

Covid-19 Recognition by Chest CT and Deep Learning

Lin Yang^{1,*}, Dimas Lima²

¹ School of Computing and Mathematical Sciences, The University of Leicester, University Road, LE1 7RH, United Kingdom

² Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900, Florianópolis, Brazil

Abstract

INTRODUCTION: The current RT-qPCR approach to identify Covid-19 diseases is slow and non-optimal for a large number of candidates.

OBJECTIVES: Several studies have demonstrated that deep learning can help healthcare professionals diagnose Covid-19 patients. The deep learning model proposed in this paper significantly enhanced the accuracy of identifying Covid-19 patients compared to prior approaches.

METHODS: This paper applies transfer learning and deep residual network ResNet152V2 to detect Covid-19 patients with the help of CT scan images. Monte Carlo Cross-Validation has been applied to obtain an accurate and valid result.

RESULTS: The proposed model can identify Covid-19 disease with an overall accuracy of 95.06%, along with an average precision and recall of 97.19% and 92.81%, respectively. It also obtained a specificity of 93.14% and a F1-score of 94.96%.

CONCLUSION: The performance of this proposed ResNet152V2 model is superior to most of the current Covid-19 detection models.

Keywords: Covid-19, deep learning, deep transfer learning, ResNet152V2.

Received on 02 October 2021, accepted on 07 December 2021, published on 07 January 2022

Copyright © 2022 Lin Yang *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.7-1-2022.172812

*Corresponding author. Email: ylin131@gmail.com, dimaslima@ieee.org

1 Introduction

1.1 Background

World Health Organization (WHO) Coronavirus (Covid-19) Dashboard shows that by the end of August 2021, the cumulative number of confirmed cases of Covid-19 Pneumonia worldwide exceeded 200 million, with 454,424,290 deaths [1]. Since the outbreak of the Coronavirus pandemic, many variants of the virus have emerged. In just over a year, WHO has named 11 mutant strains [2]. Among them, four mutant strains - Alpha, Beta, Gamma, and Delta are listed as "strains of concern," as of 31 August, 193, 141, 91, and 170 countries and regions have reported finding the four mutant strains of Alpha, Beta, Gamma, and Delta respectively. Current research suggested

that the longer each pandemic virus has been circulating globally, the greater the likelihood of mutation [3]. Therefore, being able to quickly diagnose Covid-19 patients and accelerate vaccination is critical to the global fight against the epidemic. While Europe and North America have so far received more than 90 doses of the Covid-19 vaccine per 100 people, Africa has received only 7.6 doses per 100 people [4].

Public health experts have warned that inadequate vaccination due to global vaccine inequity allows for the long-term spread of Coronavirus in developing countries [5]. Therefore, under the condition of vaccine shortages, a rapid diagnosis of Covid-19 and priority treatment of Covid-19 patients could reduce coronavirus mortality and aid early recovery.

Currently, RT-qPCR (reverse transcription quantitative polymerase chain reaction) based assays are the standard for diagnosing of Covid-19 [6]. Despite the high sensitivity of this method, it is still too complex and takes several hours to

perform, which is unable to achieve rapid and immediate detection [7]. Therefore, it is essential to develop a diagnostic strategy that is faster and easier to perform than RT-qPCR.

Today, deep learning is a modern technique in medical image analysis research, achieving considerable results in organ segmentation, lesion detection, and classification tasks [8]. It can help reduce the workload of doctors in diagnosis, increase the efficiency of their decision-making. It also improves the accuracy and reliability of their diagnosis and treatment decisions and, more importantly, provides new diagnostic and treatment solutions that doctors would not have been able to provide without the relevant tools [9]. With deep learning, doctors can no longer rely on long-term experience and detailed review to determine a patient's condition. Patients can quickly obtain a more objective result after an examination. CT images can reveal hairy glassy shadows and high tissue density, which are critical in viral pneumonia caused by coronavirus [10]. With the help of rapid diagnosis of deep learning and CT scan images, staff can immediately prioritize treating patients with potential Covid-19 pneumonia.

1.2 Related work

Covid-19 recognition using deep learning methods is a hot subject that has garnered much interest lately. In this area, promising findings by using various Convolutional Neural Networks (CNNs) have been presented and continue to emerge. CT scan images and X-ray images are the two primary kinds of datasets utilized to identify Covid-19. Numerous deep learning models have been developed and successfully used to identify Covid-19. A ResNet 50 model for discriminating Covid-19 from non-Covid19 use chest CT scans was proposed in Ref [11]. They feed the ResNet-based model with the wavelet coefficients of the whole image without clipping any portions of the image. The accuracy of the final result was 92.2%. Researchers in Ref [12] investigated five different deep CNN learning models, AlexNet, Vgg16, Vgg19, GoogleNet, and ResNet50, in identifying Covid patients. The researchers used classical digital augmentation with CGAN to enhance classification performance for all five pre-trained models. The outcome indicated ResNet50 had achieved the best accuracy of 82.91%. Several pre-trained models including Vgg16, Vgg19, InceptionV3, InceptionResNetV2, Xception, DenseNet121, DenseNet169 and DenseNet201 were investigated for detecting Covid-19 patients in Ref [13]. The result showed that DenseNet201 was the highest performance model. It achieved 85% Accuracy. Ref [14] designed COVID-Net, a residual-based CNN model, with a dataset of 13,975 CXR images for identifying Covid-19 patients. The result indicated that COVID-Net obtained an average accuracy of 93.3%. In Ref [15], a light CNN design was proposed based on the SqueezeNet, for the efficient detection of COVID-19 CT images with respect to other pneumonia and healthy CT images. The architecture allowed an accuracy of 85.03%. The researchers in Ref [16]

investigated 15 different pre-trained CNN models to find the most suitable one for this task. VGG19 obtained the highest classification accuracy of 89.3%.

