



**Explainable Machine Learning Techniques in Medical Image Analysis  
Based on Classification with Feature Extraction**

**Dr. B. Dwarakanath**

*Assistant Professor, Department of Information Technology, SRM Institute of Science and Technology,  
Ramapuram, Chennai, India  
dwarakab@srmist.edu.in*

**Dr. Gitanjali Shrivastava**

*Assistant Professor, Department of Law, Symbiosis Law School, Pune, India  
dr.gitanjali10@gmail.com*

**Dr. Rohit Bansal**

*Associate Professor, Department of Management Studies, Vaish College of Engineering, Rohtak,  
Haryana, India  
rohitbansal.mba@gmail.com*

**Praful Nandankar**

*Assistant Professor, Department of Electrical Engineering, Government College of Engineering,  
Nagpur, India  
pppful@gmail.com*

**Dr. Veera Talukdar**

*Registrar, Kaziranga University, Jorhat, Assam, India  
bhaskarveera95@gmail.com*

**M Ahmer Usmani**

*Assistant Professor, Department of Computer Science and Engineering, Bharati Vidyapeeth Deemed  
to be University, Department of Engineering and Technology, Navi Mumbai, India  
mausmani@bvucoep.edu.in*

<b>Article History</b>	<b>Abstract</b>
Received: 13 July 2022 Revised: 20 September 2022 Accepted: 26 October 2022	Animals are also afflicted by COVID-19, a virus that is quickly spreading and infects both humans and animals. This fatal viral disease has an impact on people's daily lives, health, and economy of a nation. Most effective machine learning method is deep learning, which offers insightful analysis for examining a significant number of chest x-ray pictures that have a significant bearing on COVID-19 screening. This research proposes novel technique in lung image analysis for detection of lung infection due to COVID using Explainable Machine learning techniques. Here the input has been collected as COVID patient's lung image dataset and it has been processed for noise removal and smoothening. This processed image features have been extracted using spatio transfer neural network integrated with DenseNet+ architecture. Extracted features has been classified using stacked auto Boltzmann encoder machine with VGG-19Net+. With the transfer learning method integrated into the binary classification process, the suggested algorithm achieves good classification accuracy. The experimental analysis has been carried out for various COVID dataset in terms of accuracy, precision, Recall, F-1score, RMSE, MAP. The proposed technique attained accuracy of 95%, precision of 91%, recall of 85%, F_1 score of 80%, RMSE of 61% and MAP of 51%.

<p>CC License CC-BY-NC-SA 4.</p>	<p><b>Keywords:</b> <i>lung image analysis, lung infection, Explainable Machine learning, classification, COVID -19, feature extraction.</i></p>
--------------------------------------	--

## 1. Introduction

Covid-19 is a serious medical problem where many individuals pass away every day. This virus sickness not only impacts a particular nation, but also causes suffering throughout the entire planet. Several different viruses (including SARS, MERS, the flu, etc.) [1] entered the scene in the last ten years, although they only last a few days or weeks. Due to the availability of vaccinations created by these experts, few of these viruses are diagnosed even though many scientists are working on them. The Covid-19 sickness is currently affecting everyone in the world [2], and the most crucial issue is that no one country's experts have been able to develop a vaccine for it. Plasma therapy, X-ray imaging, and many more forecasts came into play in the meantime, but no precise cure for this fatal illness has been discovered. Covid-19 [3] is a disease that claims lives of individuals every day, and it is exceedingly expensive to diagnose for a nation, a state, and a patient. Online repositories like Github and Kaggle had X-ray photos of healthy persons and Covid-19-infected patients [4] for study as of March 2020. A pandemic epidemic sickness called COVID-19 has begun to endanger people on a global scale. It is crucial to distinguish between people who have covid-19 infection and healthy patients. To lower the risk of patients who have not been affected by Covid-19, dialysis of infected patients with Covid-19 requires additional vigilance and must be treated under very strict procedures. The coronavirus family is a large group of enclosed, positive-sense, single-stranded RNA viruses that infect both birds as well as mammals. These viruses can occasionally infect people, leading to mild to major respiratory illnesses. But unlike SARS and MERS, COVID-19 symptoms might show much more quickly or in a milder form, allowing the illness to spread via asymptomatic people and contributing to present pandemic. Although WHO has stressed necessity of widespread testing as well as contact tracing to more effectively combat epidemic, not all nations have necessary laboratory equipment as well as reagents [5]. The research community has studied CT and X-ray imagery as alternatives to time-consuming RT-PCR testing since onset of the COVID-19 pandemics. In fact, early studies using COVID-19 images found lung lesions in patients who weren't in a life-threatening condition and even in those who had recovered. The classification of the chest X-ray/CT scan pictures was used to automate the process of COVID-19 detection utilizing ML and DL both of which are branches of artificial intelligence (AI). According to a review of the literature, there are more DL-based models than ML-based models that can solve this particular classification challenge. The majority of these investigations revealed high classification performance in terms of accuracy, recall, precision, and F1-measure. However, majority of these classification models were developed and evaluated using much smaller datasets, typically just two [6].

Contribution of this research is as follows:

1. To propose novel method in lung image analysis for detection of lung infection due to COVID using Explainable Machine learning techniques.
2. the input has been collected as COVID patient's lung image dataset and it has been processed for noise removal and smoothing.
3. This processed image features have been extracted using spatio transfer neural network integrated with DenseNet+ architecture.
4. Extracted features has been classified using stacked auto Boltzmann encoder machine with VGG-19Net+

## 2. Related works

Deep learning techniques have recently received a lot of attention because they have the potential to significantly improve the way that many healthcare-related problems are handled. By utilising radiology datasets and data science classifiers, data scientists have been concentrating on improving the detection, analysis, and subsequent prediction of various diseases. For quick detection of COVID-19 situations, the author in [7] presented the algorithm nCOVnet based on the data leaking principle.

Although the authors' experiments yielded a detection accuracy of 88%, they did not provide a clear visualisation of the cases of COVID-19 they had discovered on the CXR pictures. DeQueueNet is a model that has been suggested by work in [8] for classifying X-ray pictures of patients into positive and negative categories while simultaneously identifying the COVID-19. For categorising COVID-19 on CT volumes, 3D CNN have also been utilized in recent publications. For COVID-19 classification, these 3D CNNs enable spatiotemporal modelling of CT volumes. A 3D CNN known as C3D was created by Work [9]. DeCoVNet was also suggested by author [10] for 3D volume modelling. Three 3D ResNet, ResNet-18, ResNet-50, and ResNet-101, were proposed in work [11]. Additionally, 3D-SqueezeNet, 3D-SuffleNet, 3D-MobileNet-V1, and 3D-MobileNet-V2 have been introduced by authors [12]. The 3D CNN deep learning models were used by the authors [13] to find COVID-19 on 3D CT scans. As inputs to 3D CNN model, approach [14] segments individual CT pictures as well as a set of CT images. The photos are subjected to 3D convolution by the model, and the results are then fed into AlexNet and ResNet methods for final COVID-19 classification. Similar to this, the author in [15] used 3D CNN models to learn COVID-19 from 3D CT scans. Procedure convolves CT volumes with 3D filters, and then applies bag technique to merge the 3D convolution outputs. After that, fully connected layers are used to derive final COVID-19 forecasts. 3D ResNet-18 was used in work [16] to identify COVID-19 on 3D CT scans. This study demonstrated how COVID-19 and common pneumonia can be distinguished using the 3D-ResNet-18 model. A deep model for COVID19 identification (COVID-Net) was proposed by author [17], and it achieved 92.4% accuracy when identifying normal, non-COVID pneumonia, and COVID-19 classes. Deep learning method was created in Work [18] using 224 validated COVID-19 pictures. For two and three classes, their model had success rates of 98.75% and 93.48%, respectively. Using chest X-ray pictures in conjunction with the ResNet50 model, author [19] attained a 98% COVID-19 detection accuracy. Work [20] used X-ray pictures as a classifier to categorise the features collected from several convolutional neural network (CNN) models.

### 3. System Model

This section discusses novel technique in lung image analysis for detection of lung infection due to COVID using Explainable Machine learning techniques. Here the input has been collected as COVID patient's lung image dataset and it has been processed for noise removal and smoothening. This processed image features have been extracted using spatio transfer neural network integrated with DenseNet+ architecture. Extracted features has been classified using stacked auto Boltzmann encoder machine with VGG-19Net+. The proposed architecture is shown in figure-1.

