Transfer Learning-Based Deep Learning Models for Screening Covid-19 Infection from Chest CT Images

Dr. S. Malliga
Kongu Engineering College, Department of Computer Science and Engineering
Perundurai, Erode, Tamil Nadu,
India.

Dr. S. V. Kogilavani
Kongu Engineering College,
Department of Computer Science and Engineering
Perundurai,Erode, Tamil Nadu,
India

R. Deepti
Kongu Engineering College,
Department of Computer Science and Engineering
Perundurai, Erode, Tamil Nadu,
India

S. Gowtham Krishnan
Kongu Engineering College,
Department of Computer Science and Engineering
Perundurai, Erode, Tamil Nadu,
India

G. J. Adhithiya
Kongu Engineering College,
Department of Computer Science and Engineering
Perundurai, Erode, Tamil Nadu,
India

Abstract— As the global prevalence of Covid-19 rises, accurate diagnosis of Covid-19 patients is critical. The biggest issue in diagnosing people who test positive is the non-availability or scarcity of testing kits, as Covid-19 spreads rapidly in the community. To prevent Covid-19 from spreading among humans as an alternative quick diagnostic method, an automatic detection system is required. We propose in this study to employ Convolution Neural Networks to detect corona virus-infected patients using Computed Tomography (CT) images. In addition, we look into the transfer learning of deep convolutional neural networks like VGG16, inceptionV3, and Xception for detecting infection in CT scans.To find the best values

for hyper-parameters, we use Bayesian optimization. The study comprises of comparing and analysing the employed pre-trained CNN models. According to the data, all trained models are more than 93 percent correct. Pretrained models such as VGG16, InceptionV3, and Xception have attained more than 97 percent precision. Furthermore, our method introduces novel methods for classifying CT scans in order to detect the Covid-19 infection.

Keywords— Covid-19, CT Images, Convolution Neural Networks, VGG16, InceptionV3, Xception

I. INTRODUCTION

Coronavirus (SARS-CoV-2) is a kind of SARS virus found in 2019. It has various mutant variants, some of which are dangerous and cause a pandemic of respiratory sickness. Symptoms appear two to fourteen days after being exposed to the virus. A person infected with the coronavirus can be communicable to others for up to two days before symptoms develop and for 10 to 20 days after symptoms appear, depending on immunity and disease severity. The Covid-19 pandemic is now sweeping the globe. Every day, more person dies from pneumonia caused by the SARS-CoV-2 virus.People who test positive for Covid-19 should find out if they are infected as soon as possible so they can avoid contact with others, keep away from them, and alert close contacts. Currently, a laboratory test of nose and throat samples is required for a formal diagnosis of Covid-19. Reverse Transcript - Polymerase Chain Reaction is the name of this test (RT-PCR). The RT-PCR approach necessitates the use of specialised test kits, which may be in short supply and need at least 24 hours. It is not completely accurate, and a second RT-PCR or other test may be necessary to confirm the diagnosis. Covid-19 is a lung disease. Practitioners can diagnose people with Covid-19 symptoms using chest imaging while they wait for RT-PCR or RT-PCR, based on the person's Covid19 symptoms.

Because Covid-19 test kits are in poor availability, a Chinese study proposes that chest radiography (X-rays) and chest Computed Tomography (CT) scans can help with diagnosis. Both, particularly Covid-19, may be indicative of lung disease abnormalities. Chest CT is more sensitive than chest X-ray in detecting early Covid-19 illness. A CT chest scan is a specialist imaging test that uses X-rays to create 3D chest pictures. Chest CT scans are a more accurate imaging approach for detecting lung issues. A chest CT scan, which is used to diagnose pneumonia, may be done fast and easily.According to recent research, CT offers a sensitivity of 98 percent for Covid-19 infection, compared to a sensitivity of 71% with RT-PCR testing. As a result, CT has emerged as one of the most important tools for early detection and diagnosis of Covid-19 [1, 2]. Chest CT scans, on the other hand, can produce both false-negative and false-positive results. As a result, new computer-aided lung CT diagnosis techniques are crucial for properly confirming suspected infections, screening patients, and monitoring viral outbreaks. With the rapid advancement of artificial intelligence, computer vision techniques, which were originally employed for image recognition, have been applied to medical imaging, including CT images [3].Deep learning algorithms based on medical imaging have been developed to extract image information, including spatial correlations. CNN, in particular, demonstrated encouraging results in feature extraction. The CNN excelled in a variety of computer vision tasks [4, 5]. CNN has emerged to become the premier deep learning approach for medical image categorization in recent years, thanks to its self-learning capabilities.

CNN-based algorithms for Covid-19 detection, such as Resnet [34], VGG 16 [34], inception, and exception Net, have been presented [6], and these strategies employ very few training samples from medical image datasets. CNNs have been constructed for huge datasets using a variety of methodologies such as parameter optimization, data augmentation, and transfer learning to achieve unmatched efficiency.In addition there are multiple hybrid betworks such as RCoNet [37], SD Net [36] etc., developed to increase the accuracy percentage in Covid detection.However, the current approach has significant flaws in datasets [27], forecasting negative values [26], and removing CAP from COVID-19 [28]. To address the existing shortcoming of the CNN model, the proposed method created a modified CNN architecture employing transfer learning and fine tuning method.

In this research, we create a collection of models that use pre-trained VGG16, InceptionV3, and Xception architectures to automatically identify and detect Covid-19 infection using CT images collected from the web. To fine-tune pre-trained models for re-purposing, we employ two strategies: feature extractor and fine-tuner. We freeze the convolution base of the pre-trained models for feature extraction, and for fine-tuning, we train some convolution layers while freezing others. Another component of this research is that it use Bayesian optimization to discover the best configuration for hyper-parameters utilised in training multiple CNN architectures. The following sub - sections have made substantial contributions to this work:

(i) Examining the performance of CNN variations using transfer learning, fine-tuning, and rigorous simulations.

