Artificial Intelligence Evolution
http://ojs.wiserpub.com/index.php/AIE/

UNIVERSAL WISER PUBLISHER

Research Article

A Novel Approach to Detect Abnormal Chest X-rays of COVID-19 Patients Using Image Processing and Deep Learning

Subhagata Chattopadhyay

Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be) University, Bengaluru 561203 Karnataka India
E-mail: schattop@gitam.edu

Received: 5 June 2021; **Revised:** 26 June 2021; **Accepted:** 7 July 2021

Abstract: The study proposes a novel approach to automate classifying Chest X-ray (CXR) images of COVID-19 positive patients. All acquired images have been pre-processed with Simple Median Filter (SMF) and Gaussian Filter (GF) with kernel size (5, 5). The better filter is then identified by comparing Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) of denoised images. Canny's edge detection has been applied to find the Region of Interest (ROI) on denoised images. Eigenvalues [-2, 2] of the Hessian matrix (5 × 5) of the ROIs are then extracted, which constitutes the 'input' dataset to the Feed Forward Neural Network (FFNN) classifier, developed in this study. Eighty percent of the data is used for training the said network after 10-fold cross-validation and the performance of the network is tested with the remaining 20% of the data. Finally, validation has been made on another set of 'raw' normal and abnormal CXRs. Precision, Recall, Accuracy, and Computational time complexity (Big(O)) of the classifier are then estimated to examine its performance.

***Keywords*:** COVID-19, simple median filter, Gaussian filter, Canny's edge detection, region of interest, Hessian matrix, eigenvalues, Feed-Forward Neural Network

1. Introduction

Biological vision is still superior to computerized vision due to its inherent robustness through millions of years of evolution and still evolving relentlessly in the era of plasma screens and related pixels. On one hand, researches are being conducted to understand the physiology of the biological vision, its cascading path, and the related system. On the other hand, attempts are being made to develop a computer vision system so that it can match with that of the biological vision system by automating the whole process of vision. The visual cortex is the area of the brain, situated at the back of the head, is the principal area for image processing, which is a complex structure responsible for image (i) acquisition, (ii) analysis (feature extraction, finding the edges, contours, and angles of the images, compression, cropping, denoising, and many more), (iii) facilitating registration, (iv) recall or identification, and finally (v) creating an overall perception around it by mingling metacognition in a lightning speed. While images are acquired through the eyes and the corresponding signals (pixel intensities, frequencies, and volume) are transmitted to the Visual cortex via the Optic nerve. Studies have identified that V1 cells in the visual path are the main cells to perceive orientation,

DOI: https://doi.org/10.37256/aie.222021977

contour, and edges of any given image [1]. Edge-detection, therefore, poses to be one of the most important image analysis methods, which is performed by visual cortex cells quite efficiently [2]. Hence, to improve computerized vision technology, edge-detection and in turn, the Region of Interest (ROI) identification of an image are presently two key research areas.

Biologically humans are vision-dominated creatures. Without visionary function, life seems to be non-existing. Through eyes, we can capture large images and their RGB color, texture, contour, and orientation and so on, in a much faster way. The visual cortex can differentiate among the images based on the mentioned features of the image. The brain can store large images, retrieve and analyze them based on their original shape and size much quickly, which is still not possible for computer vision to do so. Hence, to match with the speed of biological vision, image dimension reduction is a popular research area in computer vision [3]. There are several methods of dimensionality reduction. Popular methods are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). However, these methods have their limitations, such as information loss, especially in large image data [4]. Hessian matrix has been used for dimensionality reduction due to its various advantages over PCA and LDA [5]. Its Eigenvalues provide direction to the key edges to define the ROIs. Hence, these techniques are used for image enhancement [5].

Deep Learning (DL) in CXR analysis of COVID-19 is a current trend of Computer vision research [6]. Most of the studies are conducted with Convolutional Neural Network (CNN), originally introduced by Fukushima as 'neocognition' [7] for its obvious advantages, such as (i) image feature recognition and (ii) high precision and accuracy, though the computational complexity of CNN is quite high [8]. Neural network imitates biological NN in the brain and was first proposed by McCulloch and Pitts [9], then Rosenblatt [10], Werbos [11], and in turn by many others [12] as the process of its mathematical evolution and CNN is an extension of that evolution and focused towards computer vision. The key objective of the CNN is to imitate biological vision. NNs are popular classification (supervised learning) as well as clustering (unsupervised learning) techniques. CNN and ANN follow supervised learning, although these can also be modified for unsupervised or deep learning.

In a recent study, [13] used deep learning neural net for solving a multi-class issue of COVID-19 cases. The classes are divided into (i) normal X-ray, (ii) abnormal X-ray, and (iii) Pneumonia. The classifier has been developed on the DarkNet model using 'you only lok one time' object detection system. There are 17 convolutional layers for implementation and can assist radiologists in diagnosing CXRs of COVID-19 patients. The developed classifier is 98.8% accurate in classifying binary classes (i) and (ii). For multi-class, the accuracy is found to be 87.02%.

Deep CNN classifier, developed using Xcepian architecture and pre-trained with ImageNet datasets has been studied for detecting abnormal COVID-19 CXRs [14]. It is called CoroNet, which is found to be 89.6% accurate and 93% precise in detecting normal, abnormal, and pneumonitis.

CovXNet is another neural net-based classifier, developed using multi-dilation CNN for automatic COVID-19-included and other types of pneumonia. Transferable multi-receptor feature optimization method has been used for feature selection, based on which the classifier makes the diagnosis. The accuracy of CovXNet is found to be 97.4% for COVID vs. Normal, 96.9% for COVID vs. Viral pneumonia, 94.7% for COVID vs. Bacterial pneumonia, and 90.2% for multi-class COVID vs normal vs Viral vs. Bacterial pneumonia [15].

Artificial Neural Network (ANN) has its advantages, however, has not been used as wide as CNN for deep learning in COVID-19 image analysis due to the apparent superiority of CNN over ANN. This is the gap in the literature and opens up the scope of the present work. ANN's property of fast parallel processing, sequential and batch mode of training, and classification accuracy remain the key motivation behind its use.

The work aims to develop a fully connected Feed Forward Neural Network (FFNN) classifier using Eigenvalues of Hessian matrix as the training data sets of a set of COVID-19 positive CXR images, which are denoised and their ROIs are identified. The *objective* is to develop an ANN-based image classifier using a newly proposed hybrid method and examining its performance.

