

## Research Article

# Towards Grading Chest X-rays of COVID-19 Patients Using A Dynamic Radial Basis Function Network Classifier

Subhagata Chattopadhyay 

Acculi Labs Pvt. Ltd. Bangalore 560098, Karnataka, India  
E-mail: [subhagata.chattopadhyay2017@gmail.com](mailto:subhagata.chattopadhyay2017@gmail.com)

**Received:** 30 August 2021; **Revised:** 25 September 2021; **Accepted:** 22 October 2021

**Abstract:** The high volume of COVID-19 Chest X-rays and less number of radiologists to interpret those is a challenge for the highly populous developing nations. Moreover, correct grading of the COVID-19 stage by interpreting the Chest X-rays manually is time-taking and could be biased. It often delays the treatment. Given the scenario, the purpose of this study is to develop a deep learning classifier for multiple classifications (e.g., mild, moderate, and severe grade of involvement) of COVID-19 Chest X-rays for faster and accurate diagnosis. To accomplish the goal, the raw images are denoised with a Gaussian filter during pre-processing followed by the Regions of Interest, and Edge Features are identified using Canny's edge detector algorithm. Standardized Edge Features become the training inputs to a Dynamic Radial Basis Function Network classifier, developed from scratch. Results show that the developed classifier is 88% precise and 86% accurate in classifying the grade of illness with a much faster processing speed. The contribution lies in the dynamic allocation of the (i) number of Input and Hidden nodes as per the shape and size of the image, (ii) Learning rate, (iii) Centroid, (iv) Spread, and (v) Weight values during squared error minimization; (vi) image size reduction (37% on average) by standardization, instead of dimensionality reduction to prevent data loss; and (vii) reducing the time complexity of the classifier by 26% on average. Such a classifier could be a reliable assistive tool to human doctors in screening and grading COVID-19 patients and in turn, would help faster management of the patients as per the stages of COVID-19.

**Keywords:** COVID-19, chest X-ray, radial basis function net, classification, image processing, edge features

## Abbreviations

BPN	Back Propagation Net
BV	Biological Vision
CO	Computed Output
CNN	Convolutional Neural Network
COVID-19	Corona Virus Disease 2019
CT	Computerized Tomography
CV	Computer Vision
CXR	Chest X-ray

DRBFN	Dynamic Radial Basis Function Net
EF	Edge Features
ED	Euclidean Distance
FP	False Positive
FN	False Negative
FFNN	Feed Forward Neural Net
GF	Gaussian Filter
HL	Hidden Layer
IL	Input Layer
KMC	K-Means Clustering
m	Mild
M	Moderate
MRI	Magnetic Resonance Imaging
OL	Output Layer
Q1	Quartile 1
Q2	Quartile 2
Q3	Quartile 3
RBF	Radial Basis Function
ROI	Regions of Interest
S	Severe
SE	Squared Error
TO	Target Output
USG	Ultrasonography
VC	Visual Cortex

## 1. Introduction

Biological vision (BV) is one of the most important sensory systems in humans and is still evolving continuously. Photons dissipated by any structure are captured by our Retina as pixels. The signal thus generated in the Retina then travels to the Occipital or Visual cortex (VC) via Optic nerves of each eye and the pixels having high values play a critical role in capturing the first visual impression of any object. Pixels on or near the *edges* of an object possess high values, which are therefore commonly (i) captured, (ii) registered or stored, (iii) retrieved, and (iv) processed in VC. The V1 cells in the vision path play a key role in that edge detection, orientation, and assessment of axial rotation of any image of interest to the brain [1]. Computer vision (CV) attempts to mimic BV for image processing by computer algorithms. Hence, edge detection to identify salient Edge Features (EF) of any image is the key to its successful implementation [2].

Feature selection is an important domain of image processing and various machine learning and deep learning techniques and algorithms have been used so far for text document clustering [3], Aquila optimizer that studies the feature behavior of Aquila when catching its prey [4], novel multilevel thresholding of COVID-19 CT images using the Arithmetic Optimization Algorithm (AOA) [5], machine learning classifiers [6], and so forth.

Medical imaging has been popular for screening, diagnosis, and prognostic purposes for decades [7-8]. X-ray radiographs are still the modality of choice due to their easy availability and low cost to the patients, especially in the developing nations, where other modalities such as Computerized Tomography (CT), Ultrasonography (USG), and Magnetic Resonance Imaging (MRI) facilities are not so easily available and are costlier than that of X-rays. COVID-19 virus predominantly affects the respiratory system. Hence, a Chest X-ray (CXR) remains crucial to the diagnosis of severity.

In a CXR image, lungs and their accessory organs are found to be grossly overlapped on each other because of their normal anatomical position, spread, and axial twists, especially the mediastinum region [9]. When pathological, their alignment, texture, color, and shape are altered, which can be identified by several edge detection algorithms and the deformed regions may be the Regions of Interest (ROI) to the human doctors. The Edge Features (EF), which

are nothing but the higher pixel values on and near the edges of an ROI, therefore, store the significant information to classify the image into normal or abnormal with the severity grade based on the extent of deformities, represented by changes in the pixel values.

In CV research, classifiers are gaining lots of popularity due to the ongoing COVID-19 pandemic [10]. Several studies are reported using hard classifiers, such as Support Vector Machines, Random Forest, Logistic regressions, Multiple linear regressions, Neural networks with both adaptive and deep learning techniques, and so forth in classifying normal and abnormal images [10]. Detail review of the current state of the art literature has been conducted in this study, which finds most of the classifiers show good performance. However, due to the space constraint, the studies have not been described in detail here. Interested readers may go through the comprehensive works of [11-13], and others.