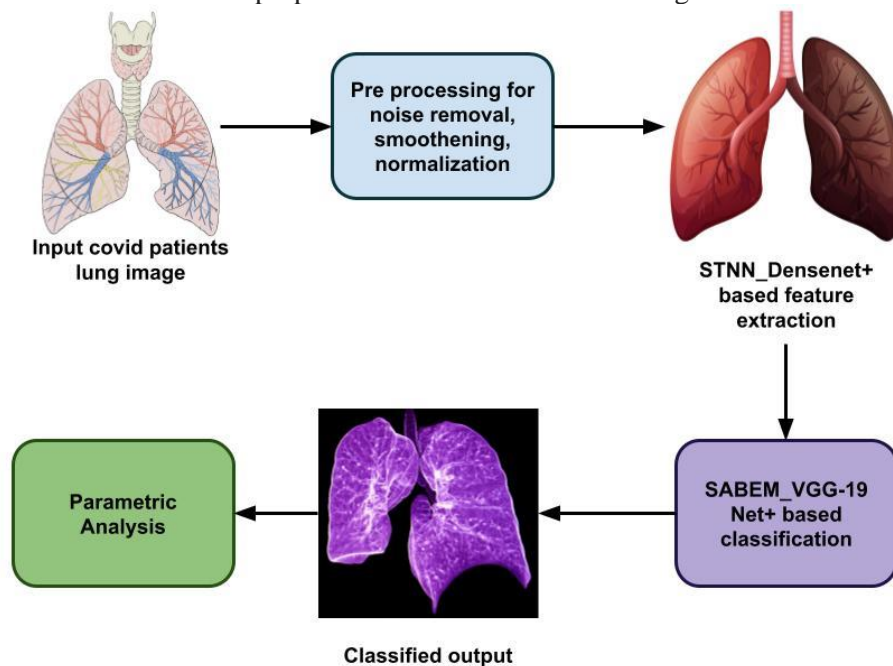


Figure 1. Overall proposed architecture

The various datasets' resolutions range from  $1112 \times 624$  to  $2170 \times 1953$  pixels. In the combined dataset that was released, these were preprocessed and scaled down to a lower resolution of  $299 \times 299$  pixels. The Portable Network Graphics (PNG) format is used for all of the photos. Each of the four categories has a different frequency of presence in terms of the number of photographs for each of the aforementioned categories. With a total of 10,192 photos, or around 48% of the sample, the Normal category had the highest representation. The COVID-19 image count, on the other hand, is 3616, or around 17% of the overall dataset. The number of lung opacity images is 6012, or about 28% of the overall dataset. With a total of 1345 photos, or 6% of the dataset, the final category (Viral Pneumonia) has the lowest representation.

### 3.1 Spatio transfer neural network integrated with densenet+ architecture based feature extraction

To make the interest point detector resilient to variations in illumination, we apply a contrast stretching function in the spatio-temporal domain. A sigmoid function given in Eqn. (1) is applied to normalised video as 3-D data to stretch intensity level of image and create stronger corners in space-time.

$$f_c(I(x, y, t)) = \frac{1}{1 + \exp(-\gamma(I(x, y, t) - c))} \quad (1)$$

$$A(x, y, t) = g(*; \sigma_i^2, \tau_i^2) \cdot \begin{pmatrix} L_{cx}^2 & L_{cx}L_{cy} & L_{cx}L_{ct} \\ L_{cx}L_{cy} & L_{cy}^2 & L_{cy}L_{ct} \\ L_{cx}L_{ct} & L_{cy}L_{ct} & L_{ct}^2 \end{pmatrix} \quad (2)$$

where  $I_e(x, y, t) = f_e(I(x, y, t))$  and  $L_{ex} = \partial_x(I_e)$ . Corner function is found as Eqn. (3):

$$R(x, y, t) = \det A(x, y, t) - k(\text{trace } A(x, y, t))^3 \quad (3)$$

The implicit activation rule for the deep network would be to learn a function, F, that in any example translates certain random combinations of x d and x d to yd and uniform distribution, in a pixel-wise manner. RLU activation of deep networks, which coincidentally contains identical activations, reduces oscillation that may ensue. Eqn. (4) gives the definition of the selectivity-based function, F.

$$F(X^i) = \begin{cases} \hat{y}d, & \text{if } i = d \\ 0, & \text{if } i = \tilde{d} \end{cases} \quad (4)$$

$$G(X^{-i}) = \begin{cases} X_d^{(s)}, & \text{if } i = d \\ 0, & \text{if } i = \tilde{d} \end{cases} \quad (5)$$

Using a robot manipulator as an example, arm's movement comprises both rotation around the centroid and translation of the centre of mass. The dynamic equation is Eqn. (6) when the manipulator is examined using the Newton-Euler equation.

$$\tau = M(\theta)\theta + V(\theta, \dot{\theta}) + G(\theta) \quad (6)$$

As a result, according to control theory, a differential state is required for robot control. By up-sampling and inverse filtering decomposed coefficients, wavelet synthesis can use the reconstruction of multi-signal wavelet decomposition. Eqn. (7) gives that as a result:

$$A_{j,n}^k X = M^{-1/2} \sum_k h(k - Mn) A_{j+1}^k X + M^{-1/2} \sum_k^{(i)} (k - Mn) D_{j+1}^{(i)} X \quad (7)$$

Equation (7) states that using the approximation and detail components at level  $j + 1$ , approximation component at level  $j$  is rebuilt. Time scales of all the time series are same because they were all decomposed at same time. Since size of the decomposed components is equal to that of the original data, exploration techniques like histograms, power spectra, and principle components can be directly used to determine the traits and geographic significance of these spatiotemporal fluctuations at each scale. Eqn. (8) can be used to represent the mapping of multi-signal time series  $X(t)$  to integer approximation in range of  $(m, n)$ :

$$f: \mathbb{R}_{R(X(x))} \mapsto \mathbb{R}_{(m,n)}^+ \quad (8)$$

$$x'_i = \left\{ m + (n - m) \frac{x_i - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)} \right\} \quad (9)$$

where  $(x)$  is a function that seeks to locate the nearest integer to the actual value of  $x$ . Equation (9) can be used to project entire dataset from original value range  $(a, b)$  into new value range  $(m, n)$  with minimal changes to the data's spatial and temporal distribution for each deconstructed feature series cube. Equations (10) display the temporal memory state equations for forgetting gate, input gate, and input differentiation in STNN unit.

$$\begin{pmatrix} f_t \\ i_t \\ k_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} (W \cdot [x_t, h_{t-1}^t])$$

$$\begin{pmatrix} f_t^{t'} \\ i_t^t \\ k_t^{t'} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} (W \cdot [x_t, S_r^{t-1}]) \quad (10)$$

Equations (11) display the spatial memory equations for the forgetting gate, input gate, and differentiation in the STNN unit.

$$\begin{pmatrix} f_t^{t'} \\ i_t^t \\ k_t^{t'} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \tanh \end{pmatrix} (W \cdot [x_t, S_r^{t-1}]) \quad (11)$$

$$\begin{pmatrix} d_t \\ p_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \tanh \end{pmatrix} (W \cdot [x_t - x_{t-1}, h_{t-1}^t])$$

When  $l = 1, s_t^{l-1} = S_t^l, s_{t-1}^{l-1} = s_{t-2}^l, S_1^t$

Eqn. (12) provides the revised temporal cell state and spatial cell state.

$$\begin{aligned} C_t^t &= f_t \circ C_{t-1}^t + i_t \circ k_t + d_t \circ p_t \\ O_t &= \sigma(w_0 \cdot [h_{t-1}^t, c_t^t, s_t^l, x_t] + b_0) \end{aligned} \quad (12)$$

Eqn. (13) provides the equation for the output gate in the STNN unit.

$$h_t^t = O_t \circ \tan(C_t^l, S_t^l) \quad (13)$$

Deep neural networks can learn extremely complex relationships through transfer learning, which can be reduced through the use of regularisation and dropout [18]. If the training dataset is indeed too tiny, it is obviously possible to use the augmentation techniques of the training dataset. It involves applying minor mathematical adjustments to the geometric shape to produce different versions of the same image. This helps the neural networks' ability to anticipate the future by preventing overfitting.

Traditional network methods include 1 layers, with lth layer's output, z<sub>l</sub>, and every layer implementing the non-linear transformation H<sub>l</sub>(.). Method used in this paper extends the concept of dense connection to the volumetric image categorization task. Eqn. (14) is used to define l specifically:

$$z_l = H_l([z_0, z_1, z_2, \dots, z_{l-1}]) \quad (14)$$

Where [...] is the concatenation function and z<sub>0</sub>, z<sub>1</sub>, ..., z<sub>l</sub> are 3D feature volumes created in earlier layers. Figure 1 depicts a dense unit's structure (top). The three primary operations of the function H<sub>l</sub>(.) are a batch normalisation (BN) layer to reduce the internal covariate transform. A dense unit, shown in Figure 1 (bottom), consists of one layer in a dense block, with each layer in block connected to all subsequent levels. When layers are connected closely together, features are used more effectively and each layer's feature expansion is slower than it would be with typical CNNs. The models are therefore smaller and contain fewer parameters than other networks.

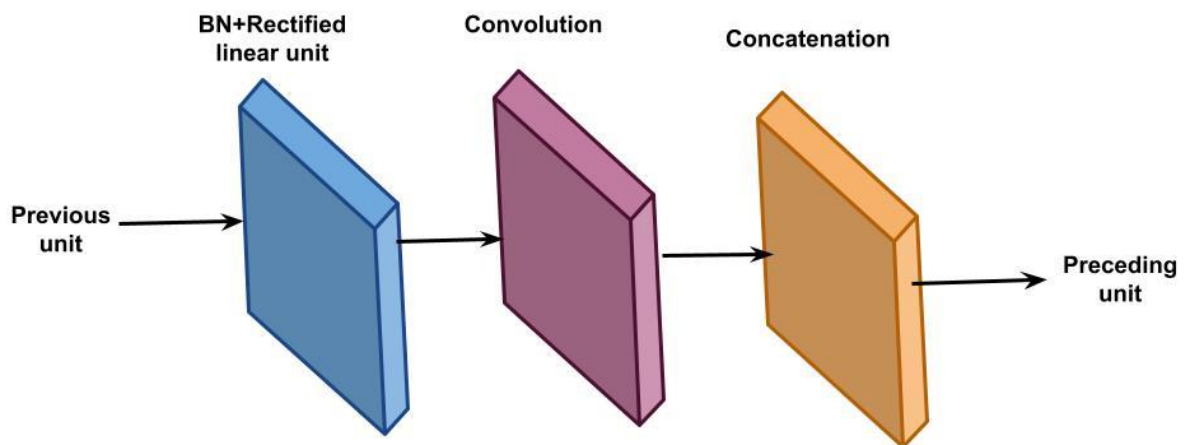


Figure 2. Architecture of a Dense Unit

To create separate networks with distinctive compositions and the ability to extract various properties, numerous experiments were carried out with various hyper-parameter sets. The growth rate was one practical hyperparameter that was shown to improve the network's outputs. l th layer has  $g_0 + g(l)$

input volume-features if each function HI creates  $g$  volume-features. Compact layers, such as those in the DenseNet with  $g = 12$ , are possible. Suggested solution involves using a probability-based fusion ensemble method [26], which combines probabilistic results of final classification layer from many separate networks.

