



## AI-based wavelet and stacked deep learning architecture for detecting coronavirus (COVID-19) from chest X-ray images<sup>☆</sup>

Rajkumar Soundrapandian<sup>a</sup>, Himanshu Naidu<sup>b</sup>, Marimuthu Karuppiah<sup>c,\*</sup>, M. Maheswari<sup>d</sup>, Ramesh Chandra Poonia<sup>e</sup>

<sup>a</sup> School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India

<sup>b</sup> ServiceNow, Hyderabad, Telangana 500081, India

<sup>c</sup> School of Computer Science and Engineering & Information Science, Presidency University, Bengaluru, Karnataka 560064, India

<sup>d</sup> Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai 600119, India

<sup>e</sup> Department of Computer Science, CHRIST (Deemed to be University), Bengaluru, Karnataka 560029, India



### ARTICLE INFO

Dataset link: <https://github.com/himanshunaid/u/wavstacovnet-19>

#### Keywords:

WavStaCovNet-19

COVID-19

X-ray images

Deep learning

Data augmentation

ResNet50

VGG19

Xception

DarkNet19

Discrete wavelet transform

Accuracy

### ABSTRACT

A novel coronavirus (COVID-19), belonging to a family of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), was identified in Wuhan city, Hubei, China, in November 2019. The disease had already infected more than 681.529665 million people as of March 13, 2023. Hence, early detection and diagnosis of COVID-19 are essential. For this purpose, radiologists use medical images such as X-ray and computed tomography (CT) images for the diagnosis of COVID-19. It is very difficult for researchers to help radiologists to do automatic diagnoses by using traditional image processing methods. Therefore, a novel artificial intelligence (AI)-based deep learning model to detect COVID-19 from chest X-ray images is proposed. The proposed work uses a wavelet and stacked deep learning architecture (ResNet50, VGG19, Xception, and DarkNet19) named WavStaCovNet-19 to detect COVID-19 from chest X-ray images automatically. The proposed work has been tested on two publicly available datasets and achieved an accuracy of 94.24% and 96.10% on 4 classes and 3 classes, respectively. From the experimental results, we believe that the proposed work can surely be useful in the healthcare domain to detect COVID-19 with less time and cost, and with higher accuracy.

### 1. Introduction

A certain strain of coronavirus, also referred to as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), causes coronavirus disease. The first case of coronavirus was identified in November 2019 in Wuhan city, China [1]. Since then, this contagious disease has been spreading worldwide and has created a pandemic situation. As of March 13, 2023, 681.529665 million confirmed cases, 6.811898 million confirmed death cases and 654.429347 million recovered cases were recognized across the globe [2]. The profuse demand for medical treatment and vaccines critically affects all the developed and developing countries, such as the USA, India, and Brazil. Reports have shown that this coronavirus can easily be transmitted from person to person if there are no safety measures, such as wearing masks and social distancing. In this context, early detection of the virus-affected people and isolation could be key factor in restricting the transmission of the virus to healthy populations.

<sup>☆</sup> This paper is for special section VSI-covid. Reviews were processed by Guest Editor Dr. Sunil Kumar Singh and recommended for publication.

\* Corresponding author.

E-mail address: [marimuthume@gmail.com](mailto:marimuthume@gmail.com) (M. Karuppiah).

Currently, reverse transcription polymerase chain reaction (RT-PCR) is being used to detect and screen COVID-19 patients [3]. This test is quite time-consuming, as it takes a minimum of four to six hours, sometimes even more, to generate a result. Furthermore, the total positive rate of RT-PCR was reported to be 30%–60%, which could easily mislabel a virus-affected person as unaffected. This may lead to failure in the isolation of affected patients, and thus, a huge population of healthy people could be infected. An alternative method for COVID-19 detection is to analyse chest scan images such as X-rays and CTs. Chest scan images are regularly used for the diagnosis of pneumonia. Chest CT images have an advanced sensitivity for the diagnosis of COVID-19, and X-rays show the visual indices correlated with COVID-19. The descriptions of COVID-19 pneumonia chest images are denoted by the presence of ground glass opacities (GGOs). In earlier diagnoses of COVID-19, the GGO pattern was recognized by an obscured pulmonary vascular area, and these abnormal patterns were being interpreted only by physicians/radiology experts. Considering the large number of image samples provided to the small number of trained physicians/radiologists, the rate of early diagnosis with higher accuracy could be increased by automatic diagnosis (of such abnormalities) procedures. Technologies such as AI, machine learning (ML), and deep learning (DL) are considered efficient tools for tackling such problems.

At present, a publicly small COVID-19 image dataset is available, which could help AI researchers train ML models for the automatic detection of COVID-19 patients from chest X-ray images. This work employs one such small dataset, which was relabelled with the help of an experienced radiologist. In addition, for negative COVID-19, sample subsets of viral pneumonia, bacterial pneumonia, and normal images from the chest X-ray pneumonia dataset were used. The combined dataset had 5475 chest X-ray images.

In this paper, a computer-aided diagnosis system (CADs) using a deep learning framework (WavStaCovNet-19) is proposed, which is used to identify COVID-19 efficiently, economically, and with low misclassification rates. The CADs is designed by pretrained deep learning models (ResNet50, VGG19, Xception, DarkNet19) to extract distinct features from the COVID-19 chest X-ray images. These pretrained models are then combined by the stacking technique WavStaCovNet-19. The WavStaCovNet-19 was trained and tested with 4 classes (COVID-19, viral pneumonia, bacterial pneumonia, normal), 3 classes (COVID-19, pneumonia, normal), and binary classes (COVID-19, non-COVID-19). Since the proposed WavStaCovNet-19 achieved significantly accurate classification results, it can effectively be applied in real-world COVID-19 diagnosis scenarios.

### 1.1. Contributions

The major contributions of the paper are:

- For the classification and diagnosis of COVID-19, the proposed WavStaCovNet-19 framework fetches the input chest X-ray images in the wavelet image format and then gives them to the stacking model for distinct feature extraction.
- The dataset consists of 5475 images incorporating COVID-19, viral pneumonia, bacterial pneumonia, and normal X-ray images, where the image labels were prepared by a skilled radiologist.
- A customized loss function is applied in the proposed WavStaCovNet-19 model for identifying COVID-19 positive X-ray images.
- The WavstacovNet-19 model was developed to help physicians identify COVID-19 more effectively and efficiently.

### 1.2. Organization of paper

The rest of the paper is organized as follows. Section 2 briefly presents related work of the COVID-19 detection models that use chest X-ray images and CT images. Section 3 outlines the preliminary concepts required to develop the proposed method. Section 4 elaborately presents the proposed WavStaCovNet-19 model. The experimental results and a performance analysis are discussed in Sections 5 and 6 concludes the paper.

## 2. Related work

This section reviews some recent progress made in the fields of AI, and DL techniques for the detection of COVID-19. Table 1 summarizes various methods, modalities, number of images used, and classification types of the state-of-the-art. Wherein, the detection and diagnosis of COVID-19 are carried out using X-ray and CT images. Irrespective of the type of image used, all the mentioned works in Table 1 strived to gain the maximum accuracy rate. For instance, the rate of sensitivity and specificity of the works related to the detection of COVID-19 using X-ray images are as follows:

The deep learning-based anomaly detection model proposed by Zhang et al. [4] attains 96.00% sensitivity (COVID-19 cases), 70.65% specificity (non-COVID-19 cases) and, 95.13% area under the curve rate. The inceptionv3 for binary classification (COVID-19 and normal) model developed by Salman et al. [5] obtained 100% accuracy, sensitivity, specificity, and an F1-score. The DeTrac model constructed by Abbas et al. [6] achieved an accuracy of 95.12% for multiclass classification. A COVID-Net model was built by Wang et al. [7], gave an accuracy of 93.3%. Sethy [8] introduced the ResNet50+SVM combination model achieved the highest accuracy of 98.66% for the 3-class (COVID-19, pneumonia, normal) problem. The transfer learning model designed by Apostolopoulos and Mpesiana [9] achieved an accuracy of 98.75% for 2 classes, while an accuracy of 93.48% was obtained for 3 classes. Kassania et al. [27] deployed automatic detection using various deep architectures (MobileNet, DenseNet, Xception, ResNet, InceptionV3, InceptionRes-NetV2, VGGNet, NASNet), generated a 96% accuracy for the 2-class (COVID-19-20, Normal-117) problem. Marques et al. [10] proposed efficientNet CNN model obtained an accuracy of 99.62% for binary classification and 96.70% for multiclass classification. Gupta et al. [11] created InstaCovNet-19 model it obtained 99.08% accuracy on 3 class

