Eksplorium *p*-ISSN 0854-1418
Volume 46 No. 1, May 2025: 1376–1398 *e*-ISSN 2503-426X

An Enhanced Diagnostic and Classification Process of Covid-19 Chest X-Ray Images Using Ensemble Convolutional Neural Network (ECNN)

N. Sundaravalli[1], Dr. R. Vidyabanu[2]

[1]Research Scholar, PG & Research Department Of Computer Science,

L.R.G Govt. Arts College For Women, Tirupur – 4.

shinisuba99@yahoo.com

[2]Assistant Professor, PG & Research Department Of Computer Science,

L.R.G Govt. Arts College For Women, Tirupur-4.

vidhyabanu@yahoo.com

Article Received: 22 Feb 2025, Revised: 24 April 2025, Accepted: 04 May 2025

Abstract*:* Coronavirus disease 2019 (COVID-19) is an epidemic disease caused by the Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) virus and became a pandemic disease in 2020. Since the arrival of this health crisis, accurate prediction and efficient diagnosis of COVID-19 remains a significant challenge to medical experts due to the limitations of current detection methods, such as blood tests and chest scans, which can be cost, time-consuming and error-prone. In this context, fast and accessible diagnostic technical tools are needed. During this period, the medical Chest X-ray (CXR) image plays an important role to diagnose COVID-19 patients effectively. This current research work takes the COVID-19 affect chest X -ray and proposes a new method of image de-noising based on using Discrete Wavelet Transform (DWT) and Total Variation Regularization (TVR). Features are extracted using VGG16, InceptionV3, DenseNet and MobileNet. Optimization is performed using Grasshopper Optimization Algorithm (GOA) and Enhanced GOA (EGOA). DL-based approaches like Convolutional Neural Network (CNN) and Ensemble CNN (ECNN) are used for classifying Covid-19, normal or healthy and Pneumonia cases from X-ray images. From the outcomes, it is obvious that the proposed mechanism offers improved Precision, Accuracy, F1-Score and Recall. (ECNN) are used for classifying Covid-19, normal or healthy and Pneumonia cases from X-ray images. From the outcomes, it is obvious that the proposed mechanism offers improved Precision, Accuracy, F1-Score and Recall.

Keywords: Convolutional Neural Network (CNN), Discrete Wavelet Transform (DWT), Total Variation Regularization (TVR), Grasshopper Optimization Algorithm (GOA), VGG16, InceptionV3, DenseNet and MobileNet, Ensemble CNN , Covid-19, X-ray image.

1. INTRODUCTION

Global epidemic of Covid-19 was a major threat to health throughput the world. Detection of Covid-19 is a challenge as there is a rapid spread of disease. There were about 1.6 million cases diagnosed with Covid-19 which increased exponentially. In this paper, Covid-19 cases are determined using Deep Learning (DL)-based schemes by observing chest X-ray images.

In DL, a branch of Machine Learning (ML), system aims at extracting deep features using hierarchical algorithm. DL may be used effectively in processing image and video, pattern recognition as well as Natural Language Processing (NLP). Amid present DL algorithms, Convolutional Neural Network (CNN) uses 2D data and is appropriate for image as well as video processing. AlexNet is a basic CNN algorithm [1]. It includes layers like convolutional, pooling and Fully-Connected (FC) layers. Convolution layer includes several filters with learnable co-efficients. Feature map is formed by convolving every filter with image. Subsequently, pooling layer is employed for reducing feature map dimension. Following convolution and pooling layers, FC layers are involved in transforming 2D feature maps to feature vector. Classifier is used for categorizing input features. In learning step, co-efficients as well as factors of filters are updated to get best values. The system extracts suitable features from image given as input.

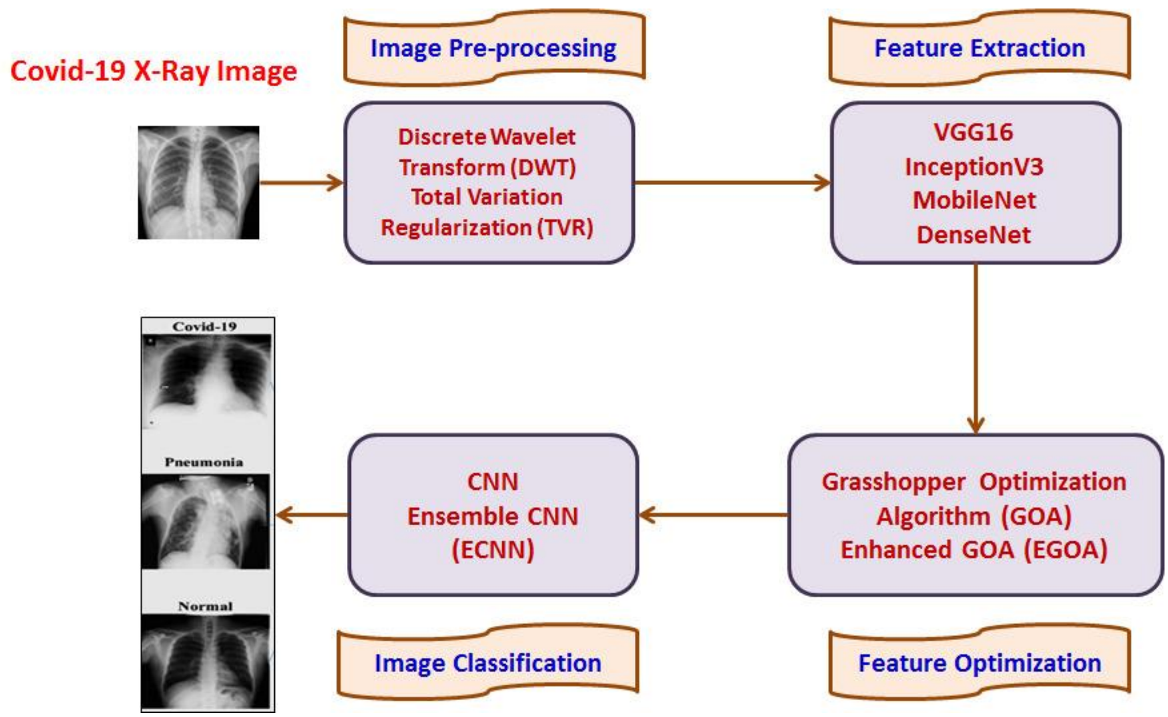

Figure 1: Proposed Architecture

The proposed architecture is shown in Figure 1. The input Covid-19 X-Ray image is preprocessed using DWT and TVR. Features like color, shape, edges, texture, mean and standard deviation are extracted using VGG16, InceptionV3, DenseNet and MobileNet. The combined features are optimized using GOA and Enhanced GOA (EGOA). Finally, classification is performed using CNN and Ensemble CNN (ECNN).

1.1 The Present work

The current paper focuses on image de-noising of COVID-19-affected chest X-rays and proposed Deep Learning (DL) and Machine Learning or a hybrid-based approach are employed for feature extraction and classification process. The remaining sections of this work are well organized as follows, Section II contains, a review related work of this study, Section III is concerned with the images are pre-processed using Total Variation Regularization (TVR) and Discrete Wavelet Transform (DWT). Section IV resolves the features extracted using VGG16, InceptionV3, DenseNet and MobileNet. Section V Optimization is performed using Grasshopper Optimization Algorithm (GOA) and Enhanced GOA (EGOA). Section VI expounds, medical images are classified using DL-based approaches like Convolutional Neural Network (CNN) and Ensemble CNN (ECNN). Section VI Interpret, Experimental Result and finally, Section VII concludes the study.