1.3 Motivation

The majority of the above studies utilized a dataset of just a few hundred Covid-19 images, which is inadequate for developing accurate and robust deep learning methods [17-20]. Insufficient data may affect the performance of proposed methods. Further, in most studies, there was a data imbalance problem [21-23], with one class having more images than the other. It affects the accuracy of models. Moreover, most of the studies have not used cross-validation while training deep learning models.

This paper used a balanced CT scan dataset with more than two thousand chest images. The Monte Carlo cross-validation method has been used to validate the deep learning model. A residual network based deep learning model is utilized to distinguish between chest CT scans for Covid-19 and non-Covid-19 symptoms. A comparison analysis is conducted to compare the proposed method with the state-of-the-art approaches. The rest of the paper is organized as follows. Section 2 presents project materials. Section **Error! Reference source not found.** describes the proposed methodology, including CNN, transfer learning, and proposed ResNet152V2 architecture. The experiment results and evaluation are shown in section 4. The conclusion of this paper and the future work is given in section 5.

2 Dataset

In this paper, a CT scan dataset named SARS-CoV-2 is used to conduct the Covid-19 recognition task. The dataset is publicly available at [24]. The CT scan images from this dataset were gathered from 120 individuals of both sexes at Sao Paulo, Brazil, hospitals. It comprises 2481 CT scan images, from which 1252 images are Covid-19 positive, 1229 scans are Covid-19 negative. These CT scans images are accessible in the PNG format and have a spatial resolution of 104 * 119 to 416 * 512 pixels. 1229 images were chosen from a total of 1252 Covid-19 infected data to avoid data imbalance. Fig. 1 illustrates CT images from this dataset. The first row includes CT scans of Covid-19 patients, and the second row shows scan images of patients with a non-Covid-19 diagnosis. Table 1 shows the number of data from each category.

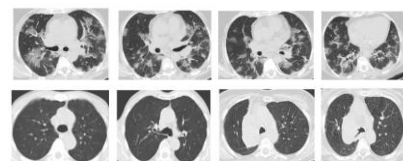


Fig.1 Covid-19 infected (1st row) and Non-Covid-19(not normal) (2nd row) chest CT scans dataset

Table 1. Chest CT scan dataset distributions

Chest CT scan dataset	Classes	Number of samples
	Covid-19	1229
	Non-Covid-19	1229

Image Normalization

The pixel-value representation of images is needed in an image processing procedure. The range of pixel values of an image is in $[0, 255]$. In order to train the deep learning model correctly, The pixel values of images need to be rescaled to the range of $[0, 1]$ to meet the requirement of the inputs. This normalizing method is utilized in both training, validation, and test datasets. Moreover, the images in the dataset are in different sizes. However, the deep learning network needs to be fed with fixed-sized data. Therefore, images are rescaled to 224×224 to satisfy the size requirement of inputs for pre-trained models in this experiment.

Data Augmentation

Methods of data augmentation were used to efficiently expand the number of training samples without the acquiring new images [25]. It is a very effective technique for enhancing the model's generalization capacity. A larger dataset may achieve higher classification accuracy in deep learning than a smaller sample. However, having a large dataset is not always feasible [26-29]. A data augmentation strategy may assist in increasing the quantity of the data. Geometric modifications such as image rotation and flipping were conducted to augment the CT images used in this study. The pixels can be fully reorganized by flipping and rotating images horizontally, vertically, from 0 degrees to 360 degrees and they can also be zoomed in and out.

3 Methodology

In this section, the method applied to the proposed model is explained. The principle of a convolutional neural network for feature extraction and classification is discussed. Moreover, the concept of transfer learning and deep residual network ResNet152V2 are included. In addition, this section also introduces the Monte Carlo Cross-Validation (MCCV) method.

3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) was first introduced by LeCun in 1989 [30]. CNNs are the most widely used deep learning algorithms nowadays. It has been extensively applied in target detection, face recognition, computer vision, and prediction. CNNs are a kind of feedforward neural network capable of extracting features from data via the use of convolutional architecture. Compared with other deep neural networks, CNN has many attractive

characteristics: 1) Each neuron is not connected to all neurons in the preceding layer, but to a subset of them, which effectively reduces parameters and accelerates convergence; 2) A set of connections may share the exact weights, to reduce the number of parameters. 3) A pooling layer down-samples images using image local correlation concept, allowing for data reduction while preserving valuable information [31].

Between the late 1990s and 2000, CNN was more capable of big, various, and complicated problems thanks to numerous enhancements to the CNN architecture and technique. The evolution of CNN covers various topics, including the change of processing units, parameter and hyperparameter optimization methodologies, layer design patterns, and connection. In 2012, CNN-based applications gained popularity following AlexNet's great performance on the dataset in ImageNet. Since then, CNN has been developed with significant advancements, most of which are credited to the rearrangement units of processing and the design of new blocks. Layer visualization was introduced with CNN to aid in comprehending of the feature extraction phases. It moved the tendency away from low spatial resolution feature extraction in deep architectures such as Vgg.

Nowadays, the majority of modern architectures are based on the Vgg topology that is simple and homogeneous. For example, Google's deep learning department pioneered the concept of the split, transform, and merge, coining the term "inception block" to refer to the corresponding block. For the first time, the genesis block introduced the branching notion inside a layer, allowing for the features extraction at various spatial scales. In 2015, with the development of ResNet, the idea of skip connections gained popularity for training deep CNNs. Following that, this notion was adopted by the majority of subsequent networks, including Inception-ResNet, Wide ResNet, and ResNeXt. These new architectural designs, such as ResNeXt, Xception, etc., investigated the learning capacity of CNN on multi-level transformations, with the work of extending the width of the network or by introducing a new notion of cardinality. As a result, the focus of the study changed away from parameter optimization and connection readjustment and toward network architecture improvement. This paradigm shift spawned a slew of novel architectural concepts, including channel boosting, spatial and exploitation focused on spatial and feature maps, and information processing based on attention.