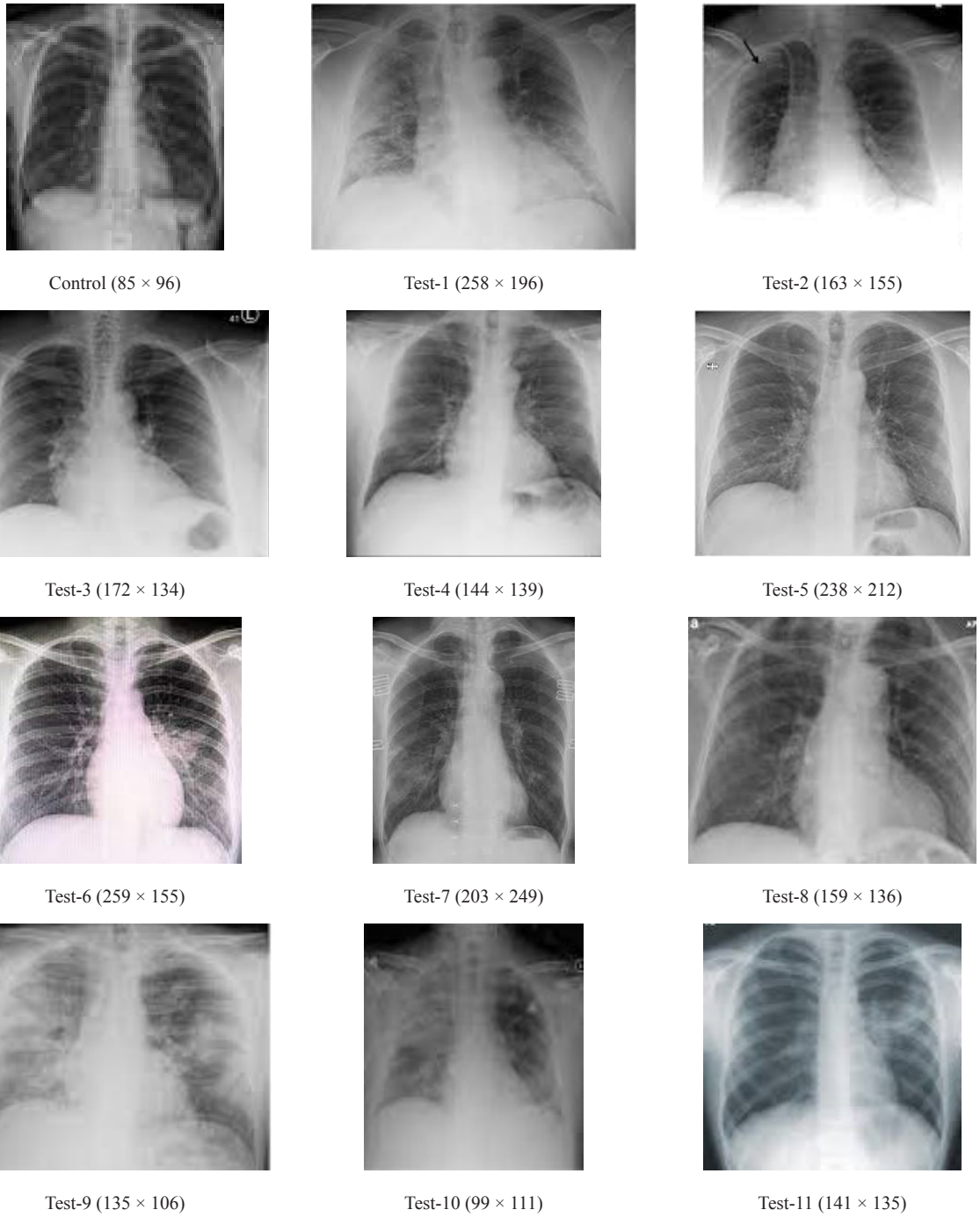
The importance of the work is to develop a classifier that should be able to classify normal and abnormal COVID-19 CXRs and contribute to assisting human doctors to decide on the treatment plan based on the extent of the lung involvement by the virus.

The rest of the paper is organized as follows. Section II describes the material and methods of the experiment. Section III shows the results and explains those. Section IV concludes the work and proposes future extensions.

2. Materials and method

This section discusses CXR image acquisition; the pre-processing and processing methods; and the development of the FFNN classifier in detail.

2.1 *Image acquisition*



Control (85 × 96) Test-1 (258 × 196) Test-2 (163 × 155)

Test-3 (172 × 134) Test-4 (144 × 139) Test-5 (238 × 212)

Test-6 (259 × 155) Test-7 (203 × 249) Test-8 (159 × 136)

Test-9 (135 × 106) Test-10 (99 × 111) Test-11 (141 × 135)

Figure 1. CXR images with their size

source: https://www.google.com/search?q=covid+chest+x+ray+images+copyright+free&tbm=isch&ved=2ahUKEwjv9OOooe7wAhUFcSsKHZEDAW4Q2-cCegQIABAA&oq=covid+chest+x+ray+images+copyright+free&gs_lcp=CgNpbWcQAzoCCAA6BggAEAgQHjoECAAQGFDPvwJY1-cCYKbuAmgAcAB4AYABsQKIAZAWkgEIMC4xMS4zLjGYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=v96xYK-ZDIXirQGRh4TwBg&bih=657&biw=1366

All CXR images are acquired from Google images. There is a total of 12 images-one normal (Control) and eleven (Test1-Test11), having abnormal findings are selected for training and testing the classifier. It is worth noting that ten new CXRs are taken for validation of the classifier. A sample of training images and their shapes are shown in Figure 1.

2.2 *System information*

Image data pre-processing, processing, and development of the proposed FFNN classifier are performed using Python 3.8.3 programing language with Spyder editor version 4.1.5, preloaded with *matplotlib*, *skimage*, *numpy*, *pandas*, *OpenCV*, and *sklearn*. All simulations were primarily run on Windows 10 Pro 64 bits OS × 64-based Processor Intel(R) Core TM @ 2.80 GHz.

2.3 *Pre-processing*

All RGB-colored images are gray-scaled at the outset of the study, followed by denoising. Images have noise within them as a natural phenomenon during capturing, processing and registering by a device. X-rays are no exception. Hence, all images are denoised using two popular filters, Simple Median Filter (SMF), which is a non-linear filter, and Gaussian Filter (GF), which is a linear filter and are referred by Equations (1) and (2) and (3), respectively. The kernel size is kept as (5, 5) for denoising. The Python package used is *OpenCV*.

For all pixels in a neighborhood of '*w*', the following equation can express the working principle of an SMF,

$$I(k,l) = median\{x(i,j),(I,j)\epsilon w\} \tag{1}$$

Where '$I(k, l)$' represents the image obtained after filtering; '$x(i, j)$' are the pixels with '(i, j)' coordinates; 'w' is the location, which is centered around location (k, l) of the image [16].

The formulae of Gaussian function in 1D and 2D are as follows:

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

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

$$G_{(x,y)} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

Where 'x' and 'y' represent horizontal and vertical axes respectively, '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. That's why this operation results in the blurring of the image with edge preservation in a much better way [17].

As mentioned before, it is desired that the images obtained by good filtering techniques should be less noisy. Hence, the performance of the individual filtering techniques is assessed by computing *Mean Squared Error (MSE)* and *Peak Signal-to-Noise Ratio (PSNR)*, which are two popular techniques to estimate noise level in a pre-processed image [18-19].

MSE-represents the cumulative squared error between the denoised and original images (refer to Equation (10)). Hence, lower values are appreciated.

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - K(i,j))^2 \quad (4)$$

Where 'm, n' represents image matrix '$I(I, j)$' (original image) and '$K(i, j)$' (denoised image).

PSNR-represents the peak error of an image and is expressed as a ratio of peak or maximum signal value and the power of the distorting noise effect. Therefore, higher values are appreciated.

$$PSNR = 20 \cdot \log_{10}(MAX / \sqrt{MSE}) \quad (5)$$

Where *MAX* is the peak/maximum value of any pixel and *MSE* is the Mean Squared Error among pixels of original and denoised images. As will be detailed later in Section 3.1, Table 1 shows the performance of the denoising techniques on original images by estimating MSE and PSNR.

After pre-processing (denoising), the images are processed for edge detection-based Region of Interest (ROI) identification and computing Eigenvalues of the Hessian matrix, which are explained below.

2.4 *Processing 1-Identifying ROI using Canny's edge detection*

Regions of Interest (ROIs) of a CXR consist of several anatomical structures, which are grossly overlapped on each other. Hence, it is a challenge to identify each of these distinctly for diagnosis. Common structures which describe an ROI have been shown in Figure 2.