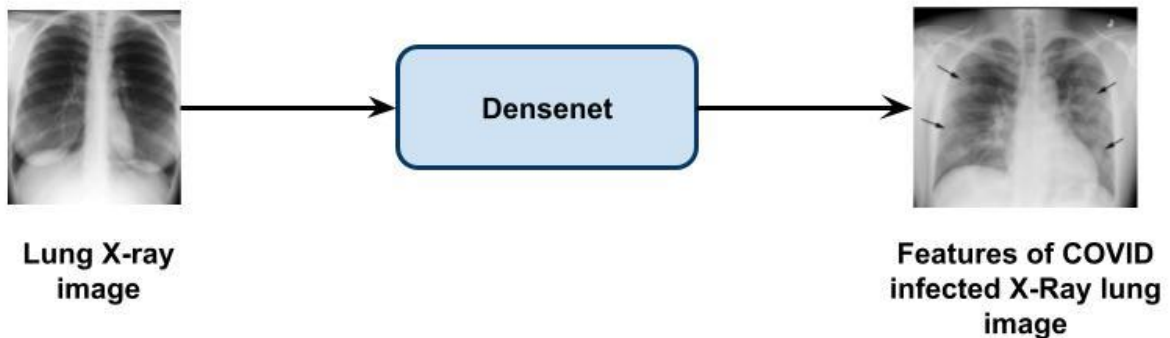


Figure 3. Architecture of proposed DenseNet framework for feature extraction

The network automatically learns the weight of each feature channel, boosts the valuable characteristics in accordance with the weight, and suppresses the features that are unhelpful for the task at hand. The convolutional layer's channel-wise feature recalibration is explicitly modelled in architecture.  $k$ th feature map of the  $g$ th layer is stored as  $X_{g,k}$ , and squeeze process is evaluated using Eqn. (15): This is accomplished by employing global average pooling.

$$X'_{g,k} = F_{sq}(X_{g,k}) = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H X_{g,k}(i, j) \quad (15)$$

where  $g \in \{1, 2, \dots, N-1\}$  and  $k \in \{1, 2, \dots, C\}$ . The next step is an excitation operation that creates weight for each channel using two fully connected (FC) layers. We designate the input of excitation operation in  $g$ th layer as  $(X'_{g,1}, X'_{g,2}, \dots, X'_{g,C})$ . The results of the excitation operation can then be written using Eqn in the  $g$ th layer. (16)

$$(X''_{g,1}, X''_{g,2}, \dots, X''_{g,C}) = F_{ex}(X'_{g,1}, X'_{g,2}, \dots, X'_{g,C}) = \sigma(W_2 \delta(W_1)) \quad (16)$$

where  $\widetilde{X}_{g,k}$  denotes the weight of the  $g$ th layer's  $k$ th channel. ReLU and Sigmoid functions are represented by and respectively. By rescaling  $X_{g,k}$  with weight  $X''_{g,k}$ , which is derived using Eqn(17), the final result is produced

$$\widetilde{X}_{g,k} = F_{Re}(\cdot) = X_{g,k} X''_{g,k}. \quad (17)$$

#### Stacked Auto Boltzmann Encoder Machine With VGG-19Net+ Based Classification:

Chest X-ray pictures are utilised directly for encoding and decoding in Fig. 1b. Encoder portion of the layer 1 network is used to extract features, and the decoder portion restores the input's original feature map. In contrast to Fig. 1b, LRN and Max-Pooling operate the input feature map of Fig. 1 first. The network of Autoencoder 2 of entire stacked autoencoder diagnosis method is shown in Figure 1c.

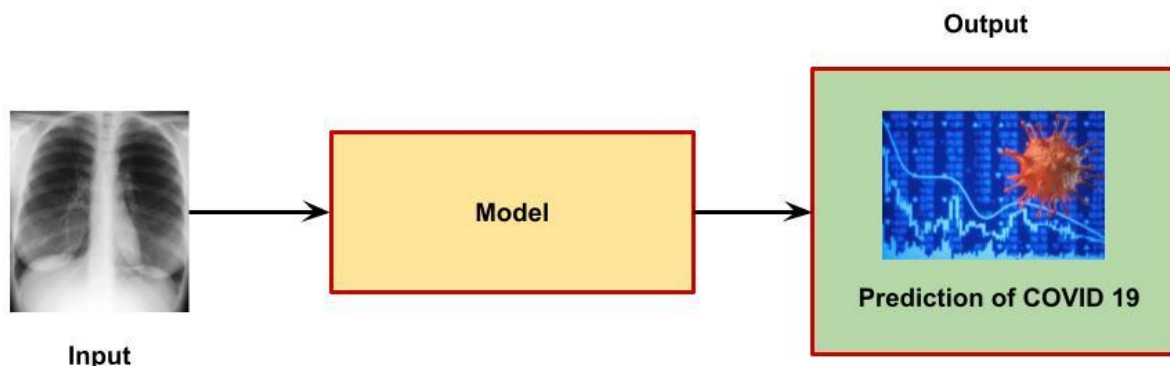


Figure 4. Stacked autoencoder model structure, overall architecture of stacked autoencoder detection model, b stacked autoencoder layer 1 structure, c stacked autoencoder layer 2 structure

The layer 2 network is the same as the layer 3 and layer 4 networks. It is significant to notice that first four layers of autoencoder networks have various loss functions. Reconstruction loss is loss of layer 1 autoencoder network, as illustrated in Eq. (18):

$$J_1 = \frac{1}{N} \sum_{i=1}^N L(f_{\text{deroding } 1}(\mathbf{w}, \mathbf{x}), \mathbf{x}) \quad (18)$$

In order to as closely mimic the real data as possible, the mean squared error loss function term was used for model's first four layers. After iteration training, the loss function's value decreases steadily until it is at its lowest point. According to Eq. (19), the layer 2 loss function is based on layer 2 reconstruction loss as well as J1 loss function of layer 1 autoencoder network as regularisation term.

$$J_3 = J_2 + \frac{1}{N} \sum_{i=1}^N L(f_{\text{dinandang}}, h2) \quad (19)$$

$$h1 = f_{\text{encoding}}(w, x) \quad (20)$$

Layers 3 and 4's loss functions are comparable to layer 2's. Reconstruction loss of layer 4 plus loss functions of first 3 layers of autoencoder network are what make up layer 4's loss function, as indicated in Eq (21).

$$J_3 = J_2 + \frac{1}{N} \sum_{i=1}^N L(f_{\text{decanding } 3}, h2)$$

$$J_4 = J_3 + \frac{1}{N} \sum_{i=1}^N L(f_{\text{decandings } 4}, h3) \quad (21)$$

where h2 and h3 are results of layer 2 f encoding and layer 3 f encoding, respectively. Additionally, as indicated in Eq. (22): "We add the loss function of each of the four layers autoencoder networks as the regular term when calculating the classification loss".

$$J_{\text{elacipioasion}} = \sum_{i=1}^4 J_i + \frac{1}{N} \quad (22)$$

Where w stands for the last layer's parameter matrix, y is CT image's label, h5 is layer 5's output, and L 0 is the loss function term L crossentropy, as stated in Eq. (23):

$$-\mathbf{y} \log(f_{\text{eluruffination}}(\mathbf{w}', \mathbf{h5})) \quad (23)$$

Cross-entropy loss was used as the loss function term for model's last layer in order to calculate the probabilities of COVID-19 and non-COVID-19. Boltzmann machine proposed method has one visible layer, L hidden layers, and one output layer. They also have a connection to the output layer. We search for the logarithmic likelihood's maximum.  $L(\theta) = -\sum_{i=1}^t \log p(\bar{v}_i, \underline{u}_i)$ ,  $(\bar{v}_i, \underline{u}_i) \in V$  by Eqn. (24),

$$\sum_{k=1}^{L-1} \sum_{i=k+1}^L \sum_{j=1}^{S_k} \sum_{i=1}^{S_i} \underline{h}_i^{(k)} \bar{w}_{ij}^{(k,h)} \underline{h}_j^{(k)} - \sum_{i=1}^n d_i u_i - \sum_{k=1}^L \sum_{i=1}^{S_k} \sum_{j=1}^n \underline{h}_i^{(k)} \underline{u}_j^{(k)} u_j \quad (24)$$

Where  $V = \{(\bar{v}_i, \underline{u}_i) \mid i = 1, 2, \dots, t\}$  is a set of training pairs.