**Table 1**  
Summary of COVID-19 detection and diagnostic methods.

Reference	Method	Modality	Number of images	Classification
Wang et al. [4]	DL based anomaly detection	Chest X-ray	100-COVID-19, 1431-Non-COVID-19	Binary class
Salman et al. [5]	Inceptionv3	Chest X-ray	130-COVID-19, 130-Normal	Binary class
Abbas et al. [6]	DeTraC	Chest X-ray	105-COVID-19, 80-Normal 11-SARS	Multiclass
Wang et al. [7]	COVID-Net	Chest X-ray	358-COVID-19, 8066-Normal cases 5538-Non-Covid cases	Multiclass
Sethy et al. [8]	CNN+SVM	Chest X-ray	127-COVID-19, 127-Normal 127-Pneumonia	Multiclass
Apostolopoulos et al. [9]	VGG19	Chest X-ray	224-COVID-19, 504-Normal, 700 Pneumonia	Binary class Multiclass
Marques et al. [10]	EfficientNet	Chest X-ray	504-COVID-19, 500-Normal, 504 Pneumonia	Binary class Multiclass
Gupta et al. [11]	InstaCovNet-19	Chest X-ray	361-COVID-19, 1341-Normal, 1345-Pneumonia	Binary class Multiclass
Mahmud et al. [12]	COVXNet	Chest X-ray	305-COVID-19, 1583-Normal, 1493-Viral pneumonia 2780-Bacterial pneumonia	Binary class Multiclass
Ozturk et al. [13]	DarkCovidNet	Chest X-ray	150-COVID-19, 500-Normal, 500-Pneumonia	Binary class Multiclass
Pereira et al. [14]	classification	Chest X-ray	90-COVID-19, 1000-Normal, 55-Pneumonia	Multiclass
Khan et al. [15]	CoroNet	Chest X-ray	284-COVID-19, 310-Normal, 327- Viral pneumonia 330 -Bacterial pneumonia	Multiclass
Ismael et al. [16]	ResNet50+SVM	Chest X-ray	180-COVID-19, 200-Normal	Binary class
Minaee et al. [17]	ResNet50	Chest X-ray	184-COVID-19, 5000-Normal	Binary class
Gour et al. [18]	Stacked	Chest X-ray	546-COVID-19, 1139-Normal 1355 Pneumonia	Multiclass
Bargshady et al. [19]	Inception Transfer Learning	Chest X-ray	5427-COVID-19, 2628-Normal	Binary class
Al-Monsur et al. [20]	Covid-EnsembleNet	Chest X-ray	1133-COVID-19, 2615-Normal 1565 Pneumonia	Multiclass
Xu et al. [21]	Location attention classification	Chest CT	219-COVID-19, 175-Healthy	Multiclass
Amyar et al. [22]	Multitask DL model	Chest CT	449-COVID-19, 425-Normal 495-Other infections	Multiclass
Wang et al. [23]	FCGNet	Chest CT	320-COVID-19, 320-Healthy	Binary class
Pathak et al. [24]	Deep Transfer Learning (ResNet32)	Chest CT	413-COVID-19, 439-Normal or pneumonia	Binary class
Ardakani et al. [25]	ResNet101	Chest CT	510-COVID-19, 510-Non-COVID-19	Binary class
Sozes et al. [26]	ResNet50	Chest CT	120-COVID-19, 150-Normal	Binary class

problems and 99.53% accuracy on 2 class problems. Mahmud et al. [12] proposed stacked multiresolution COVXNet model that generated accuracies of 97.4% and 90.3% for 2-class and 4-class problems, respectively. In another study, the DarkCovidNet model proposed by Ozturk et al. [13] gave accuracies of 98.08% and 87.02%, respectively, for 2-class and 3-class COVID-19 detection

problems. Pereira et al. [14] proposed two classification approaches, namely, multiclass classification and hierarchical classification achieved an F1-Score of 0.65 and 0.89 for multiclass and hierarchical classification, respectively. Furthermore, Khan et al. [15] established a pretrained CoroNet model achieved an overall accuracy of 89.6%. Another study where Ismael and Sengur [16] proposed a deep learning model using different pretrained deep CNN models and SVM classifiers achieved an accuracy score of 94.7% (ResNet50+SVM). Similarly, A transfer learning-based approach has been advised by Minaee et al. [17] obtained sensitivity rates of 100%, 100%, 100%, and 98% for ResNet18, ResNet50, SqueezeNet, and DenseNet121, respectively. Gour and Jain [18] developed a stacked ensemble CNN model that obtained a 97.27% accuracy for 3 classes and a 98.30% accuracy for a binary class. Bargshady et al. [19] used the Inception V3 transfer learning model and achieved an accuracy of 94.2%. Al-Monsur et al. [20] proposed Covid-EnsembleNet detect COVID-19 and secured an accuracy of 97.56%. Loey et al. [28] used the AlexNet, ResNet18, and GoogleNet for COVID-19 detection and obtained accuracy of 66.67%, 69.47% and 80.56%.

Likewise, the research works on the detection of COVID-19 using CT images achieve the accuracy rate as follows: Xu et al. [21] built location-attention classification model yielded an accuracy of 86.7%. Amyar et al. [22] proposed a multitask deep learning model that attained an accuracy of 94.67%. Wang et al. [23] proposed the FGCNet model for COVID-19 classification, with an accuracy of 97.71%. A transfer learning-based model suggested by Pathak et al. [24] obtained an accuracy of 96.22%. Ardakani et al. [25] deployed ten pretrained CNN models (AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101 and Xception) for COVID-19 diagnosis. In addition, Gozes et al. [26] utilized the ResNet-50 architecture attained 94% sensitivity and a 98% specificity score. Moreover, unlike the above-mentioned detection methods using X-ray and CT images, some researchers have tried to diagnose the disease using health data collected during pandemics. For instance, Patrizi et al. [29] designed a behavioural and cost-based health data acquisition model that collects data from wearable devices.

### 3. Preliminary techniques

This section elaborates the detailed aspects of discrete wavelet transforms, convolutional neural networks, transfer learning, stacking techniques and some of the pretrained convolutional neural networks along with their highlight points. These concepts were required to develop the proposed WavStaCovNet-19 model.

#### 3.1. Discrete wavelet transform

Wavelet transformation is a form of transformation domain processing. This technique manipulates the orthogonal transform of the image instead of the original image. The wavelet transform decomposes images using functions that are located in the real and Fourier space. Thus, it manages to capture both the location and frequency information. A typical wavelet transform can be expressed by Eqs. (1) and (2).

$$W_{(s,\tau)} = \int_{+\infty}^{-\infty} x(t)\psi_{(s,\tau)}^*(t) \quad (1)$$

$$\psi_{(s,\tau)}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \quad (2)$$

$s$  and  $\tau$  are the scale and translation parameters, respectively.  $W$  is the wavelet transform, and  $\psi$  is the mother wavelet.

A discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled, not the wavelets that are continuously sampled. This results in a sparse representation of natural data, allowing significant compression. Mathematically, the chosen DWT mother wavelet can be represented in Eq. (3).

$$\psi_{(s,\tau)}(t) = \frac{1}{\sqrt{2^n}}\psi\left(\frac{t-k\times 2^n}{2^n}\right) \quad (3)$$

where the scale and transform parameters have been expressed as powers of 2.

2-D DWT utilizes the combination of low-pass filtering  $\varphi(t)$  and high-pass filtering  $\psi(t)$  to extract various wavelet coefficients by downsampling the image. Each level of DWT decomposition obtains a set of four subimages. These subimages are obtained depending upon one approximation coefficient and three detailed coefficients (horizontal, vertical and diagonal). The approximation coefficient is generated by the application of low-pass filtering (LPF) on the rows and subsequently the columns. The horizontal detail coefficient is generated by low-pass filtering on the rows and then high-pass filtering (HPF) on the columns. Similarly, the vertical detail coefficient is generated by high-pass filtering on the rows and then low-pass filtering on the columns. Last, a diagonal detail coefficient is created by high-pass filtering on the rows and then the columns. This coefficient generation is concisely illustrated in Fig. 1. Additionally, a sample single-level decomposition is given in Fig. 2.

Based on the desired level of decomposition, the approximation coefficient can be further decomposed to generate the next level of coefficients. The procedure continues until the desired level of transformation is achieved.

The transformation procedure necessitates the selection of the most suitable base wavelet functions. Preceding research works have recognized such wavelet functions for 1-D signals. However, such results cannot be reliably extrapolated to 2-D images. Thus, some of the most popular wavelet functions were utilized, including Daubechies, Haar, Coiflet and Discrete Meyer. Two-level DWT decomposition is utilized for better approximation, which engenders denoised images.

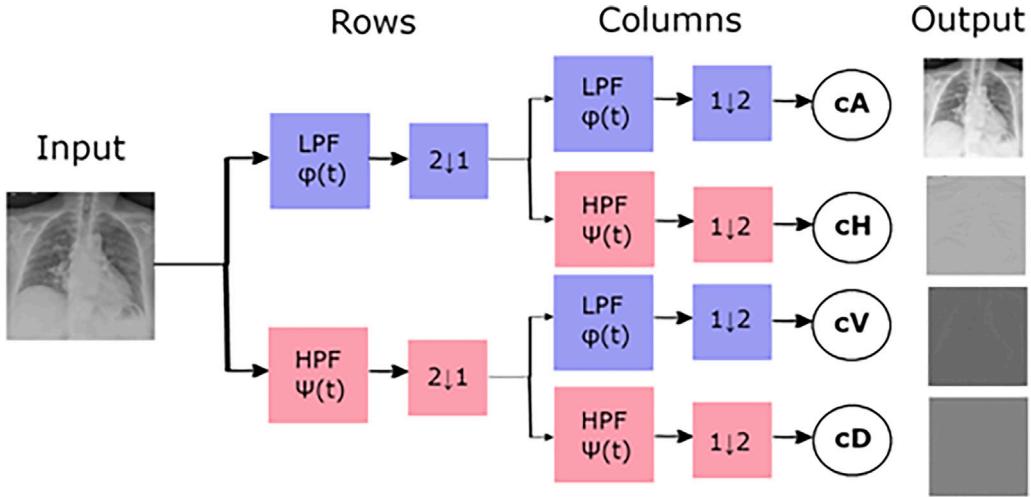


Fig. 1. DWT single level decomposition.

