International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2023, 7 (2), 53-60
DOI: 10.36287/ijmsit.7.2.3
Received: October 27, 2023; Accepted: December 5, 2023

COVID-19 and Non-COVID-19 Classification from Lung CT-Scan Images Using Deep Convolutional Neural Networks

Özlü Dolma[1*]

[1*]*Department of Business Administration, Pamukkale University, Denizli, Türkiye (odolma@pau.edu.tr) (ORCID: 0000-0002-3947-898X)*

Abstract – In this study, three different convolutional neural network (CNN) architectures have been used for SARS-COV-2 infection (COVID-19) detection from lung Computerized Tomography (CT) scan images. The dataset comprises 2481 lung CT-scan images, of which 1252 are positive for COVID-19 infection. First, a simple CNN, LeNet-5, was trained from scratch, which resulted in poor classification performance with an accuracy value of 0.78. Then, to overcome the drawback of the limited availability of data, the convolutional bases of two pre-trained networks, VGG-16 and MobileNet, were leveraged to extract features from the dataset. On top of the feature extraction outputs, new classifiers were trained. When the VGG16 and the MobileNet CNN's convolutional bases were used for feature extraction, accuracy values of 0.974 and 0.984 were obtained, respectively. The findings indicate that using pre-trained CNN models for feature extraction and then training a simpler, fully connected network structure for classification successfully differentiates CT-scan images of patients with COVID-19 infection from the ones without COVID-19 infection.

Keywords *– COVID-19, Convolutional Neural Networks, Feature Extraction, Classification, Deep Learning*

Citation: Dolma, O. (2023). COVID-19 and Non-COVID-19 Classification from Lung CT-Scan Images Using Deep Convolutional Neural Networks. International Journal of Multidisciplinary Studies and Innovative Technologies, 7(2): 53-60.

I. INTRODUCTION

COVID-19 is a contagious and infectious disease caused by the severe acute respiratory syndrome coronavirus 2 (SARS-COV-2). Computed tomography (CT) is a widely used modality for disease or lesion detection, including COVID-19 infection. The diagnosis based on CT-scan images usually depends on a specialist's (e.g., radiologist, physician, clinician) visual evaluation of the images. Computer-aided diagnosis (CADx) can be used as an alternative method for medical diagnosis to reduce the burden on human interpreters by enabling automated analysis of huge amounts of medical images and to support and improve the decision-making process. In CADx, machine learning methods, including deep learning, are utilized to analyze past samples of patient data and develop a model that can be used to predict the disease outcome of a new patient [1].

While CADx systems are already being used as an aid for the detection and interpretation of diseases [2] and are commercially available for lung nodule detection on chest radiography or thoracic CT [3], in numerous studies, convolutional neural network (CNN) based deep learning algorithms were employed to interpret CT-scan or X-ray images automatically and to predict various lung diseases. CNN architectures have also been utilized for the diagnosis of COVID-19. [4] and [5] provided lists of previous studies where different CNNs models were applied to detect COVID-19 infection. They compared the studies in terms of the dataset used, the CNN models or other techniques adopted, and the performance metrics achieved.

In this study, it was aimed to develop a framework to perform the automatic detection of COVID-19 infection in lung CT-scan images using CNN-based deep learning algorithms. For this purpose, given the availability of only a small dataset, a simple LeNet-5 CNN model was first trained, which resulted in poor classification performance but provided a baseline for what can be achieved when a model is trained from scratch. To overcome the drawback of a limited amount of data, pre-trained models, VGG-16 and MobileNet, were used for feature extraction, which were then integrated with a new classifier composed of a small number of layers. Each model's performance was evaluated separately. The classification performances of the proposed methods were assessed and compared with respect to evaluation metrics of accuracy, loss, precision, recall, and the AUC (Area under the curve) of the ROC (Receiver operating characteristic).

The rest of this study is organized as follows: First, an overview of the dataset is provided. Next, the theoretical framework and methodology concerning the research question are presented. Then, the experimental results are presented and interpreted. Finally, the study findings are evaluated, and some directions for future work are proposed.

II. MATERIALS AND METHOD

A. An Overview of the Dataset

The dataset was retrieved from the Kaggle datasets repository. It is titled by Kaggle dataset owners as "SARS-COV-2 CT-Scan Dataset" and released under the CC BY-NC-SA 4.0 license. The dataset contains 1252 positive CT scans for SARS-COV-2 infection (COVID-19) and 1229 CT scans for patients non-infected by SARS-COV-2. The data have been collected from real patients in a hospital in Sao Paolo,

Brazil [6]. A sample of CT-Scan images is provided in Figure 1.