We use the Contrastive Divergence technique to train the Boltzmann machine for classification. Eqn. (25) defines the conditional distributions required for the negative phase.

$$P(\underline{v} | \underline{v}^{(1)}) = \mathcal{N}(\underline{a}_i + \underline{b}_i + \sigma_i \sum_{j=1}^{S_1} w_{ij} \underline{h}_j^{(1)}, \sigma_i^2 l)$$

$$P(u_j | H1) = \text{sigm} \left( d_j + \sum_{k=1}^L \sum_{i=1}^{S_k} \underline{h}_i^{(k)} \underline{u}_j^{(k)} \right)$$

$$P(\underline{h}_j^{(1)} | \bar{v}, \underline{H}_{-1}, \underline{u}) = \text{Sigm} \left( \underline{c}_f^{(1)} + \sum_{i=1}^m \frac{J^T w_{ij}}{\sigma_i} + \sum_{i=2}^L \sum_{i=1}^{S_i} w_{ji}^{(1,1)} \underline{h}_j^{(1)} + \sum_{i=1}^n \underline{u}_{il}^{(1)} u_i \right) \quad (25)$$

where

$$\underline{H}_{-i} = (\bar{h}^{(1)}, \bar{h}^{(2)}, \dots, \bar{h}^{(i-1)}, \bar{h}^{(i+1)}, \dots, \bar{h}^{(L)})$$

Every calculated value is stored on a hard drive. Multiconnected Boltzmann machine training uses adaptive learning rate. The reconstruction error is taken into consideration while using the LR (learning rate) search process. We determine the initial value of the reconstruction error in the first epoch. A significant challenge is selecting the variable learning rate's initial value. The results of the studies show that the initial estimation of the learning rate is greatly influenced by the input signal, or input data. For the starting value of the learning rate, which we chose to be between 0.01 and 0.001, we get successful outcomes. All neural network-related data, such as hidden units, biases, offsets, and weights, is preserved in the hard disc after the first epoch. The epoch number  $k + 1$  is put into action. We raise the module of the learning rate and go on to the next epoch if the  $k + 1$ -th reconstruction error is less than the  $kk$ -th error. In this situation, it is best to leave the learning rate's indicator alone. We go back to the  $k$ -th epoch loading the data from the disc if the newly discovered error is greater than the one that was previously estimated. Additionally, we modify the learning rate's sign and lower its module. This strategy guarantees a steady decrease in the reconstruction error. Prior to anything else, a number representing the smallest permissible module of learning rate should be established. A learning rate that is nearly zero means that it is impossible to find a smaller reconstruction mistake. This explains why Eqn(26) inequality exists

$$|\lambda_k| < v \quad (26)$$

creates a stop criterion for the deep Boltzmann machine training. Many neural network developers only run a predetermined number of epochs before stopping, regardless of any stop criteria. There is a genuine risk of falling into a local minimum or a plateau even when two neighbouring values of the reconstruction error are quite near to one another. The figure 3 depicts the LR search process. The mean field approximation controls the states of hidden neurons throughout the positive phase. The visible and output units' initial values in the negative phase are taken from the data. The initial hypotheses for the Gibbs sampling process in the negative phase are the values of the hidden units in the positive phase.

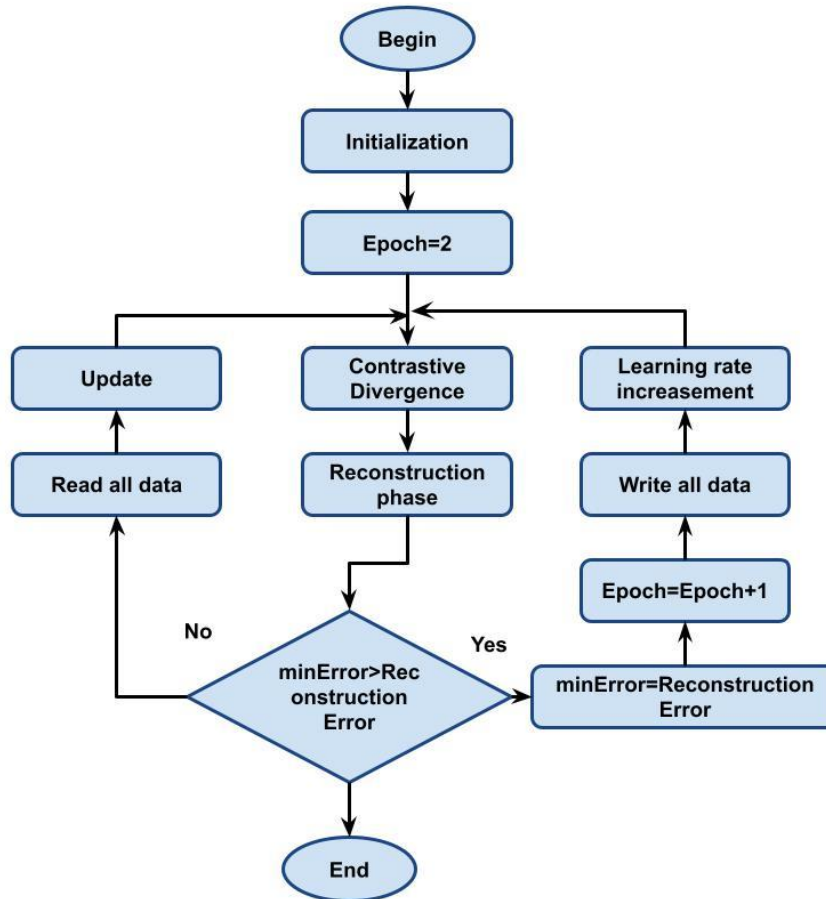


Figure 5. Flowchart of Boltzmann architecture

The target  $y$ , which is a class label, is formatted in "one out of  $C$ " manner.  $Y = e_c$ , where  $e_c$  is a vector with all values set to 0 except at location  $c$ , which is set to 1, if  $x$  belongs to class  $c$ .

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = -\mathbf{d}^T \mathbf{y} - \mathbf{c}^T \mathbf{h} - \mathbf{b}^T \mathbf{x} - \mathbf{h}^T \mathbf{W} \mathbf{x} - \mathbf{h}^T \mathbf{U} \mathbf{y}, \quad (25)$$

Eq(26) represents the probability for a certain combination of  $x$ ,  $y$ , and  $h$ .

$$p(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \exp(-E(\mathbf{x}, \mathbf{y}, \mathbf{h})) / Z \quad (26)$$

where  $Z$  is a normalising constant that guarantees that the distribution defined by  $p(\mathbf{x}, \mathbf{y}, \mathbf{h})$  is correct. The following conditional distributions of method are themselves tractable by Eqn. (27) despite the fact that  $Z$  is typically hard to evaluate:

$$\begin{aligned} p(\mathbf{h} | \mathbf{x}, \mathbf{y}) &= \prod_{j=1}^H p(h_j | \mathbf{x}, \mathbf{y}) \\ p(h_j = 1 | \mathbf{x}, \mathbf{y}) &= \text{sigm}(c_j + \mathbf{W}_j \cdot \mathbf{x} + \mathbf{U}_j \cdot \mathbf{y}) \\ p(\mathbf{x}, \mathbf{y} | \mathbf{h}) &= p(\mathbf{y} | \mathbf{h}) \prod_{i=1}^D p(x_i | \mathbf{h}) \\ p(x_i = 1 | \mathbf{h}) &= \text{sigm}(b_i + \mathbf{h}^T \mathbf{W}_i) \\ p(\mathbf{y} = \mathbf{e}_c | \mathbf{h}) &= \frac{\exp(\mathbf{d}^T \mathbf{y} + \mathbf{h}^T \mathbf{U} \mathbf{y})}{\sum_{c'=1}^C \exp(\mathbf{d}^T \mathbf{e}_{c'} + \mathbf{h}^T \mathbf{U} \mathbf{e}_{c'})} \end{aligned} \quad (28)$$

where  $(1/(1 + \exp(v))) = \text{sigm}(v)$  where the  $j$ th row of matrix  $A$  is denoted by the notation  $A_j$ , while the  $i$ th column is denoted by  $A_i$ . The posterior class probability distribution given some input  $x$  may also be demonstrated given (29), and it has the closed form.



$$p(\mathbf{y} = \mathbf{e}_c | \mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y} = \mathbf{e}_c, \mathbf{h} | \mathbf{x}) = \frac{\exp(-F(\mathbf{x}, \mathbf{y}))}{\sum_{c'=1..C} \exp(-F(\mathbf{x}, \mathbf{e}_{c'}))} \quad (29)$$

where  $F(\mathbf{x}, \mathbf{y})$  is referred to as free-energy  $F(\mathbf{x}, \mathbf{y}) = -\mathbf{d}^T \mathbf{y} - \sum^H \text{softplus}(c_j + \mathbf{W}_j \cdot \mathbf{x} + \mathbf{U}_j \cdot \mathbf{y})$  using  $\text{softplus}(v) = \log(1 + v \exp)$  Different methods can be used to train the ClassRBM. By minimising average negative. Simple stochastic gradient descent can accomplish this. Another choice is to generatively train ClassRBM by minimising negative joint log-likelihood  $\log p. (x_t, y_t)$ . It is impossible to precisely compute the required gradients. However, a good approximation using Eqn(30) is provided by the Contrastive Divergence (CD) method.

$$\mathbf{E}_{\mathbf{h} | \tilde{x}_t, \tilde{y}_t} \left[ \frac{\partial E(\tilde{x}_t, \tilde{y}_t, \mathbf{h})}{\partial \theta} \right] \quad (30)$$

