COVID-19 DETECTION AND ANALYSIS ON CXRs USING VARIOUS CNN MODELS

A Capstone project Report

submitted in partial fulfilment for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINERING [CSE]

By

Kemal Mudie Tosora (18BCE2433)

Under The guidance of

Prof. Rajkumar S

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May 12, 2022

DECLARATION

We hereby declare that the Project report entitled "***COVID-19 DETECTION AND ANALYSIS ON CXRs USING VARIOUS CNN MODELS***" submitted by us to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of B**. Tech** in **Computer science and engineering** is a record of bonafide work carried out under guidance of **Prof. Rajkumar S**.

We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Signature of the candidate:

Name: Kemal Mudie Tosora

Reg. No: 18BCE2433

CERTIFICATE

This is to certify that the project report entitled "**COVID-19 DETECTION AND ANALYSIS ON CXRs USING VARIOUS CNN MODELS**" submitted by **Kemal Mudie Tosora 18BCE2433**. School of Computer Science and Engineering, Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of **Bachelor of technology** in **Computer science and engineering** is a record of bonafide capstone project undertaken by him/her under my supervision.

The report fulfils the requirements as per the regulations of this Institute and in my opinion, meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Signature of the Supervisor

<Rajkumar S> **SUPERVISOR**

<Associate Professor>

Date:

Date:

Internal Examiner

External Examiner

Approved by

Head of Department

Department of Computer science and Engineering.

School of Computer Science and engineering

ACKNOWLEDGEMENT

We avail this opportunity to express our indebtedness to our guide Dr. Rajkumar S, Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore, for his valuable guidance, constant encouragement, kind and inspired guidance offered at various stages for the execution of this dissertation work.

We would like to extend our gratitude to Prof. Rajkumar S, Head of Department of computer Science and Engineering for providing valuable assistance and insight during the work process. And last but not the least, we thank all those friends who helped us in the course of this entire dissertation work.

Kemal Mudie Tosora(18BCE2433)

Executive Summary

The gold standard and most widely used COVID-19 detection technique is real-time polymerase chain reaction (RT-PCR). However, RT-PCR kits are costly and take 6–9 hours to verify infection in a patient. Due to the lower sensitivity of RT-PCR, it provides high false-negative results. To resolve this problem, radiological imaging techniques like chest X-rays and CT scans are used as alternatives to detect and diagnose COVID-19. In this paper, chest X-rays are preferred over CT scans. The rationale behind this is that X-ray machines are available in most hospitals, and X-rays have lower ionizing radiation than CT scans. Moreover, COVID-19 reveals some radiological signatures that will be easily detected through CXRs. For this, radiologists are required to research these signatures.

However, it's a time-consuming and error-prone task. Hence, there's a necessity to automate. Therefore, our objective is to develop an automatic DL system for the detection of COVID-19 samples from healthy and pneumonia cases using CXR images. Because of the scarcity of the COVID-19 dataset, we used transfer learning techniques, where we examined 3 pre-trained CNN models (VGG-16, ResNet-50, and EfficientNetB3) to search out the most suitable one for this task. A total of 15,153 images (3616 COVID-19 cases, 10,192 healthy and 1345 pneumonia cases) are employed. To perform cross-validation, the samples are divided into 80% for training and 20% for testing. From the experiment, EfficientNetB3 achieves the highest classification accuracy of 99%, with average precision, recall, and F1 scores of 98.3%, 99.9%, and 98%, respectively, and is followed by a VGG-16 accuracy of 98%, with a mean precision of 98.6%, a recall of 97.7%, and an F1 score of 97.7%. The efficacy of deep transfer learning approaches for detecting COVID-19 cases using CXR scans was demonstrated in this study.

Table of Contents

List of figures

List of tables

List of terms and abbreviations

COVID	19-CoronaVirus Disease of 2019
CT	Computed Tomography
RT	PCR- reverse transcription-polymerase chain reaction
PCR	Polymerase Chain Reaction
CNN	Convolutional Neural Network
TL	Transfer Learning
CXR	Chest -X-Ray
ML	Machine Learning
AI	Artificial Intelligence
DL	Deep Learning
GAN	Generative Adversarial Networks
ROC	Receiver Operating Characteristic
FC	Fully Connected
LR	Learning Rate
IB	Identity Block

CB	Convolutional Block
CL	Convolution Layer
AUC	Area Under the Curve

1. INTRODUCTION

1.1. Objective

This study uses chest X-ray images to overcome the lower sensitivity and FPR of RT-PCR. chest X-rays were preferred over CT scans because they are more accessible and cost-effective. Aside from that, X-rays (0.2 mSv) provide less ionizing radiation than CT scans (0.7 mSv). To counter data scarcity and imbalanced data we employed various pre-processing, training techniques, and data augmentation techniques to balancing the samples. The three state-of-the-art pre-trained CNN models (VGG-16, ResNet-50, and effinentNetB3) are used as feature extractors to extract feature vectors from the input images, and the model is compared using a variety of hyperparameters.To conduct a comparative study, we designed a new CNN model and compared its performance to three state-of-the-art CNN models that applied the transfer learning approach (TL).

1.2. Motivation

While RTPCR is by far the most effective way of COVID-19 detection. This method is very time consuming and requires special kits that may not be available in remote regions of a country due to geological, social and economic barriers. On the contrary, the rapid antigen test looks for the presence of antigens of the virus from a nasal swab but suffers from higher rate of false negatives. The serological test looks for the antibodies produced by the immune system against the virus from the blood sample of the patient. However, it only checks the few antibodies during or after recovery and does not help in early virus detection. CT scan and x-ray scans, both use invisible ranges of electro-magnetic spectrum to detect any kind of anomaly, used for early detects and have high clinical relevance. In this paper, we found out the chest x-ray tests are economically affordable, and the results are relatively easy to use. Chest X-ray tests are easily available, have portable versions and low risk of radiation. On the other hand, CT scans have high risk of radiation, are expensive, need clinical expertise to handle and are non-portable. This makes the use of X-ray scans more convenient than CT scans.

1.3. Background and Literature survey

The new COVID-19 (coronavirus disease 2019) pandemic has caused over 516,476,402 million illnesses and 6,258,023 million deaths worldwide as of May 12th, 2022, with a total of 11.5+ million vaccination doses provided [1]. As a result of the virus's ongoing dissemination over the world, the SARS-CoV-2 mutations produced new COVID-19 wave. Among the variants identified were the Beta, XE, and omicron types. There were fears that new variant based on the original SARS-CoV-2 strain might be more extremely contagious. The number of daily fatalities (68.5k) and daily new cases (17.5 million) based on the new COVID strain from mid-April to mid-May 2022. Due to its ease of use in clinical settings, reverse transcription-polymerase chain reaction (RTPCR) is the primary tool used to track the transmission of this contagious virus. Due to the fact that real-time PCR uses primers from different genes, RNA variations in the SARS-CoV-2 genome can affect the primer and probe- target sites, resulting in false negatives (FN). The design of the real-time PCR is based on the conserved regions within the viral genomes; however, variability may result in mismatches between primers and probes and the target sequences, resulting in lower test performance and a greater likelihood of false-negative (FN) results. The development of a covid-19 detection system that can overcome the limitations of RT-PCR is essential to improving healthcare. Imaging has proven to be a promising method for detecting and identifying covid-19 since the outbreak began [2] [3] [4]. Machine learning-based medical image classification techniques have revolutionized the healthcare industry by achieving human-level accuracy. As the pandemic began, both the medical and deep learning communities were interested in CXR images.[5]. In the opinion of many medical professionals, ML technologies could help hospitals fight epidemics through analysis of CXR and CT scans [6]. In country such as China and Italy [7], hospitals have implemented ML-driven X-ray scan analyzers, as well as AI projects to enhance COVID-19 patient triaging (i.e., to discharge, to admit , or to give ICU care) and hospital resource allocation [7] . CXR and CT scans had radiation doses of 0.02 and

0.7 mSv [8] [9], respectively, for PA film. CXR patients are less likely to develop radiation-related disorders including cancer because to the decreased radiation exposure [9]. When a CT scan and an x-ray image are examined for multiple class classification, CXRs are preferred as the first imaging modality when pneumonia is anticipated [10] since the symptoms are similar to covid and the pneumonia signature is easier to identify in CXR than in CT scan images [11]. Furthermore, CXR scans are less expensive than CT scans, making them more cost effective for hospitals and patients. covid-19 on CXR chest images may also be distinguished from other kinds of pneumonia by exploiting the predictive capabilities of ML [12]. It is feasible to identify covid-19 cases from those caused by other forms of pneumonia (viral and bacterial) using DL techniques [13]. To develop DL methods for covid detection label data is mandatory but getting such data require experts. Researcher around the word create data set and make it open for public :[2] and [14] crated Kaggle repository with a total of 15k Xray image where 3600 is covid, 1345 viral pneumonia and 10k normal and Chest-xray8 with a total of 108k crated by [15]. The number of labeled data for covid-19 is very low and difficult to train new CNN models from scratch. To overcome this data scarcity in this study the concept of transfer learning (TL) with fine tuning methods utilizes and examined 3 state of art CNN models and compared with our custom CNN models.

The study by [16] suggested DarkCovidNet for automated covid-19 identification in binary classification (covid-19 vs non-covid-19) and multiclass classification (covid-19 vs non-covid 19 vs Pneumonia). The model attained a 98 % accuracy for binary instances and an 87 % accuracy for multi-class cases, and the author employed DarkNet as a classifier, with 5-fold cross validation used for performance evaluation for both binary and multiple class cases. In [17] a transfer learning with fine-tuning -based approach proposed for COVID-19 that achieved 96.78% classification accuracy for binary classes and 94.72% for multiple classes using Mobile Net v2. The training and performance evaluation is carried out using 10-fold cross-validation.

CapsNet with capsule networks for covid-19 identification utilizing chest X-ray images was proposed in [18]. The suggested networks obtained 97.24 percent and 97.24 percent accuracy for binary (covid-19, and No-Findings) and multi-class classification (covid-19, and No-Findings, and Pneumonia) classification, respectively. The amount of X-ray images is restricted, which causes overfitting issues. To circumvent this, the authors used data augmentation as follows: width and height shift range of 0.2, and horizontal and vertical flip as True. The data is divided into ten folds for training and performance testing.

In [19], in contrast, to transfer learning, the researcher suggested a CNN-based architecture was trained from scratch to recognize features in chest X-ray images. The suggested CNN model was employed as a deep feature extractor, with the extracted output feeding ML algorithms such as k-nearest neighbor, SVM, and decision tree. The Bayesian optimization approach was used to tune the hyperparameters of the ML models. The ML model was optimized using the Bayesian optimization approach, which utilized 70% of the total datasets for training and 30% for testing. Finally, using SVM as a classifier, we acquired a high classification accuracy of 98.97%, a sensitivity of 89.39%, a specificity of 99.75%, and an F-score of 96.72%.

In [20], for data preprocessing, images were rebuilt using stacking and fuzzy approaches, the feature vector retrieved using (MobileNetV2 and SqueezeNet) models, processed using a method called Social Mimic optimization, and the final features categorized by SVM, obtaining an accuracy of 99.27 %.

[21] demonstrated TL approaches and analyzed 15 pre-trained CNN models. The author employed 860 images for the experiment: 260 covid-19, 300 non-covid-19, and 300 pneumonia patients. To explore model performance, 70%, 15%, and 15% of the data for each class are utilized for training, validation, and testing, respectively. VGG19 had the greatest classification accuracy of 89.53%, with an average precision of 90%, a recall of 90%, and an F1 score of 90%.

The study by [22] proposed 2 medical image classification engines, named COV19-CNNet, which is based on the CNN architecture, and covid-ResNet, based on the residual neural network (ResNet) architecture for classification of 3 classes. For performance check, a total of 1000 data sets were used, where 210 covid-19, 350 viral pneumonia, and 350 normal, with a ratio of 60% for training and 20% for validation and testing of CXR images. Finally, they achieved a high classification accuracy of 97.61% from COV19-ResNet and 94.28% from COV19-CNNet.

The study [23] introduced TL-based image classification algorithms and fine-tuned three pre-trained models (Restnet18, Googlenet, and Alexnet). The GAN network was utilized to solve the overfitting problem caused by the limited amount of data points, and it increased the train data. The authors chose the models because of the minimal number of layers in their architectures, which decreases model complexity and consumes less memory, reducing training execution time. The three-model was evaluated in three different scenarios. In the initial instances, the model was tested for 4 categories (COVID-19, non-COVID-19, pneumonia bacterial, and viral pneumonia), and Googlenet attained an accuracy of 80.6 %. In the second scenario, which included three classes (non-covid-19, covid-19, and pneumonia bacterial), Alexnet achieved an accuracy of 85.2 %, and in the third scenario, which included only two classes (COVID-19 and non-COVID-19), GoogleNet achieved a high classification accuracy of 100 % for testing and 99.9 % for validation.

In the study by [4] examined the performance of three DL-based models in their study (Inception V3, Xception, and ResNeXt). For data input, the authors employed data augmentation with the following parameters: resize image to 128 x128 x 3, 10° range of random rotation, horizontal flip set to true, and zoom range of 0.4. Furthermore, they used the LeakyReLU activation function to overcome the drawback of relu inactivation due to zero slope. A total of 6432 CTR scans were used, with 965 for validation and 5467 for training. The Xception model has the highest accuracy of 97.97 % based on the experiment analysis.

Another work [24] detected COVID-19 instances in X-ray and CT scan images using a modified VGG16 CNN-based model. Finally, the researchers found that COVID-19 can be identified with both CT and X-ray images. The suggested X-ray, on the other hand, had a better accuracy score of 97.6 % than the CT scan-based model, which had an accuracy of 93 %.

To distinguish the COVID-19 virus from CXRs, a hybrid DL model based on CNN and GRU was developed [5]. CNN was employed in the study for feature extraction, while GRU was used as a classifier. A total of 424 CXR pictures from three classes (COIVD-19, Pneumonia, and Normal) were used, and the dataset was then divided into three sets: 70% training, 10% validation, and 20% testing, producing acceptable accuracy of 0.96 precision, 0.96 recall, and f1-score of 0.95.

The research provided [25] a combination of two CNN architectures (Xception and ResNet50V2) to improve the capability of multiple feature extraction for multiple class classification. For performance evaluation, 5-fold cross validation is used, and the approach achieves acceptable accuracy, precision, recall, and F1-Score, with values of 0.93, 0.96, 0.99, and 0.98, respectively.

In the study, [26] used the Inception-ResNet-v2 architecture to categorize x-ray images into 3 groups (Normal, Viral Pneumonia, and COVID-19) using the transfer learning approach. The model's performance was examined using different epoch counts; with 32 iterations, the model attained an accuracy of 0.966 on the 29th epoch.

The first model proposed by [27] is efficient-net B3 initialized weights with noisy student training. The second model is SE-Resnext, which integrates squeeze and excitation methods with the resnext model, and the third is efficient net B3, which has been improved through the application of adversarial training procedures. There are 17958, 3432, and 1200 photos used for training, testing, and validation, respectively. Finally, the model's output was ensembled and obtained 95 % accuracy, 95.22 % sensitivity, and 95.77 % specificity on a test set.