A common CNN architecture consists of alternate convolutional layers and pooling layers, followed by fully connected layers. Along with the feature mapping, regulatory methods such as dropout are added to maximize the performance of CNN. The following describes the function of CNN components design in further detail.

Convolutional layer

The convolutional layer is made up of a collection of convolutional filters for extracting features. The convolutional filter divides the image into small pieces, which are called the receptive field. The receptive field

facilitates the extraction of feature patterns. Fig.2 [32] shows the filters convolve with images by a predefined set of weights that multiply with the pixel values of receptive fields.

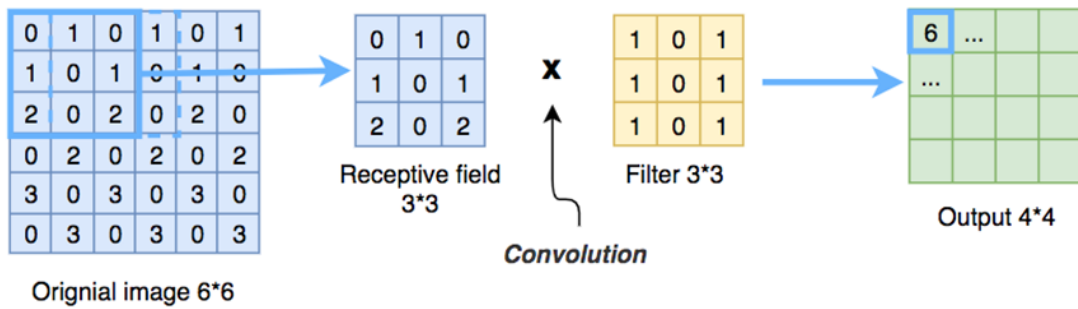


Fig. 2. Principal of convolutional layers

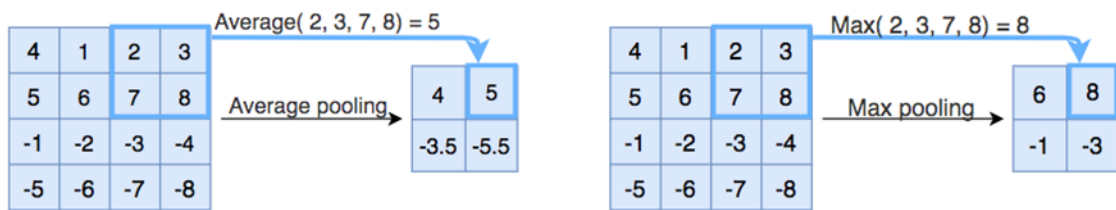


Fig. 3. Average pooling and Max pooling

Pooling

Pooling enables the extraction of a set of features' translational invariance by ignoring the local positional shift of features [33]. The feature patterns that emerge as a result of the convolution operation can occur in various positions throughout the image. After features are extracted, their precise placement becomes irrelevant on the condition that their relative position to others is kept. Pooling is down-sampling. It summarizes the feature data in a certain region of a feature map and outputs the prominent response within this limited region. reducing the dimension of the feature maps regulates the network's amount of parameters and computation loads and also improves overfitting. CNNs employ a variety of pooling formulations, including maximum and average pooling, as indicated in Fig. 3 [33].

Activation function

The activation function acts as a decision-making mechanism that aids in recognition of complicated patterns in data. It adds non-linearity to the network, mapping linear input to non-linear to solve real-world problems. The learning process can be accelerated by selecting a suitable activation function. Sigmoid, softmax, and ReLU are some popular activation functions. Fig. 4 and Fig. 5 are the graphical plot of ReLU and softmax activation functions. Sigmoid and softmax are usually utilized in the last layer of a classification model for binary and multi-class classification. ReLU, on the other hand, is commonly implemented in the convolutional layer, it aid in overcoming the vanishing gradient problem

[34]. The following equations were used to calculate the ReLU and softmax values.

$$f(x) = \max(0, x) \tag{1}$$

$$f_i(\vec{a}) = \frac{e^{a_i}}{\sum_k e^{a_k}}, i = 1, 2, \dots, k \tag{2}$$

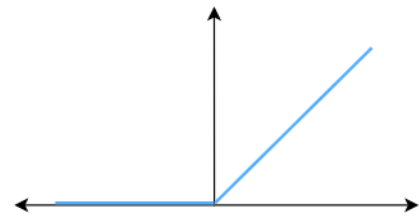


Fig. 4. ReLU plot for negative and positive inputs

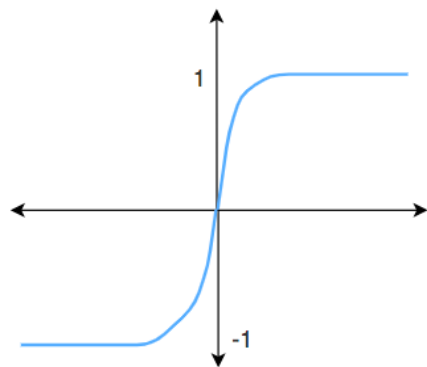
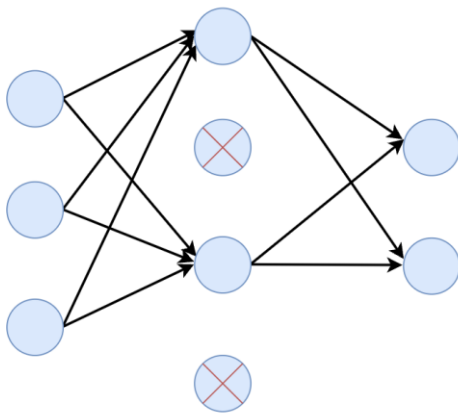


Fig. 5. Softmax line plot representation**Dropout**

Dropout is known to reduce overfitting. Overfitting is one common problem that occurs during model training. It appears when the accuracy of the training dataset is greater than the validation accuracy or when the training dataset generates lower errors than the validation dataset does. Overfitting suggested that the trained model is too complicated for the input dataset. Dropout is one of the regularization approaches that is utilized to reduce the model complex and the training loss. It randomly disregards some neurons in a layer by setting their weights to zero, the red cross in Fig. 6 suggests the neuron is dropped. This methodology results in a simpler network with fewer parameters [35].