B. LeNet-5 Architecture

Given the limited availability of training images, in the current study, first, a simple CNN architecture LeNet-5, with a small number of parameters, was adopted to train a CNN model from scratch. [7] introduced the LeNet-5 model. It is among the earliest CNN architectures developed for image recognition tasks. [7] trained and tested various versions of LeNet-5 on the MNIST database, which is a dataset of images of handwritten digits. The original model of LeNet-5 is composed of an input layer and seven other subsequent layers. It requires 32x32 pixel image input. There are three convolutional layers, the first two of which are followed by subsampling layers. Convolutional layers have 6, 16, and 120 feature maps, respectively, where filters with sizes 5x5 and stride 1 are used. Subsampling layers perform average pooling using filters of size 2x2 with stride 2. The network concludes with a fully connected layer with 84 units and the output layer. The layers up to the fully connected layer are equipped with scaled hyperbolic tangent activation function. The LeNet-5 architecture adapted for the binary classification problem and an image input size of 32x32 is depicted in Figure 2. The figure was adapted based on the LeNet figure retrieved from [8].

In the current study, for the configuration of the training process of the LeNet-5 model, the optimizer (the way in which the gradient of the loss will be used to update the parameters) was specified as "Adam" (Adaptive moment estimation) with a learning rate of 0.0001, the loss function was specified as "binary cross-entropy," and the metrics were specified as a list of performance metrics evaluated. The number of times the training loop is iterated over the dataset (i.e., number of epochs) was set to 50, and the number of training examples to be used within each epoch in order to compute the gradients for one weight update step (i.e., the batch size used within each epoch) was set to 30.

The LeNet-5 CNN model was previously applied by other researchers to detect patients infected with coronavirus pneumonia using CT-scan images (e.g., [5], [9], [10], [11]). In particular, [10] used the LeNet-5 network to extract features from CT-scan images. Then they applied "eXtreme Gradient Boosting" (XGBoost, [12]) for the classification of the images.

C. Leveraging Pretrained Models

Given the high number of parameters involved in convolutional network structures, the availability of a huge amount of data is relevant to train a model without overfitting. Leveraging a pre-trained model that was previously trained on a large dataset is a commonly adopted approach to deep learning on small image datasets to overcome the problem of overfitting [13]. Moreover, if the model was pre-trained on large and general datasets, the feature hierarchies learned by this model can effectively be used on new problems, even if the classification task of the new problem is entirely different from that of the original dataset [13].

The amount of SARS-COV-2 CT-Scan data in this study is limited to train a CNN model from scratch. To resolve the overfitting problem, VGG-16 [14] and MobileNet [15] convolutional networks pre-trained on ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [16] were used to extract features from the SARS-COV-2 CT-Scan dataset. That is, the dataset was run through the convolutional base of a pre-trained network to extract features, and then a new classifier was trained on top of the feature extraction output. This process was completed separately for these two CNN architectures, and then the evaluation metrics were compared.

For feature extraction with a pre-trained model, only the convolutional bases of the pre-trained networks are used, letting aside the densely connected classifier layers of the original structures. In CNNs, the convolutional base has two types of hidden layers, namely, convolution layers and pooling layers. Convolution layers are used to find instances of small patterns in the image, whereas pooling layers are used to condense them into a smaller summary image. In the early convolution layers of the network, local features, such as edges, colors, and textures of the input image, are identified. In the layers that come later, these low-level features are combined to form higher-level compound features [17]. After feature maps are extracted, they are introduced to a fully connected network used to implement classification. In the densely connected classification part of the network, the representations learned by the classifier will be specific to the set of classes on which the model was trained and will only contain information about the presence probability of a class in the image, thus, densely connected features are not relevant for the new classification problem [13]. Accordingly, to conduct feature extraction, using only the convolutional base of the pre-trained model is more appropriate.

D. VGG16 Architecture

In a study, [14] evaluated very deep convolutional networks (16–19 weight layers) for large-scale image classification on the ILSVRC dataset. They tested six different convolutional network configurations, including VGG16. VGG16 is a CNN architecture in which an image is passed through stacks of convolutional layers where 3x3 filters with stride 1 are used, and the "same" padding is applied. The convolutional layers are followed by five max-pooling layers. Max-pooling is carried out using 2x2 filters with stride 2. Finally, three fully-connected layers are followed by a "softmax" output layer. All hidden layers are equipped with the ReLU activation function. In total, there are 16 weight layers and about 138 million parameters. The VGG16 architecture's convolutional base is presented in Figure 3. The figure was adapted from [18].