*The research question* that this work has attempted to address is to develop a deep learning-based *dynamic* Radial Basis Function Net (DRBFN) image classifier, which is acceptably *precise, accurate, lossless, and fast* in processing the image. Current research on COVID-19 image processing is dependent upon the applications of Convolutional Neural Network (CNN) and its various extensions. Using DRBFN, grading COVID-19 CXRs, and in turn, validating the diagnostic accuracy and precision is hence a novel approach adopted in this study. It is worth mentioning that no comparison has been made between existing deep learning-based classifiers and the developed classifier as its performances vary across different image data. Hence, such a comparison would be scientifically illogical.

The structure of the remaining part of the paper is as follows. Section 2 describes the Material and method; Results are shown and discussed in Section 3 and 4, respectively, and Section 5 concludes the paper.

## 2. Material and methods

The purpose of the study is to develop an automatic, fast, and accurate COVID-19 CXR analyzer, which would be helpful to the human doctors to grade the stage of the illness and consider the treatment protocol accordingly to save lives. To accomplish the task, the following methods have been adopted to train the computer concerning the human BV:

- 2.1. Acquisition of COVID CXRs
- 2.2. Denoising of the acquired CXRs as the pre-processing step
- 2.3. ROI detection and Edge feature (EF) extractions as pixel values and its' standardization, and
- 2.4. Construction of the DRBFN classifier and its training, testing, validation, and computational time complexity estimation.

*System information:* All coding for the experiment has been conducted on Python 3.8.3 with Spyder editor version 5.0.0, preloaded with *skimage*, *NumPy*, *OpenCV*, and *pandas* for image processing and *matplotlib*, *seaborn* for visualization. All computations are run on Windows 10 Pro 64 bits OS x 64-based Processor Intel (R) Core TM @ 2.80 GHz.

### 2.1 CXR acquisition

COVID CXR images (N = 100), in which, forty-three are pre-diagnosed as 'mild' ('m' class label), thirty-four as 'moderate' ('M' class label), and twenty-three as 'severe' ('S' class label) are obtained. A group of radiologists did the clinical grading and assigned class labels for each image. Appropriate ethical measures have been taken to preserve patients' privacy. Seventy percent CXRs are used for network *training* to find the best parameters while the remaining thirty percent of data are used for *testing* the performance of the network.

For *validation*, another set of CXRs (N = 50) have been collected. These are also pre-diagnosed by the radiologists as 'm' (17), 'M' (15), and 'S' (18), respectively.

In the next step, raw CXRs are denoised or filtered.

### 2.2 Filtering

Any image is normally noisy and the noise affects its quality. Hence, as an image pre-processing method, all CXR images have been denoised using the Gaussian Filtering (GF) technique having the kernel size of  $5 \times 5$ , using equations

1 and 2 [14]. The advantage of such type of filter is that it can preserve the originality of the image by maintaining the edges, which are useful to derive the significant features of the image and are fundamental aspects of this work [14].

$$G_{(x)} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

In the case of 2D Gaussian, it is the product of two 1D Gaussians, one of each dimension, and is represented as follows,

$$G_{(x,y)} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Where ‘x’ and ‘y’ represent horizontal and vertical axes, ‘σ’ is the standard deviation of the Gaussian distribution. Values obtained from this distribution are used to build a convolutional matrix that in turn is applied to the original image. Each pixel’s new value is then set to a weighted average of that pixel’s neighborhood. The original pixels receive high weights (i.e., the high Gaussian value) and the neighborhood pixels receive lower values (i.e., smaller weights) as their distances from the original pixels increase. This operation, that’s why results in blurring of the image with ‘edge preservation’ in a much better way [15], and therefore has been used in this work. After denoising, the images, their Edge Features (EFs) are extracted. However, blurring of images can also be prevented and the original sharpness of the image can be preserved by ‘mask adjusting’ the GF, which in turn, fine-tunes the appropriate edge-angle data and preserves the sharpness of the image [16].