2. RELATED WORK

In this section, works done by various authors related to pre-processing of images, feature extraction, optimization and classification.

2.1 Data Processing

Van Breugel et al (2020) have proposed an optimization framework using multi-objectives for selecting factors which reduce loss function for balancing faithfulness as well as smoothness of derivative.[1] Less number of factors is involved for selecting a hyper-parameter. For unbeknown ground-truth data, a heuristic for choosing a hyper-parameter depending on power spectrum and data's temporal resolution is proposed. Ideal value of hyper-parameter is used across diverse differentiation schemes, thus unifying diverse numerical differentiation schemes and facilitating unbiased comparison of outcomes. In the loss function, Root Mean Square Error (RMSE) of integral and total variance of derivative is used as metrics.

Patel & Kashyap (2022) have proposed 2D Flexible Analytical Wavelet Transform (FAWT) which decomposes images which are pre-processed into sub-bands.[2] The statistical significant attributes are determined and Principal Component Analysis (PCA) is employed for identifying appropriate features which are graded by using Student's t-value algorithm. Classification is performed using Least Square-SVM. Based on results, the proposed scheme offers improved classification accuracy, specificity, sensitivity and F1-score using 10-fold Cross-Validation (CV).

Tallapragada et al (2023) have propounded DW-optimized model for Covid-19 diagnosis as well as feature extraction. It filters raw images in pre-processing stage for eliminating needless noises and enhances image quality by using MMG filtering scheme. It reduces feature dimensionality using modified DW-based MobileNet model. Convolutional Aquila Covid network model is designed for classifying Covid cases. Performance of propounded model is determined based on accuracy, sensitivity, specificity, F-score, precision, Negative Predictive Value (NPV) as well as Positive PV (PPV).[3]

2.2 Image Feature Extraction

Kong & Cheng (2022) have propounded a scheme for X-ray images of chest based on fusing of features of DenseNet and VGG16. It includes attention mechanisms to extract deep features. [5] It uses ResNet for segmenting efficient image information for rapidly achieving precise classification. The proposed scheme offers better mean accuracy. Teodoro et al (2023) have used diverse CNN models which are pre-trained for extracting features from images and identifying suitable features. [6] A framework which chooses a combination of CNN and classification models is presented for classifying cases into with and without Covid-19 ones, and performance is analyzed based on features determined using CNNs.

Srinivas et al (2024) have proposed an integrated model including Inception V3 with VGG16 for predicting Covid-19 from chest X-rays for pre-processing his study. [7] and identifies images with diverse color intensities and sizes. He elucit IV3-VGG includes the following blocks contain Initial block which designed for VGG-16, next 2 blocks deal with Inception V3 and the last block includes Average pooling, FC, dropout and Softmax layers. The proposed scheme offers better accuracy. Karthi (2023) have proposed Gradient Mutated Leader Algorithm (GMLA) for altering DenseNet weights by merging Mutated leader's behavior for leading members to update position with Gradient Descent Algorithm (GDA) for attaining computation efficiency with robust convergence that aids in getting best solution for regulating weights for efficient classification. [8] It uses data augmentation by rotating, zooming, shifting and flipping for getting balanced data involving huge data sizes that make efficient classification. The performance is assessed based on Precision, True Positive Rate (TPR), True Negative Rate (TNR) and accuracy. Kaya & Gürsoy (2023) have propounded Transfer Learning (TL) mechanism based on MobileNetV2 for classifying Covid-19 from images.[9] Additional schemes are proposed for increasing performance of the model. The models offer mean accuracy rates for cases with 5-fold CV. The model is assessed and trained on datasets using standard training schemes.

2.3 Classification

Bhosale & Patnaik (2023) have proposed PulDi-COVID, a CNN model for identifying different diseases like Covid-19, viral-Pneumonia etc., by using CXI. It performs training using TL models on CXI of Covid-19 and lung diseases.[10] The ensemble models handle constraints of DL by making forecasts with diverse classifiers. A model which combine DL and ensemble learning is proposed. It is formed by including several DL models that includes Covid-19 case detection with deep NN produced CXI.

Sitaula & Shahi (2023) have represented every tweet by combining syntactic as well as semantic information. Syntactic information is produced from Bag of Words (BoWs) scheme, while semantic information is produced from both fast Text and domain based schemes. [11] Multi-channel CNN (MCNN), an ensemble of CNNs designed for capturing multi-scale information and performing improved classification is proposed. Efficacies of propounded feature extraction scheme and MCNN model are assessed for categorizing into positive, neutral and negative cases. The proposed scheme offers improved classification accuracy.

3. IMAGE PRE-PROCESSING

Input X-Ray images are pre-processed to remove noise using Discrete Wavelet Transform (DWT) and Total Variation Regularization (TVR).

3.1 Discrete Wavelet Transform (DWT)

Most of the signals are discrete instead of being continuous leading to Discrete Wavelet Transform (DWT).

- Specific values of 'a' and 'b' are used
- Owing to signal's discrete representation, integrals which define co-efficients should be numerically computed

Orthogonal family of wavelets may be got for mother wavelet by assigning,

$$a = a_0^m \quad (1)$$

$$b = n.b_0 \quad (2)$$

where 'm' and 'n' signify integers, $a_0 > 1$ and $b_0 > 0$ denote dilation and translation parameters respectively.

To guarantee wavelets 'ψ_a^b' for fixed value of 'a', for cover '$f(x)$' with increase in 'm', $b_0 = \beta.a_0^m$. For quick computation of wavelet coefficients, $\beta = 1$ and $a_0 = 2$. By selecting $b_0 < 2^m$, redundant wavelet family $(b_0 > 2^m)$ leads to partial demonstration of transformed function. Thus, $b_0 = 2^m$ is an ideal choice leading to orthogonal family.

With values of 'a' and 'b', DWT of function ($f(x)$) is given by,

$$W.f(m,n) = \langle \varphi_m^n, f \rangle = \int_{-\infty}^{\infty} \varphi_m^n(x) \times f(x)\, dx \quad (3)$$

where,

$$\varphi_m^n(x) = 2^{-\frac{m}{2}} \times \varphi\left(\frac{x - n \times 2^m}{2^m}\right) \quad (4)$$

Inverse transform is shown in Eq. (3).

$$f(x) = \sum_{m,n} \varphi_m^n(x) \times W.f(m,n) \quad (5)$$

It must be seen that though integral defining '$W.f(m,n)$' is within boundless interval, it is effective on limited interval in case mother wavelet comes with dense support, and may be numerically approximated.

3.2 Total Variation Regularization (TVR)

Total Variation Denoising (TVD) also called TVR or Total variation Filtering (TF) aids in removing noise (filter). Signals with extreme and probably bogus details have increased TV with increased integral of image gradient (Rudin et al 1992). Reducing TV of signal which is a close match to signal eliminates undesirable detail while conserving significant details like edges. [12]

This scheme is better when compared to simple schemes like median filtering and linear smoothing that lessens noise but smooths the edges. In contrast, TVD is an effective filter which preserves edges i.e., instantaneously conserving edges while smoothing noise in flat areas even when Signal-to-Noise Ratio (SNR) is reduced [13].