**Fig. 6** Dropout in a CNN layer**Fully connected layer**

The fully connected layer is typically utilized for classification at the end of CNN. It receives the output from the feature extraction period and analyzes the output of all preceding layers globally. As a result, it creates a non-linear combination of chosen features for the purpose of classifying data [36].

Optimizer

Optimizers are algorithms or methods that alter a neural network's learning rate and weights in order to minimize losses. Adam (Adaptive Momentum Estimation), gradient descent, and SGD (Stochastic Gradient Descent) are the three main optimizers used in neural networks. Gradient descent is the most straightforward optimizer. It estimates how weight is changed through backpropagation, although it has a couple of drawbacks, including the fact that it takes a long time. SGD makes the training faster than gradient descent because it updates the model parameters more often than gradient descent that updates the parameters just once. It also has some drawbacks, including a considerable variation in model parameters as

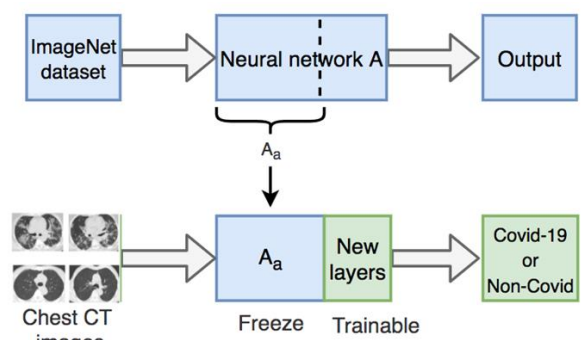
a result of frequent updates. Adam is regarded as the best optimizer since it is faster and more efficient [36].

Loss function

The neural network employs a measure for quantifying the error called the loss function, such that it minimizes the error as far as possible. The loss function is calculated after each iteration, and weights are restructured to minimize the error. The high value of the loss measure would make it more difficult for the model to modify neuronal weights. CrossEntropy is a kind of loss function that is used in binary and multi-class classification. It is calculated by taking the difference of average actual and predicted probability of the defined class. Categorical CrossEntropy is used in this paper as the loss function with one hot labelled targets [37].

3.2 Transfer learning

The majority of deep learning algorithms face three major challenges: lack of data, source data mismatch, and low computation power. Traditionally, deep learning needs a great amount of training data. Moreover, a typical deep learning model requires calculation and update millions of parameters in run-time, which is computationally expensive [37]. Furthermore, when training data is small, more labeled data are added to increase the dataset, even it is from a different source of a dataset. This gives the impact of significantly degrade the performance of deep learning models. Numerous solutions to the first two problems have been proposed, including data argumentation, cloud computing, etc. However, there are drawbacks for each of these solutions, including less efficiency, high cost, or low-security issue. Recently, Transfer learning has become a popular approach as a means of resolving all three difficulties [38].

**Fig. 7** Transfer learning principle

Transfer learning's primary goal is using knowledge gained from previous tasks in another area to solve the target task, rather than starting with a massive amount of data to build from scratch. Thus, transfer learning can handle the most critical issue: the lack of labeled training data. Additionally, the necessary amount of time and computational power needed to train a deep learning

model can be significantly reduced by reusing previously gained knowledge from tasks in other areas. Fig. 7 transfer learning can solve distribution mismatch by combing knowledge from one or more areas [39]. There are three types of transfer learning, transductive, inductive, and unsupervised transfer learning. Inductive learning is used in this paper.

Inductive transfer learning

Inductive transfer learning is characterized as the source task and the target task are distinct, whether the domains are identical. Under this scenario, no matter labeled data is accessible in the source domain, it is typically available in the target domain. In the case of no labeled data available in the source domain, inductive learning is comparable to self-taught learning. There are three types of commonly used learning approaches in inductive transfer learning: instance-based learning, feature-based learning and parameter-based learning [40].

In the target domain, inductive transfer learning is often utilized to build a target model from a limited amount of data that are labeled. The critical aspect of the instance-based learning method is determining which portion of the source data is adaptable in the target domain for training a new model. TrAdaBoost, a well-known transfer learning algorithm in instance-based inductive learning, can repeatedly reweight the instances of the source domain as a means of extracting useful information from the source domain [41].

Feature-based learning algorithms in inductive transfer learning frequently seek to identify domain-invariant features to reduce model error and domain variation. Thus the majority of feature-based inductive learning methods are primarily concerned with effectively extracting shared features across the source and destination domains [42].

Parameter-based learning methods are used in this paper. They are valid on the premise that shared parameters exist across models from the source and target domains. Therefore they are not helpful for situations involving significant domain shifts. Under this scenario, parameter-based learning studies from multi-task learning methods. Except that in inductive transfer learning, the loss function for the target domain task has a higher weight than it does in the source domain compared with the same distributed weights in multi-task learning [43].

3.3 Customized ResNet152V2 model

ResNet is bringing the notion of residual learning into CNNs and developing a successful approach for deep

illustrates the terminology of transfer learning. Moreover,

network training. It altered the CNN architectural race. ResNet152 is developed with 152 layers, which is eight times deeper than Vgg but requires less computing effort.

He et al. [44] demonstrated that ResNet improved by 28% on the well-known COCO image recognition benchmark dataset. ResNet's superior performance on image identification demonstrated that representational depth is critical for a large number of visual recognition asks.

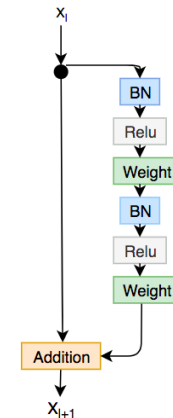


Fig. 8 ResNetV2 architecture

ResNet version 2 showed in Fig. 8 is an improved version of the original ResNet. It uses a pre-activation (put Relu and BN before weight layers) instead of post-activation in the original ResNet to achieve better classification outcomes [45].

