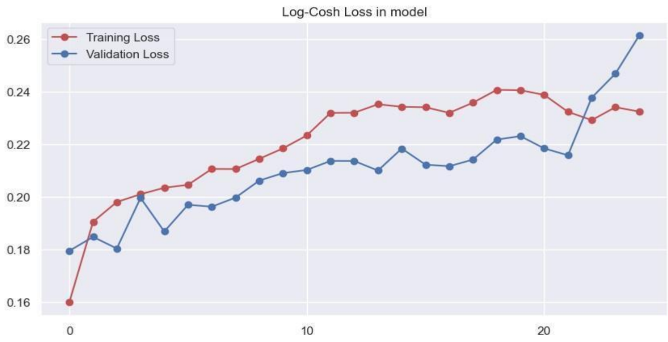


Figure 12

Log-Cosh Loss



Selected model evaluation result tables and plots.

Table 4

Selected model evolution result table

Model Evaluation Results		
	loss	accuracy
	0.0083	0.9963
(%)	0.83%	99.63%

Figure 13

Confusion Matrix

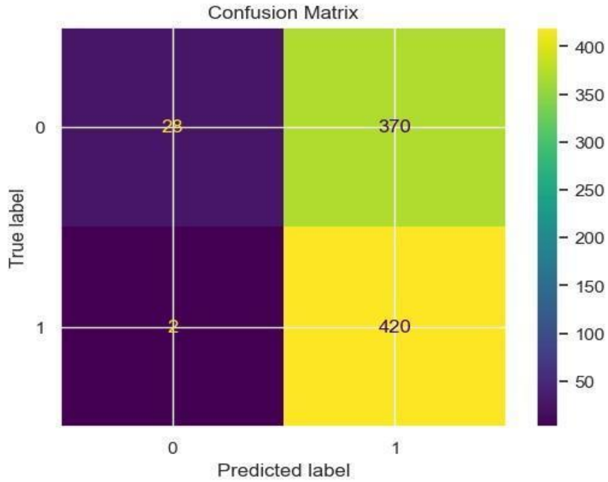


Figure 14

ROC Curve

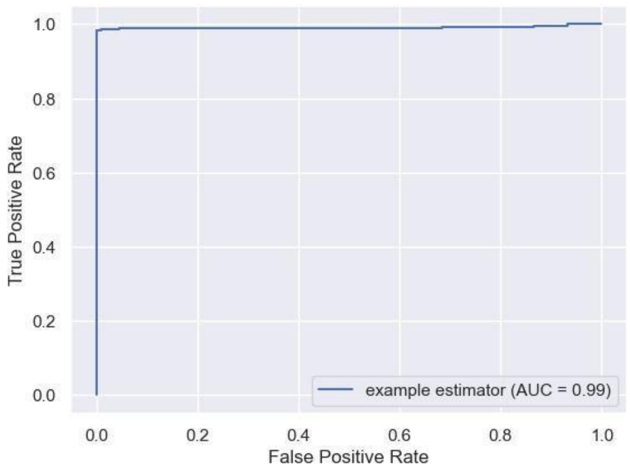


Table 5

Classification Report

Classification Report				
	precision	recall	f1-score	support
0	0.93	0.07	0.13	398
1	0.53	1	0.69	422
accuracy			0.55	820
macro avg	0.73	0.53	0.41	820
weighted avg	0.73	0.55	0.42	820

Figure 15

Normal chest x-ray image GradCAM plot

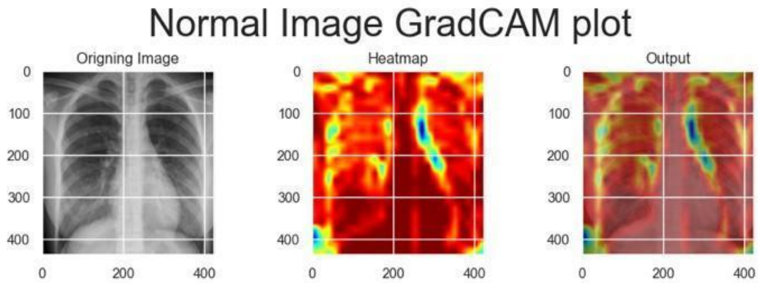


Figure 16

Virus pneumonia chest x-ray image GradCAM plot taken as non-Covid image

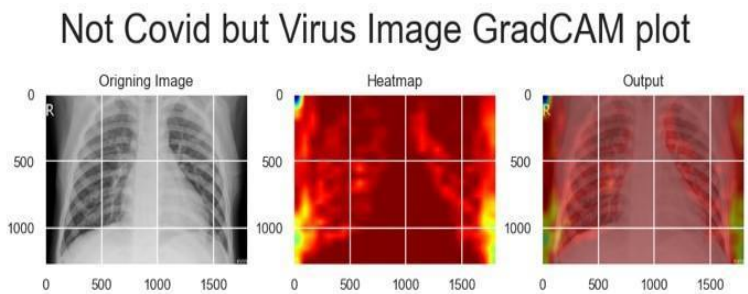
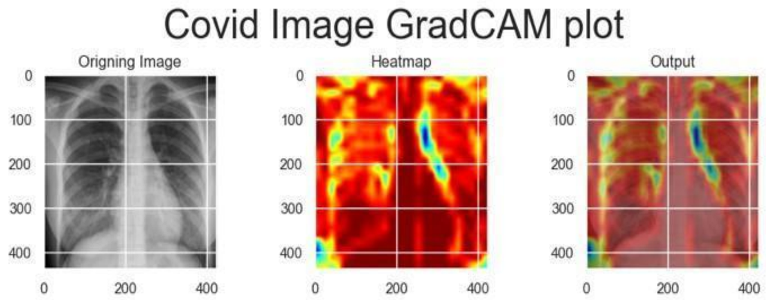