To address the data scarcity, [28] proposed a self-augmentation approach in the feature space rather than the data space using RICA, which stands for reconstruction independent component analysis. CNN is used to extract relevant features and generate low-dimensional augmented features, while bidirectional LSTM (BiLSTM) is used for classification. They evaluated three publicly available data sets and obtained 98 % of them right.

Table 1. Comparison with the prior studies that classified CXR images into multiple classes

Author	Data modularity	Covid dataset	Classification model used	Trained and validation matched	Accuracy %
[16]	Chest X-ray images	(500 +)	DarkNet	5-fold cross-validation	87.02
[17]	Chest X-ray images	(224 +)	VGG-19 + MobileNet v2	10-fold cross-validation	96.78
[18]	Chest X-ray images	(1050 +)	CapsNet	10-fold cross-validation	84.22
[19]	Chest X-ray images	(153 +), after augmentation (918 +)	Deep CNN model	Train: Test 70%: 30%	98.97
[20]	Chest X-ray images	(295 +)	MobileNetV2 + SqueezeNet	Train: Test 70%: 30%	99.27

[21]	Chest X-ray images	(295 +)	VGG 16,19+ ResNet 50,101,152+ ResNetV2 50,101,152+DenseNet121,169, 201 + MobileNet + InceptionV3 + InceptionResNetV2 + Xception	5-fold cross-validation	89.3
[22]	Chest X-ray images	(350 +)	COV19-CNNet + COV19-ResNet	Train: Test: Validation 60%: 20:20%	97.61%
[23]	Chest X-ray images	(79 +) + GAN	ALexnet + Googlenet + Resnet18	Train: Test 80%: 20%	85.19
[4]	Chest X-ray images	(5467 +)	InceptionV3+ Xception + ResNet	Train: validation 80%:20%	97.97%
[5]	Chest X-ray images	(142 +)	GRU+CNN	Train: Validation: Test 70%:10%:20%	97%
[25]	Chest X-ray images	(300 +)	Xception +ResNet50V2 +CoroNet	5-fold cross-validation	93.412%
[26]	Chest X-ray images	(2893 +)	Inception-ResNet-v2	Train: Test 80%:20%	96.6%

[27]	Chest X-ray images	(6534 +)	efficient-net + SE-Resnext + efficient net adversarial	Train: Test: Validation 70% :20%:10%	95.92%
[28]	Chest X-ray images	(2232 +)	ResNet-50+SAM+BiLSTM	5-fold cross-validation	97.26%

2. PROJECT DESCRIPTIONS

2.1. Materials and methods

2.1.1. Dataset collection

CXRs images were used in this investigation. The database by [2][29] was selected for this study. The database consists of 15K CXR scans, including 3.5K covid-positive photos, 10K normal images, and 1345 pneumonia images as show in Table 2. The images in the database came from a variety of sources, such as a German medical school, SIRM, GitHub, Kaggle, and Twitter, as well as certain public repositories [29] [2], which physicians and academics had uploaded for study purposes. Every image in the database has 1024 x 1024 pixels and is saved in PNG format and Figure 1 shows some sample of images.

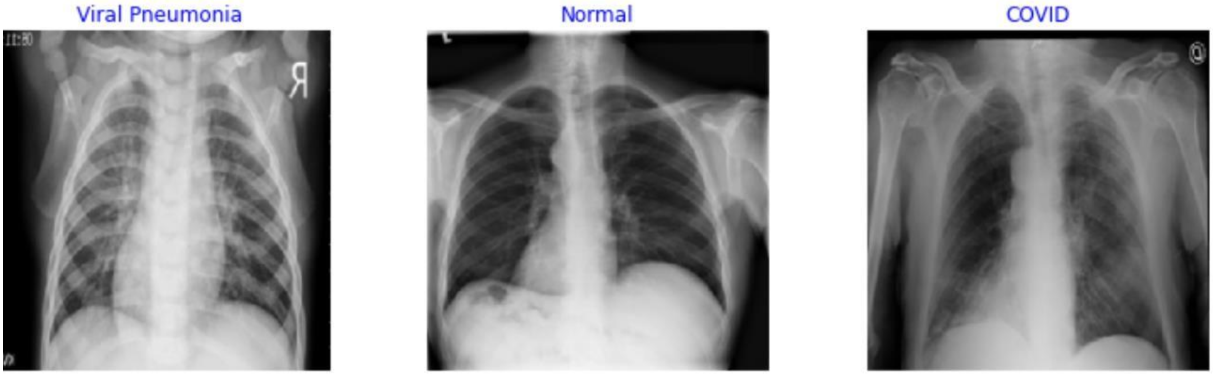


Figure 1. Sample X-ray images from the used dataset.

Table 2. List of all data sets for 3 classes (covid-19, normal, viral) and the numbers of datasets used for Train and Testing

Data source	Class label	samples	Testing	Modality
[29] [2]	Infected -covid	3616	240	X-Ray images
	health	10192	240	
	Infected Pneumonia	1335	240	

2.1.2. Data preprocessing

The original sizes of the images before being put into the model vary, ranging from 1024x1024 to a minimum of 540 x 540. These images are then downsized to 234 × 224 RGB images so that the processing layers can quickly distinguish the regular images. Furthermore, contrast enhancement and image normalization procedures are used to modify the pixel intensity values and reduce unwanted noise. Hidden information that resides within the low range of grey level images is revealed by increasing the image intensity value. For creating training and testing instance and to preprocess labeled data into the model Image Data Generator from keras was imported.

2.1.3. Data augmentation

Data augmentation, i.e., increasing the quantity of obtainable data without gathering new data by applying various operations to the available data, has proven to be effective in image classification [29]. The technique was used by the winners of the ImageNet classifier competition [30],[31], and it's widely employed by researchers to extend the training data, thereby avoiding overfitting [32]. We chose to apply data augmentation techniques in this study because of the limited number of images in the COVID-19 radiography database, particularly for the COVID-19 class, which had just 3616 samples. Each data that found in each instance will labeled with class name by Image Data Generator and all the preprocessing and augmentation scaling explained in Table 3.

Table 3. The names of each pre-processing and augmentation techniques with range values

Techniques	Range
Rescale Factor	1/255
Shear range	0 to 0.1 Rad counterclockwise
Zoom range	0.9 to 1.1
Channel shift range	150
RandomHorizontalFlip	True
RandomVerticalFlip	True
Height shift range	10%
Rotation range	-90 to 90
Train-test split ratio	80%:20%
Width shift range	10%

Normalize	Mean = 0.485, 0.456, 0.406,
	STd= 0.229, 0.224, 0.225

Shuffle	True

3. TECHNICAL SOFTWARE SPECIFICATION

3.1. Experimental and hyper parameters setup

The pre-trained (VGG16, Resnet50, and efficientNetB3) and custom design CNN models used in this study were tested on Kaggle since it offers a complete Keras and Torch library, as well as all DL and data analysis packages, all are important packers are pre-installed and we use it without compatibility issues. It also includes a fast GPU and TPU. For all three pretrained models, the images in the dataset were scaled to 224x224 pixels.

For the implementation, TensorFlow 2.8 with 2.7 Keras API and Adam optimizer with a learning rate of 1e-5 and a dropout probability of 0.5 was used. A categorical cross-entropy loss function was employed for adjusting model weights, minimize loss, and to evaluate model performance based on ground truth probabilities during training. And all the models were fine-tuned and tried with 80% train and 20% test data with 22 epochs for 32 batch sizes. The cross-entropy loss for n class is defined as in equation (1):

$$\mathcal{LCE} = -\sum_{i=1}^{m} y_i \log q_i \text{, for m classes.} \tag{1}$$

Where, y_i is the truth label and qi are the classifier (SoftMax) probability for the ith class.

Table 4 The Hyper parameters setup details for our three proposed model (VGG-16, ReNet-18 and efficient Net-B3, custom design CNN models)

List of hyperparameters	Setup
Dataset collection	Kaggle, previous publication, GitHub
Input	Data collected for 3 categories (covid-19, normal, viral pneumonia)
Input image Re size	224x224x3
Environment	Kaggle with required libraries
Directories	Divide the data into two parts: training (80%), testing (20%), and create subfolders for each folder establishing three classes (covid-19, normal, viral pneumonia)
Libraries used	NumPy, matplotlib, sklearn, scikitplot, Keras, torch
Batch size	32

Epoch	20
Optimizer	Adam
Learning rate (LR)	0.003
Loss	Categorical cross entropy
dropout probability	0.5
activation function	Relu
Output classifier	softmax
Model	Sequential
Training and testing	Created the proposed model using 3 pre-trained networks (efficient net-B3, ResNet-50, VGG16) and custom design CNN model
Model validation	Identified the best performing model from 3 pre-trained model as mentioned above with training and testing data and test the performance of custom design CNN model. Generate performance metrics 1. Confusion metric 2. ROC curve

4. DESIGN APPROACH AND DETAILS

4.1. Transfer learning approach

Due to the difficulties of gathering and labelling data, which must be done by experts, there is a paucity of training data necessary for machine learning models in COVID detection. To circumvent the shortage of training datasets for the image classification tasks, transfer learning (TL) using pre-trained models from the ImageNet dataset was employed. TL aids in image recognition rate by providing a large dataset for learning features. Many studies have shown that TL is successful and efficient in medical image categorization tasks. Furthermore, training a CNN model on a limited dataset will not provide significant results.

Traditional ML algorithms decouple learning from modelling and data analysis. As a result, the previously trained model's knowledge is not retained nor communicated across models. However, with deep transfer learning, knowledge such as weights and features of the pre-trained model can be used to train a new model. Deep learning is more accurate and quicker than traditional ML techniques, and it requires less training data sets. For deep learning-based medical image classification, which requires fine-tuning of the pre-trained framework, a transfer learning approach is often utilized. Input data contains classes that were previously unknown to the network architecture. After making certain network adjustments, we may employ pre-trained models for multiple tasks.

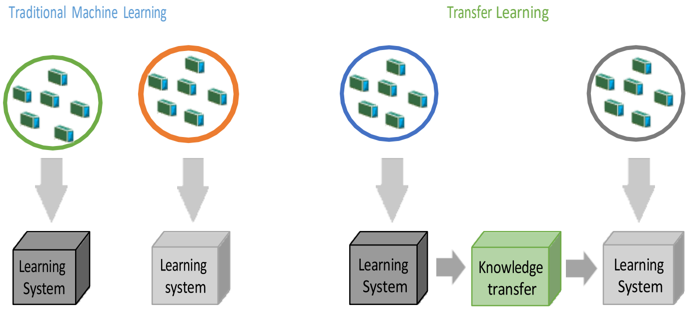


Figure 2. The concept of knowledge transfer in Traditional machine learning (ML) and in transfer learning (TL)

There are two ways to exploit the capabilities of pre-trained CNNs. As part of the first approach, features are extracted using transfer learning; these techniques rely on a pre-trained model to extract the features. As a result, the weights of a pre-trained model cannot be used for new tasks since the extracted features must be fed into a new classifier that also needs to be trained from scratch. Using the convolutional basis, a new classifier will be trained on the output of the previously trained network. As a result, this technique requires a large amount of data, and the numbers of covid CXR images available is insufficient to train it from scratch. The second approach involves fine-tuning the network for new tasks. As a result, the weight of the pre-trained model is the first value of the new task and is updated as the network is being trained.

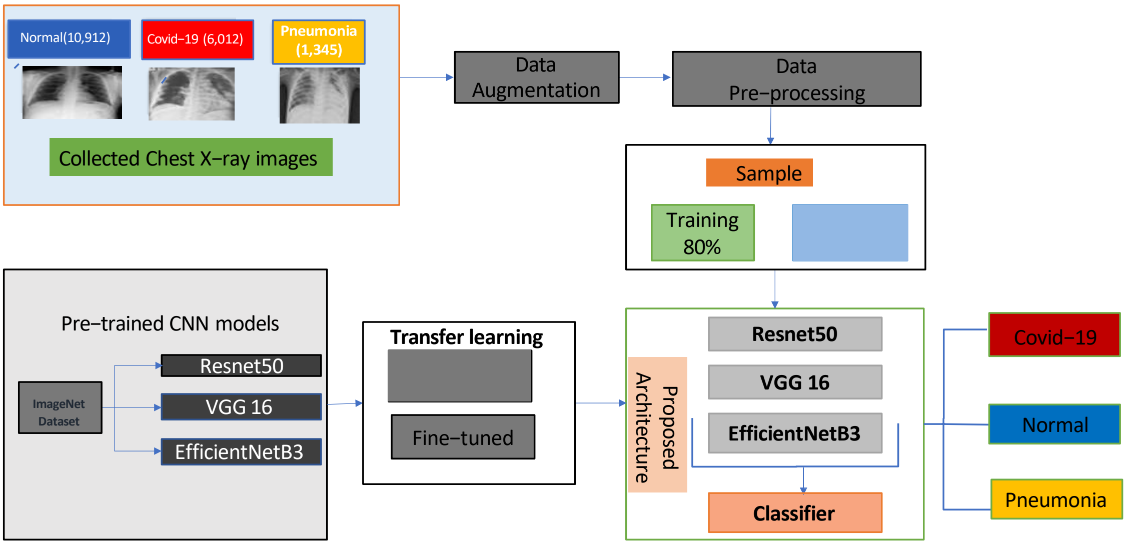


Figure 3. Proposed model using transfer learning

In this study, due to the scarcity of covid-19 CXR images, we only fine-tune the last layer of three well-known pre-trained models VGG16 [30], ResNet50 [31], and Efficient Net [32] and use the pre-trained model as the feature extractor and train them on the prepared data set to classify CXR images into three classes: covid-19, normal, and viral pneumonia with reasonable accuracy. The next section provides a short explanation of the architecture of these models and how we utilized for covid-19 identification.

4.1.1. VGG-16 detection of Covid-19

The VGG-16 architecture was introduced by [30]. As shown in Figure 4, it is a 16-layer network, the first 13 of which are Conv layers, two of which are FC layers, and the last of which is a SoftMax classifier. As shown in Table 5, the number of channels began with 64 in the first layer and increased by two factors as it passed through each max-pooling layer until it reached 512. Using VGG-16 has 2 advantages: firstly, its power to extract features at a low level by utilizing a smaller kernel size, which is ideal for CXR images with fewer parameters than its counterpart VGG-19 model. Second, it has excellent feature extraction capability for covid-19 from CXR image classification [33]. As we have a limited number of covid-19 CXR images, we use a fine-tuning strategy, which is one of the transfer learning techniques in this case. As in the table Table 6, we employ ImageNet's pre-trained weight of VGG-16 and fine-tune its last 4 layers; this layer is the flattening layer, whose role is to convert the data into a two-dimensional array to create a single long feature vector fed into an Fc layer. Once we have the transformed vector, we then feed it into a FC dense layer with a depth size of 64 units, and ReLU is used as an activation function after each Fc layer. Then, for regularization, we apply dropout with a probability of 0.35. Finally, for the output layer, we used the SoftMax classifier as we have multiple classes. For the model optimization, the Adam optimizer is used with an LR of 0.0003. Figure 5 depicts the proposed model, and the model summary is tabulated in Table 6 after fine-tuning.