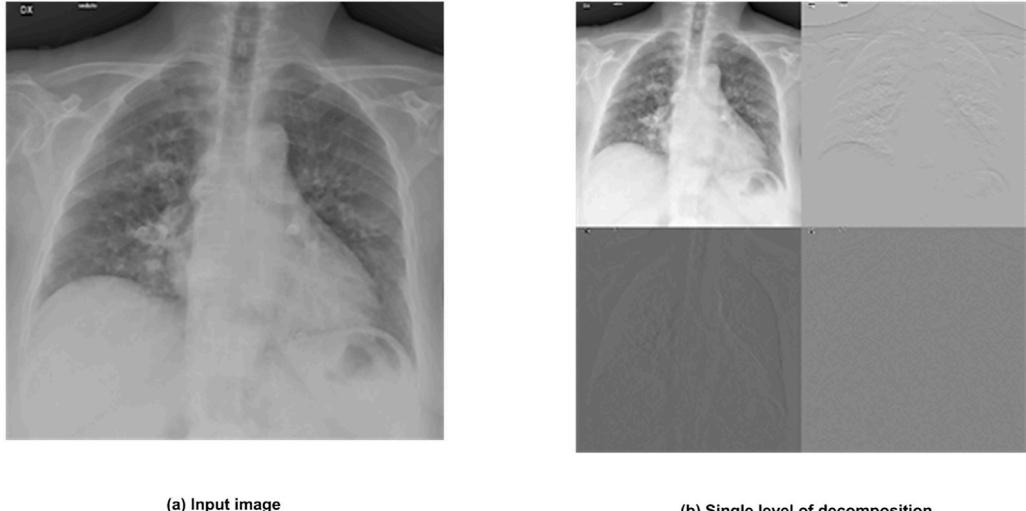


Fig. 2. Sample of DWT single level decomposition.

### 3.2. Convolutional neural network

A convolutional neural network (CNN) is a prolific machine learning technique that has been tremendously effective in image and video applications. A typical CNN is composed of multiple convolutional layers stacked together. The convolutional layer is the pith of the CNN, which in turn is composed of multiple image kernels. The objective of each kernel is to learn and extract meaningful features from the images. These kernels share weights among themselves, which makes a CNN more efficient for image analysis compared to other networks such as feedforward neural networks.

The following layers are the most essential components of several state-of-the-art CNNs:

**(A) Convolutional Layer:** The convolutional layer is the most essential part of the network. The first operation carried out by this layer is the convolution operation. The convolution operation used in the layer is given in Eq. (4).

$$S_{ij} = (I \times K)_{ij} = \sum_{a=\lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=\lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a, j-b} K_{\frac{m}{2}+a, \frac{n}{2}+b} \quad (4)$$

Here,  $i$  and  $j$  are image dimensions, and  $m$  and  $n$  are filter dimensions.

After the convolution, the resultant output is processed using an activation function. The predominant objective of the activation function is to introduce nonlinearity into the input. The ReLU activation function is known to be the most prolific activation function; however, other activation functions, such as tanh and sigmoid, are also widely used.

- (B) Pooling Layer:** The pooling layer is used for dimensionality reduction of the images. These layers summarize image features in fixed patches for downsampling. Each convolution layer in general is followed by a pooling layer. Max pooling and average pooling are the most widespread pooling methods, which take the maximum and average pixel values of a neighbourhood, respectively.
- (C) Batch Normalization Layer:** Although the batch normalization layer is not a desideratum like the pooling layer, it has been gaining attention due to the improved performance that several research activities have reported. It is used to standardize the outputs of a convolution layer and then pass it on to the next convolution layer. This consequently stabilizes the learning process and improves the convergence rate.
- (D) Fully Connected Layer:** The fully connected layer (dense layer) is the core of the typical feedforward network; however, it is merely used in CNNs as the final group of layers for the generalization of features and the final output. In this layer, each neuron is connected to every neuron of the next fully connected layer. For classification applications, the dense layer is generally followed by a sigmoid (for 2 classes) or a softmax function (for > 2 classes) as the output function.
- (E) Dropout layer:** Deep neural networks that are fed with large datasets have the propensity for overfitting on the dataset instead of generalizing for the intended application. This reduces the testing accuracy while maintaining a high training accuracy, belying the practical unavailability of the network. To tackle this issue, regularization techniques such as dropout are used. The dropout layer randomly 'drops out' neurons of the preceding layer during training. This prevents hidden neurons from coadapting, and effectively simulates training of a large number of slightly different networks. This layer is generally utilized on dense layers.

### 3.3. Transfer learning

Transfer learning is a machine learning technique that aims at improving learning efficiency. It involves leveraging the pretrained existing work of one application as the starting point of another application. The weights of the pretrained network can be utilized in the following ways:

- (A) Direct application:** This is the simplest form of transfer learning. The pretrained network is directly used to obtain the results on the data. This technique does not involve any training.
- (B) Feature extraction:** In this technique, the pretrained network is exploited to process the data and extract features. This is done by truncating the output layer of the network and extracting outputs from the last few layers (generally dense layers). These obtained outputs can then be utilized as features and eventually fed to other classifiers.
- (C) Fine-tuning:** This technique, as the name suggests, involves fine-tuning the pretrained network for the specified application. This fine-tuned network can be directly used for the output, or even can be subsumed as part of a larger framework. This is the most prolific technique under transfer learning.

Transfer learning becomes essential when there is an exiguity of available data. The pretrained networks generally manage to converge over diminutive datasets more efficiently than the untrained networks.

### 3.4. Stacking technique

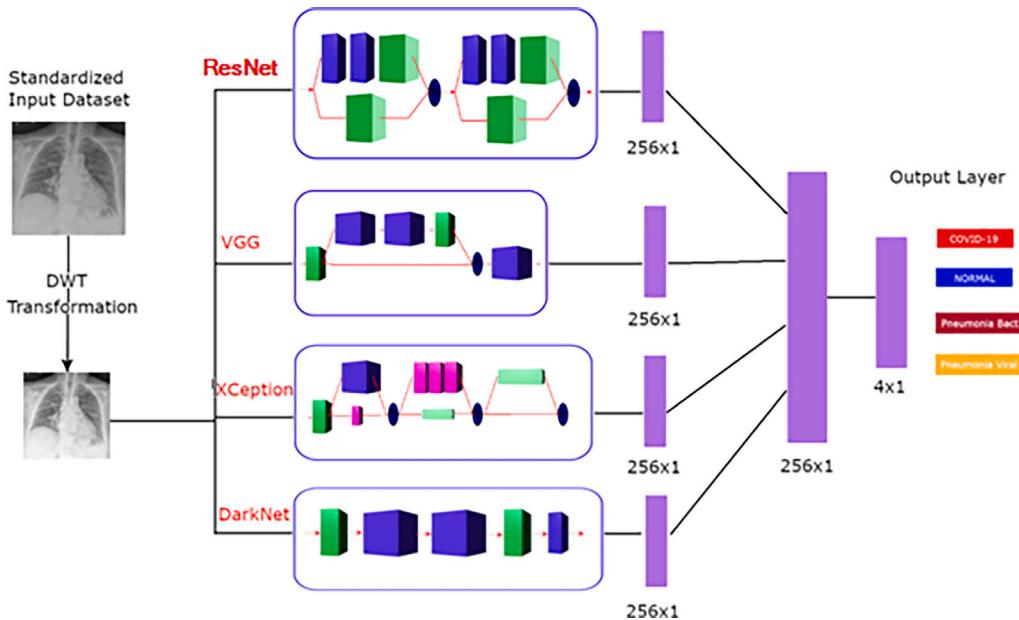
Stacking is an ensemble technique used on neural networks that forms a cohesive framework from individual neural networks. The individual predictions given by each subnetwork are combined into one matrix and then fed to the ultimate neural network, called the estimate learner. Based on the configuration of the technique, the different networks can contribute to the final prediction, either equally, or based on weights, depending on the individual performances.

The subnetworks can either be trained individually, or the entire framework can be trained as a single network, although the former is generally preferable due to the higher level of personalized fine-tuning for each network. The advantage of this ensemble technique can be attributed to the fact that its performance is at least as good as its individual networks. This ensures a zero-risk and a better guarantee of improvement in accuracy.

## 4. Proposed method

The proposed method (WavStaCovNet-19) is a deep convolutional network architecture that fine-tunes the pretrained CNNs to fit on the DWT feature maps of the X-ray images. The utilization of various DWT mother wavelets and transfer learning, together, help in tackling the inadequacy issues of the available COVID-19 datasets.

The primary objective of the proposed method is to discern the difference between four states, three states and two states of health using thoracic X-ray images and eventually identify the correct state. The architecture of WavStaCovNet-19 is outlined in Fig. 3.



**Fig. 3.** WavStaCovNet-19 architecture.

#### 4.1. Preprocessing

Before training the networks, the chest X-ray images are preprocessed by DWT. As mentioned in Section 3.1, surplus DWT images are created for COVID-19 due to the diminutive presence of COVID-19 class images. This is obtained by employing the prominent wavelet coefficients. For the other classes, since there is a surplus of images, only the Daubechies function is utilized. The sample result of the 2-level DWT approximation is shown in Fig. 4. These generated transformations are passed as input to the stacked network. The networks are trained individually over the generated dataset so that each network can capture its own set of features, which will be used later by the main estimate learner. The WavStaCovNet-19 framework is explained in detail in Section 4.2.