3.2.1 1D signal series

For digital signal (y_n), TV is defined as,

$$V_{\grave{y}} = \sum_n |\grave{y}_{n+1} - \grave{y}_n| \quad (6)$$

For input signal ($\dot{x}_n$), the main aim of TVD is to determine an approximation ($\grave{y}_n$) with reduced TV when compared to ‘$\dot{x}_n$’ but is ‘near’ to ‘$\dot{x}_n$’. Closeness is Sum of Square Error (SSE):

$$E_{\dot{x}}^{\grave{y}} = \frac{1}{n}\sum_n (\dot{x}_n - \grave{y}_n)^2 \quad (7)$$

TND problem leads to reduction of discrete function over ‘y_n’:

$$E_{\dot{x}}^{\grave{y}} + \lambda . V_{\grave{y}} \quad (8)$$

By discriminating the function with respect to ‘$\grave{y}_n$’, respective Euler-Lagrange is derived which is numerically combined with ‘$\dot{x}_n$’ as primary condition. As this is a convex function, schemes from convex optimization may be employed for minimising it and finding solution ‘$\grave{y}_n$’ [14].

3.2.2 Properties of Regularization

Regularization factor (λ) plays a dominant part in denoising. In case $\lambda = 0$, smoothing is not done and outcome is same as reducing SS. As $\lambda \to \infty$, nevertheless, TV term plays a dominant role that forces the outcome to have lesser TV with less cost like noisy signal. Regularization factor is dominant in attaining accurate quantity of noise exclusion.

3.2.3 2D Signal Images

In 2D signals ($\grave{y}$) like images, TV norm is given by,

$$V_{\grave{y}} = \sum_{i,j} \sqrt{\left|\grave{y}_{i+1}^{j} - \grave{y}_{i}^{j}\right|^2 + \left|\grave{y}_{i}^{j+1} - \grave{y}_{i}^{j}\right|^2} \quad (9)$$

It is isotropic and is non-differentiable. Variation which may be easier to minimalize is anisotropic form.

$$V_{\mathrm{Aniso}_{\grave{y}}} = \sum_{i,j} \sqrt{\left|\grave{y}_{i+1}^{j} - \grave{y}_{i}^{j}\right|^2 + \left|\grave{y}_{i}^{j+1} - \grave{y}_{i}^{j}\right|^2} = \sum_{i,j} \grave{y}_{i+1}^{j} - \grave{y}_{i}^{j} \sqrt{\left|\grave{y}_{i+1}^{j} - \grave{y}_{i}^{j}\right| + \left|\grave{y}_{i}^{j+1} - \grave{y}_{i}^{j}\right|} \quad (10)$$

Standard TVD problem is of form,

$$\min_{\grave{y}} \left[E_{\dot{x}}^{\grave{y}} + \lambda . V_{\grave{y}}\right] \quad (11)$$

Where, ‘E’ represents ‘L_2’ norm which is 2D

When compared to 1D signal image, handling denoising is non-trivial. Primal Dual is recently used [15].

4. FEATURE EXTRACTION

Feature extraction system includes the following 2 steps:

- Offline training
- Online retrieval

AlexNet is used in offline training for determining deep features. This network comprises of 5 convolution layers, 3 maxpooling and 3 FC layers, whereas 'ReLu' function is employed in activation function. 'Softmax' classifier is used in classifying images into Covid-19, Pneumonia and normal ones. Once system is trained, system is capable of classifying input images by determining deep features. Appropriate features are determined from different layers. Features are obtained from 'FC_8' layer of CNN.

To determine a set of feature vectors, images must be sent to the system for storing vectors in the bank called feature matrix. After training, the learned factors of CNN and feature matrix are available in online image retrieval. In this step, an image taken by user is passed through trained CNN. The values of last layer of CNN are taken as a feature vector. Images similar to query image is determined from database. To perform this, retrieval system is considered as an optimization problem. To deal with the problem, a heuristic model is proposed for performing search and retrieval using modified GOA.

In this paper, features are extracted using Visual Geometry Group 16 (VGG16), InceptionV3, DenseNet and MobileNet.

4.1 Visual Geometry Group 16 (VGG16)

VGG16 is formed by sequentially using 33 kernel-sized filters, 11 and 5 filters in Convolutional Layers (CLs). 224×224 RGB image is given as input. VGG16 is a deeper CNN model. VGG-19 CNN is trained using several images from database. [16] Network involves a depth with 19 layers which is efficient in categorizing images of numerous classes. Main aim is to maintain convolution size constant when designing a deep network.

The basic design is to substitute huge kernels with smaller ones, and increasing CNN model's depth VGG16 is highly consistent in performing diverse classification tasks. It includes 5 blocks involving 41 layers comprising of 16 layers involving learnable weights, 13 CLs along with 3 FCC layers from learnable layers (Khan et al 2020) [16][17][18]. Initial 2 blocks comprise of 2 CLs, whereas last 3 blocks include 3 CLs which involve kernels of 3×3 size along with padding 1. Layers are split using maxpooling layers which make use of filters of size 2×2 and padding 1.The CL's output is 4096 which represents quantity of neurons in FCC. VGG16 may use 134 million factors that increase complexity of VGG16 in contrast to other pre-trained models [19][20].

4.2 InceptionV3

Google teams offered InceptionV3 CNN (Szegedy et al 2016). InceptionV3 involves label smoothing, 7×7 convolutions which are factorized, secondary classifier for transmitting label information along network, and batch normalization for layers on the side head. It performs small convolutions for faster training involving small grids to handle computational costs. Several optimization schemes like factorized convolutions, regularization, reduction of dimension and parallelized computations may be used in InceptionV3 model for relaxing limitations and enable model compliance.

Framework of InceptionV3 focuses on some issues in former inceptionV1 like auxiliary classifiers with batch normalization as well as representation bottleneck by including kernel factorization. The framework includes several kinds of kernels in same level. This structure focuses on solving the challenge related to high variation in position of salient areas in input images which are taken into consideration [21]. InceptionV3 uses a filter $(1 \times 7 \ \& \ 1 \times 5)$ instead of huge filter$(7 \times 7 \ \& \ 5 \times 5)$ [22]. Further, a block of 1×1 convolution is used for facilitating improved feature representation.

It takes an input image, performs mapped parallel computations which are shaped into 3 CLs with filters of size 3×3 or 5×5. Output from layers is combined into a layer that signifies output layer (ensemble). By employing parallel layers, the amount of memory involved can be reduced, increasing model's capacity and not depth.

4.3 DenseNet

In Dense CNN (DenseNet) [23], every layer is directly associated with each other layer leading to densely linked CNN. There are '$\frac{l(l+1)}{2}$' direct connections involving 'l' levels. Average of feature maps from former layers are not found but combined and used as input in every layer. This demands less number of factors in contrast to traditional CNN that facilitates feature reclaim by eliminating redundant feature maps. In dense blocks, feature maps' sizes are kept constant within block with varying number of filters. The in-between layers are called Transition Layers which are accountable for image down-sampling by using normalization of batch, 1×1 convolution and 2×2 pooling layers.

The framework is improved using ResNet model (Alzubaidi et al 2021). It is based on a linking model with dense connections rather than a direct one in hidden layers of CNN. Features which are extracted or features map is given to model. The quantity of training factors is reduced in contrast to other CNN models due to direct feature synchronization to ensuing layers [24] [25]. DenseNet utilizes features, making structure more effective. Thus, performance is improved DenseNet includes composition layer, ReLU activation function along with dense blocks. Last layer includes a collection of FC layers [26].

4.4 MobileNet

MobileNet is a light-weight CNN model which depends on inverted residuals which involve linear bottleneck which forms short connections amid thin layers [27]. It involves a module with reversed residual structure. By using MobileNetV2, contemporary object recognition as well as semantic segmentation is performed. The framework includes a fully CL involving 32 filters along with 19 remaining bottleneck layers. Accuracy may be improved by eliminating ReLU6 from output of every bottleneck module.