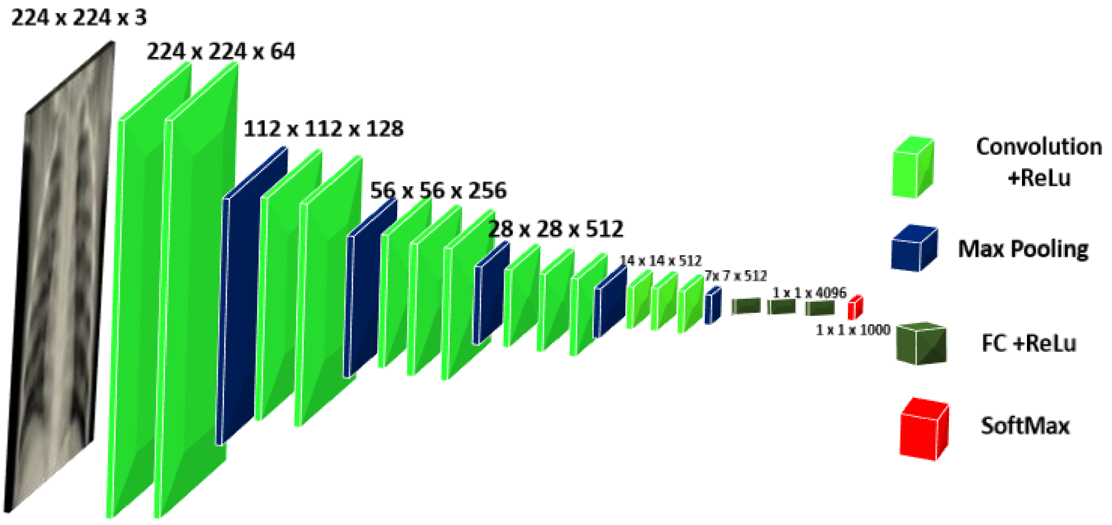


Figure 4. VGG-16 base line model architecture based on the original paper

Table 5. The List of parameters in the original VGG-16 model

Layer Name	Input size	Kernel size	Stride	Output size
conv1-64	224× 224 × 3	3x3	1	224 × 224 × 64
conv1-64	224× 224 × 64	3x3	1	224 × 224 × 64
Max-pool	224× 224 × 64	2x2	2	112 × 112 × 64

conv2-128	112×112 × 64	3x3	1	112 × 112 × 128
conv2-128	112× 112 × 128	3x3	1	112 × 112 × 128
Max-pool	112× 112 × 128	2x2	2	56 × 56 × 128
conv3-256	56× 56 × 128	3x3	1	56 × 56 × 256
conv3-256	56× 56 × 256	3x3	1	56 × 56 × 256
conv3-256	56× 56 × 256	3x3	1	56 × 56 × 256
Max-pool	56× 56 × 256	2x2	2	28 × 28 × 256
conv4-512	28× 28 × 256	3x3	1	28 × 28 × 512
conv4-512	28× 28 × 512	3x3	1	28 × 28 × 512
conv4-512	28× 28 × 512	3x3	1	28 × 28 × 512
Max-pool	28× 28 × 512	2x2	2	14 × 14 × 512
conv5-512	14× 14 × 512	3x3	1	14 × 14 × 512
conv5-512	14× 14 × 512	3x3	1	14 × 14 × 512
conv5-512	14× 14 × 512	3x3	1	14 × 14 × 512
Max-pool	14× 14 × 512	2x2	2	7 × 7× 512
fc	1× 1 × 25088	1x1	-	1 × 1 × 4096
fc	1× 1 × 4096	1x1	-	1 × 1 × 4096
fc	1× 1 × 4096	1x1	-	1 × 1 × 1000

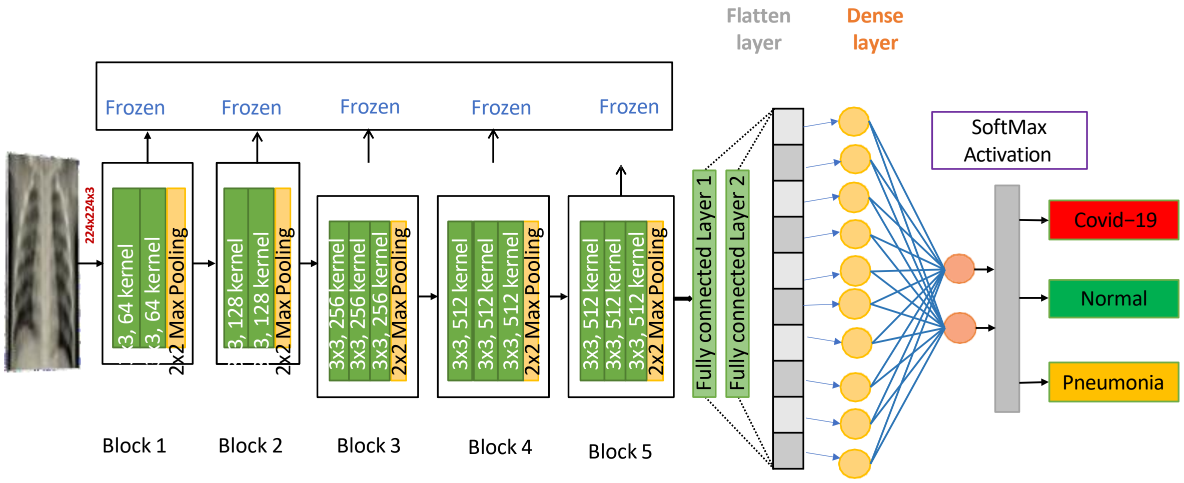


Figure 5. The proposed VGG-16 CNN architecture for multiple categories

Table 6. Architecture Details of our proposed VGG-16 model's

Layer type	Output shape	Parma
input_1 (InputLayer)	224 × 224 × 3	0
block1_conv1 (Conv2D)	224 × 224 × 64	1792
block1_conv2 (Conv2D)	224 × 224 × 64	36928
block1_pool (MaxPooling2D)	112 × 112 × 64	0
block2_conv1 (Conv2D)	112 × 112 × 128	73856
block2_conv2 (Conv2D)	112 × 112 × 128	147584
block2_pool (MaxPooling2D)	56 × 56 × 128	0
block3_conv1 (Conv2D)	56 × 56 × 256	295168
block3_conv2 (Conv2D)	56 × 56 × 256	590080
block3_conv3 (Conv2D)	56 × 56 × 256	590080
block3_pool (MaxPooling2D)	28 × 28 × 256	0
block4_conv1 (Conv2D)	28 × 28 × 512	1180160
block4_conv2 (Conv2D)	28 × 28 × 512	2359808
block4_conv3 (Conv2D)	28 × 28 × 512	2359808
block4_pool (MaxPooling2D)	14 × 14 × 512	0
block5_conv1 (Conv2D)	14 × 14 × 512	2359808
block5_conv2 (Conv2D)	14 × 14 × 512	2359808
block5_conv3 (Conv2D)	14 × 14 × 512	2359808
block5_pool (MaxPooling2D)	7 × 7× 512	0
flatten (Flatten)	25088	0
dense (Dense)	64	1605696
dropout (Dropout)	64	0
dense_1 (Dense)	3	195
Total params: 16,320,579 Trainable params: 1,605,891 Non-trainable params: 14,714,688		

4.1.2. ResNet-50 detection of Covid-19

ResNet50 was proposed by [34] and the main breakthrough with ResNet18 was the ability to train very deep neural networks with 150+ layers, as well as particular skip connections to overcome the problem of Vanishing Gradient during backward propagation in very deep neural networks. Skip connections allow information to flow while also assuring that top levels work equally well as lower layers by skipping one or two layers.

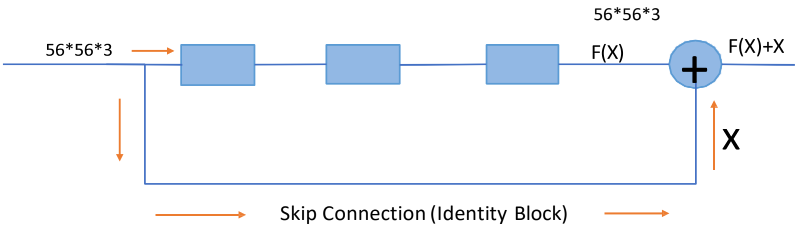


Figure 6. Identity block (input size=output size), When the input of the network and the output of the network is the same then the Identity block employed.

ResNet50 employs two blocks depending on whether the input and output dimensions are the same or different. The identity block (IB) is used when the input image size equals the output image dimension, while the convolutional block (CB) is used when the input and output dimensions do not match. The input image for Resnet 50 is 224x224 pixels in size with RGB channels then this input image is subjected to a 7x7 convolution with a stride of 2, producing a 112 x 112-pixel feature map. The feature map is then max pooled to a size of 56 by 56 pixels. The max-pooled feature map is subjected to a convolution block sequence. Only three layers (out of 50) feature stride 2. When stride convolution is used on a feature map to minimize image size, it results in a dimension difference between the input and output image sizes, hence a convolution block is used at this period. Using 1x1 convolution image down sampling

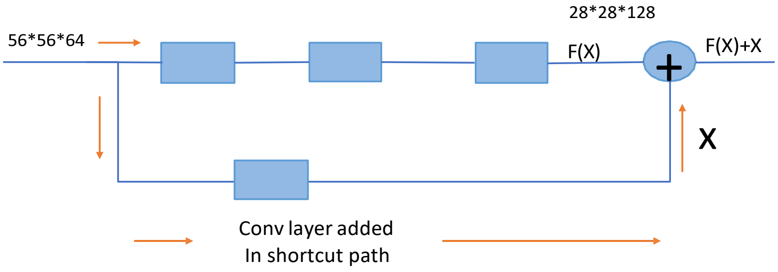


Figure 7. convolutional block (input! = output), during this time convolutional block utilized, the block has extra 2D convolution block followed by batch normalization and Relu, this helps to down sample the input image size to match with the output image size.

methods, the input and output image sizes will be equalized. After going through the last convolution block, the feature map is 7x7 pixels in size. The resulting feature map is then average-pooled to form a 512- or 2048-dimensional vector, which implies that the channel average reduces each 7x7 channel to 1x1.

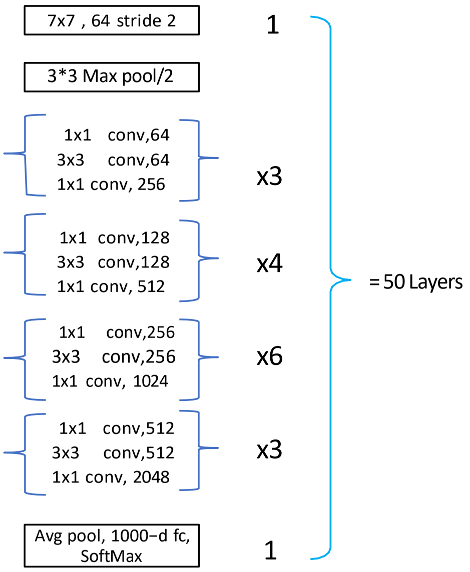


Figure 8. The architectures of the baseline model (Resnet 50) as mentioned on

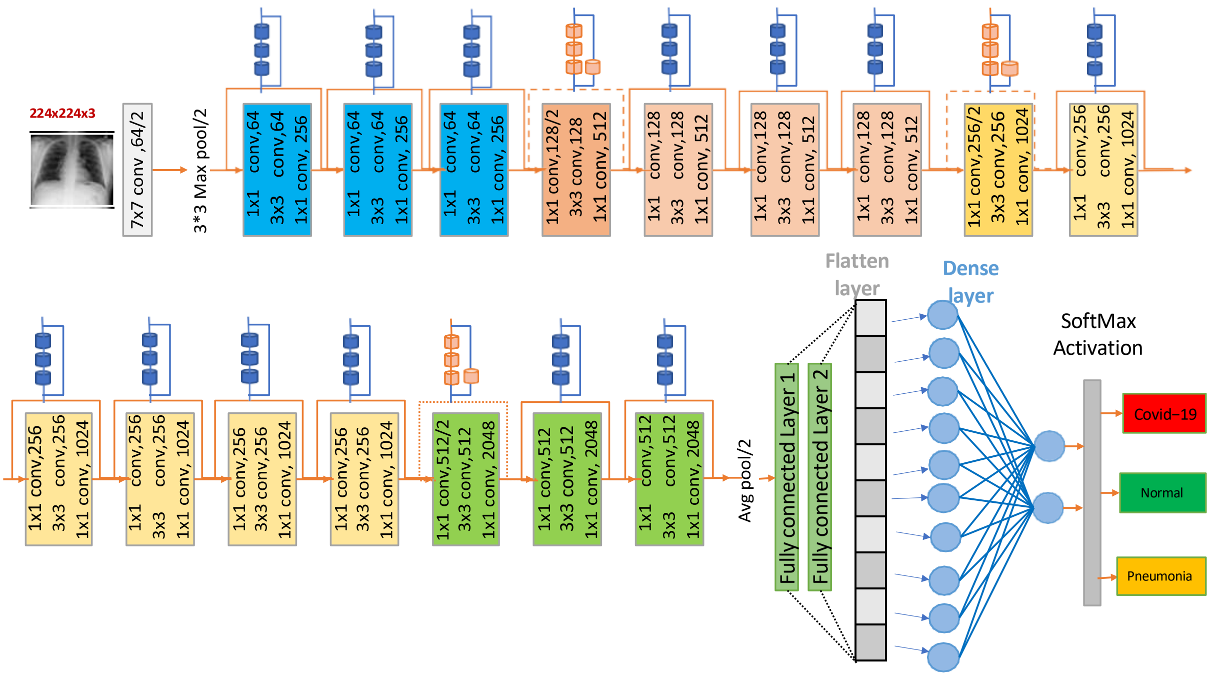
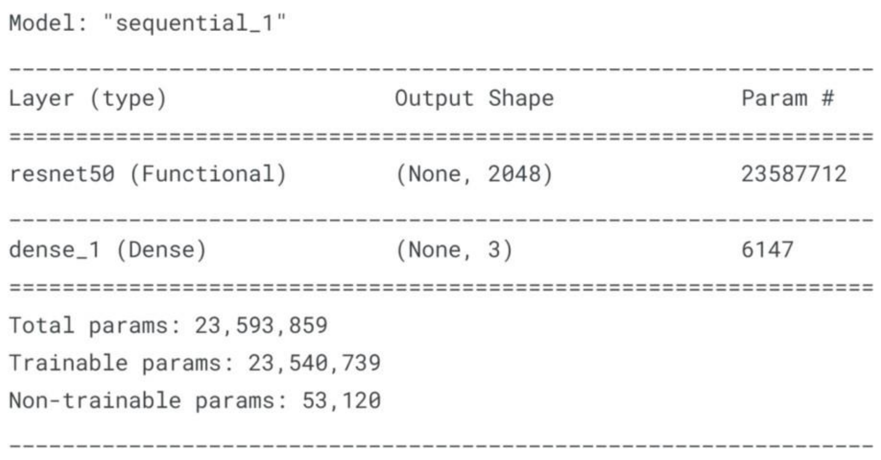


Figure 9. A proposed ResNet-50 architecture for multiple categories.



```
Model: "sequential_1"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Functional)        (None, 2048)              23587712

_________________________________________________________________
dense_1 (Dense)              (None, 3)                 6147
=================================================================
Total params: 23,593,859
Trainable params: 23,540,739
Non-trainable params: 53,120

_________________________________________________________________
```

