Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 15, No. 3, 2023 Article ID IJIM-1640, 16 pages
DOI: http://dx.doi.org/10.30495/ijim.2023.15261.817
Research Article
Science and Research Branch (IAU)

# An Ensemble Deep Learning Model for Detection Covid-19 from CT Scan Images

M. Ghasemnezhad M.I. *, H. Izadkhah †‡

Received Date: 2022-04-12 Revised Date: 2022-08-11 Accepted Date: 2022-08-13

**Abstract**

Diagnosis of covid-19 using deep learning on CT scan images can play an important role in helping doctors. In this paper, by combining EfficientNet-B2 and vision transformers ($ViT-1-32$) neural networks a new deep transfer learning is proposed. For evaluation, con-fusion matrix, precision, accuracy, recall, and F1 score are used. The experimental results are 0.9838 for validation accuracy, 0.9667 for test accuracy, and 0.9839 for accuracy.

*Keywords* : Deep Learning; Convolutional Neural Network; CT Scan; Covid-19; Transfer Learning; Ensemble Methods.

## 1 Introduction

Coronaviruses are single-stranded positive-sense RNA viruses with a positive polarity that infect humans and other animals. On December 31, 2019, a new strain of coronavirus was isolated from patients with pneumonia of unknown cause in Wuhan city and named severe acute respiratory syndrome coronavirus (SARS-Cov-2) by the ICTV (International Committee on the Classification of Viruses Committee). According to the announcement of the World Health Organization on March 11, 2020, Covid-19 became an international concern [9, 25, 1, 26]. Several diagnostic methods have been used for detecting the coronavirus in clinical, research, and public health laboratories. A diagnostic test method should have sufficient sensitivity and accuracy to make appropriate clinical decisions rapidly during a pandemic [8]. Nucleic acid amplification by reverse transcription polymerase chain reaction (RT-PCR) is the most widely used method for direct SARS-CoV-2 detection [8, 17, 19].

### *1.1* Diagnosis methods

While the RT-PCR and rapid test are currently the standard way used for early diagnosis of the disease, the sensitivity shown by these tests is not optimal. The false positive and false negative results of these tests have increased the difficulty of early identification and isolation of patients, and thus caused a further delay in making deci-

*Department of Computer Engineering, University College of Daneshvaran, Tabriz, Iran.
†Corresponding author. izadkhah@tabrizu.ac.ir, Tel:+98(914)1015690.
‡Department of Computer Science, University of Tabriz, Tabriz, Iran.

sions, and taking the proper actions [26, 8, 17]. Due to the lack of hospitalization, and getting necessary treatments, the highly contagious nature of the virus, the low sensitivity of RT-PCR tests, as well as its time-consuming procedure, a patient may even pass away until the diagnosis result is determined. In addition, the highly infectious nature of this virus increases the risk of infecting more people. Therefore, rapid diagnosis of Covid-19 is necessary to treat and control this disease [19]. Radiologists use X-rays, computed tomography (CT), and ultrasound screening tests to confirm the presence of COVID-19. Ground-glass opacity in the early stage and pulmonary embolism that shows linear stabilization in the late stage are identified as two remarkable patterns in diagnosing the virus infection [17]. By analyzing these images, radiologists can help doctors in the early diagnosis of Covid-19 [25]. Tao et al. reported the diagnostic significance and accuracy of chest CT images in RT-PCR in covid-19. Their findings show that chest CT has a high sensitivity for the diagnosis of Covid-19 [26]. On the other hand, Guan et al. reported radiographic abnormalities of positive cases of Covid-19 such as interstitial abnormalities, bilateral abnormalities, and ground-glass opacities in both CT and chest X-ray images [26].

Visual diagnosis is time-consuming and laborious as it requires specialized manpower and ultimately might not be accurate enough. Automated analysis of Covid-19 can reduce the workload of hospital staff by rapid diagnosis [19]. Artificial Intelligence-based solutions can be used for the automatic detection of Covid-19 [31]. Machine learning, deep learning, and AI-based approaches have been used for the detection and classification of various diseases. Thus, as an alternative, AI-based solutions can provide efficient solutions that can help in the automatic learning of feature patterns from CT scan images, which can enhance the radiologists' decision-making process and lead to more effective management of the situation [17]. Although these methods may never replace human care professionals, they seem to be an effective solution to fight the virus. These methods should be used as complements to support the medical staff. New methods are therefore required to effectively support the medical manpower [16].

### *1.2* Convolutional Neural networks

Convolutional neural networks (CNNs) are widely used for image analysis, which makes them a popular approach for detecting COVID-19 from CT or CXR images. CNN architecture mainly consists of three types of layers including convolutional, pooling, and fully connected layers. Convolution layers extract features from images using convolution kernels. Pooling layers reduce the resolution of feature maps based on operations such as average or max-pooling to achieve shift-invariance, and finally, Fully-connected layers are used for classification based on obtained feature maps from previous layers. By using several layers, the kernels of the first convolutional layer are used to extract the low-level features of the image such as edges, and the subsequent layers extract the high-level features from the image. Convolution layers extract image features at a high level. Although different methods such as Support Vector Machine (SVM) can be used [20].

### *1.3* Transfer Learning

Well-labeled data in medical imaging are abundant and rarely available due to the high cost and necessary workload of radiology experts. A couple of other alternative techniques are available for training a model efficiently on a smaller dataset: data augmentation and transfer learning [32].

Transfer learning is a method in which knowledge gained from one domain can learn in another domain. Thus, a deep neural network can be trained on one domain with enough data, and the obtained knowledge from that domain can be used to train the network with little data from another domain [20]. Put differently, transfer learning is a common and effective strategy to train a network on a small dataset, where a network is trained on an extremely large dataset, such as ImageNet containing 1.4 million images with 1000 classes, then reused on any other task [32].

In fact, in transfer learning, it is assumed that the features extracted from a set with a lot of data can be used in a smaller set with a smaller amount of data. This portability of learned generic features is a unique advantage of deep learning that makes it useful in various domain tasks with small datasets. Currently, many models pre-trained on the ImageNet challenge dataset are available and accessible to the public, along with their learned kernels and weights, such as Alex Net, VGG, ResNet, Inception, and DenseNet [32]. Using this method may even have better results than using human experts [20].

Two different strategies of transfer learning exist for image classification. In the first strategy, the trained network is used to extract features, and in the second, the same network can be well-adjusted on the images of corona patients. The results of using these strategies may be contradictory, but in general, using transfer learning dramatically improves classification accuracy. Using this method can sometimes even outperform human experts [20].

### *1.4* Efficient Net Architecture

To get better accuracy, the scale of ConvNets should be increased. For example, ResNet can be scaled up from ResNet18 to ResNet200. In most cases, the depth or width is increased to change the scale. There is another method that is less commonly used, but is very popular, and that is enlarging the models by changing the resolution of the images. In the models before this model, it was used only by changing one of the dimensions, depth, and width or image size. While it is possible to change the two-dimensional and three-dimensional scale at will, this type of scaling is both extremely tedious and often does not achieve the desired accuracy and efficiency [30]. Efficient Net group consists of 8 models between B0 and B7, and the number of calculated parameters does not increase as the layer of the model increases even though the accuracy increases considerably. Despite other CNN models, Efficient Net uses a new activation function called Swish instead of the Rectifier Linear Unit (ReLU) activation function [30].