$x_t$  and  $y_t$  are the outcomes of a one-step Gibbs sampling chain, initialised at training example  $x_t$  and  $y_t$ . where  $\theta$  is any parameter of the ClassRBM. We obtain the following stochastic gradient update from CD by Eqn. (31) when we take into account  $h_t = \text{sigm}(c + \mathbf{W}x_t + \mathbf{U}y_t)$  and  $\tilde{h}_t = \text{sigm}(c + \mathbf{W}x_t + \mathbf{U}y_t)$ :

$$\begin{aligned} \mathbf{b} &\leftarrow \mathbf{b} + \lambda(x_t - \tilde{x}_t) \\ \mathbf{c} &\leftarrow \mathbf{c} + \lambda(\mathbf{h}_t - \tilde{\mathbf{h}}_t) \\ \mathbf{d} &\leftarrow \mathbf{d} + \lambda(y_t - \tilde{y}_t) \\ \mathbf{W} &\leftarrow \mathbf{W} + \lambda(\mathbf{h}_t \mathbf{x}_t^T - \tilde{\mathbf{h}}_t \tilde{x}_t^T) \\ \mathbf{U} &\leftarrow \mathbf{U} + \lambda(\mathbf{h}_t \mathbf{y}_t^T - \tilde{\mathbf{h}}_t \tilde{y}_t^T) \end{aligned} \quad (31)$$

The process of doing both discriminative and generative parameter updates with different learning rates is known as hybrid generative/discriminative learning. We may mandate that the concealed units' activity be mutually exclusive throughout the set's vectors in order to take advantage of this knowledge.

$$\sum_{s=1}^{|\mathbf{X}|} h_j^{(s)} \in \{0,1\} \forall j = 1 \dots H \quad (32)$$

For example, just one hidden unit  $h(s)j$  should be active across all vectors  $x(s)$  for all hidden unit location  $j$ . So, using Eq(33), we define energy function

$$E(\mathbf{X}, \mathbf{y}, \mathbf{H}) = -\mathbf{d}^T \mathbf{y} - \sum_s \mathbf{b}^T \mathbf{x}^{(s)} - \sum_s \mathbf{c}^T \mathbf{h}^{(s)} - \sum_s (\mathbf{h}^{(s)T} \mathbf{W} \mathbf{x}^{(s)} + \mathbf{h}^{(s)T} \mathbf{U} \mathbf{y}) \quad (33)$$

where the identical connection matrix  $\mathbf{U}$  connects the target  $\mathbf{y}$  to every hidden layer. This version of the ClassRBM for sets will be referred to as ClassSetRBMXOR. For an example of ClassSetRBMXOR, see Figure 2. It can be demonstrated that ClassSetRBMXOR has simple conditional distributions as well, while being more complex than the ClassRBM for single vectors. By using Eq(34), the conditional distribution of hidden layers is created

$$\begin{aligned} p(\mathbf{H} | \mathbf{X}, \mathbf{y}) &= \prod_{j=1}^H p\left(\left\{h_j^{(s)}\right\}_{s=1}^{|\mathbf{X}|} | \mathbf{X}, \mathbf{y}\right) \\ p\left(h_j^{(s)} = 1 | \mathbf{X}, \mathbf{y}\right) &= \frac{\exp(\text{act}(\mathbf{x}^{(s)}, \mathbf{y}))}{1 + \sum_{j'=1}^X \exp(\text{act}(\mathbf{x}^{(s)}, \mathbf{y}))} \\ p\left(h_j^{(s)} = 0 | \mathbf{X}, \mathbf{y}\right) &= \frac{1}{1 + \sum_{j'=1}^{|\mathbf{X}|} \exp(\text{act}(\mathbf{x}^{(s)}, \mathbf{y}))} \end{aligned} \quad (34)$$

where  $\text{act}(\mathbf{x}^{(s)}, \mathbf{y}) = c_j + \mathbf{W}_j \cdot \mathbf{x}^{(s)} + \mathbf{U}_j \cdot \mathbf{y}$  and statement  $h_j^{(s)} = 0$  is a shorthand for  $h_j^{(s)} = 0 \forall s = 1, \dots, |\mathbf{X}|$ . input and target vectors' distribution are given by Eqn. (35)

$$\begin{aligned} p(\mathbf{X}, \mathbf{y} | \mathbf{H}) &= p(\mathbf{y} | \mathbf{H}) \prod_{s=1}^{|\mathbf{X}|} \prod_{i=1}^D p\left(x_i^{(s)} | \mathbf{h}^{(s)}\right) \\ p\left(x_i^{(s)} = 1 | \mathbf{h}^{(s)}\right) &= \text{sigm}(b_i + \mathbf{h}^{(s)} \mathbf{W}_{-i}) \\ p(\mathbf{y} = \mathbf{e}_c | \mathbf{H}) &= \frac{\exp(\mathbf{d}^T \mathbf{y} + \sum_{s=1}^{|\mathbf{X}|} \mathbf{h}^{(s)T} \mathbf{U} \mathbf{y})}{\sum_{c'=1}^C \exp(\mathbf{d}^T \mathbf{e}_{c'} + \sum_{s=1}^{|\mathbf{X}|} \mathbf{h}^{(s)T} \mathbf{U} \mathbf{e}_{c'})} \end{aligned} \quad (35)$$

Given  $\mathbf{X}$  and  $\mathbf{y}$ , one can sample each element of  $\mathbf{H}$  and then sample new values for vectors in  $\mathbf{X}$  and for  $\mathbf{y}$  to perform Gibbs sampling on these conditional distributions. It is also possible to compute the target posterior  $p(\mathbf{y} | \mathbf{X})$  effectively. It is demonstrated that it has the subsequent Eqn (36)

$$p(\mathbf{y} = \mathbf{e}_c | \mathbf{X}) = \frac{\exp(-F^{\text{XOR}}(\mathbf{X}, \mathbf{y}))}{\sum_{c'=1..C} \exp(-F^{\text{XOR}}(\mathbf{X}, \mathbf{e}_{c'}))} \quad (36)$$

where free-energy  $F^{\text{XOR}}(\mathbf{X}, \mathbf{y})$  is now by Eqn. (37)

$$F^{\text{XOR}}(\mathbf{X}, \mathbf{y}) = -\mathbf{d}^T \mathbf{y} - \sum_{j=1}^H \text{softplus}^{\text{softmax}}(\mathbf{X}) + \mathbf{U}_j \cdot \mathbf{y} \quad (37)$$

with  $\text{softmax}_j(\mathbf{X}) = \log \left( \sum_{s=1}^{|\mathbf{X}|} \exp(c_j + \mathbf{W}_{j,\mathbf{x}^{(s)}}) \right)$  is seen as a soft version of max function of  $c_j + \mathbf{W}_{j,\mathbf{x}^{(s)}}$  over set of input vectors.

<b>Algorithm of SABEM:</b>
<p>Si is the current number of a training pair  begin  The mean field approximation  The value <math>k</math> indicates the number of iterations  The positive phase  fork = 0 to <math>K - 1</math> do  begin  The first hidden layer  for <math>j = 1</math> to <math>s_1</math> do  begin  <math>\underline{h}_{1j}^{[K+1]}: \text{Sigm} \left( \underline{c}_j^{(1)} + \sum_{i=1}^m \frac{v_j^T w_{ij}}{\sigma_i} + \sum_{i=2}^L \sum_{i=1}^{s_i} \tilde{w}_{fi}^{(1,l)} \underline{h}_{ii}^{[K]} + \sum_{i=1}^n U_{ii}^{(1)} u_i \right)</math>  end <math>j</math>  for <math>q = 2</math> to <math>L - 1</math> do  begin  Next hidden layers  for <math>j = 1</math> to <math>s_q</math> do  begin  <math>\underline{h}_{qj}^{[K+1]} := \text{Sigm} \left( \underline{c}_j^{(q)} + \sum_{i=1}^{q-1} \sum_{i=1}^{s_i} \left( \underline{h}_{ii}^{[K+1]} \right)^T \tilde{w}_{ij}^{(l,i)} + \sum_{l=q+1}^L \sum_{i=1}^{s_l} \tilde{w}_{ji}^{(q,l)} \underline{h}_{ii}^{[K]} + \sum_{i=1}^n U_{ii}^{(q)} u_i \right)</math>  end <math>j</math>  end <math>q</math>  end for all  Approximate values for each unit of each hidden layer  are obtained after <math>K</math> iterations.  Af := <math>\left( \bar{h}_1^{[k]}, \bar{h}_2^{[k]}, \dots, \bar{h}_L^{[k]} \right)</math>;</p>

VGGNet is a multilayered DNN. VGGNet is used on the ImageNet dataset and is based on the CNN model. The simplicity of the VGG-19, which has  $3 \times 3$  convolutional layers stacked on top to expand with depth level, makes it handy. Max pooling layers were utilised as a handler in VGG-19 to decrease the volume size. 4096 neurons were employed with two FC layers. Figure 4 demonstrates how the VGGNet DNN used the vessel segmentation photos as input data. Convolutional layers were utilised for the feature extraction during the training phase, and some of the convolutional layers also have max pooling layers attached to them to lessen the dimensionality of the features. 64 kernels were used in first convolutional layer to extract features from input images. To create feature vector, fully connected layers were employed. For dimensionality reduction as well as feature selection of image data for improved classification results, collected feature vector was further put through PCA as well as SVD. Using PCA and SVD to minimise the highly dimensional data is a substantial undertaking. Because PCA and SVD are quicker and quantitatively more stable than other reduction approaches, they are more beneficial. Last but not least, 10-fold cross validation was used during testing to categorise DR photos using softmax activation technique.