Figure 10. A detailed of the proposed ResNet50 model summary after fine turning

4.1.3. Efficient Net for detecting Covid-19

The Efficient Net is a set of seven CNNs, numbered b1 through b7, with b0 acting as the benchmark network [35]. By demanding fewer parameters and FLOPS, the Efficient Net surpasses previous CNNs in terms of accuracy and efficiency [32] [35]. Efficient Net is 8.4 times lesser of parameters (model complexity) and 6.1 times faster than the best available state-of-the-art models [32] based on the following factors. Compound scaling is the systematic and reasonable scaling of three variables: depth, breadth, and resolution [32] [36]. Second, mobile inverted bottleneck Conv blocks (MBconv) with squeeze and excite minimization were introduced [32] [35]. Finally, pointwise and depth-wise convolution are employed. EfficientNet works in three different ways: Figure [] shows a full description of the proposed

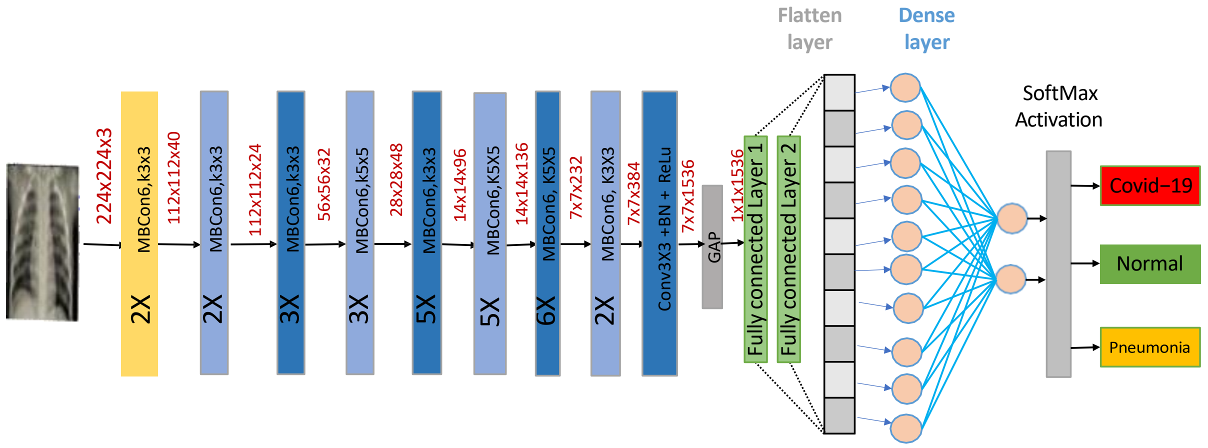


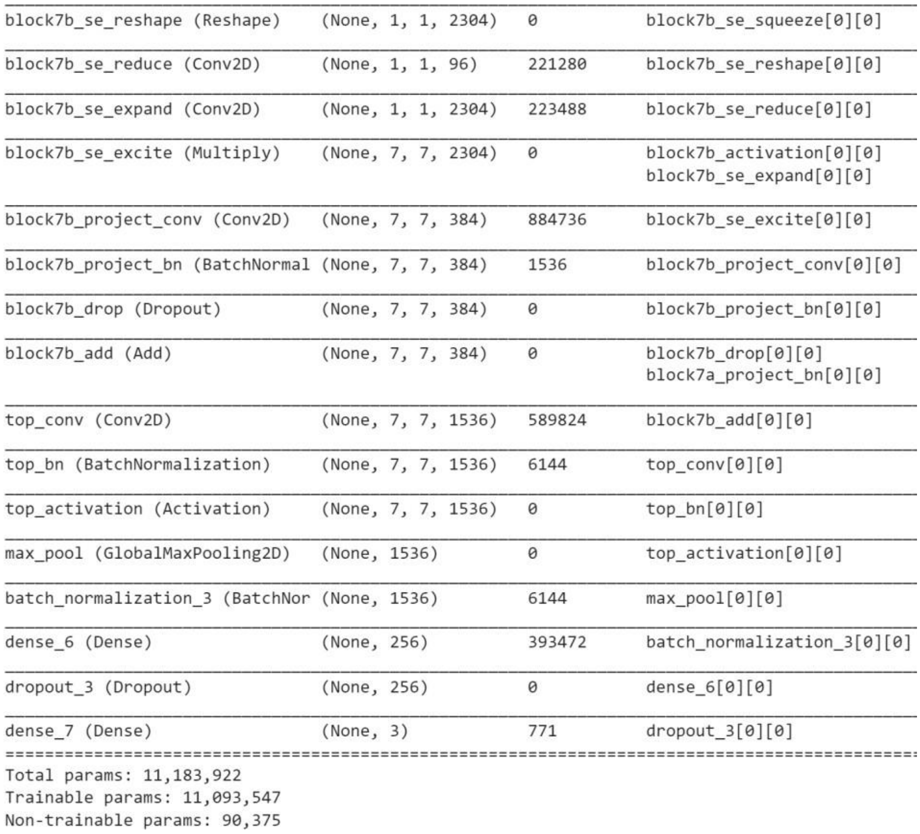
Figure 11. Proposed efficient net model after fine tuning

efficient net model after fine tuning.

a. Both depth-wise and point-wise convolution are used in an Efficient Net: Regular CNN is performed on all input channels, whereas depth-wise convolution is performed on a single channel at a time [36]. The depth-wise convolution channel output is projected onto a new channel space using pointwise convolution, which is a 1 x 1 convolution followed by a depth-wise convolution to transform the channels of the input features.

b. Inverted residual block: The MBConv layer, which is an inverted residual block (IRB) initially used in MobileNetV2 [37], serves as the foundation for EfficientNet. ResNet blocks are made up of two layers: one that squeezes the channels and one that extends them [32] [37] [34]. It connects skip connections to rich channel layers in this manner.

c. The bottleneck layer: To reduce information loss from Relu, we utilize linear activation functions for specific layers (the bottleneck layer because the number of channels is squeezed at some of these places in the network), which is a concept that comes from mobile net V2 [32] . According to the author [37], Relu does not operate as well in the last layer as it does in the first.



```
________________________________________________________________________________________________
block7b_se_reshape (Reshape)    (None, 1, 1, 2304)   0           block7b_se_squeeze[0][0]
________________________________________________________________________________________________
block7b_se_reduce (Conv2D)      (None, 1, 1, 96)     221280      block7b_se_reshape[0][0]
________________________________________________________________________________________________
block7b_se_expand (Conv2D)      (None, 1, 1, 2304)   223488      block7b_se_reduce[0][0]
________________________________________________________________________________________________
block7b_se_excite (Multiply)    (None, 7, 7, 2304)   0           block7b_activation[0][0]
                                                                 block7b_se_expand[0][0]
________________________________________________________________________________________________
block7b_project_conv (Conv2D)   (None, 7, 7, 384)    884736      block7b_se_excite[0][0]
________________________________________________________________________________________________
block7b_project_bn (BatchNormal (None, 7, 7, 384)    1536        block7b_project_conv[0][0]
________________________________________________________________________________________________
block7b_drop (Dropout)          (None, 7, 7, 384)    0           block7b_project_bn[0][0]
________________________________________________________________________________________________
block7b_add (Add)               (None, 7, 7, 384)    0           block7b_drop[0][0]
                                                                 block7a_project_bn[0][0]
________________________________________________________________________________________________
top_conv (Conv2D)               (None, 7, 7, 1536)   589824      block7b_add[0][0]
________________________________________________________________________________________________
top_bn (BatchNormalization)     (None, 7, 7, 1536)   6144        top_conv[0][0]
________________________________________________________________________________________________
top_activation (Activation)     (None, 7, 7, 1536)   0           top_bn[0][0]
________________________________________________________________________________________________
max_pool (GlobalMaxPooling2D)   (None, 1536)         0           top_activation[0][0]
________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 1536)         6144        max_pool[0][0]
________________________________________________________________________________________________
dense_6 (Dense)                 (None, 256)          393472      batch_normalization_3[0][0]
________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 256)          0           dense_6[0][0]
________________________________________________________________________________________________
dense_7 (Dense)                 (None, 3)            771         dropout_3[0][0]
================================================================================================
Total params: 11,183,922
Trainable params: 11,093,547
Non-trainable params: 90,375
```

Figure 12. Detailed of the proposed efficient Net -B3 model summary after fine turning

Table 7. Illustrate the depth, resolution, and width scaling of Efficient Net-B3

Block No. (j)	Leyer (Gi ())	Resolution (Ui x Wi	No of layers (Li)
1	Conv 3x3	300x300	1
2	MBconv1,	150x150	2
3	MBconv6,	75x75	3
4	MBconv6,	38x38	3
5	MBconv6,	19x19	5
6	MBconv6,	10x10	5
7	MBconv6,	10x10	6
8	MBconv6,	10x10	2
9	Conv 1x1	10x10	1
10	Global pooling	10x10	1
11	Dense Layer	10x10	1

4.1.3.1. Compound scaling

Controlling the width and scaling up helps the network add more feature mappings to each layer, whilst scaling the depth helps the network add more layers and increase the resolution of input images by scaling up the resolution [32] [38] [36]. The concept of compound scaling is related to the depth, breadth, and resolution formulated in Equation (2),(3) ,(4).

$$\mathrm{depth(d)} = \alpha^{\theta} \tag{2}$$

$$\mathrm{width\ (w)} = \beta^{\vartheta} \tag{3}$$

$$\alpha * \beta^2 * \gamma^2 = 2, \tag{4}$$

Whele, $\alpha \geq 1$, $\beta \geq 1$, $\gamma \geq 1$.

θ is a user-specified coefficient that controls how many resources are accessible whereas α, is a parameter to control the depth (d), β is a parameter to control the width (ω) and γ is a parameter to control the resolution (r). for the effinentNet-B0(The base model), the value of θ, α, β and γ is 1, 1.2, 1.1 and 1.15 respectively [32]. By Fixing α, β and γ value and changing the value of θ based on resources availability will give more complex EfficientNet models [36].

4.2. Using custom CNN architecture (CNN) approach

In deep learning tasks, CNN performs excellently [39] and often used to deal with image data classification algorithm like in imgeNet, and its promising result in area of traffic signal processing, captioning of pictures, recognition of voice , translation of different languages , and cancer cell detection have attracted attention in the DL community. CNN builds feature maps by using input image convolution with kernel values. This feature map is connected to preceding layers by kernel weights, which are updated during training via a feed backward process. A pooling layer is used to simplify the complexity of the following levels. CNN have different layers which are explained below.

4.2.1. Convolution layer (CL)

It is the first and most significant layer in the CNN layer, and it uses a kernel to extract features from an input image and build a feature map.

4.2.2. Kernel

In CNN, a kernel is used to extract high-level visual features such as edges. The kernel is a matrix that traverses the input data, performs the dot product with the input matrix data sub-region, and returns the matrix of dot products as the output. The stride value instructs the kernel to shuffle the input data. When the stride parameter is set to 2, for example, the kernel advances the input matrix by two columns of pixels.

Table 8. 11 Layer Custom design CNN model

Batch_normalization_1_input: InputLayer ↓	Input:	[(None, 224,224,1)]
	Output:	[(None, 224,224,1)]

Batch_normalization_1: BatchNormalization ↓	Input:	[(None, 224,224,1)]
	Output:	[(None, 224,224,1)]
Conv2d_3: Conv2D	Input:	[(None, 224,224,1)]
	Output:	[(None, 224,224,64)]
↓		
Max_Pooling2d_3: MaxPooling2D	Input:	[(None, 224,224,64)]
	Output:	[(None, 112,112,64)]
↓		
Conv2d_4: Conv2D	Input:	[(None, 112,112,64)]
	Output:	[(None, 112,112,64)]
↓		
Max_pooling2d_4: MaxPooling2D ↓	Input:	[(None, 112,112, 64)]
	Output:	[(None, 56,56,64)]
Droupout_3: Dropout ↓	Input:	[(None, 56,56,64)]
	Output:	[(None, 56,56,64)]
Conv2d_5: Conv2D ↓	Input:	[(None, 56,56,64)]
	Output:	[(None, 54,54,32)]
Max_pooling2d_5: MaxPooling2D ↓	Input:	[(None, 54,54,32)]
	Output:	[(None, 27,27,32)]
Dropout_4: Dropout ↓	Input:	[(None, 27,27,32)]
	Output:	[(None, 27,27,32)]

Flatten_1: Flatten ↓	Input:	[(None, 27,27,32)]
	Output:	[(None, 23328)]
Dense_2: Dense ↓	Input:	[(None, 23328)]
	Output:	[(None,128)]
Dropout_5: Dropout	Input:	[(None, 128)]
	Output:	[(None,128)]
↓		
Dense_3: Dense	Input:	[(None, 128)]
	Output:	[(None, 3)]

4.2.3. Activation function

It is used to determine NN's output and maps the resultant values between 0 and 1. There are many activation functions, including sigmoid, tanH, and ReLU. Relu is by far the most extensively utilized of these since it turns negative numbers to zero.

$$\mathrm{Fx} = \max(0, \mathrm{x}) \tag{5}$$

4.2.4. Pooling Layer

Pooling layers are used in CNNs to minimize the spatial size of feature maps. Max-pooling and Avg-pooling are two pooling layers that have been suggested. Max-pooling performed well and is widely used for dimensionality reduction. and the most common size is 2x2 max pooling. The maximum value of the matrix is chosen using max pooling. In contrast, avg- pooling delivers the average of all the values.

$$\mathrm{pool}_{i,j} = \underset{\mathrm{p}}{\mathrm{Max}}\, \mathrm{f}'(\mathrm{x})_{\mathrm{i+p,j+p}} \tag{6}$$

4.2.5. Flatten layer

In our model, we utilized a flattening layer after the pooling layer to reduce the pooled feature map to a single column matrix, which was then fed into the NN to be processed.

4.2.6. FC Layer

Fc layer is highly used parameter [39]. All the node in an FC layer is directly linked to every node in the layers above and below it. It is a feed forward neural network that uses the Softmax activation function as a classifier after receiving input from the preceding pooling layer.

$$\text{Softmax}\ (x_i) = \frac{\exp\ (x_i)}{\sum_j \exp\ (x_j)} \qquad (7)$$

Table 9. 11 Layer Custom design CNN model summary

Layer (type)	Output Shape	Param #
batch_normalization_6 (BN)	224, 224, 1	4
conv2d_18 (Conv2D)	224, 224, 64	640
max_pooling2d_18 (MaxPooling)	112, 112, 64	0
conv2d_19 (Conv2D)	112, 112, 64	36928
max_pooling2d_19 (MaxPooling)	56, 56, 64	0
dropout_18 (Dropout)	56, 56, 64	0
conv2d_20 (Conv2D)	54, 54, 32	18464
max_pooling2d_20 (MaxPooling)	27, 27, 32	0
dropout_19 (Dropout)	27, 27, 32	0
flatten_6 (Flatten)	23328	0
dense_12 (Dense)	128	2986112
dropout_20 (Dropout)	128	0
dense_13 (Dense)	3	387
Total params: 3,042,535 Trainable params: 3,042,533 Non-trainable params: 2		

5. RESULTS AND DISCUSSIONS

5.1. Performance evaluation Metrics

This study in order to evaluate the performance of the transfer learning method and our proposed method confusion matrix was used and it composed of the following elements:

a. correct identification of covid-19 cases (Truly Positives, TP),

b. wrong classification of covid-19 cases (Falsely Negatives, FN),

c. correct identification of healthy cases (Truly Negatives, TN),

d. wrong classification of healthy cases (Falsely Positives, FP).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Sensitivity/Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1} - \text{Score} = \frac{2 * \text{Precision x } 2 * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{12}$$

Where:

1. TP denotes accurately predicted covid-19 cases,

2. FP denotes health or Viral pneumonia cases categorized as Covid-19 cases,

3. TN denotes normal, or Viral pneumonia cases classified as non-novid-19 cases, and

4. FN denotes Covid-19 cases classified as health or viral pneumonia cases.

In addition to the above, performance was evaluated using ROC (Receiver Operating Characteristic) curves. These are graphs that show the variation of True Positive Rate (TPR), or Sensitivity as specified in Eq. (13), in relation to False Positive Rate (FPR), where FPR is defined as follows:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} * 100 \tag{13}$$

The ROC graph demonstrates the performance of the highlighted classification model over all categorization thresholds.

