



Deep learning empowered COVID-19 diagnosis using chest CT scan images for collaborative edge-cloud computing platform

Vipul Kumar Singh¹ · Maheshkumar H. Kolekar¹

Received: 30 November 2020 / Revised: 1 May 2021 / Accepted: 7 June 2021 /

Published online: 28 June 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The novel coronavirus outbreak has spread worldwide, causing respiratory infections in humans, leading to a huge global pandemic COVID-19. According to World Health Organization, the only way to curb this spread is by increasing the testing and isolating the infected. Meanwhile, the clinical testing currently being followed is not easily accessible and requires much time to give the results. In this scenario, remote diagnostic systems could become a handy solution. Some existing studies leverage the deep learning approach to provide an effective alternative to clinical diagnostic techniques. However, it is difficult to use such complex networks in resource constraint environments. To address this problem, we developed a fine-tuned deep learning model inspired by the architecture of the MobileNet V2 model. Moreover, the developed model is further optimized in terms of its size and complexity to make it compatible with mobile and edge devices. The results of extensive experimentation performed on a real-world dataset consisting of 2482 chest Computerized Tomography scan images strongly suggest the superiority of the developed fine-tuned deep learning model in terms of high accuracy and faster diagnosis time. The proposed model achieved a classification accuracy of 96.40%, with approximately ten times shorter response time than prevailing deep learning models. Further, McNemar's statistical test results also prove the efficacy of the proposed model.

Keywords Chest CT scan · COVID-19 · Deep Learning · Diagnosis · Edge Computing · MobileNet V2

✉ Vipul Kumar Singh
vipul_1911ee13@iitp.ac.in

Maheshkumar H. Kolekar
mahesh@iitp.ac.in

¹ Department of Electrical Engineering, Indian Institute of Technology, Patna, India

1 Introduction

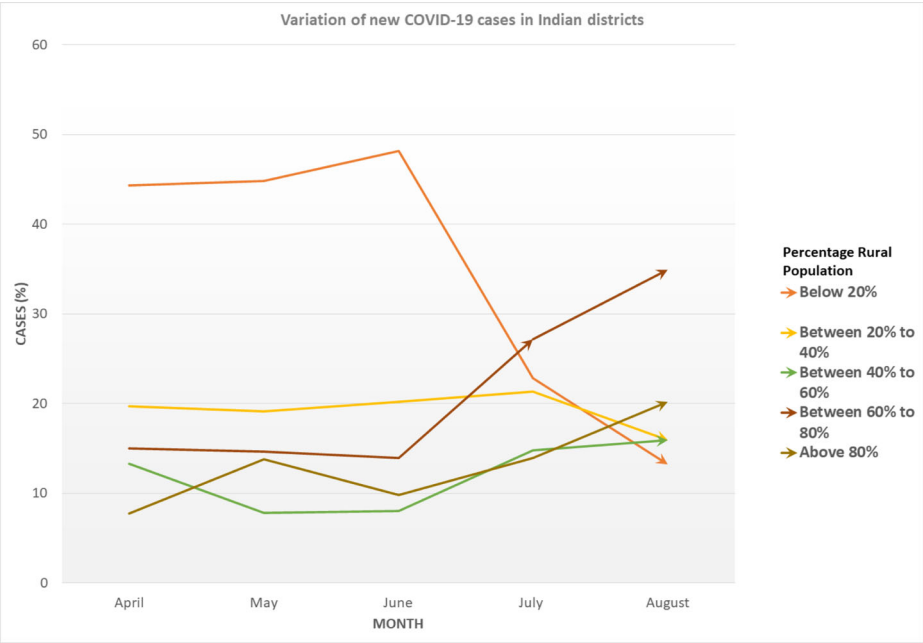
In December 2019, World Health Organization (WHO) received information about an outbreak of novel coronavirus infectious disease among humans across China's Hubei province. WHO named the contagious disease as COVID-19, where '19' represents the year 2019, when the virus was first identified [15]. Later, on 11th March 2020, this outbreak was officially declared as a pandemic by the then director of WHO [14]. COVID-19 is induced by a family of viruses called coronavirus that infects both humans and animals. A higher percentage of the infected patients show symptoms like fever, dry cough, and tiredness. Many other symptoms, such as loss of taste or smell, discoloration of fingers or toes, etc., are also noticed in a small percentage of patients [45]. Once this virus infects a person, it takes an average time of 5–6 days (incubation period), ranging between 1 to 14 days after exposure to develop symptoms. As of 11th September 2020, the WHO's diagnostic guidelines include details of laboratory testing for the detection of the unique viral sequence by Nucleic Acid Amplification Testing (NAAT), such as real-time reverse-transcription polymerase chain reaction (rRT-PCR) [17]. In these rRT-PCR tests, the samples of the throat or nasal swabs of the symptomatic individual are taken and tested for the traces of the virus. Several Rapid Diagnostic Tests (RDTs) are also developed to diagnose the infection within 30 minutes.

1.1 Motivation

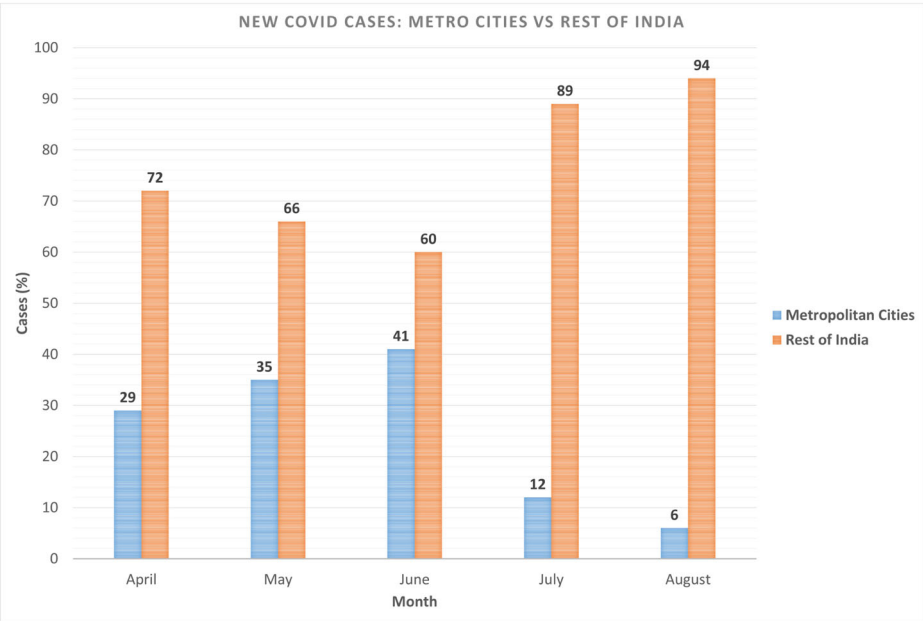
Clinical diagnostic tests such as rRT-PCR are time-consuming and are not readily available because of their low production and high cost. And these clinical tests also face several criticism over false-negative predictions. While experts are engaged in managing COVID-19 cases in metropolitan and major cities, a significant surge in the confirmed cases of the infection is observed in remote and rural areas of developing countries like India and Brazil. Figure 1 shows the shift of COVID-19 cases from cities to rural areas in India [9]. With a lack of proper medical facilities and trained medical personnel, combating community spread in remote and rural areas is quite challenging. Hence, there is a need for an easily accessible computer-aided diagnostic system that can be used for the rapid diagnosis of the virus.

Since the COVID-19 virus mainly targets the respiratory system of the human body, chest radiography images are playing a significant role in COVID-19 diagnosis [28]. Usage of chest CT images has been suggested over chest X-ray images as 3D representation in CT images provide better localization of the infection. Three symptoms can primarily be used to diagnose COVID-19 patients: Ground-Glass Opacities (GGO), consolidation, and pleural effusion [54]. The chest CT scan of COVID-19 patients will have gray patches because of the presence of ground-glass opacities. Pleural effusion is a health condition when fluid gets accumulated around the lungs. With cases of COVID-19 increasing tremendously, it is tedious for radiologists to manually analyze large numbers of CT scans with no scope of errors.

Artificial Intelligence (AI) tools such as deep learning have immense potential to process and draw insights from the massive amount of data [6]. Researchers have actively used advanced computer vision techniques and artificial intelligence tools to assist clinicians [7, 36, 38]. Alzubi et al. [7] used an ensemble of weight optimized neural network for automatic diagnosis of lung cancer. Liang et al. [39] have used a residual convolutional neural network to diagnose pediatric pneumonia using chest X-ray images. Inspired by the promising results, several studies have taken into account advanced AI technologies for diagnosing



(a) Variation of confirmed COVID-19 infections in Indian districts.



(b) Variation of confirmed COVID-19 infections in Indian metropolitan cities.

Fig. 1 Representation of newly registered cases of COVID-19 infection across India from April 2020 to August 2020

COVID-19 infection using chest radiographical images. Rahman et al. [48] have presented an extensive comparison of several image enhancement techniques and the U-Net model to detect COVID-19 infection. Gifani et al. [53] used an ensemble of five CNN architectures for screening of COVID-19 disease using chest CT scan images. Recently, Basset et al. [2] have investigated a semi-supervised learning approach for effective segmentation of COVID-19 infection in chest CT scan images. It was observed in the study that the proposed few shot learning based model generalizes well using a small number of training examples. Although conventional AI strategies provide invaluable aid to clinicians, most sophisticated deep learning architectures are computationally quite expensive for real-time applications.

Internet of Things (IoT) is the next-generation technology that can provide apt solutions to a broad spectrum of problems, be it traffic management, waste management, or health-care services [31]. Several studies have leveraged IoT and cloud computing technologies to develop remote health monitoring services in the past [4, 60]. However, these cloud-based systems do not efficiently provide the required latency for healthcare applications because of high time delay. Also, providing the sufficient bandwidth required to transfer a massive amount of image data to the cloud in remote areas is not feasible. An extremely reliable and low latency network configuration will be the prime requirement to efficiently train deep learning algorithms using chest CT scan images over the cloud [47].