Deep learning architectures aim to find more efficient approaches with smaller models. Unlike other state-of-the-art models, EfficientNet scales the depth, width, and resolution uniformly, and achieves more efficient results while the model is scaled down. The first task in the hybrid scaling method is to find a network to re-establish the relationship between the different scaling dimensions of the base network assuming resource constraints. Thus, a suitable scaling coefficient is specified for depth, width, and resolution dimensions, which are then applied to scale the baseline network to the target network [2]. The hybrid scaling method seems reasonable because if the input image is larger, the network needs more layers to increase the receiving field and more channels to capture finer-grained patterns on the larger image [30].

### *1.5* Vision Transformer Architecture

Vision transformers (ViT) have recently demonstrated significant results in image processing while requiring fewer computational resources in comparison to other similar architectures. Based on self-attention architectures, the transformer has become a leading model in natural language processing (NLP). Vision transformers, in particular, perform very well when trained on sufficient data, outperforming state-of-the-art CNNs with four times fewer computational resources. One of the advantages of transformers is their computational efficiency and scalability. With these techniques, models of unprecedented size, with more than 100 billion parameters, can potentially be trained [7].

Vision Transformer has fewer image-specific induction biases such as translation equation and location than CNN, and therefore cannot be generalized when trained on insufficient amounts of data. ViT achieves excellent results when it is pre-trained to a sufficient scale and transferred to tasks with fewer data points [5].

### *1.6* Ensemble Methods

Neural networks are designed to recognize patterns in data, and their ability to learn is very

high. However, there is a drawback that each time they are trained, a different set of weights is obtained, resulting in different predictions. To solve this problem, a better and more successful approach, known as ensemble approach, was proposed, which trains several models instead of one model and obtains final predictions by combining the results of each trained model [6]. Ensemble methods are the most effective approaches in machine learning that generally outperform individual models. But group models increase the algorithmic cost and complexity of the model [29]. The Ensemble models have the following specifications:

- They build multiple and diverse predictive models from adapted versions of the training data with resampling or reweighting.
- The predictions of these models are combined in different ways, most of the time by normal averaging or by voting, sometimes with their weights and sometimes without them.

Since the idea of an ensemble method is to build a large model class of predictions whose elements are carefully selected, they may achieve a better overall prediction [29]. Another technique for building an ensemble model is to repeatedly re-optimize and average the solutions. This can be achieved by reinforcing a group of "weak learners" as your set. A weak learner is a weak model that still describes some important features of the data. Hence, it makes sense that compounding over the appropriate set of weak learners produces a strong learner. In general, there are two main principles for generating ensemble models. First, combining models represents a stronger model class than a simple selection of one of them. As a result, the weighted sum of predictions from a set of models can achieve better performance than individual predictions because linear combinations of models produce less bias than any individual model. The predictions that the ensemble models combine remain distinct because they are based on diverse assumptions that cannot be easily integrated, and they have different parameters with diverse estimates. In fact, ensemble models only improve model selection techniques while the models in the ensemble achieve different predictions [29].

## 2 Related Work

Since the onset of COVID-19, many models based on deep learning have been proposed to diagnose the disease. These models are mainly trained using X-ray or CT scan images. Because this epidemic is ongoing and spreading at a very high rate, there are not enough labeled radiographic images of good quality to train a neural network. Due to this, the data set is sparse and also highly unbalanced in almost all the introduced models. Therefore, most of the presented models in the literature use transfer learning-based architecture [17].

In [17], VGG16 and ResNet50 base models were described and various performance improvement techniques were discussed. This study integrated VGG16 and ResNet50 models to present their proposed deep learning model used for the detection of COVID-19 from CT scan images. Also, data augmentation and various performance improvement techniques have been utilized to improve the learning capabilities of base models.

In [19], a novel deep Convolutional Neural Network-Long Short Time Memory (CNN-LSTM) model was presented which is trained to extract features directly from raw data rather than using Transfer models. For this purpose, X-ray images (based on different categories) have been collected from different databases. Their study has examined seven various Scenarios of Bacterial, Viral, COVID-19, and Healthy from the chest X-ray imagery in 4 classes to provide high accuracy to separate classes from each other. The integration of the CNN and Long-Short Time Memory (LSTM) networks can reduce feature dimensions, and improves stability, the training process, the speed of convergence, and detection accuracy. In the presented model an end-to-end classifier is used which it does not need any feature extraction or feature selection. Thus, the optimal features of every class are learned with the deep CNN-LSTM model, automatically. In this

study, five convolutional layers and three LSTM layers are combined.

Alhudhaif, et al. [1] developed a reliable deep CNN model based on a transfer learning approach to identify the patients infected by COVID-19 pneumonia by utilizing CXIs. The generalization of the model to minimize the possible biases and increase the reliability of the model is the main idea of this work, which is done by radiologists by removing the CXIs bias. Three different pre-trained architectures, DenseNet-201, ResNet-18, and SqueezeNet, were used to train the model and assessed by the calculated confusion matrices. The results showed that the proposed model can be used to classify covid19 patients in binary form with high-performance parameters. The model was trained and tested using real data previously confirmed by their clinical diagnosis of pneumonia.

Suto, et al. [27] employed several deep learning classifiers such as deep neural network (DNN) and several pre-trained CNNs like residual neural network (Res-Net50), visual geometry group network 16 (VGG16), and inception network V3 (In-ceptionV3) for transfer learning. These models are used in many fields and image review projects. Like the general classifier projects, in their study for the diagnosis of Covid-19 disease, different parameters were adjusted to obtain more accurate results. For DNN, they considered the batch size as 32, the number of epochs as 50, the 'adam' optimizer, and the learning rate as 0.0001 with weight decay. Once again, regularization was used to reduce overfitting in deep learning models. Then, the flattened layer was added to these pre-trained models, which flattens the input to one dimension. Then, they used a dense layer with 64 neurons, the 'relu' activation function, and the regularize as 0.001, respectively. Before applying the dense layer and after, a dropout is used to prevent overfitting. Finally, three classes are defined with softmax activation function. Therefore, one trainable layer is considered for the ResNet50 model and 62 trainable layers for InceptionV3 [27].

For the purpose of transfer learning, Marques et al. [16] have used the EfficientNetB4 model with a $global - average - pooling2d$ layer added to it, which minimizes overfitting by reducing the total number of parameters. In addition, a sequence of three dense layers with ReLu activation functions have been added. In total, a 30% dropout rate was chosen randomly to avoid overfitting. Finally, an output dense layer includes two output units in the case of binary classification, and three output units for multi-class classification, with softmax activation function, added to create the presented automatic detection system.