The whole architecture of the modified ResNet152V2 model is indicated in Fig. 9. It can be observed that the original ResNet 152V2 is pruned after feature extraction layers and a new average pooling layer, a fully connected layer with 64 filters, and an output layer with softmax activation function are added to classifying Covid-19 images. A dropout layer with 0.5 ratios is also added to prevent overfitting. The original ResNet152V2 model was pre-trained on the ImageNet dataset for feature extraction purposes in this study, which means the model retains its fine-tuned feature extraction weights and hyper-parameters from ImageNet. The feature extraction layers were frozen to avoid information loss during future training. The primary goal of freezing the weights of pre-trained CNNs was to utilize their feature extraction capability.

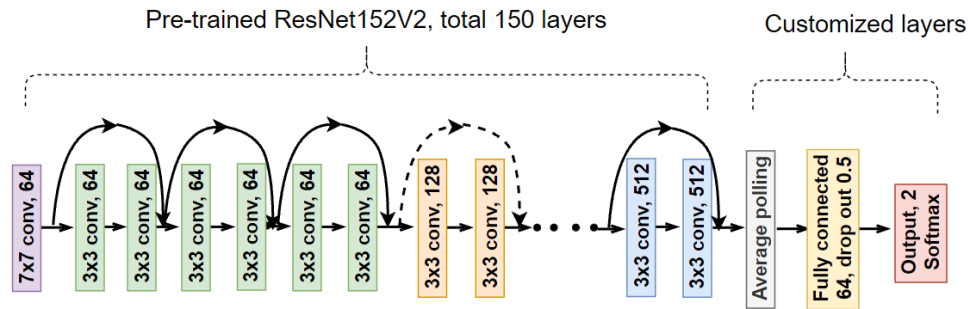


Fig. 9 Architecture of customized ResNet152V2 model

3.4 Monte Carlo Cross-Validation

Monte Carlo Cross-Validation (MCCV) is also known as repeated random subsampling CV. It was first introduced by Picard and Cook [46], the methodology of MCCV is basically to perform the holdout method several times. Each time the dataset is randomly divided into a training set and a validation set so that the model is trained and validated several times individually, the results of these validations are averaged to get a final result that is considered more accurate and valid. This approach provides a better measure of model performance than a single validation (holdout). Compared to k-fold cross-validation, this method offers better control over the

number of times the model is trained and validated and the ratio of the training and validation sets [47].

In this study, the dataset was randomly divided into training, validation, and test subsets with 60%, 20%, and 20%, respectively. The proposed model was trained using training data. After each training epoch, the models were validated against the validation set. Following training, the model was evaluated by the testing dataset, and its performance was quantified using several evaluation metrics. This procedure is repeated five times to make the prediction stable and trustworthy, as shown in Fig. 10, and the results are averaged to obtain a more trustworthy conclusion.

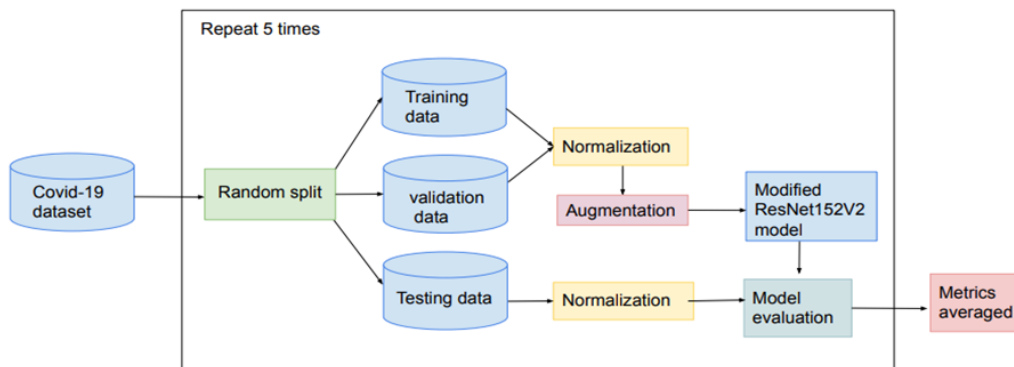


Fig.10 Monte Carlo Cross-Validation with five times repeating

4 Experiment result and evaluation

4.1 Tools

The network was implemented on Jupiter notebook with Python 3.7, Keras 2.3.1, TensorFlow 2.4.0, Scikit-Learn 0.20.4. It was trained on a PC with Intel Xeon CPU E5-2680 v2, 16 GB RAM, and Nvidia RTX2070S GPU.

4.2 Performance metrics

A confusion matrix is a table that summarizes the outcomes of a classification problem's prediction. The amount of right and wrong predictions is summarized and classified using numbers [48]. The predicted and real values are combined in four distinct ways, which are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP means the prediction by the model is positive, and the actual result is positive(true). Similarly, TN indicated that the prediction and the actual outcome are both negative. On the other hand, FP suggested the model predicts a positive result. Still, the real output is negative(false), and FN shows there is a wrong negative prediction; the actual outcome is positive.

The confusion matrix reveals the types of errors that are occurring. For example, As shown in Fig.13 (a), the model predicts 231 True Positives, 14 True Negatives, 9 False Negatives, and 236 False Positives.

		Actual value	
		Positive	Negative
Predicted value	Positive	TP	FP
	Negative	FN	TN

Fig. 11 confusion matrix

Four classification metrics are used to evaluate the trained model via test dataset: accuracy, precision, recall, and F1-score [48].