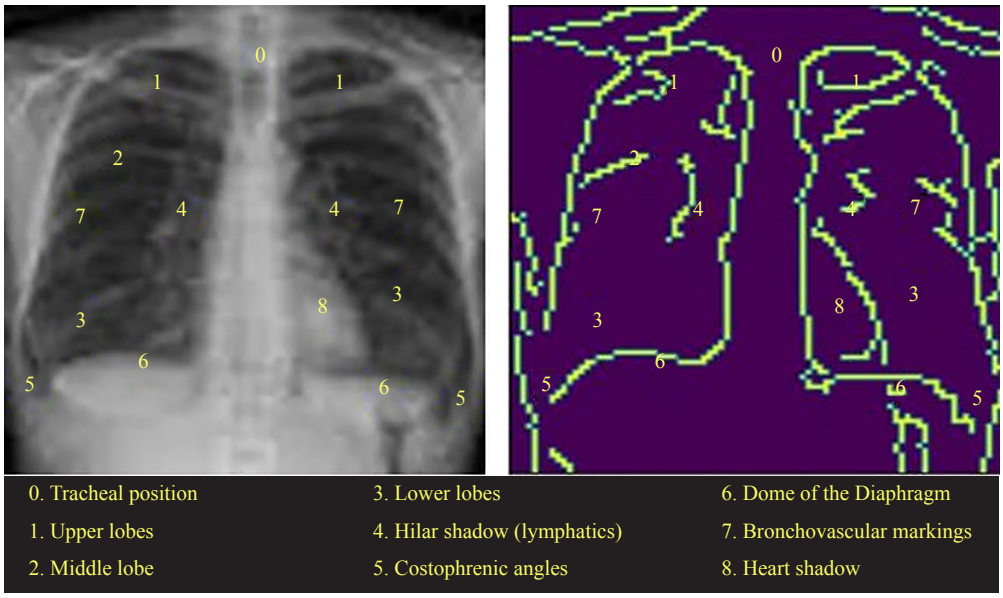


Figure 2. ROI of a CXR

From the figure, it is evident that edges of individual structures need clear demarcations to understand any pathology therein. For 'abnormal' CXR images, the edges are found to be obliterated or distorted. Although the human brain processes the edges efficiently, yet, there could be diagnostic fallacies and biases to decide on which one is 'abnormal' or 'normal'. Hence, computer vision can be assisting to clear such human bias or error. An edge detection operator that applies a multi-stage algorithm to detect a wide range of edges in an image is therefore opens a wide research scope. In this work, Canny's multistage edge detector algorithm has been used [20]. The algorithm runs in five intertwined steps, such as (i) smoothening or denoising by using GF, (ii) finding intensity gradient of the image,

(iii) non-maximum suppression to avoid spurious edges, (iv) double thresholding to get potential edges, and (v) edge tracking based on strong edges and removing weaker edges by the process called Hytsteresis [21]. The Python package used is *OpenCV*.

2.5 *Processing 2-Identifying Eigenvalues of Hessian matrix of the ROIs*

The Hessian matrix describes the 2nd order local image intensity variations around the selected voxel or volumetric pixel [22]. Eigenvector decomposition extracts an orthonormal coordinate system, which is aligned with the second-order structure of the image to obtain the said matrix. It is important to note that, (i) having the Eigenvalues, (ii) knowing the assumed model of the structure under investigation, and the (iii) resulting theoretical behavior of the Eigenvalues, the decision can be made if the analyzed voxel belongs to the structure that is under search [22]. In this work, to obtain the Hessian matrix Eigenvalues, the *SkImage* package is used. As a method of dimensionality reduction, a (5 × 5) Hessian matrix has been created taking each ROI. Eigenvalues vary between -2 to 2 for the ease of implementation and are essentially the non-zero values in the matrix. Hessian matrix is a symmetric matrix and hence to prevent redundant data use, out of 5 × 5 = 25 values, (5 + 4 + 3 + 2 + 1 = 15) values are considered as the inputs to the FFNN. The resultant matrix is then converted into a 1D array for each ROI. For twelve images, therefore, there are twelve ROIs and hence, twelve Hessian matrices and the Eigenvalues of each are represented by the respective 1D array. Each of the arrays has been assigned a class label of '0' for 'normal' CXR and '1' for abnormal for the ease of classification. Now, each of these 2D arrays (1D array of Eigenvalues plus corresponding class label) constitutes one set of training data. Therefore, there are twelve 2D training datasets finally, each obtained for each of the twelve images and in turn, are fed into the Feed-Forward Neural Network (FFNN) for class prediction of COVID-19 CXRs.

2.6 *Classifier development-Feed-Forward Neural Network (FFNN)*

The fully connected type of Feed-Forward Neural Network (FFNN) consists of one Input, Hidden, and Output layer, each. In this work, the detailed structure of the FFNN consists of fifteen nodes in the Input layer, which is equal to the size of the half-Hessian matrix of 15 values of each image containing 'zero' and 'non-zero' values (Eigenvalues); there is a single node in the Output layer, where the final output is calculated; the number of nodes in the Hidden layer (the layer between Input and Output layer) is kept as ten, as studies have shown that the number of nodes in the Hidden layer could be anything between the numbers of nodes in the Output and Input layer [23]. The connecting weights (W) between "Input-to-Hidden" and "Hidden-to-Output" nodes are set as random float values between -1 and 1. It is worth noting that, Input node must receive input value ≠ 0 (the Eigenvalue) for activation. The transfer function of Input nodes is kept as 'linear' (LTF). The transfer functions of the Hidden nodes are 'log sigmoidal' (STF), while the transfer function of the Output node is kept as 'linear' (LTF) for the simplicity of the model. The input signals move from Input to Output layer through the Hidden layer and the Input signals are nothing but the 2D array of Eigenvalues of the Hessian matrices obtained from the ROIs, identified by Canny's edge detection of an image, as mentioned earlier. All input values are processed by applying the principle of 'sequential' processing. The *Sklearn* package has been used to develop the FFNN classifier. A detailed parametric study has been conducted before finding the threshold level (see Appendix), which decodes on 'normal' and 'abnormal' CXR images after the final output is computed and matched with it. It has been observed that abnormal CXRs show a trend in having high computed output values through Output node, while lower values are obtained for normal' CXRs. Therefore, the 'threshold for classification' has been set as the mid-point, i.e., 0.5 (see Appendix). Output value ≤ 0.5 results into 'Normal' CXR and 'Abnormal' otherwise. The constructed FFNN has been trained with 80% of the Eigenvalue data of ROIs after 10-fold cross-validation and tested with the remaining 20% of the data. Validation of the model is finally performed with another set of five 'normal' and 'abnormal' CXRs, each. Figure 3 shows the "15-10-1" architecture of the FFNN. Epochs for training are varied from 100 to 1000 and the prediction error is estimated through Mean Square of the difference between the calculated output and the actual output. Output calculation of the FFNN is generically presented as below:

Step-1 Input to Input Nodes (IIN): If and only if the input values to the Input nodes are ≠ 0, each node is activated, else remain inactive.

Step-2 Output of Input nodes (OIN): it is the same as IIN due to the LTF operation.

Step-3 Multiplication of OIN with the connectors' weight values (W_{ij}): OIN (where '*i*' varies from 1 to 15) is

multiplied with the weight values of their respective connector with the Hidden node (where '*j*' varies from 1 to 10)

Step-4 Input to the Hidden node (IHN): summation of all multiplied values (OIN × W_{ij}).