#### 4.2. WavStaCovNet-19 model in detail

The proposed WavStaCovNet-19 model is a cohesive system of 4 neural networks, 3 of which are already pretrained over ImageNet, namely, ResNet-50, VGG-19 and Xception. The last network, DarkNet-19, is the simplest among all the networks used; it is created from scratch and is fine-tuned over the dataset. Each of these neural networks has a prolific history of success in object detection applications. They are shortlisted only after rigorous testing on the generated dataset. The four networks, with their disparate architectures, offer unique feature extraction properties leading to the final network classification.

**(A) ResNet50:** ResNet stands for Residual Networks, which has proven to be useful in practical computer vision applications. Winning with the ImageNet 2015 challenge, ResNet-152 is considered a breakthrough technology because it was the first successful neural network with a depth of 150+ layers. Prior to the introduction of ResNet, training of extremely deep networks proved to be a tribulation because of the prevalent problem of vanishing gradients. ResNet approached the issue by introducing the concept of skip connections.

The integrated network employs the ResNet-50 version, which is widely used as the starting point for transfer learning. Keras applications contain a ResNet-50 model pretrained over the ImageNet dataset. This model was extracted and trained over the generated dataset. The prediction layer of the network ( $1000 \times 1$ ) is replaced with two dense layers, the prefinal dense layer is  $256 \times 1$ , while the final layer is dependent on the level of classification. For 3-class classification, layer  $3 \times 1$  is used, while for 4-class classification, layer  $4 \times 1$  is used. The prefinal layer is used as the input for the estimate learner.

**(B) VGG19:** VGG is a CNN created by Oxford's Visual Geometry Group in 2014, aiming to improve upon the performance of its predecessors, such as AlexNet. It has shown tremendous success in computer vision, and its utilization is still extant. The VGG-19 variant used by the integrated network is 19 layers deep. A pretrained VGG-19 network from Keras applications was imported and fine-tuned over the dataset. The adjustments made are similar to those made for ResNet-50. The prediction layer ( $1000 \times 1$ ) is replaced with similar dense layers. After the training phase is completed, the output of the prefinal layer ( $256 \times 1$ ) is fed to the estimate learner.

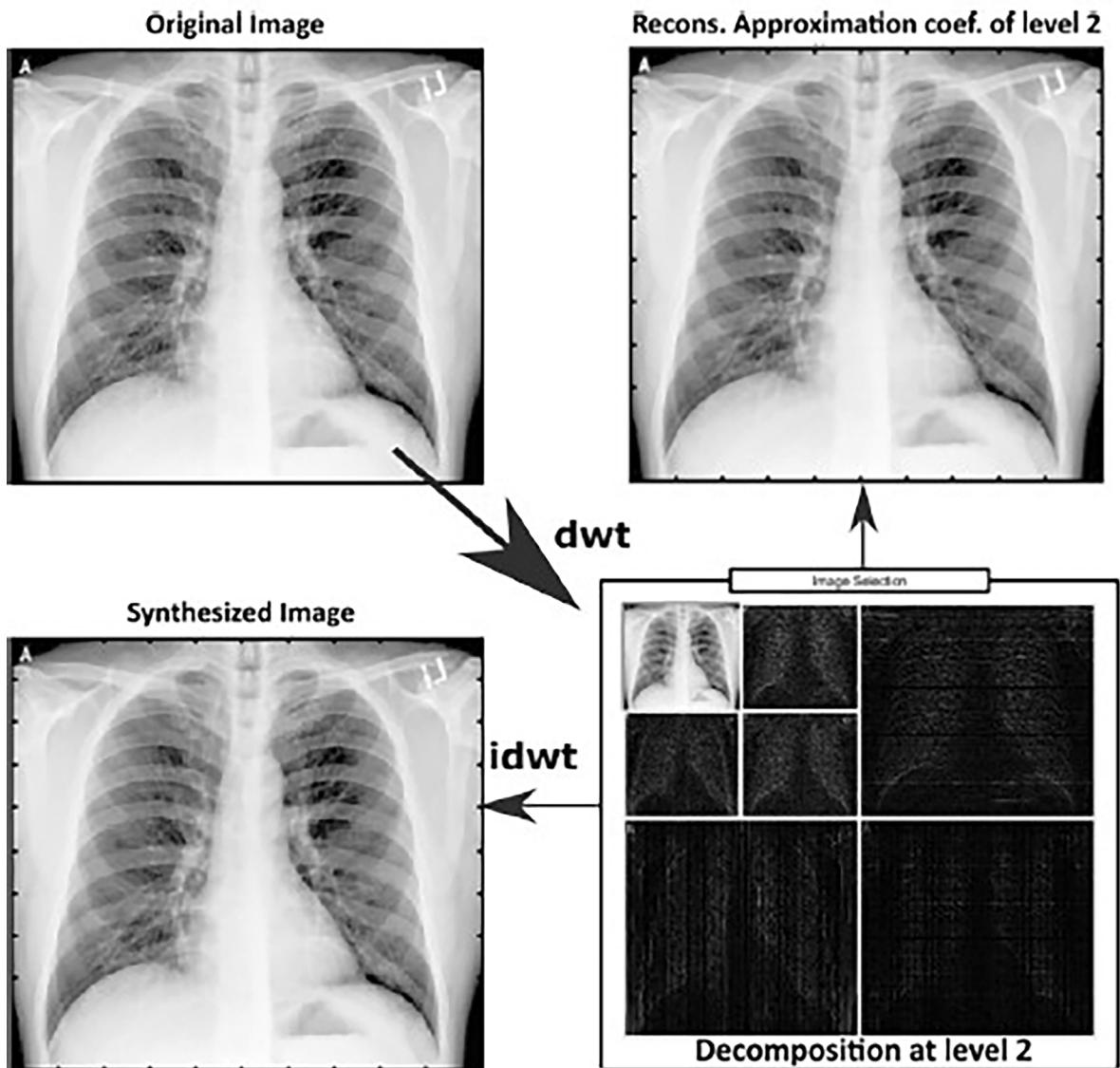


Fig. 4. The sample result of 2-level dwt approximation.

**(C) Xception:** Xception network, created by Google, stands for Extreme version of Inception. The network is 71 layers deep, making use of modified depthwise separable convolution. This modification has helped Xception outperform its predecessor, Inception-v3, despite being of equal complexity. A pretrained Xception model is also available on Keras applications. This model is imported and fine-tuned over the dataset and then subsumed by the integrated network. Unlike ReNet-50 and VGG-19, the default input image size accepted by Xception is  $299 \times 299 \times 3$ . However, the TensorFlow networks can adjust with nondefault image sizes by padding. The prediction layer, as mentioned for ResNet-50 and VGG-19, is replaced with two dense layers. The feature vector is obtained from the prefinal layer ( $256 \times 1$ ).

**(D) DarkNet19:** The DarkNet network is a CNN that forms the backbone of the YOLO architecture. The network, like VGG19, contains 19 convolutional layers. The disparate nature of the generated dataset from the ImageNet dataset incentivized the training of one of the four networks from scratch so that the fine-tuning could be more personalized for the dataset. DarkNet19, being the simplest network, is chosen for such a task. The network is created from scratch on the TensorFlow 2.3.0 library and trained over the dataset. The DarkNet network created for this framework has a prefinal layer of  $256 \times 1$ . After the training phase is completed, this prefinal layer is utilized as the feature vector for the estimate learner.

The training of each of the networks involves commonalities in the choice of activation function, optimizer, loss function and output function.

**Table 2**

The training parameters for experimental setup.

Training parameters	ResNet50	VGG19	Xception	DarkNet19	WavStaCovNet-19
Activation function	Leaky ReLU				
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning Rate	1e-5	1e-5	1e-5	1e-5	1e-5
Batch Size	20	20	20	20	20
Loss function	Crossentropy	Crossentropy	Crossentropy	Crossentropy	Crossentropy
Epochs	100	100	100	100	100

The activation function used in this framework is the leaky ReLU. This function has an advantage over the vanilla ReLU, as it does not face the dying-neurons issue because it does not completely cut off the negative inputs. The formula of the leaky ReLU is as follows in Eq. (5):

$$f(x) = \begin{cases} 0.15x & \text{for } x < 0 \\ x & \text{for } x > 0 \end{cases} \quad (5)$$

For the optimization in the training process, the Adam optimizer is used. Adam triumphs over other optimizer choices, such as AdaGrad and RMSProp, in factors such as memory requirement, computational efficiency and adaptability. Eqs. (6) and (7) represent the formulas for Adam optimization:

$$n_t = b_1 \times n_{t-1} + (1 - b_1) \times g_t \quad (6)$$

$$s_t = b_2 \times s_{t-1} + (1 - b_2) \times g_t^2 \quad (7)$$

where  $n_t$  and  $s_t$  are the first and second moments,  $g_t$  is the gradient, and  $b_1$  and  $b_2$  are the hyperparameters.

As mentioned in the Introduction (Section 1), a customized loss function was developed to curb the presence of false negatives in the COVID-19 class. Sparse categorical cross-entropy is the base loss for the function, and an additional regularization parameter is added. This parameter punished the network more severely in the presence of COVID-19 false negatives. Eq. (8) represents the loss function formula.

$$L(t, p) = SCCE(t, p) + |(1 - p(covid))| \quad (8)$$

where  $L$  is the loss function, and  $t$  and  $p$  are the true classes and predicted class values by the framework, respectively, for each image.  $SCCE$  is the sparse categorical cross-entropy function.