Shome et al. [28] used the ViT L-16 model for the initial stage of their model, which is the "Large" variant with a patch size of 16 $\times$ 16. The pre-trained MLP prediction block was removed and a set of untrained feed-forward layers are added to form the custom MLP block. Batch normalization is a neural network layer that allows other layers of the model to learn more independently. It is used to make the output of the previous layers more natural and to scale the activation in the input layer. Learning becomes more efficient when batch normalization is used, and it may also be used as a regularization to avoid overfitting the model [28]. The first dense layer consists of a Gaussian error linear unit (GELU)-based activation with 120 neurons. GELU has been widely used in revolutionary transformer models, and also in vision transformers due to its deterministic nonlinearity that encapsulates a stochastic regularization effect, which leads to a major performance boost in most models with complex transformer architectures. The last dense layer has the softmax activation function, and L2 regularization is used to minimize overfitting as much as possible.

Lee et al. [15] proposed an ensemble method of Faster R-CNN, and achieved the best performance in the COCO object detection challenge in 2016 [28]. To build the ensemble detector, the CNN models to be used must be selected first. They proposed a model selection method based on AP (average accuracy) vectors in order to obtain the complementary benefits of each model. The diversity between models is calculated as the cosine distance between the category-wise AP

**Table 1:** Distribution of Dataset images for training, validation and testing.

| Data Type | Categories | | Sum |
|---|---|---|---|
| | normal | covid | |
| train | 861 | 876 | 1737 |
| valid | 184 | 188 | 372 |
| test | 184 | 188 | 372 |
| Sum | 1229 | 1252 | 2481 |

vectors. If the cosine distance between two models is greater than the threshold, the model with higher mAP is selected and the other is discarded. The regional proposals generated from each selected model are combined into a set of regional proposals.

# 3 Dataset

In this research, a dataset of 2481 lung CT scan images including 1252 infected and 1229 healthy cases are used [3]. From the entire set of dataset images, 70% are used for training, 15% for testing, and 15% as a validation set to avoid overfitting problems. The images are stored in three separate sub-folders named Test, Valid, and Train within the two main folders named Covid and Normal (Table 1).

## *3.1* Data Augmentation

To enhance the training procedure and guarantee the validity of the experiments, all the images were normalized following the pre-trained CNN architecture standards. To accommodate various architectures, all the images were resized to 224 × 224 pixels [1]. Deep learning models largely overfit an insufficient amount of data. So, in order to have an effective network training that can be generalized well, we need a large amount of data. Data augmentation refers to generating new data using existing data [13]. Also, the training dataset was enhanced by performing random horizontal and vertical rotation (0-180 degrees) along with random cropping and resizing (224 × 224). A sample of the generated images is shown in Figure 1.

# 4 Methodology

## *4.1* Model Implementation

In the case of computer vision, transfer learning can save a considerable amount of time required for building an accurate model. Using a learned model obtained while solving other problems can be a more efficient way than building a new ideal model from scratch, as it may involve multiple complexities. This way, the previous learning experience of the model can be transferred to the current model architecture.

In this study, instead of building a model from scratch, pre-trained models and transfer learning were used. First, the EfficientNet-B2 model was implemented with pre-trained weights. Then, considering the number of categories defined in this re-search (0 for infected and 1 for healthy), the number 1 was replaced with the output value of the last layer of the model linear function: Linear(in-features=1408, out-features)=1, bias=True)).

After the model was trained with the appropriate number of epoches and the results were recorded, in the next step, the $ViT-l-32$ model was entered and changed exactly like the previous model (Linear(in-features=1024, out-features=1, bias= True)) and the previous tests were repeated on this model. Finally, to achieve the final goal of building a new model, both previously tested models were implemented in a separate code set, and the output of the last layer of both models was locked by the *Identity()* module. Then a new model was created and within this new model the previous models were called as model modules. The output of these two models was combined using the *torch.cat()* function. Given that the EfficientNet-B2 model delivers 1408 features and the $ViT-l-32$ model delivers 1024 features as output, in the last layer of the new model (linear layer), the number of input and output features of the function are considered 2432 and 1, respectively. The schematic of the designed model is displayed in Figure 2.

The model is implemented by Python version 3 using PyTorch library. PyTorch is a Python supported library for building deep-learning models.

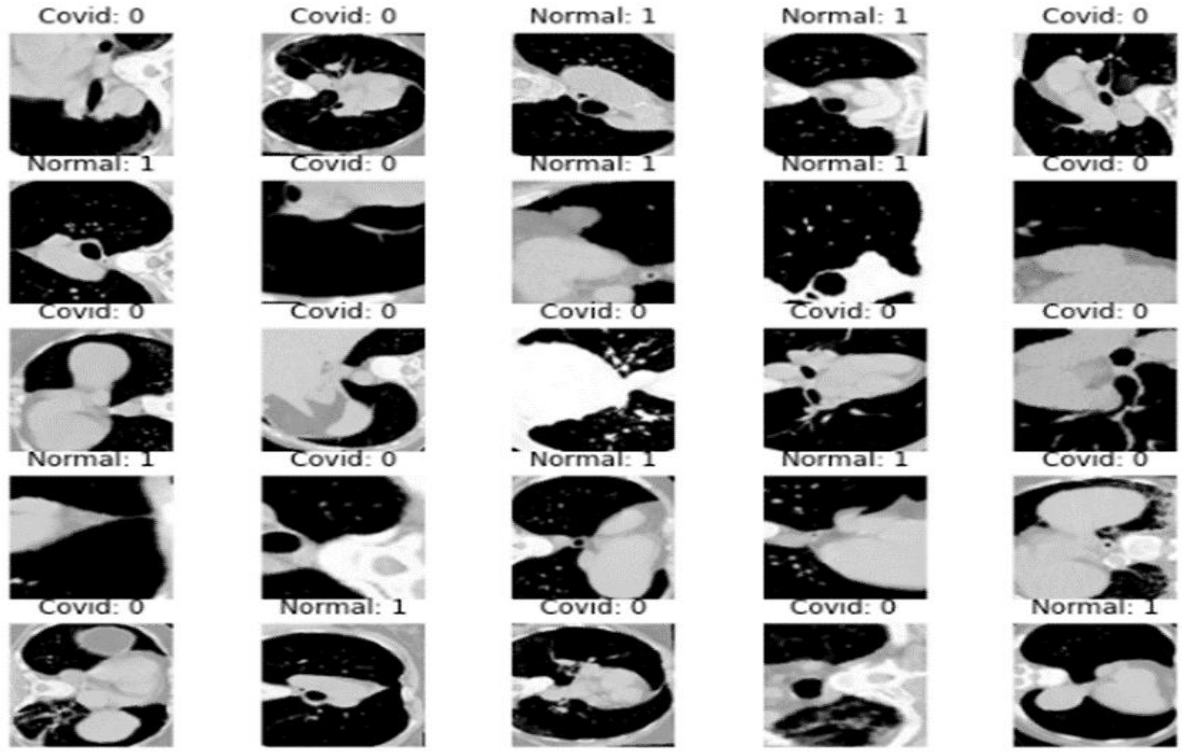


**Figure 1:** Example of CT scan images used to train the proposed model. The images of a healthy lung are labeled 1 and the lung of a Covid-19 patient is labeled 0.