Step-5 Output of the Hidden node (OHN): summated IHN values are processed with STF operation, shown in Equation (6). It gives the OHN value.

$$STF = \frac{1}{1+e^{-\sum IHN}} \tag{6}$$

Step-6 Multiplication of OHN with the connectors' weight values (W_{jk}): OIN (where '*j*' varies from 1 to 10) is multiplied with the weight values of their respective connector with the Output node (where '*k*' = 1).

Step-7 Input to the Output node (ION): summation of all multiplied values (OHN × W_{jk}).

Step-8 Output of the Output node (OON): same as ION due to the set LTF.

Step-9 Checking by threshold: OON ≤ 0.5 or > 0.5.

Step-10 Training the classifier: this is done with 80% of the input data constituted by the Eigenvalues of Hessian matrix after 10-fold cross-validation to prevent data over-fitting and examine the generalization of the predictive model [24-25].

Step-11 Classification: OON values > 0.5 might be the 'abnormal' CXRs, else 'normal'.

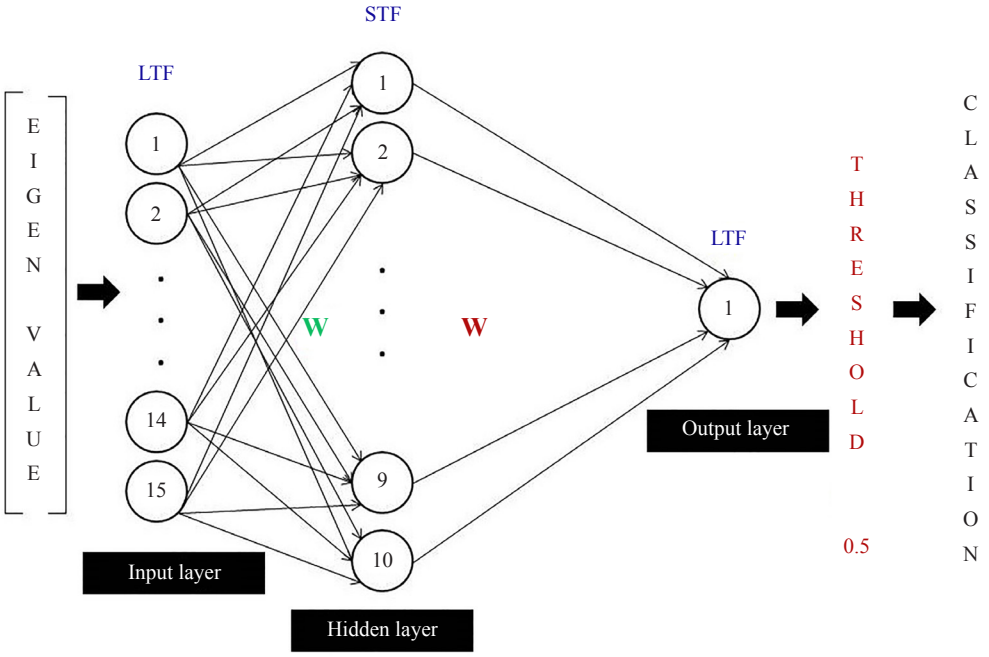


Figure 3. The architecture of the FFNN

Step-12 Computing Precision (*P*), *Recall (R) and Accuracy* (*A*) of the classifier using Equations (7), (8) and (9), respectively, where '*TP*', '*FP*', '*TN*', and '*FN*' are True Positive, False Positive, True Negative, and False Negative, respectively.

$$P = \frac{TP}{TP+FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

In the following section, results obtained through experiments are shown and discussed.

Step-13 Computing time complexity (Big(O)) of the classifier per epoch size: It is calculated as the number of pixels viewed for THE number of times. In all images, each pixel (n) is viewed two times-vertically and horizontally [26]. Hence, the computational complexity would be

$$Big(O) = (2n) \tag{10}$$

A flow diagram of all steps in the present study can be found in Figure 4.

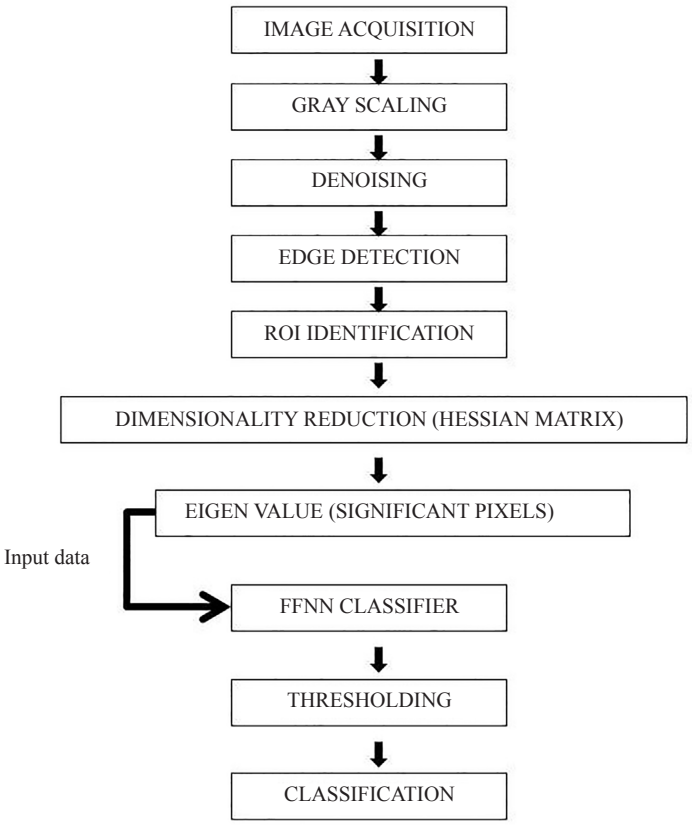


Figure 4. Flow diagram of the study

In the next section, the results of the experiments are shown and discussed.

3. Results and discussions

In this section the results of the following operations have been shown and explained:

I. Pre-processing: image denoising/filtering with SMF and GF.

II. Processing 1: ROI identification of each image with Canny's edge detection.

III. Processing 2: obtaining Hessian matrix of the ROIs for dimensionality reduction and computing its Eigenvalues that form the input training and testing datasets for each image.

IV. Feed-Forward Neural Network (FFNN) construction: a threshold-based CXR Classification, testing, validation, and time complexity (Big(O)) using Equation (10) estimation to examine classifier's performance.

3.1 *Preprocessing*

Noise removal is an important step of image processing. In this work, SMF and GF are used to denoise all images used for training the classifiers. The quality of denoising has been estimated through MSE and PSNR. Figure 5 shows the denoised images and the respective MSE as well as PSNR values to compare which one is a better denoising method for these CXR images could be found in Table 1.

Below, Table 1 shows the denoising performances of SMF and GF, respectively as estimated by MSE and PSNR values.

From Table 1, it is evident that the and average and standard deviation of MSE and PSNR values of SMF and GF are very close to each other; however, SMF shows lower MSE values and higher PSNR values, which are desired quality of a good denoising method. Hence, CXR images, denoised with SMF are considered for ROI selection using Canny's algorithm, where denoising is usually done by Gaussian technique. It is also important to mention here that a study has shown that the X-ray images are prone to Poisson type of noise and SMF works better on such type of noise, compared to GF [27].