(ii) A comparison of the accuracy, precision, sensitivity, and specificity metrics of classic and new CNN architectures;

(iii) Using Bayesian optimization to optimise hyper-parameters.

The study's key uniqueness is that it employs transfer learning with two fine-tuning procedures for pre-trained models. We evaluate the performance of the three CNN architectures by adapting previously trained models constructed on the Image-net data set and evaluating them on the CT image data set via transfer learning. To the best of our knowledge, no other study in the literature has included transfer learning with two fine-tuning procedures, as well as Bayesian Optimization, for classifying CT images.

The rest of the article is structured as follows: The second section addresses recent research on CNN-based deep learning algorithms for detecting Covid-19 infection in CT images. Section 3 delves deeply into datasets, deep neural network topologies, fine-tuning techniques, and Bayesian optimization. The fourth section goes into experimental setup, hyper-parameter adjustment, and performance measurements. Section 5 summarises the study's findings and compares the models generated. Finally, in Section 6, we provide a summary of the proposed work.

II. LITERATURE SURVEY

Within a short period of time, research on the diagnosis of Covid 19 infection increased dramatically. We addressed some of the most recent deep learning algorithms in automated Covid 19 detection in this part.

Mishra et al. [7] evaluated various deep CNN-based techniques for detecting the presence of Covid19 in chest CT images. A strategy based on decision fusion is also proposed, which integrates forecasts from numerous models to produce a final prediction. The experimental results suggest that the proposed decision fusion-based strategy can achieve more than 86% on all performance criteria. The authors of [8] create a deep learning-based method for automatically segmenting and quantifying infection regions as well as the entire lungs from chest CT scans. To segment Covid-19 infection sites in CT scans, the “VB-Net” neural network is used in the deep learning-based segmentation method. The new DL-based segmentation approach has been trained on 249 Covid-19 CT scans and validated on 300 CT scans.

Pathogen-confirmed Covid-19 cases, together with those who had previously been diagnosed with normal viral pneumonia, were studied by Wang et al. [9]. They adjusted the inception transfer-learning model and then applied internal and external validation to obtain the algorithm. To ensure accuracy, we performed both external and internal validation, and the results showed an overall accuracy of 89.5%, with a specificity of 0.88 and a sensitivity of 0.87. The external testing dataset yielded a total accuracy of 79.3%, a specificity of 0.83, and a sensitivity of 0.67. DenseNet-121 CNN model for classification and identification of patients with Covid-19 was done in [6] and the experimental analysis explored several indices including sensitivity, specificity, and AUC, in addition to testing on several data sets, all of which were extensively explored and reported in the study.

Chest CT scans have been collected from 88 patients diagnosed with Covid-19 patients in hospitals in two Chinese provinces, 100 patients infected by pneumonia, and 86 healthy people for comparison and modeling [10]. A deep CT diagnostic system for the identification of Covid-19 patients was developed based on the data. The authors claim that their models could precisely distinguish Covid-19 patients from bacterial pneumonia patients. A collection of five pre-trained convolutional neural network-based models (ResNet50, ResNet101, ResNet152, InceptionV3, and Inception-ResNetV2) were proposed in [11] to identify patients who had pulmonary infection with the coronavirus. Three alternative binary classifications, each consisting of four classes, were implemented using five-fold cross-validation (Covid-19, normal, viral pneumonia, and bacterial pneumonia). The pre-trained ResNet50 model has the best classification accuracy according to the performance results.

[12] employs a CNN to determine if individuals are infected or not. In addition, multi-objective differential evolution is employed to fine-tune CNN's initial parameters. Extensive studies are conducted out on chest CT scans utilizing the proposed and competing for machine learning techniques. A thorough examination demonstrates that the suggested model can correctly categories chest CT images.

To segment the peripheral and transition zones of the prostate from MRI images, a Bayesian deep neural network [29] was built. Technically, it outperforms the U-net and USE-Net models. Experiment findings reveal that the Bayesian neural network performs well in CT-based models as well. To aggregate the level of Covid infection from the CT picture, a parallel partial decoder known as Inf-Net [35] was constructed. In this case, reverse and explicit edge attention are employed to improve boundary and segmented learning utilising semi-supervised learning.

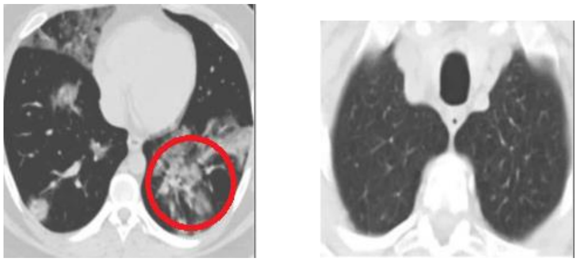
As evidenced by the studies mentioned above, deep learning architectures are increasingly being used in the identification of Covid-19 infection from CT images, However, some gaps in the use of deep learning architectures must be filled, such as faster training time, proper tuning of hyper-parameters, fewer parameters, and so forth. In this paper, we use transfer learning to try to reduce training time and find the best set of hyper-parameters for the proposed pretrained models.

III. MATERIALS AND METHODS

The primary purpose of this research is to develop and compare a few models for classifying CT images using various CNN architectures. We suggest using both conventional and contemporary CNN architectures, assessing their performance, and identifying models that detect Covid-19 infection with high accuracy. This section details all of the materials and methods used in this study.

A. Dataset

From training to testing classification models, an appropriate dataset is necessary at all phases of object recognition. A total of 968 CT images have been collected from various sources, including Kaggle, and other websites, and divided into two categories: Covid-19 and non-Covid-19. Fig. 1 depicts a sample of each type of CT image. The Covid-19 infection is distinguished by the red circle in Fig. 1(a). There are 459 images with Covid-19 infection and 521 images without infection.