Unlike Keras, PyTorch is flexible and gives the developer more control and has faster GPU acceleration. It also provides an uncomplicated way to switch computations between CPU and GPU.

Moreover, due to the hardware limitations, all the steps of this study were implemented on Google Colaboratory, which is widely known as Google Colab. The Google Colab provides 12 GB of RAM and 358 GB of hard disk space at a time.

## *4.2* Parameter Setting

As mentioned earlier, to train the network, a set of parameters and meta-parameters must be set. In this paper, to compare the performance of the proposed model with two basic transfer learning models, all parameters were set the same for all models. The parameters are as follows:

**Batch Size.** Deep networks are large by design, and the training process requires large amounts of data. Therefore, implementations typically divide the training set into a (potentially large) series of batches of some fixed size. During a training process, each category is processed in the order. However, training samples in a batch are likely to be processed in parallel. On the one hand, groups with small sizes seem desirable due to their tendency to converge in fewer rounds. On the other hand, large batch sizes offer more data parallelism which in turn improves computational efficiency and scalability [4]. Using a moderate batch size helps to achieve a smoother learning process for the model. A batch size of 32 or 64 provides a smooth learning curve in most cases, regardless of dataset size and the number of samples. Even if your hardware environment has large RAM to accommodate a larger batch size, most studies still recommend a size of 32 or 64 [27, 18]. In this study, due to the limitation of the hardware provided by Google Club, the number 32 was considered for the batch size parameter.

**Loss Function.** Deep learning algorithms use the stochastic gradient descent approach for optimization and target learning. To learn a target more accurately and faster, we need to ensure that our mathematical representation of the tar-

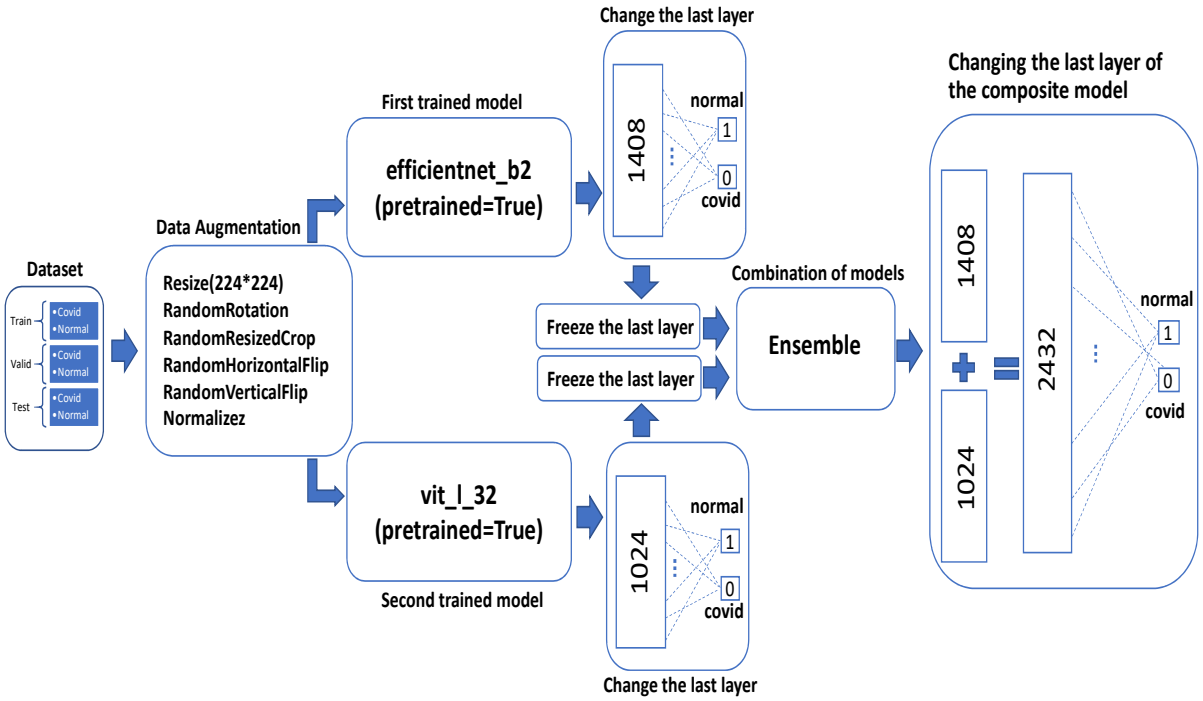


**Figure 2:** Schematic Diagram of the Proposed Model.

gets, also known as loss functions, is able to cover even edge cases. The introduction of loss functions in machine learning has traditional roots, for example, binary cross-entropy is derived from the Bernoulli distribution, and stratified cross-entropy is derived from the polynomial distribution [14]. In this study, the loss function BCEWITHLOGITSLOSS() is used, this loss function actually combines a sigmoid function layer and the BCEloss() loss function in a single class. This version is numerically more stable than using a simple Sigmoid followed by a BCEloss, as we use the log-sum-exp trick for numerical stability by combining the operations in one layer [21].

**Optimization Algorithm.** The role of an optimizer is to update the weight parameters and then minimize the error function or loss function, where the error is the difference between the actual value and the predicted value. To accomplish this, we run many iterations with various weights. Selecting an optimizer for training the network is a challenging task. Deep learning uses an iterative rule. It uses multiple parameters to tune or techniques to analyze data. It is essential to be able to train models quickly to complete the iterative cycle quickly to increase prediction accuracy and speed [10].

In this paper, three optimizers: stochastic gradient descent (SGD), adaptive gradient (AdaGrad), and adaptive moment estimation (Adam) were tested for training, validation, and testing, and finally, by comparing the results obtained in different conditions and states, Adam's optimizer was selected for final testing.

Adam is an optimization algorithm that can be used instead of stochastic gradient descent to update network weights. This technique computes adaptive learning rates for each parameter. Additionally, Adam keeps an exponentially decaying average of past gradients similar to momentum. Adam is a popular algorithm in the field of deep learning because it achieves good results quickly [22].

**Learning Rate.** The learning rate (LR) which is defined in the framework of the optimization algorithm, determines the length of each step or, put simply, the number of updates of the weights in each iteration [18]. The learning rate is certainly

the most important meta-parameter that has a major impact on the performance of the model. In fact, the learning rate indicates the size of the step that gradient descent takes toward the local optimum. A very small learning rate makes the training time long. On the contrary, too large a learning rate makes the model not converge [11]. In the case of Adam's optimizer, the default value is 0.001, which is an excellent choice for most scenarios [18]. In this study, the default value of 0.001 was used.