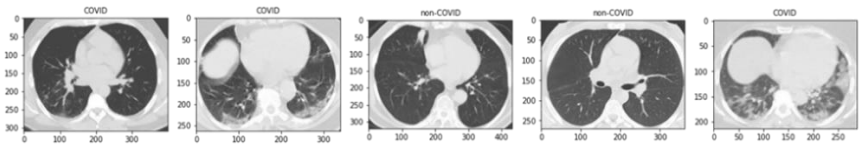

Fig. 1 Samples of CT-scan images

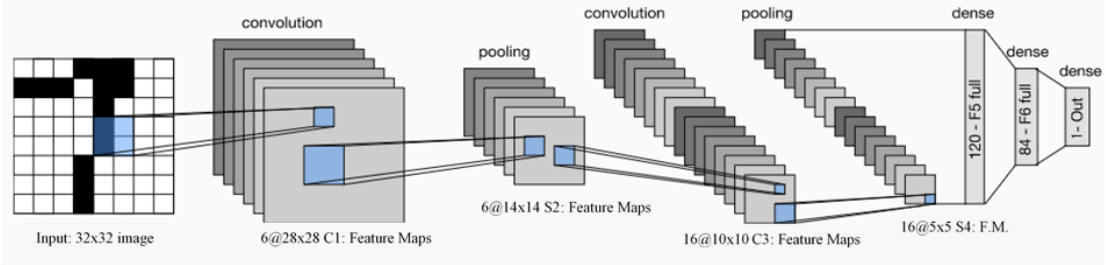

Fig. 2 The LeNet-5 CNN architecture

E. MobileNet Architecture

[15] introduced MobileNets models, replacing the standard convolutions with computationally efficient depthwise separable convolutions. They reduced the computational cost by partitioning the usual one-step operations of filtering and combination (convolution) into two steps using factorized convolutions called depthwise separable convolutions. These convolutions are composed of two layers: depthwise convolutions and pointwise convolutions [15]. Depthwise convolutions apply a single filter per input channel. Then, the pointwise convolution, a simple 1x1 convolution, is used to compute a linear combination of the output of the depthwise convolution [15]. MobileNet uses 3x3 depthwise separable convolutions. Using depthwise separable convolution can reduce accuracy but only at a small rate [15].

When the depthwise and pointwise convolutions are counted separately, the standard MobileNet structure has 28 layers [15]. The first layer is a full convolution layer. All other layers are depthwise separable convolutions followed by batch normalization and ReLU nonlinearity, except for the final fully connected layer, which has no nonlinearity and feeds into a softmax layer for classification [15]. There is a final average pooling that reduces the spatial resolution to 1 before the fully connected layer [15]. Downsampling is conducted with "strided" convolution in the depthwise convolutions and the first layer [15]. In the current study, only the convolutional basis of the structure, layers up to the average pooling layer, were utilized for feature extraction.

[15] have also investigated smaller and faster MobileNets using two model-shrinking hyperparameters; width multiplier and resolution multiplier. These multipliers take values between 0 and 1. The width multiplier is used to make a network thinner uniformly at each layer [15]. The resolution multiplier is applied to the input image, and every layer's internal representation is subsequently reduced at the same rate [15]. They showed that the smaller and less computationally expensive MobileNets that adopt these shrinking hyperparameters trade off a reasonable amount of accuracy to a substantial reduction in computational cost. In the current study, the default values for the multipliers were adopted (i.e., the width multiplier = 1.0 and the resolution multiplier = 1.0).

F. Methodology

Once the feature maps were extracted using the convolutional bases of the VGG16 and MobileNet architectures, the tensor output from the process was flattened and fed into a new classifier, which was trained from scratch. It is a regular feedforward neural network used to implement the classification. This last simple network is composed of a few fully connected layers. The first layer is a dense layer with 256 neurons and Rectified Linear Unit (ReLU) activation function. ReLU is a non-linear activation function with the advantage of faster learning and is widely employed as the default activation function in the convolutional layers of CNNs [19]. This layer is followed by a "dropout" layer with a dropout rate of 20%. Adding dropout is a commonly used regularization technique developed by [20], which involves randomly excluding a portion of the output neurons at every training step [19]. This technique has been proven successful in addressing the overfitting problem [21]. The last layer is a sigmoid output layer that estimates class probabilities for binary classification. Finally, the loss function is the binary cross-entropy, and the optimizer is the Root Mean Square Propagation (RMSprop). RMSprop is a variant of stochastic gradient descent (SGD). The default learning rate for RMSprop is 0.001. In this study, it was set to 0.00001 for MobileNet and to 0.00002 for VGG16. The models were trained with a batchsize of 30, and the networks were trained for 50 epochs. The configuration of the fully connected layers is the same for the two networks.