### 2.3 Edge Feature (EF) extraction

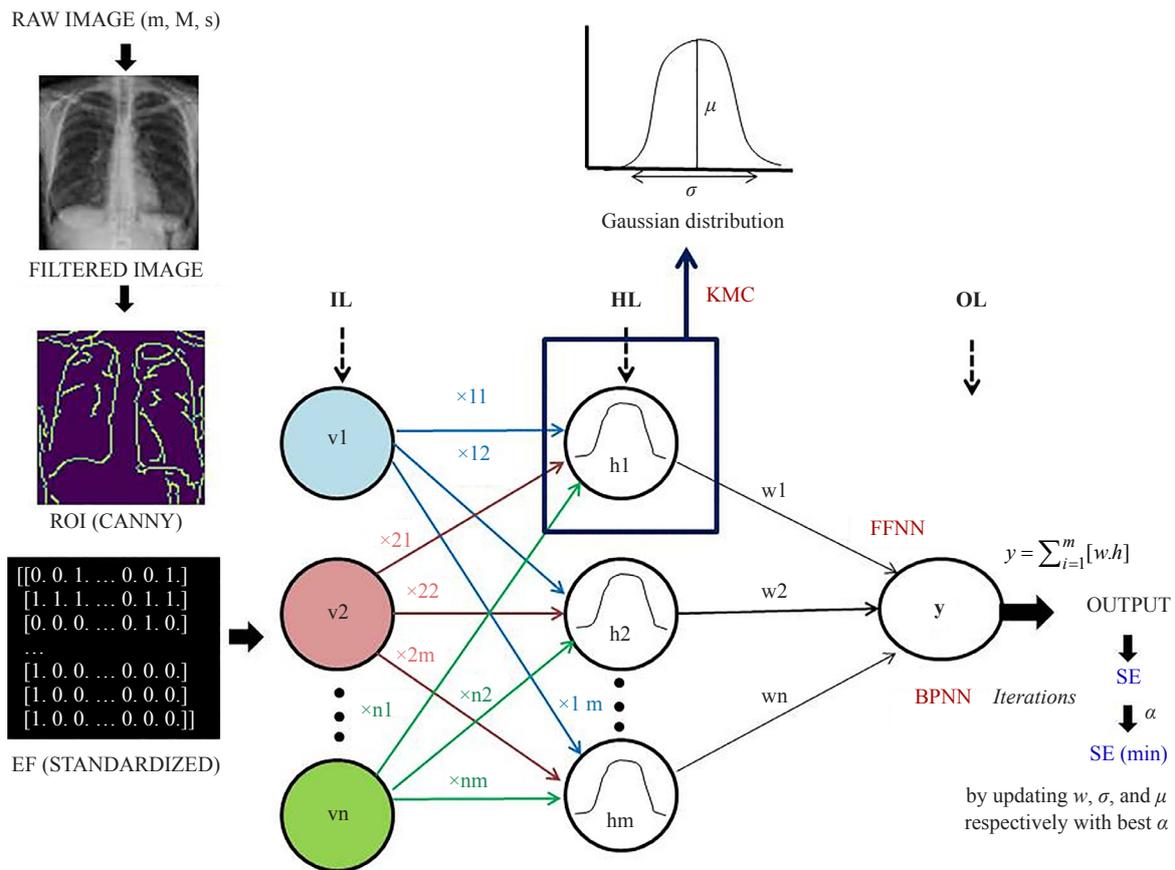
It is an important step towards medical image processing as the anatomical structures in a medical image are grossly overlapped on each other and need to be demarcated to identify the underlying pathology, where the EFs are distorted or missing [17]. In this work, Edge Features (EF) have been extracted for getting the Regions of Interest (ROI) using Canny’s EF extraction technique [18]. The reason for using the EF method is because human vision depends on ‘edges’ by activating its V1 cells of the visual path [19] for registering, retrieval, and analysis. To mimic Biological vision, Computer vision, therefore, attempts to detect the pixels of the edges, especially the high-value pixels, which are the ‘pixels of interest’ within the ROIs of any image.

Once the EF pixel values are obtained, these are then standardized (*stdz*) by subtracting the median (Q2) of each ‘x’ pixel found in each variable and then divided by the inter-quartile range (Q3-Q1) of ‘x’ using equation 3. In this equation, ‘ $x_i$ ’ is nothing by the value of  $i^{th}$  pixel of the corresponding EFs variables (refer to Figure 2).

$$stdz(x)_i = \frac{(x_i - Q2(x))}{((Q3 - Q1)(x))} \quad (3)$$

### 2.4 Construction of a Dynamic Radial Basis Function Network (DRBFN)

A Radial Basis Function Network (RBFN), originally proposed by [20] consists of three layers-Input (IL), Hidden (HL), and Output (OL). Hidden and Output layers are connected by randomly assigned weight vectors (see Figure 1). The input layer possesses one neuron for each categorical variable and thus the number of IL nodes varies according to the shape of the image as soon as it is fed into the network, which means that the number of IL nodes is dynamically assigned according to the shape of the input image and hence the developed RBFN is called as DRBFN. Nodes of IL take standardized EF pixel values as the inputs to it. The transfer functions of the IL nodes are linear. Hence, the same value is transferred to the HL nodes. The activation of each IL node occurs only when the standardized EF pixels values are non-zero.



**Figure 1.** Architecture of the DRBFN and classification method

HL has a variable number of neurons. The optimal number could be determined by the training process for each training image (see Figure 3). Each HL node consists of a ‘Radial Basis Function (RBF)’, which is centered on a point ( $\mu$ ) with dimensions similar to the number of the predictor variables. The spread or radius ( $\sigma$ ) of the RBF function may be different for each dimension. The centers (i.e., the mean or  $\mu$ ) and spreads (i.e., the standard deviation or  $\sigma$ ) are determined by the training process (see Figure 3). These two parameters are pivotal to store the EF pixels of ‘m’, ‘M’, and ‘S’ images while training the classifier. While any ‘ $x_i$ ’ input vector from the IL is fed into the hidden neuron, it computes the Euclidean distance (ED) of ‘ $x_i$ ’ (‘I’ vary from 1 to ‘N’) from the neuron’s center point ( $\mu$ ) and then applies the RBF kernel function to this distance using the spread values ( $\sigma$ ). The resulting value is passed to the OL, where summation takes place. K-Means Clustering (KMC)-based semi-supervised training algorithm has been used to identify the cluster centroids, which represent the center points ( $\mu$ ) of Hidden neurons from which EDs between each of ‘ $x_i$ ’ and ‘ $\mu$ ’ are computed [21]. Using the information of each ‘ $\mu$ ’, corresponding ‘ $\sigma$ ’ values are computed and stored after the completion of the training. It is important to mention here that, the training process involves parametric study to determine the following parameters as the training progresses; however, all parameters need to be initialized as stated below,

- The number of nodes in the IL is initialized as 100, which will change according to the Image shape, dynamically
- Number of neurons in the HL (initialized with the same number as in the Input nodes)
- Coordinates of the center ( $\mu$ , initialized with 0.5) of each HL RBF
- Radius ( $\sigma$ ) of each RBF function in each dimension (initialized with 1.0)
- Weights ( $w$ ) are applied to the RBF outputs as they are passed to the summation layer (initialized with any random float value between 0 and 1)
- Learning rate ( $\alpha$ ) is initialized with 0.05, which is required for Squared Error (SE) correction, iteratively during

training.