Table 10. Compression of the three models for the best epoch in terms of accuracy and loss throughout training and validation

Model	Best Epoch	Batch size	Training loss	Train accuracy	Validation loss	Validation accuracy
VGG-16	22	32	0.083	0.964	0.089	0.973
ResNet-50	22	32	0.005	0.99	0.157	0.969
EfficientNet-B3	22	32	1.413	0.997	1.440	0.966
Custom CNN	22	32	0.045	0.984	0.106	0.972

Table 11. The proposed model performance on classifying covid-19, non-covid-19 and Viral-pneumonia

Classification model	Class	Precision %	Recall %	F1-score %	AUC	Accuracy %
EfficientNet-B3	Covid-19	97	1.00	98	98	97
	Normal	99	93	96	99	
	Pneumonia	96	98	97	97	
VGG-16	Covid-19	97.5	1.00	98.7	99	97.9
	Normal	97.8	95.8	96.8	99	
	Pneumonia	98.3	97.9	98.1	97	
ResNet-50	Covid-19	98.7	96	97	99	97.6
	Normal	96	98	97	98	
	Pneumonia	98.7	98	98	98	
Custom CNN	Covid-19	96.2	96.2	94.8	97	95
	Normal	94.3	93.9	94.1	96	
	Pneumonia	94.6	95.0	94.8	96	

5.1.1. Training and validation phase in terms of Loss vs accuracy

For each fine-tuned model and custom CNN, the accuracy and loss values for their best epochs in the training and validation processes are given in Table 10 and displayed in Figure 13 - Figure 20. When comparing the number of epochs required by each model to attain the minimal validation loss, it is shown that the efficient NetB3, VGG16, and ResNet50 (using transfer learning) achieved minimum validation losses of 1.440, 0.089, and 0.157, respectively, after just 22 epochs, whereas the custom CNN model (without transfer learning) achieved a minimal validation loss of 1.069 for the same number of epochs. With 22 epochs, the efficient NetB3, VGG16, ResNet50, and custom CNN models reach 96.67, 97.3, 96.7, and 97.2 validation accuracy, respectively. Specifically, VGG-16 shows a minimum validation loss of 0.089. This suggests that these models are capable of quickly learning the differences between COVID-19, normal, and pneumonia. However, when train loss and accuracy are taken into account, efficient NetB3 (99.7%) and ResNet-50 (99%) have the best training accuracy, while VGG-16 has the lowest (96.4%). As a result of this data, it can be inferred that among the three models, the efficient NetB3 model has the best training and validation performance and gets considerable results from the custom CNN model (98.4) without transfer learning.