MobileNet is capable of handling restricted hardware resources as the model involves low-latency and reduced power. A tradeoff exists amid several factors like latency, resolution and accuracy [28]. The Depth Separable Convolutional (DSC) along with point-based convolutional kernels is employed for producing feature maps. It is a method of factorization that substitutes standard convolution with a quicker one. It uses depth-based kennel 2-D filters for filtering spatial dimensions of image. The filter's size is '$\mathrm{D_k} \times \mathrm{D_k} \times 1$', where '$D_k$' represents filter size that is less than size of images taken as input. Point-wise convolutional filter is applied for filtering depth dimension. Depth filter's size is '$1 \times 1 \times n$', where 'n' represents quantity of kernels. They extract every DSC from point-wise convolution by using batch normalization along with ReLU function. Last FCC is associated with Softmax layer to generate final classification output. Depth-wise convolution involves reduced computation complexity. It is 7 times faster in contrast to convolutional networks. It can be used when there is restricted hardware [29]. Features extraction involves fusion of several features.

5. OPTIMIZATION

Feature optimization is performed using Grasshopper Optimization Algorithm (GOA) and Enhanced GOA (EGOA).

5.1 Grasshopper Optimization Algorithm (GOA)

GOA, a meta-heuristic optimization algorithm mimics grasshoppers swarm for solving optimization problem. The insects fly as a group wherein, every grasshopper involves a specific distance to others and movement is modeled as shown below.

$$\mathbf{X_i = S_i + G_i + A_i} \tag{12}$$

Where, '$\mathbf{X_i}$' signifies position of '$\mathbf{i^{th}}$' grasshopper, '$\mathbf{S_i}$' denotes social interaction and '$\mathbf{A_i}$' signifies wind influence.

Interaction plays a major role of the algorithm that is represented using following Eq. (13).

$$\mathbf{S_i = \sum_{\substack{1 \le j \le n \\ j \ne i}} s\left(d_i^j\right) \times \tilde{d}_i^j} \tag{13}$$

Distance amid '$\mathbf{i^{th}}$' and '$\mathbf{j^{th}}$' grasshopper is represented as a unit vector ($\tilde{\mathbf{d}}_{\mathbf{i}}^{\mathbf{j}}$), and '**s**' signifies a function for representing social force determined using Eq. (14).

$$\mathbf{s_d} = \mathbf{f} \times \mathbf{e}^{-\frac{\mathbf{d}}{\mathbf{l}}} - \mathbf{e}^{-\mathbf{d}} \tag{14}$$

Where, '**f**' signifies attraction intensity, '**l**' denotes attractive length scale and '**d**' represents distance amid grasshoppers

Though suitable model for grasshopper movement is used for dealing with optimization problem, Eq. (15) is used in enhanced form of the model for efficiently solving the problem.

$$x_i^d = c \times \left(\sum_{\substack{1 \le j \le n \\ j \ne i}} \frac{UB_d - LB_d}{2} \times s\left(\left| x_j^d - x_i^d \right| \times \frac{x_j - x_i}{2} \right) \right) + \tilde{T}_d \tag{15}$$

Where, 'UB_d' and 'LB_d' denote upper and lower bound respectively, and '$\tilde{T}_d$' represents target location in 'D^{th}' dimension. 'c' signifies volume for dealing with exploration and exploitation. It must be reduced by raising the iteration number. An efficient scheme for performing this reduction is given in Eq. (16).

$$c = c_{max} - l \times \frac{c_{max} - c_{min}}{L} \tag{16}$$

5.2 Enhanced GOA (EGOA)

A multi-dimensional fitness function may be used for retrieving images. For reducing the computation cost along with time, 1D function is propounded. K-best particles signify 'K' images for retrieval. Further, GOA is employed to deal with continuous variables, whereas a discrete algorithm is needed due to discrete indices of images. Variables are converted to discrete form. In the meantime, every discrete grasshopper signifies image index in the database. In the fitness function, Euclidian distance is determined amid features of query image and those of grasshoppers.

$$F_i = \sum_{i=1}^{d} \|r - v_i\| \tag{17}$$

Where, 'd' refers to quantity of image dimensions, 'r' denotes Query feature vector and 'v_i' signifies 'i^{th}' vector in feature matrix. The factor 'i' denotes image index in database which is the rounded state of grasshopper 'i'.

$$i = [j + 0.5] \tag{18}$$

Where, [.] represents integer part and 'j' signifies location of grasshopper

Exploration and exploitation play a dominant role in GOA for performing a proper search in possible space and best solution that reduces the fitness function is determined. Grasshopper model is proposed as follows:

$$x_i = \begin{cases} 10 \times ran \times m_i, \text{if } l \gg \theta \\ m_i, l < \theta \end{cases} \quad (19)$$

Where, 'm_i' is same as Eq. (15) and 'd' is taken as '1'

$$m_i = \left(\sum_{\substack{1 \le j \le n \\ j \ne i}} c \times \frac{UB-LB}{2} \times s\left(|x_f - x_i| \times \frac{x_j - x_i}{d_i^j} \right) \right) + \tilde{T}_d \quad (20)$$

In Eq. (19), 'ran' represents a variable in the range [0, 1]. '$10 \times$ran' signifies a number within range [0, 10]. The factor 'θ' is a volume which has a control on trade-off amid exploration as well as exploitation. This leads to suitable search as well as exploration when iteration is not more than 'θ' and gives the best answer in ensuing iterations. When EGOA converges at the best solution, 'k' best grasshoppers in former iteration are chosen as 'k' nearest images to given image.

Algorithm 1: Image retrieval using EGOA

A: Offline step

Resize images and feed them to CNN for training

Record ideal parameters

Form feature matrix by sending images to trained CNN

B: Offline retrieval

Resize query images to a size of 224×224 and give as input to CNN

Determine query features from FC8

Initialize EGOA

Randomly distribute particles

$C_{max} = 2, C_{min} = 0.004, Iter_{Max} = 100$

Determine fitness of particles using Eq. (17)

While ($l < Iter_{Max}$)

Sort particles

Calculate 'c' by using Eq. (16)

For (every particle)

If

Determine location of particle using $x_i = 10 \times m_i \times ran$

else

Determine location of particle using $x_i = m_i$

end

Return (particles to borders in case they are left)

End while

Choose ‘K’ best particles obtained in previous iteration

Round ‘K’ best location using $i = [j + 0.5]$

Denote images that the number is represented by ‘K’ best particles

6. CLASSIFICATION

The images are classified into Pneumonia, Covid-19, normal X-Ray images using CNN and Ensemble CNN (ECNN).

6.1 Convolutional Neural Network (CNN)

CNN is a kind of DL NN architecture which is usually employed in computer vision that facilitates a computer to comprehend and infer image or visual data. Artificial NNs (ANNs) offer better performance. NNs are used in diverse datasets including images, text and audio. NNs find their applications in diverse areas like predicting word sequence (Recurrent NN) and image classification (CNN). A NN includes 3 layers.

- **Input Layer:** Input is given through this layer. Quantity of neurons represents the quantity of features in the data.
- **Hidden Layer:** Input from former layer is fed to this layer which has many neurons more than amount of features. There may be several hidden layers based on the model and size of data. Output from every layer is determined by finding the product of former layer’s output involving learnable weights along with learnable biases. It includes an activation function that makes network non-linear.
- **Output Layer:** Output of hidden layer is sent to logistic function that transforms the class’ output into probability scores of classes.