Table 1. MSE and PSNR values of SMF and GF

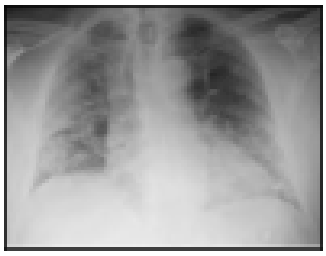
	SMF		GF	
CXR	MSE	PSNR	MSE	PSNR
Control	28	30.21	29.83	32.04
Test1	7.88	37.9	7.54	31.77
Test2	11.77	34.27	10.86	28.66
Test3	16.2	30.92	16.92	26.11
Test4	15	35.14	16.11	34.67
Test5	15.89	33.55	15.51	30.51
Test6	45.07	20.68	45.98	27.71
Test7	12.1	34.84	14.78	36.83
Test8	18.46	32.96	13.68	35.05
Test9	15.46	33.12	18.91	29.45
Test10	11.91	35.76	11.72	36.74
Test11	11.96	35.34	10.12	37.37
Average	17.475	32.89083	17.66333	32.2425
Standard deviation	10.01108	4.377107	10.54941	3.854018

Control Original

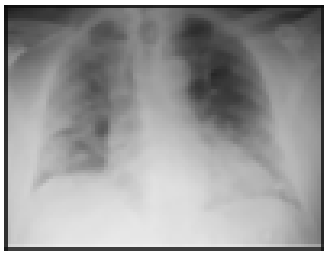
Median Filter

Gaussian Filter

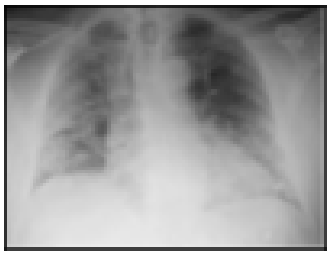
Test-1 Original



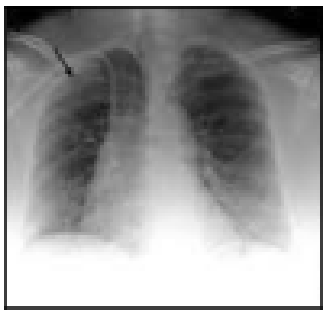
Median Filter



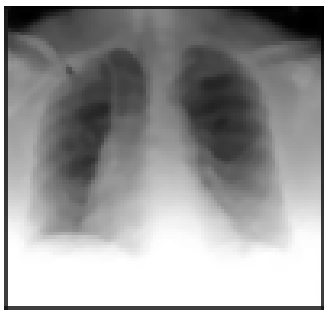
Gaussian Filter



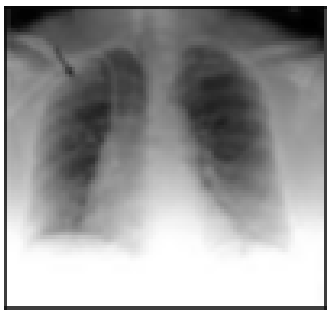
Test-2 Original



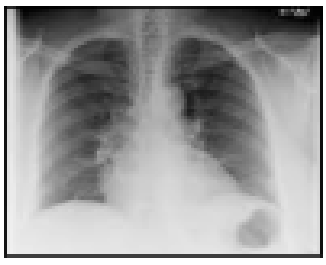
Median Filter



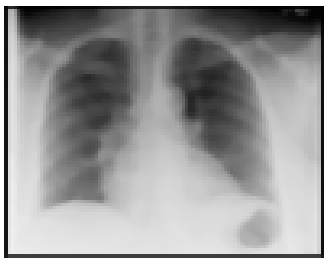