For classification applications, functions such as sigmoid (2 classes) and softmax (multiple classes) are widely used. In this application, the softmax function is utilized owing to the multiclass classification. The formula of the softmax function is given in Eq. (9):

$$S(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{100} e^{z_k}} \quad \text{for } j = 1, 2, 3 \quad (9)$$

The feature vectors obtained from the individual networks are passed on to the estimate learner, which in turn trains over these vectors. The estimate learner is composed of two dense layers. The prefinal dense layer is  $256 \times 1$ , while the final layer, whose configuration is dependent on the number of classes involved in classification, would have  $3 \times 1$  and  $4 \times 1$  layers for 3-class and 4-class classification, respectively. These layers are trained over the feature vectors obtained from the subnetworks.

The training of the estimate learner also shares commonalities with that of the individual networks. It utilizes the leaky ReLU as the activation function, the Adam optimizer for network learning, a customized loss function and the softmax output function, as mentioned in Eqs. (5)–(9). The final layer of the estimate learner attempts to predict the correct class. The complete procedure of WavStaCovNet19 is given in Algorithm 1.

#### 4.3. Parameter setup and notations

For training and testing using ResNet50, VGG19, Xception, DarkNet19 and concatenation of ResNet50, VGG19, 17 Xception and DarkNet19 networks (WavStaCovNet-19) based on section s42. The training parameters are given in Table 2. In addition, the notation used in the proposed WavStaCovNet-19 method are tabulated in Table 3.

#### 4.4. Computational complexity of the proposed method

The significant steps in the proposed WavStaCovNet-19 method are preprocessing, separate model generation, stacked model and classification. The sub-tasks in preprocessing are

- Resize input in  $O(1)$  time
- DWT runs in  $O(n)$  time

The overall time is  $O(n)$  for preprocessing.

**Algorithm 1** : WavStaCovNet-19 4-class Classification

---

**Input** X-ray Image Dataset (Variable size)  $x = [\text{Normal}, \text{Bacterial}, \text{Viral} \text{ and } \text{COVID-19}]$

**Output** Probability Distribution for Diseases  $y = [P(y\text{NORMAL}), P(y\text{BACTERIAL}), P(y\text{VIRAL}), P(y\text{COVID-19})]$

- 1: We selected 1300 images for each of the normal, bacterial pneumonia, viral pneumonia and COVID-19 classes.
- 2: procedure preprocessing( $x$ )
- 3:  $x \leftarrow \text{resize}(x)$  //Standardize the images to a standard size and format ( $224 \times 224 \times 3$  and PNG respectively).
- 4:  $[\text{LL}, \text{LH}, \text{HL}, \text{HH}] \leftarrow \text{DWT2}(x)$  //Generate Daubechies DWT transformation for each image.
- 5: Retrieve the 1300 approximation coefficients (LL). //Generate COVID-19 image approximation coefficients, using multiple wavelet functions (Daubechies, Haar, Coiflet, etc).
- 6: end procedure preprocessing
- 7: procedure separate\_model\_generation (LL)
- 8: //Neural network training configuration details:
  - //Initializer: Random normal distribution
  - //Activation function: Leaky ReLU
  - //Optimizer: Adam
  - //Loss function: Customized as given in Equation (8)
  - //Output function: Softmax
- 9: Train ResNet50, VGG19, Xception, and DarkNet19 and generate models
- 10: Initialisation of parameters Batch size = 20, epochs=100, learning rate=1e-5
- 11: for  $i=1$  to epochs do
- 12:   Train(ResNet50, training\_image)
- 13:   Feature extraction from training\_image
- 14:   model1=save(ResNet50)
- 15: end for
- 16: for  $i=1$  to epochs do
- 17:   Train(VGG19, training\_image)
- 18:   Feature extraction from training\_image
- 19:   model2=save(VGG19)
- 20: end for
- 21: for  $i=1$  to epochs do
- 22:   Train(Xception, training\_image)
- 23:   Feature extraction from training\_image
- 24:   model3=save(Xception)
- 25: end for
- 26: for  $i=1$  to epochs do
- 27:   Train(DarkNet19, training\_image)
- 28:   Feature extraction from training\_image
- 29:   model4=save(DarkNet19)
- 30: end for
- 31: procedure separate\_model\_generation
- 32: procedure Stacked\_model
- 33: for  $m=1$  to validation\_images do
- 34:   for  $n=1$  to total\_no.\_models do
- 35:      $[\text{SM1}_m, \text{SM2}_m, \text{SM3}_m, \text{SM4}_m] = \text{model}(n).\text{predict}(\text{validation\_images}(m))$
- 36:   //SM-Separate Model
- 37:   end for
- 38:    $F = \text{concatenate}([[\text{SM1}_m, \text{SM2}_m, \text{SM3}_m, \text{SM4}_m]])$
- 39: end for
- 40: Training of softmax classifier on  $F$
- 41: Stacked\_model = Train( $F$ , val)
- 42: Classification of normal, bacterial, viral and COVID-19 images
- 43: Classification\_output = classify(Stacked\_model, test\_images)
- 44: //Feed the testing images and retrieve the output from the estimate learner final layer:  $\text{Dense}_2(4 \times 1) \rightarrow \text{Probability distribution} \rightarrow y$
- 45: end procedure Stacked\_model

---

**Table 3**  
Notation used throughout this paper.

Parameter	Description
$W$	Wavelet transform
$\psi$	Mother wavelet
$s$	scale parameter
$\tau$	Translation parameter
$S_{ij}$	Convolution operation $S$ of $i,j$ image
$f(x)$	Leaky ReLU function $f$ of $x$
$L(t, p)$	Loss function $L$ of $t, p$
$t$	True class
$p$	predicted class
$SCCE(.)$	Sparse categorical cross-entropy
$S(.)$	Softmax function

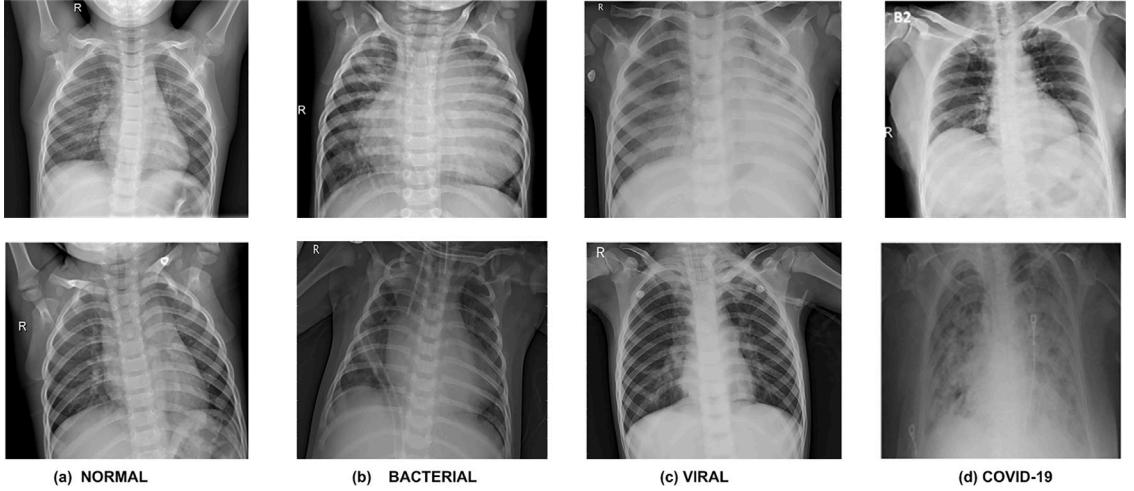


Fig. 5. Chest X-ray images.

For separate model generation, the sub-tasks are

- Convolution operation  $2cwh(M - w + 1)(N - h + 1)k$
- Fully Connected Network  $2nm$

Total operations for each separate model is  $2cwh(M - w + 1)(N - h + 1)k + 2nm$ . Hence time taken for separate model approximately  $O(n^2)$ . From the each separate model, the time taken for stacked model is  $4 \times O(n^2)$ . Finally for classification takes  $O(n^2)$  The overall time complexity is  $O(n) + 4 \times O(n^2) + O(n^2) \equiv O(n^2)$

## 5. Experimental results and analysis

In this section, we discuss the dataset, the evaluation criteria, the results and the performance comparisons. The classification method was implemented using the Python 3.8.3 programming language with TensorFlow 2.3.0 on the Windows 10 platform with an Intel Core i7 8 GB RAM processor and an NVIVIDA GeForce MX150 graphics card.

### 5.1. Dataset

Two datasets were utilized for the experiment, the COVID-19 imagedata collection repository and chest X-ray images for normal and pneumonia [30].

At the time of the experiment, the COVID-19 image repository contained 478 chest X-ray images for the COVID class. The images were split into training and testing subsets in a 60:40 ratio, resulting in 286 images for training and 192 for testing. Owing to the fact that this obtained dataset size is diminutive by itself for neural network training, surplus COVID-19 DWT transformations were generated to rectify the deficiency. Further details of the utilized DWT transformations are given in Section 3.1.

The chest X-ray dataset had 1341 normal, 2530 bacterial pneumonia and 1345 viral pneumonia images in the training folder. For unbiased training, 1300 images were used in each class. The testing folder had 234 normal, 242 bacterial pneumonia and 148 viral pneumonia images, and all these images were deployed for testing. The sample chest X-ray images are shown in Fig. 5.

The image distribution of the final dataset is given concisely in Table 4.

**Table 4**

Number of images for each class in datasets used.