CNN is extended form of ANN which is mainly used for extracting feature from grid-based matrix dataset. It comprises of several layers including input, convolutional, pooling layer as well as Fully Connected (FC) layers. The CL applies filters on input for extracting features, whereas pooling layer down samples image to decrease computation, while FC layer offers final forecast. The network learns ideal filters by performing backpropagation as well as gradient descent. Complete CNN framework is also called covnets which involves several layers, and each layer converts a volume to another using differentiable function.

- **Input Layer:** Input as a sequence of images is given to the model through this layer.

- **Convolutional Layer (CL):** This layer is used for extracting feature from dataset. It uses a collection of learnable filters called kernels to images. Filters or kernels involve small matrices. It moves over input and determines dot product amid kernel weight and respective image patch. Layer's output is called feature maps.

- **Activation Layer:** By applying activation function to output of former and activation layers, non-linearity can be incorporated into network. It applies element-based activation function to output of convolution layer.

- **Pooling layer:** This is periodically included and its chief function is to reduce volume which hastens computation, lessens memory as well as avoids overfitting. Maxpooling and average pooling are considered as common layers.

- **Flattening:** Flattened feature maps are 1D vectors obtained following convolution as well as pooling layers so that they may be sent to a entirely linked layer for classification or regression.

- **FC Layer:** It takes input from former layer and determines regression task or final classification.

- **Output Layer:** Output from FCs is sent to logistic function (Sigmoid or Softmax) for classification that transforms output of every category into probability score of every class.

6.2 ENSEMBLE CNN (ECNN)

Proposed deep Ensemble CNN (ECNN) framework involves 2 stages (base classifiers and fuse schemes). In previous stage, 4 modified models are used: InceptionV3, DenseNet121, MobileNetV2 as well as VGG16 pre-trained CNN classifier. In the latter stage, decisions of base classifiers are combined (first stage). Fusion schemes like majority voting as well as product rule are employed in propounded ECNN. Majority voting uses value of prediction of base classifier. Product rule uses probabilities of base classifiers (pre-trained model).

6.2.1 Enhanced Deep Pre-Trained Models

Once pre-trained models like VGG16, InceptionV3, MobileNet, DenseNet are adapted, they act as base classifiers in propounded ECNN for automatic classification of images. Standard former pre-trained models determine diverse features from training images to differentiate amid Covid-19, Pneumonia and normal images. Nevertheless, pre-trained models involve convolution layers as well as filter sizes for determining diverse features from images. Pre-trained models are not common in determining distinguishing features from training images taken as input (Ghosh et al 2021).

Nevertheless, initial weights affect classification performance as pre-trained models of CNN are non-linear designs. The pre-trained models learn complex relations from training data using back propagation as well as stochastic optimization (Ahmad et al 2021). A block-based fine-tuning scheme is used for adapting typical CNN models for handling heterogeneity in image classification.

Benchmark images are loaded, and training as well as testing images is pre-processed for feeding them to pre-trained models (resized) (Simonyan & Zisserman 2014; Szegedy et al 2016). Images are rescaled to 1/255 as in former studies. Once dataset is split for training as well as testing, pre-trained models: DenseNet, VGG16, MobileNet and InceptionV3 are loaded without modifying weights (Huang et al 2017; Sandler et al 2018).

FCC as well as softmax layers are circumvented from pre-trained models. The layers are designed for returning classes from dataset. Two layers which are dense with several hidden neurons are included for strengthening essential data-centric feature learning from every pre-trained model (Ahmad et al 2021). ReLU which is a non-linear activation function that permits learning complex relationships of data, follow these layers (Garbin et al 2020). Subsequently, 0.3 dropout layer (Deniz et al 2018) is included for reducing training time as well as overfitting issues during classification (Boumaraf et al 2021).

After every pre-trained model, final FCC with Softmax layer is included. FCC is a Feed-Forward Neural Network (FFNN) that takes flattened input from final pooling layer. Depending on quantity of classes, amount of neurons in FCC is assigned a value of 4. Softmax layer is added to top of every model for training features and producing classification output depending on maximum probability. The algorithm includes phases of block-based fine-tuning mechanism for every model in propounded ECNN.

6.2.2 Ensemble Fusing Schemes

ECNN is pre-trained for automated classification of Pneumonia, Covid-19 and normal case from the input images. Dynamic models are trained on dataset and assessed. Probabilities of output of pre-trained models are linked to generate feature vector which is of 16D. Every individual with Softmax generates 3 probabilities depending on quantity of classes.

Diverse combination schemes including majority voting as well as product rule are included for producing a decision for query image. In case of majority voting, every base classifier assigns a label (predicted) to assess the sample. It determines number of votes of labels from fundamental classifiers. Class which gets maximum amount of votes is chosen as decision for ECNN (majority voting) as shown below.

$$P(I) = \max_{j=1 \text{ to } c} \sum_{t=1}^{T=4} d_t^j \tag{21}$$

Where, '$P(I)$' represents decision for query image (I), 't' signifies base classifier, '$T = 4$' represents number of base classifiers, 'c' denotes class label. 'd_t^j' signifies class label 'j' for 'I' by 't'. Decision of 'I' is class 'j' with maximum occurrence.

In case of product rule, output of posterior probability $\left(p_t^j(I)\right)$ for every label 'j' is produced by base 't' for 'I'. Class with increased probability of product is taken as final decision. Eq. (22) represents product rule in ECNN.

Algorithm shows ECNNs with majority voting as well as product rule.

p-ISSN 0854-1418
e-ISSN 2503-426X

$$P(I) = \max_{j=1 \text{ to } c} \prod_{t=1}^{T=4} p_t^j(I) \quad (22)$$

Initially, CNN model is used for handling heterogeneity in images by using block-based fine-tuning scheme for every pre-trained model. It determines added abstract features from images which help in raising discrimination within class. Ensemble learning is used for improving performance of pre-trained models. Final decision concerning query images is more precise.

Figure 2 shows the steps followed in proposed ECNN. The pre-processed images are split into both training and testing datasets which are further sent to fine-tuned pre-trained models for feature extraction. Then, majority voting is carried out followed by optimization and classification.