A collaborative edge-cloud computing framework can be used as an effective solution for the issues mentioned above. Ali et al. [5] have used edge/fog computing architecture for the real-time prediction of traffic flow. Rahman et al. [46] have presented the application of edge computing framework for quality of life monitoring using deep learning. Kong et al. [37] developed an edge computing-based deep learning framework for real-time face mask detection. There have been few works to develop lightweight and efficient deep learning architectures for automatic screening of COVID-19 infection [35, 41]. However, these studies failed in investigating the model complexity and latency for the application in edge computing platforms.

The primary goal of this research work is to develop a compact yet sophisticated deep learning framework for remote COVID-19 diagnosis using chest CT scan images with minimum latency. The proposed model being highly efficient in terms of size and performance, can be easily deployed on low-power mobile and edge devices for a fast diagnosis process. To the best of our knowledge, no such framework has been proposed yet for COVID-19 management. The major research contributions of this study can be summarized as:

- This study has introduced a MobileNet V2 based fine-tuned transfer learning model for COVID-19 diagnosis using chest CT scan images.
- This study has introduced a collaborative edge-cloud framework for the management of COVID-19 in remote and rural areas.
- The performance of the proposed model is evaluated using a benchmark chest CT scan image dataset.
- The developed deep learning model takes approximately 43 msec to diagnose the chest CT scan images making it highly suitable for real-time screening.
- The proposed model was further optimized in a flat buffer format leading to a significant reduction in size without compromising the performance.
- The efficacy of the proposed model is compared with other state-of-the-art transfer learning models using McNemar's statistical test.

The rest of the paper is organized as follows: Section 2 summarizes the related state-of-the-art studies focusing on COVID-19 diagnosis. Section 3 provides the background

information related to the technologies used in this study. In Section 4, the proposed architecture for COVID-19 diagnosis is described. Section 5 presents the edge-cloud computing platform for COVID-19 management in remote and rural areas. Section 6 includes the results of the extensive experiments performed on the proposed model. Section 7 is the concluding section. It discusses about possible future improvements of this study.

2 Related work

Towards the end of the year 2020, researchers have proposed a significant number of state-of-the-art methods for COVID-19 diagnosis. In line with the objective of this research, we have presented few state-of-the-art studies for COVID-19 diagnosis using artificial intelligence tools in Table 1. It can be observed from Table 1 that most of the researchers have used pre-trained models and proposed transfer learning models for feature extraction and classification. Goldstein et al. [23] have used an ensemble of pre-trained ResNet 34, ResNet 50, ResNet 152, VGG 16 and CheXpert model for automatic diagnosis of COVID-19 using chest X-ray images. Abbas et al. [1] employed a deep Convolutional neural network (CNN) model called Decompose Transfer and Compose model for the classification of COVID-19 using chest X-ray images from multiple sources. ResNet 50 based transfer learning model was used by Pathak et al. [44] for identification of COVID-19 using chest CT images. Ardakani et al. [8] used an ensemble of machine learning classifiers, namely k-Nearest Neighbor (KNN), Decision Tree (DT), Naive Bayes (NB), and Support Vector Machine (SVM) for the diagnosis of coronavirus infection using twenty radiological features from chest CT images.

Chen et al. [12] used a set of radiological and clinical features from chest CT scan images for screening of COVID-19. But the study was performed on very limited data.

Due to the lack of benchmark datasets, researchers have used Generative Adversarial Networks to augment the training data. Jiang et al. [34] generated synthetic chest CT scan images using CGAN.

It can be observed from Table 1 that in most of the previous work, either the study is performed on a small dataset or the artificial intelligence techniques developed are too complex to be used in resource constraint environments; moreover they are relatively more time-consuming. Thus, this study aims to develop a compact deep learning-based framework for rapid diagnosis of COVID-19 infection, which is both efficient and performs on par with other state-of-the-art methods.

3 Background technologies

3.1 Cloud and edge computing for healthcare solutions

The significant increase in the number of electronic devices and the use of internet and web resources for various multiple reasons has posed a challenge to the infrastructure and resource management of the IT industry. In addition to this, the huge data being generated and the increased access of local servers has brought troubles to several service providers. In such circumstances, the idea of shared virtual resources and servers has become a boon, which is called 'CLOUD'. The overwhelmed data is transferred to these virtual servers and platforms. This sharing of data has made all the internet services accessible with ease to the

Table 1 State-of-the-art studies focusing on COVID-19 diagnosis

Study	Dataset	Methodology	Results/Remarks
Chen et al. [12]	CT scan: 70 COVID-19 and 66 Non-COVID	Used 41 radiological features and 26 clinical features	Area under curve: 0.986 Study is performed on limited data.
Song et al. [57]	CT scan: 777 COVID-19 and 505 bacterial pneumonia	Feature pyramid network on top of ResNet 50 transfer learning model (DRE-Net)	Accuracy: 86% Area under curve: 0.96 Accuracy score needs to be addressed.
He et al. [25]	CT scan: 349 COVID-19 and 397 Non-COVID	Proposed a self transfer learning model using DenseNet 169	Accuracy: 86% F-1 score: 0.85 Area under curve: 0.94 Accuracy score is not optimal.
Zhang et al. [62]	Two in house X-ray dataset X-VIRAL: 5977 viral and 18774 healthy	Confidence aware anomaly detection model (CAAD) X-COVID: 106 COVID-19 and 107 Non-COVID	Accuracy: 72.77% Sensitivity: 71.70% Area under curve: 0.836 Performance on X-COVID dataset needs to be improved.
Hemdan et al. [26]	X-ray: 25 COVID-19 and 25 Non-COVID	VGG 19 and DenseNet 201 based transfer learning model	Accuracy: 90% Sensitivity: 90% Time: 4 sec Failed to address the diagnosis time.
Ismael et al. [32]	X-ray: 180 COVID-19 and 201 Non-COVID	Deep transfer learning features extracted from ResNet 50 with linear SVM classifier	Accuracy: 94.7% Sensitivity: 91% Area under curve: 0.99

Table 1 (continued)

Study	Dataset	Methodology	Results/Remarks
Das et al. [16]	X-ray: 219 COVID-19, 1340 normal and 1345 pneumonia	VGG 16 transfer learning model	Accuracy: 97.67% Sensitivity: 96.54% Failed to minimize the computational cost.
Brunese et al. [10]	X-ray: 3003 Pulmonary diseases (250 COVID-19) 3520 healthy	Two VGG 16 based transfer learning models, first model to discriminate healthy cases from pulmonary diseases and second to discriminate COVID-19 cases from other pulmonary diseases	Accuracy: 97% (avg.) Sensitivity: 91% (avg.) Time: 2.5 s High computational cost.
Goldstein et al. [23]	X-ray: 1191 COVID-19 and 1135 Non-COVID	X-ray images are segmented using U-Net model and ensemble of four transfer learning models (ResNet 50, ResNet 152, CheXpert and VGG 16)	Accuracy: 90.5% Sensitivity: 88.8% Sensitivity score needs to be addressed.
Mishra et al. [42]	CT scan: 360 COVID-19 and 397 Non-COVID	Ensemble of VGG 16 Inception V3, ResNet 50, DenseNet 121 and DenseNet 201	Accuracy: 88.34% Sensitivity: 88% Ensemble of transfer learning model is not feasible to deploy on low power edge devices
Loey et al. [40]	CT scan: 345 COVID-19 and 397 Non-COVID	Increased training and validation data using Conditional Generative Adversarial Network (CGAN) and ResNet 50 transfer learning model	Accuracy: 82.91% Sensitivity: 77.66% False negative rate is not optimal.

users and less complicated to the providers. These cloud technologies even gave solutions to problems of Big Data like data analytics, computing, and storage.

Cloud computing (CC) is one of the prominent emerging technologies in recent applications as it provides additional computational abilities and various services on shared and virtual servers or hardware. These expanded computational capabilities have led to the compatibility of complicated machine learning and deep learning algorithms. The raw or unprocessed data is uploaded to the cloud, where further data processing and analytics are performed, thereby extracting efficient information from the data.

Despite all these benefits, Cloud Technologies have a few drawbacks in terms of usage. The latency is one issue where the delay is increased in uploading the data to the cloud servers and getting back the results. Data security is also a significant concern as the resources are shared among users, and there is lack of confidentiality. Information regarding the processing operations of data by the cloud providers is not disclosed to the users. Complete access to the data is not provided to the clients by the cloud service providers. Internet access to obtain the services of CC needs to be better; a weak or inconsistent connection may bring more trouble and hindrance to the clients.

With these drawbacks, there is the necessity to perform maximum possible computations at the edge level closer to the user to make things easy and quick. Edge computing is a new paradigm that can be used to limit the amount of data being sent over the network and thereby improve the speed of the communication links [55]. Edge computing refers to the empowering technologies that allow maximum possible computation in proximity to the source of information.

Gaura et al. examined the benefits of data mining at the edge of the network for three algorithms, namely: Linear Spanish Inquisition Protocol (L-SIP), ClassAct, and Bare Necessities (BN) [20]. They found a significant reduction in the network's energy requirements and hence extension of battery life because of reduced data packet transmission. The comparison of edge and cloud computing on several metrics is shown in Fig. 2. Hence, we can clearly understand the necessity of Edge computing in Healthcare applications where the data is a matter of life and death and requires much faster and safer computations and analytics. In the applications of edge computing, the placement of edge servers also play a prominent role. Wang et al. addressed this in their study of edge server placement in mobile edge computing networks for smart cities [59]. Assuming the edge servers to be homogenous, they proposed a mixed integer programming (MIP) edge server placement algorithm.