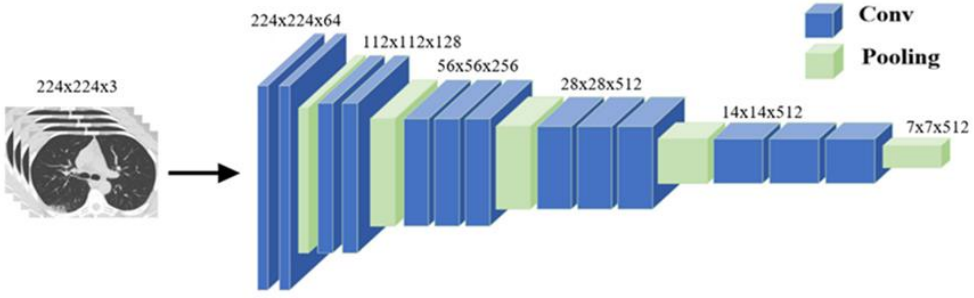

Fig. 3 VGG16 convolutional base

The dataset composed of 2481 lung CT-scan images was split into training (60%), validation (20%), and test (20%) sets. All the images with varied sizes were scaled to a uniform size of 224x224 pixels for training the models backboned by the pre-trained VGG-16 and MobileNet structures. For training the LeNet-5 model, the images were resized to 32x32 since it requires that size of image pixel input. Regarding preprocessing, images represented by pixel values ranging from 0 to 255 were normalized from 0 to 1 by dividing their pixel values by 255.

III.RESULTS

For each of the proposed deep learning models, the validation process was run along with the training process to observe the progress of the model performance across the number of epochs. Next, the evaluation metrics of accuracy, loss, precision, recall, and the AUC of the ROC curve were utilized to compare the classification performances of the proposed models on the test set.

Confusion matrices were utilized to investigate the prediction performance of the models further. In the confusion matrix, the correspondence between the predicted labels and the ground truth is presented in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). In the current study, TP is the number of samples that are correctly classified by the model as being positive for COVID-19 infection that are actually positive, and TN is the number of samples that are correctly classified by the model as being negative for COVID-19 infection that are actually negative. On the other hand, FP is the number of errors in which the model incorrectly indicates the presence of COVID-19 infection when the disease is actually not present, whereas FN is the number of the opposite errors in which the model incorrectly indicates the absence of COVID-19 infection when it is actually present.

The accuracy is the ratio of correct predictions over all predictions. The loss is the binary cross-entropy loss. The ROC curve is created by plotting the true positive rate (Recall) (TPR = TP/(TP+FN)) against the false positive rate (FPR = FP/(FP + TN)) at various classification threshold values ranging from 0 to 1. It summarizes all the confusion matrices produced at each threshold value. AUC is the area under the curve. Precision (TP/(TP+FP)) indicates the probability that a patient with a positive prediction truly has COVID-19 infection.

The accuracy and loss curves for the training and validation of the LeNet-5 model are depicted in Figure 4. Their progress through the epochs did not indicate overfitting after 50 epochs; however, both the training error and the validation error were fairly high, and the curves were close to each other, indicating underfitting.

The model achieved an accuracy of 0.78 and a loss of 0.47 on the training set. Using the validation dataset, it achieved an accuracy of 0.77 and a loss of 0.49. On the training set, the model reached a precision and a recall of 0.77 and 0.80, respectively. On the validation set, the model reached precision and recall values of 0.72 and 0.83, respectively. The model performed poorly both on the training and validation datasets. When the model performance was evaluated on the test set, the LeNet-5 model reached an accuracy of 0.78, a loss of 0.48, a precision of 0.75, a recall of 0.86, and an AUC value of 0.85. The confusion matrix in Figure 5 presents the classification performance of the LeNet-5 model on the test set. The model demonstrated poor performance in classifying images having COVID-19 lesions (i.e., low recall). The model made more mistakes in classifying CT-scan images of patients not infected by COVID-19 (i.e., high FPR). In the case of pandemics, however, the consequences of misclassifying COVID-19 patients may be worse; thus, having a high recall value may be more prominent.

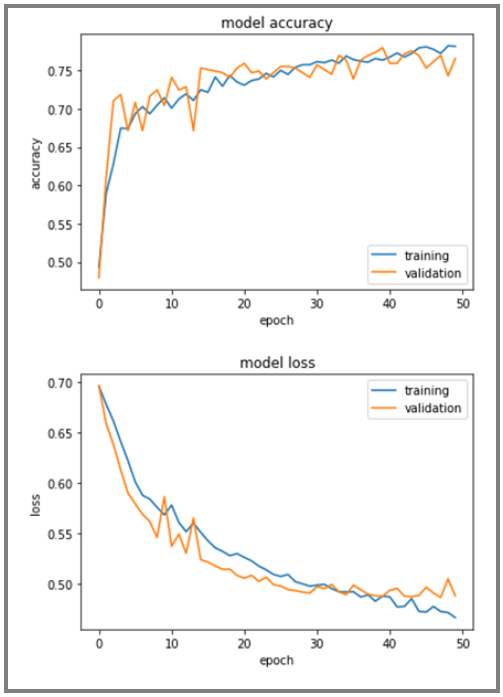

Fig. 4 The training and validation accuracy and loss curves for the LeNet-5 model