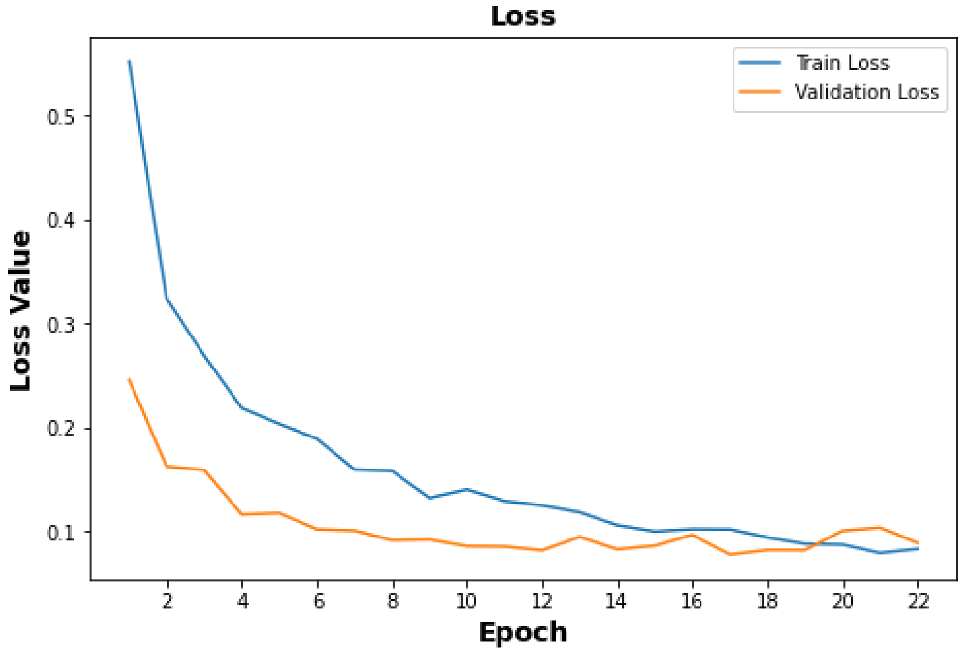


Figure 13. Loss plot of VGG16

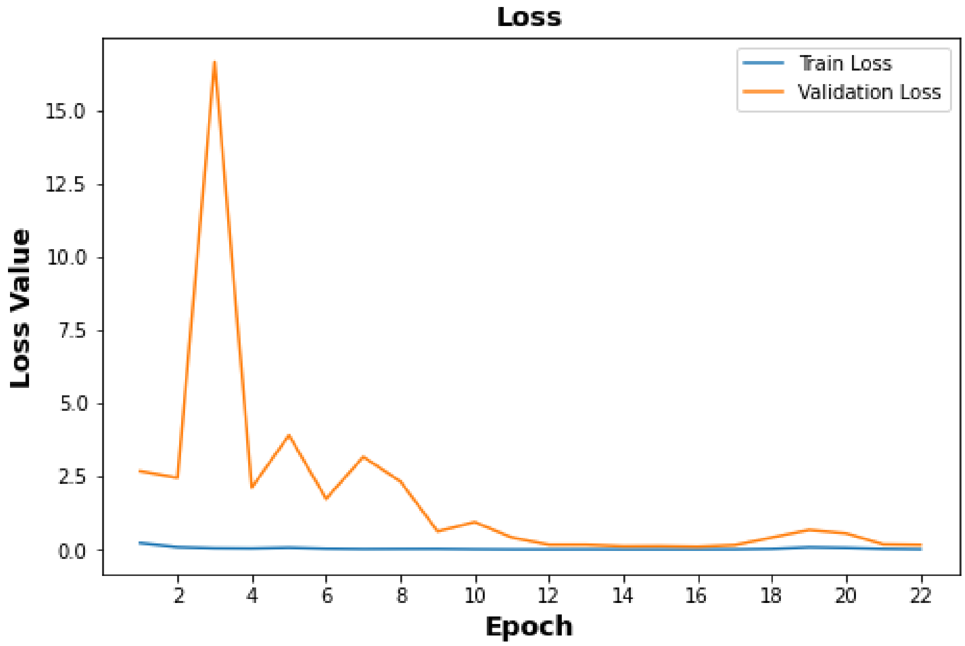


Figure 14. Loss plot of ResNet-50

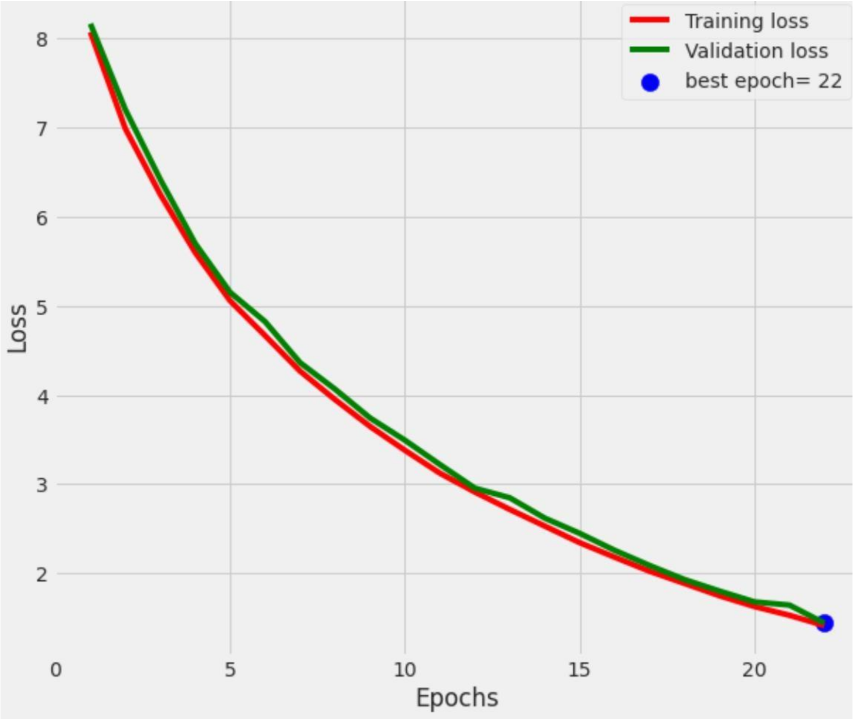


Figure 15. Loss plot of efficient NetB3

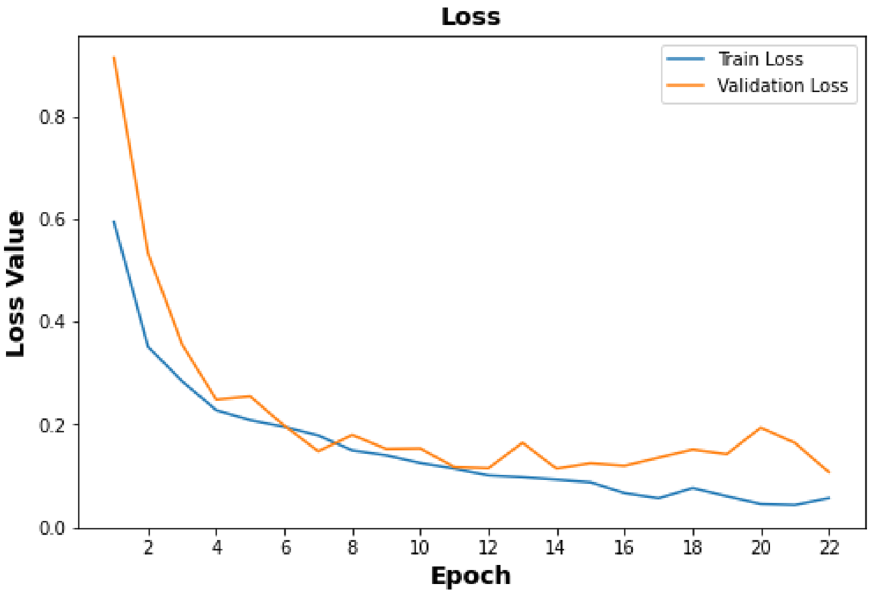


Figure 16. Loss plot of 11-layer custom design CNN model

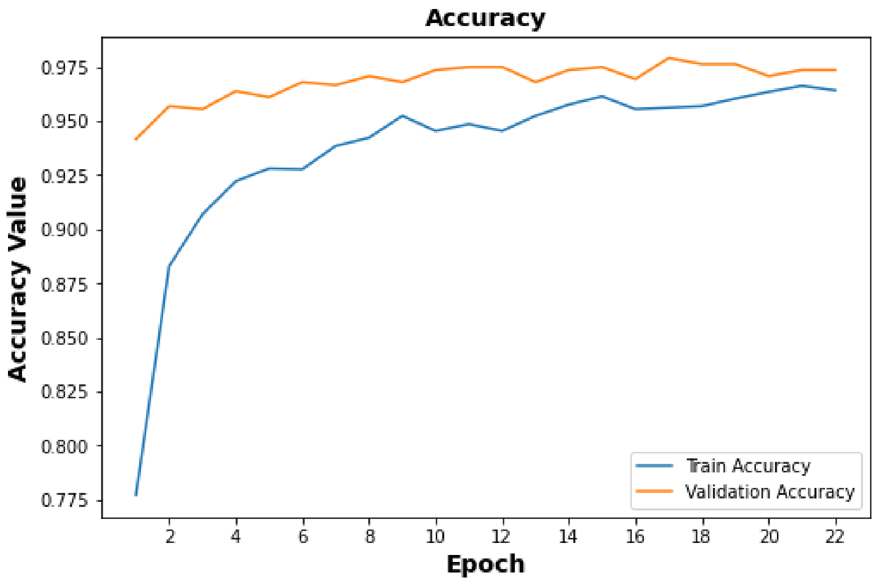


Figure 17. Accuracy plot of VGG16

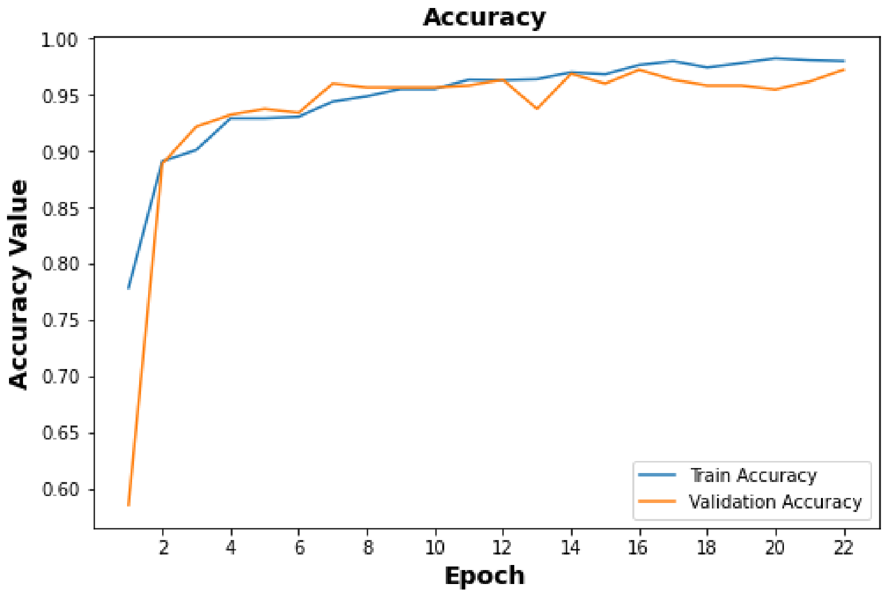


Figure 18. Accuracy plot of ResNet-50

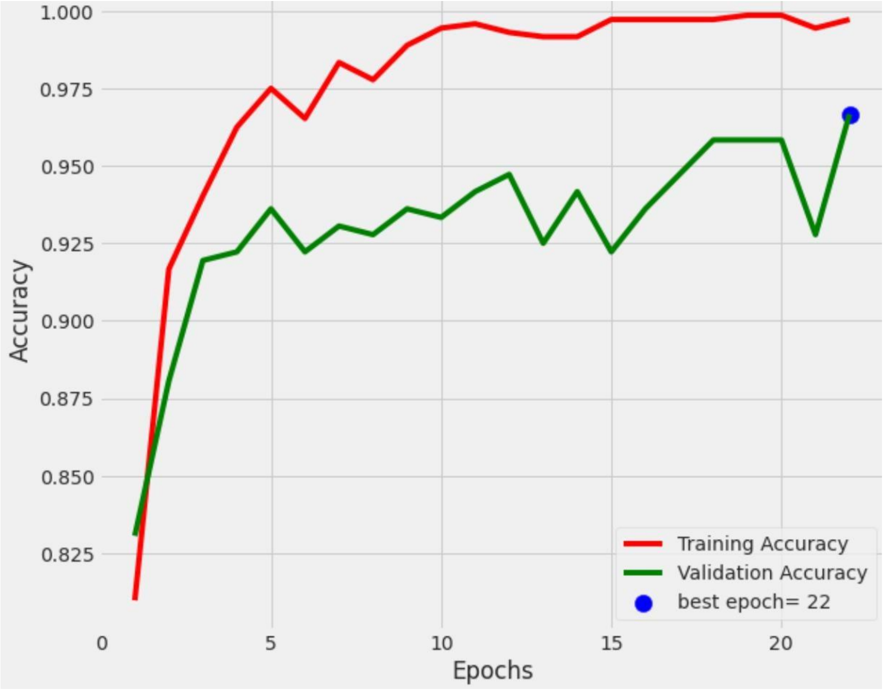


Figure 19. Accuracy plot of efficient NetB3

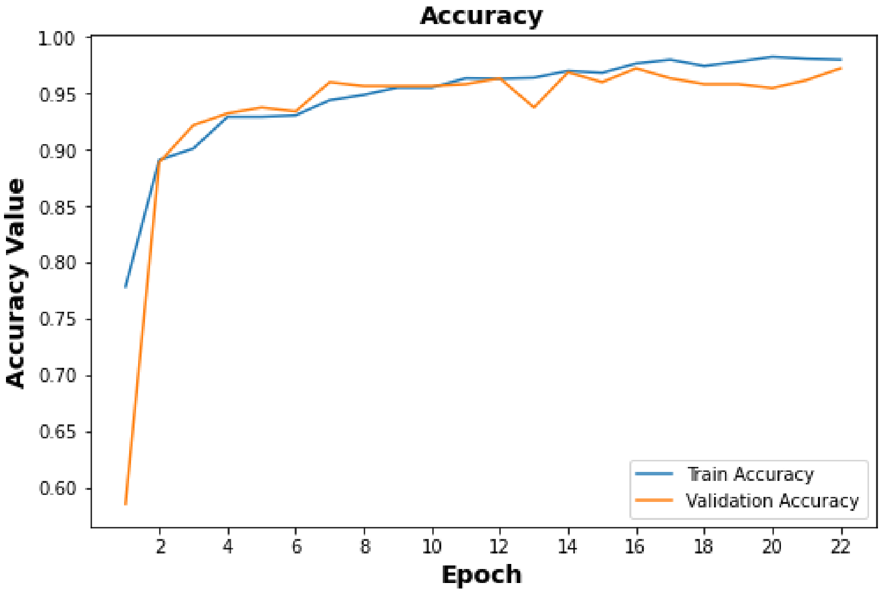


Figure 20. Accuracy plot of 11-layer custom design CNN model

5.1.2. Confusion matrix

The confusion matrix shows the number of images that the model correctly and mistakenly detected. The confusion matrix for all three models (efficient NetB3, VGG16, and ResNet50) and the custom CNN model was summarized in figures 17, 18, 19, and 20. During the validation, all models behaved well. Out of 240 validation images per class, the EfficientNetB3 model accurately categorized 239 COVID-19 instances (True Positive). The model misidentified 1 image as normal (false negative), but the model shows high false negative in the viral pneumonia class and has the lowest false positive and false negative in the COVID class. The VGG-16 model predicts 235 images as COVID (True Positive) and erroneously classifies 5 images as viral pneumonia. ResNet 50 wrongly predicted 3 images as pneumonia and 1 image as normal (false negative) out of 240 images. whereas the custom CNN model predicted 230 images as COVID and erroneously predicted 4 images as viral pneumonia and 6 images as normal. This suggests that using transfer learning will help to get high accuracy in the confusion matrix shown in Figure 21, Figure 22, Figure 23 and Figure 24. When comparing the three models based on TP, TN, FP, and FN from the pretrained models, VGG-16 had the fewest errors (15/240) while predicting whether the image was COVID-19, normal, or pneumonia.

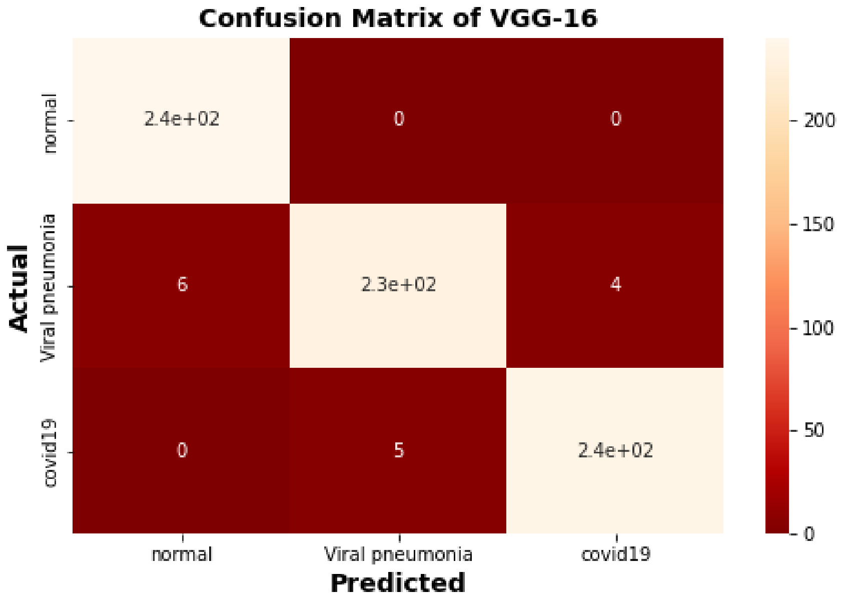


Figure 21. For multiple classifications, the proposed VGG16 model's confusion matrix

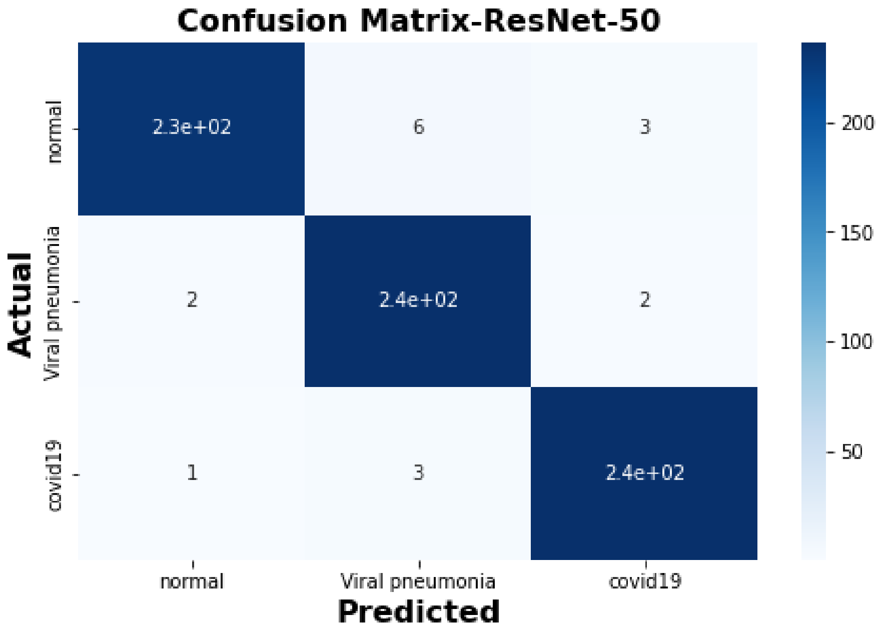


Figure 22. The proposed ResNet-50 models' confusion matrix for multiple classifications

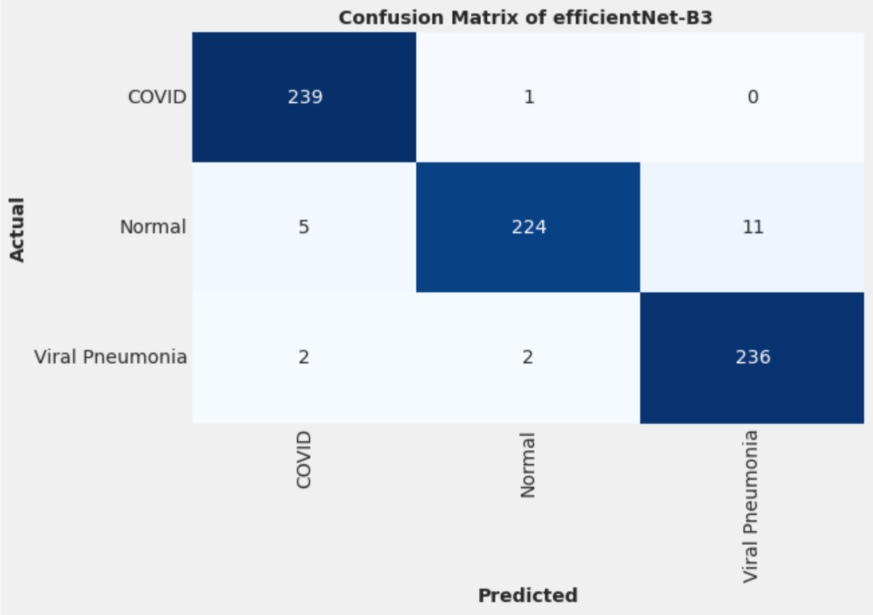


Figure 23. The proposed EfficientNetB3 models' confusion matrix for multiple classifications

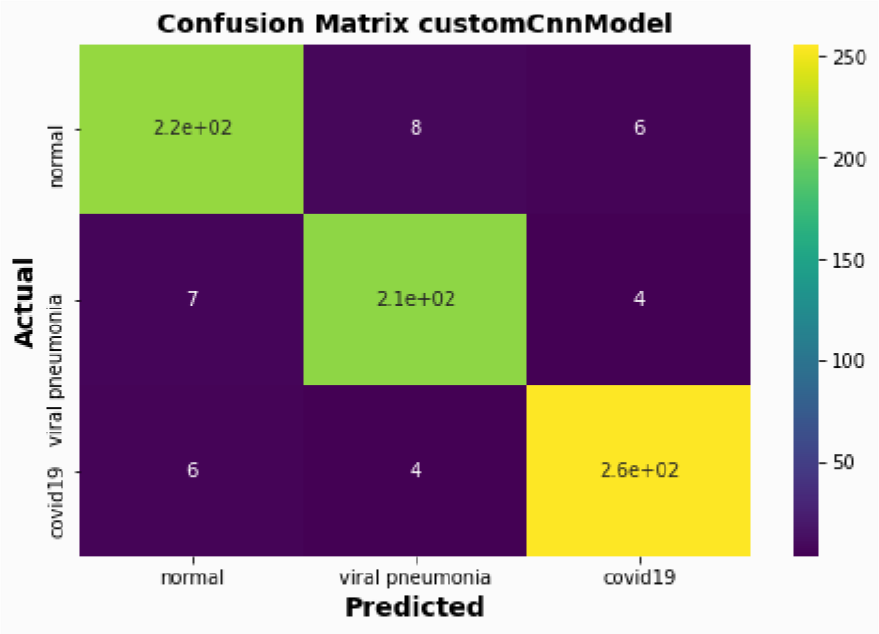


Figure 24. The proposed custom models' confusion matrix for multiple classifications

5.1.3. The receiver operating characteristic (ROC) and Area Under Curve (AUC)

Because recall (sensitivity) and specificity rates fluctuate when cut-off values are altered, it's difficult to compare different models just on their recall (sensitivity) and specificity rates. We must examine the comparison for all possible threshold values to determine how these models compare overall. One way of evaluating performance is to use the receiver operating characteristic (ROC) curve, which plots the true positive rate (TPR) as a function of the false positive rate (FPR). The Receiver Operator Characteristic (ROC) curve is a classification assessment metric that is often used. It's made by graphing TPR against FPR at different thresholds. The Area Under Curve (AUC) is a metric that measures a classifier's ability to distinguish between classes and ranges from 0 to 1. The closer the AUC value to 1, the higher the classifier's performance. Figures 23, 24, and 25 show the ROC curves for the three models with AUC values. ResNet-50 and VGG-16 has a higher AUC than the other models, ResNet-50 scored 0.98 both normal and pneumonia classes and .99 for covid class, whereas VGG-16 scored maximum of 0.99 AUC for both COVID and normal and 0.97 for its pneumonia classes. The efficient B3 model achieved 0.99,0.98,0.97 for its normal, pneumonia, covid class respectively. It's worth noting that the AUC may not be a good predictor of model performance for severely unbalanced test sets (as it will be very high).