Dataset/Classes	COVID-19	Normal	Bacterial	Viral
Number of Images in Training Set	286	1341	2530	1345
Number of Images in Training Set Used	1300 (Augmented)	1300	1300	1300
Number of Images in Testing Set	192	234	242	148
Number of Images in Testing Set Used	192	234	242	148

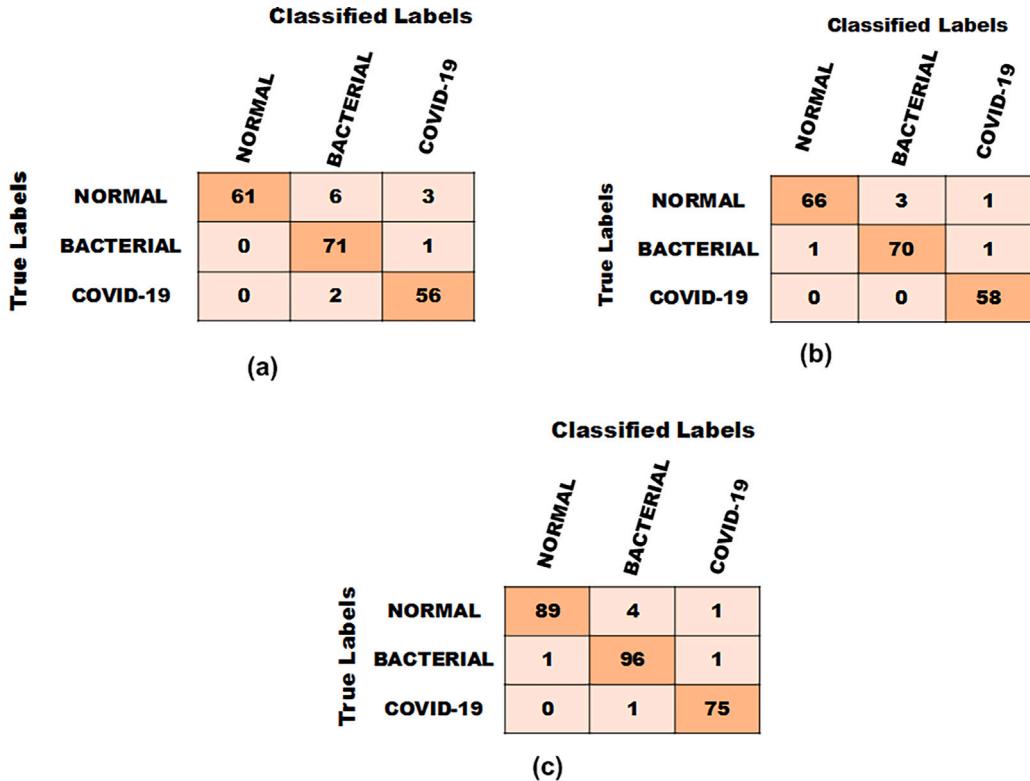


Fig. 6. WavStaCovNet-19's 3-class classification (a) Fold 1 (b) Fold 2 (c) Fold 3.

## 5.2. Evaluation criteria

The proposed WavStaCovNet-19 framework's performance was measured across 3 folds using confusion matrices for both 3-class and 4-class classification. A confusion matrix records the true sample classes along with the predicted sample classes by the classification method. These records help in calculating the true positives (TP), false-positives (FP), false negatives (FN) and true negatives (TN) for each class. Confusion matrix-based metrics were utilized to record and analyse the performance parameters, namely, accuracy, precision and recall, the equations for which are given in Eqs. (10)–(12).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (10)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (11)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (12)$$

## 5.3. WavStaCovNet-19 method performance

As mentioned in Section 5.2, the confusion matrices have been recorded for WavStaCovNet-19's performance evaluation. The confusion matrices for all 3 folds of both the 3-class and 4-class classification are represented in Figs. 6 and 7.

The confusion matrices for 3-class classification by WavStaCovNet-19 shown in Fig. 6 indicate excellent performance for the COVID-19 images. Furthermore, no true COVID-19 images were classified as normal, indicating the complete absence of false-positives, which is a desideratum in today's pandemic situation. The classification method proves efficacious for the bacterial pneumonic images with an insignificant number of false-positives and negatives.

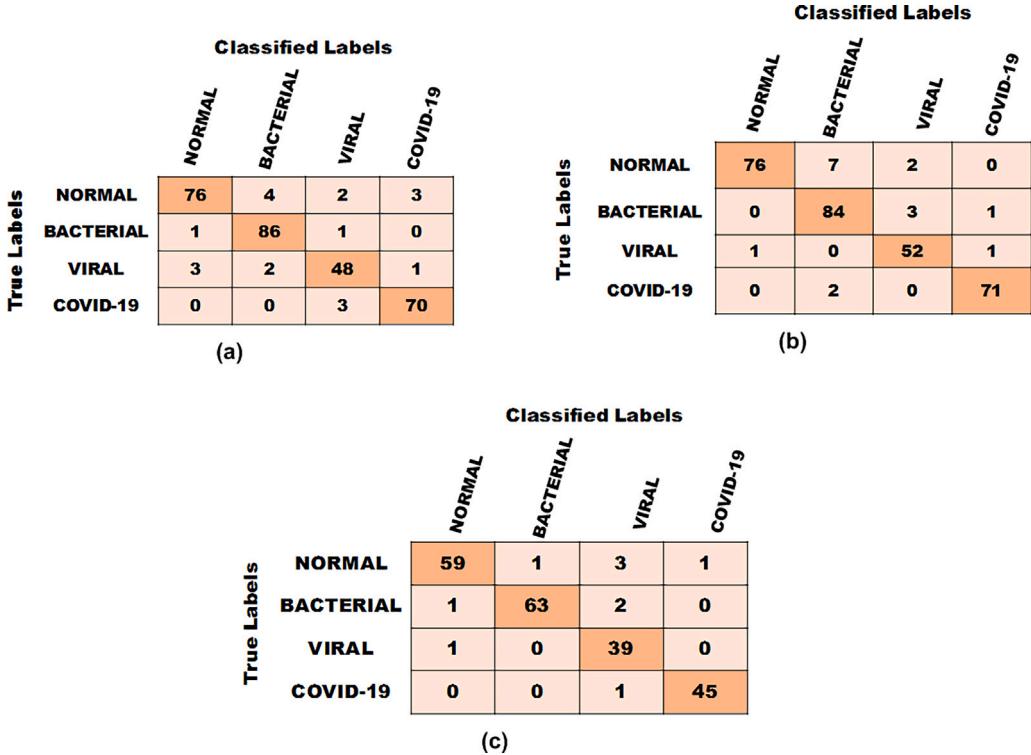


Fig. 7. WavStaCovNet-19's 4-class classification (a) Fold 1 (b) Fold 2 (c) Fold 3.

**Table 5**  
WavStaCovNet-19's performance accuracy.

Classification type	Fold	Accuracy (%)	Total images
3-classes	1	94.00	200
	2	97.00	200
	3	97.74	268
	Overall	96.10	668
4-classes	1	93.33	300
	2	94.33	300
	3	95.37	216
	Overall	94.24	816

**Table 6**  
WavStaCovNet-19's performance precision and recall.

Metric	Classification type	Normal	Bacterial	Viral	COVID-19
Precision	3-classes	99.08%	93.67%	–	95.34%
	4-classes	96.78%	93.57%	89.10%	96.37%
Recall	3-class	92.30%	97.93%	–	98.43%
	4-class	90.17%	96.28%	93.92%	96.87%

For the 4-class classification, WavStaCovNet-19 performs very well for all classes, but the classification effectiveness slightly attenuates for the normal class, resulting in the presence of a modicum of false-positives. However, the presence of false negatives in COVID-19 is again nonexistent. The method also seems to be capable of discerning differences between COVID-19 and other viral diseases.

Based on the confusion matrices, the precision and recall can also be calculated for each class. For such a calculation of each class metric, that specific class is considered positive, and all the rest of the classes are consequently negative. For a more concise performance representation, the accuracy, precision and recall charts are recorded in Tables 5 and 6.

The performance charts in Tables 5 and 6 further reaffirm the inferences drawn from the confusion matrices. The overall accuracy highlights the efficacy of the method for both 3-class and 4-class classifications. The high precision and recall values for COVID-19