Figure 1 shows the structure of a DRBFN, developed in this work. Here, ‘v’ stands for ‘variables’ that vary from 1 to ‘n’. The number of Input nodes equals the number of ‘v’ (dynamically assigned according to the shape of the image); ‘x<sub>i</sub>’ (‘i’ varies from 1 to ‘N’) values come under each ‘v’, which is nothing by the standardized values of the EFs, fed to the HL nodes (h), the number of which varies from 1 to ‘m’ and the objective is to have N > m [22], which could be achieved with the training; ‘w’ is the connector weight vectors [0, 1] between the Hidden and Output nodes; summation (‘y’) takes place in the Output node, as mentioned above. A linear transfer function is chosen in the Input and Output nodes, while Hidden nodes have Gaussian transfer functions (equation 4).

$$h(x) = \exp \left\{ -\frac{(x - \mu)^2}{\sigma^2} \right\} \quad (4)$$

As mentioned Input nodes are activated only when pixel values of  $x_i \neq 0$ . Therefore, it also reduces the input pixel data size as ‘0’s are discarded. Size reduction helps to gain the processing speed of the classifier.

Seventy percent of the standardized pixels values of EF has been used to train the classifier with COVID-19 affected CXR images and tested with the remaining set of thirty percent CXRs after *10-fold cross-validation*. Squared Error (SE) is computed for the output of each training image using equation 5. In this equation, TO stands for Target Output, which is already predicted by the radiologists, while CO refers to Calculated Output, which are the outputs given by the classifier. It is worth noting here that the classification is done by a Feed-Forward Neural Network (FFNN) and SE correction (minimization) is done with a Back Propagation Net (BPN) with learning rate ( $\alpha$ ).

$$SE = (TO - CO)^2 \quad (5)$$

The minimum target SE value is set as 25% (i.e., 75% accuracy) as doctors’ average clinical accuracy is about 71% in reality [23]. Unless the target SE is achieved, training continues by updating the ‘number of hidden nodes’, ‘ $\alpha$ ’, ‘ $\mu$ ’, ‘ $\sigma$ ’, and ‘w’ values, respectively, based on these values, the DRBFN classifier is reconstructed for testing (with remaining thirty percent of the input data) and validation (a new set of fifty CXRs of COVID-19 affected patients). Accuracy and Precision, using equations 6 and 7, respectively are computed for fifty images, which have already been pre-diagnosed by the radiologists.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

In these equations, *TP*, *FP*, *TN*, *FN* refers to *True Positive*, *False Positive*, *True Negative*, and *False Negative*, respectively. These are computed based on the radiologists’ diagnoses of the CXRs as mild ‘m’, moderate ‘M’, and severe ‘S’.

Computational Time Complexity (*Big(O)*) has been calculated to process each of fifty CXR images using equation 8.

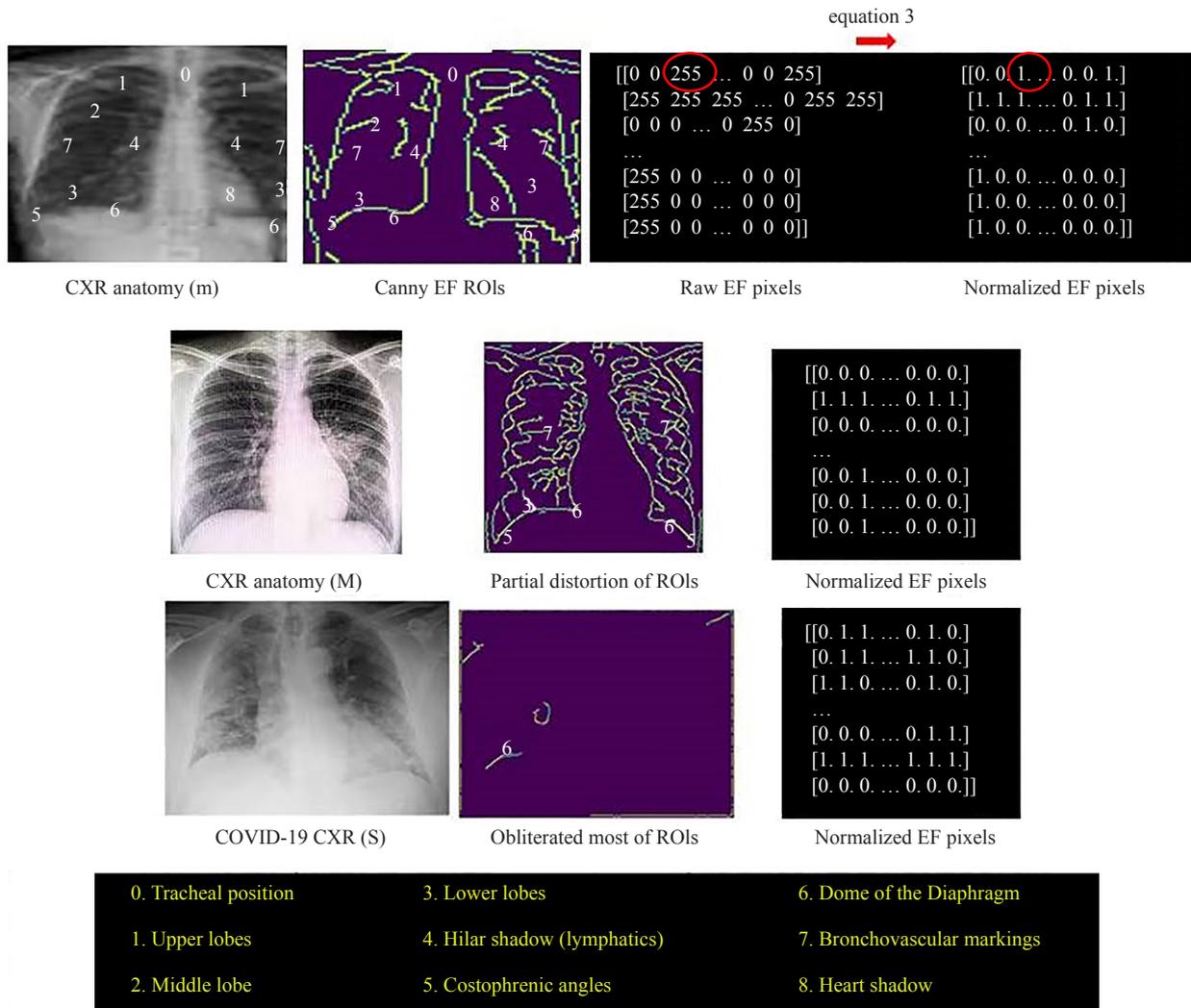
$$Big(O) = (2N) \quad (8)$$

Where ‘N’ denotes the number of *stdz* pixels. Average *Big(O)* has been plotted against the processing of each CXR image for visualization (see Table 2).