**Number of Epochs.** Providing a complete set of samples to the network is called an epoch. Therefore, the number of epochs is a measure of the number of times a complete set of input vectors has been presented to the network. A group of samples presented for training purposes is called a training set. In most cases, the training sets are presented in a random order, which varies from epoch to epoch. The relative success or failure of a neural network often depends on the preparation of the input vectors. All weights are simultaneously adjusted at the end of each period based on the error factors in each layer. The weights can be initially assigned random values. However, if some insights are available regarding a particular application, this information can be reflected through an appropriate choice of initial weights. Although the formation of the initial map occurs quickly, it may take many cycles to reach final convergence during the experiment [23]. Sometimes, just increasing the number of epochs to train the model provides better results, although this comes at the cost of increased computation and training time [18].

In this paper, the training of models with the values of 10, 20, 30, 50, and 100 epochs was tested. By increasing the length of the training up to 50 epochs, although more time is spent on training, the validity of the obtained results was higher. However, the results obtained in 100 epochs were not much different from 50 epochs. Finally, using the training length, 50 epochs were considered for conducting the final tests.

### *4.3* Evaluation Criteria

To evaluate the performance of the used models, Confusion matrix, precision, accuracy, recall, and F1 score have been used [25, 17, 31, 24].

The confusion matrix shows the number of correct and incorrect predictions that are summarized with count values and broken down by each class. TP shows the correctly detected positives. TN shows false positives. FN is used for correctly detected negatives, and finally, FP is used for false positives.

Accuracy shows to what degree the model correctly predicts the output. Using the accuracy value, you can immediately see if the model is trained correctly or not and how efficient it is in general (Eq. 4.1).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{4.1}$$

Precision shows how true the result is when the result is predicted as positive. When the value of false positives is high, the precision criterion will be a suitable criterion (Eq. 4.2).

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

When the value of false negatives is high, the Recall criterion would be suitable. When a model has a low recall value, it means that this model considers many infected people as healthy which is not pleasant (Eq. 4.3).

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

F1 score is a suitable measure to evaluate the accuracy of a test. This measure reflects both the Precision and Recall values. In an uneven class distribution, F1 score would be more useful (Eq. 4.4).

$$F1 = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{4.4}$$

## 5 Results and Discussion

In this paper, to avoid overfitting the model, the validation accuracy value of each epoch was compared with the value of the previous epoch and

the highest validation value was saved in a file, then the stored model was called to run the test function and the folder images were validated. Due to saving the best validation accuracy, there is no need to use the early stop function. Also, to reach a fair approximation, each of the base models and the proposed models were trained three times, 50 epochs, and then, based on the above-mentioned evaluation criteria, the average results of each model were compared once separately, and then all together.

In each round, the training was validated using the Train, and Valid folder images and the highest obtained amount was recorded for validation accuracy. To test the performance of the model, the saved model was called and tested with the images of the Test folder, which included 188 lung images of infected people and 184 lung images of healthy people. The results are shown in Tables 2 to 5.

The confusion matrices (a) to (c) (Figure 3) represent the first to third round of the EfficientNet-B2 model, (d) to (f) the $ViT-l-32$ model, and (g) to (i) the proposed model. In the first round (a), 4 out of 188 Covid (i.e., infected people) images, and 13 out of 184 healthy images were incorrectly diagnosed. The incorrect diagnoses were 4 Covid and 3 healthy images for the second round (b), and only 2 Covid and 6 healthy images in the third round. In the first round of the next model (d), 9 Covid and 44 healthy images were classified. The incorrect diagnoses were 90 Covid, and 100 healthy images in the second round (e), and in the third run (f), while only 47 Covid images were correctly recognized, 52 healthy images were incorrectly classified as Covid. In the first round of the proposed model (g), only 13 healthy images have been detected incorrectly, and all the Covid images have been correctly classified. In the second round (h), the number of incorrect diagnoses was 2 for Covid and 5 for healthy images. Finally, in the third round (i), only one covid and 8 healthy images were misdiagnosed.

Figures 4, 5, and 6 display the accuracy of training and validation, and the loss of training and validation obtained during the training process (three times and 50 rounds each time) for EfficientNet-B2, $ViT-l-32$, and the proposed model, respectively. The validation accuracy of each training round is compared with the previous rounds and the highest value is recorded. The average validation accuracy for the EfficientNet-B2 model was 96%, for the $ViT-l-32$ model was 70%, and for the proposed model was 98%, which means that the proposed model has the potential to achieve excellent performance. It is worth mentioning that the experiments show a tendency to lose oscillatory instability, which might be due to the small batch size and the number of datasets [33].

## 6 Conclusion and Future Work

When many patients are hospitalized, doctors and healthcare workers may not be able to take appropriate measures. However, if Covid-19 cases are detected early, isolation from the community and prompt access to possible treatments can greatly reduce the rate of transmission of Covid-19. An appropriate tool is therefore required to detect positive and negative cases of Covid-19 in a more feasible way [2]. The current research suggests a deep learning model built from the combination of two pre-trained transfer learning models, EfficientNet-b2 and $ViT-l-32$, to diagnose Covid-19 infection from CT scan images of people's lungs. The average results obtained after three times training were as follows: validation accuracy of 98.38%, the overall accuracy of 97.39%. Moreover, for the Covid class, the Precision of 95.6%, recall of 99.46%, and 97.48% for the F1 score, and for the healthy class, the Precision of 99.44%, recall of 95.28% and 97.30% for the F1 score were obtained. Implementing this model can significantly improve the speed and accuracy of diagnosis while reducing treatment costs. Moreover, this tool can support radiologists in making quick diagnoses. Meanwhile, the use of this method in hospitals can significantly increase productivity and help save the costs of treating patients for hospitals.

One of the main challenges in this research and in testing the proposed model was finding

**Table 2:** Results of Efficient Net-B2 model training.

| Class | Evaluation Criteria | First | Second | Third | Average |
|---|---|---|---|---|---|
| All | best-valid-accuracy | 0.9811 | 0.9354 | 0.9677 | 0.9614 |
| | test-accuracy | 0.9500 | 0.9500 | 0.9500 | 0.9500 |
| | Accuracy | 0.9543 | 0.9811 | 0.9784 | 0.9712 |
| covid | Precision | 0.9340 | 0.9839 | 0.9687 | 0.9622 |
| | Recall | 0.9787 | 0.9787 | 0.9893 | 0.9822 |
| | F1-score | 0.9558 | 0.9813 | 0.9789 | 0.9720 |
| normal | Precision | 0.9771 | 0.9783 | 0.9888 | 0.9814 |
| | Recall | 0.9293 | 0.9836 | 0.9673 | 0.9600 |
| | F1-score | 0.9526 | 0.9810 | 0.9780 | 0.9705 |