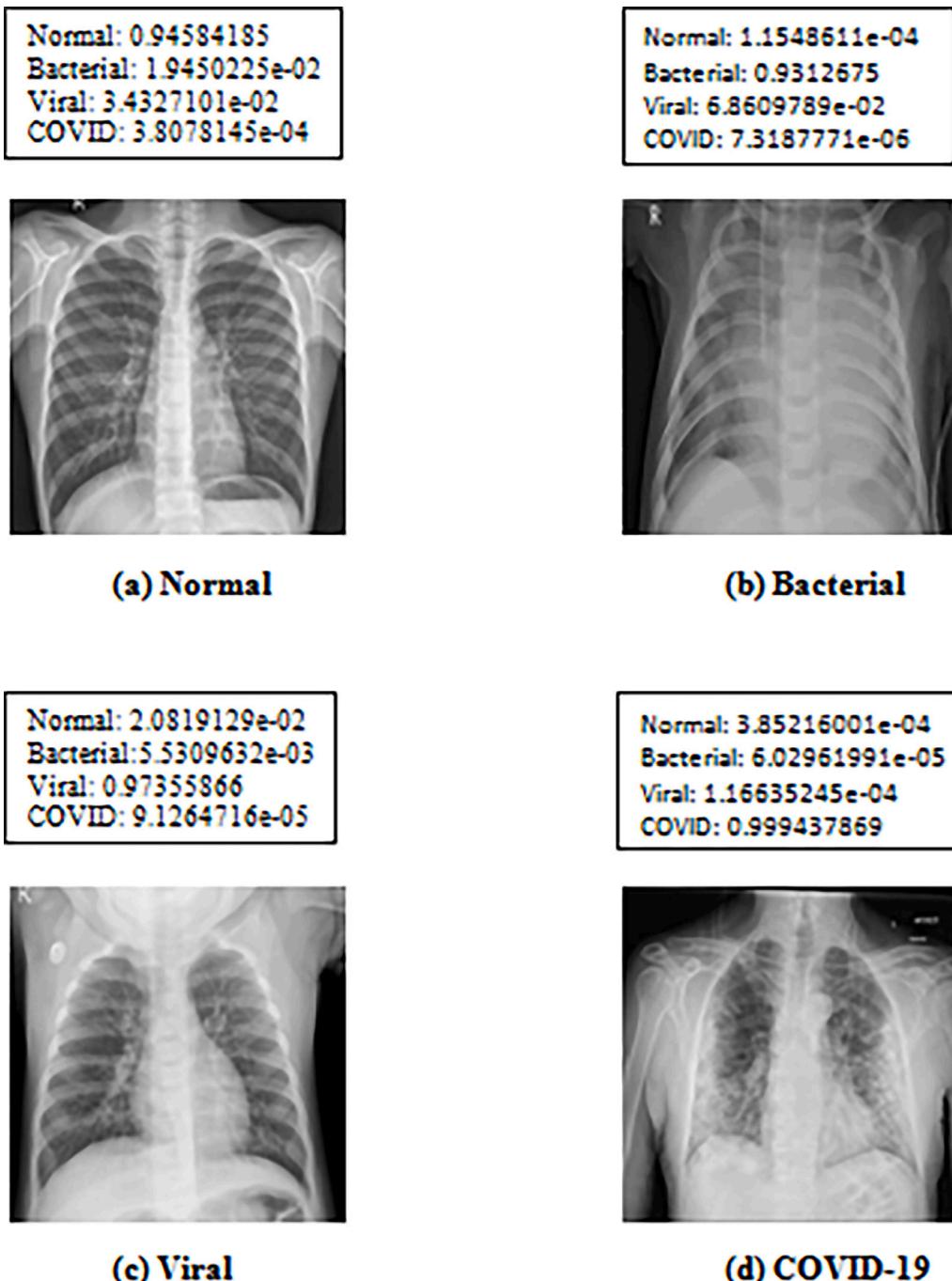


Fig. 8. WavStaCovNet-19's sample classification results (Given in the softmax output).

corroborate the utility of the customized loss function, which was focused on reducing the COVID-19 classification inaccuracy. The relatively lower recall values for the normal class reaffirm the attenuation of the method's performance for that class.

The sample classification results of the proposed method are also illustrated in Fig. 8.

#### 5.4. Performance comparison

To evaluate the efficacy of the stacking method, WavStaCovNet-19 was compared to its individual subnetworks and other existing classification models in the 4-class classification. The comparison with the individual subnetworks determines the utility of the

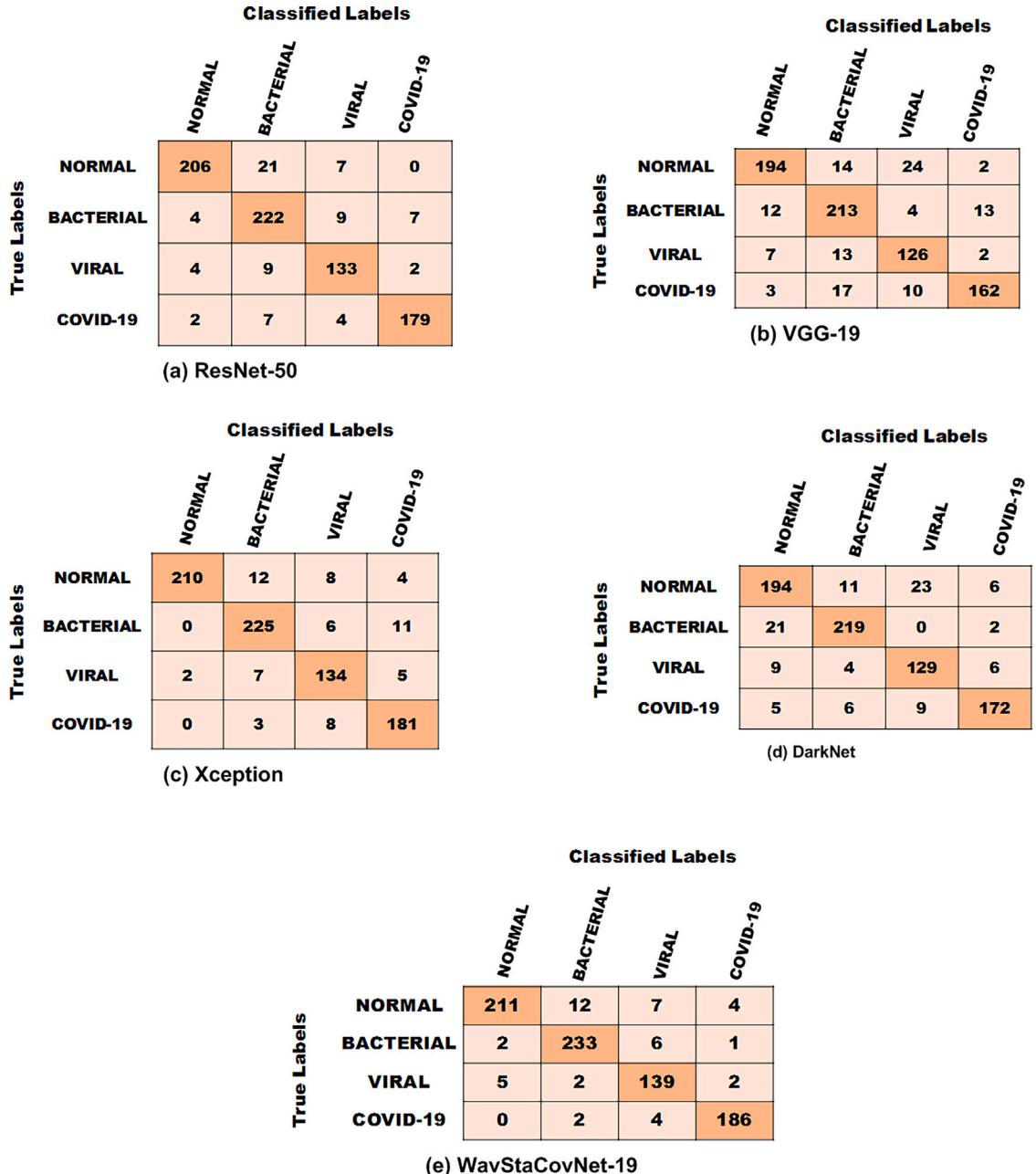


Fig. 9. Individual submodels and WavStaCovNet-19 confusion matrices.

stacking method. Theoretically, the stacking ensemble method is supposed to perform at least as well as its individual methods. The performance of these individual subnetworks along with the overall WavStaConNet-19 have been recorded in Fig. 9 in the form of confusion matrices.

For a more concise comparison, the performance accuracy of the individual networks is also recorded in Table 7. To check the synergy of the DWT transformation with neural networks, the individual subnetworks were also trained on nontransformed images. The COVID-19 dataset was augmented using kerasImageGenerator in place of extra DWT transformations. The performance accuracy of the tested networks on nontransformed images is also recorded in Table 7. Furthermore, the running time of proposed method for classification of testing images in batches of 32 images is shown in Table 8. Besides, the scalability analysis of proposed method is shown in Fig. 10.

Every tested subnetwork employed the same training functions, leaky ReLU activation, Adam Optimizer, customized loss function and softmax output function with the same hyperparameters as in the method employed for comparison.

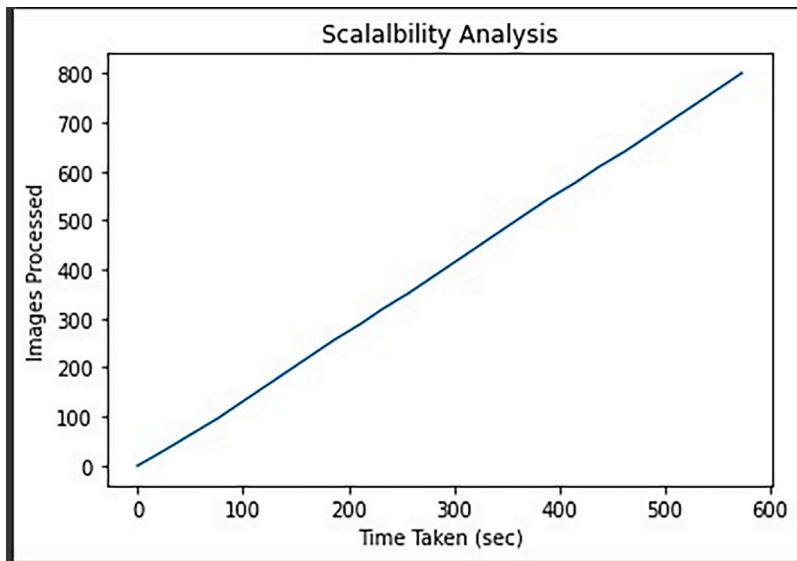


Fig. 10. The scalability analysis of proposed method for test images.

Table 7

Performance accuracy of the methods tested for 4-class classification.

Classification method	Average without DWT	Average with DWT
ResNet-50	87.00%	90.68%
VGG-19	83.12%	85.17%
Xception	84.55%	91.91%
DarkNet	79.17%	87.50%
WavStaCovNet-19	–	94.24%

Table 8

The running time of the proposed method for classification of testing images in batches of 32 images.