In the next section (Section III), the results of the experiments are shown and in turn, explained in Section IV.

### 3. Results

In this section, results are shown. Explanations of the results are made in Section IV. Figure 2 shows how (i) anatomical ROIs of a CXR is defined and (ii) EFs pixels are standardized for processing through IL nodes, which are activated with non-zero pixels values as these are pixels of interest of ‘m’, ‘M’, and ‘S’ grades of COVID-19 CXR images.



**Figure 2.** CXR anatomy, Canny EFs (ROIs), and Standardized EFs of ‘m-M-S’ images

Next, the results of the parametric study for one CXR image are shown in Figure 3. Similarly, for the remaining CXRs, parametric studies are conducted to find optimum (i)  $\alpha$ , (ii) training iterations, (iii) number of HL nodes (mean(hn)), (iv)  $\mu$  values, (v)  $\sigma$  values, (vi) ‘w’ values, all mapped against SE values using backpropagation. With these optimum values, DRBFN is reconstructed for testing with the remaining thirty percent CXRs and validation using a new set of fifty CXRs.

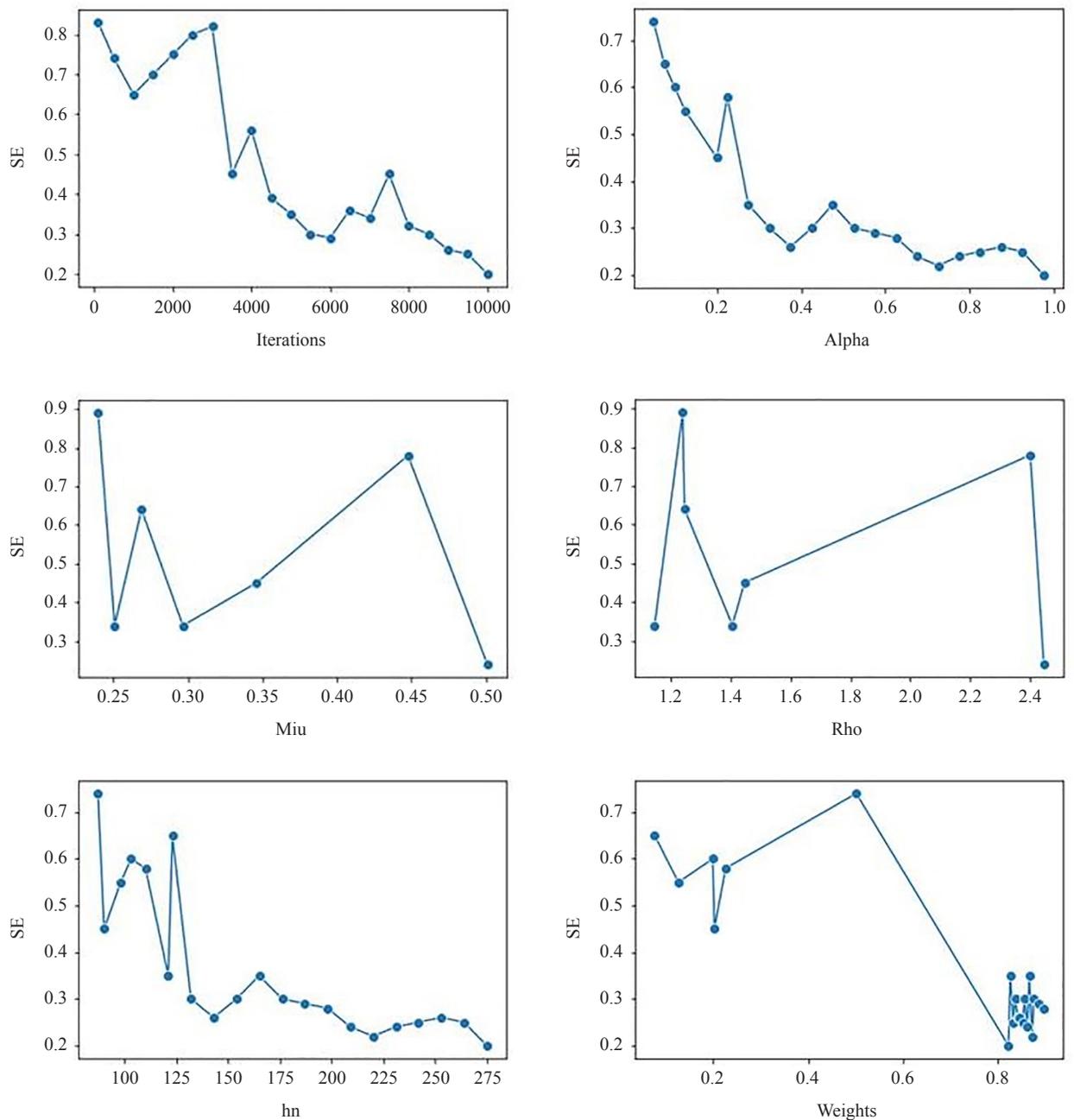


Figure 3. Results of the parametric study

Optimum values of Iterations, learning rate (alpha), number of Hidden nodes (hn), Center point (miu), Spread (rho), and Weights achieved during the training of the DRBFN classifier. It can be noted that the least SE can be achieved with 10,000 iterations, learning rate or  $\alpha$  value of 0.93, a center point or  $\mu$  value of 0.5, spread value ( $\sigma$ ) of 2.42, number of hidden nodes of 275, and weight values around 0.9. With these parameters, the DRBFN is reconstructed for testing.