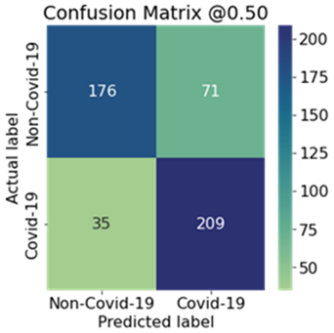

Fig. 5 Confusion matrix (LeNet-5 model)

This unsatisfactory result might have been obtained either due to the fact that the LeNet-5 model was too simple to distinguish between the images with lesions of COVID-19 infection from the ones without such lesions or the features did not provide enough information to make good predictions. Two remedies to overcome this problem might be: (i) selecting a more powerful model with more parameters and (ii) feeding better features to the learning algorithm [19]. In fact, when the features were extracted using the pre-trained models and then fed into a new classifier composed of a few fully connected layers, significantly better prediction performances were obtained. The accuracy and loss curves for the training and validation of the model backboned by the VGG-16 model are depicted in Figure 6.

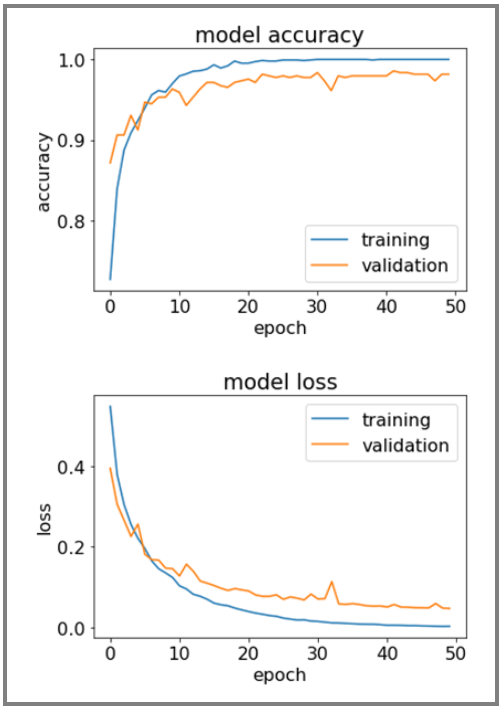

Fig. 6 Training and validation analysis over 50 epochs using VGG16 for feature extraction

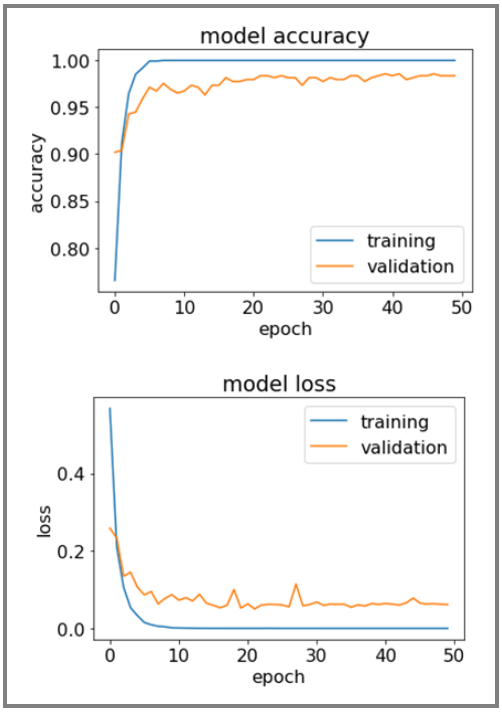

Fig. 7 Training and validation analysis over 50 epochs using MobileNet for feature extraction

As shown, the training accuracy steadily increases and reaches 1 after 30 epochs. The validation accuracy has a similar trend and peaks at 0.95 before 10 epochs and reaches 0.98 after 50 epochs. On the other hand, while the training loss continues to decrease until 50 epochs and reaches zero, the validation loss stalls after 40 epochs at about 0.05.

The accuracy and loss curves for the training and validation of the model backboned by the MobileNet architecture are shown in Figure 7. The training accuracy achieves 1 earlier than was the case for the model with feature extraction based on VGG16. Here, as well, the validation accuracy has a similar trend as the training accuracy and reaches an accuracy level of 0.98 after almost 17 epochs and stays at that level for the subsequent epochs. So, 0.98 was the highest accuracy level that could be achieved on the validation set. While the training loss decreases swiftly to zero after 12 epochs, the validation loss continues to decrease to 0.06 until 20 epochs and stays at that level for the remaining epochs.

Figure 8 compares the proposed deep learning models' training and validation analysis progresses. As seen, learning a simple, fully connected network for classification preceded by a more sophisticated pre-trained CNN architecture for feature extraction outperforms learning a standard CNN with a simple structure with a limited amount of data available. Although overall, the classification performances were comparable, the model employing the MobileNet CNN's convolutional base for feature extraction demonstrated slightly better performance in achieving the highest accuracy and the lowest loss levels with less number of epochs.