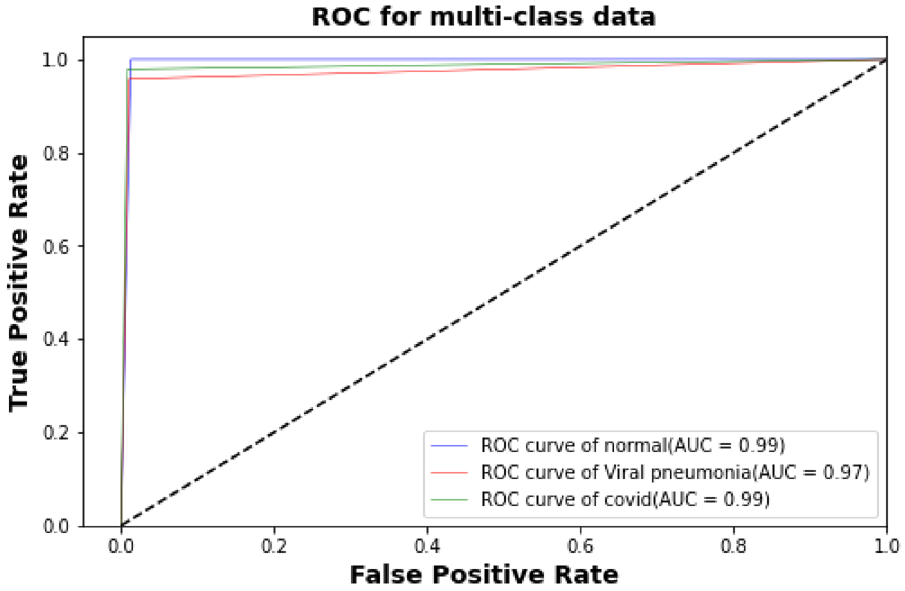


Figure 25.True positive rate (TPR) vs False positive rate (FPR) of VGG16 model for multiple classifications

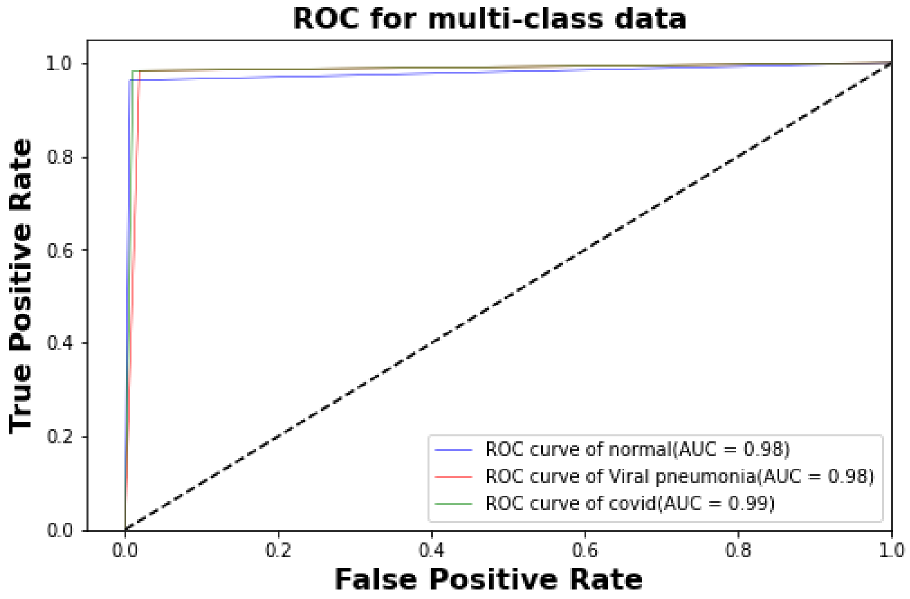


Figure 26.True positive rate (TPR) vs False positive rate (FPR) of ResNet50 model for multiple classifications

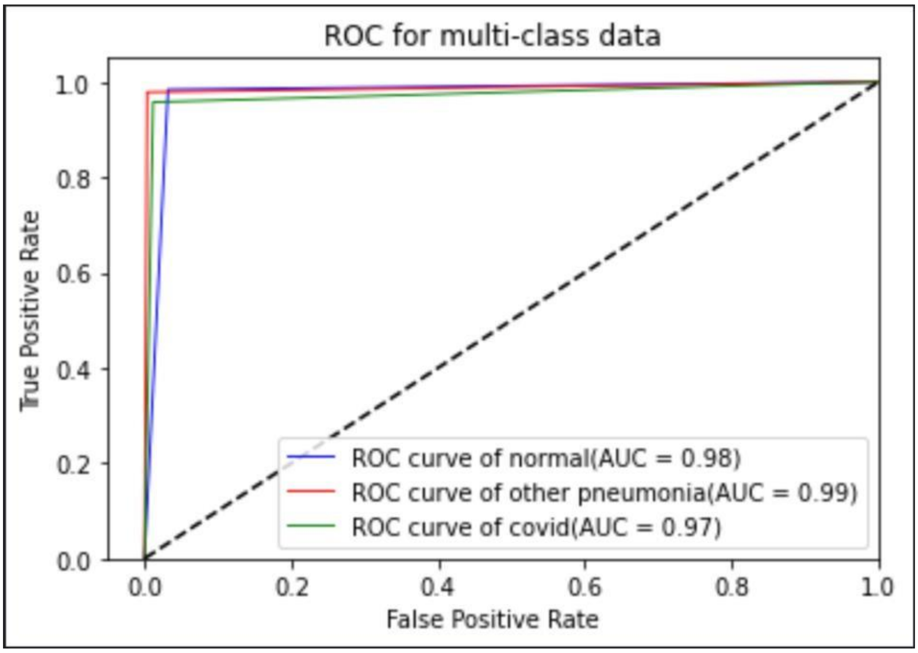


Figure 27. True positive rate (TPR) vs False positive rate (FPR) of EfficientNetB3 NetB3 model for multiple classifications

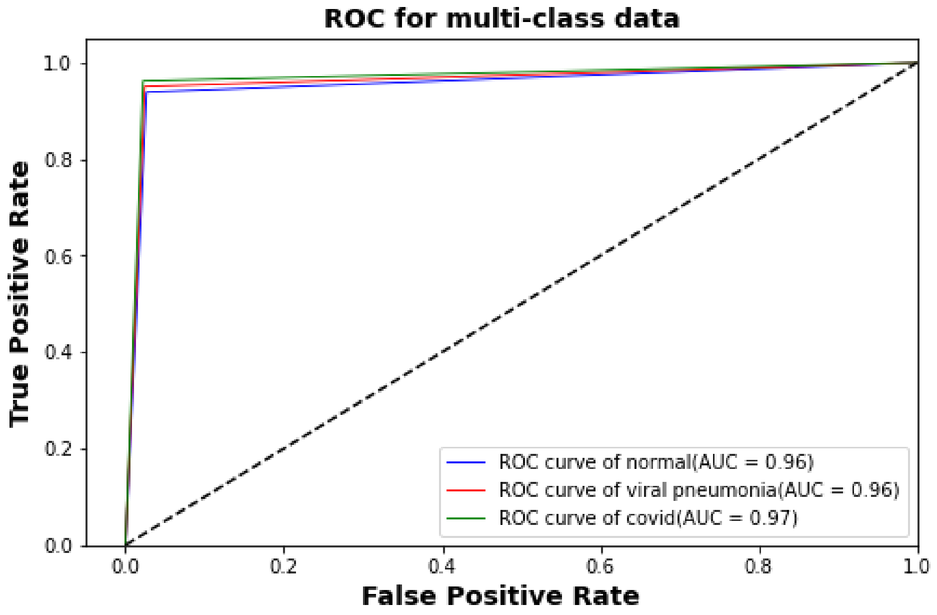
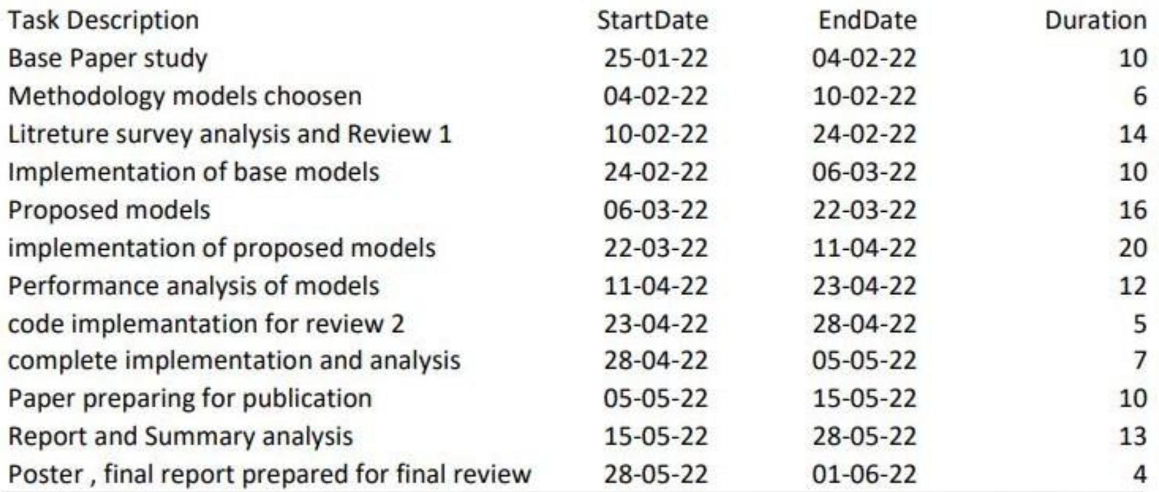
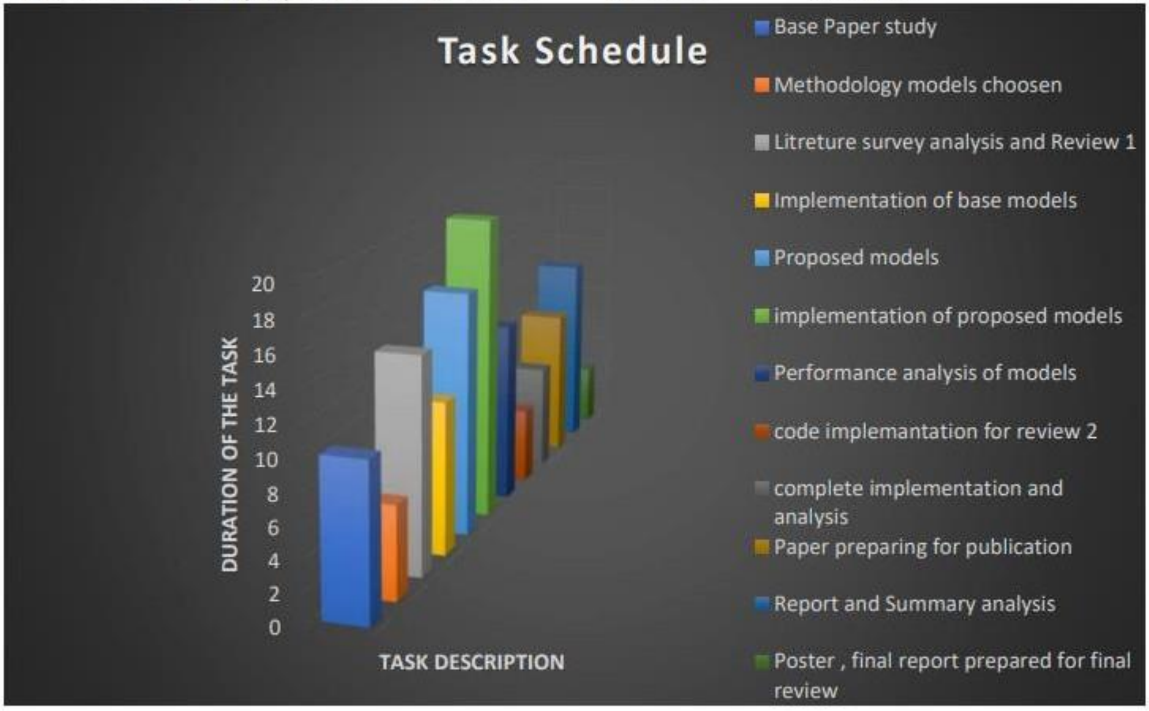


Figure 28. True positive rate (TPR) vs False positive rate (FPR) of custom CNN model for multiple classifications

6. TASK SCHEDULE



Task Description	StartDate	EndDate	Duration
Base Paper study	25-01-22	04-02-22	10
Methodology models choosen	04-02-22	10-02-22	6
Litreture survey analysis and Review 1	10-02-22	24-02-22	14
Implementation of base models	24-02-22	06-03-22	10
Proposed models	06-03-22	22-03-22	16
implementation of proposed models	22-03-22	11-04-22	20
Performance analysis of models	11-04-22	23-04-22	12
code implemantation for review 2	23-04-22	28-04-22	5
complete implementation and analysis	28-04-22	05-05-22	7
Paper preparing for publication	05-05-22	15-05-22	10
Report and Summary analysis	15-05-22	28-05-22	13
Poster , final report prepared for final review	28-05-22	01-06-22	4



7. COST ANALYSIS

This project has been implemented in a software and hence the minimalized.

8. SUMMARY

Using the transfer learning approach, three cutting-edge pre-trained neural networks (eficentNetB3, VGG16, and ResNet50) and a custom-designed CNN model that was trained from scratch were tested with the same hyperparameters. VGG-16 was the most accurate out of the pre-trained models, obtaining 98.48% accuracy, followed by ResNet-50 and efficient Net-B3, achieving 97.9% and 97%, respectively. The custom-designed CNN model achieved 95% accuracy. We observe excellent repeatability and reliability for transfer learning models that are consistently retrained several times based on statistical analysis. According to the results, a transfer learning model can achieve 98.48% recognition accuracy at a learning rate of 0.003 for COVID-19 detection. Our findings demonstrate that detection models built with transfer learning techniques can perform better than custom-designed CNN models on a variety of classification tasks for non-COVID-19, COVID-19, and Viral pneumonia images. As a result of using the weight of a pre-trained model, the proposed model doesn't need to learn about features like edges, shapes, and holes. In addition, the TL will eliminate the need for a large set. The medical field could benefit greatly from this greatly, as we all know it takes medical experts to label data for viruses. Covid-19 infection and viral pneumonia in CXR scans exhibit similar characteristics that are difficult to interpret for radiologists. In our study, transfer learning-based models learned the features and classified the images with reasonable accuracy after fewer than 22 epochs of training. These high accuracies demonstrate that the deep transfer learning models were able to find something distinctive in the CXR images, which allowed them to distinguish the images accurately.

Table 12. Performance evaluation of custom design CNN models and pre-trained architecture with existing literature

Study	Methods (model)	Class	Accuracy (%)
[28]	ResNet-50+SAM+BiLSTM	3	97.26
[16]	DarkCovidNet	3	87.02
[17]	VGG-16	3	96.73
[40]	COVIDiagnosis-Net	3	98.3
[2]	Sg-SqueezeNet	3	98.3
[41]	DenseNet	3	88.9
[42]	Tailored CNN	3	92.3

[43]	ALexnet+Googlenet+Resnet18	3	97.2 (ALexnet)
			98.3(Googlenet)
			99.6(Resnet18)

[19]	Proposed deep CNNs	3	98.7
[44]	EfficientNet-B4	3	96.70
[45]	COV-VGX	3	98.92
[46]	CoroDet	3	94.2
In this study	VGG-16+ResNet-50+Efficient Net-B3+ Costume CNN	3	97.9 (VGG-16) 97.6 (ResNet-50) 97 (Efficient Net-B3) 95(Custom CNN)

In contrast, the custom-designed CNN model that trains without the concept of transfer learning underperforms when compared to the pre-trained model. A comparison will be made between our fine-tuned and custom-designed CNN model and the recent studies that have multiple classifications, too. For multiple classifications, [44] used a deep scaled efficient net, but the accuracy obtained is lower than our proposed efficientNetB3. This could be due to a smaller number of training images. When comparing our custom-designed CNN model that worked on multiple classifications of CXR images, we achieved our 2nd height by achieving 95, which is below than [19] and better than 22 layer CoroDet by [46] as in table Table 12, our proposed VGG-16 model has the highest accuracy, whereas [17] has the second-highest accuracy. A DL algorithm can assist healthcare staff in diagnosing COVID-19 with minimal processing of chest CXR scans.

9. References

[1] WHO, “WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data,” 2022. https://covid19.who.int/ (accessed May 12, 2022).

[2] M. E. H. Chowdhury et al., “Can AI Help in Screening Viral and COVID-19 Pneumonia?,” IEEE Access, vol. 8, pp. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.

[3] M. Al Mehedi Hasan, J. Shin, and F. Parvin, “Deep Transfer Learning Based Detection of COVID-19 from Chest X-ray Images,” ACM Int. Conf. Proceeding Ser., pp. 64–70, 2021, doi: 10.1145/3460238.3460249.

[4] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, “Deep learning based detection and analysis of COVID-19 on chest X-ray images,” Appl. Intell. (Dordrecht, Netherlands), vol. 51, no. 3, pp. 1690–1700, 2021, doi: 10.1007/s10489-020-01902-1.

[5] P. M. Shah et al., “Deep GRU-CNN model for COVID-19 detection from chest X-rays data,” IEEE Access, vol. 10, pp. 35094–35105, 2021, doi: 10.1109/ACCESS.2021.3077592.