Appendix-1a shows the precision and accuracy of the reconstructed classifier on fifty new COVID-19 affected CXR images. Here, the grades 'Computed by the classifier' are matched with that of the 'Actual output given by the radiologists' using equations 6 and 7. It is found that the classifier is 86% accurate and 88% precise in grading COVID-19 CXRs.

Big(O) has then been computed for each CXR processing during the above validation and the results can be seen in

## 4. Discussions

In COVID-19 infection, the respiratory system is predominantly affected. Diagnoses are made based on the grade of the involvement, which are reflected in the CXR image. The paper has addressed this issue by (i) pre-processing each image using GF to reduce noise within the image and preserving the edges, (ii) identifying the ROIs of the image using Canny edge detector, (iii) EFs, thus extracted from the ROIs, are standardized for RBFN training. Figure 2 shows the normal anatomical positions of the important structures, which are numbered accordingly in both the original and Edge-detected CXR images. The pixel values of the EFs [0, 255] are then standardized with equation 3 and it becomes [0, 1]. For the images, the median (Q2) equals '0' and the inter-quartile distance (Q3-Q1) is '255'. Hence, the final standardized input values have become binary, which is fed into the RBFN classifier (refer to Figure 1). It is important to note that the number of '1's is consistently increasing as the images are distorted due to the generation of more and more edges with the amount of distortion, i.e., pixels gain high values as the images lose their original texture (refer to Figure 2).

The shape of the sample images, i.e., 'm' = mild ( $96 \times 85$ ); 'M' = Moderate ( $155 \times 159$ ), and 'S' = Severe COVID ( $196 \times 258$ ) are shown in Figure 2. Each column represents the 'variables', while the rows are the pixels values [0, 1] under each column. The number of input nodes is dynamically assigned as the number of columns. Image size is further reduced as pixel values, which are '0's, are discarded automatically as input nodes are pre-conditioned not fire with pixel values equal to '0'. Thus, 'pixels of interest', i.e., pixels with values of '1' after standardization can be preserved, and in turn, would be used for training the RBFN classifier. Similarly, EFs have been identified and standardized for all CXR images (m, M, and S), which have been fed into the network and its training is conducted to minimize the error in diagnosis (SE), generated through iterations. Before testing the performance of the RBFN classifier, a battery of parametric studies have been conducted during its training (using 70% of the input data) to identify the (i) the optimum number of neurons in the Hidden layer by applying 'Data structure-preserving criterion technique' proposed by [24], (ii) coordinates of the center ( $\mu$ ) and its spread ( $\sigma$ ) in each hidden-layer RBF function by KMC [22], and (iii) weights ( $w$ ) applied to the RBF outputs as they are passed through the summation or output layer using least square method [25]. Iteration range is kept from 100 to 10,000 while training the RBFN. Optimum values of the above parameters can be seen in Figure 3 and are used for reconstructing the RBFN into a DRBFN classifier for testing (with the remaining 30% of the input data) and validating its performance. In the DRBFN classifier, the training EF pixel data for 'm', 'M', and 'S' are stored, on which the EF pixel data for new images are mapped for diagnosis.

Performance of the developed DRBFN classifier has been tested on fifty new COVID-19 affected CXR images by measuring the accuracy and precision using equations 6 and 7 and can be seen in Appendix-1a. In this table, the predicted classification of CXR images is validated with that of the radiologist's diagnosis. There are three classes of COVID-19 affected CXR images-'mild or m', 'Moderate or M', and 'Severe of S' grades. Misclassifications are interpreted as class predicted for 'm' but the actual class is of higher grade, i.e., 'M' or 'S' is an 'FN' case and 'FP' otherwise. When classes ('M' or 'S') match for predicted and actual, it is the 'TP' case, while the 'TN' case is considered when classes match for 'm'. The average accuracy and precision of the classifier are 86% and 88%, respectively, which is about 15% and 17% above expectation, which was originally 75%. It is worth mentioning that the DRBFN classifier can diagnose 'mild' cases with 88% accuracy, 'Moderate' cases with 86% accuracy, and 'Severe' cases with about 78% accuracy, i.e., all are well above the expected accuracy rate of 75% while developing the classifier.

Gaining classifier speed is another contribution of this study by reducing the image size and dynamically allocating the number of nodes in the Input and Hidden layer, ' $w$ ', ' $\sigma$ ', ' $\mu$ ' with iterations while minimizing SE value with the help of optimum ' $\alpha$ '. Appendix-1b shows that the Big(O) is reduced by 26% on average as the size of the images is reduced due to the standardization method by 37% to gain speed by the DRBFN classifier.

## 5. Conclusions and future research

The work proposes a *dynamic, loss-less, and fast* RBFN classifier using deep learning method for grading 'mild',

‘moderate’, and ‘severe’ COVID-19 infection in the lungs with 86% average accuracy and 88% precision, which are far better than that of the accuracy and precision of the human doctors, which is around 71% [23]. The *contribution* lies on (a) dynamic selection of Input nodes, Hidden nodes, ‘ $\mu$ ’, ‘ $\sigma$ ’, and ‘ $w$ ’ values while training, where ‘ $\mu$ ’ and ‘ $\sigma$ ’ values store EF standardized pixel information for comparing each the new CXR image for diagnosis during validation; (b) EF size has been reduced by allowing pixels with ‘ $> 0$ ’ values instead of reducing the number of variables i.e., the dimension to preserve the ‘pixels of interest’ for a given image; and it allows (c) higher speed (average 26% reduction in the time complexity) in classifying as the size of the input pixels is reduced by the average of 37% after the EF pixel values are standardized.