#### 4. Experimental Analysis

The complete implementation of the suggested method is carried out using the Python tool, and the following specifications were taken into account for the experiment: a PC running Ubuntu, 4GB of RAM, and an Intel i3 processor.



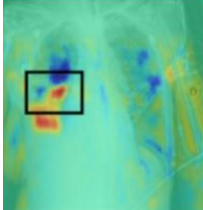
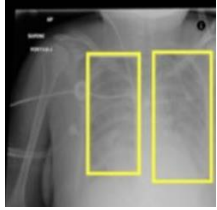

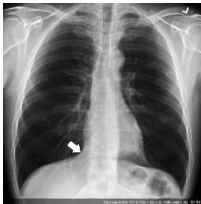
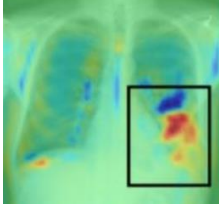
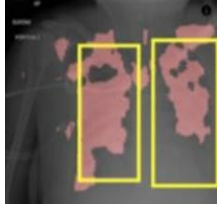


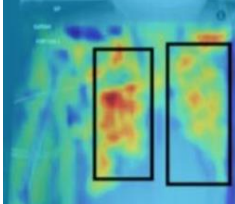

4.1 Database description

**Covid-19 Open Research Dataset (CORD-19):** Scientific papers on COVID-19 and associated earlier coronavirus studies can be found in COVID-19 Open Research Dataset (CORD-19), which is a growing database. Extensive collection of metadata and structured full text papers in CORD-19 are intended to make it easier for development of text mining as well as data retrieval methods. Over 200K people have downloaded CORD-19 since it was made available, and it has formed the foundation for other COVID-19 text mining and discovery methods. In this article, we present an overview of how CORD-19 has been utilised, detail mechanics of dataset building, highlighting obstacles and important design decisions, and describe numerous shared tasks built around dataset.

**COVID-19 X-ray dataset:** The suggested dataset was created using the Chest X-ray Images Pneumonia dataset. Each image category has its own sub-folder within the dataset's two main directories. There are four categories and 306 X-ray photos in JPEG format.

**CheXNet:** The neural network CheXNet has undergone two training sessions. With 112,120 images from 30,805 patients, ChestX-ray14 is one of the largest chest X-ray datasets. US National Institutes of Health acquired the photographs. It includes photos without any results and 14 distinct ailments.

Table 1. Comparative analysis in chest X-ray image classification for various datasets

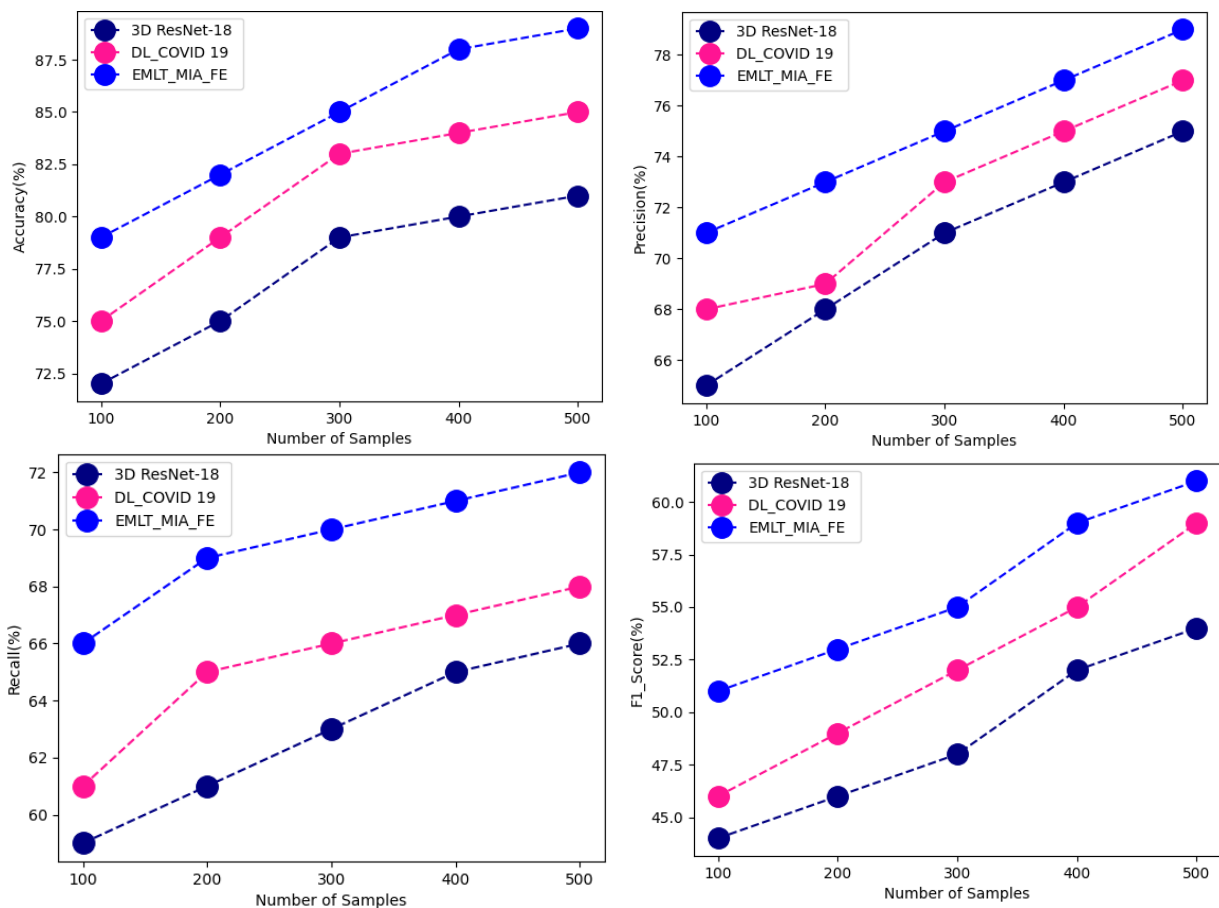
Dataset	Chest X-ray Input Image	Pre-processed image	Extracted features of input image	Classified lung image
CORD-19 Dataset				
COVID-19 X-ray Dataset				
CheXNet Dataset				

The various stages of processing input image is shown by table-1. From the above table the input has been processed based on various input datasets using proposed technique.

Table 2. Comparative analysis between proposed and existing technique

Dataset	Techniques	Accuracy	Precision	Recall	F1_Score	RMSE	MAP
CORD-19 Dataset	3D ResNet-18	81	75	66	54	41	32
	DL_COVID 19	85	77	68	59	43	34
	EMLT_MIA_FE	89	79	72	61	45	36
COVID-19 X-ray Dataset	3D ResNet-18	85	81	73	63	47	39
	DL_COVID 19	88	83	75	65	49	42
	EMLT_MIA_FE	92	85	78	71	52	44
CheXNet Dataset	3D ResNet-18	89	88	81	73	55	46
	DL_COVID 19	93	89	83	75	59	48
	EMLT_MIA_FE	95	91	85	80	61	51

The above table-2 shows comparative analysis between proposed and existing technique where the various input datasets has been processed and analysed. The parameteric analysis has been carried out in terms of accuracy, precision, recall, F\_1 score, RMSE and MAP. The datasets being compared are CORD-19, COVID-19 X-ray, and CheXNet. One parameter for assessing classification method is accuracy. Informally, accuracy is percentage of predictions that our method correctly predicted. Accuracy is defined as follows in formal language: Accuracy is the quantity of accurate forecasts. sum of all projections. ML classification algorithm's accuracy can be used to determine how frequently a data point is correctly classified. Accuracy is defined as the percentage of correctly predicted data points among all the data points. One indicator of the model's performance is precision, or quality of a successful prediction. Total number of accurate positive predictions is divided by total number of real positives to determine precision. Recall increases as more positive samples are found. By evaluating harmonic mean of a classifier's precision and recall, F1-score integrates both into a single metric. RMSE, sometimes referred to as RMS deviation, is one of the techniques most frequently used to evaluate accuracy of forecasts. It illustrates the Euclidean distance between measured true values and forecasts. Using a model and a prior probability or belief about the model, MAP entails computing a conditional probability of observing the data. For ML, MAP offers an alternative probability framework to maximum likelihood estimation.



