Pooria Dehghan Chenari

Abstract

Corona virus disease or covid-19 is one of the most widespread and dangerous diseases that appeared during 2019 and took the lives of millions of people. The main involvement of covid-19 disease is on the lung tissues. Diagnosis of covid-19 disease on lung tissue is mostly possible using chest radiology images and chest images using CT scan. In this article, a wavelet transformation was performed using chest CT scan images, and then the necessary implementations were performed on these images using machine learning. The experiments were performed on two popular VGG16 and ResNet34 models and the empirical results showed that pruned ResNet34 model achieved 95.47% accuracy, 0.9216 sensitivity, 0.9567 F-score, and 0.9942 specificity with 41.96% fewer FLOPs and 20.64% fewer weight parameters on the SARS-CoV-2 CT-scan dataset. The results of our experiments showed that the proposed method significantly reduces the run-time resource requirements of the computationally intensive models and makes them ready to be utilized on the point-of-care devices.

Keywords: Covid-19, Deep learning, CT-scan images, Accuracy, X-ray images

1.Introduction

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by the coronavirus SARS-CoV-2. The first known case was identified in Wuhan, China, in December 2019. Most scientists believe the SARS-CoV-2 virus entered into human populations through natural zoonosis, similar to the SARS-CoV-1 and MERS-CoV outbreaks, and consistent with other pandemics in human history. Social and environmental factors including climate change, natural ecosystem destruction and wildlife trade increased the likelihood of such zoonotic spillover. The disease quickly spread worldwide, resulting in the COVID-19 pandemic.

The symptoms of COVID-19 are variable but often include fever, fatigue, cough, breathing difficulties, loss of smell, and loss of taste. Symptoms may begin one to fourteen days after exposure to the virus. At least a third of people who are infected do not develop noticeable symptoms. Of those who develop symptoms noticeable enough to be classified as patients, most (81%) develop mild to moderate symptoms (up to mild pneumonia), while 14% develop severe symptoms (dyspnea, hypoxia, or more than 50% lung involvement on imaging), and 5% develop critical symptoms (respiratory failure, shock, or multiorgan dysfunction). Older people are at a higher risk of developing severe symptoms. Some complications result in death. Some people continue to experience a range of effects (long COVID) for months or years after infection, and damage to organs has been observed. Multi-year studies are underway to further investigate the long-term effects of the disease.

COVID-19 transmission occurs when infectious particles are breathed in or come into contact with the eyes, nose, or mouth. The risk is highest when people are in close proximity, but small airborne particles containing the virus can remain suspended in the air and travel over longer distances, particularly indoors. Transmission can also occur when people touch their eyes, nose or mouth after touching surfaces or objects that have been contaminated by the virus. People remain contagious for up to 20 days and can spread the virus even if they do not develop symptoms.

1.1Symptoms and signs

The symptoms of COVID-19 are variable depending on the type of variant contracted, ranging from mild symptoms to a potentially fatal illness. Common symptoms

include coughing, fever, loss of smell (anosmia) and taste (ageusia), with less common ones including headaches, nasal congestion and runny nose, muscle pain, sore throat, diarrhea, eye irritation, and toes swelling or turning purple, and in moderate to severe cases, breathing difficulties. People with the COVID-19 infection may have different symptoms, and their symptoms may change over time. Three common clusters of symptoms have been identified: one respiratory symptom cluster with cough, sputum, shortness of breath, and fever; a musculoskeletal symptom cluster with muscle and joint pain, headache, and fatigue; and a cluster of digestive symptoms with abdominal pain, vomiting, and diarrhea. In people without prior ear, nose, or throat disorders, loss of taste combined with loss of smell is associated with COVID-19 and is reported in as many as 88% of symptomatic cases.

Published data on the neuropathological changes related with COVID-19 have been limited and contentious, with neuropathological descriptions ranging from moderate to severe hemorrhagic and hypoxia phenotypes, thrombotic consequences, changes in acute disseminated encephalomyelitis (ADEM-type), encephalitis and meningitis. Many COVID-19 patients with co-morbidities have hypoxia and have been in intensive care for varying lengths of time, confounding interpretation of the data.

Of people who show symptoms, 81% develop only mild to moderate symptoms (up to mild pneumonia), while 14% develop severe symptoms (dyspnea, hypoxia, or more than 50% lung involvement on imaging) that require hospitalization, and 5% of patients develop critical symptoms (respiratory failure, septic shock, or multiorgan dysfunction) requiring ICU admission.

1.2 Diagnosis

1.2.1Imaging

Chest CT scans may be helpful to diagnose COVID-19 in individuals with a high clinical suspicion of infection but are not recommended for routine screening. Bilateral multilobar ground-glass opacities with a peripheral, asymmetric, and posterior distribution are common in early infection Subpleural dominance, crazy paving (lobular septal thickening with variable alveolar filling), and consolidation may appear as the disease progresses. Characteristic imaging features on chest radiographs and computed tomography (CT) of people who are symptomatic include asymmetric peripheral ground-glass opacities without pleural effusions.

Many groups have created COVID-19 datasets that include imagery such as the Italian Radiological Society which has compiled an international online database of imaging findings for confirmed cases. Due to overlap with other infections such as adenovirus, imaging without confirmation by rRT-PCR is of limited specificity in identifying COVID-19. A large study in China compared chest CT results to PCR and demonstrated that though imaging is less specific for the infection, it is faster and more sensitive.

1.2.2Coding

In late 2019, the WHO assigned emergency ICD-10 disease codes U07.1 for deaths from lab-confirmed SARS-CoV-2 infection and U07.2 for deaths from clinically or epidemiologically diagnosed COVID-19 without lab-confirmed SARS-CoV-2 infection.

1.3. Organization of paper

The rest of the paper is organized as follows. Section 2 briefly presents related work of the COVID-19 detection models that use chest X-ray images and CT images. Section 3 outlines the preliminary concepts required to develop the proposed method. Section 4 elaborately presents the proposed WavStaCovNet-19

model. The experimental results and a performance analysis are discussed in Sections 5 and 6 concludes the paper.

2. Literature review

Several studies and research work have been carried out in the field of diagnosis from medical images such as computed tomography (CT) scans using artificial intelligence and deep learning. DenseNet architecture and recurrent neural network layer were incorporated for the analysis of 77 brain CTs by Grewal et al. RADnet demonstrates 81.82% hemorrhage prediction accuracy at the CT level. Three types of deep neural networks (CNN, DNN, and SAE) were designed for lung cancer classification by Song et al. The CNN model was found to have better accuracy as compared to the other models. Using deep learning, specifically convolutional neural network (CNN) analysis, Gonzalez et al. could detect and stage chronic obstructive pulmonary disease (COPD) and predict acute respiratory disease (ARD) events and mortality in smokers. During the outbreak of COVID-19, CT was found to be useful for diagnosing COVID-19 patients. The key point that can be visualized from the CT scan images for the detection of COVID-19 was ground-glass opacities, consolidation, reticular pattern, and crazy paving pattern.A study was done by Zhao et al. to investigate the relation between chest CT findings and the clinical conditions of COVID-19 pneumonia. Data on 101 cases of COVID-19 pneumonia were collected from four institutions in Hunan, China. Basic clinical characteristics and detailed imaging features were evaluated and compared. A study on the chest CTs of 121 symptomatic patients infected with coronavirus was done by Bernheim et al. . The hallmarks of COVID-19 infection as seen on the CT scan images were bilateral and peripheral ground-glass and consolidative pulmonary opacities. As it is difficult to obtain the datasets related to COVID-19, an open-sourced dataset COVID-CT, which contains 349 COVID-19 CT images from 216 patients and 463 non-COVID-19 CTs, was built by Zhao et al. . Using the dataset, they developed an AI-based diagnosis model for the diagnosis of COVID-19 from the CT images. On a testing set of 157 international patients, an AI-based automated CT image analysis tools for detection, quantification, and tracking of coronavirus was designed by Gozes et al. The accuracy of the model developed was 95%. The common chest CT findings of COVID-19 are multiple ground-glass opacity, consolidation, and interlobular septal thickening in both lungs, which are mostly distributed under the pleura . A deep learning–based software system for automatic COVID-19 detection on chest CT was developed by Zheng et al. using 3D CT volumes to detect COVID-19. A pre-trained UNet and a 3D deep neural network were used to predict the probability of COVID-19 infections on a set of 630 CT scans. Out of 1014 patients, 601 patients tested positive for COVID-19 based on RT-PCR and the results were compared with the chest CT. The sensitivity of chest CT in suggesting COVID-19 was 97% as shown by Ai et al. . In a series of 51 patients with chest CT and RTPCR tests performed within 3 days by Fang et al. , the sensitivity of CT for COVID-19 infection was 98% compared to RT-PCR sensitivity of 71%. An AI system (CAD4COVID-Xray) was trained on 24,678 CXR images including 1540 used only for validation while training. The radiographs were independently analyzed by six readers and by the AI system. Using RT-PCR test results as the reference standard, the AI system correctly classified CXR images as COVID-19 pneumonia with an AUC of 0.81

3. Methodology

This section elucidates the details of the proposed important weights-only transfer learning approach. The overall method can be divided into different parts; pruning, finetuning, and training. In a nutshell, first, we propose to prune the filters that are least important from the convolutional layers. The pruned models were then retrained using the ImageNet dataset in the second part of the study. Finally, the resulting fine-tuned models were then used for training and testing on the SARS-CoV-2 CT-scan dataset. Figure 1 shows the overall pruning, fine-tuning, and training pipeline. Further, the main contributions of the proposed work includes: 1) A novel important weights-only transfer learning approach for the

classification of COVID-19 CTscan images. 2) To reduce the models' run-time resource requirements, we proposed transferring only the significant weights by pruning the least important weights. 3) In order to identify the less less important filters of the model, we evaluate the importance of each filter based on their absolute sum. 4) The effectiveness of the proposed work is validated through multiple experiments with both the models; unpruned pre-trained models and the pruned. The experiments are performed on the SARS-CoV-2 CT-scan dataset . The following subsections contain the details of the proposed methodology.

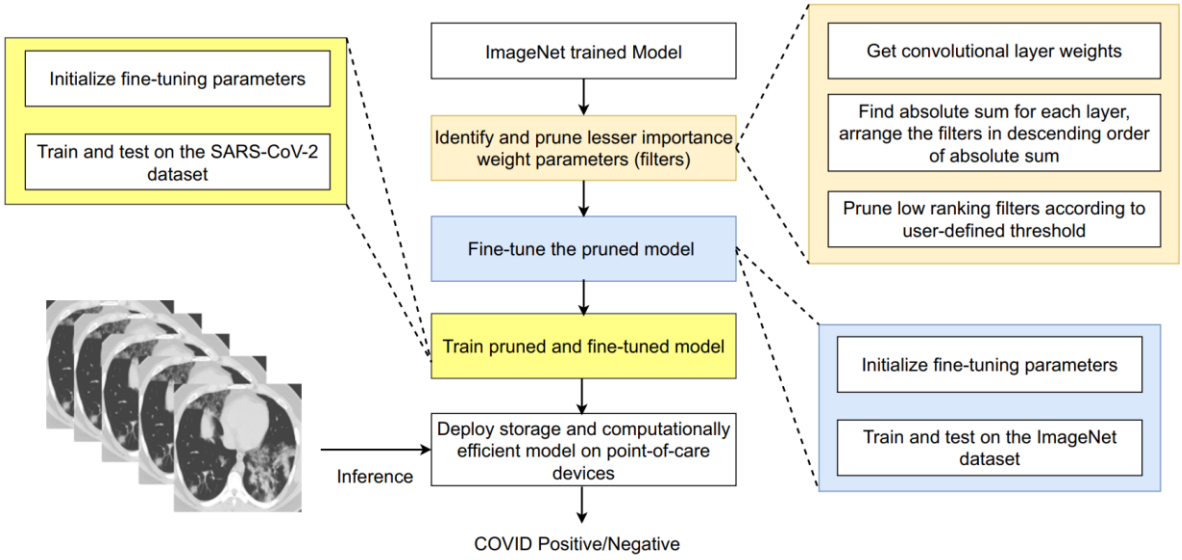


Fig. 1 Proposed important weights-only transfer learning approach

3.1 Step I: prune the least important filters

Convolutional filters are the backbone of any CNN architecture. However, in earlier research, it is found that not all filters are essential, and the removal of a few filters can be done without a significant accuracy loss . In this context, filter pruning has become popular in recent years. It does not necessitate any model architectural changes. The resulting model can be deployed without requiring any additional hardware or software for acceleration. Particularly in our research work, a layer-by-layer filter pruning was performed. One of the important tasks in filter pruning is to assess the filters' importance. The filter's magnitude, impact on loss/error, and batch normalization parameter can all be used to evaluate the filter's relevance. Different approaches can find distinct pruning filters. In order to determine the optimal criterion, we conducted several experiments. To find the number of pruning candidate filters using the batch normalization parameter, the model needs to be trained from the scratch. There are already pre-trained models on the ImageNet dataset, hence, training from scratch makes it computationally expensive. In a different experiment, the filters were pruned based on their impact on the loss/error to determine whether the removal of the filters increases or decreases the loss/error. We found that pruning the filters based on their impact of the loss/error is a time-consuming process. It requires the model to be trained after the removal of each filter. The magnitude of the weight parameters impact filters activations. The filter with a smaller magnitude generates weaker activation. Hence, we considered small magnitude (absolute sum) filters less important than the filters with larger magnitude Each convolutional layer k in the model produces the output as A(u×v×c), where u, v, and c correspond to the height, width, and the channels, respectively. The generated output A(u×v×c), works as an input for the k + 1 convolutional layer. Each convolutional filter generates one feature map when all the feature maps are combined they produce the feature maps of size A(u×v×c). In the proposed approach, filters that fail to meet the evaluation criterion

were pruned since the objective was to remove the less significant filters from the trained model. Further, we employed a binary masking approach to designate whether or not any filter in the layer would be pruned. All the filters of the layer were first sorted in decreasing order by their absolute sum. The aim of the research was to create an optimal model for point-of-care devices with the minimal number of convolutional filters while least compromising the model performance. Let No represent the original model, Np represent the pruned model and No has K convolutional layers, the kth layer is given by k[l] , and l ∈ (1, 2,... ,K). The number of filters for layer k[l] is ns and the generated activation map is given by Qmap. The activation Qmap works as input for the subsequent layer. Further, Gk[l] = [g1, g2,...,gns] represent the set of filter for layer k[l] . The original model weights of layer k[l] is Wk[l] o = [w1, w2,...,wns] and pruned model weights of the k[l] layer is Wk[l] p = [p1, p2,...,pnr], ns ——————————= nr. For dataset D = (xi, yi) N i=1 and a pruning threshold pt , the filter pruning problem is formulated as:

$$\min_{\mathcal{G}} \quad \mathcal{L}(\mathcal{G}; \mathcal{D}) \tag{1}$$

$$= \min_{\mathcal{G}} \quad \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{G}; (x_i, y_i)) \tag{2}$$

In (2), L(.) is the standard cross entropy loss. If the 1- norm or the absolute sum of filter gi is given by γ and γ ∈ R, the norm of filters gi is

———————————————————————————————————

γ

———————————————————————————————————

and defined as

$$\|\gamma\| = \sum_{i=1}^{c} \sum_{j=1}^{a} \sum_{e=1}^{b} \|W_{i,j,e}\| \tag{3}$$

For every γ , γ ≥ 0 for all γ ∈ R, and γ =0 iff γ =0. Let U = [u1, u2, ..., uns] represent the relative filter index for layer k[l] and U ∈ (0, 1). The set of filters of the layer are ranked in descending order according to their absolute sum. Then, the pruning threshold pt finds the number of filters to be pruned (X) for each layer, If γgi is than X[v], where v=0, the relative filter index U is to zero otherwise to one as

$$U_{n_s} = \begin{cases} 0 & if \|\gamma_{g_i\|} < \|X[v]\| \\ 1, & otherwise \end{cases} \tag{4}$$

3.2 Step II: re-train pruned models

The pruning of convolutional filters from the model leads to performance degradation. Therefore, it is required to re-train the pruned architecture on the original dataset before applying transfer learning. In this regard, before training the resulting pruned architectures on the CT-scan dataset, the resulting architectures are re-trained on the ImageNet dataset with standard data splits. The pre-trained models were used from the PyTorch deep learning framework. After pruning, during fine-tuning the pruned models, standard hyper-parameters were used for training and testing purposes.

3.3 Step III: training on the COVID dataset

After pruning and fine-tuning the pre-trained models, the SARS-CoV-2 dataset was utilized for training and testing the pruned models. The process of training pre-trained models on another dataset is referred to as transfer learning. Since we transferred only important weights of the pretrained models, we called this an important weight-only transfer learning. The models were trained with data splits discussed in Section 4.1. On the SARS-CoV-2 dataset, we conducted various experiments with the unpruned models and the ImageNet re-trained pruned models. The base and re-trained models were trained and tested in the following ways.

- Training the entire model (both original and the pruned & fine-tuned) on the SARS-CoV-2 dataset.
- Training the last layer of the model (both pruned & finetuned and the original) on the SARS-CoV-2 dataset.
- Training the entire dense layers on the SARS-CoV2 dataset (both VGG16 original and the pruned & fine-tuned).

4. Experimental setup and results

This section includes the detail of the dataset, CNN models, experimental setup, and evaluation metrics used to perform the experiments and measure the performance of the original & the pruned models.

4.1 Dataset

Deep learning algorithms require a large amount of labelled data to learn the distinguishing characteristics from images. In our study, we used a publicly available SARS-CoV2 CT-scan dataset to validate the effectiveness of the selective transfer learning method. The SARS-CoV-2 dataset comprises 2482 images. Out of the 2482 images, 1252 belong to the COVID-19 infected, and 1230 images belong to the COVID-19 non-infected class. The dataset was prepared by collecting images of actual patients in the hospital in Sao Paulo, Brazil. Figure 2 shows some random COVID-19 positive as well as COVID-19 negative images. The dimension of the images in the dataset is 224×224. The dataset is available for download from the link 1. As shown in Table 1, the dataset was divided into three parts as 68% training, 17% validation, and 15% testing.

Table 1 Normal and infected images from the SARS-CoV-2 CT-scan dataset

Class	Training	Validation	Testing	Total Images
Normal	836	208	186	1,230
Infected	851	212	189	1,252
Total	1,687	420	375	2,482

4.2 CNN models used

The experiments were performed on two CNN classification models, VGG16 and ResNet34. VGG16 is a popular CNN architecture that achieved 92.7% top-5 test accuracy in
ILSVRC 2014 challenge on the ImageNet dataset. VGG16 consists of convolutional, dense, and pooling layers, 3 × 3 filters for convolutional and 2 × 2 for max pooling, respectively. The model accepts 224 × 224 input images followed by two convolution layers with 64 filters and a max pooling to reduce the

output height and width to 112 × 112 × 64. Further, two convolutional layers with 128 filters are used, followed by a max pooling layer that reduces the activation size to 56 × 56 × 128. Similarly, three convolutional layers with 256 filters are followed by a pooling layer that reduces the output activation to 28 × 28×256. Finally, there are two stacks of three convolutional layers with 512 filters, separated by pooling layers. Next, dense layers with 4096 nodes accept the output of the last pooling layer which is 7 × 7 × 512. The dense layer is followed by one more dense layer with 4096 nodes. Finally, the model has the softmax layer with 1000 nodes. ResNet34 is another well-known architecture that performed better than VGG16 in ILSVRC challenge in the year 2015 and also archived first place in the competition. The ResNet34 design is made up of four residual blocks and is based on skip connections. The first block comprises six convolutional layers, each of which has 64, 3 × 3 filters. Eight convolutional layers and 128, 3 × 3 filters make up the second block. 256, 3 × 3 filters are used in the third block, consisting of 12 convolutional layers. The final block consists of six convolutional layers with 512 filters. Finally, it is followed by an average pooling and a softmax layer with 1000 nodes.

4.3 Evaluation metrics

The standard evaluation metrics were used to validate the performance of the CNN models. The evaluation metrics used in the experiments were confusion matrix, accuracy, precision, F1-score,

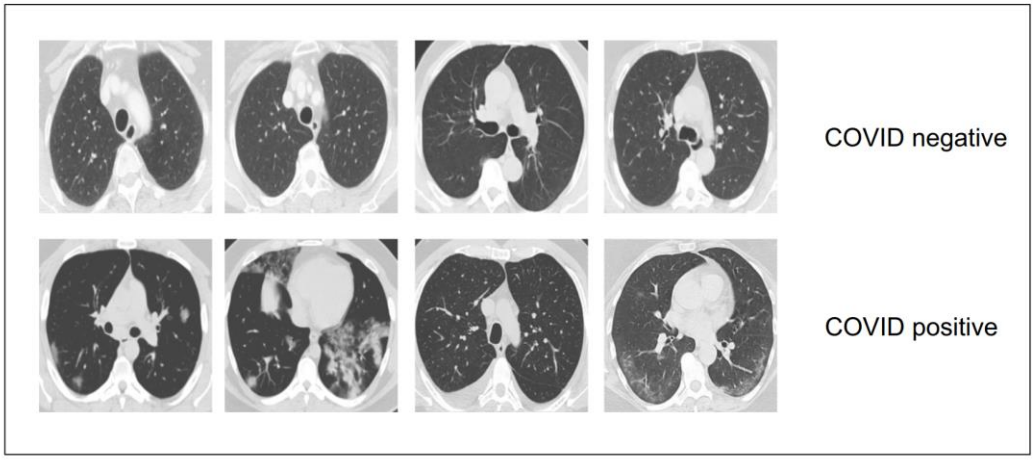


Fig. 2 COVID-19 positive (bottom row) and negative (top row) images from the dataset

recall/sensitivity, and specificity. The different evaluation metrics are defined as: (In the below equations, FN = false negatives, TN = true negatives, TP = true positives, FP = false positives).

Confusion matrix: The confusion matrix is one of the important evaluation metrics to measure the performance of binary classification problems. It measures the performance by comparing the actual values and the predicted values. When the actual sample is positive/negative and the model classifieds it as a positive/negative, it is known as true positive (TP) and true negative (TN), respectively. When the actual sample is negative (a person does have COVID), and the model predicted it as positive, it is known as a false positive (FP). Finally, if the actual sample is positive (a person has COVID) and the model predicted it as negative, it is known as false negative (FN).

Accuracy: Accuracy is defined as the total number of the correct classification made by the model to the total number of samples in the dataset. It is the most widely used evaluation metric to evaluate the classification performance and it is given by

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Precision: Precision is the ratio of the correct positives samples identified by the model out of the total number of positive predictions made by the model. The precision value ranges from zero to one; if the model does not have any false positives (FP = 0), it will always have a precision of one.

$$precision = \frac{TP}{TP + FP} \quad (6)$$

Sensitivity or recall: Sensitivity or recall is defined as the measure of what proportion of actual positives are identified correctly. If the model has no false negatives (FN = 0), then the recall will be one, which states that all the actual positives samples were classified correctly.

$$sensitivity/recall = \frac{TP}{TP + FN} \quad (7)$$

Specificity: It is the ratio of the total true negatives identified by the model to the sum of true negatives and false positives. It is given by

$$Specificity = \frac{TN}{TN + FP} \quad (8)$$

F1-score: The harmonic mean of precision and recall is used to calculate the F1-score. The value of the F1-score also varies between zero and one, where the values close to one are considered best. F1-score is defined as

$$F1 - score = \frac{2 \times recall \times precision}{recall + precision} \quad (9)$$

ROC curve: The receiver operating characteristic (ROC) curve is another important metric to measure the performance of binary classification problems. ROC curve is plotted between the true positive rate (y-axis) and false-positive rate (x-axis) for varying thresholds between zero and one.

4.4 Training VGG16 and ResNet34 on the SARS-CoV-2 dataset

The PyTorch deep learning framework was utilized to implement the experiments. NVIDIA DGX-1 V100 supercomputer was used as computing power. The initial experiments were performed on the VGG16 pruned and original model. The following were the hyper-parameters that were utilized for training,

validating, and testing. The models were trained for 200 epochs using a 0.001 learning rate. The training was carried out with a 0.9 momentum stochastic gradient descent (SGD) optimizer. All of the images were normalized before being fed into the model. The images were presented to the model in a mini-batch size of 32. In addition, the images were resized to 256 × 256 and were center cropped at 224 × 224. Further, the experiments were also performed with and without data augmentation on original and pruned models. Random horizontal flip, random vertical flip, and 20-degree random rotation were mainly applied to the training images. Two experiments were carried out on the ResNet 34 model, one to train only the last softmax layer and the other to train the entire model. The VGG16 model, on the other hand, was subjected to three sorts of experiments: training entire layers, dense layer, and the last layer. Atraining, validation, and test in the ration of 68%, 17%, and 15%, respectively. The result of training the original and pruned models on the SARS-CoV-2 CT-scan dataset are summarized in Tables 2, and 3, respectively. A detailed discussion of the result is given in the next sections given in Table 1, the dataset was split into

Table 2 Different performance measures for the ResNet34 and VGG16 on the test data (original pre-trained models)

Model trained	Augmentation	Accuracy	Precision	Recall	F1-score	Specificity	ROC-AUC
ResNet34	No	95.73	0.9471	0.9676	0.9572	0.9474	0.9931
ResNet34	Yes	97.87	0.9947	0.9641	0.9792	0.9944	0.9967
VGG16, dense	No	90.13	0.9206	0.8878	0.9039	0.9162	0.9667
VGG16, dense	Yes	89.87	0.9471	0.8647	0.9040	0.9405	0.9797
VGG16, all	No	90.40	0.9153	0.8964	0.9058	0.9121	0.9660
VGG, all	Yes	96.53	0.9630	0.9681	0.9655	0.9626	0.9957

Table 3 Different performance measures for the ResNet34 and VGG16 on the test data (pruned models)

Model trained	Augmentation	Accuracy	Precision	Recall	F1-score	Specificity	ROC-AUC
ResNet34	No	94.93	0.9312	0.9670	0.9488	0.9326	0.9888
ResNet34	Yes	95.47	0.9947	0.9216	0.9567	0.9942	0.9974
VGG16, dense	No	89.33	0.8730	0.9116	0.8919	0.8763	0.9669
VGG16, dense	Yes	89.33	0.8889	0.8984	0.8936	0.8883	0.9698
VG16, all	No	93.07	0.9418	0.9223	0.9319	0.9396	0.9744
VG16, all	Yes	92.80	0.9630	0.9010	0.9309	0.9595	0.9878

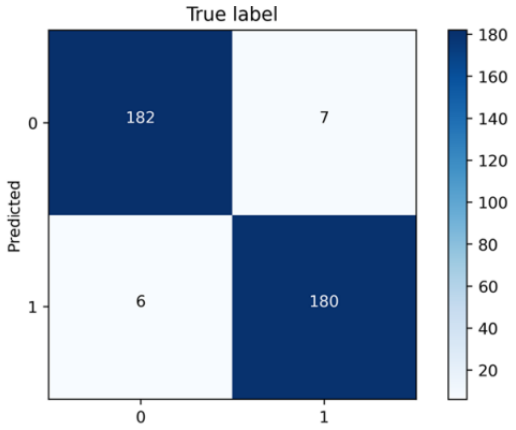
4.5 Results

The results of all the experiments utilizing the SARS-CoV2 dataset are discussed in this section. The section also contains the detail of the comparative study, complexity analysis, and statistical test. In this study, first, the experiments were performed with the pertained deep learning models VGG16 and ResNet34. Table 2 details the various evaluation metrics calculated after using test data on trained models. Table 2 shows that when the complete model was trained with data augmentation, the VGG16 model achieved higher accuracy. In this case, the model achieved 0.9630 precision, 0.9681 sensitivity, 0.9655 F1- score, 96.53% accuracy, 0.9626 specificity, and 0.9957 AUC. The ResNet34 model achieved higher accuracy when the entire model was trained with data augmentation. In this case, the model achieved 0.9947 precision, 0.9641 sensitivity, 0.9792 F1-score, 97.87% accuracy, 0.9944 specificity, and 0.9967 AUC. Table 3 shows the detail of different evaluation metrics calculated after applying test data on the pruned models. The pruned version of the VGG16 achieved 93.07% accuracy, 0.9223 sensitivity, 0.9319 F1-score, and 0.9396 specificity, and 0.9744 AUC. On the pruned version of the ResNet34 model 95.47% accuracy, 0.9216 sensitivity, 0.9567 F1-score, 0.9942 specificity, and 0.9974 AUC was achieved. In addition, for the ResNet34 and VGG16 models, Figs. 3 and 4 illustrate the confusion matrix, ROC curve, and precision-recall curve, respectively. On the basis of #parameters, #FLOPs, and accuracy, in

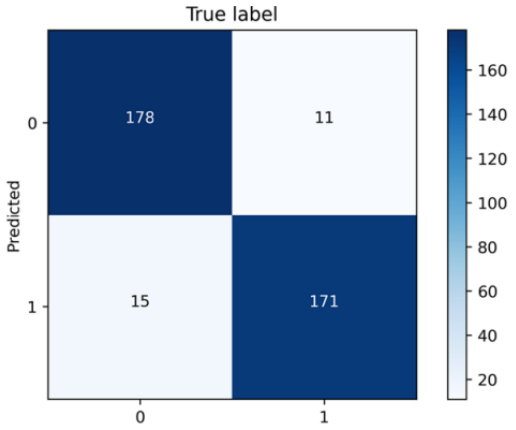
Table 4 we compared the original & pruned VGG16 and ResNet34 models. It is evident from Table 4 that the pruned VGG16 has 41.66% less weight parameters, and the #FLOPs also reduced by 77.47%. On the other hand, ResNet34 pruned model has 20.64% less weight parameters, and the #FLOPs were also reduced by 41.96%.

Table 4 Pruned and original model comparison, Augmentation = Aug, Million = M, Billion = B

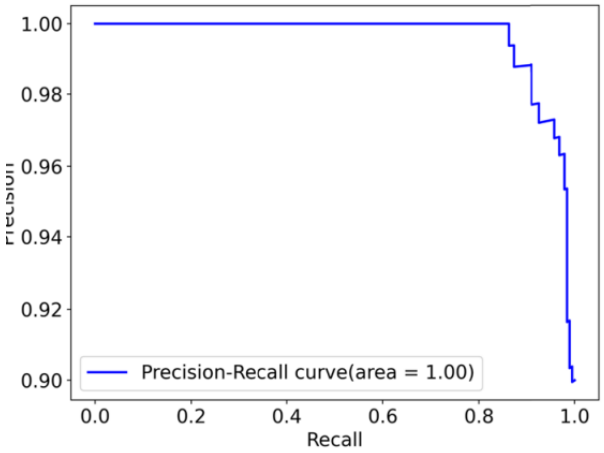
		Original			Pruned					
Model	Aug.	Para (M)	FLOP (B)	Acc.	Para (M)	FLOP (B)	Acc.	%Para ↓	%FLOP ↓	Acc(±)
ResNet34	No	21.28	3.67	95.73	16.89	2.13	94.93	20.64	41.96	-0.80
ResNet34	Yes	21.28	3.67	97.87	16.89	2.13	95.47	20.64	41.96	-2.40
VGG16, dense	No	134.26	15.49	90.13	78.33	3.49	89.33	41.66	77.47	-0.80
VGG16, dense	Yes	134.26	15.49	89.87	78.33	3.49	89.33	41.66	77.47	-0.53
VG16, all	No	134.26	15.49	90.40	78.33	3.49	93.07	41.66	77.47	2.67
VG16, all	Yes	134.26	15.49	96.53	78.33	3.49	92.80	41.66	77.47	-3.73



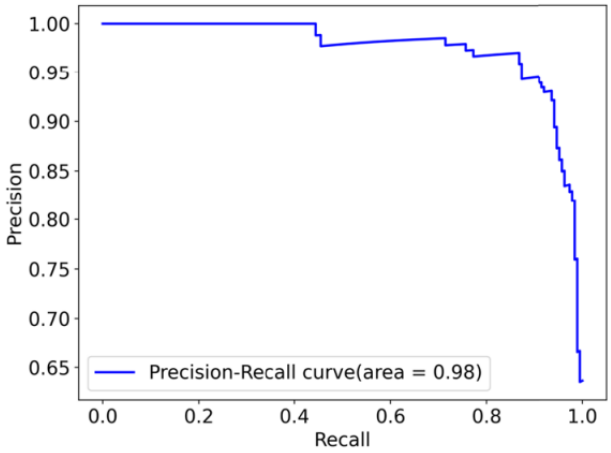
(a) Confusion matrix (original)



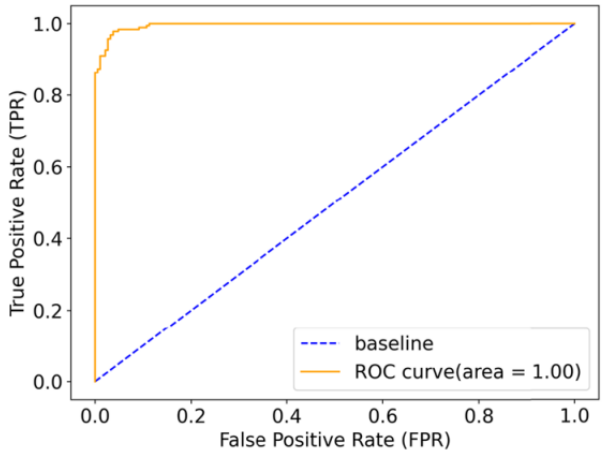
(b) Confusion matrix (pruned)



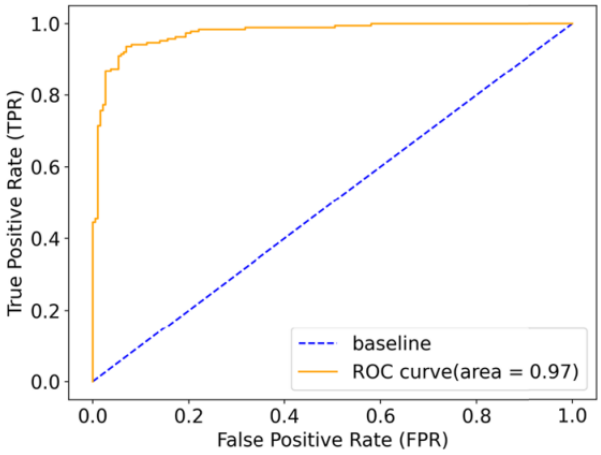
(c) Precision-recall curve (original)



(d) Precision-recall curve (pruned)



(e) ROC curve (original)



(f) ROC curve (pruned)

Fig. 3 The confusion matrix, precision-recall curve, and ROC curve for the VGG16 original (left) and pruned (right) model

4.6 Complexity analysis

The complexity of the proposed work was analyzed based on the time taken to classify the single image and the whole test set. For this, the pruned and original models were deployed on GPU & CPU. In addition, the complexity of the VGG16 model was assessed based on the number of parameters, and the FLOPs decreased layer-by-layer. Table 5 shows the inference time of the VGG16 and ResNet34 pruned and original models on CPU & GPU. It should be noted here from Table 5 that the pruned models have significant improvement in the inference time compared to the original models. Further, the test set was applied 50 times to record the inference time and the average was taken. The CPU and GPU inference time of the models for a single image and an entire set is shown in Figs. 5 and 6, respectively. It is also evident from Figs. 5 and 6 that the CPU and GPU time is less for the pruned models. Table 6 compares the complexity of the pruned and original VGG16 models in terms of weight parameters and FLOPs. The 13 layers of the VGG 16 model are divided into five blocks, where each block further follows the pooling layer. While counting the FLOPs, addition and multiplication were considered as a single operation. The pruned model contains fewer FLOPs and weight parameters, resulting in a considerable improvement in model inference performance, as shown in Table 6.

Table 5 Inference time (seconds), #parameters, FLOPs, and filters of the models

	Original model		Pruned model	
Metric	VGG16	ResNet34	VGG16	ResNet34
Parameters (M)	134.26	21.28	78.33	16.89
Parameter reduction (%)	0	0	41.66	20.64
FLOPs (B)	15.49	3.67	3.49	2.13
FLOPs reduction (%)	0	0	77.47	41.96
Convolutional filters	4224	8512	2073	7362
Convolutional filter reduction (%)	0	0	50.92	13.51
GPU inference-time, single image (s)	0.005219	0.004637	0.004154	0.004250
GPU inference-time, test set (s)	1.957030	1.738801	1.557787	1.593882
CPU inference-time, single image (s)	0.158153	0.058811	0.056285	0.040023
CPU inference-time, test set (s)	59.307411	22.054049	21.106688	15.008648

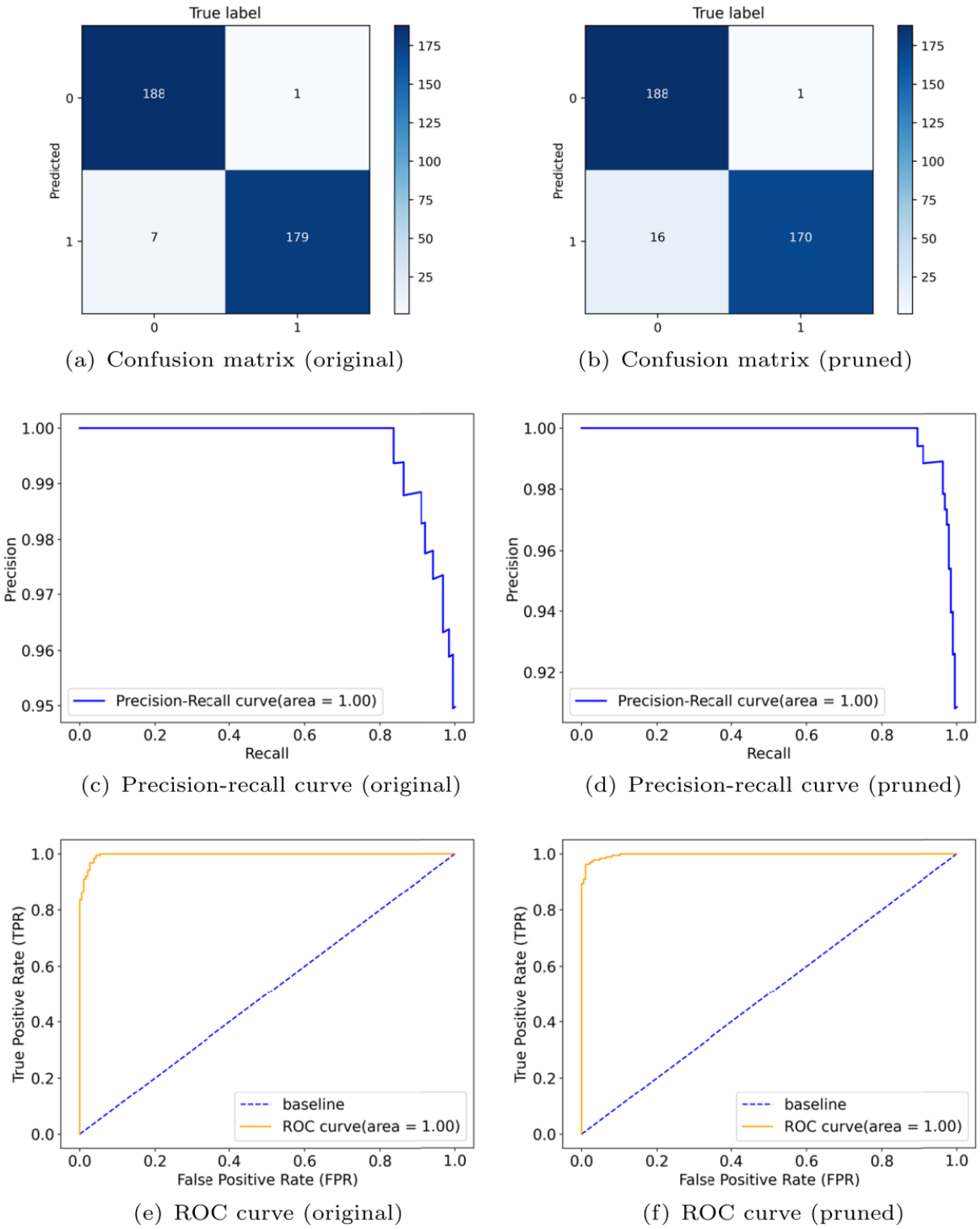
Table 6 VGG16 model complexity analysis

Convolutional block and layers		Before pruning		After pruning		Reduction	
Block	Filter, Layer	Parameters	FLOPs	Parameters	FLOPs	%Para red	%FLOP red
Conv block 1	64, 2	38720	1952448512	8374	424589312	78.37	78.25
Conv block 2	128, 2	221440	2782560256	40822	514002944	81.57	81.53
Conv block 3	256, 3	1475328	4629839872	348173	1093413440	76.40	76.38
Conv block 4	512, 3	5899776	4626628608	1306313	1024720144	77.86	77.85
Conv block 5	512, 3	7079424	1388269568	1837569	360513188	74.04	74.03
FC1	4,096 (neuron)	102764544	102764544	58007552	58007552	43.55	43.55
FC2	4,096 (neuron)	16781312	16781312	16781312	16781312	0.00	0.00
FC3	2 (neuron)	8194	8194	8194	8194	0.00	0.00
	Total	134.26M	15.49B	78.33M	3.49B	41.66	77.47

4.7 Paired statistical test

The inference-time was used as a dependent variable in a paired statistical t-test to validate the performance of the VGG16 and ResNet34 original and pruned models. The mean inference-time difference between the original and pruned models before and after pruning was compared using a paired sample t-test. For this, the hypotheses were established (null and alternate). The null hypothesis was that the mean inferencetime of the original and pruned models was the same ($H0 : \mu o = \mu p$). The mean

inference-time of the two models was not the same under the alternate hypothesis H1 : μo ——————————————= μp. The inference-time of the original and pruned models was determined by evaluating them on the test set with various test set splits. The test set was split into one, two, three, four, five, and ten equal parts. On each test split, the original and pruned models were evaluated, and the model inference time was recorded. α was set to 0.05 as the significance level. The VGG16 model had a p-value of less than 0.001 and a t-value of 4.504. There is sufficient evidence to reject the null hypothesis because
the p-value is smaller than alpha. The ResNet34 model has a t-value of 2.735 and a p-value less than 0.001. With alpha = 0.05, the ResNet34 model has a lower p-value, indicating that there is enough evidence to reject the null hypotheses.



(a) Confusion matrix (original)

(b) Confusion matrix (pruned)

(c) Precision-recall curve (original)

(d) Precision-recall curve (pruned)

(e) ROC curve (original)

(f) ROC curve (pruned)

Fig. 4 The confusion matrix, precision-recall curve, and ROC curve for the ResNet34 original (left) and pruned (right) model

4.8 Comparison

with other methods. The proposed method was compared with existing stateof-the-art methods and the result

are presented in Table 7. The authors of the research proposed a stacked ensemble of heterogeneously pretrained CNN models. The ensemble model was created by the combination of VGG19, ResNet101, Densenet169, and WideResNet50-2. No data augmentation was applied on the SARS-CoV-2 CT-scan dataset. As a result, the model failed to achieve good classification accuracy. The authors achieved 91.5% accuracy, 0.915 sensitivity, and 0.915 F-score. In addition, also proposed a stacked ensemble method. The authors claimed that the stacked ensemble method achieved higher recall and accuracy. The authors achieved 94% accuracy, 0.98 sensitivity, and 0.94 F-score. The methods proposed by and did not make any optimization and were less suitable for the point-of-care devices. The authors of the research concluded that the combination of transfer learning with segmentation methods such as U-Net improves the classification performance. Transfer learning with U-Net architecture outperformed other state-of-the-art transfer learning-based CNN methods. Without segmentation, the authors achieved 89.31% accuracy, 0.8240 sensitivity, 0.8860 F-score, and 0.9634 specificity. On the other hand, the authors achieved 89.92% accuracy, 0.8680 sensitivity, 0.8967 F-score, and 0.9309 specificity with the Fig. 5 Inference time (CPU and GPU) for single image segmentation scheme. However, both the schemes failed to achieve competitive performance. Moreover, also proposed a transfer learning-based approach for SARS-CoV-2 CT-scan classification. Particularly, the authors worked with the VGG16 and DenseNet201 models. The accuracy, sensitivity, F-score, and specificity for the VGG16 model were 95.45%, 0.9523, 0.9549, and 0.9567, respectively. The accuracy, sensitivity, F-score, and specificity achieved by the DenseNet201 model were 96.25%, 0.9629, 0.9629, and 0.9621, respectively. The results of the research were improved compared to the other methods. However, the storage, energy, and computational requirement of the pre-trained models were high. Hasan et al worked with the DenseNet121 convolutional architecture and obtained 0.95 sensitivity, 0.89 F-score, and 92% accuracy. The authors of the research proposed a redesigned COVID-Net for improved performance along with the objective of contrastive learning for cross-site learning. The authors achieved 90.83% accuracy, 0.8589 sensitivity, and 0.9087 F1-score. Angelov and Almeida Soares worked with the GoogleNet, ResNet, and AdaBoost methods. AdaBoost achieved higher accuracy compared to GoogleNet and ResNet. However, the GoogleNet model showed higher sensitivity. The authors of the research achieved 95.61% accuracy with the VGG16 pre-trained model. The authors also implemented gradCAM-based color visualization to interpret the predictions. It should be noted from Table 7 that none of the existing methods takes into consideration the constraints of the point-of-care devices. The last two-column of Table 7 shows that the existing methods make no reduction in FLOPs and learnable weight parameter. Further, it can be seen from Table 7 that the proposed method significantly reduces the inference-time needs of the models and also achieved competitive performance. Furthermore, our pruned ResNet-34 model achieved 95.47% classification accuracy, 0.9216 sensitivity, 0.9567 F-score, and 0.9942 specificity. On the other hand, the VGG16 pruned model achieved 93.07% accuracy, 0.9223 sensitivity, 0.9319 F1-score, and 0.9396 specificity. The pruned VGG16 and ResNet34 models reveal that the pre-trained models are over-parameterized and that removing low-importance parameters enhances the model's performance for point-of-care devices.

5.discussion

Early detection and treatment of infectious diseases play an important role in medical diagnosis. Many researchers have recently recommended radiological imaging-based approaches, given the present constraints of reverse transcription-polymerase chain reaction (RT-PCR)-based testing for diagnosing COVID19. Furthermore, with the development of AI-based technology, significant progress in automated medical diagnosis has been made. However during our research, it was found that high-performance techniques such as deep learning methods need high computational resources. For widespread benefits,

the trained deep learning model must be deployed in pointof-care devices. However, the point-of-care devices have limited resources to execute the large, trained models. Motivated by the deep learning models' ability to generate the diagnosis results accurately, timely, and the limitations of the point-of-care devices, a selective transfer learning approach was suggested in this study to classify CT-scan images as COVID-19 positive or negative. The result of the study indicates that the selective transfer learning approach effectively makes the deep learning models inference efficient for point-of-care devices in the medical domain for early diagnosis. It will help speed up the diagnosis process and significantly reduce the dependability on the skilled technicians, laboratories, and make the automated diagnosis more affordable in underprivileged areas. The comparative analysis found that the proposed method performed superior to existing methods in classifying chest CT-scan images. Moreover, none of the existing methods minimizes the trained models' run-time resource requirements for point-of-care devices. The VGG16 pruned model achieved 93.07% accuracy, while the Resnet-34 pruned model achieved 95.47% accuracy. Another noteworthy finding from this study is that the VGG16 model has 41.66 percent fewer parameters and 77.47 percent less floatingpoint operations than the standard model. Similarly, the ResNet-34 model has 20.64% fewer parameters and 41.96% fewer FLOPs than the standard model. Furthermore, this research finds that pre-trained CNN architectures are over-parameterized, and that filter pruning improves inference performance. The proposed method has advantages over other existing filter pruning methods. Unlike, other methods in which to identify the pruning candidate filters, the author remove the filters one by one and evaluate model loss after each pruned filter. Removing the filters one by one is a time consuming and computational intensive task. In contrast, in the proposed method, one shot pruning is applied to find all the candidate pruning filters. Moreover, the method is also different from those in which the convolutional filters are sparsified by setting some of the weights to zero. Such methods require specialized hardware and software to process the resulting sparse model . On the contrary, the proposed method completely removes the unimportant filters and their corresponding feature maps. Unlike , the proposed method doesn't required training the model from scratch to find the less important filters. In contrast, the proposed method can be applied to prune any pretrained model. Further, the current work focuses only on COVID-19 disease; however, the proposed important weights-only learning approach can be used for other applications in point-of-care devices. Some of the applications include detecting skin lesions, Pneumonia, and Tuberculosis, to name a few. In addition to various advantages of using CT-scan-based automatic image diagnosis for COVID-19 detection, such models can help radiologists effectively detect the virus. Also, these models not only show predictions or classifications over the CTscan but can also be used to monitor the outcome of the treatment .

Table 7 Comparison of the proposed important weights only approach with other methods on the SARS-CoV-2 dataset

Method	Accuracy	Sensitivity	F1-score	Specificity	FLOP(%) ↓	Para.(%) ↓
Varied threshold	91.5	0.915	0.915	-		
Without segmentation	89.31	0.8240	0.8860	0.9634		
With segmentation	89.92	0.8680	0.8967	0.9309		
VGG16	95.45	0.9523	0.9549	0.9567		
DenseNet	96.25	0.9629	0.9629	0.9621		
GoogleNet	91.73	0.9350	0.9182	-		
ResNet	94.96	0.9715	0.9503	-		
AdaBoost	95.16	0.9671	0.9514	-		
Contrastive Learning	90.83	0.8589	0.9087	-		
DenseNet-121	92	0.95	0.89	-		
Stacked Ensemble	94	0.98	0.94	-		
DNN	95.61	-	-	-		
Ours, VGG16 pruned	93.07	0.9223	0.9319	0.9396	77.47	41.66
Ours, ResNet34 pruned	95.47	0.9216	0.9567	0.9942	41.96	20.64