Accuracy refers to the model's percentage of accurate predictions. It is defined as the ratio of accurate to incorrect predictions as shown in the following equation:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

Precision is a metric that indicates how accurate a model is in classifying a sample as positive. It is defined as the ratio of True Positives to False Positives as indicated in the following equation:

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Recall refers to a model's capacity to identify positive samples. A higher recall value results in a greater number of positive samples being identified. It is computed as the ratio of True Positives to False Negatives as suggested in equation (5):

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

F1-score quantifies precision and sensitivity in a balanced manner. It is the weighted mean of precision and recall as shown in the following equation:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (6)$$

Specificity measures the number of negative samples that have actually been identified as negative.

$$Specificity = \frac{TN}{TN + FP} \quad (7)$$

4.3 Result analysis

After applying MCCV (5 times hold out), it can be observed from Table 2 that the customized pre-trained ResNet152V2 achieved the highest accuracy, precision, F1-score, and specificity with 95.51%, 97.45%, 95.42%, and 93.73% in the fifth run. The fourth run of model training obtained the lowest prediction metrics with an accuracy of 94.29%, recall of 91.43%, F1-score of 94.12%, specificity of 91.89%, and precision of 96.97%. The final averaged accuracy, precision, recall, specificity, and F1-score of modified pre-trained ResNet152V2 were as follows: 95.06%, 97.19%, 92.81%, 93.14%, and 94.96%. All measures are detailed in Table 2. Fig. 12 provides a more intuitive illustration of results for five times of model training.

The confusion matrix results are shown in Fig 13, and it can be observed that the customized pre-trained ResNet152V2 model performed better in identifying non-Covid-19 images than identifying Covid-19 images. On the first run, the model misclassified 23 images, which is slightly better than the results in the second run. The model correctly recognized 466 images; 24 images were misclassified on the second run. And on the third run, this customized model made 6/245 and 18/245 misclassifications in recognizing non-Covid-19 and Covid-19 images, respectively. The model recognized 224 out of 245 Covid-19 images on the fourth run, t, with just 21 images not recognized. The model wrongly classified 22 out of the 490 images in the test set on the last run.

Table 2. Performance metrics for MCCV (five times)

Run	Accuracy	Precision	Recall	F1	Specificity
1	0.9531	0.9625	0.9429	0.9526	0.9440
2	0.9510	0.9784	0.9224	0.9496	0.9266
3	0.9510	0.9742	0.9265	0.9498	0.9300

4	0.9429	0.9697	0.9143	0.9412	0.9189
5	0.9551	0.9745	0.9347	0.9542	0.9373
Average	0.9506	0.9719	0.9281	0.9496	0.9314

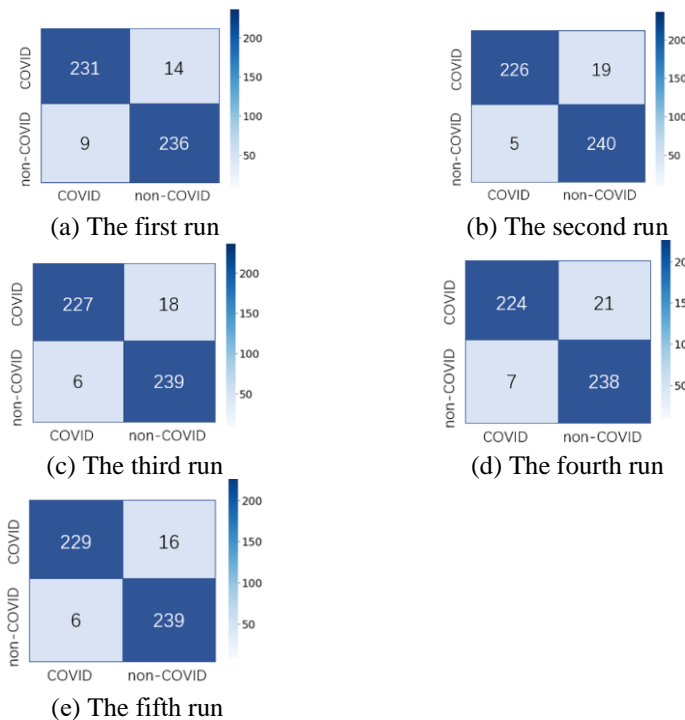


Fig. 12 Evaluation metrics comparison in five times of run

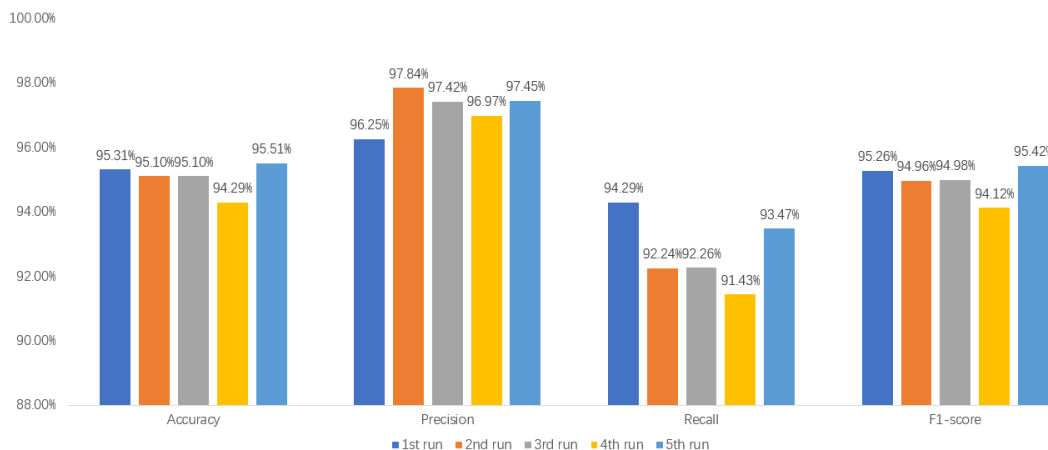


Fig.13 Confusion matrix of five times model evaluation

4.4 Compare with state-of-the-art approaches

Six state-of-the-art approaches were chosen for the comparison study. Table 6 displays the summary of the results of each model. Fig. 14 shows a more intuitive

manner of the performance of all models. It can be found that the model proposed in this paper outperforms the majority of classifiers. In terms of accuracy, it topped the RseNet50 with wavelet coefficients approach by 2.94% and the ResNet50 with CGAN enhancement method by 12.15%. When compared to Do's customized DenseNet201, the accuracy is increased by 10.06%. Additionally, the proposed model improved accuracy by

1.76% to Wang's COVID-Net. And it improves the accuracy of the customized SqueezeNet approach by 10.03%. In comparison to the customized Vgg19 approach, the model proposed in this paper was observed to be 5.76% more accurate. In comparison to the

preceding models, the proposed model employs image augmentation to improve image generalization and produce a more accurate result using the MCCV approach.