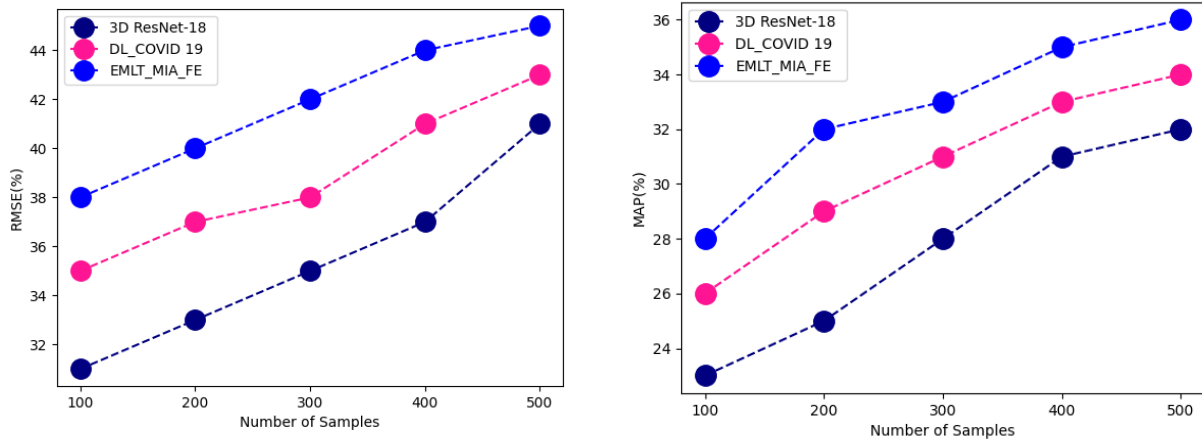
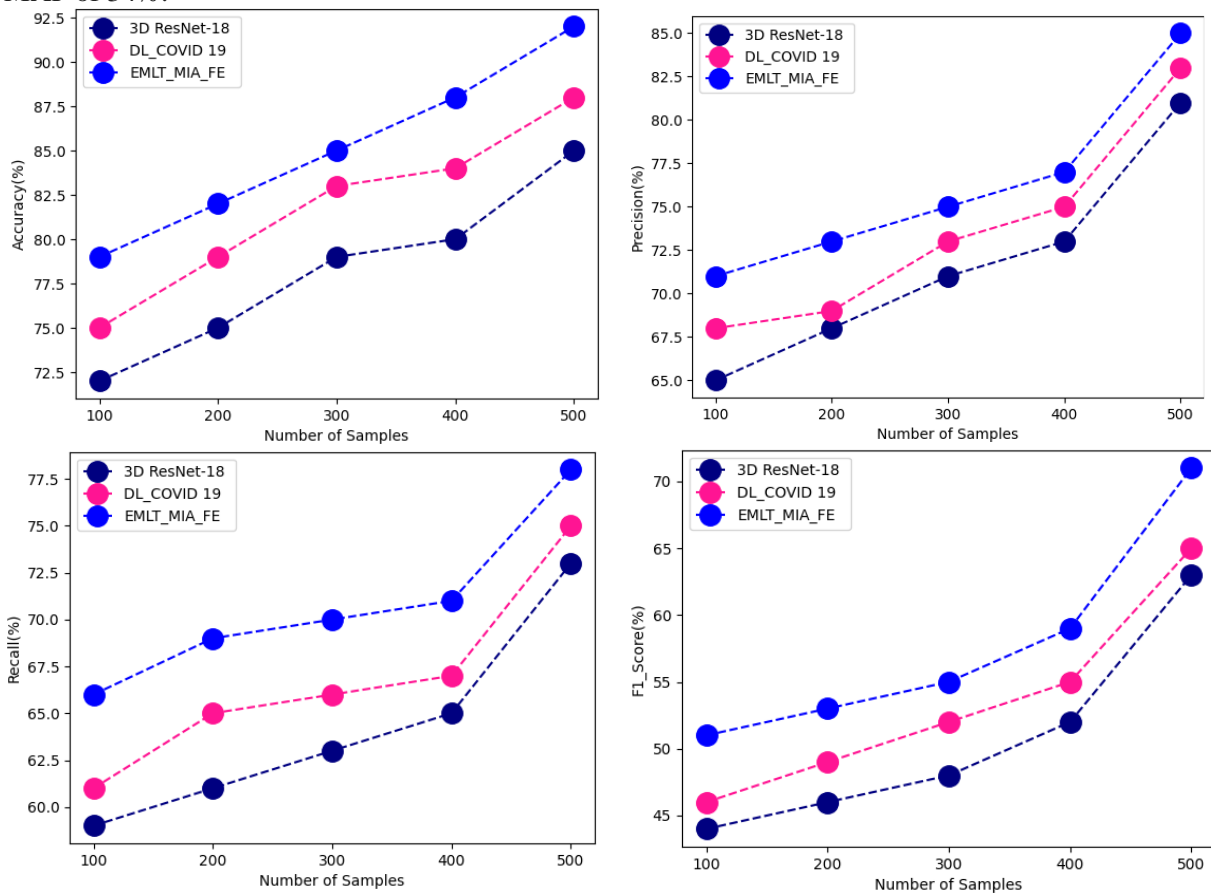


Figure 6. Comparative analysis between proposed and existing technique in terms of accuracy, precision, recall, F<sub>1</sub> score, RMSE and MAP for CORD-19 Dataset

Above figure-6 shows comparative analysis between proposed and existing technique for CORD-19 Dataset. The proposed technique attained accuracy of 89%, precision of 79%, recall of 72%, F<sub>1</sub> score of 61%, RMSE of 45% and MAP of 36%; while existing 3D ResNet-18 attained accuracy of 81%, precision of 75%, recall of 66%, F<sub>1</sub> score of 54%, RMSE of 41% and MAP of 32%, DL\_COVID 19 attained accuracy of 85%, precision of 77%, recall of 68%, F<sub>1</sub> score of 59%, RMSE of 43% and MAP of 34%.



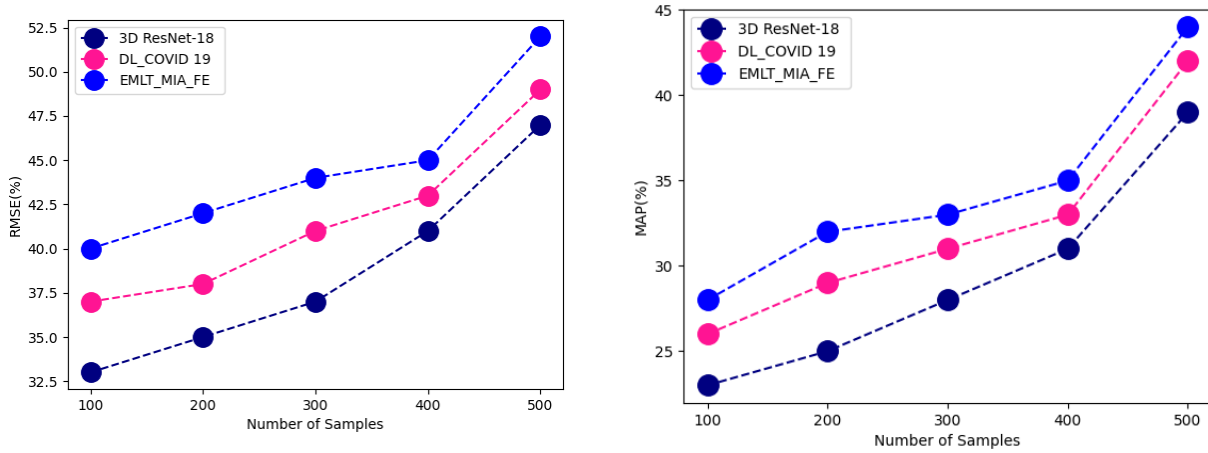
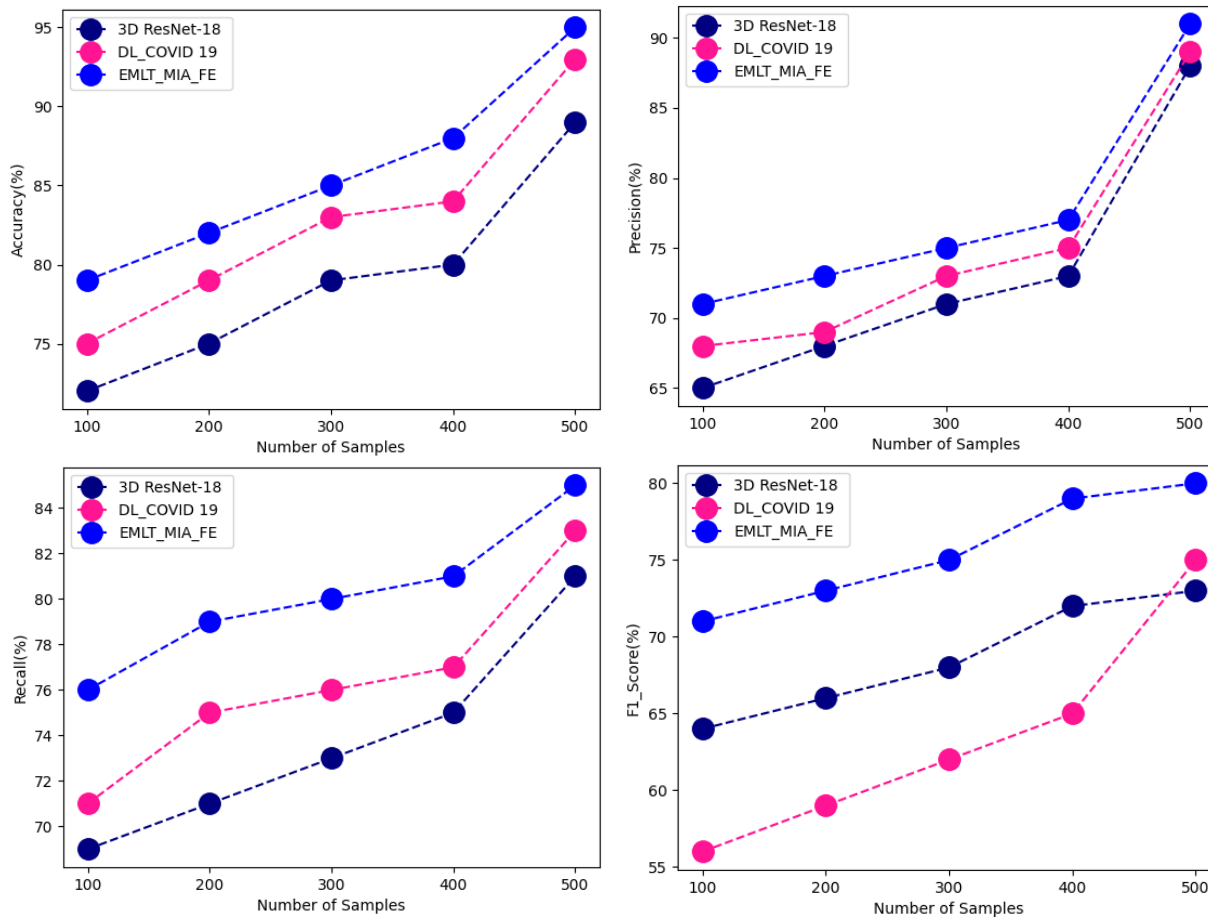