Table 1 compares the classification performance of the models employing pre-trained CNNs of VGG-16 and MobileNet for feature extraction on the test set. When the VGG16 CNN's convolutional base is used for feature extraction, an accuracy of 0.974 was obtained. The loss was 0.087. When the feature extraction process was conducted by employing the MobileNet CNN's convolutional base, a higher accuracy of 0.984 was achieved. The loss was 0.072.

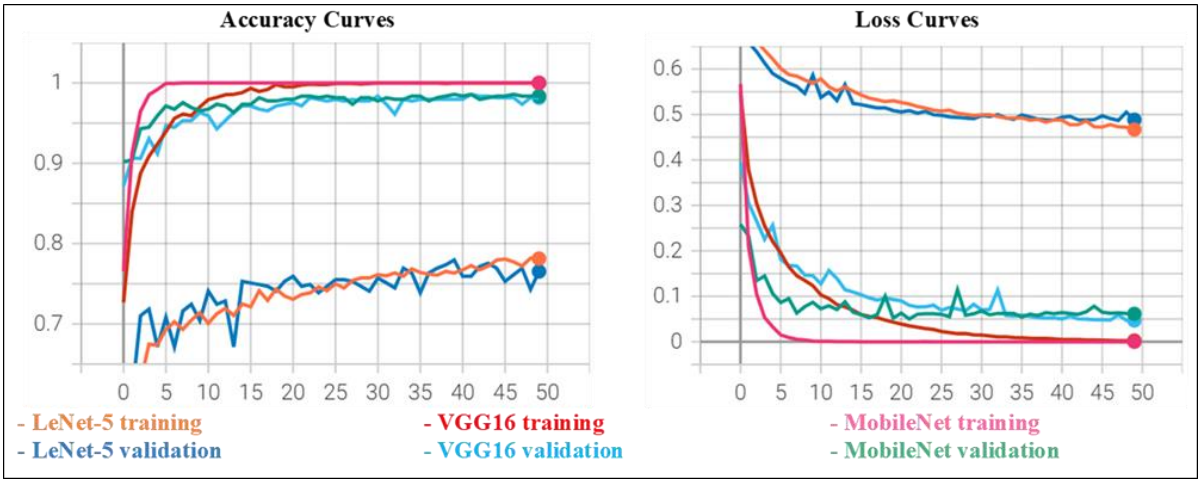

Fig. 8 Comparison of the training and validation analysis results of the three models

Table 1. Comparison of the results obtained using pre-trained CNNs for feature extraction

CNN	Accuracy	Loss	Recall	Precision	AUC
VGG16	0.974	0.087	0.971	0.975	0.995
MobileNet	0.984	0.072	0.988	0.980	0.995

With respect to the proportion of COVID-19 samples that were correctly classified, the network structure backboned by the MobileNet convolutional base had a slightly better performance (RecallMobilNet = 0.988) than the one backboned by the VGG16 convolutional base (RecallVGG16 = 0.971). The precision values indicate that both models were successful in correctly classifying positive results. AUC values were also high and comparable for both models. Overall, the findings indicate that the methodology of using a complex pre-trained CNN for feature extraction and then a simpler fully connected network structure for classification is useful for differentiating CT-scan images with COVID-19 from Non-COVID-19. Both models were more accurate than the simple LeNet-5.

The confusion matrices in Figure 9 and Figure 10 present in detail the classification performance of these models on the test set. Overall, leveraging pre-trained models for feature extraction significantly reduced the number of FPs and FNs compared to the case when the LeNet-5 model is trained from scratch and used to make binary classification. The model backboned by the MobilNet convolutional base had a slightly better performance in minimizing the classification errors. In COVID-19 detection, it may be essential to correctly classify each infected sample to reduce the risk of an outbreak. When the MobileNet convolutional base was utilized for feature extraction, a slightly higher TPR was achieved. Nevertheless, both models' prediction performance was comparable.

Figure 11 compares the classification performance results on the test set using ROC and AUC for the three methods employed. The ROC curve displays concurrently TPR (recall) and FPR (the fraction of Non-COVID-19 patients that were incorrectly classified as infected by COVID-19) for all possible threshold values. It is ideal to have a ROC curve that hugs the top left corner, indicating a high TPR and a low FPR [17]. The area under the (ROC) curve (AUC) quantifies the overall performance of a classifier, summarized over all possible thresholds, and the larger the AUC, the better the classifier [17]. Figure 11 reveals that the classification performance of the models on the test set employing pre-trained CNNs of VGG-16 and MobileNet architectures for feature extraction is significantly better than that of the LeNet-5 model trained from scratch.