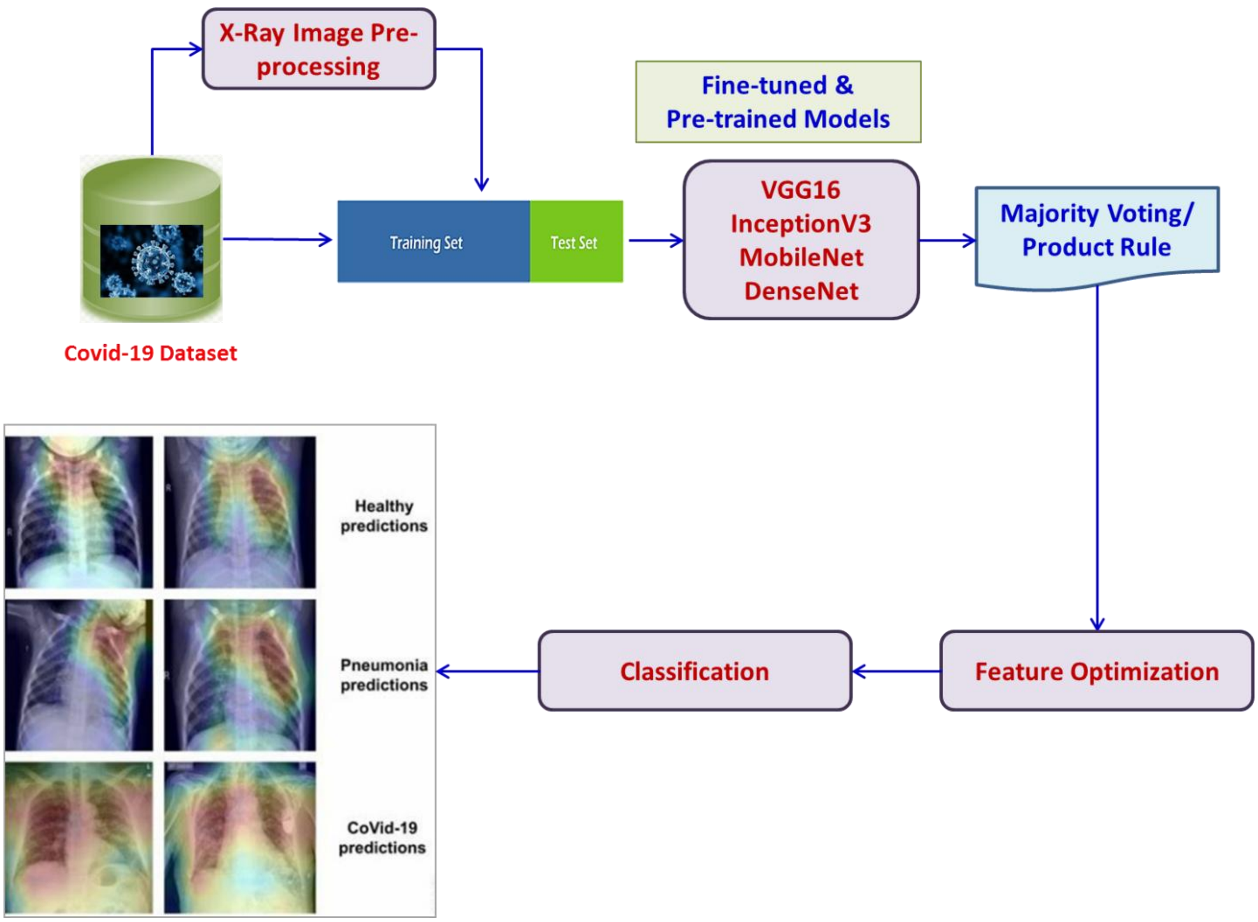

Figure 2: Proposed ECNN with Adaptive and Pre-trained Models

Algorithm 2: Constructing and training pre-trained models

Input:
T - Training data
n - Number of samples

$T = [x_1, x_2, \ldots, x_n]$

Class: $y = [y_1, y_2, \ldots, y_n]$

M - Pre-trained CNN models

Output: adaptM - Adaptive models

for (every I in M)

Eliminate last FCC as well as Softmax layers

Include 512 neurons and ReLU to Dense1 layer

Include 64 neurons and ReLU to Dense2 layer

Include dropout layer

Include 3 neurons to FCC layers (based on classes)

Include Softmax layer (output probabilities)

Set values of hyper-parameters

Construct final model (adaptI)

Train adapI on 'T'

Add adapI to adapM

end for

Algorithm 3: Ensemble of 'adaptM' and assessing ensemble model on images

Input:

adaptM - Adaptive models

D - Set of test images

Samples: $R = [x_1, x_2, \ldots, x_n]$

Class: $y = [y_1, y_2, \ldots, y_n]$

Output: Class forecast for 'D' by using 'ECNN(majority_voting)' & 'ECNN(product_rule)'

for (j in D)

    for (every I in adaptM)

        Assess performance of 'I' based on test data 'j'

    P[j, I] = Probability of every class for 'j' when using 'I'

    V[j, I] = Prediction of 'j' when using 'I' end for

Determine ensemble prediction for 'j' by using majority voting & V[j, :] (ECNN(majority_voting))

Determine ensemble prediction for 'j' by using product rule (ECNN(product_rule)) & p[j, :]

end for

7. RESULTS AND DISCUSSION

Feature extraction as well as fine-tuning aids in leveraging knowledge from a model which is pre-trained on huge source task. This improves performance on target task. Former features are more general which may be used across several tasks, while latter layers signify more precise details of classes in original dataset. Figure 3 and Figure 4 shows the Accuracy analysis and model prediction respectively.

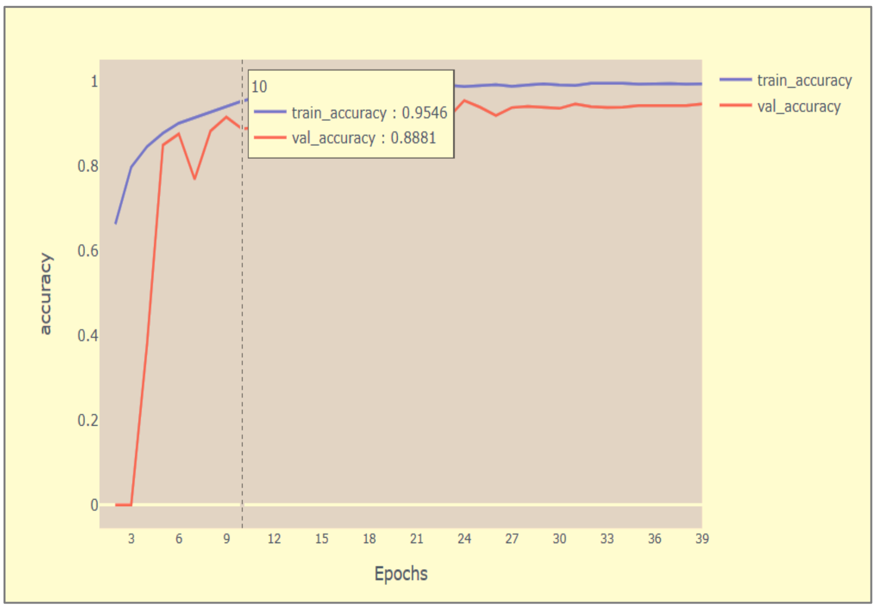

Figure 3: Accuracy Analysis for Testing and Training

Table 1 : Comparison of Pre-Processing Methods

Filters	PSNR	MS	SSIM
DWT	27	0.6	0.96
TVR	38	0.28	0.99

The Table 1 clearly shows the performance comparison of proposed TVR de-nosing method with DWT. The Peak signal-to-noise ratio , Mean Square Error, and structural similarity index measure are the valuation metrics used to evaluate the performance of the proposed TVR de-noising algorithm. Thus The result shows that the proposed method TVR has high PSNR 38, MS 0.28 and SSIM 0.99 value when compared with DWT.