Figure 7. Comparative analysis between proposed and existing technique in terms of accuracy, precision, recall, F<sub>1</sub> score, RMSE and MAP for COVID-19 X-ray Dataset

Above figure-7 represents comparative analysis between proposed and existing technique for COVID-19 X-ray Dataset. Proposed technique attained accuracy of 92%, precision of 85%, recall of 78%, F<sub>1</sub> score of 71%, RMSE of 52% and MAP of 44%; while existing 3D ResNet-18 attained accuracy of 85%, precision of 81%, recall of 73%, F<sub>1</sub> score of 63%, RMSE of 47% and MAP of 39%, DL\_COVID 19 attained accuracy of 88%, precision of 83%, recall of 75%, F<sub>1</sub> score of 65%, RMSE of 49% and MAP of 42%.



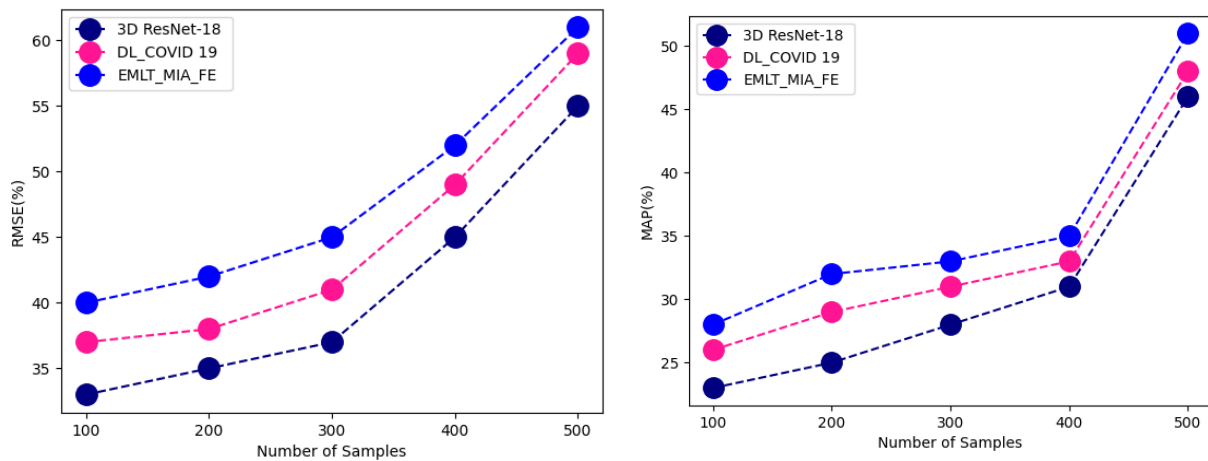


Figure 8. Comparative analysis between proposed and existing technique in terms of accuracy, precision, recall, F<sub>1</sub> score, RMSE and MAP for CheXNet Dataset

Above figure-8 shows comparative analysis between proposed and existing technique for CheXNet Dataset. The proposed technique attained accuracy of 95%, precision of 91%, recall of 85%, F<sub>1</sub> score of 80%, RMSE of 61% and MAP of 51%; while existing 3D ResNet-18 attained accuracy of 89%, precision of 88%, recall of 81%, F<sub>1</sub> score of 73%, RMSE of 55% and MAP of 46%, DL\_COVID 19 attained accuracy of 93%, precision of 89%, recall of 83%, F<sub>1</sub> score of 75%, RMSE of 59% and MAP of 46%.

## 5. Conclusion

This research propose novel method in lung image analysis for detection of lung infection due to COVID using Explainable Machine learning techniques. The processed input image features has been extracted using spatio transfer neural network integrated with DenseNet+ architecture. Extracted features has been classified using stacked auto Boltzmann encoder machine with VGG-19Net+. When using COVID-19 detection methods or any other ML/DL approaches, segmentation of lungs from X-ray can increase accuracy of results. To compare the algorithms, the categorised lungs are validated using intersection and union scores. Preprocessed X-ray images have higher classification accuracy for all three classes than raw, unedited images. The proposed technique attained accuracy of 95%, precision of 91%, recall of 85%, F<sub>1</sub> score of 80%, RMSE of 61% and MAP of 51%.

## References

- [1] Luo, W., Yu, Z. Y., Xiao, S. J., Zhu, A. X., & Yuan, L. W. (2015). Exploratory method for spatio-temporal feature extraction and clustering: An integrated multi-scale framework. *Isprs International Journal of Geo-Information*, 4(4), 1870-1893.
- [2] Hu, K., Zheng, F., Weng, L., Ding, Y., & Jin, J. (2021). Action Recognition Algorithm of Spatio-Temporal Differential LSTM Based on Feature Enhancement. *Applied Sciences*, 11(17), 7876.
- [3] Mesmakhosroshahi, M., & Kim, J. (2012, November). Improving spatio-temporal feature extraction techniques and their applications in action classification. In *2012 Visual Communications and Image Processing* (pp. 1-6). IEEE.
- [4] Ruiz, J., Mahmud, M., Modasshir, M., Shamim Kaiser, M., & Alzheimer's Disease Neuroimaging Initiative, F. T. (2020, September). 3D DenseNet ensemble in 4-way classification of Alzheimer's disease. In *International Conference on Brain Informatics* (pp. 85-96). Springer, Cham.
- [5] Li, D., Fu, Z., & Xu, J. (2021). Stacked-autoencoder-based model for COVID-19 diagnosis on CT images. *Applied Intelligence*, 51(5), 2805-2817.
- [6] Zhang, K., Guo, Y., Wang, X., Yuan, J., & Ding, Q. (2019). Multiple feature reweight densenet for image classification. *IEEE Access*, 7, 9872-9880.

- [7] Tsanev, G. S. (2017). Deep multiconnected Boltzmann machine for classification. *Amer. J. Eng. Res.*, 6(5), 186-194.
- [8] Louradour, J., & Larochelle, H. (2011). Classification of sets using restricted Boltzmann machines. *arXiv preprint arXiv:1103.4896*.
- [9] Mateen, M., Wen, J., Song, S., & Huang, Z. (2018). Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry*, 11(1), 1.
- [10] Subramaniam, U., Subashini, M. M., Almakhlles, D., Karthick, A., & Manoharan, S. (2021). An expert system for COVID-19 infection tracking in lungs using image processing and deep learning techniques. *BioMed Research International*, 2021.
- [11] McDermott, C., Łacki, M., Sainsbury, B., Henry, J., Filippov, M., & Rossa, C. (2021). Sonographic diagnosis of COVID-19: A review of image processing for lung ultrasound. *Frontiers in big Data*, 4, 612561.
- [12] Oulefki, A., Agaian, S., Trongtirakul, T., & Laouar, A. K. (2021). Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images. *Pattern recognition*, 114, 107747.
- [13] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S. B. A., ... & Chowdhury, M. E. (2021). Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in biology and medicine*, 132, 104319.
- [14] Born, J., Wiedemann, N., Cossio, M., Buhre, C., Brändle, G., Leidermann, K., ... & Borgwardt, K. (2021). Accelerating detection of lung pathologies with explainable ultrasound image analysis. *Applied Sciences*, 11(2), 672.
- [15] Ranjbarzadeh, R., Jafarzadeh Ghouschi, S., Bendechange, M., Amirabadi, A., Ab Rahman, M. N., Baseri Saadi, S., ... & Kooshki Forooshani, M. (2021). Lung infection segmentation for COVID-19 pneumonia based on a cascade convolutional network from CT images. *BioMed Research International*, 2021.
- [16] Bhattacharya, S., Maddikunta, P. K. R., Pham, Q. V., Gadekallu, T. R., Chowdhary, C. L., Alazab, M., & Piran, M. J. (2021). Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. *Sustainable cities and society*, 65, 102589.
- [17] Bhattacharya, S., Maddikunta, P. K. R., Pham, Q. V., Gadekallu, T. R., Chowdhary, C. L., Alazab, M., & Piran, M. J. (2021). Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. *Sustainable cities and society*, 65, 102589.
- [18] Sangeetha, S. K. B., Afreen, N., & Ahmad, G. (2021). A Combined Image Segmentation and Classification Approach for COVID-19 Infected Lungs. *J. homepage*, 8(3), 71-76.
- [19] Mu, N., Wang, H., Zhang, Y., Jiang, J., & Tang, J. (2021). Progressive global perception and local polishing network for lung infection segmentation of COVID-19 CT images. *Pattern Recognition*, 120, 108168.
- [20] Tello-Mijares, S., & Woo, L. (2021). Computed tomography image processing analysis in COVID-19 patient follow-up assessment. *Journal of Healthcare Engineering*, 2021.