Figure 17

Covid Positive patient image GradCAM Plot



Conclusion and Future Research

COVID-19 is a naval viral disease which is causing COVID-19 pneumonia. Viruses belonging to this family can cause the same pneumonia and can be easily detected in the future, using this model. The advantage of it is early detection in a general X-ray imaging device. Chest X-rays are commonly used to identify upper respiratory infections by chest physicians, and this can be a vital addition. The proposed and proven method is more accurate than the previously developed methods mentioned in the literature even though it merely takes the binary classification method into account. The fact that the balance data set performs better than the imbalance data set suggests that the balance dataset helped avoid overfitting.

We cannot predict what sort of diseases can be triggered in the future, but respiratory diseases are more common and hard to prevent spreading. Early detection is the key to control the spread. Most viral diseases cause bacterial illnesses. So, hope to focus to build the COVID-19, bacterial and viral pneumonia detection model and do a comparison with previous training models like ResNet50, and VGG16.

References

Amruthambrish, A. (2020). HASH_Directors - Covid19. Retrieved from www.kaggle.com: https://www.kaggle.com/code/amruthambrish/hash-directors-covid19/data

Apostolopoulos, I. D., Mpesiana, T. A. (2020). Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med, 635-640.

Asnaoui, K. el, Chawki, Y. (2020). Using X-ray images and deep learning for automated detection of coronavirus disease.

Belman-López, C. E. (Apr. 2022). Detection of COVID-19 and other pneumonia cases using convolutional neural networks and X-ray images. Ingenieria.

Blumenfeld, A., Greenspan, H., Konen, E. (2018). Pneumothorax detection in chest radiographs using convolutional neural networks. 3.

Capobianco, E., Dominietto, M. D., Aiello, M., Gunraj, H., Wang, L., Wong, A. (2019). COVIDNetCT: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases From Chest CT Images. Frontiers in Medicine.

Castillo, O., Melin, P. (2020). Forecasting of COVID-19 time series for countries in the world based on a hybrid approach combining the fractal dimension and fuzzy logic. Chaos Solitons Fractals,.

Cohen, J. P. (2020, September 05). Covid chest x-ray dataset. Retrieved from Github: https://github.com/ieee8023/covid-chestxray-dataset

Degerli, A. (2020, September). COVID-19 Infection Map Generation and Detection from Chest X-Ray Images. Retrieved from http://arxiv.org/abs/2009.12698

Elshennawy, N. M., Ibrahim, D. M. (2020). DeepPneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images," .

Groza, V., Kuzin, A. (2020). Pneumothorax. Segmentation with Effective Conditioned PostProcessing in Chest X-Ray. International Symposium on Biomedical Imaging Workshops. ISBI Workshops 2020.

Hassantabar, S., A. (2020). Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches. Chaos Solitons Fractals, 110-170.

Ismael, A. M., Şengür, A. (2020). Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl.

Jayawardana, R., S. (n.d.). ANALYSIS OF OPTIMIZING NEURAL NETWORKS AND ARTIFICIAL INTELLIGENT MODELS FOR GUIDANCE, CONTROL, AND NAVIGATION SYSTEMS Utilisation of Fixed wing Drones for mars Exploration View project Optimizing Neural Networks in Engineering Applications. Retrieved from www.irjmets.com : www.irjmets.com

Liu, S., Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. 3rd IAPR Asian Conference on Pattern Recognition (ACPR),. 3rd IAPR Asian Conference on Pattern Recognition (ACPR),.

Mahmud, T., A. Rahman, M., Fattah, S. A. (2020). 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 optimisation. Comput Biol Med.

Narin, A., Kaya, C., Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Analysis and Applications, 1207–1220.

Nour, M. M., Zhu, N., Ali, T. A. A., Hameed, A. S., Mutar, M. L. (2021). Transfer Learning to Detect COVID-19 Automatically from XRay Images Using Convolutional Neural Networks. Int J Biomed Imaging, 1-9.

Ozturk, T., M. T. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med.

Praveen. (2020). Pneumonia dataset. Retrieved from www.kaggle.com: https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset

RADDAR:. (2020). RICORD COVID-19 X-ray positive tests Data sets. Retrieved from www.kaggle.com/datasets: https://www.kaggle.com/datasets/raddar/ricordcovid19-xray-positive-tests

Saimanasa, Chadal (2020). COVID19 chest XRAY analysis - Dataset. Retrieved from kaggle: https://www.kaggle.com/code/saimanasachadal

Sohrabi, C. (Apr, 2020). World Health Organisation declares global emergency: A review of the 2019 novel coronavirus (COVID-19). International Journal of Surgery, 71-76.

Taresh, M. M., Zhu, N., Ali, T. A. A., Hameed, A. S., Mutar, M. L. (2021). Transfer Learning to Detect COVID-19 Automatically from XRay Images Using Convolutional Neural Networks. Int J Biomed Imaging, 1-9.

Tahir, A. (2020). Deep Learning for Reliable Classification of COVID-19, MERS, and SARS from Chest X-Ray Images CRediT authorship contribution statement.

Zargari Khuzani, A., Heidari, M., Shariati, S. A. (2021). COVID-Classifier: an automated machine learning model to assist in the diagnosis of COVID-19 infection in chest X-ray images. Sci Rep.