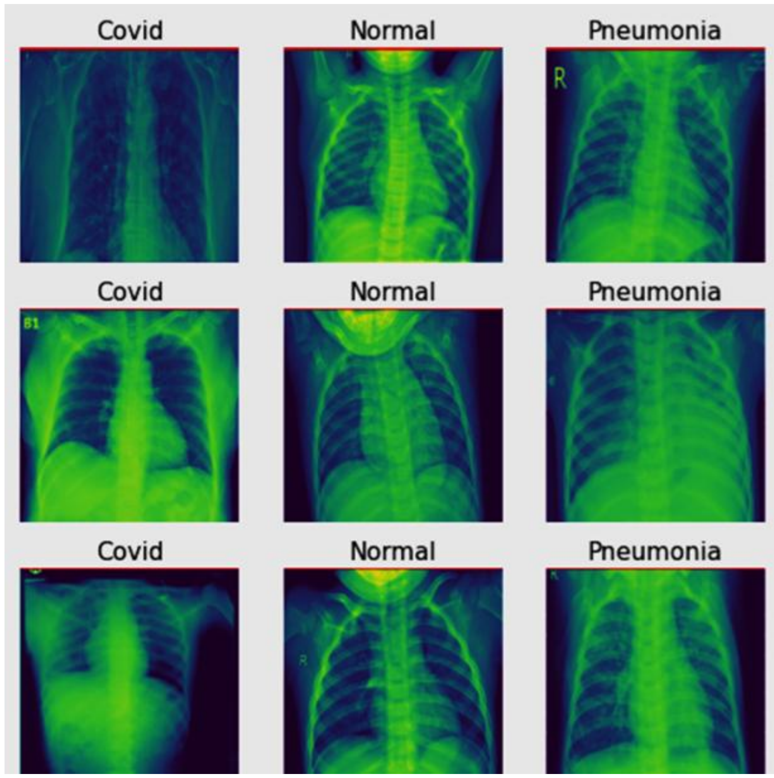

Figure 4: Model Prediction

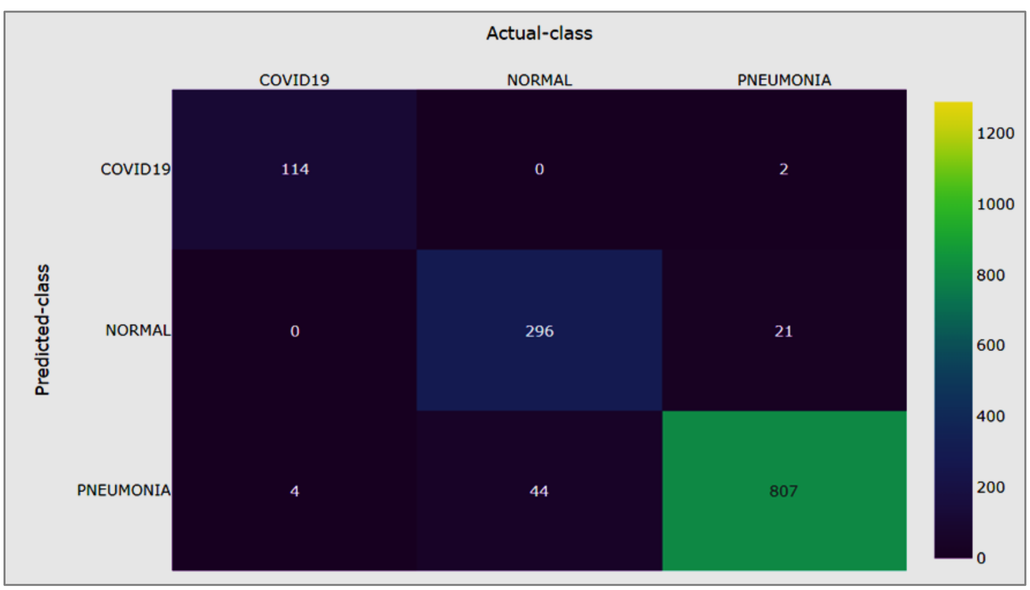

Figure 5: Prediction of Classes using ECNN

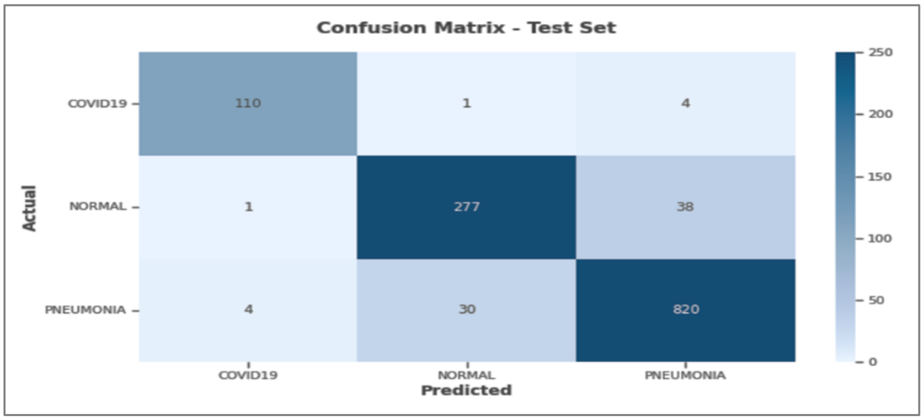

Figure 6: Confusion Matrix obtained using ECNN

Table 2: Classification Performance Analysis of CNN and ECNN

Algorithm	Accuracy	Precision	Recall	F_Measure
CNN	89	88	87	87
ECNN	98.7	97	98.3	98.2

Table 2 illustrates the performance of CNN and ECNN based on Accuracy, Recall, Precision and F_Measure. The result shows that the proposed scheme ECNN offers better result in terms of improved Accuracy 98.7%, Precision 97%, Recal 98.5%l and F1-Score 98.3%. In Figure 5 and Figure 6 show the Confusions matrix for Three- Class (Covid, Normal, Pnuemonia) classification of validation.

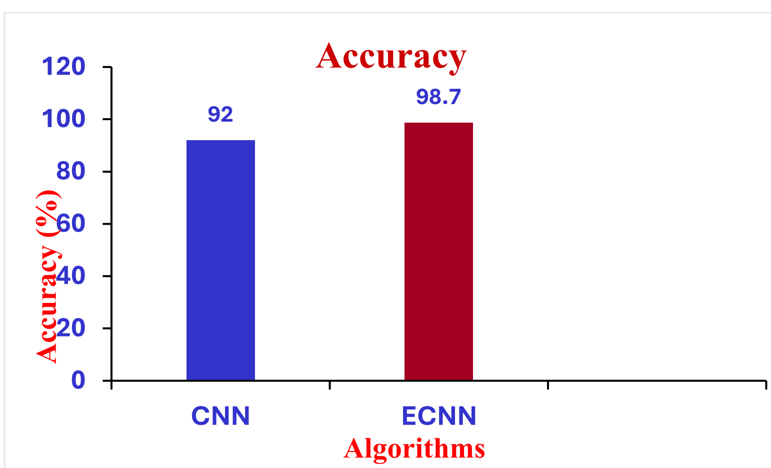

Figure 7: Accuracy

From Figure 7, it is evident that ECNN offers 7% better Accuracy in contrast to CNN.

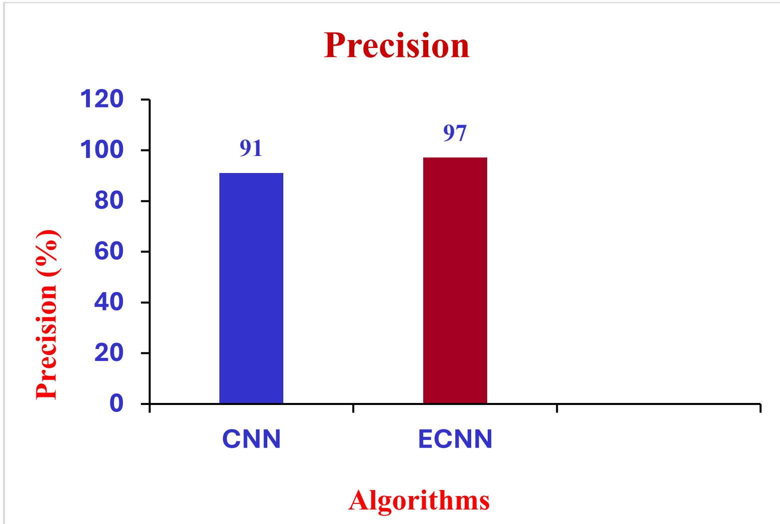

Figure 8: Precision

From Figure 8, it is obvious that ECNN offers 6% better Precision when compared to CNN.