Gaussian Filter



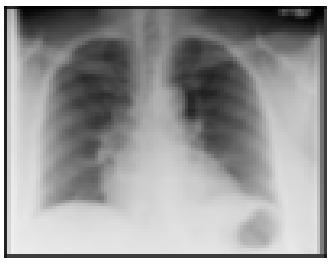
Test-3 Original



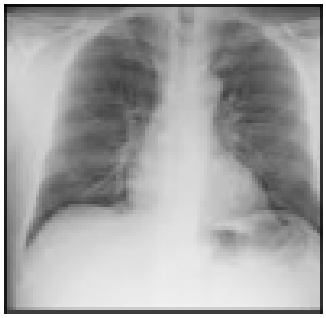
Median Filter



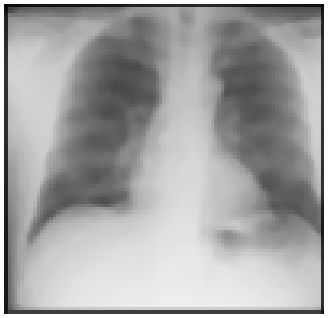
Gaussian Filter



Test-4 Original

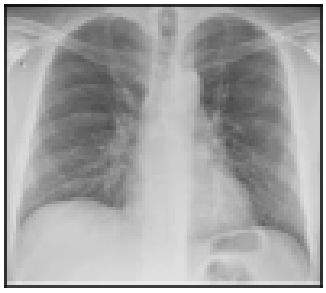


Median Filter

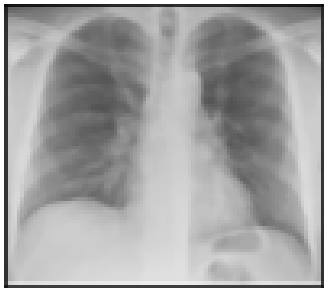


Gaussian Filter

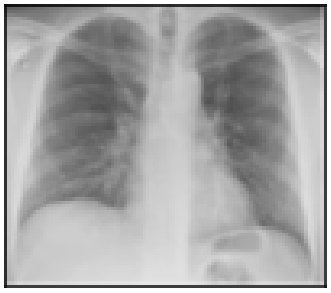
Test-5 Original



Median Filter



Gaussian Filter

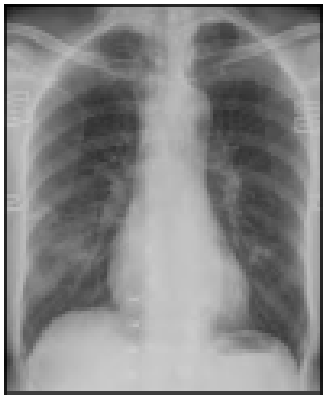


Test-6 Original

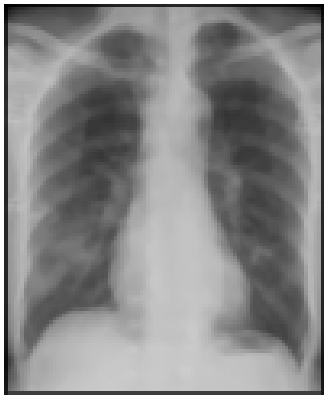
Median Filter

Gaussian Filter

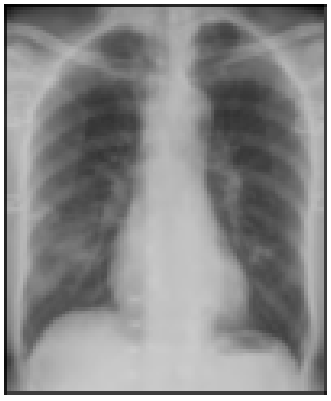
Test-7 Original



Median Filter



Gaussian Filter



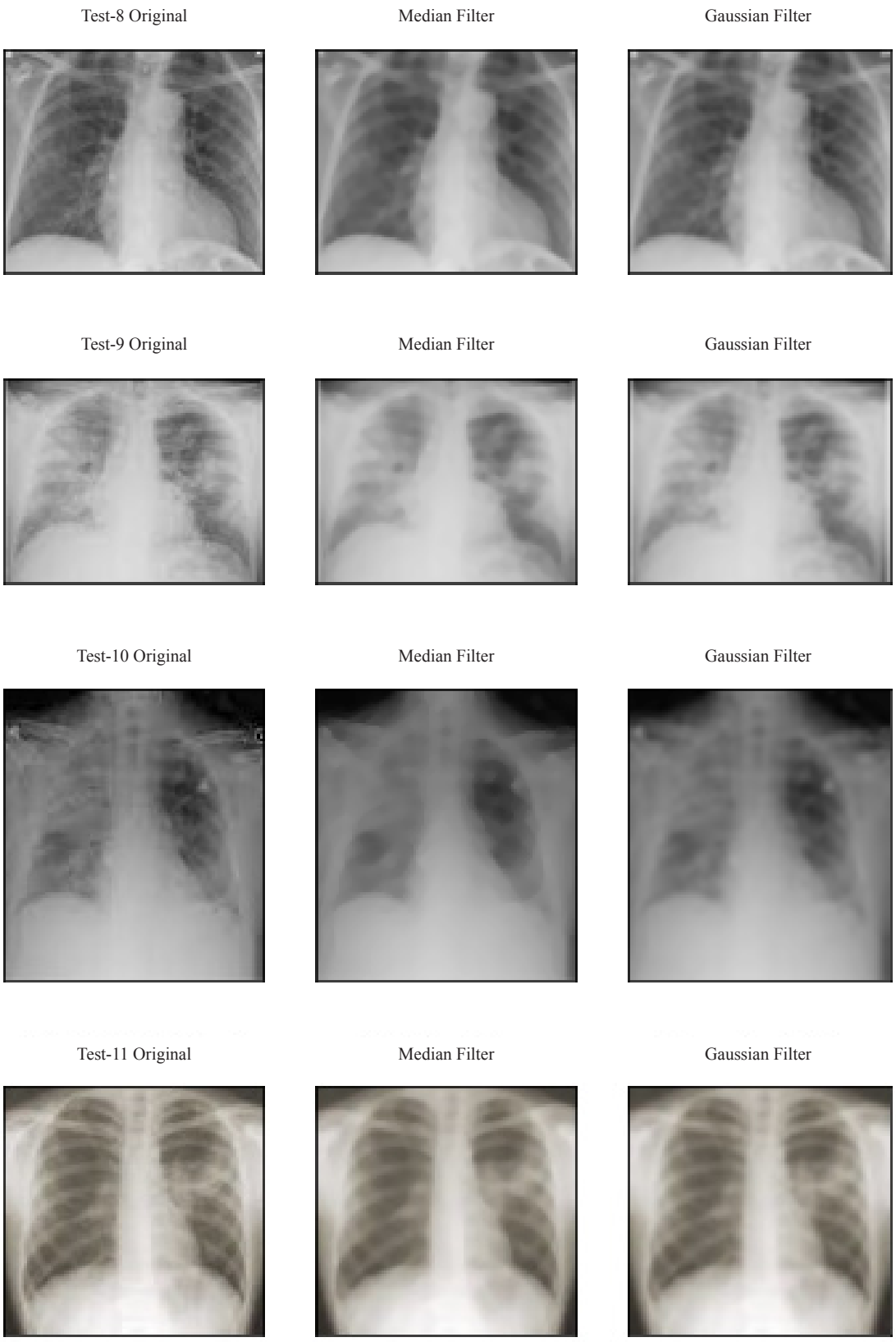


Figure 5. Denoising by SMF and GF and their respective MSE and PSNR values

3.2 *Processing 1-ROI identification*

Following noise removal, Canny's multistage algorithm has been applied for edge detection of each training CXR image. The edges thus detected are shown in Figure 6.