(a) (b)

Fig. 1 CT image of lungs (a) Covid-19 positive image and (b) non-Covid-19 image

B. Image augmentation

Training a neural network necessitates a large amount of data. The more information the network has, the more features it will be able to learn. As we have a limited number of images, we have applied image augmentation techniques. When we don't have enough data to train deep learning models, image augmentation is an effective technique. We have applied the image augmentation techniques such as rotation, shifting, flipping, blurring, scaling, cropping, padding, translation, and affine transformation. Additionally, we have used in-place/on-the-fly data augmentation. The primary advantage of using this augmentation is that it provides augmentation in real-time. That is, while a model is still being trained, it generates augmented images on the fly and ensures that each epoch the model receives new variations of the images. The images in the dataset are pre-processed before training to improve feature extraction and consistency. All of the images in this study have been normalized and resized to 224*224 pixels (for VGG16) and 299*299 pixels (for InceptionV3 and Xception) pixels at 96*96 dots per inch. The total number of images in the dataset has increased to 3562 after augmentation. The dataset has been split into 70% training, 15% validation, and 15% test sets to test the various CNN architectures.

C. Methods

Several baseline CNN architectures for image recognition applications have been developed and successfully applied to complex visual imagery tasks. While traditional network architectures like LeNet, AlexNet, and VGG16 were entirely made up of stacked convolutional layers, modern architectures like ResNet50, InceptionV3, and Xception look for novel and innovative ways to build convolutional layers to maximize learning efficiency. In this study, three CNN architectures, VGG16, InceptionV3, and Xception, have been tested for their ability to detect infection in CT images. These models are summarized in the following sections.

Basic CNN:

CNN architecture is a popular deep learning tool for image classification and an important technique for modeling complex processing in applications that use large amounts of data. It is on the cutting edge of image classification tasks and has been programmed to derive visual patterns directly from input images [13, 14]. Lecun et al. [15] successfully solved the problem of handwritten digit classification using CNN with a gradient-based algorithm. It then advanced to the forefront of several object recognition tasks, and it is now used in a variety of other domains such as object tracking and identification, text and action recognition, and so on.

A CNN consists of several artificial neuron layers. Artificial neurons are mathematical functions that compute the weighted sum of inputs to produce an activation value. When neurons in a CNN are fed pixel values, they choose different visual features. A CNN has two basic building elements namely feature extraction and classification blocks.

Convolution and pooling layers comprise the feature extraction block. The classification block has been flattened and is fully connected. Filters in a convolutional layer represent lower-dimensional features of the input data. To generate feature maps, the filters convolve the entire input. The feature maps are then subsampled, and the pooling layers reduce the feature maps' dimensionality. The final convolution layer's feature map is flattened into an array. Finally, we have fully connected two or three hidden layers and an output layer that classifies between two or more classes using a classifier. CNN is used as a feature extractor and classifier in image processing applications, despite their use in image processing and classification. Fig. 2 depicts a typical CNN.

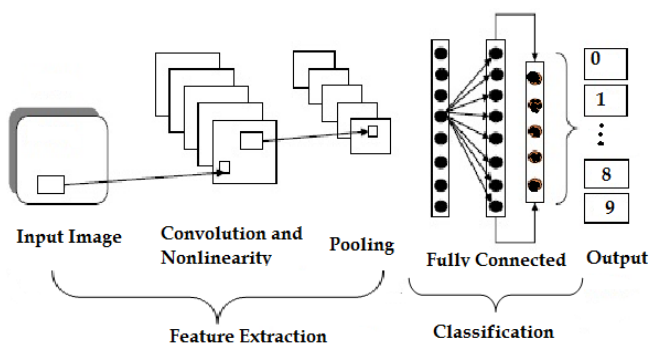


Fig. 2. Architecture of CNN

CNN can automatically learn hierarchical feature representations, which is an important feature. The first few layers of CNN are often responsible for detecting basic features such as horizontal, vertical, and diagonal edges, among others. The output of these layers is transferred to the intermediate layers, which are responsible for extracting more sophisticated features such as corners and edges. As we progress deeper into the network, the layers begin to recognize higher-level elements such as objects, faces, and other such things as well. This means that the features computed by the first layers are generic and can be applied to a wide range of problems, whereas the features computed by the later layers are specific to the dataset and task at hand. In comparison to conventional feed-forward neural networks, CNN has a significant advantage in that it requires a smaller number of neurons and hyper-parameters. Several baseline CNN architectures for image recognition applications have been developed and successfully applied to complex visual imagery tasks. In this study, we choose to use pre-trained models such as VGG16, InceptionV3, and Xception to create proposed models. The rationale for selecting these pretrained models is presented below.

The VGG16 object-recognition model is a common object-recognition model with up to 16 layers. VGG16, which was designed as a deep CNN, outperforms ImageNet on several tasks and datasets. VGG16 is still one of the most widely used image-recognition models today. So, we propose to use this model.

InceptionV3 was one of the first models to use batch normalization. It also employed the factorization approach to make computations more efficient. InceptionV3 would parallelize the operations and perform convolutions and batch normalization in parallel before concatenating the results, rather than executing processes linearly and adding more parameters and complexity. With directed acyclic graphs, InceptionV3 offers sophisticated processing.

Xception is a CNN built entirely of depth-wise separable convolutional layers. Xception combined ResNets and Inception with depth-wise separable convolutions to create a light, yet high-performance network. On the ImageNet dataset, Xception attained more accuracy compared to InceptionV3. We shall discuss these ground-breaking CNN designs in the following sections.

CNN variants:

VGG16:

A 16-layer neural network, VGG16, presented in 2014 by Simonyan and Ziserman of Oxford University's Visual Geometry Group Lab [16] was revealed to be even deeper than AlexNet but with significantly reduced computational complexity because the architecture utilizes multiple 3x3 kernel-size filters in place of a single large kernel filter. VGG16 has 13 convolutional and 3 fullyconnected layers, all of which carry the ReLU from AlexNet. VGG16's architecture is depicted in Fig. 3. VGG19, a deeper form of VGG16, has also been constructed, consisting of 19 layers in total (16 convolution layers and 3 fully connected layers).