The *limitation* of the study is that, in a multi-class problem, in reality, the boundaries between any two grades are not crisp and often there are overlapping. Hence, soft computing techniques such as fuzzy set and fuzzy logic-based hybrid classifier are more powerful and holistic in solving those class overlaps as the literal intermediary zones. The author is working on this issue as a *future extension* of this study.

## Conflict of interest

The author declares that there is no conflict of interest.

## References

- [1] Lv C, Xu YL, Zhang XL, Ma SP, Li S, Xin P, et al. Edge detection based on primary visual pathway. *Lecture Notes in Computer Science*. Springer, Cham; 2017.
- [2] Versaci M, Morabito FC. Image edge detection: A new approach based on fuzzy entropy and fuzzy divergence. *International Journal of Fuzzy Systems*. 2021; 23: 918-936.
- [3] Abualigah LM. *Feature selection and enhanced krill herd algorithm for text document clustering in studies in computational intelligence*. Springer Nature; 2019. p. 11-19.
- [4] Abualigah L, Yousri D, Elaziz MA, Ewees AA, Al-qaness MAA, Gandomi AH. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*. 2021; 157: 107250. Available from: <https://doi.org/10.1016/j.cie.2021.107250>.
- [5] Abualigah L, Diabat A, Sumari P, Gandomi AH. A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of COVID-19 CT images. *Processes*. 2021; 9(7): 1155. Available from: <https://doi.org/10.3390/pr9071155>.
- [6] Ayoub A, Mahboob K, Javed AR, Rizwan M, Gadekallu TR, Abidi MH, et al. Classification and categorization of COVID-19 outbreak in Pakistan. *Computers, Materials and Continua*. 2021; 69(1): 1253-1269.
- [7] Agrawala AK. *Machine recognition of patterns*. New York: IEEE Press; 1977.
- [8] Onoe M, Takagi M. An automated microscope for digital image processing, Part I: Hardware. *Digital Processing of Biomedical Images*. Tokyo, University of Tokyo Press; 1976. p. 17.
- [9] Puddy E, Hill C. Interpretation of the chest radiograph. *Continuing Education in Anaesthesia Critical Care & Pain*. 2007; 7(3): 71-75.
- [10] Bhattacharya S, Maddikunta PKR, Pham Q-V, Gadekallu TR, Chowdhary CL, Alazab M, et al. Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. *Sustainable Cities and Society*. 2021; 65: 102589.
- [11] Rahman T, Khandakar A, Qiblawye Y, Tahir A, Kiranyaz S, Kashem SBA, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine*. 2021; 132: 104319.
- [12] Song Y, Zheng SJ, Li L, Zhnag X, Zhang XD, Huang ZW, et al. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2021; 1. Available from: <https://doi.org/10.1109/TCBB.2021.3065361>.
- [13] Huang S, Yang J, Fong S, Zhao Q. Artificial intelligence in the diagnosis of COVID-19: challenges and perspectives. *International Journal of Biological Sciences*. 2021; 17(6): 1581-1587.
- [14] Chattopadhyay S. A novel approach to detect abnormal chest X-rays of COVID-19 patients using image processing and deep learning. *Artificial Intelligence Evolution*. 2021; 2(2): 23-41.

- [15] Haddad RA, Akansu AN. A class of fast gaussian binomial filters for speech and image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1991; 39: 723-727.
- [16] Jundang N, Ongkittikul S, Chutisowan K, Suwatcharakulthorn J. Development of gaussian filter for preserving edge by edge template. *16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. 2019. Available from: <https://doi.org/10.1109/ECTI-CON47248.2019.8955188>.
- [17] Zhang Y, Miao S, Mansi T, Liao R. Unsupervised X-ray image segmentation with task driven generative adversarial networks. *Medical Image Analysis*. 2020; 62: 101664.
- [18] Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986; 8(6): 679-698.
- [19] Poo MM, Du JL, Ip NY, Xiong ZQ, Xu B, Tan TN. China brain project: Basic neuroscience, brain diseases, and brain-inspired computing. *Journal of Neuron*. 2016; 92(3): 591-596.
- [20] Broomhead DS, Lowe D. Multivariable functional interpolation and adaptive networks. *Complex Systems*. 1988; 2: 321-355.
- [21] MacQueen JB. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. 1967; 5(1): 281-297.
- [22] Dash SK, Behera AK, Dehuri S, Cho S-B. Radial basis function neural networks: a topical state-of-the-art survey. *Open Computer Science*. 2016; 6: 33-63.
- [23] Richens JG, Lee CM, Johri S. Improving the accuracy of medical diagnosis with causal machine learning. *Nature Communications*. 2020; 11: 3923.
- [24] Mao KJ, Huang G. Neuron selection for RBF neural network classifier based on data structure preserving criterion. *IEEE Transactions on Neural Networks*. 2005; 15(6): 15531-1540.
- [25] Zakharov AV, Peach ML, Sitzmann M, Nicklaus MC. A new approach to radial basis function approximation and its application to QSAR. *Journal of Chemical Information and Modeling*. 2014; 54(3): 713-719.