[6] H. Kim, “Outbreak of novel coronavirus (COVID-19): What is the role of radiologists?,” Eur. Radiol., vol. 30, no. 6, pp. 3266–3267, 2020, doi: 10.1007/s00330-020-06748-2.

[7] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, “Covid-19 image data collection: Prospective predictions are the future,” arXiv Prepr. arXiv2006.11988, 2020.

[8] C. Us, “What are the Radiation Risks from CT ?,” Public Health, pp. 2005–2007, 2006.

[9] 2017 FDA, “What are the Radiation Risks from CT? | FDA.” https://www.fda.gov/radiation-emitting-products/medical-x-ray-imaging/what-are-radiation-risks-ct (accessed May 12, 2022).

[10] C. Dennie et al., “Canadian Society of Thoracic Radiology/Canadian Association of Radiologists Consensus Statement Regarding Chest Imaging in Suspected and Confirmed COVID-19,” Can. Assoc. Radiol. J., vol. 71, no. 4, pp. 470–481, 2020, doi:

10.1177/0846537120924606.

[11] C. Hani et al., “COVID-19 pneumonia: A review of typical CT findings and differential diagnosis,” Diagn. Interv. Imaging, vol. 101, no. 5, pp. 263–268, May 2020, doi: 10.1016/j.diii.2020.03.014.

[12] S. Hassantabar, M. Ahmadi, and A. Sharifi, “Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches,” Chaos, Solitons and Fractals, vol. 140, Nov. 2020, doi: 10.1016/j.chaos.2020.110170.

[13] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib, “SPECIAL SECTION ON SCALABLE DEEP LEARNING FOR BIG DATA A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor,” doi: 10.1109/ACCESS.2020.2978629.

[14] T. Rahman et al., “Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images,” Comput. Biol. Med., vol. 132, no. November 2020, p. 104319, 2021, doi: 10.1016/j.compbiomed.2021.104319.

[15] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 3462–3471, 2017, doi: 10.1109/CVPR.2017.369.

[16] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” Comput. Biol. Med., vol. 121, no. April, p. 103792, 2020, doi: 10.1016/j.compbiomed.2020.103792.

[17] I. D. Apostolopoulos and T. A. Mpesiana, “Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks,” Phys. Eng. Sci. Med., vol. 43, no. 2, pp. 635–640, 2020, doi: 10.1007/s13246-020-00865-4.

[18] S. Toraman, T. B. Alakus, and I. Turkoglu, “Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks,” Chaos, Solitons and Fractals, vol. 140, Nov. 2020, doi:

10.1016/j.chaos.2020.110122.

[19] M. Nour, Z. Cömert, and K. Polat, “A Novel Medical Diagnosis model for COVID-19 infection detection based on Deep Features and Bayesian Optimization,” Appl. Soft Comput., vol. 97, p. 106580, 2020, doi: 10.1016/j.asoc.2020.106580.

[20] M. Toğaçar, B. Ergen, and Z. Cömert, “COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches,” Comput. Biol. Med., vol. 121, no. March, 2020, doi: 10.1016/j.compbiomed.2020.103805.

[21] M. M. Rahaman et al., “Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches,” J. Xray. Sci. Technol., vol. 28, no. 5, pp. 821–839, Jan. 2020, doi: 10.3233/XST-200715.

[22] A. Keles, · Mustafa, B. Keles, and · Ali Keles, “COV19-CNNet and COV19-ResNet: Diagnostic Inference Engines for Early Detection of COVID-19,” vol. 1, p. 3, doi: 10.1007/s12559-020-09795-5.

[23] M. Loey, F. Smarandache, and N. E. M. Khalifa, “Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning,” Symmetry (Basel)., vol. 12, no. 4, 2020, doi: 10.3390/SYM12040651.

[24] S. Latisha, A. C. Halim, R. Ricardo, and D. Suhartono, “COVID-19 Detection Model on Chest CT Scan and X-ray Images Using VGG16 Convolutional Neural Network,” pp. 532–538, 2022, doi: 10.1109/isriti54043.2021.9702839.

[25] H. I. Fitriasari and M. Rizkinia, “Improvement of Xception-ResNet50V2 Concatenation for COVID-19 Detection on Chest X-Ray Images,” 3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021, pp. 343–347, 2021, doi: 10.1109/EIConCIT50028.2021.9431916.

[26] R. Badrahadipura, S. Q. Nur Septi, J. Fachrel, I. N. Yulita, A. A. Pravitasari, and D. Agustian, “COVID-19 Detection In Chest X-Rays Using Inception Resnet-v2,” pp. 104–109, 2022, doi: 10.1109/icaibda53487.2021.9689723.

[27] S. Chaudhary and Y. Qiang, “Ensemble deep learning method for Covid-19 detection via chest X-rays,” Conf. Proc. 2021 Ethics Explain. Responsible Data Sci. EE-RDS 2021, pp. 6–8, 2021, doi: 10.1109/EE-RDS53766.2021.9708581.

[28] U. Muhammad, M. Z. Hoque, M. Oussalah, A. Keskinarkaus, T. Seppänen, and P. Sarder, “SAM: Self-augmentation mechanism for COVID-19 detection using chest X-ray images,” Knowledge-Based Syst., vol. 241, p. 108207, 2022, doi: 10.1016/j.knosys.2022.108207.

[29] T. Rahman et al., “Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images,” Comput. Biol. Med., vol. 132, p. 104319, 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104319.

[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” [Online]. Available: http://image-net.org/challenges/LSVRC/2015/.

[32] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” 36th Int. Conf. Mach. Learn. ICML 2019, vol. 2019-June, pp. 10691–10700, 2019.

[33] C. Sitaula and M. B. Hossain, “Attention-based VGG-16 model for COVID-19 chest X-ray image classification,” vol. 19, pp. 2850–2863, 2021.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[35] M. Tan et al., “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” Jul. 2018, Accessed: May 10, 2022. [Online]. Available: http://arxiv.org/abs/1807.11626.

[36] T. Ahmed and N. H. N. Sabab, “Classification and Understanding of Cloud Structures via Satellite Images with EfficientUNet,” SN Comput. Sci., vol. 3, no. 1, 2022, doi: 10.1007/s42979-021-00981-2.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[38] R. K. Gupta and J. Manhas, “Improved Classification of Cancerous Histopathology Images using Color Channel Separation and Deep Learning,” J. Multimed. Inf. Syst.,

vol. 8, no. 3, pp. 175–182, 2021, doi: 10.33851/jmis.2021.8.3.175.

[39] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[40] F. Ucar and D. Korkmaz, “COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images,” Med. Hypotheses, vol. 140, p. 109761, 2020, doi: https://doi.org/10.1016/j.mehy.2020.109761.

[41] X. Li and D. Zhu, “COVID-Xpert: An AI Powered Population Screening of COVID-19 Cases Using Chest Radiography Images.” arXiv, 2020, [Online]. Available: http://europepmc.org/abstract/PPR/PPR346252.

[42] L. Wang, Z. Q. Lin, and A. Wong, “COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images,” 2020, doi: 10.1038/s41598-020-76550-z.

[43] M. Loey, S. El-Sappagh, and S. Mirjalili, “Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data,” Comput. Biol. Med., vol. 142, p. 105213, Mar. 2022, doi: 10.1016/j.compbiomed.2022.105213.

[44] G. Marques, D. Agarwal, I. De La, and T. Díez, “Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network,” Appl. Soft Comput. J., vol. 96, p. 106691, 2020, doi: 10.1016/j.asoc.2020.106691.

[45] P. Saha, M. S. Sadi, O. F. M. R. R. Aranya, S. Jahan, and F.-A. Islam, “COV-VGX: An automated COVID-19 detection system using X-ray images and transfer learning,” Informatics Med. unlocked, vol. 26, p. 100741, 2021, doi: 10.1016/j.imu.2021.100741.

[46] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, and M. Z. Parvez, “CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images,” Chaos, Solitons and Fractals, vol. 142, Jan. 2021, doi: 10.1016/j.chaos.2020.110495.

10. Appendix

Poster

CXRcovNet: COVID−19 detection from CXR images using transfer learning approaches.

Kemal Mudie Tosora |18BCE2433 |Rajkumar S|SCOPE

INRODUCTION

The new COVID−19 (coronavirusdisease 2019) pandemichas caused over 516 million illnesses and 6.2 million deaths worldwideas of May 12th, 2022, with a total of 11.5 + million vaccination doses provided. As a result of the virus's ongoingdissemination over the world, the SARS−CoV−2 mutations produced a new COVID−19 wave. Among the variants identified were the Beta, XE, and omicron types. There were fears that a new variant based on the original SARS−CoV−2 strain might be more extremely contagious. In many places, COVID−19 detection takes place through reverse transcription polymerase chain reaction (RT−PCR) tests, which may take longer than 48 h. This is one major reason for its severity and rapid spread.

In this work, we propose an automatic detection method called CXRcovNet, an 11−layer custom−designed CNN model, and three state−of−the−art models were evaluated with CXR images to detect COVID−19. Transfer learning techniques were used on the three pre−trained CNN models (VGG−16, ResNet−50, and EfficientNetB3), while the CXRcovNet models were designed and trained from scratch. As part of the experiment, all pre−trained and CXRcovNet models were trained with the same hyperparameters and their performance was checked for matrix−like sensitivity, specificity, f1−score, precision, accuracy, and ROC curve.

MOTIVATION

The gold standard COVID−19 detection technique is real−time polymerase chain reaction (RT−PCR). However, RT−PCR kits are costly and take 6−9 hours to verify infection in a patient. Due to the lower sensitivity of RT−PCR, it provides high false−negative results.

To resolve this problem, radiological imaging techniques like chest X−rays and CT scans are used as alternatives to detect and diagnose COVID−19. In this paper, we found out the chest x−ray tests are economically affordable, and the results are relatively easy to use. Chest X−ray tests are easily available, have portable versions and low risk of radiation. On the other hand, CT scans have high risk of radiation, are expensive, need clinical expertise to handle and are non−portable. This makes the use of X−ray scans more convenient than CT scans.

Moreover, COVID−19 reveals some radiological signatures that will be easily detected through CXRs. For this, radiologists are required to research these signatures. However, it's a time−consuming and error−prone task. Hence, there's a necessity to automate. Therefore, our objective is to develop an automatic DL system for the detection of COVID−19 samples from healthy and pneumoniacases using CXR images.

SCOPE OF THE PROJECT

Our suggested CXRCOVNet and 3 pre-trained model is especially essential for countries where laboratory testing kits are unavailable. The use of a chest X-ray to prioritize patients for further RT-PCR testing, which might be useful in an inpatient setting where existing systems are confused whether to retain the patient on the ward with other patients or segregate them in COVID -19 zones.

Our research advances the prospect of an accurate, automated, quick, and low-cost technique for aiding in the diagnosis of COVID -19 using chest X-ray images. This would allow hospitals and medical clinics all around the world to diagnose illnesses in chest X-rays.

METHODOLOGY

Approach 1 is to examine covid−19 infection using three pretrained models (VGG−16, Resnet−50, and efficient Net Net−B3). The concept of transfer learning is used to fine−tune three pretrained models for covid identification in this method. The purpose of this technique is to investigate if a neural network can detect the difference between a covid−19 infection and other lung abnormalities

Approach 2, CXRcovNet, an 11−layer custom−designed CNN architecture that is trained from scratch, is used to classify covid−19 and its performance is compared to the three pretrained models (VGG−16, Resnet−50, and efficient Net Net−B3) to assess the significance of transfer learning when we have a small number of data sets.

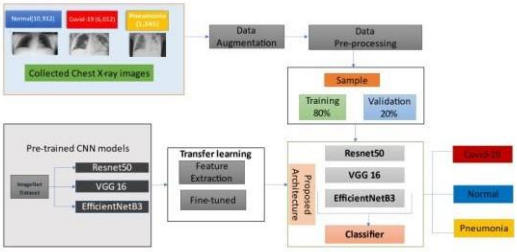


Fig 1. Proposed model diagram

RESULTS

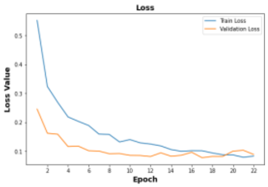
For both the fine−tuned (EficentNetB3, VGG16, and ResNet50) and CXRcovNet (11−layer custom design CNN) models, its train accuracy and loss values for their best epochs during the training and testing phase summarize in table 1 and the model performance matric in terms of Accuracy, precision, Sensitivity, F−Score, Specificity and ROC value summarize in table 2.

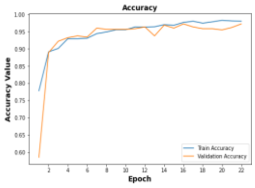
Table 1. Compression of the models for the best epoch in terms oTfrainAcc and ValLoss

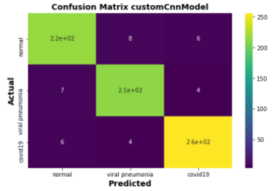
Model	Best Epoch	BS	TrainLoss	TrainAcc	ValLoss	ValAcc
VGG-16	22	32	0.083	0.964	0.089	0.973
ResNet-50	22	32	0.005	0.99	0.157	0.969
EfficientNet-B3	22	32	1.413	0.997	1.440	0.966
CXRcovNet	22	32	0.045	0.984	0.106	0.972

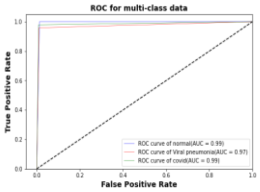
Table 2. The performance of each proposed model to classify cov-i1d9, non-covid-19, and viral-pneumonia occurrences.

Classification model	Class	Precision%	Recall %	F1-score %	AUC %	Accuracy %
EfficientNet-B3	Covid-19	97	100	98	98	97
	Normal	99	93	96	99	
	Pneumonia	96	98	97	97	
VGG-16	Covid-19	97.5	100	98.7	99	97.9
	Normal	97.8	95.8	96.8	99	
	Pneumonia	98.3	97.9	98.1	97	
ResNet-50	Covid-19	98.7	96	97	99	97.6
	Normal	96	98	97	98	
	Pneumonia	98.7	98	98	98	
CXRcovNet	Covid-19	96.2	96.2	94.8	97	95
	Normal	94.3	93.9	94.1	96	
	Pneumonia	94.6	95.0	94.8	96	









SUMMARY

The technique of transfer learning was used on three pre−trained neural networks (eficentNetB3, VGG16, and ResNet50) in the first approach, while CXRcovNet, an 11−layer custom−designed CNN model, was trained from scratch and evaluated with the same hyperparameters in the second approach.

VGG−16 was the most accurate out of the pre−trained models, obtaining 98.48% accuracy, followed by ResNet−50 and efficient Net−B3, achieving 97.9% and 97%, respectively. CXRcovNet achieved 95% accuracy. In our study, transfer learning−based models learned the properties and classified the images with minimal error after just 22 trainingepochs.

REFERENCES

[1] WHO, "WHO Coronavirus (COVID -19) Dashboard | WHO Coronavirus (COVID -19) Dashboard With Vaccination Data," 2022. https://covid19.who.int/ (accessed May 12, 2022).

[2] M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID -19 Pneumonia?," *IEEE Access*, vol. 8, pp. 132665 –132676, 2020, doi: 10.1109/ACCESS.2020.3010287.

PUBLICATIONS