**Table 3:** ViT-l-32 model training results.

| Class | Evaluation Criteria | First | Second | Third | Average |
|---|---|---|---|---|---|
| All | best-valid-accuracy | 0.9543 | 0.6317 | 0.5376 | 0.7078 |
| | test-accuracy | 0.8500 | 0.7500 | 0.4500 | 0.6833 |
| | Accuracy | 0.8575 | 0.4892 | 0.4811 | 0.6092 |
| covid | Precision | 0.9543 | 0.4949 | 0.4747 | 0.5907 |
| | Recall | 0.9521 | 0.5212 | 0.2500 | 0.5744 |
| | F1-score | 0.8710 | 0.5077 | 0.3275 | 0.5687 |
| normal | Precision | 0.9395 | 0.4827 | 0.4835 | 0.6352 |
| | Recall | 0.7608 | 0.4565 | 0.7173 | 0.6448 |
| | F1-score | 0.8408 | 0.4692 | 0.5776 | 0.6292 |

**Table 4:** Training results of the proposed model.

| Class | Evaluation Criteria | First | Second | Third | Average |
|---|---|---|---|---|---|
| All | best-valid-accuracy | 0.9784 | 0.9865 | 0.9865 | 0.9838 |
| | test-accuracy | 0.9500 | 1.00 | 0.9500 | 0.9667 |
| | Accuracy | 0.9650 | 0.9811 | 0.9758 | 0.9739 |
| covid | Precision | 0.9353 | 0.9738 | 0.9589 | 0.9560 |
| | Recall | 1.00 | 0.9893 | 0.9946 | 0.9946 |
| | F1-score | 0.9665 | 0.9815 | 0.9765 | 0.9748 |
| normal | Precision | 1.00 | 0.9889 | 0.9943 | 0.9944 |
| | Recall | 0.9293 | 0.9728 | 0.9565 | 0.9528 |
| | F1-score | 0.9633 | 0.9808 | 0.9750 | 0.9730 |

**Table 5:** Average results for base and proposed model.

| Class | Evaluation Criteria | EffcienNet-B2 | ViT-l-32 | Proposed Model |
|---|---|---|---|---|
| All | best-valid-accuracy | 0.9614 | 0.7078 | 0.9838 |
| | test-accuracy | 0.9500 | 0.6833 | 0.9667 |
| | Accuracy | 0.9712 | 0.6092 | 0.9739 |
| covid | Precision | 0.9622 | 0.5907 | 0.9560 |
| | Recall | 0.9822 | 0.5744 | 0.9946 |
| | F1-score | 0.9720 | 0.5687 | 0.9748 |
| normal | Precision | 0.9714 | 0.6352 | 0.9944 |
| | Recall | 0.9600 | 0.6448 | 0.9528 |
| | F1-score | 0.9705 | 0.6292 | 0.9730 |