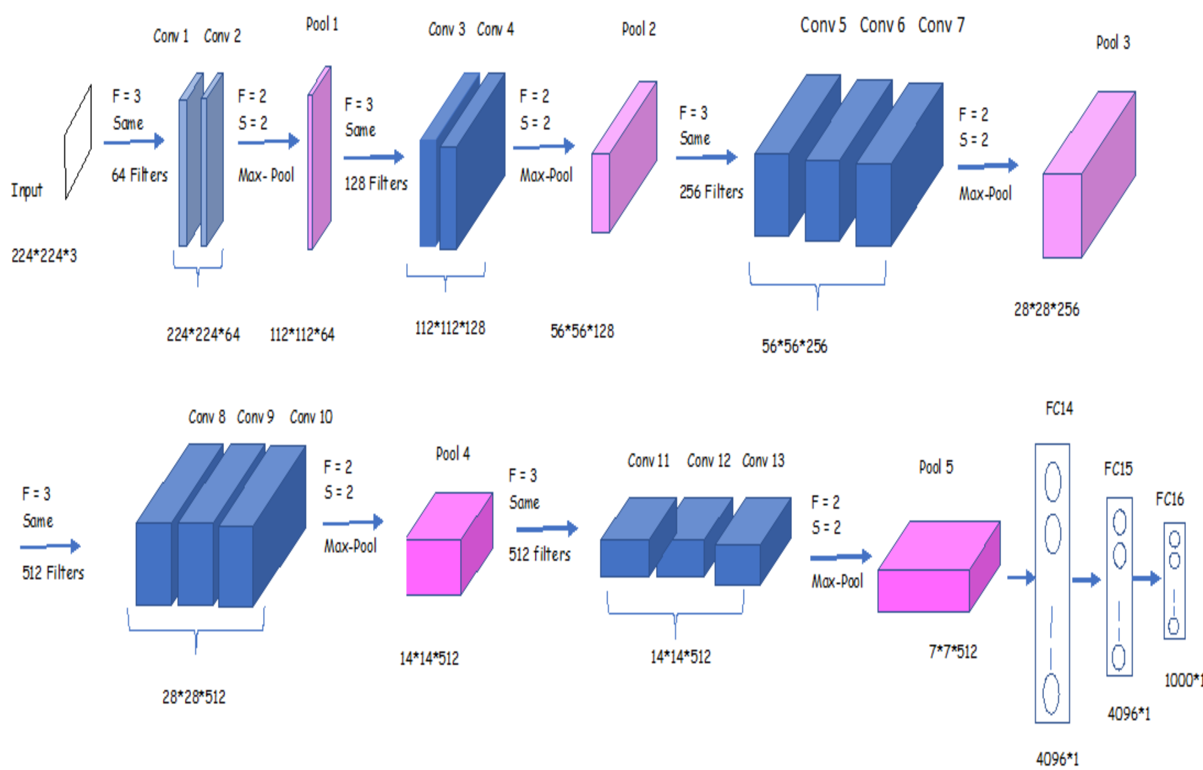


Fig. 3. VGG16 architecture with 13 convolution layers and 3 fully connected layers

Inception:

GoogLeNet has been introduced in [17] as the Inception deep convolutional architecture (InceptionV1). The initial architecture was subsequently improved in several ways: firstly, batch normalization [18] and namedInceptionV2. Smoothing of labels factorized 7 x 7 convolutions, and the addition of an auxiliary classifier to transport label information down in the network are all added in the next variant, InceptionV3[19]. Softmax is included in the last layer of the InceptionV3 architecture, which has 48 layers and a 299 x 299 pixels input layer. InceptionV3 reduces processing costs significantly while maintaining speed and precision.

Xception:

Xception is a modification to the Inception architecture that substitutes depth-wise separable convolutions for regular Inception modules. In a nutshell, the Xception architecture is a depth-separable convolution layer stack with residual connections. This model substituted depth-wise separable convolutions [20](spatial convolution performed separately on each input channel) for traditional inception modules (1*1). In most traditional classification problems, the Xception architecture outperformed VGG16, ResNet, and InceptionV3.

Explainable AI:

The key to unlock the black box for deep learning is explainable AI (XAI). XAI is an artificial intelligence model that is built to explain its aims, reasoning, and decision-making to end users.

Various terms have been used to describe attributions, such as “feature visualization” “saliency maps” and “heat map” [38]. In proposed work the major characters such as "activation maps" and "gradients" are employed to generate the attributions. This technique identifies essential input characteristics for a network decision. Single pixels, random pixels, or super-pixels in output images are perturbed by occlusion techniques, which replace them with black, grey, or blur masks. The influence of this perturbation on the input is a measure of the significance of the associated pixels in the prediction. The focus of this study is gradient-based class activation mapping (CAM).

Transfer learning:

Scholars' interest in transfer learning has lately increased. It is a method for creating new artificial intelligence models by fine-tuning previously trained neural networks [21]. It is commonly demonstrated in computer vision by the use of pre-trained models. Canziani et al. [22] used the ImageNet dataset to investigate the performance of pre-trained models on computer vision tasks. A CNN, as shown in Fig. 2, is made up of two independent components: a convolutional base and a classifier. The basic objective of the convolutional base is to extract features from an image. The basic objective of the classifier is to categorise the image based on the extracted features.The lower layers of a CNN's convolutional base learn general characteristics, whereas the classifier part and a few top layers of the convolutional base, learn specialized features.

Because we have a small sample size, which is typical in medical imaging, transfer learning, even if based on ImageNet, is beneficial [23]. The initial layers of the pretrained model are frozen in this study, and only the remaining higher layers are retrained. It would then be necessary to customize the top layers to the new data collection. We make use of the pretrained model as a classifier in this case. Because the new data set has a low degree of resemblance to ImageNet, it is important to retrain and customize the upper layers to account for the differences in the new data.

To demonstrate the performance of fine-tuning, we just use the pretrained models as feature extractors also. In this approach, we can keep the convolution base in its original form with the weights learned from ImageNet. In pretrained models, the classifier generates 1000 different output labels, but the number of neurons in the output layer can be decided based on the number of classes in our dataset. So, we can import the convolution base and add our classifier. The output from the convolution base is fed into the classifier

D.Bayesian optimization

The number of hidden units, dropout, activation function, weight initialization, and other hyper-parameters define the network structure, as well as how the network is trained, such as learning rate, momentum, batch size, epochs, and so on. The goal of hyper-parameter tuning is to find optimal values for hyper-parameters so that a loss function can be minimized and better results can be obtained. The hyper-parameters can be fine-tuned using a variety of methods, including manual, grid, random searches, and

Table 1. Hyperparameter setting of the deep learning model.

Parameter	Search Space	Description
Optimizer	Adam, RMSProp, SGD, Nadam	To optimize the input weights by comparing the prediction and the loss function
Learning rate	1e-3, 1e-4, 1e-5, 1e-6	To determine the step size at each iteration while minimizing the loss function
Activation function	Relu, Elu, LeakyRelu, and Tanh	To introduce non-linearity into the output of neurons
Number of neurons in customized layers	64,128, 56, 512,1024	To compute the weighted average of the input

Table 1. shows the hyperparameter setting in deep learning models. Bayesian optimization. Unlike Grid and Random search, Bayesian optimization makes use of previous iterations of the algorithm. Each hyper-parameter guess is independent in the grid and random searches. On the other hand, with Bayesian approaches, we get closer to the perfect solution with each selection and testing of alternative hyper-parameters. In other words, Bayesian optimization aids decision-making about which hyper-parameter combination is best for evaluating a model [24, 25]. Deep learning models can take a long time to train due to the amount of data and computing density. In these cases, Bayesian optimization can be very useful. We use Bayesian optimization in conjunction with pre-trained models to optimize the hyper-parameters of the classifier layers.

IV. Experiments

We conducted two sets of experiments, namely Feature Extractor and Fine-tuner, to compare the performance of models developed using pretrained CNN models, namely VGG16, InceptionV3, and Xception. The details are provided below.

A. Experimental platform

The proposed modelsare compiledin Graphical Processing Units (GPUs) because they consume a lot of power and require high-performance hardware to function properly. The hardware and software configurations used are listed in Table 2. We imported the necessary Keras model architectures and instantiated them with ImageNet weights.

Table 2. Experimental platform

Parameters	Details
GPU	GPU DELL EMC 740
RAM	128 GB
GPU RAM	32 GB
DISK	4TB
OS	Ubuntu
Language	Python
IDE	Jupyter notebook environment on Google Co-laboratory

B. Tuning of Hyper-parameters

In deep learning algorithms, hyper-parameters are important because they specify training details and select the appropriate hyper-parameter values. In this paper, we use Bayesian optimization to find the best hyper-parameter values while maintaining high accuracy. It is a method for finding the minimum or maximum of an objective function that employs Bayes' theorem. In this study, the objective function is accuracy, which we want to maximize. The hyperparameters tuned are optimizers, learning rate, activation feature, and neurons. Table 3 summarizes the hyperparameters we work on and their search areas.

C. Experimental setup

In the first set of trials, Pretrained models like VGG16, InceptionV3, and Xception were employed as feature extractors, and the extracted features were then utilised to train the newly added classifier. The convolution foundation was built using the weights learned from the ImageNet dataset. We retrain a few top layers of the proposed models' convolution basis in the second round of experiments. We can skip retraining the pretrained models' early layers since they learn lower-level characteristics from the input images, resulting in a large reduction in training effort. As a result, simply training the classifier and the convolutional base layers is adequate.

The best values for the hyper-parameters are determined through the application of Bayesian optimization. The Bayesian optimization process has been repeated 25 times. For each iteration of Bayesian optimization, we set the number of epochs to 100. In each iteration, we recorded the accuracy and loss. Table 3 displays the hyper-parameter values that resulted in the highest accuracy in each strategy for all models. Because more iterations did not result in significant changes, the hyper-parameters discovered after 25 iterations were deemed the best in our study. We may obtain a better set of values for hyper-parameters if we use a large search space but at the expense of a high computation time.

Table 3. Hyper-parameters with tuned values

Hyper Para meters	**VGG 16**		**Inception V3**		**Xception**	
	Feature Extractor	Fine-tuner	Feature Extractor	Fine-tuner	Feature Extractor	Fine-tuner
Optimizer	Adam	Adam	RMSProp	RMSProp	Adam	Adam
Learning rate	0.0001	0.0001	0.0001	0.0001	0.001	0.0001
Activation function	tanh	tanh	relu	relu	relu	relu
Number of neurons in customized layers	512	128	64	1024	128	256

D. Performance metrics

Following the development of the models, the next stage is to assess their effectiveness using some metrics for the test datasets. The accuracy, precision, specificity, and sensitivity of the various CNN models produced in this study were used to evaluate their performance. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) indices are used to generate these measures (FN). TP is defined as the number of correctly classified photos for each class. FP denotes the number of photos misclassified in all other classes except the correct one. FN denotes the number of photos misclassified in the relevant class.TN is the number of images correctly classified in all other classes except the correct class.

V. Results and Discussion

This section presents the performance of the proposed models and compares their performance with each other.

A. Results of proposed models

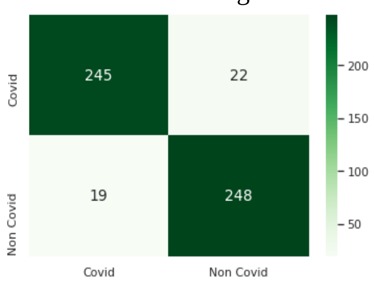
We evaluated the performance of the pretrained CNN models developed in this work. The experiments have been carried out with the tuned hyper-parameters shown in Table 3, which produced the best results during training. Table 4 displays the overall validation and testing accuracy for each of the models.

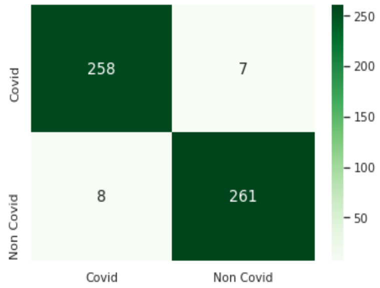
Table 4. Performance of the proposed models

Experiment scenario	VGG16		InceptionV3		Xception	
	Validation Accuracy (%)	Testing Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)
Feature Extractor	93.1	92.3	92.15	93.07	93.1	94.5
Fine-tuner	95.6	97.1	96.1	97.5	98.4	98.3

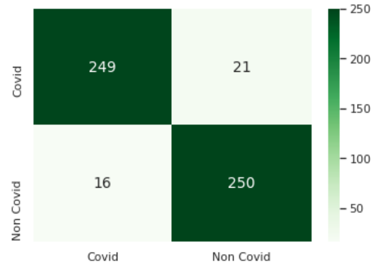
As can be seen in Table 4, using the pretrained models as feature extractors results in lower validation and testing accuracy than retraining a few top layers. The fine-tuning strategy keeps the weights of previously trained models from the initial layers and retrains them on top layers. The weights of the initial layers are usually retained because the features obtained from these layers are generic and applicable to a wide range of tasks. The latter layers, on the other hand, provide more specific features that, because they have been tailored to our dataset, can benefit from the fine-tuning approach. In Strategy 1, however, the pretrained models are simply used as feature extractors without being fine-tuned, and the top layers of the models are not retrained specifically for our dataset. As a result, the learned features are unique to ImageNet, and the accuracy is lower than in strategy 2.

We have also assessed the performance of the proposed models against each one of the classes. We used the indices TP, FP, TN, and FN to calculate the values of various performance metrics such as accuracy, precision, specificity, and sensitivity. To find the values of these indices, we use the confusion matrix obtained during training and testing the models. The confusion matrix, as we all know, is a visualization tool used in supervised learning and can be used to see how well the prediction results correspond to the actual data. Fig. 4 depicts the confusion matrix obtained while training all the models in both the scenarios we took. The diagonal elements of the confusion matrix represent correct classifications. The predicted classes are represented by X-axis, and the actual classes are represented by Y-axis. For instance, from Fig. 4 (a), it can be understood that the model classified 22 non-Covid images as Covid

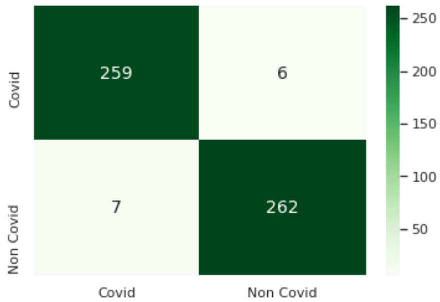




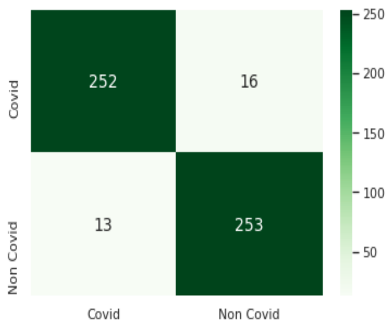
(b)



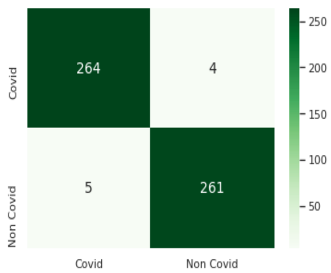
(c)



(d)



(e)



(f)

Fig. 4. Visualization results (a) VGG16 (feature extractor), (b) VGG16 (fine-tuner), (c) InceptionV3 (feature extractor), (d) InceptionV3 (fine-tuner), (e) Xception (feature extractor) and (f) Xception (fine-tuner)

(a)

Using the values of TP, FP, TN, and FN, we have calculated the metrics such as accuracy, precision, specificity, and sensitivity for each of the proposed models and presented in Table 5

Table 5. Performance of the pre-trained models

Models		Accuracy	Precision	Sensitivity	specificity
VGG 16	Feature Extractor	92.32	91.76	92.8	91.85
	Fine-tuner	97.19	97.36	96.99	97.39
InceptionV3	Feature Extractor	93.1	92.22	93.96	92.25
	Fine-tuner	97.57	97.74	97.37	97.76
Xception	Feature Extractor	94.57	94.03	95.09	94.05
	Fine-tuner	98.31	98.51	98.14	98.49

Table 5 compares the results based on performance metrics like accuracy, precision, specificity, and sensitivity. We discovered that, like overall accuracy, the fine-tuning strategy outperformed the feature extractor strategy. We believe we achieve higher accuracy because we fine-tune the top layers of the convolution base with our data set. Also, the performance of VGG16, InceptionV3, and Xception in all metrics is found to be greater than 93%. Furthermore, among all pretrained models developed in this work, the pretrained Xception model performed the best in terms of metrics. This is because Xception employs depth-wise separable convolution, which improves learning efficiency and performance.

Table 6. Comparison of the suggested method to state-of-the-art procedures.

Method	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)
ResNet-50 [30]	55.48	62.45	76.25	50.45
COVID-Net [31]	58.45	63.15	78.25	56.12
VB-Net [32]	67.57	75.91	74.25	68.64
COV-Net [33]	77.58	86.14	76.14	89.45
Ours	**93.54**	**97.57**	**93.45**	**94.25**

Table 6 illustrates a comparison of the suggested technique to state-of-the-art procedures. The term "ours" refers to the performance of the anticipated transfer learning results. Based on these findings, we may conclude that transfer learning, even if ImageNet-based, is advantageous when the sample size is limited, as it is in most medical imaging scenarios, and that when the convergence rates of the different models are compared, the pretrained models converge faster.

VI. CONCLUSIONS

Because of the Covid-19 outbreak, CT has recently become the most convenient and quickest method of diagnosis. However, difficulties arise as a result of a lack of measurable diagnosis criteria and skilled radiologists. As a result, computer-aided image classification of CT images is critical in medical image analysis.

The proposed model uses the Pretrained CNN models such as VGG16, InceptionV3, and Xception study to predict Covid-19 infection. Herein, the Transfer learning based on ImageNet was selected because of the benefits of not requiring excessively large training datasets and simply retraining the weights of a few top layers, which takes less processing resources. Furthermore, Bayesian optimization is employed to indicate the best values for hyper-parameters used during training. The experimental result of proposed method yields higher accuracy of 93% and 97% precision.The findings of this work suggest that employing pretrained models driven by Bayesian Hyper-parameter optimization and transfer learning for CT image classification to detect Covid-19 infection is a promising alternative.

References

[1] J. Lei, J. Li, X. Li, X. Qi, "CT Imaging of the 2019 Novel Coronavirus Pneumonia", Radiology, vol. 295, no. 1, Jan.2020, pp. 18-18.

[2] H. Shi, X. Han, C. Zheng, "Evolution of CT Manifestations in a Patient Recovered from 2019 Novel Coronavirus Pneumonia in Wuhan, China", Radiology, vol. 295, Feb 2020, pp. 20-20.

[3] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, L. Wang, "Learning to Navigate for Fine-Grained Classification", In Proc. European Conf. on Computer Vision, Munich, Germany, 8-14, 2018, pp. 420-435.

[4] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks", Adv. in Neural Information Proce. Systems, vol. 25, Aug.2012, pp. 1097-1105.

[5] J. Qin, W. Pan, X. Xiang, Y. Tan, G. Hou, "A Biological Image Classification Method Based on Improved CNN", Ecological Informatics, Vol. 58, July. 2020, pp. 101093.

[6] N. Hasan, Y. Bao, A. Shawon, Y. Huang, "DenseNet Convolutional Neural Networks Application For Predicting COVID-19 using CT Image", SN Computer Science, Vol. 2, No. 5, July. 2021, pp. 1-

11.

[7] K. Mishra, S. K. Das, P. Roy, S. Bandyopadhyay, “Identifying COVID19 from Chest CT Images: A Deep Convolutional Neural Networks-Based Approach", J. Healthcare Engineering, Vol. 2020, Aug. 2020, pp.1-7.

[8] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, et al., "Abnormal Lung Quantification in Chest CT Images of COVID 19 Patients with Deep Learning and its Application to Severity Prediction", Medi. Physics, vol. 48, Nov. 2021, pp. 1633-1645.

[9] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, Et Al., “A Deep Learning Algorithm Using CT Images to Screen for Corona Virus Disease (COVID-19)", European Radiology, Feb.2021, pp.1-9.

[10] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, Et Al., "Deep Learning Enables Accurate Diagnosis of Novel Coronavirus (COVID-19) with CT Images", IEEE/ACM Trans. Comput. Biology and Bioinformatics, Mar. 2021, pp. 1-1.

[11] A. Narin, C. Kaya, Z. Pamuk, "Automatic Detection of Coronavirus Disease (Covid-19) Using X-Ray Images and Deep Convolutional Neural Networks", Pattern Analysis and Applications, vol. 24, May, 2021, pp. 1-14.

[12] D. Singh, V. Kumar, M. Kaur, “Classification Of COVID-19 Patients from Chest CT Images using Multi-Objective Differential Evolution–Based Convolutional Neural Networks", European Journal of Clinical Microbiology & Infectious Diseases, Vol. 39, No. 7, April, 2020, pp. 1379-1389.

[13] Y. Lecun, Y. Bengio, G. Hinton, "Deep Learning,” Nature, Vol. 521, May, 2015, pp. 436-444.

[14] Y. Lecun, F. J. Huang, L. Bottou, "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting", in Proce. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, CVPR, Vol. 2, Washington, DC, USA, 27 June-2 July 2004, pp. II-104.

[15] Y. Lecun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, Et Al., "Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition", Neural Networks: The Statistical Mechanics Perspective, vol. 261, 1995, pp. 2.

[16] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint arXiv:1409.1556 (accessed: 2014),Apr,20105.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, "Going Deeper with Convolution”, Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, Nevada, 7-12 June 2015, pp. 1-9.

[18] S. Ioffe, C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, in Proce. Inter. Conf on Machine Learning, Lille, France, 07-09 July 2015, pp. 448-456.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", in Proces IEEE Conf. Computer Vision and Pattern Recognition, San Juan, PR, USA, 27-30 June 2016, pp. 2818-2826.

[20] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", in Proces IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, Hawaii, 21-26 July 2017, pp. 1251-1258.

[21] J. Chen, J. Chen, D. Zhang, Y. Sun, Y. A. Nanehkaran, "Using Deep Transfer Learning for Image-Based Plant Disease Identification”, Computers and Electronics in Agriculture, Vol. 173, June, 2020, pp. 105393.

[22] A. Canziani, A. Paszke, E. Culurciello, An Analysis of Deep Neural Network Models for Practical Applications, arXiv preprint arXiv:1605.07678 (accessed: 2016).

[23] M. Raghu, C. Zhang, J. Kleinberg, S. Bengio, Transfusion: Understanding Transfer Learning for Medical Imaging arXiv preprint arXiv:1902.07208 (accessed: 2019).

[24] M. I. Sameen, B. Pradhan, S. Lee, "Application of Convolutional Neural Networks Featuring Bayesian Optimization for Landslide Susceptibility Assessment", Catena, vol. 186, Mar, 2020, pp. 104249.

[25] M. Pelikan, D. E. Goldberg, S. Tsutsui, “Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms," in Proces SICE 2003 Annual Conference (IEEE Cat. No. 03TH8734), Fukui, Japan, 4-6 August 2003, pp. 2738-2743.