With the advancements in information and communication technologies, several studies are carried out to design a smart healthcare system. The general architecture of a connected smart healthcare system is shown in Fig. 3. The smart healthcare systems generally consist of IoT devices and collaborative edge-cloud computing with advanced AI technologies deployed over them.

Muhammad et al. proposed a smart healthcare framework using edge and cloud computing for voice disorder assessment and treatment [43]. The proposed system was reported to have over 99% sensitivity. Mamun et al. proposed a cloud-based framework for detection and monitoring patients with Parkinson's disease [3]. The feedforward backpropagation based artificial neural network model used in the framework gave an accuracy of 97.37% in the diagnosis of patients. Sacco et al. developed an edge-computing based telepathology system for real-time image processing of histological images [50]. In their experiments, they found the latency of the system sufficiently less for real-time applications.

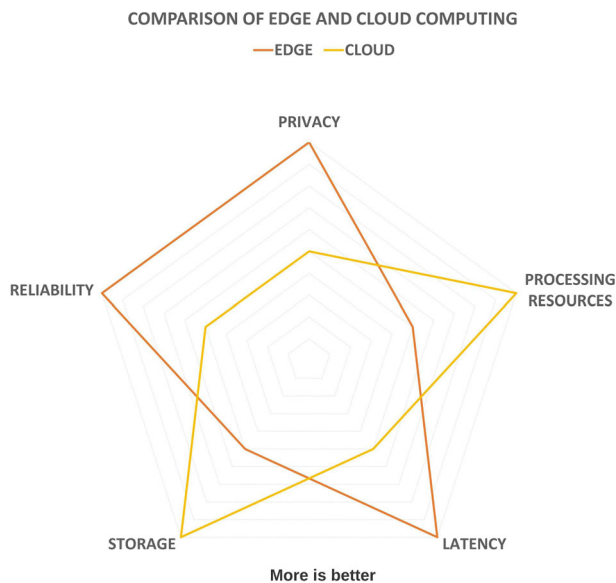


Fig. 2 Radar plot comparing the edge and cloud computing technologies in terms of privacy, resources, latency, storage, and reliability metrics

The above-discussed studies provided an efficient clinical diagnostic system that can be used as a second opinion by healthcare professionals. There have been researches to make the complex AI algorithms compatible with edge levels making the devices smarter to avoid latencies and provide much better security with no access concerns. To deploy a machine learning model on edge devices, it needs to be optimized to a comparable size and complexity.

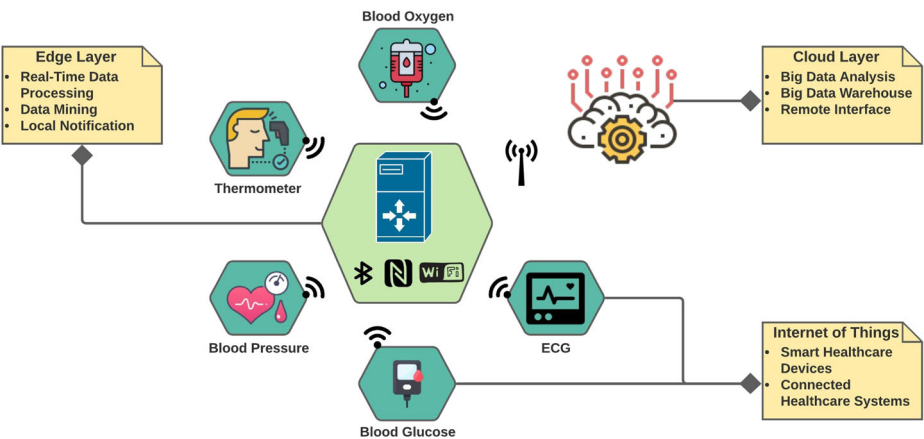


Fig. 3 Typical connected smart healthcare system architecture, which includes IoT devices, edge layer, and cloud layer

3.2 Deep learning

One of the most critical advancements in the field of computer science, which revolutionized the data mining industry, is undoubtedly deep learning. Deep learning took almost two decades to become this mature with the rise in availability of publicly accessible data, powerful parallelization of Graphic Processing units (GPUs), and development of deep learning specific hardware such as Google's Tensor Processing units (TPUs). Deep learning-based frameworks are widely used in various pattern recognition and classification applications such as image recognition [63], music recognition [21], and biomedical disease segmentation [58]. These frameworks have achieved state-of-the-art performance in majority of the applications. The human nervous system inspires many of the deep learning algorithms.

CNNs are the default choice for vision-based applications thanks to their translational invariance property. These networks consist of several convolutional and pooling layers stacked together, which extract high-level features from the image. Finally, these features are classified by one or more fully connected layers. The absence of high processing power in edge devices is the major challenge faced while deploying deep learning frameworks on edge computing platforms.

3.3 Transfer learning

Even though deep learning has gained immense success in various real-world applications, it has several limitations restricting its usage in certain applications. One such limitation is the need for a massive amount of labeled data for training of the parameters. But in the fields such as biomedical imaging, creating such a large labeled dataset is often unrealistic. Training a deep neural network from scratch using limited data indeed leads to the overfitting problem. Transfer learning is the process of sharing the knowledge gained while doing a particular task to solve different tasks, as shown in Fig. 4. Intuitively we can think of a model that is trained to classify images of mangoes and apples; the knowledge of this model can be used further to classify mangoes and pears. ImageNet dataset consisting of more than 14 million images of real-life subjects has played a significant role in popularizing the concept of transfer learning. [49].

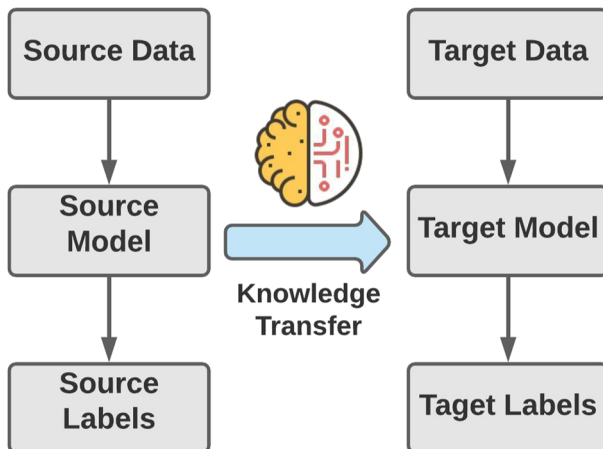


Fig. 4 Knowledge transfer process in transfer learning

There are two ways in which transfer learning is incorporated in convolutional neural networks:

1. The pre-trained transfer learning model can be used as a feature extractor. And further, a strong classifier can be trained to perform classification using the transfer learning features.
2. The weights learned by the pre-trained network using source data can be updated by employing target data and the backpropagation algorithm. As the pre-trained weights were already optimized, the model converges very quickly. This process of updating the weights is known as fine-tuning.

4 Proposed transfer learning framework

The proposed fine-tuned transfer learning framework consists of three main components: pre-processing, fine-tuned MobileNet V2 [51] model, and finally, a classifier. Figure 5 describes the proposed framework. When a chest CT scan image is given input to the framework, it pre-processes the image according to MobileNet V2 architecture requirements. MobileNet V2 architecture only supports colored images of resolution 224×224 .

In the second stage of the framework, the MobileNet V2 model pre-trained on the ImageNet dataset is used. In this study, we have employed MobileNet V2 because of the following reasons:

- The training process of the MobileNet V2 model is fast as it requires significantly fewer parameters to train compared to other state-of-the-art models.
- MobileNet V2 is one of the very few deep learning models suggested to be used for resource constraint applications on mobile and edge devices.

Finally, a fully connected dense layer with sigmoid activation is used for binary classification.

4.1 MobileNet V2 architecture

Howard et al. [27] introduced a deep learning architecture MobileNet V1 for devices with constrained computational power. This model strives to provide the user with state-of-the-art performance while being computationally efficient. With MobileNet V1, depthwise convolutions were introduced, which splits a standard convolution operation in two parts: a) a depthwise convolution in which each channel of the input image is convolved by a

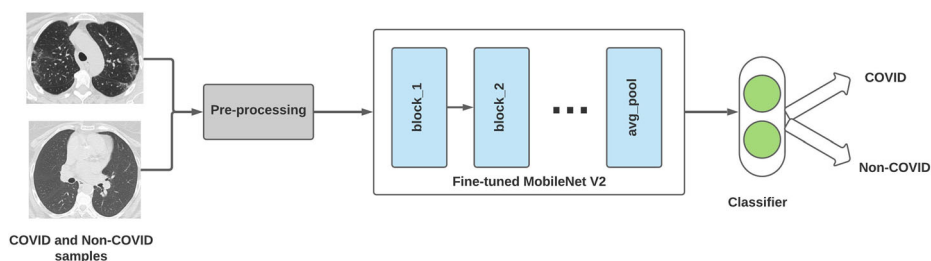


Fig. 5 The proposed transfer learning framework for screening of COVID-19 infection using chest CT scan images

single filter, b) a pointwise convolution, which computes the weighted sum of each convolution obtained. The pictorial representation of a depthwise separable convolution process is shown in the Fig. 6. This splitting of the standard convolution results in significantly lesser parameters to train.

The computational cost of any experiment is defined in terms of the number of multiplications performed. Let F represent an input feature map of dimension $D_F \times D_F \times M$, where M is the total input channel. In a standard convolution with N filters each of dimension $D_K \times D_K \times M$, the computational cost is expressed as:

$$D_K \times D_K \times D_F \times D_F \times M \times N \quad (1)$$

While in the case of depthwise separable convolution, the cost of computation is the addition of the individual computation cost of both depthwise convolution and pointwise convolution.

In depthwise convolution with M filters each of dimension $D_K \times D_K \times 1$, the computational cost is expressed as:

$$D_K \times D_K \times D_F \times D_F \times M \quad (2)$$

In pointwise convolution with N filters each of dimension $1 \times 1 \times M$, the computational cost is expressed as:

$$D_F \times D_F \times M \times N \quad (3)$$

The total computational cost for the depthwise separable convolution is expressed as:

$$D_K \times D_K \times D_F \times D_F \times M + D_F \times D_F \times M \times N \quad (4)$$

Ratio of computational cost of depthwise separable convolution to the computational cost of standard convolution can be given as:

$$\frac{M \times D_F^2 \times (N + D_K^2)}{D_K^2 \times D_F^2 \times M \times N} = \frac{1}{N} + \frac{1}{D_K^2} \quad (5)$$

Apart from depthwise convolution, Sandler et al. [51] introduced inverted residual connection in the MobileNet V2 architecture. The shortcut connections between the bottlenecks are similar to the skip connections introduced in the architecture of residual networks, allowing networks to go deep without affecting the overall performance by preserving the low-level features [24]. The basic building block of MobileNet V2 architecture is shown in Fig. 7.

The building block consists of three convolutional layers: a 1×1 expansion layer, a 3×3 depthwise convolution, and a 1×1 projection layer. When a feature map is given as input to the block, the expansion layer increases the dimension of the feature map according to the value of expansion factor (default=6). The depthwise convolution applies a convolution

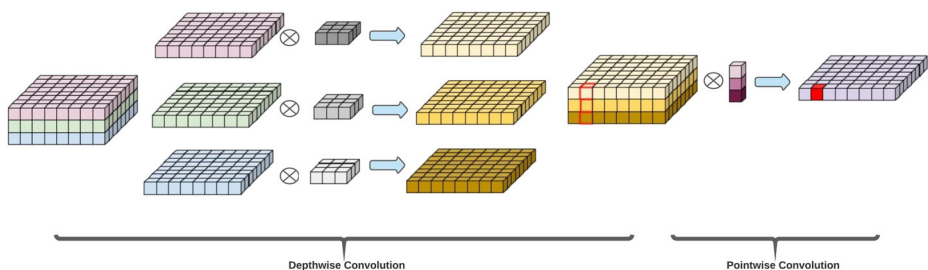


Fig. 6 Concept of Depthwise separable convolutions

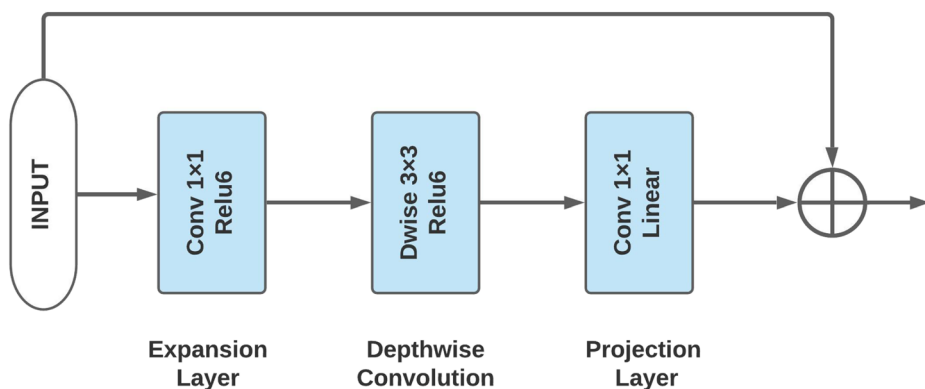


Fig. 7 Basic building block consisting of an expansion layer, a depthwise convolution, and a projection layer in MobileNet V2 architecture

filter to each channel, as shown in Fig. 6. Finally, the projection layer projects the higher dimensional data into a lower dimension, which is then passed to the next block.

The introduction of residual blocks in MobileNet V2 ensures that the feature maps flowing across the blocks do not have a very high dimension. The higher the dimension of feature maps flowing across the blocks, the higher will be the computational complexity. But to extract more abstract features from the input using depthwise convolution, the feature maps are expanded by the amount of expansion factor. In each block of MobileNet V2, the expansion layer helps to learn more abstract features by increasing the dimension of input feature map while the projection layer helps to reduce the computational complexity by reducing the dimensions of the feature map again.

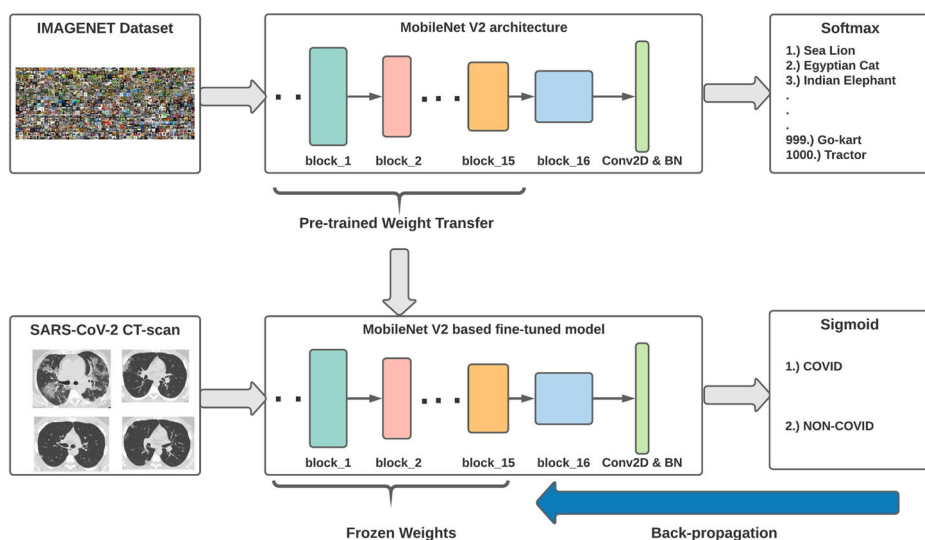


Fig. 8 The proposed MobileNet V2 based fine-tuned model. The figure depicts the transfer of knowledge learnt from the ImageNet dataset for the application of COVID-19 diagnosis

4.2 Fine-tuning MobileNet V2

In this study, considering the limited amount of image data and limited computational resources, only the weights of the last convolutional block of the MobileNet V2 architecture is fine-tuned. The final dense layer with a softmax activation function in the original MobileNet V2 architecture is replaced with a dense layer with a sigmoid activation function for binary classification. The sigmoid function is mathematically expressed as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x} \quad (6)$$

Figure 8 illustrates the fine-tuning process of MobileNet V2 architecture incorporated in this study. The algorithm used to fine-tune the MobileNet V2 architecture is described in Algorithm 1.

Algorithm 1 Fine-tuning MobileNet V2 architecture.

Input: Labeled training samples, $S=(x_k, y_k)_{k=1}^N$
 Pre-trained transfer learning model, $T=\text{MobileNet V2}$
 Number of epochs, $\xi = \xi_i$
 Learning rate = l
Output: Fine-tuned MobileNet V2 model
Function `fine_tune` (S, T, ξ, l):
 `parameters` \leftarrow `load_model(model=T, pre_trained=True, weights='ImageNet')`
 `parameters` \leftarrow `initialize(parameters, layer=block_16)`
 foreach ξ **do**
 foreach k **do**
 `activations` \leftarrow `forward_propagation_function(x_k, parameters)`
 `loss` \leftarrow `loss_function(activations, y_k)`
 `gradients` \leftarrow `back_propagation_function(activations, loss)`
 `parameters` \leftarrow `update_weight(parameters, gradients, l)`
 end
 end
End Function

5 Proposed collaborative edge-cloud computing framework

The proposed deep learning-based, collaborative edge-cloud computing framework for COVID-19 management is shown in Fig. 9. The proposed framework is a three-tier architecture consisting of the data generation layer, edge layer, and cloud layer. The data required for predicting and managing the COVID-19 is generated in the bottom layer. The collected data is then transferred to the edge layer consisting of edge devices with a pre-installed deep learning framework. These devices are capable of alerting the concerned authorities if any individual is diagnosed as COVID positive. A unique patient ID, along with the location information, is sent to the cloud layer to track and manage the spread of the virus. The subsequent subsections discuss each layer of the proposed framework.

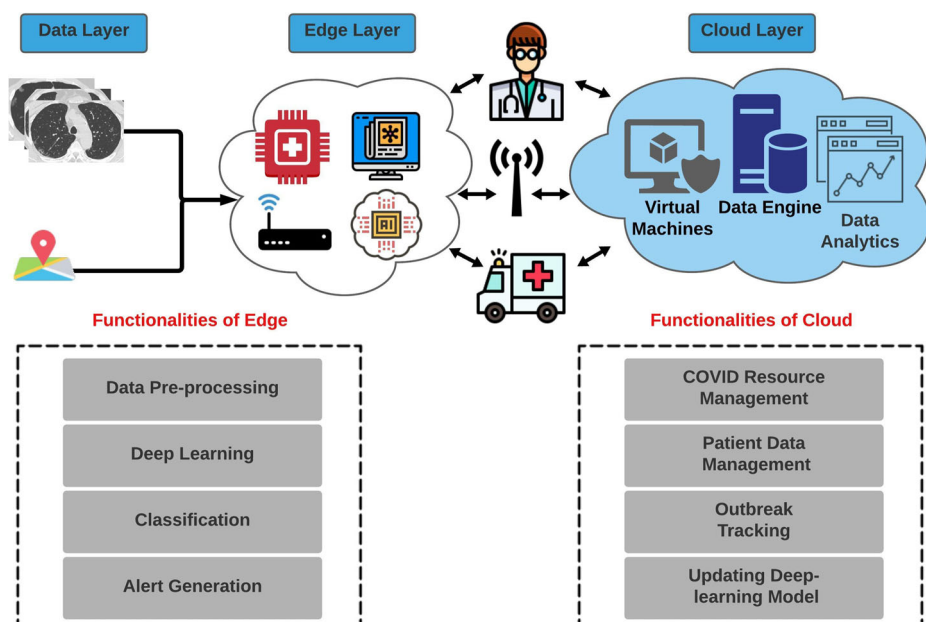


Fig. 9 Collaborative edge cloud computing framework

5.1 Data generation layer

This layer is where all the data used for COVID-19 management is generated. This study considered chest CT scan images of patients as the data required to diagnose patients as COVID positive or COVID negative. Furthermore, the patient's location data is also taken to facilitate government authorities in providing resources and services to the patient. These datasets are then transmitted to the edge layer consisting of edge devices at the location of the patient itself for further processing and classification.

5.2 Edge layer

The edge layer in this framework is used for processing and classifying the CT scan data received from the bottom layer. It also acts as a bridge between the physical data layer and the distant cloud layer. The edge layer provides low latency and reduced data traffic. The edge layer consists of devices with low computational power, which can also act as IoT devices installed with the proposed deep learning model. Separate hardware accelerators can also be used along with the edge devices to make the computation faster.

If any patient's chest CT image is diagnosed as COVID-19, an alert is sent to the concerned officials. A unique patient ID consisting of patient information and the location information is sent to the cloud layer for further operations.

5.3 Cloud layer

Due to limited data storage and computation power available on edge devices cloud layer is used to store data and perform various centralized tasks. The centralized functions in

the cloud layer are performed over multiple Virtual Machines (VMs). The data is stored in a central data warehouse, which can be accessed only by authorized medical caregivers and government authorities. In this proposed framework cloud layer is used to perform the following functions:

1. **COVID-19 Resource Management:** With an exponential increase in the number of cases, there is an acute shortage of critical medical tools worldwide. Medical devices such as ventilators and respiratory devices play a vital role in treating this disease; hence proper management of this equipment is required. There should be a right balance between demand, supply, and production of these types of equipment. For example, there are specific locations where there is an outbreak of COVID, and hospitals at these locations require more supply. The cloud can help in tracking the outbreak location, number of cases and optimizing the resources.
2. **Patient Data Management:** The cloud helps in storing the data regarding the number of patients being infected, the number of active cases, the number of recoveries, and the number of deaths due to the virus. This information needs to be continuously monitored and analyzed to obtain regular situation updates supporting necessary advancements in treating the patients and controlling the spread of the virus.
3. **Outbreak Tracking:** As the spread and infection rate of the virus is very high, it is necessary to measure the intensity of the virus outbreak in every locality and identify hotspots. The cloud data is regularly updated with the newly identified cases and their location information. This information can be made publicly available, thereby alerting the public and civic bodies.
4. **Provide Model Updates:** As the disease is novel, there is limited availability of data to train the deep learning model. Training the model on edge is very complex; hence the model can be updated on the cloud, and the updates can be downloaded on edge devices. This provision of getting updated over-the-air makes the system future proof for other health related applications.

Algorithm 2 Diagnosis of COVID-19 at Edge layer.

Input: Chest CT images and location data

Output: Classification result of input chest CT image

Step 1: Get the chest CT images from the data generation layer and generate a unique patient ID.

Step 2: Pre-process the chest CT images obtained from data generation layer to match with the size requirements of the deep learning model.

Step 3: Feed the image to the deep learning classifier and obtain classification result.

Step 4: if *Predicted Result* = *COVID* then

Step 4.1: Display the diagnosed result to the user along with the patient ID for future reference.

Step 4.2: Immediately send the patient ID and location information to cloud and alert the concerned authorities for followup with the patient.

else

Step 4.1: Display the diagnosed result to the user with a suggested re-diagnosis date.

Step 4.2: Delete the generated patient ID and location information from the memory.

end

Step 5: Exit

The proposed framework diagnoses the patients as COVID positive or COVID negative, taking input as their raw chest CT images using the process shown in Algorithm 2.

The proposed framework requires limited bandwidth as only the individual's location and identity information are uploaded to the cloud. Also, the delay in communication is hugely reduced compared to cloud-dependent architectures, thereby providing much quicker responses and results. The edge layer incorporated provides higher data security as the personal health data is not shared with any other third parties as in the cloud architectures. The diagnosis of COVID-19 infection becomes much accessible in remote areas without any significant network usage with the proposed framework.

6 Simulation results and discussion

The simulations presented in this article are performed on a HP personal computer with Intel® Core™ i5-5200 CPU and 4GB of RAM using the Python environment provided by Google Colaboratory. The TensorFlow platform is used to implement deep learning models, which offer various Application Programming Interfaces (APIs) for data analysis and data mining.

6.1 Dataset used

The chest CT images used in this study for experimentation are taken from the publicly available SARS-CoV-2 CT-scan dataset [52]. This dataset includes 1252 chest CT images of patients diagnosed as COVID positive and 1230 chest CT images of patients diagnosed with other pulmonary diseases. Figure 10 shows sample chest CT images from the dataset. We have selected the chest CT image as the dataset for diagnosing COVID-19 in this study since CT scans have proved to be better than X-ray for detecting pneumonia, which is a significant symptom of COVID-19 infection.

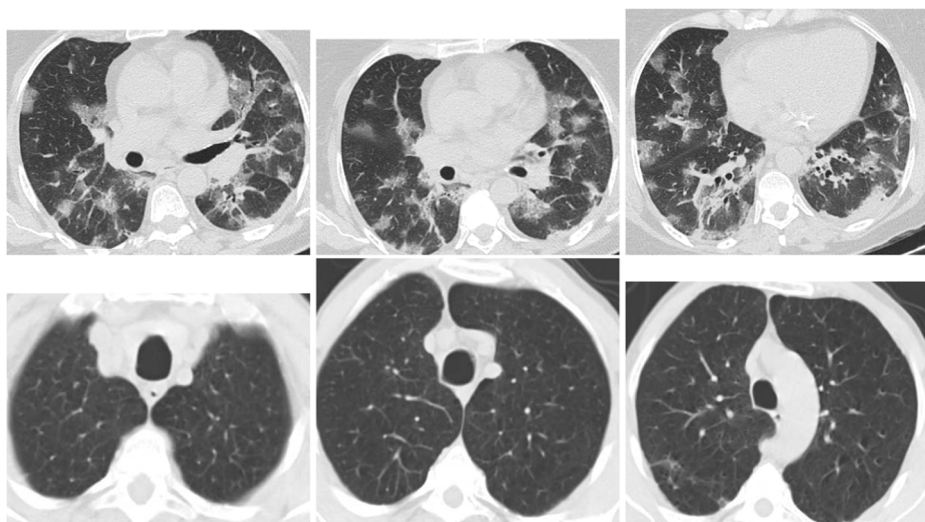


Fig. 10 Few sample images of chest CT scan from the dataset

6.2 Pre-processing

The dataset was divided in three directories:

1. Training Data: Randomly selected 1580 images from the dataset are added in this directory for the training of deep learning models.
2. Validation Data: Randomly selected 200 images from the dataset of each class are added in this directory to validate the model after every epoch.
3. Testing Data: This is the data used to evaluate the performance of the deep learning model. The model has never seen these images while training. Randomly selected 250 images of each class are added in this directory.

From Fig. 10, we can see that the images come in various sizes. All the images were brought down to a resolution of $224 \times 224 \times 3$ before feeding in the model.

6.3 Assessment metrics

Evaluation of deep learning models in healthcare applications cannot be done solely based upon the accuracy score. The proposed deep learning models must reduce false negatives and false positive events in the classification process. Diagnosing a COVID-19 infected patient as a Non-COVID patient would be very detrimental to society. Hence in this study, we compared all the transfer learning model based upon the following metrics:

1. Sensitivity (Se) : Sensitivity is the measure depicting the true positive rate of a classifier. It shows the ability of a classifier to predict true positive events correctly. It is also known as Recall. It is represented as:

$$Se = \frac{TP}{TP + FN} \times 100\% \quad (7)$$

2. Specificity (Sp) : Specificity is the measure depicting the true negative rate of a classifier. It shows the ability of a classifier to predict true negative events correctly. It is represented as:

$$Sp = \frac{TN}{TN + FP} \times 100\% \quad (8)$$

3. Precision (P) : Precision is also known as Positive Predictive Value (PPV). It shows how many positive predictions of the classifier were actually positive. It is represented as:

$$P = \frac{TP}{TP + FP} \times 100\% \quad (9)$$

4. F1-score : F1-score is the harmonic mean of both the precision and sensitivity. It can be represented as:

$$F1 - score = \frac{2 \times Se \times P}{Se + P} \quad (10)$$

5. Accuracy (Acc) : Accuracy is the fraction which the classifier can correctly predict. It is given as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (11)$$

6. Matthews correlation coefficient (MCC) : Being a Correlation coefficient MCC value lies in the range of $[-1, 1]$. It is the only statistical performance metric that can give high

scores if the model correctly predicts most of the positive and negative samples from the testing dataset [13].

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (12)$$

Where, TP, TN, FP, and FN denote true positive, true negative, false positive and false negative events respectively.

6.4 Transfer-learning model evaluation

The deep learning model developed in this study is trained using the backpropagation algorithm. For the convergence of the deep learning model, proper selection of optimizer is essential. Many optimizers have previously been used in deep learning literature, such as Stochastic Gradient Descent (SGD), RMSprop, and Adaptive Moment Estimation (Adam). Considering the advantages of adaptive optimizers [61], we have selected Adam optimizer with an initial learning rate of 10^{-4} in this work. The batch size was kept as twenty with 79 steps to facilitate the computer's memory limitations. The model was trained for 100 epochs.

In the first experiment, a baseline model of MobileNet V2 was developed. All the convolutional layers were set as non-trainable with weights learned from the ImageNet dataset. This baseline model achieved an accuracy score of 85.6%.

In our second exploration, weights of the last convolutional block, i.e., block_16 of the MobileNet V2 architecture is fine-tuned following the architecture shown in Fig. 8 in Section 4. This MobileNet V2 based fine-tuned transfer learning model performed significantly better than the baseline model by correctly predicting 96.40% of the test images.

Further, the performance of the developed deep learning model is compared with three other benchmark CNN architectures, namely VGG 16 [56], VGG 19 [56], and DenseNet 201 [29], based on the metrics mentioned above. There should be no biasing in the performance comparison; hence we selected the models having the input resolution of $224 \times 224 \times 3$.

The confusion matrices corresponding to each model are illustrated in Fig. 11. From Fig. 11, it can be observed that the proposed model is able to correctly diagnose 246 (True Positive) out of 250 COVID-19 cases giving an accuracy score of 98.4%. The DenseNet 201 [29] transfer learning model performed worst by correctly diagnosing only 230 (True Positive) out of 250 COVID-19 cases with an accuracy score of 92%.

The overall performance of any classification model can be evaluated in terms of accuracy, F-1 score, precision, and Matthews correlation coefficient. A comparative analysis of the overall performance achieved by all the developed transfer learning models is shown in Fig. 12. Figure 12 clearly depicts the superior performance of the proposed model as compared to other transfer learning models. The MCC value achieved by the proposed model is 0.929, which is the highest among all the models, which signifies that the predicted labels of the proposed model are highly correlated to the actual labels.

The clinical diagnostic tests currently being employed suffer significantly in the aspect of false negative rate. The false positive rate and false negative rate of a model can be visualized using the specificity and sensitivity scores. In Fig. 13, the sensitivity and specificity scores of all the transfer learning models are illustrated. Figure 13 shows that the proposed model has a significantly higher sensitivity score than other models that signify the least false negative rate.

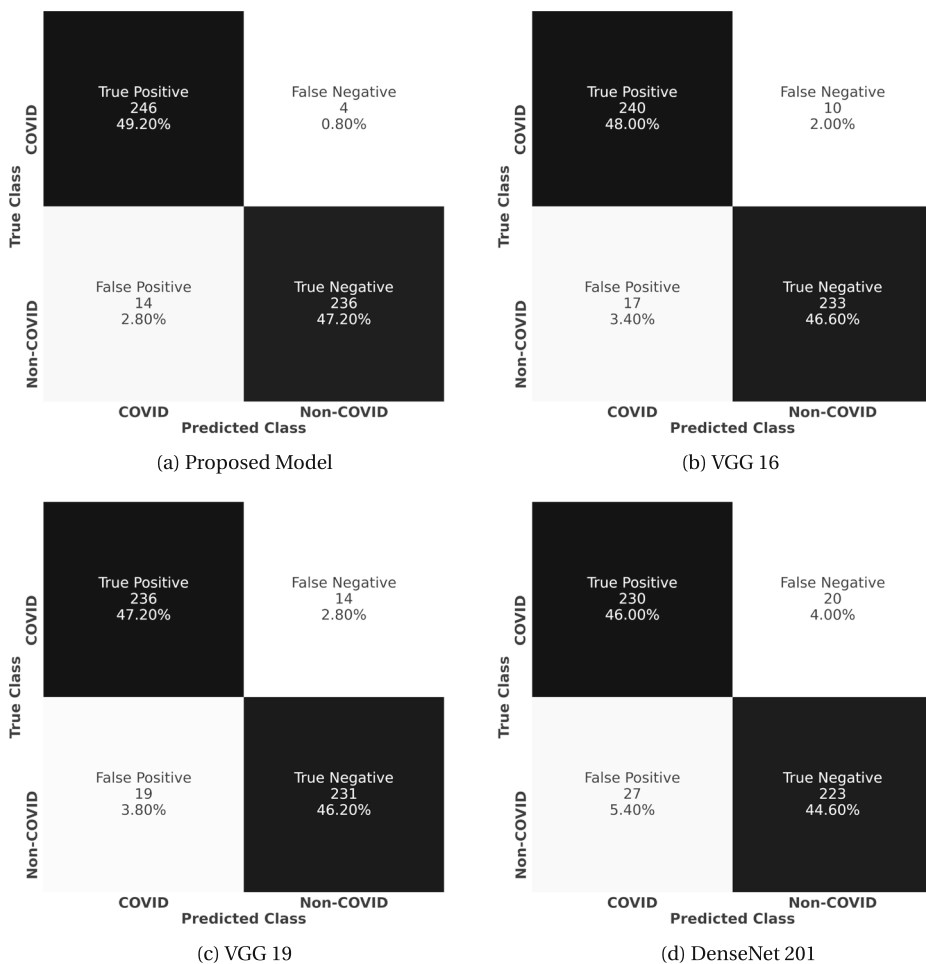


Fig. 11 Representation of confusion matrices of each transfer learning model for the prediction of COVID-19 infection on the test data

The Receiver Operating Characteristic (ROC) curve and respective Area Under Curve (AUC) value of all the developed models are shown in Fig. 14. In the ROC curve, true positive rate (sensitivity) is represented along the X-axis while false positive rate (1-specificity) is represented along the Y-axis. For an ideal classification model, the true positive rate should be maximized, keeping the false positive rate as minimum as possible. Figure 14 shows that the ROC curve of the proposed model is closest to the top left corner signifying superior performance.

The primary goal of this research work is to develop a deep learning model for edge devices to reduce the diagnosis time. The time taken by the deep learning models to diagnose 500 chest CT images present in the testing dataset is shown in Fig. 15. From this figure it is evident that the proposed model is approximately ten times faster than VGG 16 model and fifteen times faster than VGG 19 model. The proposed model takes an average of 43 msec to diagnose an input chest CT image as COVID-19 or Non-COVID.



Fig. 12 Comparison of overall performance in terms of accuracy, precision, F-1 score, and MCC values across VGG 16, VGG 19, DenseNet 201, and proposed model

6.5 Compatibility of proposed model on edge devices

To mitigate the storage issue associated with edge and mobile devices, the proposed model is further optimized in a flat buffer format. This reduced the size of the proposed model significantly, making it compatible with majority of edge and mobile devices having limited computational capabilities. A comparative analysis of all the models based on their size is shown in Fig. 16. In this study, we used the TensorFlow Lite API, which is provided by the TensorFlow platform with its default options to optimize the proposed MobileNet V2 based fine-tuned model in terms of its size and latency.

To perform inference using the model, we used Python Interpreter. The inferences obtained by the optimized model are compared with the original proposed model. In most cases, the confidence obtained by both models is similar to each other. Few sample

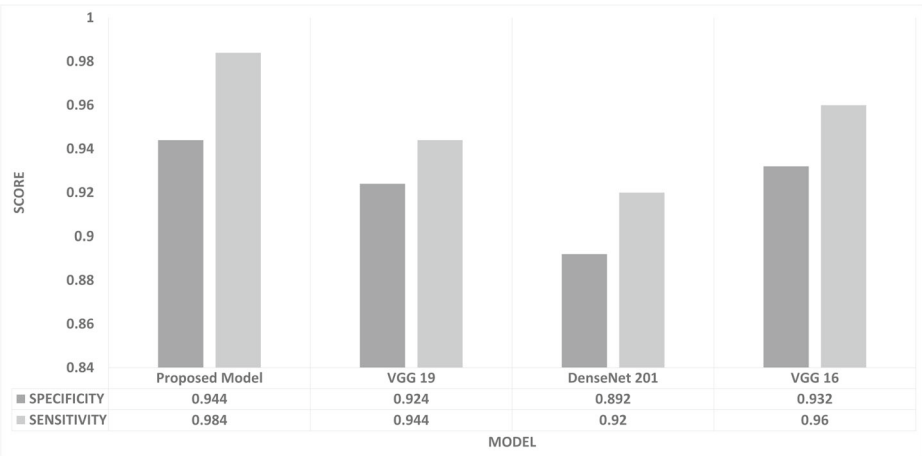


Fig. 13 Comparison of sensitivity and specificity scores of each transfer learning

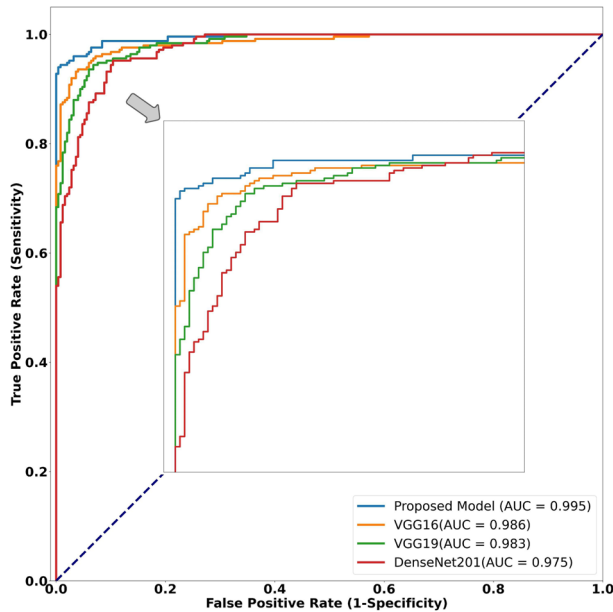


Fig. 14 Representation of performance in terms of ROC curve for all developed transfer learning models. The plot is zoomed from the top left corner for better visualization

inferences with their respective input images are shown in Fig. 17. It is depicted by the Fig. 17 that even after decreasing the size of the proposed model by more than 40% for edge deployment, it still performs exceptionally well in classifying chest CT images. Having a size of just about 8 MB, this model can be easily deployed on edge and mobile devices.

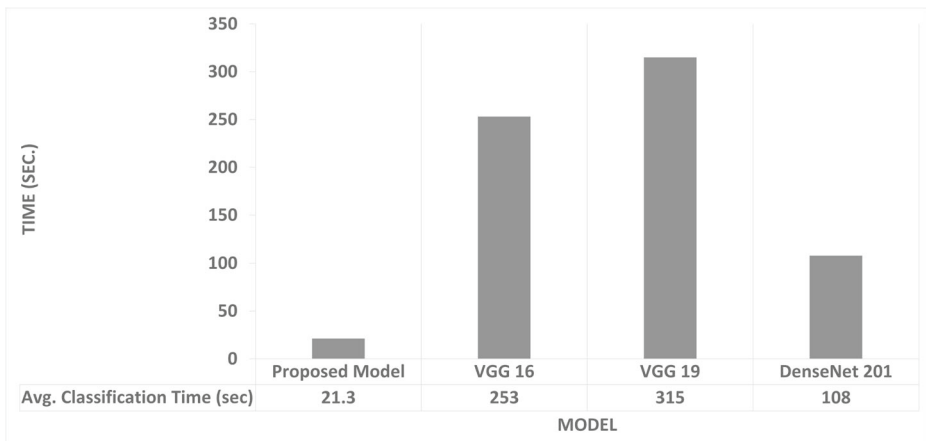


Fig. 15 Analysis of the time complexity for each transfer learning model in terms of average time taken for classification of test data images

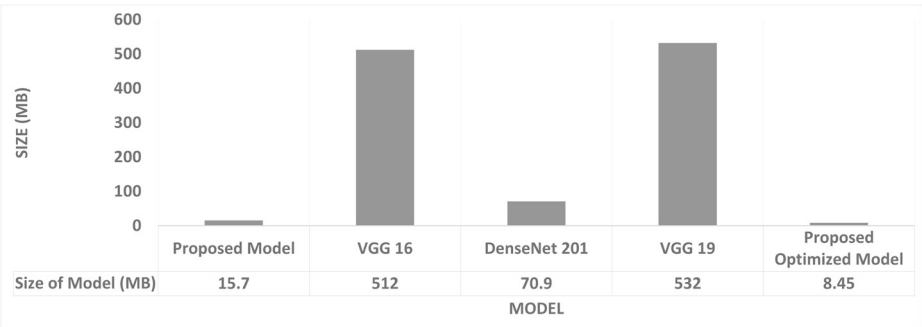


Fig. 16 Analysis of the hardware complexity for each transfer learning model in terms of disk size taken by the trained model

6.6 Statistical significance test

To statistically verify the superiority of the proposed model over other transfer learning models, we used a paired test called McNemar’s test [19]. McNemar’s test is a paired hypothesis test that initially assumes a null hypothesis (H_0) that the error rates of two

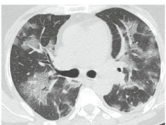
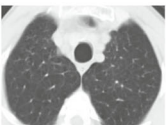


CT Image	Actual Class	Inference of Proposed Model	Inference of Proposed Optimized Model	Predicted Class
	COVID	0.000261	0.000261	COVID
	NON-COVID	0.999994	0.999994	NON-COVID
	COVID	0.967825	0.967825	NON-COVID
	NON-COVID	0.066904	0.066905	COVID

Fig. 17 Few sample inferences obtained from the model using Python Interpreter

different classification models are equal. This test is suggested primarily for the experiments having computational limitations, and repeated training of models is not possible [18].

For the same testing dataset, after obtaining the classification results from two different classification models, a contingency table is constructed as shown in Fig 18. In Fig. 18: n_{AB} is the numerical value corresponding to the number of test cases that both the models correctly predict

$n_{A'B}$ is the numerical value corresponding to the number of test cases that are correctly predicted by model B but is misclassified by model A.

$n_{AB'}$ is the numerical value corresponding to the number of test cases that are correctly predicted by model A but is misclassified by model B.

$n_{A'B'}$ is the numerical value corresponding to the number of test cases that are misclassified by both model A and model B.

The McNemar's test statistic value is expressed as:

$$\chi^2 = \frac{(|n_{A'B} - n_{AB'}| - 1)^2}{n_{A'B} + n_{AB'}} \quad (13)$$

For 95% confidence interval and 1 degree of freedom, $\chi^2_{1,0.95} = 3.8158$ or $\alpha = 0.05$.

If the p-value obtained from McNemar's test is greater than the threshold value α ($p > \alpha$), the test fails to reject the null hypothesis [30]. On the other hand, if the obtained p-value is less than the desired threshold ($p < \alpha$), the test rejects the null hypothesis hence proving that there is a difference in the error rates of both the classification models

The McNemar's test statistic value and the p-value of the comparison models are shown in Table 2. From Table 2, it can be observed that McNemar's test failed to reject the null hypothesis in the comparison of the proposed model and VGG 16. This signifies that even though the classification accuracy is different for both models, there is no significant difference in the error rates of the models statistically. But considering other factors such as less diagnosis time, high sensitivity, and compact size, the proposed model, seems to be a perfect candidate for remote diagnosis of COVID-19.

6.7 Comparison with state-of-the-art methodologies

Table 3 shows the comparison of the proposed model with other state-of-the-art studies concerned with COVID-19 diagnosis. This study intends to propose a lightweight deep learning model that can be deployed on edge devices. To encounter the computational limitations of

Fig. 18 Representation of contingency matrix developed in McNemars statistical test

		Model B	
		correct	incorrect
Model A	correct	n_{AB}	$n_{AB'}$
	incorrect	$n_{A'B}$	$n_{A'B'}$

Table 2 Results of McNemar's statistical test on the test data

Comparison	n_{AB}	$n_{AB'}$	$n_{A'B}$	$n_{A'B'}$	χ^2	p-value	Remarks
Proposed Model vs. VGG 16	459	23	14	4	1.73	0.188	Failed to reject the null hypothesis
Proposed Model vs. VGG 19	452	30	15	3	4.35	0.036	Reject the null hypothesis
Proposed Model vs. DenseNet 201	442	40	11	7	15.37	8.83×10^{-5}	Reject the null hypothesis

all the edge devices, it is necessary to validate the deep learning model in terms of its size and diagnosis time. Hence, apart from comparing the models with their respective accuracy and sensitivity scores, we also considered a comparison based upon the diagnosis time and the number of parameters in the original architecture, which signifies the model's complexity and size. Table 3 depicts the superior performance of the proposed transfer learning model in terms of accuracy, sensitivity, and diagnosis time despite its compact size.

The proposed MobileNet V2 based fine-tuned transfer learning model has the following advantages:

- The proposed model can itself extract relevant features from the chest CT scan images for diagnosis.
- Once the model gets trained with the chest CT scan images, it can diagnose COVID-19 infection in a few milliseconds.
- The proposed model is highly efficient and can be easily deployed on mobile and edge devices.
- The proposed model eliminates the need for subjective and time-consuming analysis of chest CT scan images by clinicians.

Though the proposed deep learning model is performing reasonably well in COVID-19 diagnosis, the robustness of the proposed model can be improved further with the availability of more chest CT image datasets. Also, by including other health-related data such as body temperature and peripheral oxygen saturation, i.e., S_pO_2 can help the model in further diagnosing the severity of COVID-19 infection.

Table 3 Comparison of proposed model with state-of-the-art methodologies

Study	Images	Model	Acc.	Se.	Avg. time (s)	Parameters
Azemin et al. [11]	X-ray	ResNet 101	0.72	0.77	0.132	44.71M
Jaiswal et al. [33]	CT	DenseNet 201	0.96	0.96	–	20.24M
Brunese et al. [10]	X-ray	VGG 16	0.97	0.91	2.5	138.35M
Song et al. [57]	CT	DRE-Net	0.86	0.96	–	–
Mishra et al. [42]	CT	Fusion	0.88	0.88	0.136	–
Gianchandani et al. [22]	X-ray	Ensemble	0.96	0.96	–	–
Hemdan et al. [26]	X-ray	VGG 19	0.90	0.90	4	143.67M
Pathak et al. [44]	CT	ResNet 50	0.93	0.95	–	25.64M
Proposed Model	CT	MobileNet V2	0.96	0.98	0.043	3.54M

7 Conclusion

In this paper, a novel deep learning model is proposed for COVID-19 diagnosis that can be easily deployed on edge devices. A MobileNet V2 based fine-tuned transfer learning model is proposed and trained with chest CT scan images. The proposed model outperformed other benchmark transfer learning models, achieving an accuracy, sensitivity, and MCC value of 96.4%, 98.4%, and 0.929, respectively, taking an average diagnosis time of 43 msec per image. The proposed model is further optimized to a size of 8MB. Compared with the original proposed model and other transfer learning models, the size of the optimized model is the minimum and best suitable for edge deployment. The experimental results depict that the proposed model is highly efficient for remote diagnosis of COVID-19 using collaborative edge-cloud computing platforms and can be used as an effective alternative to the time consuming clinical diagnostic tests. We intend to find the right set of hyperparameters using genetic algorithm in the future. Also, we look forward to augmenting the training data using GANs.

Acknowledgements The authors would like to acknowledge Dr. Smriti Singh from the Department of Humanities and Social Sciences at the Indian Institute of Technology Patna for providing invaluable comments on the language of the paper.