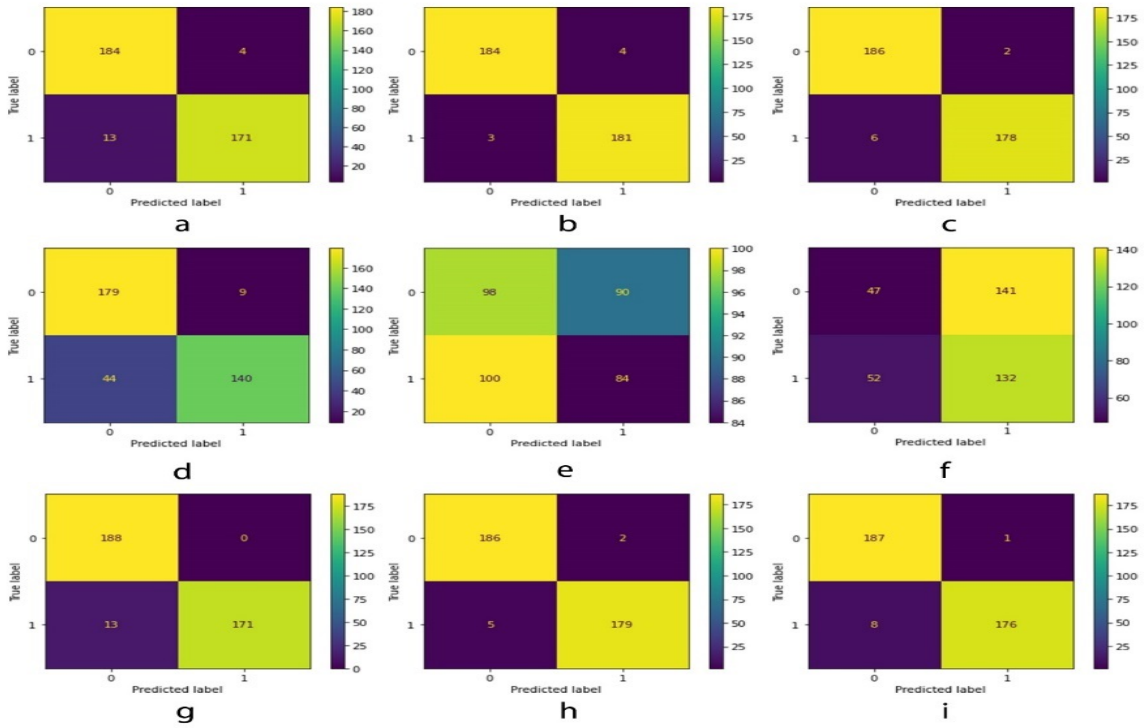


**Figure 3:** Confusion Matrix; a-c Efficient-B2, d-f: ViT-I-32, g-I: the proposed Model.

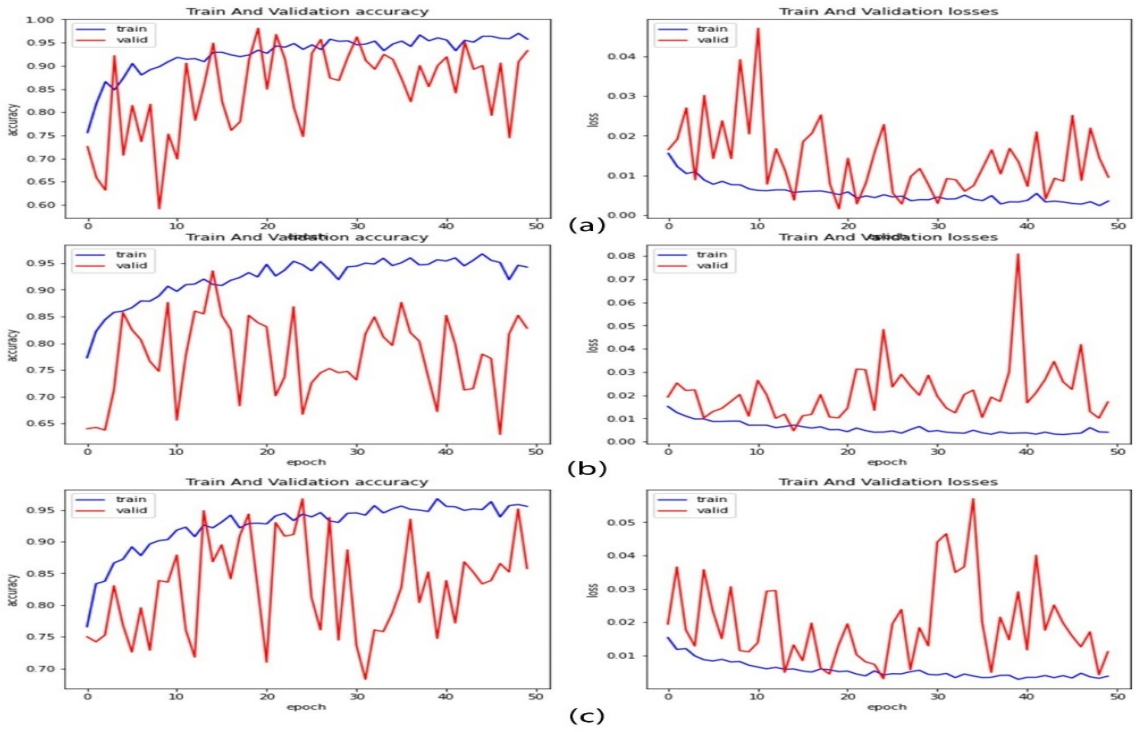


**Figure 4:** Accuracy and loss diagrams related to EfficientNet model a: first run, b: second run and c: third run.

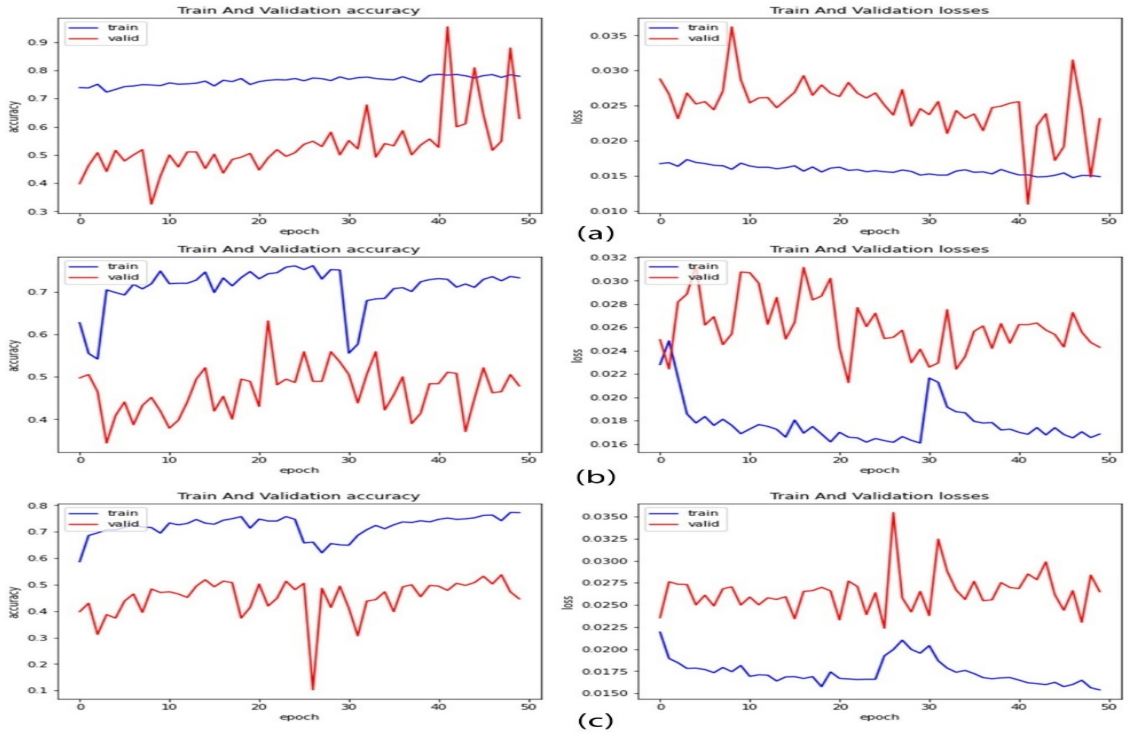


**Figure 5:** Diagrams of accuracy and loss related to ViT-l-32 model a: first run, b: second run and c: third run.

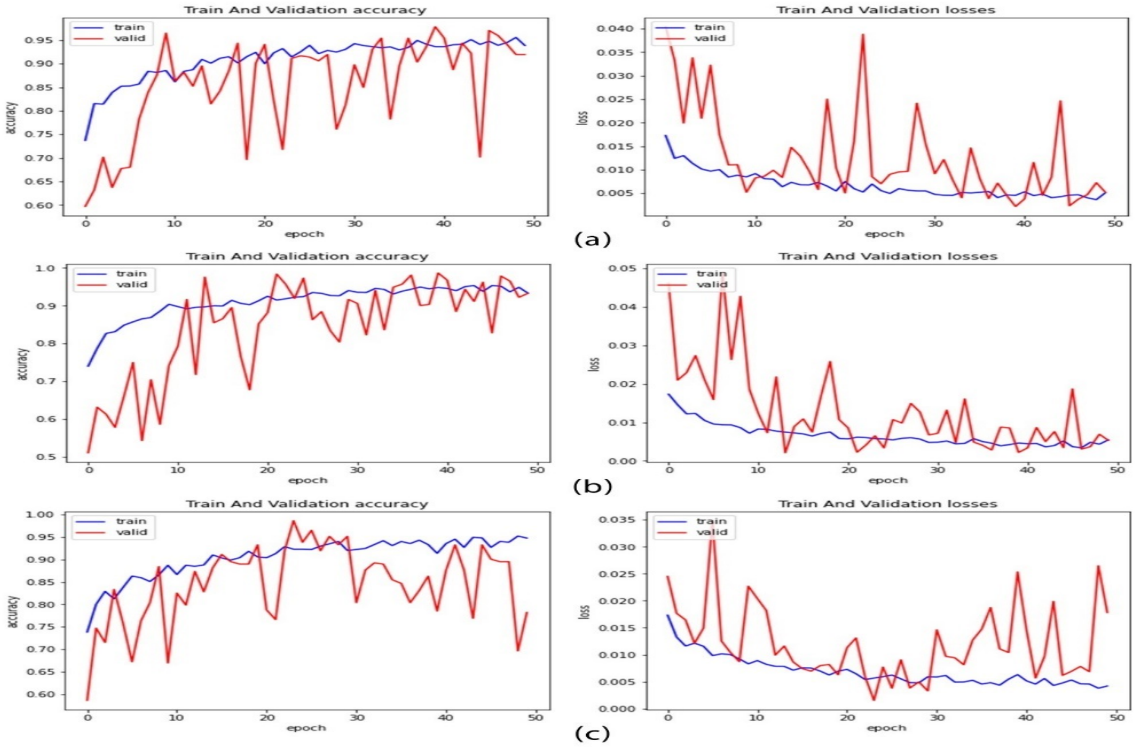


**Figure 6:** Accuracy and loss diagrams related to the proposed model. a: first run, b: second run and c: third run.

numerous high-quality training images. Collecting large amounts of high-quality data can enable the learning of diverse and higher-order features and reduce generalization errors, which can speed up the research and development process. Employing various combinations of transfer learning models can improve the learning performance of the model. The model can also be developed to apply for processing the MRI, X-ray, and ultrasound images in addition to the CT scan images.