[26] H. Ma, Q. Ye, W. Ding, Y. Jiang, M. Wang, Z. Niu, X. Zhou, Y. Gao, C. Wang,W. Menpes-Smith, E. F. Fang, “Can Clinical Symptoms and Laboratory Results Predict CT Abnormality? Initial Findings Using Novel Machine Learning Techniques in Children With COVID-19 Infections”, Frontiers in Medicine, 8, June,2021, pp.855.

[27] M. Roberts, D. Driggs, M. Thorpe, J. Gilbey, M. Yeung, S. Ursprung, A.I. Aviles-Rivero, C. Etmann, C. McCague, L. Beer, J.R. Weir-McCall, “Common Pitfalls and Recommendations for Using Machine Learning to Detect and Prognosticate for COVID-19 Using Chest Radiographs and CT Scans”, Nature Machine Intelligence, vol. 3, Mar, 2021, pp. 199-217.

[28] S. Hu, Y. Gao, Z. Niu, Y. Jiang, L. Li, X. Xiao, M. Wang, E.F. Fang, W. Menpes-Smith, J. Xia, H. Ye, “Weakly Supervised Deep Learning for Covid-19 Infection Detection and Classification from CT Images”, IEEE Access, Vol. 8, June, 2020,

pp.118869-118883.
[29] Y. Liu, G. Yang, M. Hosseiny, A. Azadikhah, S. A. Mirak, Q. Miao, S. S. Raman, K. Sung, "Exploring Uncertainty Measures in Bayesian Deep Attentive Neural Networks for Prostate Zonal Segmentation", IEEE Access, Vol. 8, Aug,2020, pp.151817-151828.
[30] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", in Proces of the IEEE Conf. Comp. Vision and Pattern Recognition, Las Vegas, Nevada, 27-30 June 2016, pp. 770-778.
[31] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng, B. Xu, "A Deep Learning Algorithm Using CT Images to Screen for Corona Virus Disease (COVID-19)", European Radiology, vol. 31, Feb, 2021 pp.1-9.
[32] X. Ouyang, J. Huo, L. Xia, F. Shan, J. Liu, Z. Mo, F. Yan, Z. Ding, Q. Yang, B. Song, F. Shi, "Dual-sampling Attention Network for Diagnosis Of COVID-19 from Community-Acquired Pneumonia", IEEE Trans. Medical Imaging, vol. 39, Aug,2020, pp.2595-2605.
[33] M. Abdel-Basset, V. Chang, H. Hawash, R.K. Chakrabortty, M. Ryan, "FSS-2019-nCov: A Deep Learning Architecture for Semi-Supervised Few-Shot Segmentation of COVID-19 Infection", Knowledge-Based Systems, Vol. 212, Jan, 2021, pp.106647.
[34] Y.H. Wu, S.H. Gao, J. Mei, J. Xu, D.P. Fan, R.G. Zhang, M.M. Cheng, "Jcs: An Explainable Covid-19 Diagnosis System by Joint Classification and Segmentation", IEEE Trans. Image Processing, vol. 30, Aug, 2021, pp.3113-3126.
[35] D.P. Fan, T. Zhou, G.P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, L. Shao, "Inf-net: Automatic Covid-19 Lung Infection Segmentation from Ct Images", IEEE Trans. Medical Imaging, Vol. 39, No. 8, May, 2020, pp.2626-2637.
[36] S. Tabik, A. Gómez-Ríos, J.L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J. L. Suárez, J. Luengo, M. A. Valero-González, P. García-Villanova, "COVIDGR Dataset and COVID-SDNet Methodology for Predicting COVID-19 Based on Chest X-Ray Images", IEEE Journal of Biomedical and Health Informatics, vol. 24, Nov,2020, pp.3595-3605.
[37] S. Dong, Q. Yang, Y. Fu, M. Tian, C. Zhuo, "RCoNet: Deformable Mutual Information Maximization and High-order Uncertainty-aware Learning for Robust COVID-19 Detection", IEEE Trans. Neural Networks and Learning Systems, vol. 32, No. 8, Feb, 2021, pp. 3401- 3411.
[38] Q. Ye, J. Xia, G. Yang, Explainable AI for COVID-19 CT Classifiers: An Initial Comparison Study, *arXiv preprint arXiv:2104.14506* (accessed: 2021).

Dr. S.Malliga is working as a Professor in the Department of Computer Science and Engineering, Kongu Engineering College, Tamil Nadu, India. She obtained PhD in the year 2010 from Anna University, Chennai. Her research area inlcudes Computer Network and Security. She has done many consultancy projects including VoIP, Web site developement for many industries. She has also offered many training programmes urses on latest technology and trends. Currently she is guiding four research scholars. She has also guided many UG and PG projects. She has published 50 articles in international journals and presented more than 40 papers in national and international conferences in her research and other technical areas. In 2018, she is awarded with Summer Reseacrh Fellowship by Indian Academy of Sciences.

Dr. S.V.Kogilavani is associated with the Department of Computer Science and Engineering as an Associate Professor at Kongu Engineering College, Tamil Nadu, India. Her research interests include Information Retrieval and Summarization. She received her Ph.D from Anna University, Chennai in the year 2013. She has presented many papers in national and international conferences and published 15 papers in national and international journals. She is also awarded with Summer Reseacrh Fellowship by Indian Academy of Sciences.

Ms.R.Deepti is doing final year Computer Science and Engineering at Kongu Engineering College. She has done projects in IOT, Web development and Deep learning. She is currently pursuing her intern at Maersk Global Services

Mr. R. Gowthamkrishnan is currently pursuing final year of Bachelor's degree in Computer Science and Engineering at Kongu Engineering College, He has done projects in Web Development, IOT and Deep Learning domain. And now he is currently undergoing intern in Zoho Corporations.

Mr. G.J. Adhithiya is doing final year Computer Science and Engineering at Kongu Engineering College. He is currently pursuing BE CSE. He has keen interest towards modern tech and web development and has done 3 mini projects so far.