References

1. Abbas A, Abdelsamea MM, Gaber MM (2020) Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. arXiv:2003.13815
2. Abdel-Basset M, Chang V, Hawash H, Chakraborty RK, Ryan M (2021) Fss-2019-ncov: A deep learning architecture for semi-supervised few-shot segmentation of covid-19 infection. *Knowl.-Based Syst* 212:106647
3. Al Mamun KA, Alhussein M, Sailunaz K, Islam MS (2017) Cloud based framework for parkinson's disease diagnosis and monitoring system for remote healthcare applications. *Futur Gener Comput Syst* 66:36–47
4. Al-Qurishi M, Al-Rakhami M, Al-Qershi F, Hassan MM, Alamri A, Khan HU, Xiang Y (2015) A framework for cloud-based healthcare services to monitor noncommunicable diseases patient. *Int J Distrib Sens Netw* 11(3):985629
5. Ali A, Zhu Y, Zakarya M (2021) A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. *Multimed Tools Appl* :1–33
6. Alsharif M, Alsharif Y, Yahya K, Alomari O, Albreem M, Jahid A (2020) Deep learning applications to combat the dissemination of covid-19 disease: a review. *Eur Rev Med Pharmacol Sci* 24(21):11455–11460
7. ALzubi JA, Bharathikannan B, Tanwar S, Manikandan R, Khanna A, Thaventhiran C (2019) Boosted neural network ensemble classification for lung cancer disease diagnosis. *Appl Soft Comput* 80:579–591
8. Ardakani AA, Acharya UR, Habibollahi S, Mohammadi A (2020) Covidiag: a clinical cad system to diagnose covid-19 pneumonia based on ct findings. *Eur Radiol* :1–10
9. Battle shifting as Covid-19 threat stalks rural India ((Last accessed date September 25, 2020)). <https://www.hindustantimes.com/india-news/battle-shifting-as-covid-19-threat-stalks-rural-india/story-GZJsSPOOpNR0RF9Yj5B59M.html>
10. Brunese L, Mercaldo F, Reginelli A, Santone A (2020) Explainable deep learning for pulmonary disease and coronavirus covid-19 detection from x-rays. *Comput Methods Programs Biomed* 196:105608
11. Che Azemin MZ, Hassan R, Mohd Tamrin MI, Md Ali MA (2020) Covid-19 deep learning prediction model using publicly available radiologist-adjudicated chest x-ray images as training data: preliminary findings. *Int J Biomed Imaging* :2020
12. Chen X, Tang Y, Mo Y, Li S, Lin D, Yang Z, Yang Z, Sun H, Qiu J, Liao Y et al (2020) A diagnostic model for coronavirus disease 2019 (covid-19) based on radiological semantic and clinical features: a multi-center study. *Eur Radiol* 30(9):4893–4902

13. Chicco D, Jurman G (2020) The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genom* 21(1):1–13
14. Coronavirus disease (COVID-19) pandemic ((Last accessed date September 25, 2020)). <https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19>
15. Coronavirus disease (COVID-19) Weekly Epidemiological Update and Weekly Operational Update ((Last accessed date September 25, 2020)). <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports>
16. Das AK, Kalam S, Kumar C, Sinha D (2021) Tlcov-an automated covid-19 screening model using transfer learning from chest x-ray images. *Chaos Solitons Fractals* :110713
17. Diagnostic testing for SARS-CoV-2: interim guidance ((Last accessed date September 25, 2020)). <https://apps.who.int/iris/handle/10665/334254>
18. Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput* 10(7):1895–1923
19. Everitt BS (1992) The analysis of contingency tables. CRC Press, Boca Raton
20. Gaura EI, Brusey J, Allen M, Wilkins R, Goldsmith D, Rednic R (2013) Edge mining the internet of things. *IEEE Sens J* 13(10):3816–3825
21. Ghosal D, Kolekar MH (2018) Music genre recognition using deep neural networks and transfer learning. In: *Interspeech*. pp 2087–2091
22. Gianchandani N, Jaiswal A, Singh D, Kumar V, Kaur M (2020) Rapid covid-19 diagnosis using ensemble deep transfer learning models from chest radiographic images. *J Ambient Intell Human Comput* :1–13
23. Goldstein E, Keidar D, Yaron D, Shachar Y, Blass A, Charbinsky L, Aharoni I, Lifshitz L, Lumelsky D, Neeman Z et al (2020) Covid-19 classification of x-ray images using deep neural networks. [arXiv:2010.01362](https://arxiv.org/abs/2010.01362)
24. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 770–778
25. He X, Yang X, Zhang S, Zhao J, Zhang Y, Xing E, Xie P (2020) Sample-efficient deep learning for covid-19 diagnosis based on ct scans. *medRxiv*
26. Hemdan EED, Shouman MA, Karar ME (2020) Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. [arXiv:2003.11055](https://arxiv.org/abs/2003.11055)
27. Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)
28. Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X et al (2020) Clinical features of patients infected with 2019 novel coronavirus in wuhan, china. *The Lancet* 395(10223):497–506
29. Huang G, Liu X, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 4700–4708
30. Islam MR, Zibran MF (2018) Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text. *J Syst Softw* 145:125–146
31. Islam SR, Kwak D, Kabir MH, Hossain M, Kwak KS (2015) The internet of things for health care: a comprehensive survey. *IEEE Access* 3:678–708
32. Ismael AM, Sengür A (2021) Deep learning approaches for covid-19 detection based on chest x-ray images. *Expert Syst Appl* 114054:164
33. 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
34. Jiang Y, Chen H, Loew M, Ko H (2020) Covid-19 ct image synthesis with a conditional generative adversarial network. *IEEE J Biomed Health Inf*
35. Karakanis S, Leontidis G (2021) Lightweight deep learning models for detecting covid-19 from chest x-ray images. *Comput Biol Med* 104181:130
36. Kaur T, Gandhi TK (2020) Deep convolutional neural networks with transfer learning for automated brain image classification. *Mach Vis Appl* 31(3):1–16
37. Kong X, Wang K, Wang S, Wang X, Jiang X, Guo Y, Shen G, Chen X, Ni Q (2021) Real-time mask identification for covid-19: an edge computing-based deep learning framework. *IEEE Internet Things J*
38. Li X, Radulovic M, Kanjer K, Plataniotis KN (2019) Discriminative pattern mining for breast cancer histopathology image classification via fully convolutional autoencoder. *IEEE Access* 7:36433–36445
39. Liang G, Zheng L (2020) A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Comput Methods Programs Biomed* 187:104964
40. Loey M, Manogaran G, Khalifa NEM (2020) A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images
41. Minaee S, Kafieh R, Sonka M, Yazdani S, Soufi GJ (2020) Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. *Med Image Anal* 65:101794

42. 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
43. Muhammad G, Alhamid MF, Alsulaiman M, Gupta B (2018) Edge computing with cloud for voice disorder assessment and treatment. *IEEE Commun Mag* 56(4):60–65
44. Pathak Y, Shukla PK, Tiwari A, Stalin S, Singh S, Shukla PK (2020) Deep transfer learning based classification model for covid-19 disease. *IRBM*
45. Q&A on coronavirus disease (COVID-19) ((Last accessed date January 28, 2021)). <https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19>
46. Rahman MA, Hossain MS (2021) An internet of medical things-enabled edge computing framework for tackling covid-19. *IEEE Internet Things J*
47. Rahman MA, Hossain MS, Alrajeh NA, Guizani N (2020) B5g and explainable deep learning assisted healthcare vertical at the edge: Covid-19 perspective. *IEEE Netw* 34(4):98–105
48. Rahman T, Khandakar A, Qiblawey Y, Tahir A, Kiranyaz S, Kashem SBA, Islam MT, Al Maadeed S, Zughaier SM, Khan MS et al (2021) Exploring the effect of image enhancement techniques on covid-19 detection using chest x-rays images. *Comput Biol Med* :104319
49. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis (IJCV)* 115(3):211–252. 10.1007/s11263-015-0816-y
50. Sacco A, Esposito F, Marchetto G, Kolar G, Schweteye K (2020) On edge computing for remote pathology consultations and computations. *IEEE J Biomed Health Inf* 24(9):2523–2534
51. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 4510–4520
52. SARS-COV-2 Ct-Scan Dataset ((Last accessed date September 12, 2020)). <https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset>
53. Shalbaf A, Vafaezadeh M et al (2021) Automated detection of covid-19 using ensemble of transfer learning with deep convolutional neural network based on ct scans. *Int J Comput Assist Radiol Surgery* 16(1):115–123
54. Sharma S (2020) Drawing insights from covid-19-infected patients using ct scan images and machine learning techniques: a study on 200 patients. *Environ Sci Pollut Res* 27(29):37155–37163
55. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: Vision and challenges. *IEEE Interner Things J* 3(5):637–646
56. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*
57. Song Y, Zheng S, Li L, Zhang X, Zhang X, Huang Z, Chen J, Zhao H, Jie Y, Wang R et al (2020) Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images. *medRxiv*
58. Vardhana M, Arunkumar N, Lasrado S, Abdulhay E, Ramirez-Gonzalez G (2018) Convolutional neural network for bio-medical image segmentation with hardware acceleration. *Cogn Syst Res* 50:10–14
59. Wang S, Zhao Y, Xu J, Yuan J, Hsu CH (2019) Edge server placement in mobile edge computing. *J Parallel Distrib Comput* 127:160–168
60. Yang G, Jiang M, Ouyang W, Ji G, Xie H, Rahmani AM, Liljeberg P, Tenhunen H (2017) Iot-based remote pain monitoring system: From device to cloud platform. *IEEE J Biomed Health Inf* 22(6):1711–1719
61. Zhang J, Karimireddy SP, Veit A, Kim S, Reddi SJ, Kumar S, Sra S (2019) Why are adaptive methods good for attention models?. *arXiv:1912.03194*
62. Zhang J, Xie Y, Pang G, Liao Z, Verjans J, Li W, Sun Z, He J, Li Y, Shen C et al (2020) Viral pneumonia screening on chest x-rays using confidence-aware anomaly detection. *IEEE Trans Med Imaging*
63. Zheng H, Fu J, Mei T, Luo J (2017) Learning multi-attention convolutional neural network for fine-grained image recognition. In: *Proceedings of the IEEE international conference on computer vision*. pp 5209–5217