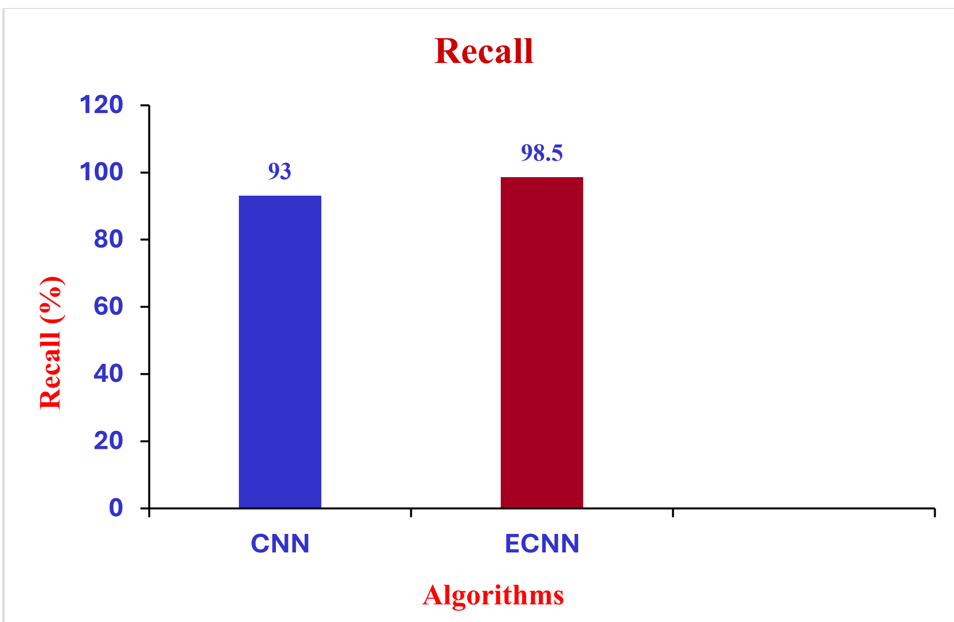

Figure 9: Recall

From Figure 9, it is evident that ECNN offers 6% better Recall in contrast to CNN.

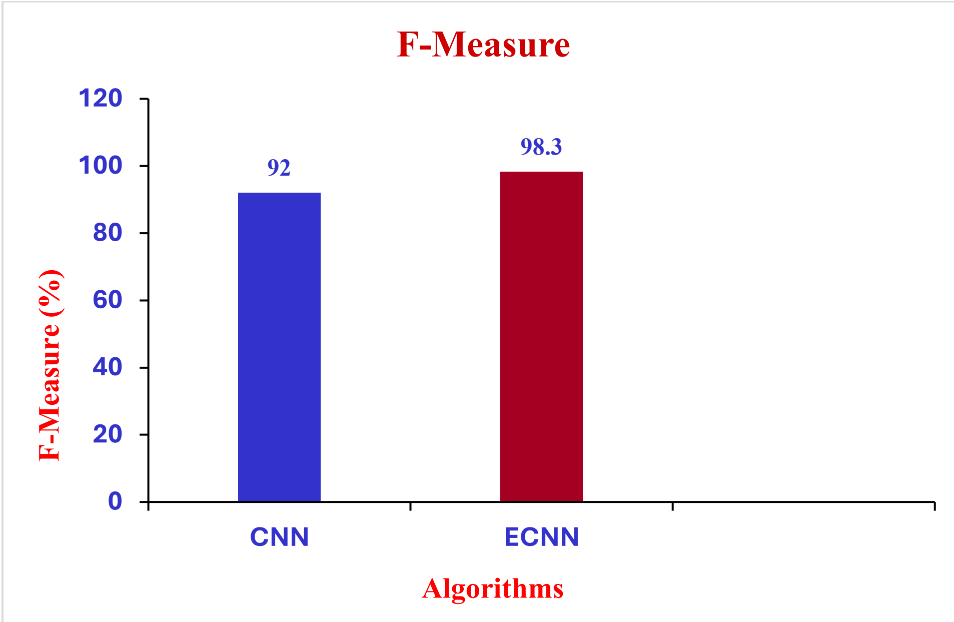

Figure 10: F-Measure

From Figure 10, it is seen that ECNN offers 6% better F-Measure when compared to CNN.

8. CONCLUSION

In this paper, the chest X-Ray images are taken as input. They are pre-processed to remove noise using DWT and TVR. Features are extracted using VGG16, InceptionV3, DenseNet and MobileNet. Optimization is performed using GOA and EGOA. DL-based methods including CNN and ECNN are used for classifying Covid-19, normal or healthy and Pneumonia cases from input X-ray images of the Chest. It is obvious that the proposed scheme ECNN offers better result in terms of improved Accuracy 98.7%, Precision 97%, Recal 98.5%l and F1-Score 98.3%.

9. REFERENCES

[1] Ketkar, N, Moolayil, J, (2021). Convolutional neural networks. Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch, Apress,197-242.

[2] Patel, R. K., & Kashyap, M. (2022). Automated diagnosis of COVID stages from lung CT images using statistical features in 2-dimensional flexible analytic wavelet transform. NIH. Biocybernetics And Biomedical Engineering, 42(3), 829-841.

[3] Tallapragada, V. S., Manga, N. A., & Kumar, G. P. (2023). A novel COVID diagnosis and feature extraction based on discrete wavelet model and classification using X-ray and CT images. Multimedia Tools and Applications, 82(17), 26183-26224.

[4] Van Breugel, F., Kutz, J. N., & Brunton, B. W. (2020). Numerical differentiation of noisy data: A unifying multi-objective optimization framework. IEEE Access, 8, 196865-196877.

[5] Kong, L., & Cheng, J. (2022). Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion. NIH, Biomedical Signal Processing and Control, 77, 103772.

[6] Teodoro, A. A., Silva, D. H., Saadi, M., Okey, O. D., Rosa, R. L., Otaibi, S. A., & Rodríguez, D. Z. (2023). An analysis of image features extracted by CNNs to design classification models for COVID-19 and non-COVID-19. Journal of signal processing systems, 1-13.

[7] Srinivas, K., Gagana Sri, R., Pravallika, K., Nishitha, K., & Polamuri, S. R. (2024). COVID-19 prediction based on hybrid Inception V3 with VGG16 using chest X-ray images. Multimedia Tools and Applications, 83(12), 36665-36682.

[8] Karthi, S., Sudha, L. R., & Navaneetha Krishnan, M. (2023). A Novel Hybrid Optimization Enabled Densenet for Covid-19 Classification using CT Images. Sensing and Imaging, 24(1), 31.

[9] Kaya, Y., & Gürsoy, E. (2023). A MobileNet-based CNN model with a novel fine-tuning mechanism for COVID-19 infection detection. Soft Computing, 27(9), 5521-5535.

[10] Bhosale, Y. H., & Patnaik, K. S. (2023). PulDi-COVID: Chronic obstructive pulmonary (lung) diseases with COVID-19 classification using ensemble deep convolutional neural network from chest X-ray images to minimize severity and mortality rates. Biomedical Signal Processing and Control, 81, 104445.

[11] Sitaula, C., & Shahi, T. B. (2023). Multi-channel CNN to classify nepali covid-19 related tweets using hybrid features. Journal of Ambient Intelligence and Humanized Computing, 1-10.

[12] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena, 60(1-4), 259-268.

[13] Strong, D., & Chan, T. (2003). Edge-preserving and scale-dependent properties of total variation regularization. Inverse problems, 19(6), S165.

[14] Little, M. A., & Jones, N. S. (2010, March). Sparse Bayesian step-filtering for high-throughput analysis of molecular machine dynamics. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4162-4165). IEEE.

[15] