



Cloud-based COVID-19 disease prediction system from X-Ray images using convolutional neural network on smartphone

Madhusudan G. Lanjewar¹ · Arman Yusuf Shaikh¹ · Jivan Parab¹

Received: 5 May 2022 / Revised: 1 November 2022 / Accepted: 4 November 2022 /

Published online: 24 November 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

COVID-19 has engulfed over 200 nations through human-to-human transmission, either directly or indirectly. Reverse Transcription-polymerase Chain Reaction (RT-PCR) has been endorsed as a standard COVID-19 diagnostic procedure but has caveats such as low sensitivity, the need for a skilled workforce, and is time-consuming. Coronaviruses show significant manifestation in Chest X-Ray (CX-Ray) images and, thus, can be a viable option for an alternate COVID-19 diagnostic strategy. An automatic COVID-19 detection system can be developed to detect the disease, thus reducing strain on the healthcare system. This paper discusses a real-time Convolutional Neural Network (CNN) based system for COVID-19 illness prediction from CX-Ray images on the cloud. The implemented CNN model displays exemplary results, with training accuracy being 99.94% and validation accuracy reaching 98.81%. The confusion matrix was utilized to assess the models' outcome and achieved 99% precision, 98% recall, 99% F1 score, 100% training area under the curve (AUC) and 98.3% validation AUC. The same CX-Ray dataset was also employed to predict the COVID-19 disease with deep Convolution Neural Networks (DCNN), such as ResNet50, VGG19, InceptionV3, and Xception. The prediction outcome demonstrated that the present CNN was more capable than the DCNN models. The efficient CNN model was deployed to the Platform as a Service (PaaS) cloud.

Keywords COVID-19 · CNN · Cloud computing · Chest X-Ray · DCNN

✉ Jivan Parab
jsparab@unigoa.ac.in

Madhusudan G. Lanjewar
madhusudan@unigoa.ac.in

Arman Yusuf Shaikh
shaikharmannx@gmail.com

¹ School of Physical and Applied Sciences, Goa University, Taleigao Plateau, Goa 403206, India

1 Introduction

Coronaviruses are a big family of viruses identified in the mid-19th century [45]. They are commonly associated with upper respiratory tract illnesses and affect many animal species. Several known spill-over events have seen animal viruses transmitted among humans. Since its outbreak in Wuhan-China, Severe Acute Respiratory Syndrome (SARS)-CoV-2 has spread worldwide and infected masses [74]. The infection spreads from the COVID-19-affected individual to the healthy by direct or indirect contact [41]. It shows no-specific symptoms ranging from multi-organ malfunction in an asymptomatic state to acute respiratory syndrome, sometimes leading to death. Commonly seen traits of COVID-19 comprise fever, headache, sore throat, cough, breathlessness, fatigue, myalgia, and pneumonia. As a result, distinguishing COVID-19 from other respiratory infections is difficult [65]. SARS-CoV-2 diagnostic methodologies have been developed using serological, nanotechnology, and molecular approaches [14]. Among these methods, RT-PCR, a test based on a molecular approach, is widely accepted as a standard diagnostics method worldwide [76]. However, it is a time-consuming, complicated operation requiring a specialised workforce [11].

CX-Ray scans have been used as a part of routine patient care to diagnose diseases and injuries involving internal organs. X-Radiation is a form of high-energy electromagnetic radiation that can penetrate the body [57]. Different tissues absorb this radiation differently, forming the foundation for X-Ray Imaging (XRI). XRI is a well-established technology in medical systems. According to research, viruses of the Coronavirus family display substantial indications in radiographic pictures [40]. The CX-Ray of the patient's thoracic cavity can help distinguish an individual affected with COVID-19 from the rest of the population. It also gives medical practitioners an insight into the extent of damage caused by the virus. CX-Ray technology offers numerous advantages over other testing procedures. CX-Ray facilities are widespread, consume less time, are affordable, and are non-invasive. These features highlight the applicability of CX-Ray for detecting COVID-19 in this healthcare crisis. Quick and precise identification of COVID-19 is crucial for controlling the virus's rapid spread and reducing the burden on healthcare systems.

A new round of development in pertinent technologies like deep learning (DL) and image recognition has resulted in a breakthrough in Artificial Intelligence (AI) [37]. AI has found a foot-hole in the medical industry for assisting professionals in diagnosis and treatment processes. Machine learning (ML) can aid medical diagnosis by identifying and detecting various ailments. ML can analyze a substantial amount of data in varied forms and produce a predictive conclusion. This versatility makes ML applicable in all manners of trials. The use of DL models like CNN in processing a large amount of data to produce complex associations within its multiple dimensions has exemplified its ability [57]. The advent of revved-up internet infrastructure has served as a boost for cloud computing. In layman's terms, it is the on-demand delivery of computing resources over the internet [6]. These include services for data storage, computation, networking solution, and other related utilities. Platform as a Service (PaaS) enables users to deploy software and coding models in real time.

This paper explores the possibility of actualizing a cloud-based real-time COVID-19 detection system with the aid of CNN. The CNN model was trained using the CX-Ray images dataset [8, 54]. Various investigations on the recognition of COVID-19 were reported in the literature using ML and DL methods, but especially CNN and DCNN methods were suitable for COVID-19 detection. At most conferences and reputed journals, CNN research is currently

a dominant subject, including the recognition of COVID-19 disease from the CXR or computed tomography (CT) scans. CNN is much less reliant on feature extraction and autonomously pulls features layer-wise to characterize input data with a hierarchical structure of elements. Due to this, the use of CNN in medical diagnostics has recently received a lot of interest. Several investigations have revealed that using DL clinical diagnosis in illnesses involving tissue, blood vessels, and joints with excellent success. Studies have been conducted on solutions based on medical imaging technologies such as magnetic resonance imaging (MRI), CT scans, etc. CX-Ray and CT scans are further investigated as alternate diagnosis techniques for COVID-19. Since the epidemic, research has focused on ways to check the COVID-19 virus. COVID-19 was detected in CX-Ray using DL algorithms in medical image processing. The CX-Ray assists clinicians in making critical decisions. The primary advantages of CX-Ray over CT scans include quick screening, small size, and simple setup, but it is difficult to identify persons with mild symptoms using CX-Ray. As performance aspects can be delicate, there is higher variation between intra- and inter-observer data assessed by professionals. As a result, there is a growing demand for computerized screening testing to assist radiologists in creating good COVID-19 assessment decisions [53].

The study attempts to generate a new understanding of this challenge by thoroughly exploring it with established models. The different sorts of research aid in better organizing research and implementing the most suitable methods to identify COVID-19. The authors tried the specialized applied research technique in this study. This study aims to improve existing procedures by changing CNN frameworks to detect COVID-19 from CX-Ray pictures. The authors employed a simple random sample strategy on the healthy and COVID-19-infected X-Ray datasets. This exploratory research must be addressed since a rapid illness forecasting system is required. CNN have shown positive performance and generalization capabilities, which could contribute to its use in various applications, including COVID-19 illness forecasting from photographs. Despite its remarkable performance, CNN presents unique challenges, such as a trade-off between efficacy and operation time [21].

On the other hand, CNN requires a considerable memory space and has a high CPU expense in categorization and testing, rendering its deployment in most scenarios impracticable on IoT devices, smartphones, and single-board computers (SOC). The inexpensive CNN-based solution is introduced to overcome the issues mentioned earlier. Additionally, due to the trial-and-error aspect of CNN, establishing a well-performing CNN requires substantial cognitive programming appropriate for a resource-constrained device [13, 26] and uploading such CNN to the cloud.

The authors examine different network structure hyperparameter configurations of CNN to ensure the highest trade-off between computational cost and predicting performance. The authors investigate input image size variations, convolutional layers with filter size, optimizers, dropout layers, batch normalization layers, etc., each with its own hyperparameters. Filters or kernels created feature maps when a CX-Ray image was applied in CNN. The feature maps draw attention to various visual components in the original CX-Ray image. As a result, the size of the filter choice may allow cost-efficient COVID-19 illness forecasting systems to significantly lower the computing price of CNN-based systems. This study also focuses on the learning rate context and reduction of vectors of input data. The CNN kernel size remains lower for optimum feature extraction with average pooling levels. The authors produced impressive results by evaluating the CNN on CX-Ray images to diagnose the disease compared to the literature reported worm in terms of accuracy, the number of layers, and

model size. Aside from that, the designed model is simple and may be accessed by mobile phones or laptops.

The primary motivation here is to design a software solution that can support doctors using android or any smartphone to identify COVID-19 disease to help physicians or doctors detect COVID-19 disease. The disease-classified history documents can be utilized to learn new connections and concepts in the future. This approach is helpful for COVID-19 diagnosis, predicting behaviour, estimating diseases, and using medicines. This research might be explored in other domains as well. It is beneficial to discover illness quickly if the health forecasting system is available on a mobile phone or other embedded device. The X-ray machine can snap pictures of the individual, which can then be submitted for the forecast. The following is the primary contribution of this work:

- CNN and DCNN models were implemented in Python.
- Selection of hyperparameters and optimizers (RMSprop and Adam) for a CNN and DCNN ideal for identifying COVID-19.
- Four state-of-the-art pre-trained DCNN (ResNet50, VGG19, InceptionV3, and Xception) models were implemented to compare the results of CNN.
- The CNN model obtained a 99.94 per cent training and a 98.81 per cent validation accuracy.
- The models' results were evaluated using the confusion matrix and Receiver Operator Characteristic (ROC) strategies, which resulted in 99 per cent precision, 98 per cent recall, 99 per cent F1 score, 100 per cent training AUC, and 98.3 per cent validation AUC.
- Compared to DCNN models, the proposed CNN model was compact and had fewer parameters.
- The comparative outcomes highlight the efficiency of the CNN was better compared to literature-reported models.
- The efficient model having a comparatively small memory size, was deployed to the PaaS cloud. More importantly, this model is accessible through a mobile phone.

The paper is structured as follows: Section 2 discusses the related works, while Section 3 discusses dataset information. The COVID-19 diagnostic techniques with CNN and Cloud computing discusses in Section 4. Section 5 presents the experiment and analyses the outcomes, while Section 6 concludes the paper and discusses future work.

2 Related works

In this section, the authors discuss the recent work reported by the researcher. Recently, ML and DL-based algorithms have been proposed by numerous investigators around the globe for COVID-19 detection. Mansour et al. [41] introduced a COVID-19 determination approach based on Feature Correlated Naïve Bayes (FCNB) and achieved a maximum of 99% detection accuracy. The FCNBP was utilized to categorize patients using the weighted Naive Bayes method with multiple variations as the feature correlation, and it was also compared to other methodologies. Das et al. [11] proposed a DCNN (DenseNet201, Resnet50V2, and Inceptionv3) model for differentiating COVID-19 from CX-Rays. The models then joined to classify a class value using a novel weighted average ensembling approach and achieved 91.62 per cent accuracy, outperforming the state-of-the-art DCNN models. Reshi et al. [57] presented

a DCNN model based on CX-Ray categorization for the diagnosis of COVID-19. The preprocessing phases of the datasets include dataset balance, image analysis by medical professionals, and data augmentation, achieving an overall accuracy of 99.5 per cent. The CNN model was evaluated in two different schemes. In the first one, the model was tested using 100 X-ray pictures from the training dataset and achieved 100% accuracy. The model was evaluated using a separate dataset of COVID-19 CX-Ray scans in the second scheme and reached 99.5 per cent accuracy. The DCNN model has 68 layers, while our proposed model has 18 layers.

Das et al. [12] suggested an autonomous Covid-19 testing procedure that uses CX-Rays to categorize individuals with this condition into 3 types: Covid-19 positive, other pneumonia illness, and no sickness using CNN, VGG-16, and ResNet-50. The model's performance with the three learning methods has been assessed, and VGG-16 outperformed CNN and ResNet-50. The TLCoV model's accuracy was 97.67%, the precision was 96.65%, the recall was 96.54%, and the F1 score was 96.59%. The total parameters used in this TLCoV is about 1,24,10,023, which is higher than our proposed model of 6,447,138 parameters and is more accurate. Ayalew et al. [5] offer a recognition and prevention strategy (DCCNet) for rapid COVID-19 diagnostics utilizing patient CX-Ray. CNN and histogram of oriented gradients (HOG) approach were suggested for timely detection using CX-Ray images acquired from the University of Gondar and online databases. The DCCNet model achieved 99.9 per cent training and 98.3 per cent test accuracy, while HOG achieved 100 per cent training and 98.5 per cent test accuracy. The hybrid model gained 99.97% and 99.67% training and testing accuracy, respectively. Indumathi et al. [24] presented strategies to categorize COVID-19-impacted locations into hazardous, moderate, and safe zones. Furthermore, they offer a DL-based technique for predicting COVID-19-affected zones using the Virudhunagar district's covid-19 dataset. Compared to the C5.0 algorithm's accuracy of 95.92 per cent and the error rate of 4.08 per cent, the DL efficiently forecasts the zones with a 98.06 per cent accuracy and an error rate of 1.94 per cent.

As a preliminary screening tool, Salau et al. [60] suggested a Support Vector Machine (SVM)-based approach for recognizing and categorizing COVID-19. The features were extracted using the discrete wavelet transform (DWT) algorithm, and the extracted features were identified using SVM and achieved a detection rate of 98.2 per cent. Yadessa et al. [72] presented a microcontroller (Arduino) and ultrasonic-based sensing devices to construct a touch-free hand-washing system for COVID-19 illness prevention. Natnael et al. [47] did cross-sectional research to evaluate the percentage of taxi drivers who utilize facemasks for the suggested precautionary efforts to manage viral transmission. Narin et al. [46] utilized DCNN models: InceptionV3, ResNet50, and InceptionResNetV2 to identify COVID-19 sickness from CX-Ray images. They constructed three distinct binary classifications with four classes (COVID-19, healthy, viral pneumonia, and bacterial pneumonia) and applied 5-fold cross-validation. Evaluation and performance findings, the ResNet50 model, delivers the best classification accuracy of 96.1 per cent, 99.5 per cent, and 99.7 per cent for Dataset-1, Dataset-2, and Dataset-3, respectively. The ResNet50 has 50 layers, and the model size was also large compared to our study. To forecast COVID-19 disease using CX-Ray, Abbas et al. [1] suggested the Decompose, Transfer, and Compose (DeTraC) model based on DCNN and acquired an accuracy of 95.12%. The accuracy performance of this work reaches up to 95.12%, but in our case, it reaches up to 98.81%. Khan et al. [27] presented a DCNN model (CoroNet) based on the Xception architecture for COVID-19 identification from CX-Ray.

CoroNet got a total accuracy of 89.6 per cent, with precision and recall rates for COVID-19 cases of 93 per cent and 98.2 per cent for 4-class instances, respectively. The proposed approach generated a classification accuracy of 95% for 3-class classification. The accuracy is less compared to our work. Furthermore, Maghddid et al. [39] suggested the DL method (AlexNet) to analyze COVID-19 cases effectively and quickly from CX-Ray and CT scans and achieved 94.1% and 98% accuracy on the AlexNet network and modified CNN, respectively. Sethy et al. [62] utilized ResNet50 to extract features from CX-Ray and the SVM classifier to classify them and achieved accuracy, FPR, F1 score, MCC, and Kappa of 95.38 per cent, 95.52 per cent, 91.41 per cent, and 90.76 per cent, respectively. Rehman et al. [56] examined the performance of AlexNet, VGG, SqueezeNet, GoogLeNet, MobileNet, ResNet with its variants, and DenseNet in detecting COVID-19. The highest level of accuracy was 98.75 per cent. Kumar et al. [29] demonstrated one study using CX-Ray pictures to identify COVID-19. For COVID-19 identification, various TL models such as EfficientNet, Xception, GoogLeNet and VGG16 plus DenseNet were used. ResNet152V2 performed the best of the four current models, with 98.15 per cent accuracy, and VGG16 plus DenseNet achieved 99.32 per cent accuracy. The proposed Ensemble method achieved 99.28 per cent accuracy in two-class categorization. The models performed with perfect accuracy, but the number of layers and size of the models were enormous compared to our study. A three-dimensional DL framework for the diagnosis of COVID-19 was discussed by Dadário et al. [10]. A total of 4356 chest CT scan images were utilized to check the performance of the suggested strategy. The results indicate excellent sensitivity and specificity. Ozturk et al. [50] used a DarkNet model as a classifier for you only look once (YOLO) real-time object identification system for autonomous COVID-19 detection employing CX-Ray images. The suggested approach was designed to offer reliable diagnostics for binary (No-Findings and COVID) and multi-class classification (No-Findings, COVID, and Pneumonia) and achieved accuracy of 98.08 per cent and 87.02 per cent for binary and multi-class classification, respectively. They created a model with 17 convolutional layers and employed numerous filtering for each one.

Yan et al. [73] built a multi-tasking AI framework for diagnosing COVID-19 in patient lungs using CT scans, and it obtained an accuracy of 89%. A system for identifying COVID-19 by classifying chest CT scans was proposed by Singh et al. [64]. CNN parameters were determined with the help of differential evolution and achieved an accuracy of 98.24%. Due to the promising results using CX-Ray and the affordability and availability of needed infrastructure, Ng et al. [48] and Huang et al. [22], in their independent findings, concluded that CX-Ray images were better than all other means of recognizing COVID-19. Zhang et al. [75] utilized a one-class classification-based anomaly detection (CAAD) framework to identify non-viral and viral pneumonia pictures. These CADD models include extracting features, finding anomalies, and predicting modules. The decision whether a patient has viral pneumonia or not was made if the anomaly result was high or the prediction score was lower. This method achieved an AUC of 83.61% and a sensitivity of 71.70%. Apostolopoulos et al. [4] created the TL-based method for classifying Covid19 from CX-Ray images. Two separate datasets were used. The first contains 1427 CX-Ray images, 224 of which were infected with COVID-19. The second dataset has 1442 CXR images, including 224 positive COVID-19 CXR images. The results indicate that DL combined with CX-Ray imaging may identify with the accuracy, sensitivity, and specificity of 96.78%, 98.66%, and 96.46%, respectively. Farooq et al. [15] presented a CNN framework (COVIDResNet) to forecast pneumonia and COVID-19. They create quick and accurate residual neural networks using cutting-edge training

approaches like cyclical learning rate discovery, progressive resizing, and discriminative learning rates. This paper describes a three-step method for fine-tuning a ResNet50 framework to increase model effectiveness and decrease training time. This is accomplished by resizing input photos progressively to $229 \times 229 \times 3$, $224 \times 224 \times 3$, and $128 \times 128 \times 3$ pixels and fine-tuning the network at each level. This method achieved an accuracy of 96.23 per cent in all classes. Kumar et al. [30] employed DL to train an intrusion detection system on the fog layer to distinguish between attack and benign network traffic. M. V. MK et al. [44] presented a complete investigation of COVID-19 detection using DL techniques and cost-effectiveness evaluation. They also compare conventional methods to improve the identification process utilizing DL techniques. Meraihi et al. [43] presented a comprehensive assessment of ML-based studies for COVID-19 identification, diagnosis, and forecast. In this survey, DL was employed in 79 per cent of the cases and supervised learning (Random Forest (RF), SVM, and Regression methods) in only 16%. Alkhodari et al. [3] proposed DL models that depend on hand-crafted features derived from original recordings and Mel-frequency cepstral coefficients (MFCC), as well as deep-activated features knowledge gained by a conjunction of CNN and bi-directional long short-term memory units (CNN-BiLSTM). The suggested DL approach used shallow and deep recordings with an average classification accuracy of 94.58 per cent and 92.08 per cent, respectively. Additionally, it was performed to detect COVID-19 individuals with a high sensitivity of 94.21 per cent, specificity of 94.96 per cent, and AUROC curves of 0.90. Guefrechi et al. [19] reported a DL system that detects COVID-19 from CX-Ray pictures utilizing three DCNN models: ResNet50, InceptionV3, and VGG16. They also used data augmentation approaches to produce an artificially huge number of CX-Ray images. This framework achieves an accuracy of 97.20 per cent, 98.10 per cent, and 98.30 per cent for Resnet50, InceptionV3, and VGG16, respectively. Table 1 depicts the details comparability of the presented work with the literature.

From Table 1, it can be seen that most of the researchers used the DL method to detect COVID-19 from CX-Ray images. Among these studies, few were complicated [5, 41], less accurate [1, 3, 4, 11, 12, 15, 19, 24, 27, 39, 43, 50, 56, 57, 60, 62, 64, 73, 75] than our proposed work, and achieved more accuracy [29, 46] than proposed work, but the number of layers and size of the model is significant. The main highlight of the proposed CNN-based method is that the CNN models have fewer parameters and layers than the literature-reported work. Apart from this, the performance of the CNN model is also good.

3 Dataset description

The CX-Ray dataset of COVID-19-positive and NORMAL cases was built by an investigator from the University of Dhaka and Qatar University in collaboration with Pakistan and Malaysia medical doctors [9]. The dataset comprises 10,192 CX-Ray pictures of the NORMAL patient and 3616 COVID-19 patients. Figure 1 shows the CX-Ray picture of COVID-19 and NORMAL patients.

4 Methodology

Figure 2 shows the workflow of the proposed CNN-based COVID-19 prediction system. The workflow consists of image dataset collection, dataset split and image resizing, CNN

Table 1 Comparison of the presented work with literature

Author	Models	Accuracy (%-per cent)
Mansour et al. [41]	FCNB	99%
Das et al. [11]	DenseNet201, Resnet50V2, and Inceptionv3	91.62%
Reshi et al. [57]	CNN	99.5%
Das et al. [12]	CNN, VGG-16, and ResNet-50	97.67%
Ayalew et al. [5]	DCCNet	99.67%
Indumathi et al. [24]	DL	95.92%
Salau et al. [60]	DWT and SVM	98.2%
Narin et al. [46]	InceptionV3, ResNet50, and InceptionResNetV2	95.12%
Abbas et al. [1]	DeTraC	95.7%
Khan et al. [27]	CoroNet	98.2%
Maghdid et al. [39]	AlexNet	98%
Sethy et al. [62]	ResNet50 and SVM	95.38%
Rehman et al. [56]	SqueezeNet, AlexNet, MobileNet, GoogLeNet, ResNet with its variants, VGG, and DenseNet	98.75%
Kumar et al. [29]	EfcientNet, XceptionNet, GoogLeNet and VGG16 plus DenseNet	99.32%
Ozturk et al. [50]	DarkNet, YOLO	98.08%
Yan et al. [73]	Multi-task AI	89%
Singh et al. [64]	CNN	98.24%
Zhang et al. [75]	CAAD	83.61% AUC and 71.70% sensitivity
Apostolopoulos et al. [4]	TL	96.78%
Farooq et al. [15]	COVIDResNet	96.23%
Alkhodari et al. [3]	CNN-BiLSTM	94.58% and AUROC curves of 0.90
Guefrechi et al. [19]	ResNet50, InceptionV3, and VGG16	97.20%
Present work	CNN	99.94% training and 98.81% validation accuracy. 100 Training AUC and 98.3% validation AUC

and DCNN modelling, performance assessment, and model deployment to the PaaS cloud. The COVID-19 and non-COVID-19 patients' CX-Ray image datasets were employed in this work. The dataset images were resized to $224 \times 224 \times 3$. CNN and DCNN models were implemented using Python. The confusion matrix technique was utilized to check the performance of all models. Finally, the model was deployed to the Heroku cloud.

4.1 CNN architecture

A CNN can take in an input image and learn crucial aspects from the image (biases and weights) that help to differentiate the images. As shown in Fig. 3, the CNN design consists of various vital parts, including convolution layers, activation functions, pooling layers, and dense or fully connected (FC) layers. The first block is the convolution layer, the most crucial part of the entire architecture, extracting features from input. In the

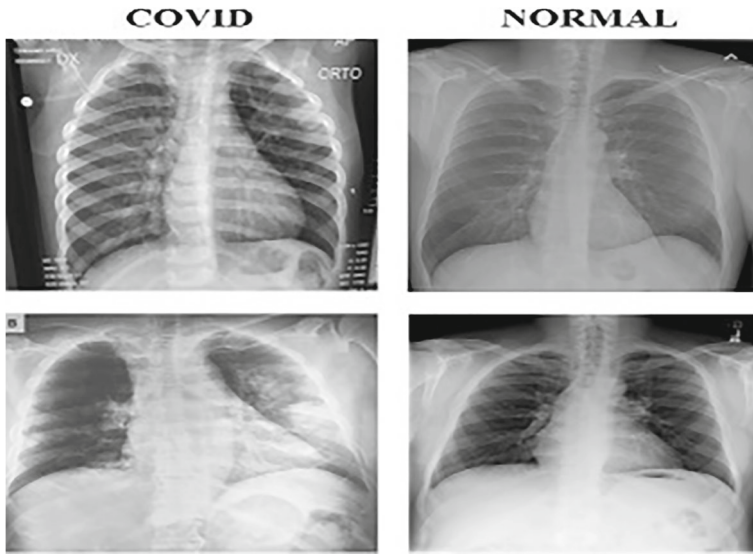


Fig. 1 COVID-19 and NORMAL patients CXR images

linear process, a 2-dimensional small array of numbers convolved across a 2-dimensional input image [31, 33]. A CNN architecture can have multiple layers where the initial layers capture the low-level features, such as colours, edges, gradient orientations, etc. [31], whereas the subsequent layers are responsible for high-level features like large shapes in the image. Hence kernels are also referred to as feature extractors. Zero paddings must be done at the input image edges to coincide with the centre of the kernel on the edge elements of the input image. Another critical parameter is the stride of the convolution operation, which is the distance between two consecutive kernel positions on the input image.

4.1.1 Activation function

In general, the output of the convolution layer is transformed using nonlinear activation functions such as sigmoid, rectified linear unit (ReLU), etc. In this work, ReLU was employed. The output ReLU is the same as the input for positive inputs; otherwise, zero [18, 59]. Model training using the ReLU function achieves better results. The below expression gives the relationship for a ReLU function.

$$h(t)_{CX} = \begin{cases} 0, & t \leq 0 \\ t, & t > 0 \end{cases} \quad (1)$$

4.1.2 Pooling layer

The pooling layer, such as average and max-pooling, incorporates a down-sampling operation and reduces the dimension of the feature map. Max pooling is widespread due to its improved convergence and speed [23].

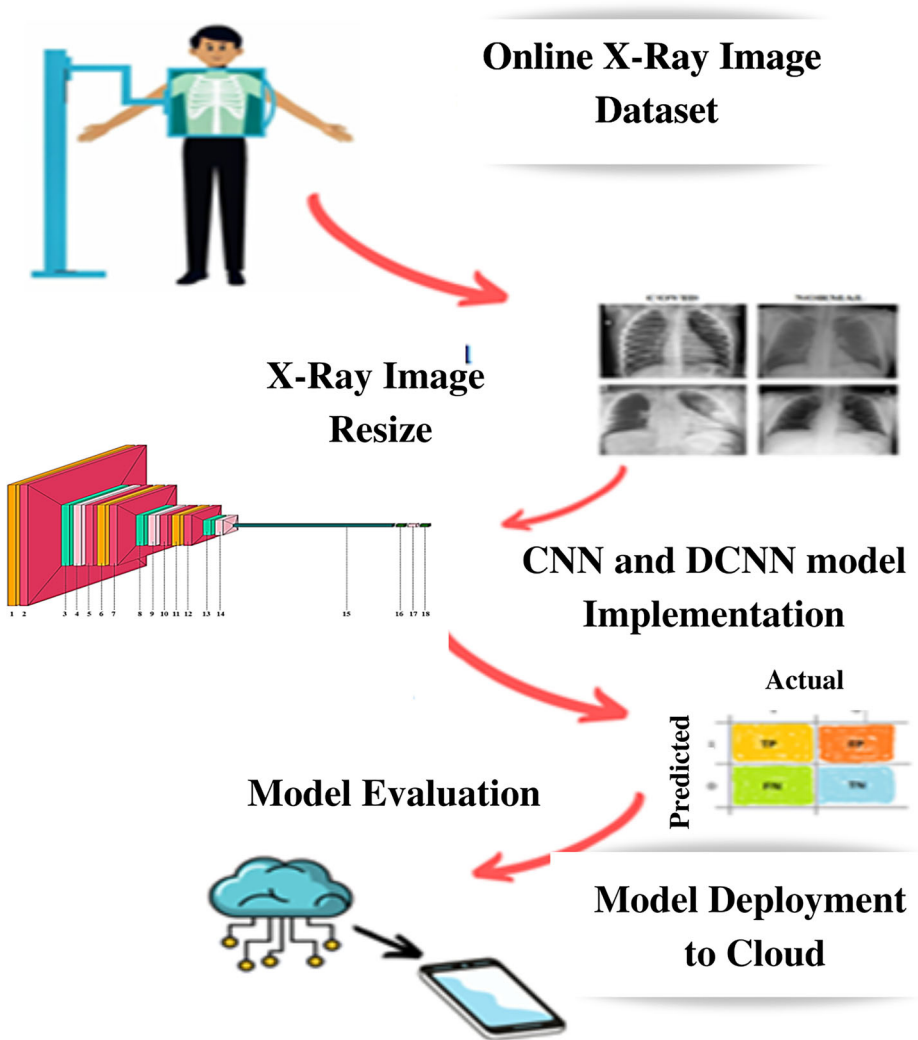


Fig. 2 Proposed methodology

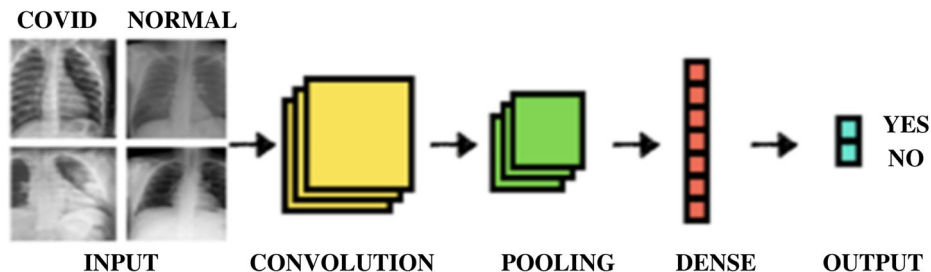


Fig. 3 Typical convolutional neural network architecture

4.1.3 Dropout layer and batch normalization (BZ)

The dropout was employed to avoid over-fitting in a CNN or DCNN by randomly selected neurons and ignored in the training process, hence the term dropout [66]. All the activation of the downstream neurons on the forward pass corresponding to the dropped-out neuron was removed effectively. Also, on the backward pass, the weight updates were not applied. The dropout process helps to increase the accuracy, but not in all cases [36], while BN speeds the training process and makes learning easier. BN is easily implementable by adding a BN layer, which is popular in CNN architectures [25, 68]. This layer also enhances the accuracy of the CNN. But this may come with some penalties for the training time [17].

4.1.4 Fully connected (FC) layer

FC layer input is attached to every one of the outputs hence the term “fully connected” and also called a dense layer. These layers perform the conventional neural network operation and comprise 88–90% of the parameters, and are responsible for taking the output of the last pooling layer. This layer’s output is the number of user-specified classification classes [28].

4.2 CNN implementation and feature extraction

The framework of the CNN is shown in Fig. 4. The input image has a $224 \times 224 \times 3$ input shape. Details of the CNN and dense layers are summarized in Table 2.

The feature extraction process is essential in the DL system. Salau et al. [61] discussed a detailed survey on the feature extraction process. This work uses CNN to extract the informative features from the CX-Ray scans. In this feature extraction process, the input image is propagated forward, halting at a predefined layer, and the outputs of that layer are used as features. CNN has these powerful, discriminative properties of feature extraction and helps forecast classes on which CNN has never been trained. Figure 5 depicts the feature extraction process using CNN. CNN allows an image to pass from the convolutional layer to the average pooling layer via the BN layer, and then the output of the average pooling layer is again applied to the BN layer. The exact process is continued up to fully-connected layers via flattened layer. The input image size was $224 \times 224 \times 3$. The first convolutional layer had a

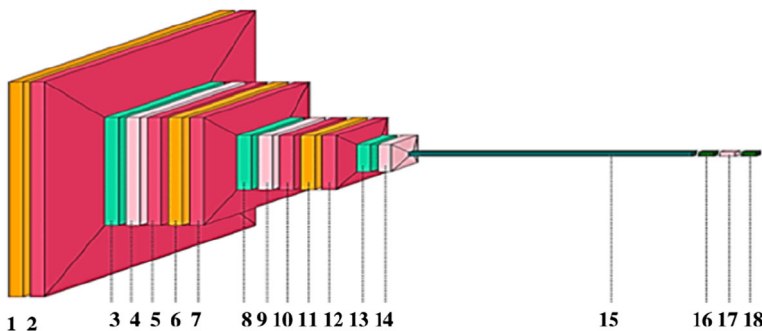


Fig. 4 Proposed CNN model architecture

Table 2 CNN details

Layer Number	Layer (type)	Kernel/Neurons Size	padding	Activation function/ Pool size/stride	Output Shape	Param #
1	conv2d (Conv2D)	16	same	Relu	(None, 224, 224, 16)	448
2	batch_normalization				(None, 224, 224, 16)	64
3	average_pooling2d			pool_size = (2,2) strides=2	(None, 112, 112, 16)	0
4	dropout			0.2	(None, 112, 112, 16)	0
5	batch_normalization_1				(None, 112, 112, 16)	64
6	conv2d_1 (Conv2D)	32	same	Relu	(None, 112, 112, 32)	4640
7	batch_normalization_2				(None, 112, 112, 32)	128
8	average_pooling2d_1			pool_size = (2,2) strides=2	(None, 56, 56, 32)	0
9	dropout_1 (Dropout)			0.2	(None, 56, 56, 32)	0
10	batch_normalization_3				(None, 56, 56, 32)	128
11	conv2d_2 (Conv2D)	64	same	Relu	(None, 56, 56, 64)	18,496
12	batch_normalization_4				(None, 56, 56, 64)	256
13	average_pooling2d_2			pool_size = (2,2) strides=2	(None, 28, 28, 64)	0
14	dropout_2 (Dropout)			0.2	(None, 28, 28, 64)	0
15	flatten (Flatten)				(None, 50,176)	0
16	dense (Dense)	128		Relu	(None, 128)	6,422,656
17	dropout_3 (Dropout)			0.3	(None, 128)	0
18	dense_1 (Dense)	2		Softmax	(None, 2)	258
Total params: 6,447,138						

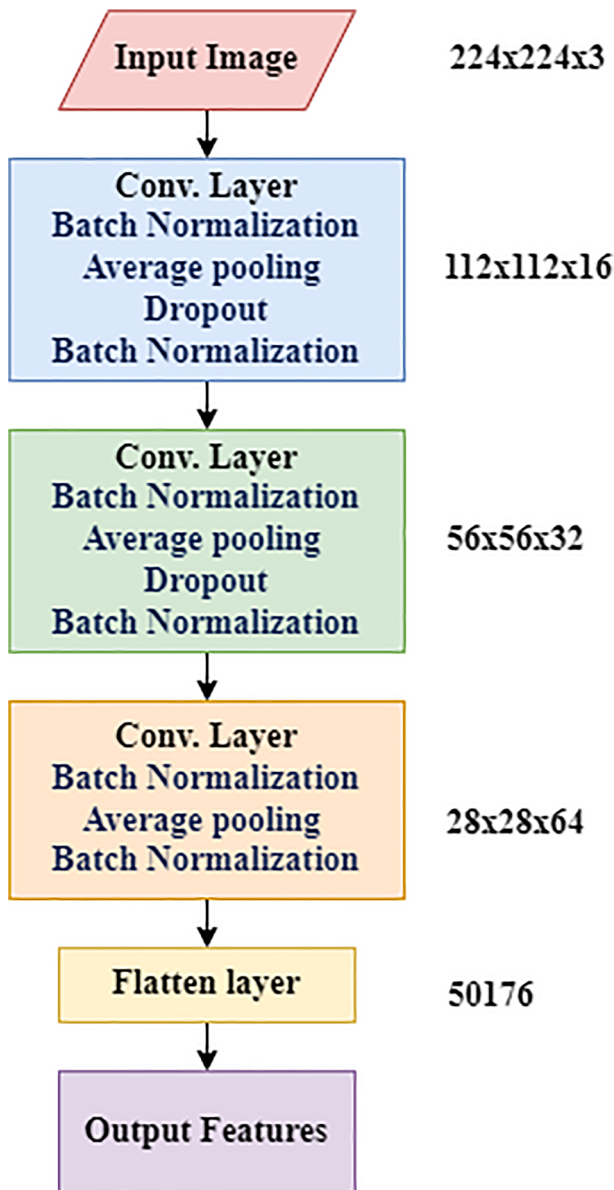


Fig. 5 CNN extracted features

volume shape of $224 \times 224 \times 16$, and the average pooling layer had a volume shape of $112 \times 112 \times 16$. The third average pooling layer's volume was $28 \times 28 \times 64$ flattened to feature vector 50,176 (Flatten Layer output). Given an N-image dataset, this feature extraction will repeat for all dataset photos, yielding a total of $N \times 50,176$ feature vectors. These extracted features were then applied to the dense layers for prediction/classification.

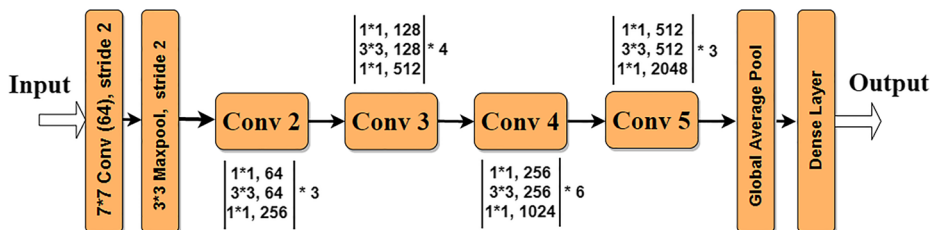


Fig. 6 Basic ResNet50 architecture

4.3 Deep convolutional neural networks (DCNN)

The ResNet50, VGG19, Xception, and DenseNet201 models were chosen for this work because of their remarkable image identification performance. The Resnet50 model has 50, VGG19 has 26, Xception has 126, and Inception V3 has 159 layers. These models have different architectures and also variations in the layers. A brief introduction of these models is as follows:

4.3.1 ResNet50

It is a pre-trained transfer learning (TL) model to simplify the training task. ResNet50 (Fig. 6) or Residual Network 50 contains 50 layers of residual networks and is trained using the ImageNet dataset. The network layers in this design are recast by learning with residual functions concerning the layer inputs. To deal with the vanishing gradient problem, ResNet proposes the skip connection concept, which prevents distortion [20, 70].

4.3.2 VGG19

VGG19 is a DCNN network consisting of convolution layers followed by FC layers. The VGG architecture was primarily introduced for classification and localization problems on images having high resolutions [63]. VGG networks consist of many convolutional layers with increasing depth and small kernels of size 3×3 in each convolutional layer, as shown in Fig. 7. In VGG19, the input layer has a standard input image size of $224 \times 224 \times 3$. The most notable achievement of VGG19 is obtaining a rate of 88% in the ImageNet database [42].

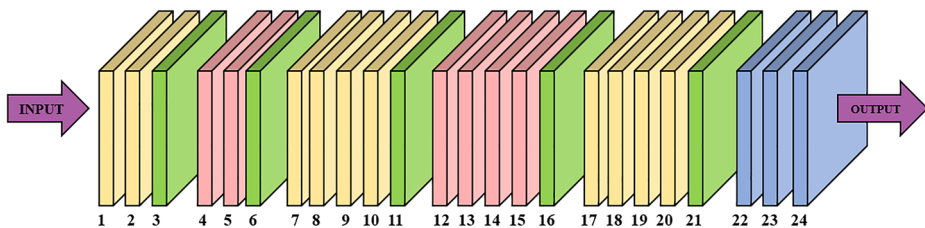


Fig. 7 Graphical representation of VGG19 (1:Conv 1–1, 2:Conv 1–2, 3:Pooling, 4:Conv 2 – 1, 5:Conv 2–2, 6:Pooling, 7:Conv 3 – 1, 8:Conv 3 – 2, 9:Conv 3–3, 10:Conv 3–4, 11:Pooling, 12:Conv 4 – 1, 13:Conv 4 – 2, 14:Conv 4 – 3, 15:Conv 4–4, 16:Pooling, 17:Conv 5 – 1, 18:Conv 5 – 2, 19:Conv 5 – 3, 20:Conv 5 – 4, 21:Pooling, 22: FC6, 23: FC7, 24: FC8)

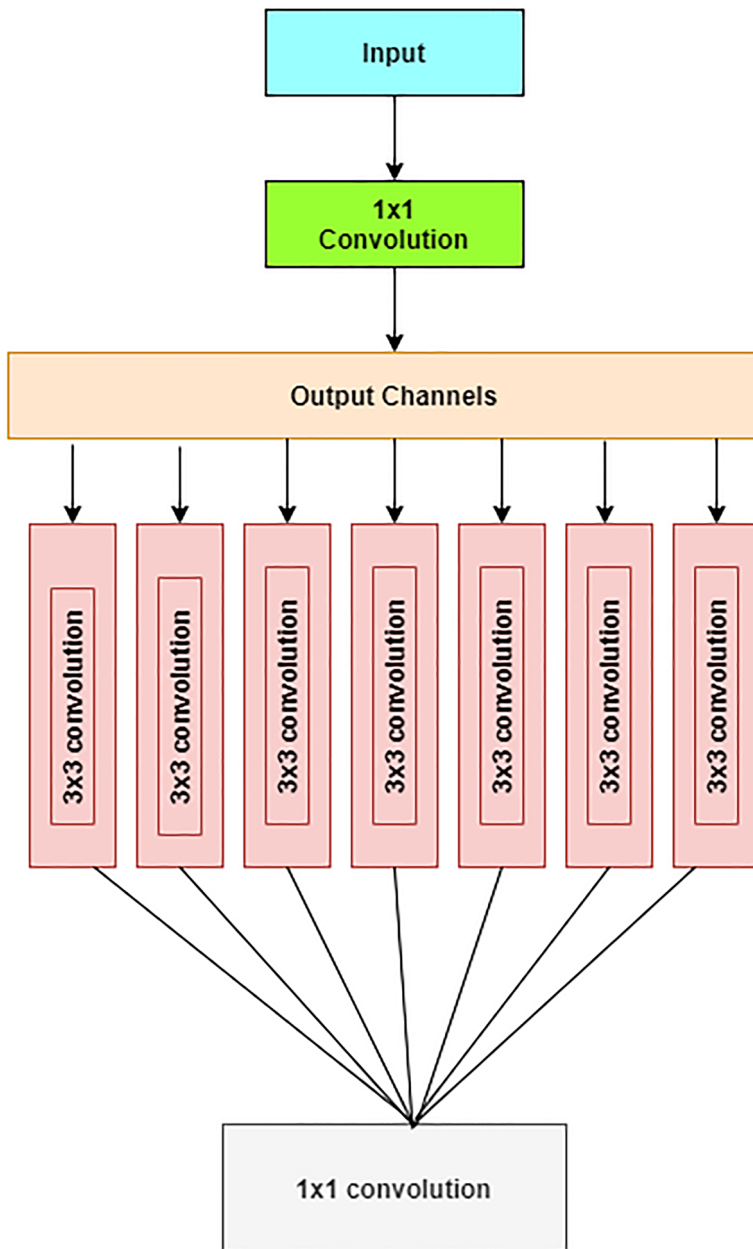


Fig. 8 Basic architecture of the Xception model

4.3.3 Xception

Xception stands for Extreme inception CNN architecture, and depth-wise separable convolution is the main idea behind Xception [7]. It is a linear stack of depth-wise detachable convolution layers containing residual connections. It consists of 36 convolution layers laid

as 14 modules, excluding the first and last modules. In the Xception architecture (Fig. 8), the spatial and depth-wise correlations are evaluated independently. Xception architecture achieves better learning efficiency but cannot reduce the number of training parameters [38, 55].

4.3.4 Inception-V3

Inception-V3 was conceptualized as an extensive network of the popular GoogLeNet [67, 68]. Inception-V3 utilizes an inception model, which joins many different-sized convolutional filters into a new filter. This approach reduces the number of training parameters, reducing the overall computational complexity by employing a bottleneck layer and a 1×1 convolutional layer. In inception-V3 architecture, pooling layers replace the FC layers, which reduces the overall number of parameters. Figure 9 shows the basic diagram of the Inception V3 architecture [49].

4.4 Cloud computing

In today's era, cloud computing has demonstrated excellent development [2]. It has promising application capabilities in various technological and industrial endeavours. Being an entity based on the internet, cloud computing, in simple words, is the on-demand availability of integrated software, hardware, and other computational resources provided to the terminal and portable devices [69]. Cloud computing combines parallel, distributed, and grid computing [2]. The user can access the available high-end computational resources from the cloud without setting up a physical infrastructure. It provides access to high-performance computing on-demand to those who do not possess the capital to acquire it. Cloud service providers offer a variety of services that are built on a shared pool of computing resources. These resources can be scaled up to meet the demand [58].

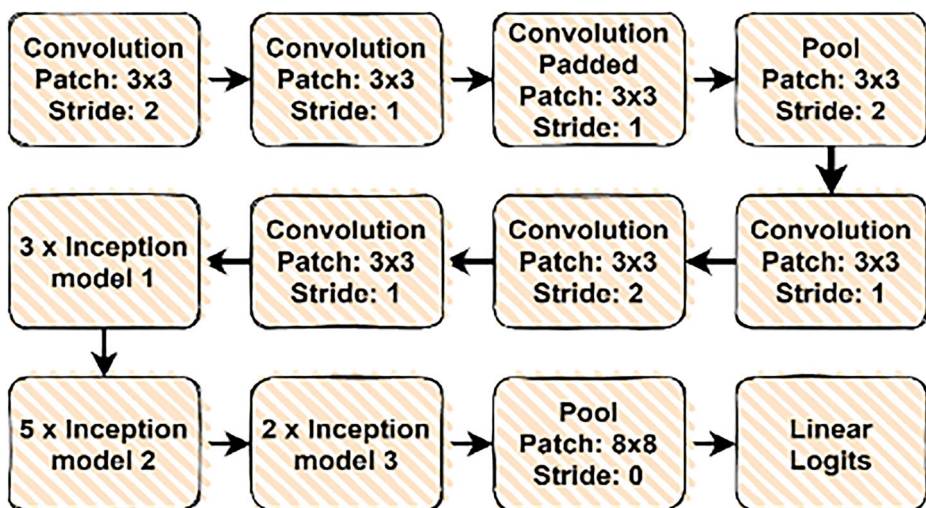


Fig. 9 Basic block diagram of Inception-v3

4.4.1 Model deployment process

In this work, the model was deployed to the Heroku cloud. Figure 10 shows the flow diagram of the proposed CNN model deployment process [33].

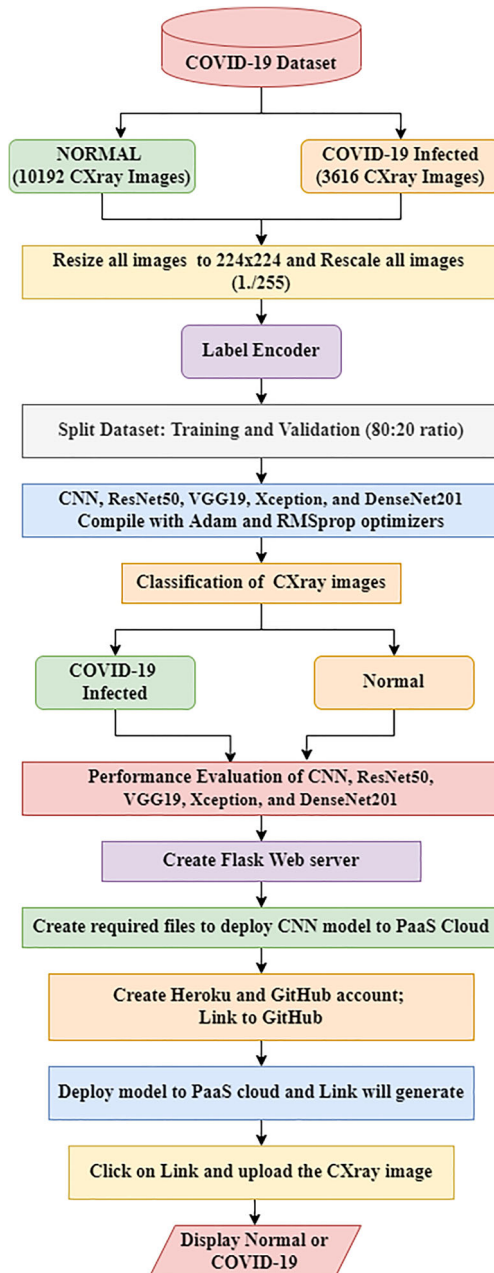


Fig. 10 CNN model deployment process on to the Heroku cloud

5 Results and discussion

The dataset includes 13,808 CX-Ray scans of Covid-19 (3616) and Normal (10,192). The database was separated into two portions (80:20): training (11,044 photos) and validation (2764 photos). The Adam and RMSprop optimizers, categorical cross-entropy loss function, and batch size of 32 were utilized.

5.1 Accuracy and AUC of CNN

Utilizing two optimizers (Adam and RMSprop), the suggested CNN model validates the different design options. This study helps evaluate model performance with various parameters and hyperparameters. As demonstrated in Fig. 11a and b, the CNN with Adam (CNN-A) and RMSprop (CNN-R) optimizers obtained 99.94 per cent and 99.98 per cent training (TA) and 98.81 per cent and 98.41 per cent validation accuracy (VA), respectively. Figure 11c and d illustrate the loss of the CNN models during training and validation, respectively. Figure 11e

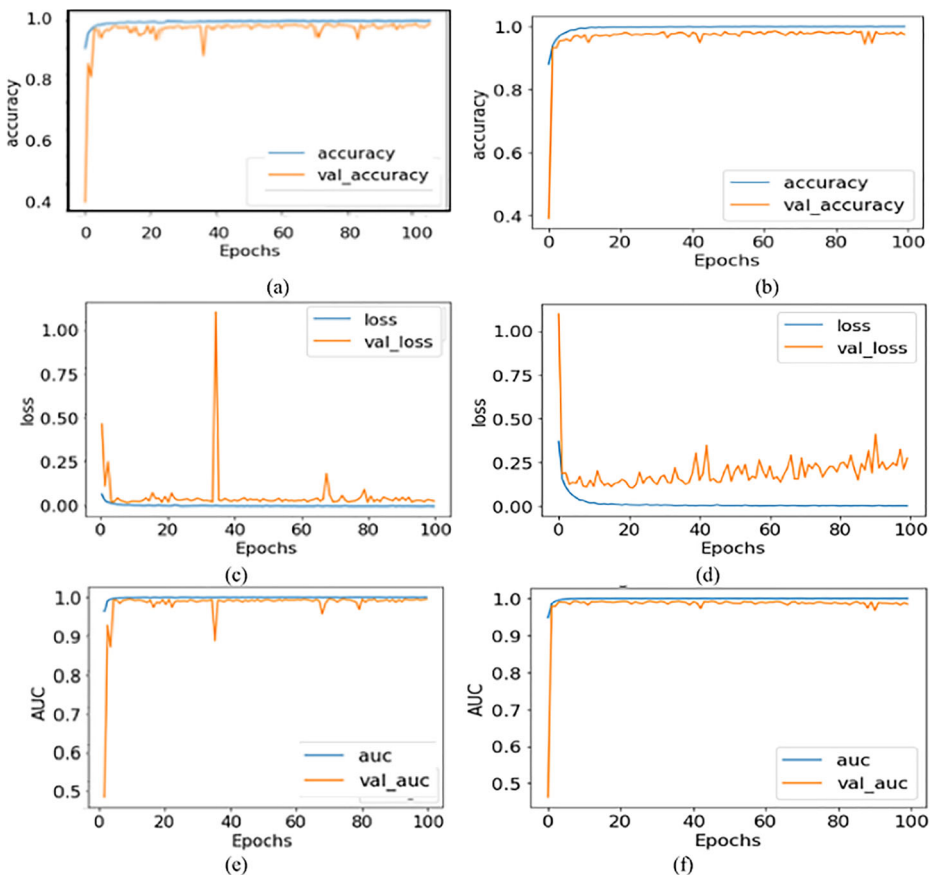


Fig. 11 a Training and validation accuracy of CNN-A, b CNN-R, c training and validation loss of CNN-A, d CNN-R, e training and validation AUC of CNN-A, and f CNN-R

and [f](#) demonstrate the AUC of the CNN models during training and validation, respectively. AUC varies from zero to one. A model with 100% erroneous identifications has an AUC of zero, while a model with 100% correct forecasts has an AUC of one. The AUC in this work is 0.9999 for the training dataset, indicating that the model predicts 99.99 per cent correctly.

5.2 Confusion matrix

The confusion matrix was used to monitor the effectiveness of every CNN and DCNN [\[31, 33\]](#).

$$\text{Acc}_{\text{CX}} = \frac{\text{TP}_{\text{CX}} + \text{TN}_{\text{CX}}}{\text{TP}_{\text{CX}} + \text{FP}_{\text{CX}} + \text{TN}_{\text{CX}} + \text{FN}_{\text{CX}}} \quad (2)$$

$$\text{Prec}_{\text{CX}} = \frac{\text{TP}_{\text{CX}}}{\text{TP}_{\text{CX}} + \text{FP}_{\text{CX}}} \quad (3)$$

$$\text{R}_{\text{CX}} = \frac{\text{TP}_{\text{CX}}}{\text{TP}_{\text{CX}} + \text{FN}_{\text{CX}}} \quad (4)$$

$$\text{F1}_{\text{CX}} = \frac{2}{\left(\frac{1}{\text{R}_{\text{CX}}}\right) + \left(\frac{1}{\text{Prec}_{\text{CX}}}\right)} \quad (5)$$

$$\text{TNR} = \frac{\text{TN}_{\text{CX}}}{\text{TN}_{\text{CX}} + \text{FP}_{\text{CX}}} \quad (6)$$

Where Acc_{CX} -Accuracy, TP_{CX} -Truly Positive, FP_{CX} -False Positive, TN_{CX} -Truly Negative, and FN_{CX} -False Negative, Prec_{CX} - Precision, R_{CX} -Recall, F1 score- F1_{CX} , Specificity or True Negative Rate (TNR).

5.3 Comparisons of CNN with DCNN

The performance of DCNN models was asses employing a confusion matrix with the proposed CNN model and compared in terms of the training accuracy (TA) and validation accuracy (VA) with loss is shown in [Fig. 12](#). The training accuracy (TA) of ResNet50-A, ResNet50-R, VGG19-A, VGG19-R Xception-A, Xception-R, Inception-V3-A, Inception-V3-R CNN-A, and CNN-R models were 94.09%, 90.78%, 100%, 99.92%, 100%, 100%, 100%, 100%, 99.94% and 99.98%, respectively. On the other hand, the VA of ResNet50-A, ResNet50-R, VGG19-A, VGG19-R Xception-A, Xception-R, DenseNet201-A, DenseNet201-R, CNN-A, and CNN-R models were 93.31%, 91.21%, 98.26%, 97.87%, 95.62%, 95.55%, 94.93%, 94.57%, 98.81%, and 98.41%, respectively. The TA of all models was above 99% except the ResNet50 model. The highest VA (98.81%) was achieved by the CNN-A model. The training AUC (TAUC) of ResNet50-A, ResNet50-R,

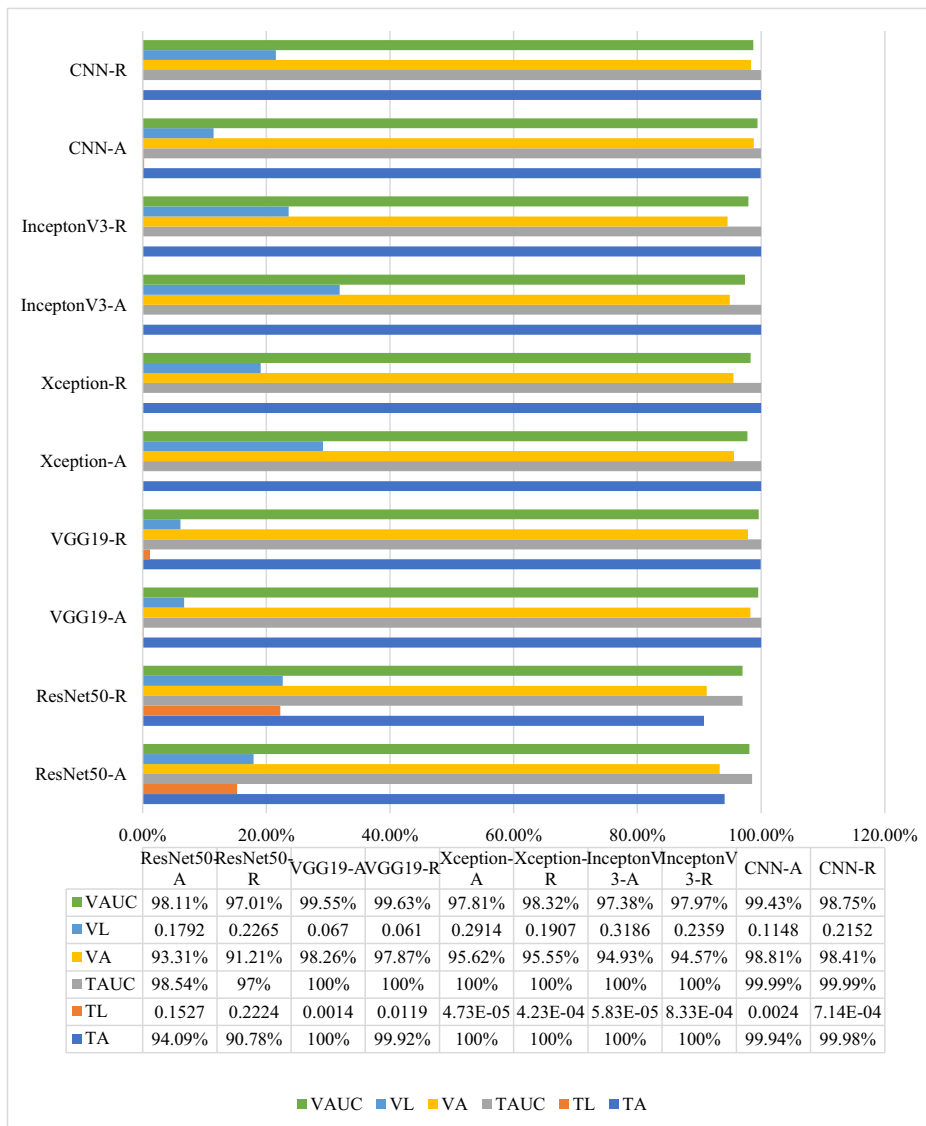


Fig. 12 Comparison of CNN and DCNN models

VGG19-A, VGG19-R, Xception-A, Xception-R, Inception-V3-A, Inception-V3-R, CNN-A, and CNN-R models were 98.54%, 97%, 100%, 100%, 100%, 100%, 100%, 100%, 99.99%, and 99.99%, respectively. The validation AUC (VAUC) of ResNet50-A, ResNet50-R, VGG19-A, VGG19-R, Xception-A, Xception-R, Inception-V3-A, Inception-V3-R, CNN-A, and CNN-R models were 98.11%, 97.01%, 99.55%, 99.63%, 97.81%, 98.32%, 97.38%, 97.97%, 99.43%, and 98.75%, respectively. The AUC performance of the VGG19-A, VGG19-R, and CNN-A was excellent. All models' training loss (TL) and validation loss (VL) were low. This performance indicated that CNN and DCNN model's performance was excellent without over-fitting.

Table 3 Model evaluation using confusion matrix on the training dataset

Models	TP	FP	FN	TN	Correct Prediction	Wrong Prediction
ResNet50-A	2563	336	139	8015	10,578	475
ResNet50-R	2668	231	581	7573	10,241	812
VGG19-A	2899	0	0	8154	11,053	0
VGG19-R	2899	0	0	8154	11,053	0
Xception-A	2899	0	0	8154	11,053	0
Xception-R	2899	0	0	8154	11,053	0
InceptionV3-A	2899	0	0	8154	11,053	0
InceptionV3-R	2899	0	0	8154	11,053	0
CNN-A	2899	0	0	8154	11,053	0
CNN-R	2899	0	0	8154	11,053	0

5.4 CNN and DCNN performance measures

Tables 3 and 4 summarize the performance of the CNN and DCNN models using Eqs. (2–5). Table 3 shows VGG19-A, VGG19-R, Xception-A, Xception-R, InceptionV3-A, InceptionV3-R, CNN-A, and CNN-R predicted all CX-Ray images (11,053) correctly. The ResNet50-A and ResNet50-R predicted 10,578 and 10,241 correctly and 475 and 812 images wrongly, respectively. It indicated that all models predicted 100% correct results on the training dataset except ResNet50-A and ResNet50-R models.

From Table 4, it can be seen that the ResNet50-A, ResNet50-R, VGG19-A, VGG19-R, Xception-A, Xception-R, InceptionV3-A, InceptionV3-R, CNN-A, and CNN-R, were predicted 2579, 2521, 2716, 2705, 2641, 2624, 2614, 2731, and 2720, CX-Ray images correctly and 185, 243, 48, 59, 121, 123, 140, 150, 33, and 44, predicted wrongly on the validation dataset, respectively. The CNN-A and CNN-R model predicted more images correctly compared to other DCNN models. The VGG-19 models also performed well in the prediction, but the Xception, InceptionV3, and ResNet50 performances were not up to the mark. The ResNet50-A and ResNet50-R mispredicted 185 and 243 CX-Ray images.

Table 4 Models evaluation using confusion matrix on the validation dataset

Models	TP	FP	FN	TN	Correct Prediction	Wrong Prediction
ResNet50-A	603	122	63	1976	2579	185
ResNet50-R	658	67	176	1863	2521	243
VGG19-A	700	25	23	2016	2716	48
VGG19-R	689	36	23	2016	2705	59
Xception-A	656	69	52	1987	2643	121
Xception-R	649	76	47	1992	2641	123
InceptionV3-A	637	88	52	1987	2624	140
InceptionV3-R	628	97	53	1986	2614	150
CNN-A	705	20	13	2026	2731	33
CNN-R	711	14	30	2009	2720	44

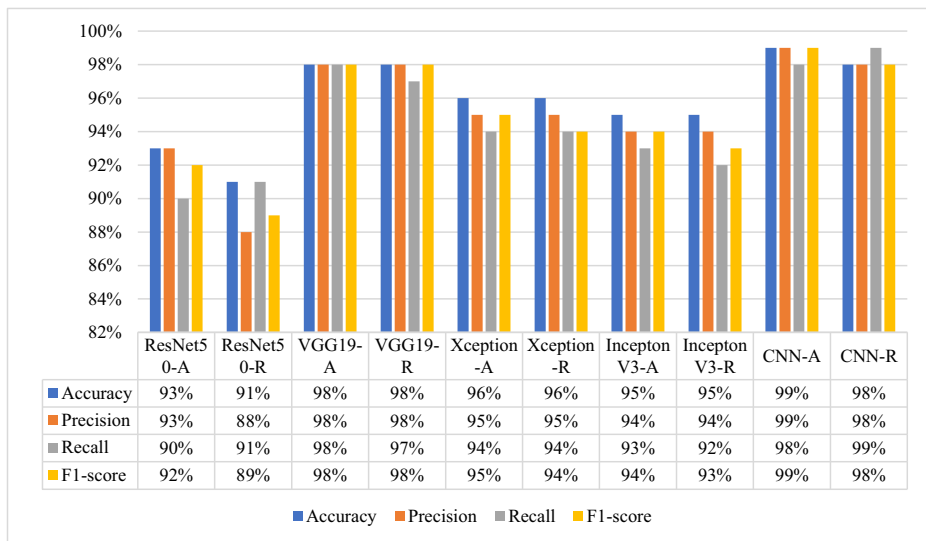


Fig. 13 CNN and DCNN models comparison

Figure 13 depicts graphical representations of evaluation criteria such as accuracy, precision, recall, and F1 score. The CNN-A (99%) and CNN-R (98%) models have fantastic Accuracy. However, if the data set is unbalanced, accuracy may not always be advantageous to model effectiveness. The precision measure aids in the monitoring of classification results. For an effective classifier, the precision value must be 1 (high). The precision of both the CNN-A and CNN-R models is 99 per cent and 98 per cent, respectively, indicating that both models performed well. A recall is the ratio of correctly identified positive cases to total positive instances. The recall value must be 1 (high) for an effective classifier. The CNN-A and CNN-R models have recall values of 98 and 99 per cent, indicating that the model performs well. Both the precision and recall measures reveal that only positive classes were examined, while the model should evaluate all categories.

As a result, an action that considers both precision and recall is essential. F1-score regards both precision and recall measurements. F1-score is the median of both measures and a better statistic to analyze model performance than accuracy [58]. These indicators aid in assessing the overall quality of the model. According to Fig. 13, the F1-score measures for CNN-A (99%) and CNN-R (98%) models were more significant than the other models. This gave a comprehensive that the suggested CNN models reliably categorize COVID-19 CX-Ray pictures. Furthermore, the F1 score represents the overall model performance. Regarding the F1-score measure, both CNN models perform more effectively than the DCNN models.

5.5 Receiver operator characteristic (ROC) curve

The ROC curves for ResNet50-A, ResNet50-R, VGG19-A, VGG19-R, Xception-A, Xception-R, InceptionV3-A, InceptionV3-R, CNN-A, and CNN-R on the training and

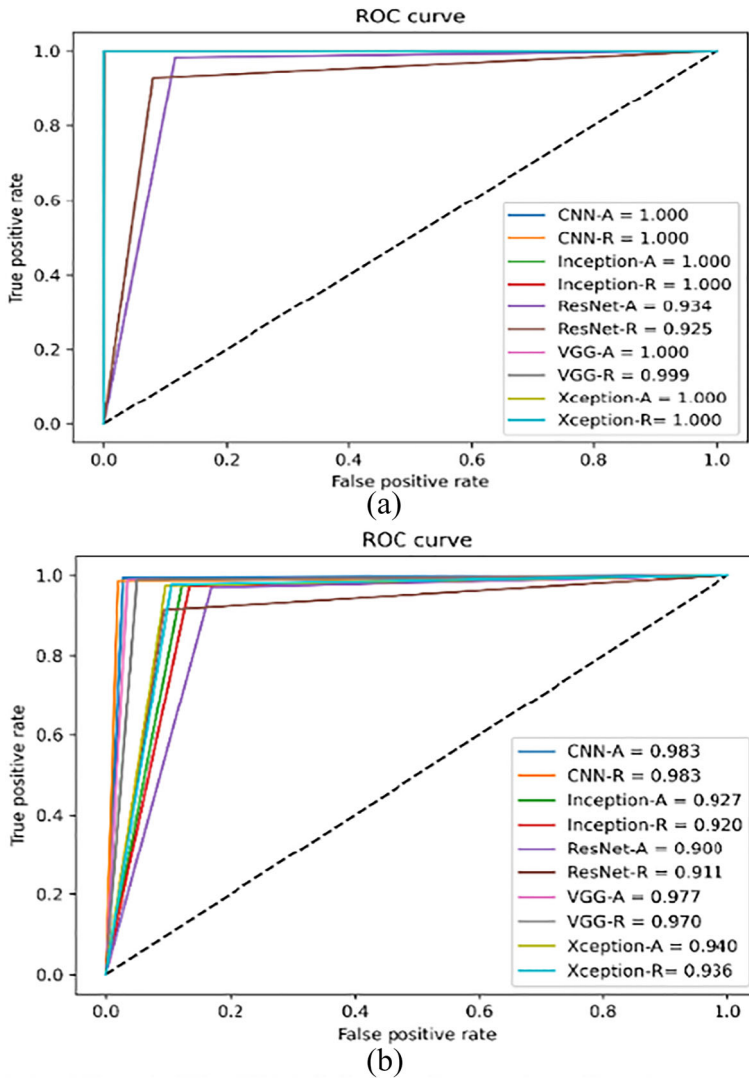


Fig. 14 **a** ROC curve for CNN-A, CNN-R, ResNet50, VGG-19, Xception, and InceptionV3 on training dataset and **b** validation dataset

validation datasets are shown in Fig. 14a and b. The ROC approach is an analytical method for assessing the efficacy of ML and data mining [16]. The ROC curve is a graphical representation of 2-class testing skills. The ROC curve should be closer to the upper left for improved classification performance. Except for ResNet50, the effectiveness of all models is good, as shown in Fig. 14a, and the curve is nearer the top left corner. As shown in Fig. 14b, the CNN-A, CNN-R, VGG-A, VGG-19-R, Xception-A, and Xception-R models function admirably. The plot demonstrates that the AUC for CNN-A and CNN-R is greater among models. The CNN-A and CNN-R models significantly outperformed in identifying COVID-19 CX-Ray pictures.

Table 5 Comparative analysis of CNN with DCNN

Models	Layers	Parameters	Memory Size (MB)	Execution time (Seconds) for 50 Steps
ResNet50-A	50	25,636,712	92.76	3000
ResNet50-R	50	25,636,712	92.02	3000
VGG19-A	26	143,667,240	77.05	3500
VGG19-R	26	143,667,240	76.86	3500
Xception-A	126	22,910,480	82.28	3000
Xception-R	126	22,910,480	81.51	3000
InceptionV3-A	159	23,851,784	85.19	2800
InceptionV3-R	159	23,851,784	84.8	2850
CNN-A	18	6,447,138	73.88	2500
CNN-R	18	6,447,138	49.28	2650

5.6 Cloud computing

In this section, the effectiveness of the CNN model is examined and compared with DCNN models in terms of layers, memory size, and parameters. The large-scale model causes significant issues with equipment energy usage and running speed. As a result, the effective implementation of a big-scale model in an embedded system or mobile device is problematic. When DL or ML are deployed on mobile devices, the accuracy of the model's outputs is not the sole consideration. As a result, creating a model with lesser parameters and a smaller memory capacity is necessary. These are critical when deploying the model to the cloud due to the associated cost. Table 5 compares the CNN with DCNN models in depth, model size, parameters, and F1 score.

The F1-score of the CNN models are 99 per cent, the highest among the models, and the CNN models contain 18 layers, which are fewer than the ResNet50 (50 layers), VGG19 (26 layers), Xception (126 layers), and InceptionV3 (159 layers) models. Generally, the number of layers enhances productivity, but a learning approach is more complicated. Other DCNN models have more parameters than the suggested CNN model. The CNN has the least memory capacity among the models at 49.28 MB. As a result of its small size, fewer parameters, and overall efficiency, the CNN model is appropriate for the proposed COVID-19 illness forecasting system using CX-Ray pictures and is suitable for deployment to the Heroku cloud. This study deploys the CNN model with Adam optimizer (CNN-A) to the cloud. The Heroku cloud link was created after the model was successfully deployed. The website opens with a single click on the Heroku cloud link, accessible on mobile devices, to submit a CX-Ray picture. The following step is to send the CX-Ray image. After sending the image to the cloud, the algorithm successfully identifies the COVID of the non-COVID individual.

Several computer-based methods have been developed for COVID-19 disease identification. The literature review in the related work section clarified various methods for identifying and classifying COVID-19 disease using CX-Ray and CT scans. Table 1 tabulates a systematic comparison of the present work CNN model approach with other ML/DL approaches. In this work, the authors use CNN and DCNN models for COVID-19 prediction due to automatic feature extraction in the CNN framework. Most researchers used a similar methodology, but either they achieved less accuracy, more parameters were used, or the model size was significant compared to our work.

Das et al. [11] used DenseNet201, ResNet50V2, and InceptionV3 models for differentiating COVID-19 from CX-Rays and achieved 91.62 per cent accuracy, but the size: 80 MB, 232 MB, and 92 MB [71], number of parameters: 20.2 M, 25.6 M, and 23.9 M [71], and depth: 402, 307, and 189 [71] of these models were pretty high compared to our proposed CNN model. The accuracy was also less compared to our method. These parameters are essential in model deployment because of the speed and cost involved. Similarly, Narin et al. [46] utilized DCNN models: InceptionV3, ResNet50, and InceptionResNetV2, Sethy et al. [62] utilized ResNet50, but the size, layers, and parameters are more. Another work proposed by Guefrechi et al. [19] used ResNet50, InceptionV3, and VGG16 and achieved an accuracy of 97.20 per cent, 98.10 per cent, and 98.30 per cent, respectively, which have more parameters, size, and depth and less accuracy. Rehman et al. [56] and Kumar et al. [29] also proposed a similar type of work, but the accuracy is less.

Reshi et al. [57] presented a DCNN model for the diagnosis of COVID-19 with image processing. They achieved 99.5 per cent, but the DCNN model has 38 layers, including six convolutional (Conv2D), six max-pooling, six dropouts, eight activation functions, eight BN, one flattens, and three dense layers. The accuracy of this method was higher than our work, but our proposed model has a total of 18 layers, including 4 convolutional, 5 BN, 3 average pooling, 4 dropouts, 1 flattened, and 2 dense layers.

Das et al. [12] suggested a CNN, VGG-16, and ResNet-50-based Covid-19 testing methodology, achieving an accuracy of 97.67 per cent. The TLCoV accuracy was more diminutive, and the total parameters used were 1,24,10,023, higher than our proposed model of 6,447,138. Indumathi et al. [24] presented a DL-based technique for predicting COVID-19 and achieved 98.06 per cent accuracy, which is less than our work. Similarly, the researcher suggested an SVM-based approach [60], DeTraC [1], CoroNet [27], AlexNet [39], Ozturk et al. [50], Yan et al. [73], Apostolopoulos et al. [4], Farooq et al. [15], and Meraihi et al. [43], but the accuracy was more diminutive.

Zhang et al. [75] used CAAD to identify non-viral and viral pneumonia pictures and achieved an AUC of 83.61% and a sensitivity of 71.70%, lower than our proposed model. Alkhodari et al. [3] proposed that CNN-BiLSTM achieved an average classification accuracy of 94.58 per cent. It was also performed to detect COVID-19 individuals with a high sensitivity of 94.21 per cent, specificity of 94.96 per cent, and AUCROC curves of 0.90, which is less than our proposed method.

Most researchers employed CNN and DCNN approaches to determine COVID-19 illness, and some achieved greater than 99 per cent accuracy. However, these models' size, depth, and parameters were greater. As a result, the custom CNN model should be designed to improve system accuracy and performance with a small size, fewer layers, and fewer parameters.

The authors investigate various CNN network hyperparameter combinations to achieve the best computational cost and performance. The input image size 224×224 was selected. The kernel size of 3×3 was chosen for optimum feature extraction with an average pooling layer [32]. BN layers were used to normalize the previous and subsequent layer's output after the convolutional layer and before the next convolutional layer. The dropout layer with 0.2 ignores 20 per cent of neurons during training to help avoid the overfitting problem. This research also focuses on the learning rate context and vector reduction of input data vectors. The Adam optimizer is chosen with a learning rate of 0.0003. The CNNs flattened layer has 50,176 extracted features used for classification using dense layers. In ML/DL, the results were often achieved at the 10th or 5000 epochs.

In this work, initially, the authors initialize epochs equal to 300; the callback method was employed to terminate the training when validation precision equals 98.81 per cent and stopped after 106 iterations. A callback is an entity that can do tasks at different training levels, like at the start or end of an epoch, before or after a single batch, etc. In ML/DL techniques, the loss function must be minimized. ML/DL algorithms frequently become trapped in local minima during model training. When there are substantial loss hurdles between local minima, global optimization issues can be exceedingly challenging. To address the problem of local minima, the authors carefully chose features, learning rate, and the number of epochs. The learning rate was initially assigned from 0.1 to 0.00001, and checked the model outcome. Additionally, the convolutional layers increased from one to three, initializing a more considerable learning rate and settling on 0.0003 with the Adam optimizer. The proposed CNN model performs well compared to the literature in Table 1. Aside from that, the model is straightforward and may be accessed via mobile phones or desktops.

6 Conclusion and recommendations

Detecting and diagnosing early-stage diseases in the health industry is always challenging. The traditional approach to detection is time-consuming, so there is a need to develop an efficient system small in size to detect COVID-19. The fundamental purpose of this effort is to support health proficiency in regions where radiotherapists are in short supply. In this work, CNN and DCNN were used to examine the COVID CX-Ray images of both COVID and non-COVID patients. The selection of the CNN and DCNN parameters and hyperparameters is essential in terms of performance, size, and computational time. These parameters were selected carefully so that the model's size should remain small with fewer CNN layers and parameters. First, the CNN model was developed with 18 layers, and then four DCNN (ResNet50, VGG19, InceptionV3, and Xception) models were implemented.

The performance of these models was compared, and it was found that the CNN model performs very well. The CNN-R achieved a 98.41 per cent validation accuracy, 98.75 per cent AUC and 98 per cent F1 score, which was small in size (49.28 MB) and had fewer parameters (6,447,138) and had an execution time of 2650 s for 50 iterations. Similarly, the CNN-A achieved a 98.81 per cent validation accuracy, 99.43 per cent AUC, and 99 per cent F1 score, which was small in size (73.88 MB) and had fewer parameters (6,447,138) and had an execution time of 2500 s for 50 iterations. The VGG-19 models also achieved similar results, but the size (77.05 MB), parameters (143,667,240), and execution time of 3500. The CNN-A is the most suitable model to deploy to the PaaS cloud. The main benefit of model deployment is that the prediction system can be accessible on mobile or laptop. Due to this, the proposed approach may help detect diseases efficiently and quickly. It reduces physicians' burden and speeds up the testing procedures for Covid-19 positive patients. This paper implemented the cloud-based framework to classify COVID-19 diseases successfully. The CNN with the petite size was deployed to the cloud successfully. The created deployed Heroku link was verified on a mobile phone and performed well. The benefit of this method is that the model is accessible anywhere on the mobile or embedded device [51, 52]. There is no need for extra hardware or device required. The CX-Ray image can be taken using a CX-Ray machine, and the CX-Ray image can be shared on any mobile. This CX-Ray image can be uploaded to the cloud to

predict the COVID-19 disease. The smaller model was also deployed to the cloud, which none of the authors did in the literature.

The authors will continue to extend this study to classify more types of CT scans with X-ray images. Various research gaps can be addressed in additional articles based on the material given. The limitation of this work is that it will predict only COVID-19 and non-COVID-19 patients. The proposed approach can be improved in the following ways:

- The CNN and ML classifiers, such as SVM, random forest, K nearest neighbour, etc., can be integrated so that CNN-extracted features can be given to the classifiers [34, 35].
- The presented model can be developed even more to provide stage-specific COVID-19 diagnosis.
- This research could be expanded to include the classification of the chest cavity, pneumonia, lung cancer, and other respiratory illnesses.

Funding The authors did not receive support from any organization for the submitted work.

Data availability The Radiography Database is available on <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>.

Declarations

Conflict of interest/Competing interests The authors declare that they have no conflict of interest. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Abbas A, Abdelsamea MM, Gaber MM (2021) Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Appl Intell* 51:854–864. <https://doi.org/10.1007/s10489-020-01829-7>
2. Alam T (2020) Cloud computing and its role in the information technology. *SSRN J*. <https://doi.org/10.2139/ssrn.3639063>
3. Alkhodari M, Khandoker AH (2022) Detection of COVID-19 in smartphone-based breathing recordings: a pre-screening deep learning tool. *PLoS ONE* 17:e0262448. <https://doi.org/10.1371/journal.pone.0262448>
4. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 43:635–640. <https://doi.org/10.1007/s13246-020-00865-4>
5. Ayalew AM, Salau AO, Abeje BT, Enyew B (2022) Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients. *Biomed Signal Process Control* 74:103530. <https://doi.org/10.1016/j.bspc.2022.103530>
6. Carretero J, Blas JG (2014) Introduction to cloud computing: platforms and solutions. *Cluster Comput* 17: 1225–1229. <https://doi.org/10.1007/s10586-014-0352-5>
7. Chollet F (2016) Xception: deep learning with depthwise separable convolutions. <https://doi.org/10.48550/ARXIV.1610.02357>
8. Chowdhury MEH, Rahman T, Khandakar A et al (2020) Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* 8:132665–132676. <https://doi.org/10.1109/ACCESS.2020.3010287>
9. COVID-19 Radiography Database. <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>. Accessed 12 July 2022
10. Dádario AMV, de Paiva JPQ, Chate RC et al (2020) Coronavirus disease 2019 deep learning models: methodologic considerations. *Radiology* 296:E192–E192. <https://doi.org/10.1148/radiol.2020201178>

11. Das AK, Ghosh S, Thunder S et al (2021) Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network. *Pattern Anal Applic* 24:1111–1124. <https://doi.org/10.1007/s10044-021-00970-4>
12. 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* 144:110713. <https://doi.org/10.1016/j.chaos.2021.110713>
13. de Venâncio PVAB, Lisboa AC, Barbosa AV (2022) An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Comput & Applic.* <https://doi.org/10.1007/s00521-022-07467-z>
14. Eftekhari A, Alipour M, Chodari L et al (2021) A comprehensive review of detection methods for SARS-CoV-2. *Microorganisms* 9:232. <https://doi.org/10.3390/microorganisms9020232>
15. Farooq M, Hafeez A (2020) COVID-ResNet: a deep learning framework for screening of COVID19 from Radiographs. <https://doi.org/10.48550/ARXIV.2003.14395>
16. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognit Lett* 27:861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
17. Garbin C, Zhu X, Marques O (2020) Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimed Tools Appl* 79:12777–12815. <https://doi.org/10.1007/s11042-019-08453-9>
18. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings*, pp 315–323
19. Guefrechi S, Jabra MB, Ammar A et al (2021) Deep learning based detection of COVID-19 from chest X-ray images. *Multimed Tools Appl* 80:31803–31820. <https://doi.org/10.1007/s11042-021-11192-5>
20. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. <https://doi.org/10.48550/ARXIV.1512.03385>
21. Huang J, Rathod V, Sun C, Recognition P et al (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, pp 3296–3297
22. Huang C, Wang Y, Li X et al (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* 395:497–506. [https://doi.org/10.1016/S0140-6736\(20\)30183-5](https://doi.org/10.1016/S0140-6736(20)30183-5)
23. Hutchison D, Kanade T, Kittler J et al (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: Diamantaras K, Duch W, Iliadis LS et al (eds) *Artificial Neural Networks – ICANN 2010*. Springer Berlin Heidelberg, Berlin, pp 92–101
24. Indumathi N, Shanmuga Eswari M, Salau AO et al (2022) Prediction of COVID-19 outbreak with current substantiation using machine learning algorithms. In: Tyagi AK, Abraham A, Kaklauskas A et al (eds) *Intelligent Interactive multimedia systems for e-healthcare applications*. Springer Singapore, Singapore, pp 171–190
25. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. <https://doi.org/10.48550/ARXIV.1502.03167>
26. Jordao A, Akio F, Lie M, Schwartz WR (2020) Stage-wise neural architecture search. <https://doi.org/10.48550/ARXIV.2004.11178>
27. Khan AI, Shah JL, Bhat MM (2020) CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Comput Methods Programs Biomed* 196:105581. <https://doi.org/10.1016/j.cmpb.2020.105581>
28. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60:84–90. <https://doi.org/10.1145/3065386>
29. Kumar N, Gupta M, Gupta D, Tiwari S (2021) Novel deep transfer learning model for COVID-19 patient detection using X-ray chest images. *J Ambient Intell Human Comput.* <https://doi.org/10.1007/s12652-021-03306-6>
30. Kumar V, Kalam S, Das AK, Sinha D (2021) Attack detection scheme using deep learning approach for IoT. In: Chaki R, Chaki N, Cortesi A, Saeed K (eds) *Advanced computing and systems for security: volume 14*. Springer Singapore, Singapore, pp 17–30
31. Lanjewar MG, Gurav OL (2022) Convolutional neural networks based classifications of soil images. *Multimed Tools Appl* 81:10313–10336. <https://doi.org/10.1007/s11042-022-12200-y>
32. Lanjewar MG, Panchbhai KG (2022) Convolutional neural network based tea leaf disease prediction system on smart phone using paas cloud. *Neural Comput & Applic.* <https://doi.org/10.1007/s00521-022-07743-y>
33. Lanjewar MG, Morajkar PP, Parab J (2022) Detection of tartrazine colored rice flour adulteration in turmeric from multi-spectral images on smartphone using convolutional neural network deployed on PaaS cloud. *Multimed Tools Appl* 81:16537–16562. <https://doi.org/10.1007/s11042-022-12392-3>

34. Lanjewar MG, Parab JS, Shaikh AY (2022) Development of framework by combining CNN with KNN to detect Alzheimer's disease using MRI images. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-022-13935-4>
35. Lanjewar MG, Parab JS, Shaikh AY, Sequeira M (2022) CNN with machine learning approaches using ExtraTreesClassifier and MRMR feature selection techniques to detect liver diseases on cloud. *Cluster Comput*. <https://doi.org/10.1007/s10586-022-03752-7>
36. Li X (2021) Artificial intelligence neural network based on intelligent diagnosis. *J Ambient Intell Human Comput* 12:923–931. <https://doi.org/10.1007/s12652-020-02108-6>
37. Li X, Chen S, Hu X, Yang J (2019) Understanding the disharmony between dropout and batch normalization by variance shift. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, pp 2677–2685
38. Lo WW, Yang X, Wang Y (2019) An Xception convolutional neural network for malware classification with transfer learning. In: 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS). IEEE, Canary Islands, pp 1–5
39. Maghdid H, Asaad AT, Ghafoor KZG et al (2021) Diagnosing COVID-19 pneumonia from x-ray and CT images using deep learning and transfer learning algorithms. In: Ağaian SS, Jassim SA, DelMarco SP, Asari VK (eds) *Multimodal image exploitation and learning 2021*. SPIE, Online Only, United States, p 26
40. Makris A, Kontopoulos I, Tserpes K (2020) COVID-19 detection from chest X-Ray images using deep learning and convolutional neural networks. In: 11th Hellenic Conference on Artificial Intelligence. ACM, Athens, pp 60–66
41. Mansour NA, Saleh AI, Badawy M, Ali HA (2022) Accurate detection of Covid-19 patients based on Feature Correlated Naïve Bayes (FCNB) classification strategy. *J Ambient Intell Human Comput* 13:41–73. <https://doi.org/10.1007/s12652-020-02883-2>
42. Mateen M, Wen J, Nasrullah et al (2018) Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry* 11:1. <https://doi.org/10.3390/sym11010001>
43. Meraih Y, Gabis AB, Mirjalili S et al (2022) Machine learning-based research for OVID-19 detection, diagnosis, and prediction: a survey. *SN Comput Sci* 3:286. <https://doi.org/10.1007/s42979-022-01184-z>
44. MK MV, Atalla S, Almuraqab N, Moonesar IA (2022) Detection of COVID-19 using deep learning techniques and cost effectiveness evaluation: a survey. *Front Artif Intell* 5:912022. <https://doi.org/10.3389/frai.2022.912022>
45. Murugan R, Goel T (2021) E-DiCoNet: Extreme learning machine-based classifier for diagnosis of COVID-19 using deep convolutional network. *J Ambient Intell Human Comput* 12:8887–8898. <https://doi.org/10.1007/s12652-020-02688-3>
46. Narin A, Kaya C, Pamuk Z (2021) Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Anal Applic* 24:1207–1220. <https://doi.org/10.1007/s10044-021-00984-y>
47. Natnael T, Alemnew Y, Berihun G et al (2021) Facemask wearing to prevent COVID-19 transmission and associated factors among taxi drivers in Dessie City and Kombolcha Town, Ethiopia. *PLoS ONE* 16: e0247954. <https://doi.org/10.1371/journal.pone.0247954>
48. Ng M-Y, Lee EYP, Yang J et al (2020) Imaging profile of the COVID-19 infection: radiologic findings and literature review. *Radiology: Cardiothorac Imaging* 2:e200034. <https://doi.org/10.1148/ryct.2020200034>
49. Nguyen LD, Lin D, Lin Z, Cao J (2018) Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, Florence, pp 1–5
50. Ozturk T, Talo M, Yildirim EA et al (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med* 121:103792. <https://doi.org/10.1016/j.combiomed.2020.103792>
51. Parab J, Sequeira M, Lanjewar M et al (2021) Backpropagation neural network-based machine learning model for prediction of blood urea and glucose in CKD patients. *IEEE J Transl Eng Health Med* 9:1–8. <https://doi.org/10.1109/JTEHM.2021.3079714>
52. Parab J, Sequeira M, Lanjewar M et al (2022) Blood glucose prediction using machine learning on jetson nanoplatfrom. In: Manshahia MS, Kharchenko V, Munapo E et al (eds) *Handbook of intelligent computing and optimization for sustainable development*, 1st edn. Wiley, pp 835–848
53. Qi X, Brown LG, Foran DJ et al (2021) Chest X-ray image phase features for improved diagnosis of COVID-19 using convolutional neural network. *Int J CARS* 16:197–206. <https://doi.org/10.1007/s11548-020-02305-w>
54. Rahaman MM, Li C, Yao Y et al (2020) Identification of COVID-19 samples from chest X-Ray images using deep learning: a comparison of transfer learning approaches. *XST* 28:821–839. <https://doi.org/10.3233/XST-200715>

55. Rahimzadeh M, Attar A (2020) A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Inf Med Unlocked* 19:100360. <https://doi.org/10.1016/j.imu.2020.100360>
56. Rehman A, Naz S, Khan A et al (2020) Improving coronavirus (COVID-19) diagnosis using deep transfer learning. *Infectious Diseases (except HIV/AIDS)*
57. Reshi AA, Rustam F, Mehmood A et al (2021) An efficient CNN model for COVID-19 disease detection based on X-Ray image classification. *Complexity* 2021:1–12. <https://doi.org/10.1155/2021/6621607>
58. Rostad S (2018) What is Heroku? A simple explanation for non-techies. In: TriFin Labs. <https://trifinlabs.com/what-is-heroku/>. Accessed 12 July 2022
59. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: *Proceedings of the 24th international conference on Machine learning - ICML '07*. ACM Press, Corvallis, pp 791–798
60. Salau AO (2021) Detection of corona virus disease using a novel machine learning approach. In: *2021 International Conference on Decision Aid Sciences and Application (DASA)*. IEEE, Sakheer, pp 587–590
61. Salau AO, Jain S (2019) Feature extraction: a survey of the types, techniques, applications. In: *2019 International Conference on Signal Processing and Communication (ICSC)*. IEEE, Noida, pp 158–164
62. Sethy PK, Behera SK, Ratha PK, Biswas P (2020) Detection of coronavirus Disease (COVID-19) based on deep features and support vector machine. *Int J Math Eng Manag Sci* 5:643–651. <https://doi.org/10.33889/IJMEMS.2020.5.4.052>
63. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. <https://doi.org/10.48550/ARXIV.1409.1556>
64. Singh D, Kumar V, Vaishali, Kaur M (2020) Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. *Eur J Clin Microbiol Infect Dis* 39:1379–1389. <https://doi.org/10.1007/s10096-020-03901-z>
65. Singhal T (2020) A review of Coronavirus Disease-2019 (COVID-19). *Indian J Pediatr* 87:281–286. <https://doi.org/10.1007/s12098-020-03263-6>
66. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958
67. Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Boston, pp 1–9
68. Szegedy C, Vanhoucke V, Ioffe S et al (2015) Rethinking the inception architecture for computer vision. <https://doi.org/10.48550/ARXIV.1512.00567>
69. Sunyaev A (2020) Cloud computing. In: *Internet computing*. Springer International Publishing, Cham, pp 195–236
70. Targ S, Almeida D, Lyman K (2016) Resnet in resnet: generalizing residual architectures. <https://doi.org/10.48550/ARXIV.1603.08029>
71. Team K Keras documentation: Keras Applications. <https://keras.io/api/applications/>. Accessed 12 July 2022
72. Yadessa AG, Salau AO (2021) Low cost sensor based hand washing solution for COVID-19 prevention. In: *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*. IEEE, Zallaq, pp 93–97
73. Yan Q, Wang B, Gong D et al (2020) COVID-19 chest CT image segmentation – a deep convolutional neural network solution. <https://doi.org/10.48550/ARXIV.2004.10987>
74. Yuen K-S, Ye Z-W, Fung S-Y et al (2020) SARS-CoV-2 and COVID-19: the most important research questions. *Cell Biosci* 10:40. <https://doi.org/10.1186/s13578-020-00404-4>
75. Zhang J, Xie Y, Pang G et al (2021) Viral pneumonia screening on chest X-Rays using confidence-aware anomaly detection. *IEEE Trans Med Imaging* 40:879–890. <https://doi.org/10.1109/TMI.2020.3040950>
76. Zhao J, Yuan Q, Wang H et al (2020) Antibody responses to SARS-CoV-2 in patients with novel coronavirus disease 2019. *Clin Infect Dis* 71:2027–2034. <https://doi.org/10.1093/cid/ciaa344>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.