Applied Intelligence
https://doi.org/10.1007/s10489-022-03893-7

Deep learning-based important weights-only transfer learning approach for COVID-19 CT-scan classification

Tejalal Choudhary[1] · Shubham Gujar[2] · Anurag Goswami[1] · Vipul Mishra[1] · Tapas Badal[1]

Accepted: 13 June 2022

Abstract
COVID-19 has become a pandemic for the entire world, and it has significantly affected the world economy. The importance of early detection and treatment of the infection cannot be overstated. The traditional diagnosis techniques take more time in detecting the infection. Although, numerous deep learning-based automated solutions have recently been developed in this regard, nevertheless, the limitation of computational and battery power in resource-constrained devices makes it difficult to deploy trained models for real-time inference. In this paper, to detect the presence of COVID-19 in CT-scan images, an important weights-only transfer learning method has been proposed for devices with limited runt-time resources. In the proposed method, the pre-trained models are made point-of-care devices friendly by pruning less important weight parameters of the model. 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 Convolutional neural network · Deep learning · Pruning · COVID-19 · Automated diagnosis

1 Introduction

In recent times, millions of people have been infected worldwide with the COVID-19 pandemic. According to the WHO, more than 231M people have been infected with the COVID-19 to date leading to 4.74M deaths worldwide [1]. In the USA alone, 42.5M people were infected, and 681K people lost their lives. Moreover, the world's second-largest populated country, India, lost 447K lives, and 33.6M people get infected [1]. COVID-19 infection causes mild to severe respiratory disease. It is more dangerous for older persons and those with other disease such as cardiovascular, diabetes, chronic respiratory illness, and cancer. The most prevalent COVID-19 symptoms are dry cough, fever, and fatigue. However, the serious symptoms include loss of smell and taste, trouble in breathing, chest discomfort, aches, and pain [2].

A wide variety of tests are available to determine the presence of COVID-19. Some of the popular test includes reverse transcription-polymerase chain reaction (RT-PCR) [3] also known as molecular test, antigen test [4] for rapid testing, antibody test [5] also known as serology test, and using radiological images; CT-scan and chest X-ray [6, 7]. The antigen test is faster and inexpensive when compared to the RT-PCR. It has high false-positive rates and is thus less reliable. The rapid antibody test looks for antibodies in the patient's blood sample. The RT-PCR test is the most

✉ Vipul Mishra
vipul.mishra@bennett.edu.in

Tejalal Choudhary
tejalal.choudhary@gmail.com

Shubham Gujar
gujar303shubham@gmail.com

Anurag Goswami
anurag.goswami@bennett.edu.in

Tapas Badal
tapas.badal@bennett.edu.in

[1] Department of Computer Science Engineering, Bennett University, Greater Noida, 201310, Uttar Pradesh, India

[2] Department of Electronics & Telecommunications, Vishwakarma Institute of Information Technology, Pune, 411048, Maharashtra, India

Published online: 18 July 2022

prevalent test that is utilized for the detection of COVID-19. If the antibody test along with RT-PCR is positive, it confirms that the person is COVID-19 positive. However, one of the main disadvantages of RT-PCR is that it is time-consuming. The COVID-19 virus spreads quickly; by the time it gets detected, it can spread to others. The COVID-19 virus can also be detected using chest computed tomography (CT) scan images. A bilateral change can be observed in these CT-scan images [7, 8]. Moreover, the examination of the CT-scan images is challenging, time-consuming, and requires skilled radiologists. The radiologist's opinion also suffers from inter-observer variability [9].

In recent years, computerized medical diagnosis have gained a lot of attention. In the medical field, artificial intelligence (AI) methods such as deep learning have been applied in a variety of diagnosis [10, 11]. In addition, a convolutional neural network (CNN) is a deep learning variant capable of learning discriminating properties from images. CNN has already proven its superior performance on different classification [12], [13, 14], segmentation [15, 16] and detection [17] applications in the past compared to the traditional methods. Further, an affordable and acute medical facility in remote areas is limited due to the lack of specialized laboratories. Point-of-care devices have the ability to play an important role in generating quick diagnosis results in this regard. For practical applications, deploying the trained deep learning model onto the point-of-care devices is must for a widespread use. However, the deep learning models suffer from over-parameterization; they contain millions of learnable weight parameters and during inference, perform a lot of floating-point operations (FLOPs) [18]. When these models are utilized for transfer learning, their generalization performance on medical datasets becomes poor. Hence, this limits their deployability on the point-of-care devices as the deep learning models need an adequate amount of resources to execute the trained model, i.e. computational, battery power, and memory [19, 20]. Another issue with deep learning models is that they were trained on the massive ImageNet dataset containing 1.2 million images from 1000 classes. However, the datasets in the medical field contain less number of classes. Therefore, a lot of weight parameters become redundant and less important in the network and it also results in low generalization. As a result, it is important to reduce run-time resource requirements before deploying trained models onto point-of-care devices. In this regard, filter pruning [21–23] has become a popular technique for reducing the amount of computational and battery power required during inference and improving the overall inference performance.

Owing to the ability of the deep learning models to perform superior in various automated medical diagnosis, the advantages of bringing intelligence to point-of-care devices and their limited resource ability motivated us to develop an inference efficient method for COVID-19 classification. Particularly, in this paper, we proposed an important weights-only transfer learning approach to automatically classify the CT-scan images as COVID-19 infected or not in the context of point-of-care devices. We propose to prune less essential filters from the convolutional layers to lower the insignificant weight parameters of the pre-trained models. Thus, while training the models on the CT-scan dataset only important weight parameters are used and CT-scan images are then classified using the pruned models. The proposed method is significantly different from [18, 21, 22], and [23]. The work presented in [18] is similar to ours, the authors worked on pruning the smaller weight filters from the convolutional layers. However, in contrast to [18] wherein the authors worked with invasive ductal carcinoma (IDC) dataset for classifying breast cancer images. In our research, we evaluate the performance of the proposed approach on the SARS-Cov-2 CT-scan dataset. Further, in [18] the authors worked with $50 \times 50 \times 3$ images, whereas the SARS-Cov-2 CT-scan dataset has relatively higher dimension images. In [21], the authors utilized batch normalization parameters to find the unimportant filters. However, according to [21], to identify the weak filters, the model needs to be trained from the scratch and thereby limits its applicability for the standard pre-trained networks. In another words, the pruning of pre-trained networks require retraining on the ImageNet dataset. Unlike [21], there is no such restriction with the proposed method. Moreover, the proposed approach differs from [22], where the authors represent pruning as an optimization problem and prune the filters depending on the next layer's statistics instead of the current layer. We believe that pruning the filters based on the current layers' filter statistics is more significant. The authors of [23] proposed asymptotic soft filter pruning in which the filters are set to zero before first training epoch and during the retraining, the authors devise an strategy to update the previously pruned filters. The same procedure is repeated for the rest of the training epochs. This also requires model to be trained from scratch to identify the pruning candidates. Moreover, setting filter weights to zero does not completely remove the filters and makes the network sparse. In contrast, our method is aimed at finding the pruning candidate filters and completely removing the filters and their respective feature maps. Following are the main contributions of the paper:

- A novel important weights-only transfer learning approach for the classification of COVID-19 CT-scan images.
- We proposed transferring only the significant weights to decrease the models' run-time resource requirements by pruning the least important weights.

- To identify the less important filters of the model, we evaluate the importance of each filter based on their absolute sum.
- Performed multiple experiments with both the models; unpruned pre-trained models and the pruned. The experiments were performed on the SARS-CoV-2 CT-scan dataset [24], and the experimental results indicate the superiority of the work presented here.

This paper has been organized as follows. Section 2 presents the related work wherein the various deep learning methods proposed for the classification of COVID-19 are summarized. Section 3 provides the detail of the proposed important weights only transfer learning approach. The experiments performed to validate the method's effectiveness, dataset & models used, their results, and analysis is elucidated in Section 4. Finally, the discussion on the proposed methods and conclusion is included in Sections 5 and 6, respectively.

2 Related work

This section contains the details of the existing research methods wherein the authors show the need to ship the deep learning models on resource-limited devices. Further, the section also provides an overview of the important transfer learning and filter pruning methods as the proposed work is based on these, followed by a review of the existing research methods proposed for COVID-19 CT-scan image classification. The popularity of the deep learning methods has opened tremendous opportunities for researchers to extend their uses for on-device practical applications on resource-limited devices. In recent years, there is a growing demand to move the inference step from high computing machines to resource-limited devices [25]. In this context, deploying trained deep learning models on resource-limited devices is becoming more prevalent for a variety of applications [26, 27]. Multiple studies have shown the need and importance of deploying deep learning models on point-of-care devices as well [28, 29]. However, due to a large number of multiply and accumulate (MAC) operations and memory access operations, a typical deep learning application can quickly exhaust resource-limited device [30]. In [31], the authors show that deep learning architectures contain enormous parameters, requiring large storage space and computational resources.

Transfer learning plays an important role to improve the performance of the deep learning models wherein the knowledge learned by the model trained on the source domain is applied to a target domain. The research has shown that transfer learning significantly enhances model performance. In [13] and [18] it was found that the model trained using transfer learning performed superior compared to the model trained from scratch. Apart from transfer learning, in recent years, to improve the inference performance of the deep learning models, filter pruning has emerged as an important technique [32]. The researchers have shown that pruning not only speeds up the inference performance but also reduces overfitting problems and helps the model in learning clear and expressive features. In a research [33], the author proposed a flexible-rate pruning method to compress and accelerate the trained models in which to identify the filters to be pruned, the authors employ a greedy-based strategy and execute an iterative loss-aware pruning procedure. In another research [34], the authors argue that the filters are affected by other filters, therefore, only the magnitude is not enough to decide the importance of the filters. The authors proposed a meta-attribute-based pruning method in which the geometric distance of filters is considered as a pruning criterion. The method of [35], pruned the filters by calculating the learned representation median in the frequency domain in contrast to the existing method that prunes less important filters in the spatial domain. In short, there are various filter pruning methods to accelerate and compress the deep learning models, however, there are limited studies evaluating and analyzing pruning methods in the context of point-of-care devices.

Moreover, the deep learning model has shown outstanding results in classifying CT-scan images over traditional machine learning methods. In a research [36], the authors proposed a multi-modality medical image diagnosis approach using deep neural networks. The non-subsampled contourlet transform (NSCT) domain method was used to construct multi-modality images. In another research [37], the authors proposed automatic segmentation and classification method for COVID-19 CT-scan images. The authors used a total of 1069 images for training and 150 images were used for validation and testing. The U-Net model was compared for segmentation and the AlexNet, DenseNet, ResNet50, InceptionV3, VGG16, VGG19, and EfficientNet were compared with the proposed architecture for classification. The segmentation architecture performed better than U-Net and obtained 88% accuracy, while the classification architecture achieved 94.67% accuracy. In a research [38], the authors proposed a transfer learning-based classification approach for the detection of COVID-19 from CT-scan images. The experiments were performed on a total of 2492 images. A pre-trained DenseNet201 architecture was used as a base model. The authors also compared the results with other CNN architectures; VGG16, InceptionResNet, and ResNet152V2. The DenseNet model outperformed other CNN models and obtained 96.25% accuracy on the test dataset. In one of the research [39], a multi-objective

differential evolution-based approach was proposed to classify COVID-19 CT-scans. Compared to the other methods, the proposed approach achieved 1.97% better accuracy.

The authors of the research [40] proposed an automated classification approach using an ensemble of densely connected convolutional networks. An ensemble model was trained using ResNet152V2, DenseNet201, and VGG16 models. The authors found that compared to the other methods, the proposed method obtained 1.27% improved accuracy. In another study [41], machine learning, deep learning, and texture analysis based approaches were compared for early diagnosis. Along with texture analysis methods, support vector machines (SVM) and K-nearest neighbors (KNN) were implemented. Moreover, a 23-layer custom CNN was also trained other than pre-trained AlexNet and MobileNetV2. In a study [42], the authors proposed a stacked ensemble of pre-trained CNN architectures. The authors implemented four different architectures and performed extensive experimentation with the stacked ensemble. The proposed model achieved 90.75% accuracy. Similarly, in another study [43], the authors proposed a stacked ensemble-based method and claimed higher recall. The proposed method achieved 94% accuracy and 0.98 recall.

In one of the research [44], a combination of transfer learning with the U-Net segmentation architecture was proposed to classify the SARS-Cov-2 CT-scan dataset. The pre-trained DenseNet-169 on U-Net segmented data achieved 89.92% classification accuracy. In a research [45], the authors proposed a novel redesigned framework from existing COVID-Net along with contrastive learning for enhanced training. The proposed method achieved 90.83% accuracy. In another research [46], a transfer learning-based DenseNet-121 architecture was proposed and achieved 92% accuracy. The authors of [47] also proposed a transfer learning-based method to classify the COVID-19 CT-scan images. The authors also implemented grad-CAM based color visualization to interpret the predictions. The method achieved 95.61% accuracy. From the study of the existing methods, it was found that none of the existing methods were designed to address the constraints of the resource-limited devices. On the other hand, our proposed approach aims at reducing the inference-time requirements of the models to enable faster and acute diagnosis in resource-limited devices.

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, fine-tuning, 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 CT-scan 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 [24]. The following subsections contain the details of the proposed methodology.

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 [48]. 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.

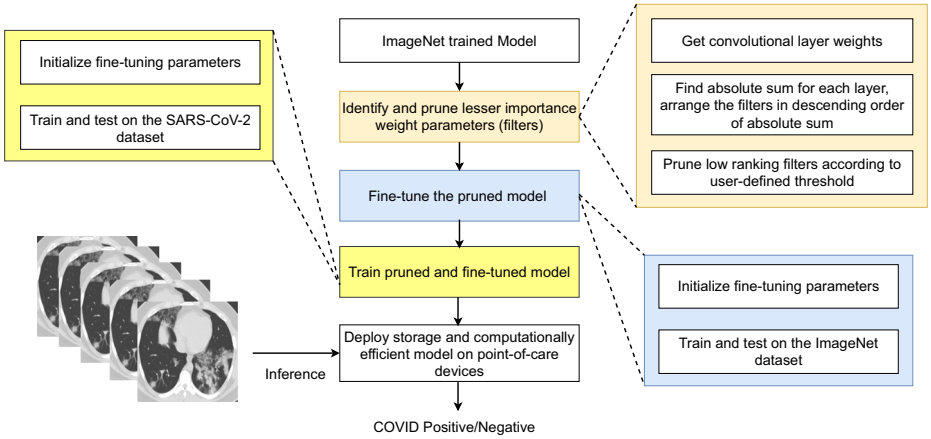


Fig. 1 Proposed important weights-only transfer learning approach

Each convolutional layer k in the model produces the output as $A^{(u\times v\times c)}$, where u, v, and c correspond to the height, width, and the channels, respectively. The generated output $A^{(u\times v\times c)}$, 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 $A^{(u\times v\times c)}$. 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 N_o represent the original model, N_p represent the pruned model and N_o has K convolutional layers, the k^{th} layer is given by $k^{[l]}$, and $l \in (1, 2, \ldots, K)$. The number of filters for layer $k^{[l]}$ is n_s and the generated activation map is given by Q_{map}. The activation Q_{map} works as input for the subsequent layer. Further, $\mathcal{G}^{k[l]} = [g_1, g_2, \ldots, g_{n_s}]$ represent the set of filter for layer $k^{[l]}$.

The original model weights of layer $k^{[l]}$ is $W_o^{k[l]} = [w_1, w_2, \ldots, w_{n_s}]$ and pruned model weights of the $k^{[l]}$ layer is $W_p^{k[l]} = [p_1, p_2, \ldots, p_{n_r}]$, $n_s \neq n_r$. For dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$ and a pruning threshold p_t, 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), $\mathcal{L}(.)$ is the standard cross entropy loss. If the $\ell 1$-norm or the absolute sum of filter g_i is given by γ and $\gamma \in \mathbb{R}$, the norm of filters g_i is $\|\gamma\|$ 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, $\|\gamma\| \geq 0$ for all $\gamma \in \mathbb{R}$, and $\|\gamma\|$=0 iff γ=0. Let $U = [u1, u2, \ldots\ldots\ldots, u_{n_s}]$ represent the relative filter index for layer $k^{[l]}$ and $U \in (0, 1)$. The set of filters of the layer are ranked in descending order according to their absolute sum. Then, the pruning threshold p_t finds the number of filters to be pruned (X) for each layer, If $\|\gamma_{g_i}\|$ is than $\|X[v]\|$, where v=0, the relative filter index U is to zero otherwise to one as

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

The filters index with value zero are less important and will be pruned, while those with value one will be preserved. In the proposed work, the filters whose corresponding index in U were zero, pruned from the model and a new architecture is created, and the weighs are copied from model N_o to N_p for the remaining filters. Further, the identified candidate pruning filters were then pruned in a single pass from the layer. For each of the convolutional layers, this process was repeated. When a filter from the layer k was pruned, the activation map associated with that filter was also pruned, resulting in reducing the #channels for the $k+1$ layer.

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 pre-trained 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 & fine-tuned and the original) on the SARS-CoV-2 dataset.
- Training the entire dense layers on the SARS-CoV-2 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-CoV-2 CT-scan dataset [24] 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.

4.2 CNN models used

The experiments were performed on two CNN classification models, VGG16 and ResNet34. VGG16 [49] 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 $112 \times 112 \times 64$. Further, two convolutional layers with 128 filters are used, followed by a max pooling layer that reduces the activation size to $56 \times 56 \times 128$. Similarly, three convolutional layers with 256 filters are followed by a pooling layer that reduces the output activation to $28 \times 28 \times 256$. 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 $7 \times 7 \times 512$. The dense layer is followed by one more dense layer with 4096 nodes. Finally, the model has the softmax layer with 1000 nodes. ResNet34 [50] 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, 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).

[1] https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset

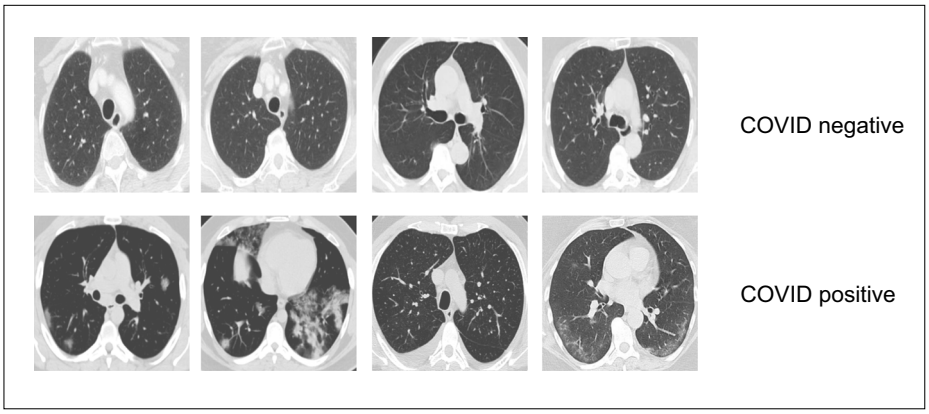


Fig. 2 COVID-19 positive (bottom row) and negative (top row) images from the dataset

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

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

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.

$$precision = \frac{TP}{TP + FP} \tag{6}$$

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.

$$sensitivity/recall = \frac{TP}{TP + FN} \tag{7}$$

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

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

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

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

$$F1 - score = \frac{2 \times recall \times precision}{recall + precision} \tag{9}$$

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. As 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

training, 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 section.

4.5 Results

The results of all the experiments utilizing the SARS-CoV-2 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%.

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

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

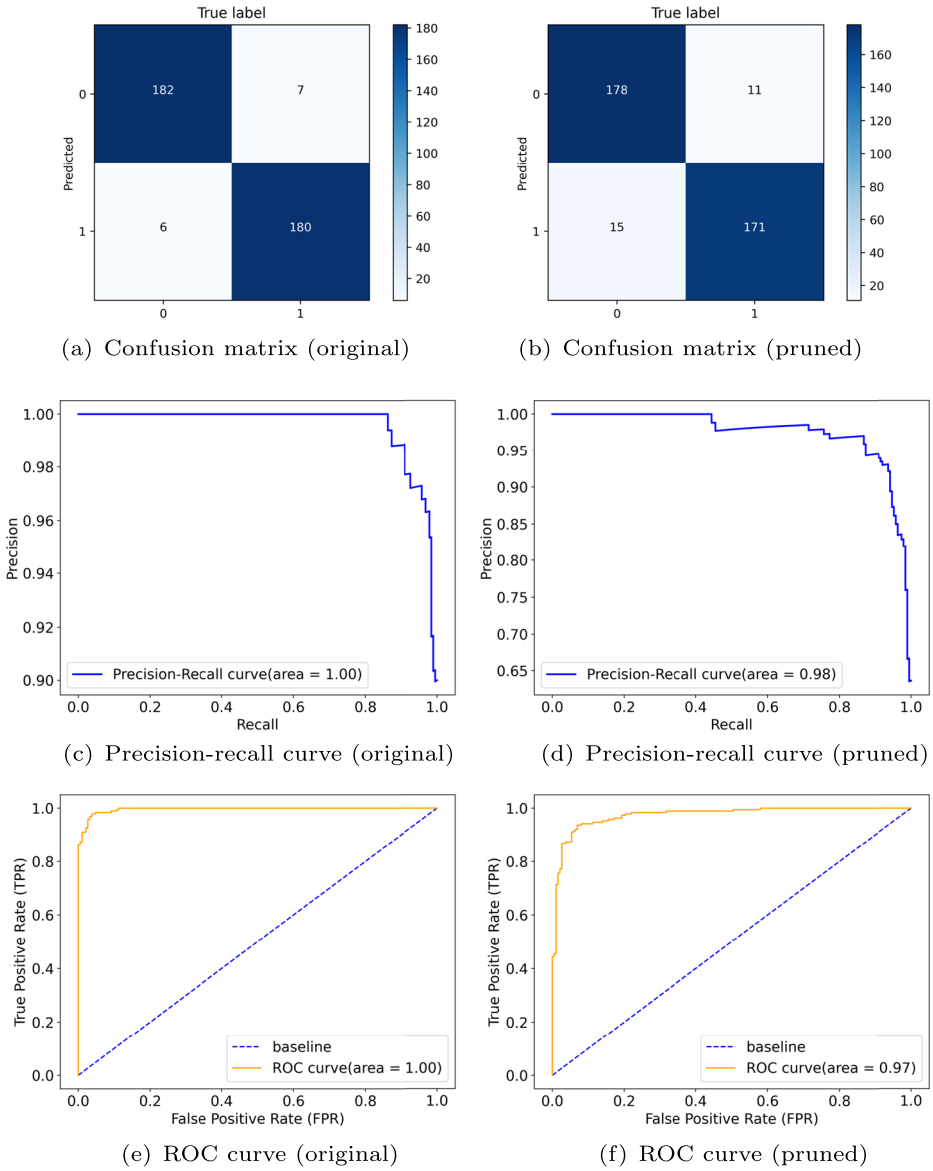


Fig. 3 The confusion matrix, precision-recall curve, and ROC curve for the VGG16 original (left) and pruned (right) model

(a) Confusion matrix (original) (b) Confusion matrix (pruned)

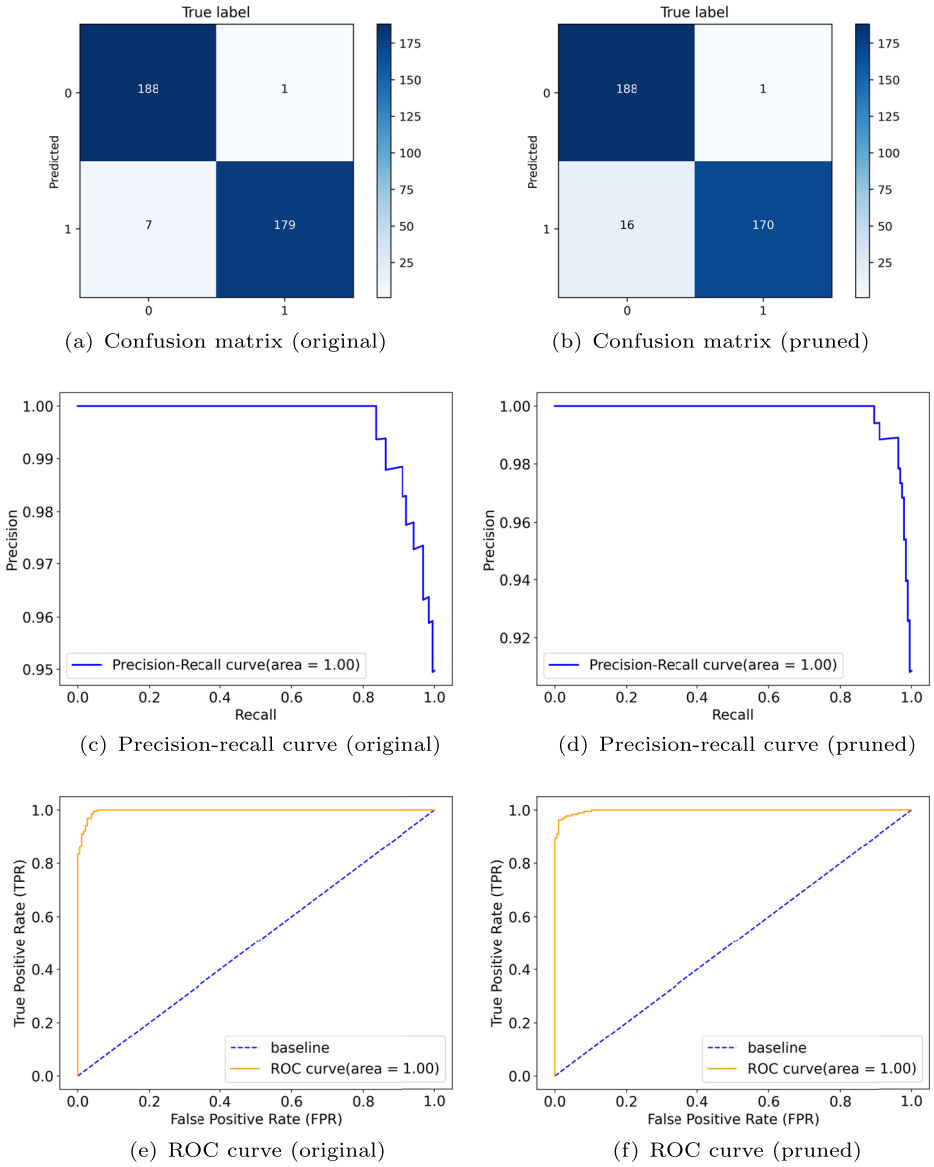
(c) Precision-recall curve (original) (d) Precision-recall curve (pruned)

(e) ROC curve (original) (f) ROC curve (pruned)

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.

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 ($H_0 : \mu_o = \mu_p$). The mean inference-time of the two models was not the same under the alternate hypothesis $H1 : \mu_o \neq \mu_p$. 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



(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

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.

4.8 Comparison with other methods.

The proposed method was compared with existing state-of-the-art methods [24, 38, 42–47, 51] and the results

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

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

Metric	Original model		Pruned model	
	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

are presented in Table 7. The authors of the research [42] proposed a stacked ensemble of heterogeneously pre-trained 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, [43] 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 [42] and [43] did not make any optimization and were less suitable for the point-of-care devices. The authors of the research [44] 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 segmentation scheme. However, both the schemes failed to achieve competitive performance.

Moreover, [38] 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 [38] were improved compared to the other methods. However, the storage, energy, and computational requirement of the pre-trained models were high. Hasan et al [46] worked with the DenseNet121 convolutional architecture and obtained 0.95 sensitivity, 0.89 F-score, and 92% accuracy. The authors of the research [45] 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 [24] worked with the GoogleNet, ResNet, and AdaBoost methods. AdaBoost achieved higher accuracy compared to GoogleNet and ResNet. However, the

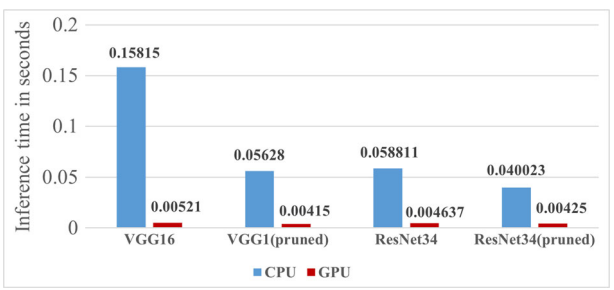


Fig. 5 Inference time (CPU and GPU) for single image

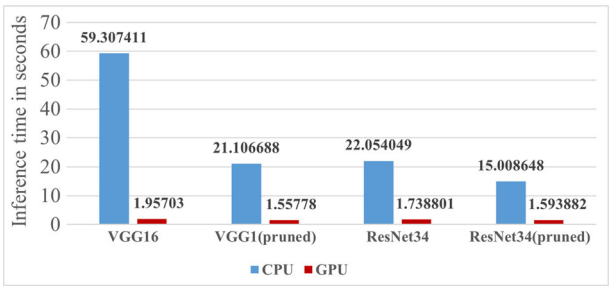


Fig. 6 Inference time (CPU and GPU) on the entire test set for all the models

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

GoogleNet model showed higher sensitivity. The authors of the research [47] achieved 95.61% accuracy with the VGG16 pre-trained model. The authors also implemented grad-CAM-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,

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.(%) ↓
[42] Varied threshold	91.5	0.915	0.915	-		
[44] Without segmentation	89.31	0.8240	0.8860	0.9634		
[44] With segmentation	89.92	0.8680	0.8967	0.9309		
[38] VGG16	95.45	0.9523	0.9549	0.9567		
[38] DenseNet	96.25	0.9629	0.9629	0.9621		
[24] GoogleNet	91.73	0.9350	0.9182	-		
[24] ResNet	94.96	0.9715	0.9503	-		
[24] AdaBoost	95.16	0.9671	0.9514	-		
[45] Contrastive Learning	90.83	0.8589	0.9087	-		
[46] DenseNet-121	92	0.95	0.89	-		
[43] Stacked Ensemble	94	0.98	0.94	-		
[47] 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**

The bold text shows our research result

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 point-of-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 floating-point 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 [48]. On the contrary, the proposed method completely removes the unimportant filters and their corresponding feature maps. Unlike [21], 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 CT-scan but can also be used to monitor the outcome of the treatment [52, 53].

6 Conclusion

In this paper, we proposed an important weights-only transfer learning method to classify the CT-scan images as COVID-19 infected or not. The proposed method has superior performance, and the trained models have reduced inference-time resource requirements. Another important conclusion of the study was that deploying the pruned models on point-of-care devices is advantageous. The computational requirement of the VGG16 and ResNet34 models was reduced by 77.47% and 41.66% in terms of the number of floating-point operations to be performed during inference. The pruned and original architectures were tested on GPU and CPU to see the practical speed-up and a significant speed-up is measured in the performance of the pruned architectures. Future studies can be aimed at developing inference-efficient models for other kinds of diseases for point-of-care devices. Moreover, the new structured filter pruning techniques can be developed to better find the filters that produce weaker activation. Further, the proposed pruning approach can be combined with other acceleration techniques to improve inference performance on point-of-care devices.

Author Contributions Tejalal Choudhary: Conceptualization, Methodology, Software, Writing - Original Draft Shubham Gujar: Software, Data Curation, Writing - Original Draft Vipul Mishra: Methodology, Data Curation, Writing- Reviewing and Editing, Validation Anurag Goswami: Methodology, Writing- Reviewing and Editing, Validation Tapas Badal: Writing- Reviewing and Editing , Supervision

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Declarations

Conflict of Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Global coronavirus (covid-19) (2021) World Health Organization (WHO) (Accessed: 28 September 2021). https://covid19.who.int/.