## APPENDIX-1a

**Table 1.** Accuracy and precision of the DRBFN classifier

#CXR	Calc_out	Act_out	TP	FP	TN	FN
1	m	S				X
2	M	S				X
3	S	S	X			
4	M	m		X		
5	M	M	X			
6	m	m			X	
7	S	M		X		
8	S	S	X			
9	M	m		X		
10	S	S	X			
11	m	S				X
12	m	m			X	
13	M	M	X			
14	M	M	X			
15	S	S	X			
16	S	S	X			
17	m	m			X	
18	m	m			X	
19	m	m			X	
20	M	M	X			
21	S	S	X			
22	S	M		X		
23	M	M	X			
24	S	S	X			
25	m	m			X	
26	M	M	X			
27	M	M	X			
28	S	S	X			
29	S	S	X			
30	S	S	X			
31	m	m			X	
32	M	M	X			
33	M	M	X			
34	M	M	X			
35	M	M	X			
36	m	m			X	
37	m	m			X	
38	m	m			X	
39	S	S	X			

#CXR	Calc_out	Act_out	TP	FP	TN	FN
40	S	S	X			
41	S	S	X			
42	m	m			X	
43	m	m			X	
44	m	m			X	
45	m	m			X	
46	m	m			X	
47	S	S	X			
48	S	S	X			
49	M	M	X			
50	M	M	X			
m	17		28	4	15	3
M	15	Accuracy:	86%			
S	18	Precision:	88%			

## APPENDIX-1b

**Table 2.** Reduction of Big(O) due to standardization of EF

#CXR	Shape (N)	Shape (n)	Size	Stdz_EF_size	Size reduction	Big(O)_org_size	Big(O)_stdz_size	Big(O) % redn
1	85	96	8160	5386	34%	0.135	0.09	0.333333333
2	223	219	48837	37116	24%	0.819	0.63	0.230769231
3	212	208	44096	19402	56%	0.3636	0.36	0.00990099
4	209	205	42845	27421	36%	0.495	0.45	0.090909091
5	123	119	14637	6001	59%	0.1404	0.108	0.230769231
6	145	141	20445	11654	43%	0.1962	0.18	0.082568807
7	156	152	23712	20867	12%	0.3888	0.36	0.074074074
8	234	230	53820	41441	23%	1.08	0.72	0.333333333
9	213	209	44517	22704	49%	0.612	0.36	0.411764706
10	209	205	42845	20994	51%	0.432	0.36	0.166666667
11	208	204	42432	14003	67%	0.351	0.27	0.230769231
12	238	234	55692	49009	12%	1.125	0.9	0.2
13	235	231	54285	42342	22%	0.8928	0.72	0.193548387
14	234	230	53820	35521	34%	0.6993	0.63	0.099099099
15	215	211	45365	24951	45%	0.855	0.45	0.473684211
16	219	215	47085	33430	29%	0.648	0.54	0.166666667
17	210	206	43260	26389	39%	0.81	0.45	0.444444444
18	221	217	47957	28774	40%	0.918	0.54	0.411764706
19	220	216	47520	23760	50%	0.54	0.45	0.166666667
20	230	226	51980	34307	34%	0.7182	0.63	0.122807018
21	240	236	56640	43613	23%	0.9477	0.81	0.145299145
22	258	254	65532	46528	29%	0.9639	0.81	0.159663866
23	78	74	5772	3867	33%	0.855	0.45	0.473684211
24	216	212	45792	25644	44%	0.792	0.45	0.431818182
25	86	82	7052	4654	34%	0.136935	0.0765	0.441340782
26	89	85	7565	4993	34%	0.083538	0.0702	0.159663866
27	98	94	9212	6817	26%	0.107856	0.0963	0.107142857
28	211	207	43677	22275	49%	0.5004	0.36	0.28057554
29	210	206	43260	21197	51%	0.6732	0.36	0.465240642
30	209	205	42845	21851	49%	0.6588	0.36	0.453551913
31	229	225	51525	34007	34%	0.8442	0.63	0.253731343
32	267	263	70221	54070	23%	1.1088	0.99	0.107142857
33	278	274	76172	41895	45%	0.79992	0.72	0.099909991
34	302	298	89996	35098	61%	0.693	0.63	0.090909091
35	311	307	95477	52512	45%	1.458	0.9	0.382716049
36	311	307	95477	60151	37%	1.8468	1.08	0.415204678
37	345	341	117645	85881	27%	2.7693	1.53	0.447513812
38	239	235	56165	39877	29%	1.116	0.72	0.35483871
39	125	121	15125	10436	31%	0.2214	0.18	0.18699187

#CXR	Shape (N)	Shape (n)	Size	Stdz_EF_size	Size reduction	Big(O)_org_size	Big(O)_stdz_size	Big(O) % redn
40	234	230	53820	29601	45%	0.6966	0.54	0.224806202
41	278	274	76172	47988	37%	1.2069	0.81	0.32885906
42	302	298	89996	65697	27%	1.989	1.17	0.411764706
43	311	307	95477	67789	29%	1.8135	1.17	0.35483871
44	311	307	95477	65879	31%	61.7022	37.17	0.397590361
45	345	341	117645	83528	29%	2.7081	1.53	0.435028249
46	239	235	56165	37631	33%	0.693	0.63	0.090909091
47	125	121	15125	8470	44%	0.1386	0.099	0.285714286
48	234	230	53820	35521	34%	0.756	0.63	0.166666667
49	98	94	9212	6080	34%	0.136359	0.0981	0.28057554
50	145	141	20445	15129	26%	0.4023	0.27	0.32885906
Avg.	215.26	211.56	50236	32083.00	37%	2.0208	1.2908	26%

Shape (N): pixel size  
Shape (n): variable size  
Stdz\_EF\_size: standardized Edge Feature (EF) pixel size  
Big(O)\_org\_size: Big(O) of the original size of the images  
Big(O)\_stdz\_size: Big(O) of the standardized size of the images, based on EF  
Big(O) % redn: % reduction of Big(O)  
Avg.: average