## References

[1] A. Alhudhaif, K. Polat, O. Karaman, Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images, Expert *Systems with Applications* 180 (2021) 115-141.

[2] U. Atila, M. Ucar, K. Akyol, E. Ucar, Plant leaf disease classification using EfficientNet deep learning model, *Ecological Informatics* 61 (2021) 101-118.

[3] P. Angelov, E. Soares, Explainable-by-design approach for covid-19 classification via ct-scan, 2020.

[4] A. Devarakonda, M. Naumov, M. Garland, Adabatch: Adaptive batch sizes for training deep neural networks, *arXiv preprint* arXiv: 1712.02029, (2017).

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, An image is worth 16x16 words: Transformers for image recognition at scale, *arXiv preprint* arXiv: 2010, 11929, (2020).

[6] M. Dua, R. Singla, S. Raj, A. Jangra, Deep CNN models-based ensemble approach to driver drowsiness detection, *Neural Computing and Applications* 33 (2021) 3155-3168.

[7] X. Gao, Y. Qian, A. Gao, Covid-vit: Classification of covid-19 from ct chest images based on vision transformer models, *arXiv preprint* arXiv: 2107.01682, (2021).

[8] B. Giri, S. Pandey, R. Shrestha, K. Pokharel, F. S. Ligler, B. B. Neupane, Review of analytical performance of COVID-19 detection methods, *Analytical and bioanalytical chemistry* 413 (2021) 35-48.

[9] K. Habas, C. Nganwuchu, F. Shahzad, Resolution of coronavirus disease 2019 (COVID-19), *Expert review of anti-infective therapy* 18 (2020) 1201-1211.

[10] M. N. Halgamuge, E. Daminda, A. Nirmalathas, Best optimizer selection for predicting bushfire occurrences using deep learning, *Natural Hazards* 103 (2020) 845-860.

[11] M. A. Hannan, D. N. How, M. H. Lipu, SOC estimation of li-ion batteries with learning rate-optimized deep fully convolutional network, *IEEE Transactions on Power Electronics* 36 (2020) 7349-7353.

[12] P. Kanani, M. Padole, Deep learning to detect skin cancer using google colab, *International Journal of Engineering and Advanced Technology Regular Issue* 8 (2019) 2176-2183.

[13] S. H. Khan, A. Sohail, M. M. Zafar, A. Khan, Coronavirus disease analysis using chest X-ray images and a novel deep convolutional neural network, *Photodiagnosis and Photodynamic Therapy* 35 (2021) 102-113.

[14] S. A. Jadon, survey of loss functions for semantic segmentation, *in 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, (2020), IEEE.

[15] J. Lee, S.-K. Lee, S.-I. Yang, An ensemble method of cnn models for object detection, *in 2018 International Conference on Information and Communication Technology Convergence (ICTC)*, (2018), IEEE.

[16] G. Marques, D. Agarwal, I. de la Torre Dez, Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural

network. *Applied soft computing* 96 (2020) 106-111.

[17] N. K. Mishra, P. Singh, S. D. Joshi, Automated detection of COVID-19 from CT scan using convolutional neural network, *Biocybernetics and Biomedical Engineering* 41 (2021) 572-588.

[18] J. Moolayil, Tuning and deploying deep neural networks, *in Learn Keras for Deep Neural Networks*, Springer, 11 (2019) 137-159.

[19] Z. Mousavi, N. Shahini, S. Sheykhivand, S. Mojtahedi, A. Arshadi, COVID-19 detection using chest X-ray images based on a developed deep neural network, *SLAS technology* 27 (2022) 63-75.

[20] S. Nabavi, A. Ejmalian, M. E. Moghaddam, A. A. Abin, A. F. Frangi, M. Mohammadi, H. S. Rad, Medical imaging and computational image analysis in COVID-19 diagnosis: A review, *Computers in Biology and Medicine* 135 (2021) 104-120.

[21] A. Paszke, S. Gross, F. Massa, et al., Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[22] S. Postalcoglu, Performance analysis of different optimizers for deep learning-based image recognition, *International Journal of Pattern Recognition and Artificial Intelligence* 34 (2020) 205-214.

[23] M. C. Purucker, Neural network quarterbacking, *IEEE Potentials* 15 (1996) 9-15.

[24] K. Sabahi, S. Shaykhivand, Z. Mousavi, M. Rajabioun, Recognition Covid-19 cases using deep type-2 fuzzy neural networks based on chest X-ray image, *Computational Intelligence in Electrical Engineering*, (2022).

[25] P. Saha, M. S. Sadi, M. M. Islam, EMC-Net: Automated COVID-19 diagnosis from X-ray images using convolutional neural network and ensemble of machine learning classifiers, *Informatics in medicine unlocked* (2021) 22:100505.

[26] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, K. Wang, Automated Detection of COVID-19 through Convolutional Neural Network using Chest x-ray images. *Plos one* 17 (2022) 17-27.

[27] M. S. Satu, K. Ahammed, M. Z. Abedin, et al., Convolutional neural network model to detect COVID-19 patients utilizing chest X-ray images, *medRxiv* 12 (2021) 45-54.

[28] D. Shome, T. Kar, S. N. Mohanty, P. Tiwari, Covid-transformer: Interpretable covid-19 detection using vision transformer for healthcare, *International Journal of Environmental Research and Public Health* 18 (2021) 110-118.

[29] A. Subasi, J. Kevric, M. Abdullah Canbaz, Epileptic seizure detection using hybrid machine learning methods, *Neural Computing and Applications* 31 (2019) 317-325.

[30] M.Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, *in International conference on machine learning* (2019), PMLR.

[31] S. Thakur, A. Kumar, X-ray and CT-scan-based automated detection and classification of covid-19 using convolutional neural networks (CNN), *Biomedical Signal Processing and Control* 69 (2021) 102-120.

[32] R. Yamashita, M. Nishio, R. K. G. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, *Insights into imaging* 23 (2018) 611-629.

[33] D. Zhang, F. Ren, Y. Li, L. Na, Y. Ma, Pneumonia detection from chest X-ray images based on convolutional neural network, *Electronics* 10 (2021) 15-29.

Mojtaba Ghasemnezhad M.I received the B.Sc. from Islamic Azad University of Ilkhchi, Iran, in 2013 and the M.Sc. from the University College of Daneshvaran, Tabriz, Iran in 2022. His research interests are digital image processing, audio signal processing, deep learning, and related fields.

Habib Izadkhah is an associate professor at the Department of Computer Science, University of Tabriz, Iran. He worked in the industry for a decade as a software engineer before becoming an academic. His research interests include algorithms and graphs, software engineering, and bioinformatics. More recently, he has been working on developing and applying deep learning to a variety of problems, dealing with biomedical images, speech recognition, text understanding, and generative models. He has contributed to various research projects, authored a number of research papers in international conferences, workshops, and journals, and also has written five books, including Source Code Modularization: Theory and Techniques from Springer and Deep Learning in Bioinformatics from Elsevier.