Table 3. Comparison with state of the art approaches

Author	Architecture	Accuracy	F1	Recall	Precision
Matsuyama, E[11]	ResNet50+wavelet coefficients	92.2%	91.5%	90.4%	/
Loey. M [12]	ResNet50+augmentation+CGAN	82.91%	/	77.66%	/
Do, C[13]	Customized DenseNet201	85%	/	79%	91%
Wang, L[14]	COVID-Net (Residual based)	93.3%	/	91%	/
Polsinelli, M[15]	Customized SqueezeNet	85.03%	86.20%	87.55%	85.01%
Rahaman, M[16]	Customized Vgg19	89.3%	90%	89%	90%
Proposed	Customized ResNet152V2	95.06%	92.81%	94.96%	97.19%

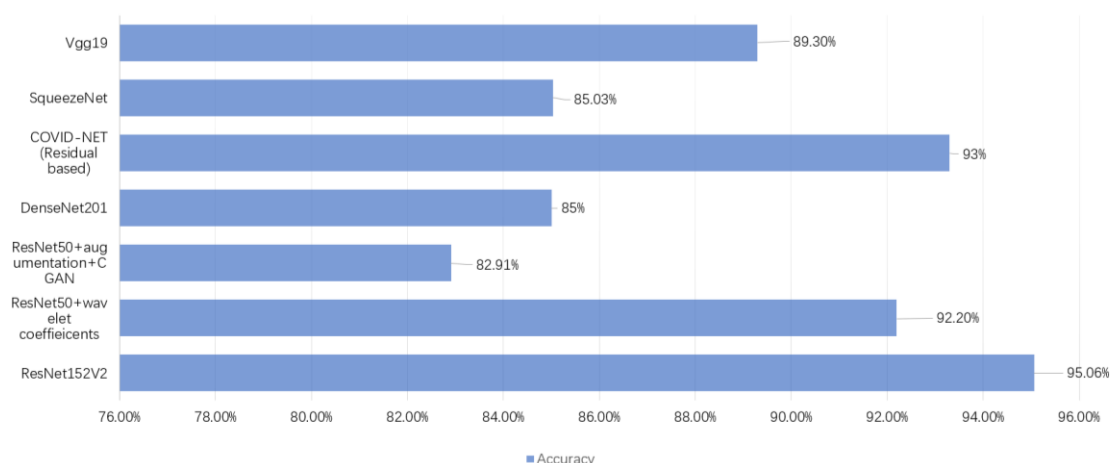


Fig. 14 Accuracy comparison with state-of-the-art approaches

5 Conclusion

This paper applies transfer learning and CNN to build a deep learning model for Covid-19 detection using a dataset containing 2458 CT scans classified as Covid-19 illness or Non-Covid-19 disease. A customized pre-trained ResNet152V2 model was designed and trained on the Covid-19 dataset. Image augmentation and MCCV were conducted to improve the performance of the proposed models. The customized ResNet152V2 model was evaluated using various measures, including accuracy, precision, recall, specificity and F1-score. After five times of repeating training, evaluation, and testing, the results were averaged to achieves an accuracy of 95.06% and sensitivity scores of 97.19% on the test

dataset. The result indicated that the proposed ResNet152V2 outperformed some state-of-the-art transfer learning models in overall accuracy. In future work, other pre-trained models will be selected to perform the Covid-19 detection task. Further, some image enhancement techniques may be used to determine whether there is room for improvement. In this study, the proposed ResNet152V2 network significantly improved Covid-19 detection performance, indicating the possibility of a completely automated and quick diagnosis of Covid-19 using deep learning, which will save time and money since the requirement for specialists in certain circumstances may be reduced.

References

- [1] WHO Coronavirus (COVID-19) Dashboard. 2021-08-20]; Available from: Covid19.who.int/.
- [2] Tracking sars-cov-2 variants. 2021-08-20]; Available from: <https://www.who.int/en/activities/tracking-SARS-CoV-2-variants/>.
- [3] Weekly epidemiological update On covid-19 - 31 AUGUST 2021. 2020-9-01]; Available from: <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---31-august-2021>.
- [4] Covid-19 vaccine tracker: View vaccinations by country. 2021-09-01]; Available from: <https://edition.cnn.com/interactive/2021/health/global-covid-vaccinations/>.
- [5] Schaefer, G.O. Making vaccines available to other countries before offering domestic booster vaccinations. 2021 2021-9-15]; Available from: <https://jamanetwork.com/journals/jama/fullarticle/2783234>.
- [6] Staff. Ask A Scientist, and Kelly says. "Why QPCR Is the Gold Standard FOR COVID-19 Testing.". 2021-8-13]; Available from: www.thermofisher.cn/blog/ask-a-scientist/why-qpcr-is-the-gold-standard-for-Covid-19-testing/.
- [7] T., B.S.A.N., RT-qPCR testing OF SARS-CoV-2: A primer. International journal of molecular sciences.
- [8] Esteva, A., et al., Deep learning-enabled medical computer vision. NPJ Digit Med, 2021. 4(1): p. 5.
- [9] AS, A., The impact of artificial intelligence in medicine on the future role of the physician. PeerJ 7:e7702, 2019.
- [10] Ross, H. What Is a Ct Scan? Procedure, Risks, and Results. 2021-08-21]; Available from: www.healthline.com/health/ct-scan.
- [11] Matsuyama, E., A Deep Learning Interpretable Model for Novel Coronavirus Disease (COVID-19) Screening with Chest CT Images. Journal of Biomedical Science and Engineering, 2020. 13(07): p. 140-152.
- [12] Loey, M., G. Manogaran, and N.E.M. Khalifa, A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images. Neural Comput Appl, 2020: p. 1-13.
- [13] Do, C.M., et al., An approach for recognizing COVID-19 cases using convolutional neural network applied to CT scan images, in Applications of Digital Image Processing XLIII. 2020.
- [14] Wang, L., Z.Q. Lin, and A. Wong, COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep, 2020. 10(1): p. 19549.
- [15] Polsinelli, M., L. Cinque, and G. Placidi, A light CNN for detecting COVID-19 from CT scans of the chest. Pattern Recognit Lett, 2020. 140: p. 95-100.
- [16] Rahaman, M.M., et al., Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches. J Xray Sci Technol, 2020. 28(5): p. 821-839.
- [17] Govindaraj, V., Deep Rank-Based Average Pooling Network for Covid-19 Recognition. Computers, Materials & Continua, 2022. 70: p. 2797-2813.
- [18] Sun, J., MFBCNNC: Momentum factor biogeography convolutional neural network for COVID-19 detection via chest X-ray images. Knowledge-Based Systems, 2021. 232: p. 107494.
- [19] Khan, M.A., Pseudo Zernike Moment and Deep Stacked Sparse Autoencoder for COVID-19 Diagnosis. CMC-Computers, Materials & Continua, 2021. 69(3): p. 3145-3162.
- [20] Zhang, Z. and X. Zhang, MIDCAN: A multiple input deep convolutional attention network for Covid-19 diagnosis based on chest CT and chest X-ray. Pattern Recognition Letters, 2021. 150: p. 8-16.
- [21] Anderson, D., Deep fractional max pooling neural network for COVID-19 recognition. Frontiers in Public Health, 2021. 9.
- [22] Yuan, S.H.Y., Introduction to the Special Issue on Computer Modelling of Transmission, Spread, Control and Diagnosis of COVID-19. Computer Modeling in Engineering & Sciences, 2021. 127(2): p. 385-387.
- [23] Zhu, W., ANC: Attention Network for COVID-19 Explainable Diagnosis Based on Convolutional Block Attention Module. Computer Modeling in Engineering & Sciences, 2021. 127: p. 1037-1058.
- [24] PlamenEduardo. Sars-cov-2 ct-scan dataset. Available from: <https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset>.
- [25] Shorten, C. and T.M. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 2019. 6(1).
- [26] Cheng, X., PSSPNN: PatchShuffle Stochastic Pooling Neural Network for an Explainable Diagnosis of COVID-19 with Multiple-Way Data Augmentation. Computational and Mathematical Methods in Medicine, 2021. 2021.
- [27] Satapathy, S.C. Covid-19 diagnosis via DenseNet and optimization of transfer learning setting. Cognitive Computation, 2021. DOI: 10.1007/s12559-020-09776-8.
- [28] Wang, S.-H., COVID-19 classification by CSHNet with deep fusion using transfer learning and discriminant correlation analysis. Information Fusion, 2021. 68: p. 131-148.
- [29] Wang, S.-H., Covid-19 Classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network. Information Fusion, 2021. 67: p. 208-229.
- [30] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Backpropagation applied to handwritten zip code recognition. Neural Computation, 1989. 1(4): p. 541-551.
- [31] Khan, A., et al., A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, 2020. 53(8): p. 5455-5516.
- [32] Li, Z., et al., A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Trans Neural Netw Learn Syst, 2021. PP.
- [33] Alzubaidi, L., et al., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data, 2021. 8(1): p. 53.
- [34] Wang, W. and Y. Yang, Development of convolutional neural network and its application in image classification: a survey. Optical Engineering, 2019. 58(04).
- [35] Labach, A., Salehinejad, H., & Valaee, S, Survey of dropout methods for deep neural networks. arXiv.org.
- [36] He, Z., Deep Learning in Image Classification: A Survey Report, in 2020 2nd International Conference on Information Technology and Computer Application (ITCA). 2020. p. 174-177.
- [37] Dong, S., P. Wang, and K. Abbas, A survey on deep learning and its applications. Computer Science Review, 2021. 40.
- [38] Pan, S.J., A Survey on Transfer Learning. IEEE

TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2020. 22.

- [39] Zhuang, F., et al., A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 2021. 109(1): p. 43-76.
- [40] Weiss, K., T.M. Khoshgoftaar, and D. Wang, A survey of transfer learning. *Journal of Big Data*, 2016. 3(1).
- [41] Dai, W., et al., Boosting for transfer learning, in *Proceedings of the 24th international conference on Machine learning - ICML '07*. 2007. p. 193-200.
- [42] Soni, A.N., Application and analysis of transfer learning-survey. *International Journal of Scientific Research and Engineering Development*, 2018. 1(2): p. 272-278.
- [43] Day, O. and T.M. Khoshgoftaar, A survey on heterogeneous transfer learning. *Journal of Big Data*, 2017. 4(1).
- [44] He, K., et al., Deep Residual Learning for Image Recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. p. 770-778.
- [45] He, K., et al., Identity Mappings in Deep Residual Networks, in *Computer Vision – ECCV 2016*. 2016. p. 630-645.
- [46] Picard, R.R. and R.D. Cook, Cross-Validation of Regression Models. *Journal of the American Statistical Association*, 1984. 79(387): p. 575-583.
- [47] Xu, Q.-S. and Y.-Z. Liang, Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 2001. 56(1): p. 1-11.
- [48] Novakovic', J.D., et al., Evaluation of classification models in machine learning. *Theory and Applications of Mathematics & Computer Science* 2017.