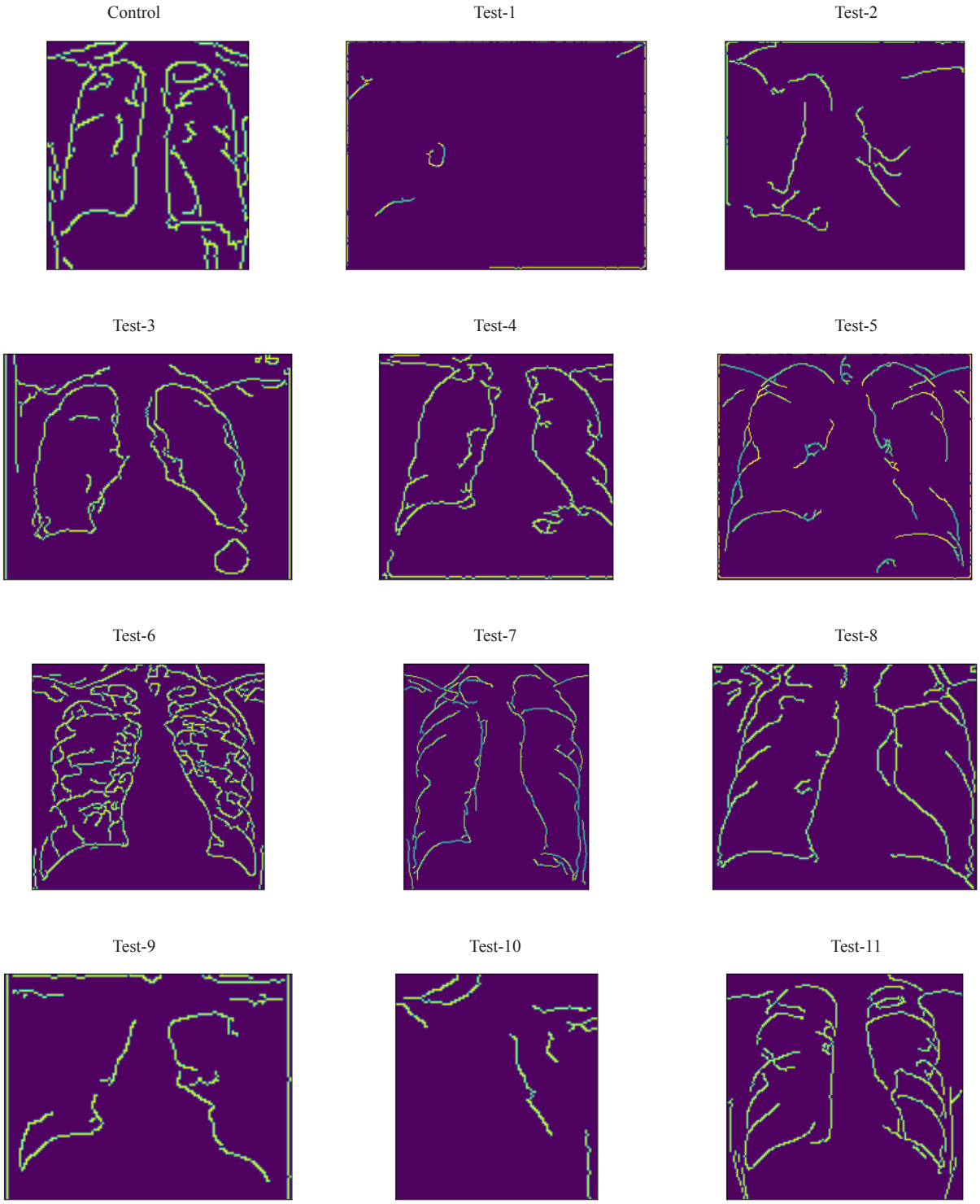


Figure 6. ROI computation by Canny's edge-detection algorithm

The edge-based demarcations of ROIs have already been explained in Figure 2. In Figure 6, in the 'Control' image, all anatomical regions have been clearly noted through edges. On the other hand, all Test images show distorted and also obliterated edges. Hence, the Test images are abnormal, while the Control is a normal CXR. The edges are therefore representative of the significant pixels, which differentiate 'abnormal' from 'normal' CXRs. It is worth noting that Canny-based image processing is computationally complex due to its multi-stage algorithm. Further smoothing required at the spatial domain reduces the processing speed further. Therefore, instead of smoothening, anatomical ROIs are only considered and their corresponding Hessian matrices are computed to find the Eigenvalues, which reduces the dimension considerably and increases the pace of the FFNN-based classification task. Hessian matrices of each ROI exploit the virtue of this pixel information and its Eigenvalues are able to represent the underlying pixel data in the ROIs. Results of the Hessian matrix and its Eigenvalues are shown in the next sub-section.

3.3 *Processing 2-finding eigenvalues of Hessian matrix*

As mentioned in section II, Eigenvalues of Hessian matrices (5 × 5) of all ROIs of the corresponding images have been computed. Hessian matrix is a symmetric matrix and can be used for dimensionality reduction. Therefore, to avoid redundancy further, out of total 25 values that lies between -2 and 2, (5 + 4 + 3 + 2 + 1) = 15 values are considered (yellow shaded), within which Eigenvalues are essentially non-zero values (red font). One sample of Eigenvalues in the matrix has been shown in Table 2.

Table 2. Eigenvalues are shown inside the yellow-colored cells

	0	1	2	3	4
0	0	0	2	0	0
1	0	1	0	1	0
2	2	0	-2	0	2
3	0	1	0	1	0
4	0	0	2	0	0

Above Table 2 showcases the Eigenvalues of the Hessian matrix of ROI of 'Control' CXR. Similarly, Eigenvalues (non-zero values) of the corresponding Hessian matrices of the ROIs of other CXRs are computed. The multidimensional matrix is then converted to a 2D matrix (values and class labels as '0' and '1.0' for normal and abnormal CXRs) to create the training inputs of the FFNN classifier.

3.4 *FFNN Classifier performance*

Eigenvalues form the training data (80% of input data) of the fully connected FFNN classifier. The performance of the FFNN classifier thus developed, has been first tested (using 20% of input data) and then validated (with ten 'raw' images) and the results are seen in Table 3 and Table 4, respectively.

As mentioned before, training of the classifier is performed using the Eigenvalues as these non-zero values represent the significant pixels of the ROIs of each CXR image. Epoch numbers are varied from 100 to 1000 and the error (MSE of classification) has been computed. Figure 7 shows the MSE vs. Epoch size parametric study. It can be noted that after the Epoch size of 800, there is no further improvement of MSE, and hence testing and validation of the FFNN classifiers are done with 800 Epoch size. Training and testing accuracy is found to be 100%. Computational time complexity has been estimated using Equation (10). Figure 8 shows the linear distribution of time complexity Vs. Epoch size.

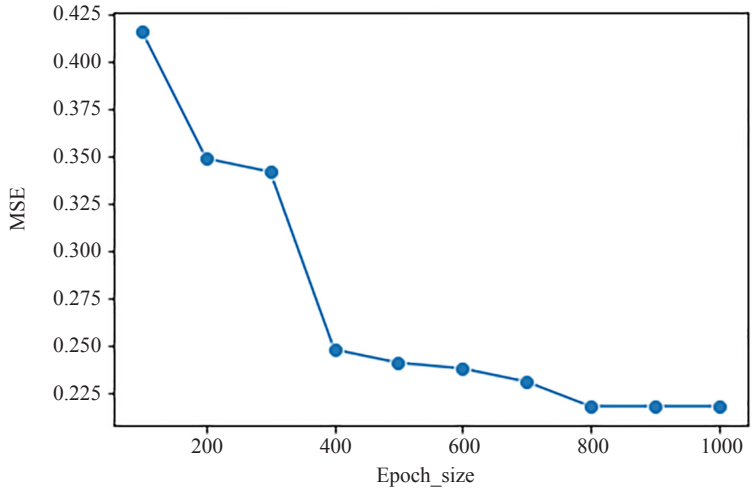


Figure 7. Epoch size vs. MSE of classification plots

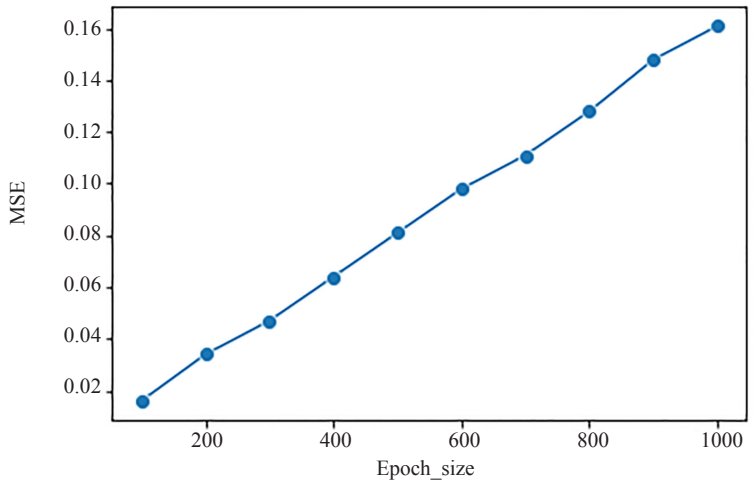


Figure 8. Epoch size vs. Time complexity plots

Table 3. Performance of the classifier on test cases

Test case	Pred_Out	Targ_Out	Threshold	Class_ass	Act_class
1	0.023	0		Normal	Normal
2	0.74	1	0.5	Abnormal	Abnormal
3	0.9	1		Abnormal	Abnormal

The accuracy of the test cases (20% of the data) is 100%. In the above table, 'Pred_Out', 'Targ_Out', 'Class_ass', and 'Act_class' refer to 'Predicted Output', 'Target Output', 'Class assigned', and 'Actual class', respectively. Next, in Table 4, validation results can be viewed.

Table 4. Performance of the classifier on validation cases

Validation case	Pred_Out	Targ_Out	Threshold	Class_ass	Act_class
1	0.013	0	0.5	Normal	Normal
2	0.64	1		Abnormal	Abnormal
3	0.76	1		Abnormal	Abnormal
4	0.52	0		Abnormal	Normal
5	0.86	1		Abnormal	Abnormal
6	0.46	1		Normal	Abnormal
7	0.98	0		Abnormal	Normal
8	0.91	0		Abnormal	Normal
9	0.12	0		Normal	Normal
10	0.99	1		Abnormal	Abnormal

From this table, Precision (P), Recall (R), and Accuracy (A) are computed using Equations (7), (8), and (9), respectively. The Accuracy is found to be 60%, while Precision and Recall are 70% and 40%, respectively. It is important to mention here that, the validation images are pre-classified by radiologists having clinical diagnostic accuracy of 71% [28]. While compared to this data, the developed classifier can interpret 'raw' images with quite an appreciable accuracy and precision.