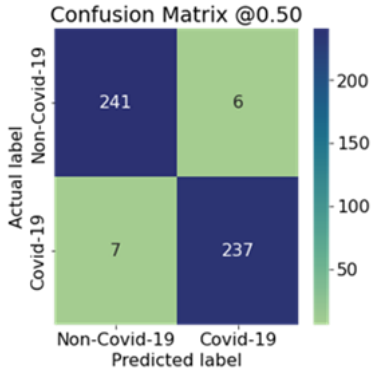

Fig. 9 VGG16 - Confusion matrix

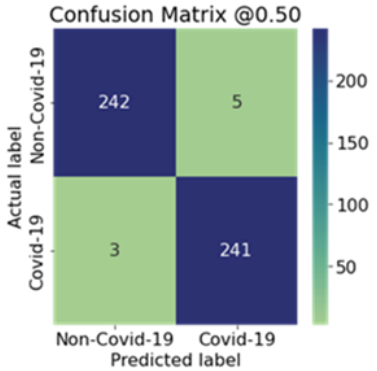

Fig. 10 MobileNet - Confusion matrix

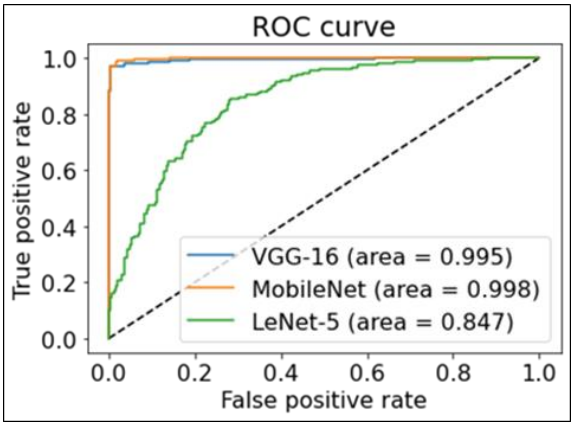

Fig. 11 Comparison of ROC curves and AUC of the three methods

IV. DISCUSSION

Due to the lack of a sufficient amount of data relevant to train a CNN model with numerous parameters successfully from scratch, the LeNet-5 CNN model with a relatively simple structure was trained first, which, however, showed poor classification performance when tested on unseen data. This result was in line with the findings of previous studies. For instance, [11] employed the LeNet-5 architecture to implement a CNN-based model for COVID-19 diagnosis using lung CT-scan images. Unlike the current study, they applied image augmentation to enlarge the dataset. Their findings were not promising, as well. They could reach an accuracy level of 86.06%, a precision of 85%, a recall of 89%, and an AUC of 0.86 for COVID-19 detection.

Similarly, [18] findings revealed that the relatively simple structure of the LeNet-5, having only five layers, rendered it less capable of extracting features, whereas the more complex structure of the VGG16 improved the feature extraction ability of the model and thus had a good classification detection ability. [10] obtained more successful performance results when they used the LeNet-5 network to extract features from CT-scan images and then applied "eXtreme Gradient Boosting" (XGBoost, [12]) for the classification of the images. With this methodology, they could achieve a rate of 0.95 for all the evaluation metrics of accuracy, recall, precision, and AUC.

This study further indicated that the methodology of using pre-trained CNNs to extract features and then training a new classifier on top of it demonstrates an effective performance in categorizing lung CT-scan images into COVID-19 and Non-COVID-19. This methodology has been adopted previously in other studies, in which the researchers either compared the classification performances of various CNN models backboned by different pre-trained CNN convolutional bases or developed their own framework based on feature extraction using pre-trained CNNs. The findings of the current study are in line with the previous research results involving a similar methodology used to solve the same binary classification problem. For instance, [22] employed pre-trained weights from several state-of-the-art CNN architectures using the ImageNet dataset, including VGG16 and another version of MobileNet called MobileNetV2, in the task of COVID-19 classification based on CT-scan images. [22] obtained more promising results using VGG16 with respect to accuracy and AUC metrics (0.84 and 0.93, respectively). [23] conducted a similar study on the same dataset analyzed in the current study (SARS-COV-2 CT-Scan Dataset), where they utilized the pre-trained models of MobileNet and VGG16 as the backbone of their deep learning framework and achieved accuracies of 0.95 and 0.94, respectively. In a similar vein, [24] analyzed the lung CT-scan images of 5191 patients, 3820 of which were COVID-19 infected, to develop a deep learning based system for the automated diagnosis of COVID-19. In that study, they developed their own model called COVIDnet and compared its performance against that of the pre-trained CNN models of MobileNet and VGG16. With 200 epochs, they could achieve an accuracy of 0.94 using MobileNet and 0.97 with VGG16.

V. CONCLUSION

This study is a preliminary work involving image classification. In a future study, the extracted features of the network can be visualized via heat maps using the Gradient-weighted Class Activation Mapping (Grad-CAM) algorithm [25] in order to have a better understanding of which parts of the CT scan images were identified as a lesion of an infection, which led to a positive COVID-19 classification. Visualizing heat maps can also be useful to locate the infection areas in the CT scan images and compare the COVID-19 with Non-COVID-19 images to understand the CNN's classification performance further. Moreover, in another study, on a dataset of lung CT scan images with radiology annotated COVID-19 lesions, region-based convolutional neural networks (R-CNNs), such as Mask-RCNN [26], can be implemented and the detection accuracy of the CNN models can be further investigated.