2. Singhal T (2020) A review of coronavirus disease-2019 (covid-19). The Indian J Pediatrics 87(4):281–286
3. van Kasteren PB et al (2020) Comparison of seven commercial rt-pcr diagnostic kits for covid-19. J Clin Virol 128:104412
4. Mak GC et al (2020) Evaluation of rapid antigen test for detection of sarscov-2 virus. J Clinic Virology 129:104500
5. Adams ER et al (2020) Antibody testing for covid-19: a report from the national covid scientific advisory panel. Wellcome Open Res ,vol 05
6. Chandra TB, Verma K, Singh BK, Jain D, Netam SS (2021) Coronavirus disease (covid-19) detection in chest x-ray images using majority voting based classifier ensemble. Expert Syst Appl 165:113909
7. Ai T et al (2020) Correlation of chest ct and rt-pcr testing for coronavirus disease 2019 (covid-19) in china: a report of 1014 cases. Radiology 296(2):E32–E40
8. Hossein H et al (2020) Value of chest computed tomography scan in diagnosis of covid-19; a systematic review and meta-analysis. Clinical Trans Imaging:1–13
9. Popović ZB, Thomas JD (2017) Assessing observer variability: a user's guide. Cardiovascular diagnosis and therapy 7(3):317
10. Tseng K-K, Zhang R, Chen C, Hassan MM (2021) Dnetunet: a semi-supervised cnn of medical image segmentation for supercomputing ai service. J Supercomput 77(4):3594–3615
11. Lee J-G et al (2017) Deep learning in medical imaging: general overview. Korean J Radiology 18(4):570
12. Sun Y, Xue B, Zhang M, Yen GG (2019) Evolving deep convolutional neural networks for image classification. IEEE Trans Evol Comput 24(2):394–407
13. Singh R et al (2020) Imbalanced breast cancer classification using transfer learning. IEEE/ACM Trans Comput Bio Bioinformatics 18(1):83–93
14. Coudray N et al (2018) Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. Nat Med 24(10):1559–1567
15. Maity A, Nair TR, Mehta S, Prakasam P (2022) Automatic lung parenchyma segmentation using a deep convolutional neural network from chest x-rays. Biomedical Signal Process Control 73:103398
16. Wang EK, Chen C-M, Hassan MM, Almogren A (2020) A deep learning based medical image segmentation technique in internet-of-medical-things domain. Futur Gener Comput Syst 108:135–144
17. Hamed G, Marey MAE-R, Amin SE-S, Tolba MF (2020) Deep learning in breast cancer detection and classification. (organization Springer):322–333
18. Choudhary T, Mishra V, Goswami A, Sarangapani J (2021) A transfer learning with structured filter pruning approach for improved breast cancer classification on point-of-care devices. Comput Biology Medicine 134:104432
19. Choudhary T, Mishra V, Goswami A, Sarangapani J (2020) A comprehensive survey on model compression and acceleration. Artificial Intell Review:1–43
20. Cheng Y, Wang D, Zhou P, Zhang T (2018) Model compression and acceleration for deep neural networks: the principles, progress, and challenges. IEEE Signal Proc Mag 35(1):126–136
21. Liu Z, Others (2017) Learning efficient convolutional networks through network slimming:2736–2744
22. Luo J-H, Others (2018) Thinet: pruning cnn filters for a thinner net. IEEE Trans Pattern Anal Mach Intell 41(10):2525–2538
23. He Y et al (2019) Asymptotic soft filter pruning for deep convolutional neural networks. IEEE Trans Cybernetics 50(8):3594–3604
24. Angelov P, Almeida Soares E (2020) Sars-cov-2 ct-scan dataset: a large dataset of real patients ct scans for sars-cov-2 identification. medRxiv
25. Chen S, Zhao Q (2018) Shallowing deep networks: layer-wise pruning based on feature representations. IEEE Trans Pattern Anal Mach Intell 41(12):3048–3056
26. Shokoohi H et al (2019) Enhanced point-of-care ultrasound applications by integrating automated feature-learning systems using deep learning. J Ultrasound Med 38(7):1887–1897
27. Pathania D et al (2019) Point-of-care cervical cancer screening using deep learning-based microholography. Theranostics 9(26):8438
28. Rahman MA, Hossain MS, Alrajeh NA, Gupta B (2021) A multimodal, multimedia point-of-care deep learning framework for covid-19 diagnosis. ACM Trans Multimidia Comput Commun Appl 17(1s):1–24
29. Holmström O., Others (2017) Point-of-care mobile digital microscopy and deep learning for the detection of soil-transmitted helminths and schistosoma haematobium. Global Health Action 10(sup3):1337325
30. Chen C, Others (2020) Deep learning on computational-resource-limited platforms: a survey. Mob Inf Syst, vol 2020
31. Dong M, Wen S, Zeng Z, Yan Z, Huang T (2019) Sparse fully convolutional network for face labeling. Neurocomputing 331:465–472
32. Xu X, Chen J, Su H, Xie L. (2022) Towards efficient filter pruning via topology. J Real-Time Image Proc:1–11
33. Li G et al (2022) Optimizing deep neural networks on intelligent edge accelerators via flexible-rate filter pruning. J Syst Archit 124:102431
34. He Y, Liu P, Zhu L, Yang Y (2022) Filter pruning by switching to neighboring cnns with good attributes. IEEE Trans Neural Netw Learn Syst
35. Zhang X, Xie W, Li Y, Lei J, Du Q (2021) Filter pruning via learned representation median in the frequency domain. IEEE Trans Cybernetics
36. Kaur M (2020) Multi-modality medical image fusion technique using multi-objective differential evolution based deep neural networks. J Ambient Intell Humanized Comput:1–11
37. Amyar A, Modzelewski R, Li H, Ruan S. (2020) Multi-task deep learning based ct imaging analysis for covid-19 pneumonia: classification and segmentation. Comput Bio Med 126:104037
38. Jaiswal A, Gianchandani N, Singh D, Kumar V, Kaur M (2020) Classification of the covid-19 infected patients using densenet201 based deep transfer learning. J Biomol Struct Dyn:1–8
39. Singh D, Kumar V, Kaur M et al (2020) Classification of covid-19 patients from chest ct images using multi-objective differential evolution–based convolutional neural networks. European J Clinical Microbio Infectious Diseases 39(7):1379–1389
40. Singh D, Kumar V, Kaur M (2021) Densely connected convolutional networks-based covid-19 screening model. Appl Intell:1–8
41. Yasar H, Ceylan M (2020) A novel comparative study for detection of covid-19 on ct lung images using texture analysis, machine learning, and deep learning methods. Multimed Tools Appl:1–25
42. Jangam E, Barreto AAD, Annavarapu CSR (2021) Automatic detection of covid-19 from chest ct scan and chest x-rays images using deep learning, transfer learning and stacking. Appl Intell:1–17
43. Jangam E, Annavarapu CSR (2021) A stacked ensemble for the detection of covid-19 with high recall and accuracy. Comput Biol Med 135:104608
44. Seum A, Raj AH, Sakib S, Hossain T (2020) A comparative study of cnn transfer learning classification algorithms with segmentation for covid-19 detection from ct scan images. IEEE:234–237
45. Wang Z, Liu Q, Dou Q (2020) Contrastive cross-site learning with redesigned net for covid-19 ct classification. IEEE J Biomed Health Inf 24(10):2806–2813

46. Hasan N, Bao Y, Shawon A, Huang Y (2021) Densenet convolutional neural networks application for predicting covid-19 using ct image. SN Comput Sci 2(5):1–11
47. Panwar H, Others (2020) A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. Chaos, Solitons Fractals 140:110190
48. Tung F, Mori G (2018) Deep neural network compression by in-parallel pruning-quantization. IEEE Trans Pattern Anal Mach Intell 42(3):568–579
49. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
50. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition:770–778
51. Mishra AK, Das SK, Roy P, Bandyopadhyay S (2020) Identifying covid19 from chest ct images: a deep convolutional neural networks based approach. J Healthcare Eng:2020
52. Paul D et al (2017) Feature selection for outcome prediction in oesophageal cancer using genetic algorithm and random forest classifier. Comput Med Imag Graph 60:42–49
53. Amyar A, Others (2018) Radiomics-net: convolutional neural networks on fdg pet images for predicting cancer treatment response