4. Conclusions and future work

This paper is an attempt to develop an image classifier by using Canny's edge-detector algorithm that provides the information of ROIs of the images. COVID-19, being still active as a pandemic and it principally affects the lungs and its accessories in the respiratory system, CXRs, which are economic, compared to Computerized Tomography (CT) and Magnetic Resonance Imaging (MRI) and hence are the mainstay for studying lung pathologies. The dimensionality of each ROI is reduced by computing the Hessian matrix and in turn, represented by the Eigenvalues of the said matrix. An FFNN classifier has been constructed, which is trained with the Eigenvalues. The performance of the classifier has been examined by measuring its Precision, Recall, Accuracy, and time complexity. Results show that the classifier, at its juvenile state, is able to predict normal and abnormal CXRs with 70% precision and 60% accuracy, which is an encouraging finding. The time complexity shows a linear relationship with the epoch size.

Contribution of the study:

i. Novel use of Hessian matrix and its Eigenvalues for dimensionality reduction and creating training data, respectively and

ii. Efficient use of Canny's edge-based ROI detection using SMF instead of GF in COVID-19 CXRs

Limitations and future scope of the study:

i. Accuracy can be further increased with more training and use of backpropagation, and

ii. A hard threshold may not be applicable for the larger volume of CXRs and needs further study.

References

[1] Poo M, Du J, Ip NY. China brain project: basic neuroscience, brain diseases, and brain-inspired computing. *Journal of Neuron*. 2016; 92(3): 591-96.

[2] Zhu MM, Xu YL, Ma HQ. Edge detection based on the characteristic of primary visual cortex cells. *IOP Conf. Series: Journal of Physics: Conf. Series*. 2018; 960: 012052.
[3] Shereena VB, David JM. Significance of dimensionality reduction in image processing. *Signal & Image Processing An International Journal*. 2015; 6(3): 27-42.
[4] Ng SC. Principal component analysis to reduce dimension on digital image. *Procedia Computer Science*. 2017; 111: 113-119. Available from: https://www.sciencedirect.com/science/article/pii/S1877050917311900.
[5] Yin X, Ng BWH, He J, Zhang Y, Abbott D. Accurate image analysis of the retina using hessian matrix and binarisation of thresholded entropy with application of texture mapping. *PLOS One*. 2014; 9(4): e95943. Available from: https://doi.org/10.1371/journal.pone.0095943.
[6] 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. *Scientific Reports*. 2020; 10(1): 19549. Available from: https://doi.org/10.1038/s41598-020-76550-z.
[7] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*. 1980; 36(4): 193-202. Available from: https://doi.org/10.1007/BF00344251.
[8] Apostolopoulos ID, Mpesiana TA. Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*. 2020; 43(2): 635-640. Available from: https://doi.org/10.1007/s13246-020-00865-4.
[9] McCulloch WA, Pitts W. A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*. 1943; 5: 115-137. Available from: https://jontalle.web.engr.illinois.edu/uploads/498-NS.21/McCulloch-Pitts-1943-neural-networks-ocr.pdf.
[10] Rosenblatt F. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. New York: Spartan; 1962. Available from: https://www.scirp.org/reference/ReferencesPapers.aspx?ReferenceID=1361084.
[11] Werbos PJ. *Beyond Regression: New tools for prediction and analysis in the behavioral sciences*. Doctoral Dissertation, Harvard University, Cambridge. 1974. Available from: https://www.scirp.org/reference/ReferencesPapers.aspx?ReferenceID=2318450.
[12] Rumelhart DE, Hinton GE, Williams GE. Learning internal representations by error propagation. *Parallel Distributed Processing*. Cambridge, MA.: MIT Press; 1986.
[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. *Computer in Biology and Medicine*. 2020; 121: 103792. Available from: https://doi.org/10.1016/j.compbiomed.2020.103792.
[14] Khan AI, Shah JL, Bhat MM. CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. *Computer Methods and Programs in Biomedicine*. 2020; 196: 105581. Available from: https://doi.org/10.1016/j.cmpb.2020.105581.
[15] 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. *Computers in Biology and Medicine*. 2020; 122: 103869. Available from: https://doi.org/10.1016/j.compbiomed.2020.103869.
[16] Zhu R, Wang Y. Application of improved median filter on image processing. *Journal of Computers*. 2012; 7(4): 838-841.
[17] Haddad RA, Akansu AN. A class of fast Gaussian binomial filters for speech and image processing. *IEEE Transactions on Signal Processing*. 1991; 39(3): 723-727. Available from: https://ieeexplore.ieee.org/document/80892/.
[18] Dabas P, Mehra R. Estimation of the image quality under different distortions. *International Journal of Advanced Trends in Computer Science and Engineering*. 2016; 8(7): 17291-17296.
[19] Jemila SJ, Therese AB. Selection of suitable segmentation technique based on image quality metrics. *The Imaging Science Journal*. 2019; 67(8): 475-480. Available from: https://www.tandfonline.com/doi/abs/10.1080/13682199.2020.1718298.
[20] Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986; 8(6): 679-698.
[21] Dhivya R, Prakash R. Ear authentication using edge detection and feature extraction. *Advances in Natural and Applied Sciences*. 2016; 10(17): 89-94.
[22] lennon310. https://stackoverflow.com. 2014. Available from: https://stackoverflow.com/questions/22378360/hessian-matrix-of-the-image/22380416. [Accessed 31th May 2021].
[23] Thomas AJ, Petridis M, Walters SD, Morgan R. On predicting the optimal number of hidden nodes. *The 2015*

International Conference on Computational Science and Computational Intelligence (CSCI'15). Monte Carlo Resort Las Vegas, USA: IEEE; 2015. p. 565-570.
[24] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: data mining, inference, and prediction. *Springer Series in Statistics*. NewYork/Berlin/Heidelberg: Springer; 2008. Available from: https://doi.org/10.1111/j.1541-0420.2010.01516.x.
[25] Duda RO, Hart PE, Stork DG. *Pattern classification*. John Wiley & Sons; 2001. Available from: https://www.scirp.org/reference/ReferencesPapers.aspx?ReferenceID=890533.
[26] Schellekens J. https://stackoverflow.com/. 2019. Available from: https://stackoverflow.com/questions/55438638/how-to-find-computation-complexity-of-image-processing-algorithm#:~:text=As%20a%20rough%20measure%2C%20you,viewed%2Fedited%20by%20the%20algorithm.&text=Its%20complexity%20is%20O(n,pixels%20in%20the%20input%20im. [Accessed 17th April 2021].
[27] Thakur K, Kadam J, Sapkal A. Poisson noise reduction from X-ray images by region classification and response median filtering. *Sadhana*. 2017; 42(6): 855-863.
[28] Richens JG, Lee CM, Johri S. Improving the accuracy of medical diagnosis with causal machine learning. *Nature Communications*. 2020; 11(1): 3923. Available from: https://doi.org/10.1038/s41467-020-17419-7.

APPENDIX

Sample of Output calculation (OON) for 100 CXR images to determine the threshold:

Image No.	Image type	OON
1	Normal	0.38
2	Abnormal	0.76
3	Normal	0.43
4	Normal	0.45
5	Normal	0.21
6	Normal	0.19
7	Abnormal	0.57
8	Abnormal	0.65
9	Abnormal	0.51
10	Abnormal	0.89
..		
91	Normal	0.43
92	Abnormal	0.87
93	Abnormal	0.79
94	Abnormal	0.63
95	Abnormal	0.65
96	Abnormal	0.98
97	Normal	0.34
98	Abnormal	0.87
99	Normal	0.35
100	Abnormal	0.63