Classification method	Time taken with DWT (in Seconds)
ResNet-50	115.8
VGG-19	103.2
Xception	107.3
DarkNet	13.6
WavStaCovNet-19(Proposed)	557.4

As the confusion matrices suggest, WavStaCovNet-19 performed much better than its individual subnetworks. The more complex networks (ResNet-50 and Xception) performed much better than the simpler networks. Possibly owing to the customized loss function, all the networks performed well for the COVID-19 class, while faltering slightly for the normal class. WavStaCovNet-19 seemed to have benefitted from all the networks because of the significant performance difference from every individual subnetwork.

The performance accuracy recorded in Table 7 highlighted the importance of DWT transformations. Except for ResNet-50, every subnetwork seemed to have a greatly improved performance due to the transformed images.

In addition, the proposed method was compared with more of the existing models. The detailed comparison is shown in Table 9. The 3-class and 4-class accuracy (if present) and the number of cases tested by the existing models are recorded in Table 9.

## 6. Conclusion

This paper proposes a robust WavStaCovNet-19 deep learning model for the detection of COVID-19 patients from chest X-ray images. The proposed model is trained with two publicly available balanced datasets containing 1300 images of each class. The proposed model obtained an accuracy of 94.24% on 4 classes (COVID-19, viral pneumonia, bacterial pneumonia, normal) and achieved an accuracy of 96.10% on 3 classes (COVID-19, pneumonia, normal). In addition, the WavStaCovNet-19 deep learning model can also be applied for the detection of other pneumonia that could affect the human body organs.

**Table 9**

Comparison of accuracy of the WavStaCovNet-19 and other existing COVID-19 detection methods for multiclass classification problem.

Authors	Method used	Modality	Accuracy (%)
Loey et al. [28]	AlexNet	Chest X-ray	66.67
Loey et al. [28]	ResNet18	Chest X-ray	69.47
Sethy et al. [8]	ShuffleNet + SVM	Chest X-ray	70.66
Wang et al. [7]	DenseNet121	Chest CT	80.12
Loey et al. [28]	GoogleNet	Chest X-ray	80.56
Xu et al. [21]	Location attentionclassification	Chest CT	86.70
Ozturk et al. [13]	DarkCovidNet	Chest X-ray	87.02
Khan et al. [15]	CoroNet	Chest X-ray	89.60
Mahmud et al. [12]	COVXNet	Chest X-ray	90.30
Minaee et al. [17]	TransferLearning	Chest X-ray	91.04
Wang et al. [7]	COVID-Net	Chest X-ray	93.30
Apostolopoulos et al. [9]	VGG19	Chest X-ray	93.48
Amyar et al. [22]	Multi-taskDL model	Chest CT	94.67
Proposed	WavStaCovNet-19	Chest X-ray	96.10

## Funding statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## CRediT authorship contribution statement

**Rajkumar Soundrapandian:** Conceptualization, Writing – original draft, Methodology, Software. **Himanshu Naidu:** Conceptualization, Methodology, Investigation, Validation. **Marimuthu Karuppiah:** Visualization, Investigation, Validation, Supervision. **M. Maheswari:** Reviewing & editing. **Ramesh Chandra Poonia:** Review & editing, Validation, Visualization.

## Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.compeleceng.2023.108711>.

## Data availability

The code and data of the proposed AI-based wavelet and stacked deep learning architecture will be made openly accessible upon publication.

<https://github.com/himanshunaidu/wavstacovnet-19>.

## References

- [1] Zheng YY, Ma YT, Zhang JY, Xie X. COVID-19 and the cardiovascular system. *Nat Rev Cardiol* 2020;17(5):259–60.
- [2] Worldometers. Worldometers information. 2023, URL <https://www.worldometers.info/coronavirus/>, [Accessed 09-01-2023].
- [3] Wang W, Xu Y, Gao R, Lu R, Han K, Wu G, et al. Detection of SARS-CoV-2 in different types of clinical specimens. *JAMA* 2020;323(18):1843–4.
- [4] Zhang J, Xie Y, Li Y, Shen C, Xia Y. Covid-19 screening on chest x-ray images using deep learning based anomaly detection. 2020, arXiv preprint [arXiv:2003.12338](https://arxiv.org/abs/2003.12338).
- [5] Salman FM, Abu-Nasser SS, Alajrami E, Abu-Nasser BS, Alashqar BA. Covid-19 detection using artificial intelligence 2020.
- [6] Abbas A, Abdelsamea MM, Gaber MM. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Appl Intell* 2021;51(2):854–64.
- [7] Wang L, Lin ZQ, Wong A. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Sci Rep* 2020;10(1):1–12.
- [8] Sethy PK, Behera SK. Detection of coronavirus disease (covid-19) based on deep features. *Preprints* 2020.
- [9] Apostolopoulos ID, Mpesiana TA. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med* 2020;43(2):635–40.
- [10] Marques G, Agarwal D, de la Torre Díez I. Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. *Appl Soft Comput* 2020;96:106691.
- [11] Gupta A, Gupta S, Katarya R. InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray. *Appl Soft Comput* 2021;99:106859.
- [12] Mahmud T, Rahman MA, Fattah SA. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Comput Biol Med* 2020;122:103869.
- [13] Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med* 2020;121:103792.
- [14] Pereira RM, Bertolini D, Teixeira LO, Silla Jr CN, Costa YM. COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios. *Comput Methods Programs Biomed* 2020;194:105532.
- [15] Khan AI, Shah JL, Bhat MM. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Comput Methods Programs Biomed* 2020;196:105581.

- [16] Ismael AM, Sengur A. Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Syst Appl* 2021;164:114054.
- [17] Minaee S, Kafieh R, Sonka M, Yazdani S, Soufi GJ. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Med Image Anal* 2020;65:101794.
- [18] Gour M, Jain S. Automated COVID-19 detection from X-ray and CT images with stacked ensemble convolutional neural network. *Biocybern Biomed Eng* 2022;42(1):27–41.
- [19] Bargshady G, Zhou X, Barua PD, Gururajan R, Li Y, Acharya UR. Application of CycleGAN and transfer learning techniques for automated detection of COVID-19 using X-ray images. *Pattern Recognit Lett* 2022;153:67–74.
- [20] Al-Monsur A, Kabir MR, Ar-Rafi AM, Nishat MM, Faisal F. Covid-EnsembleNet: an ensemble based approach for detecting Covid-19 by utilising chest X-ray images. In: 2022 IEEE world AI IoT congress. 2022, p. 351–6.
- [21] Xu X, Jiang X, Ma C, Du P, Li X, Lv S, et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering* 2020;6(10):1122–9.
- [22] Amyar A, Modzelewski R, Li H, Ruan S. Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation. *Comput Biol Med* 2020;126:104037.
- [23] Wang SH, Govindaraj VV, Göriz JM, Zhang X, Zhang YD. Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. *Inf Fusion* 2021;67:208–29.
- [24] Pathak Y, Shukla PK, Tiwari A, Stalin S, Singh S. Deep transfer learning based classification model for COVID-19 disease. *Irbm* 2020.
- [25] Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Comput Biol Med* 2020;121:103795.
- [26] Gozes O, Frid-Adar M, Greenspan H, Browning PD, Zhang H, Ji W, et al. Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis. 2020, arXiv preprint arXiv:2003.05037.
- [27] Kassania SH, Kassanib PH, Wesolowskic MJ, Schneidera KA, Detersa R. Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning based approach. *Biocybern Biomed Eng* 2021;41(3):867–79.
- [28] Loey M, Smarandache F, M. Khalifa NE. Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning. *Symmetry* 2020;12(4):651.
- [29] Patrizi N, Tsipropoulou EE, Papavassiliou S. Health data acquisition from wearable devices during a pandemic: A techno-economics approach. In: ICC 2021-IEEE international conference on communications. 2021, pp. 1–6.
- [30] ChestXrayImages. Chest X-ray images (pneumonia). 2022, URL <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>, [Accessed 26-12-2022].

**Rajkumar Soundrapandiyan** is currently working as an Associate Professor in the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India. He has completed a Ph.D. at the Vellore Institute of Technology, Vellore, India (2017). His research interests include computer vision, visual perception, object detection, medical image processing, infrared image processing, biometrics, information hiding, and network security.

**Himanshu Naidu** is a software engineer at ServiceNow, Hyderabad with a passion for software development, problem solving and machine learning. Earning his bachelor's degree at Vellore Institute of Technology, he graduated with distinction. Adept at full-stack development and system design, he has over 4 years of professional experience as a software developer.

**Marimuthu Karuppiah** received the Ph.D. degree in computer science and engineering from VIT University, Vellore, India, in 2015. . He is currently a professor with the Department of Computing Science and Engineering, Presidency University, Bengaluru, India. His current research interests include cryptography and wireless network security, in particular, authentication and encryption schemes.

**M. Maheswari** received the Ph.D. degree in Computer Science and Engineering from Anna University, India, in December 2019. She is currently an Associate Professor in the School of Computer Science and Engineering at the Sathyabama Institute of Science and Technology in Chennai. Her research interests include machine learning, recommender systems, computer networks, and data mining.

**Ramesh Chandra Poonia** received the Ph.D. degree in computer science from Banasthali University, Banasthali, India, in July 2013. He is currently an Associate Professor at the Department of Computer Science, CHRIST (Deemed to be University), Bengaluru, India. His research interests include sustainable technologies, cyber-physical systems, and intelligent algorithms for autonomous systems.