ACKNOWLEDGMENT

Statement of Research and Publication Ethics

The author declares that this study complies with Research and Publication Ethics.

REFERENCES

[1] H. P. Chan, L. M. Hadjiiski, and R. K. Samala, "Computer-aided diagnosis in the era of deep learning," *Medical Physics*, 47(5), pp. e218-e227, 2020.

[2] N. Petrick, B. Sahiner, S.G. Armato III, A. Bert, L. Correale, S. Delsanto, M.T. Freedman, D. Fryd, D. Gur, L. Hadjiiski, Z. Huo, Y. Jiang, L. Morra, S. Paquerault, V. Raykar, F. Samuelson, R.M. Summers, G. Tourassi, H. Yoshida, B. Zheng, C. Zhou, and H. P. Chan, "Evaluation of Computer-Aided Detection and Diagnosis Systems," *Medical Physics*, 40(8), 2013.

[3] K. Chockley and E. Emanuel, "The end of radiology? Three threats to the future practice of radiology," *Journal of the American College of Radiology: JACR*, 13(12), pp. 1415-1420, 2016.

[4] A. Panthakkan, S. M. Anzar., S. Al-Mansoori, and H. Al-Ahmad, "A novel DeepNet model for the efficient detection of COVID-19 for symptomatic patients," *Biomedical Signal Processing and Control*, 68, pp. 1-10, 2021.

[5] M.C. Younis, "Evaluation of deep learning approaches for identification of different Corona-Virus species and time series prediction," *Computerized Medical Imaging and Graphics*, 90, pp. 1-13, 2021.

[6] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT Scans for SARS-CoV-2 identification," *MedRxiv*, 2020.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *in Proc. IEEE*, 1998, 86(11), p. 2278.

[8] (2023) Dive into Deep Learning website. [Online]. Available: https://d2l.ai/chapter_convolutional-neural-networks/lenet.html

[9] G. Hong, X. Chen, J. Chen, M. Zhang, Y. Ren, and X. Zhang, "A multi-scale gated multi-head attention depthwise separable CNN model for recognizing COVID-19," *Scientific Reports*, 11(1), pp. 1-13, 2021.

[10] E.D. Carvalho, E.D. Carvalho, A.O. de Carvalho Filho, F.H.D. De Araújo, and R.D.A.L. Rabêlo, "Diagnosis of COVID-19 in CT image using CNN and XGBoost," *in Proc. IEEE Symposium on Computers and Communications (ISCC)*, 2020.

[11] M. R. Islam and A. Matin, "Detection of COVID 19 from CT image by the novel LeNet-5 CNN architecture," *in Proc. 23rd International Conference on Computer and Information Technology (ICCIT)*, 2020.

[12] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *in Proc. KDD '16: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, p. 785.

[13] F. Chollet, *Deep Learning with Python*, 2nd ed., Manning Publications, 2021.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. *3rd International Conference on Learning Representations (ICLR)*, 2015.

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ArXiv:1704.04861*, 2017.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, 115(3), pp. 211-252, 2015.

[17] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, 2nd ed., Springer, 2021.

[18] J., Sun, X., Li, C., Tang, S. H., Wang, and Y. D. Zhang, "MFBCNNC: Momentum factor biogeography convolutional neural network for COVID-19 detection via Chest X-ray images," *Knowledge-Based Systems*, 232, pp. 1-21, 2021.

[19] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed., O'Reilly Media, Inc., 2019.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors," *ArXiv:1207.0580*, 2012.

[21] W. Zhu, W. Yeh, J. Chen, D. Chen, A. Li, and Y. Lin, "Evolutionary convolutional neural networks using ABC," *in Proc. 11th International Conference on Machine Learning and Computing (ICMLC)*, 2019, p. 156.

[22] M.T. Dang, *A Survey on Transfer Learning for COVID-19 Medical Imaging Diagnosis*. In: Pan, JS., Li, J., Ryu, K.H., Meng, Z., Klasnja-Milicevic, A. (Eds.), Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies. Springer, 2021, vol 212.

[23] A. Halder and B. Datta, "COVID-19 detection from lung Ct-Scan images using transfer learning approach," *Machine Learning: Science and Technology*, 2(4), 2021.

[24] K. S. Briskline, D. Murugan, and A. Petchiammal, "COVIDnet: An efficient deep learning model for COVID-19 diagnosis on chest CT images," *International Journal of Advanced Computer Science and Applications*, 13(11), pp. 832-839, 2022.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D.,Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, 128 (2), pp. 336-359, 2019.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *in Proc. IEEE International Conference on Computer Vision*, 2017, pp. 2961.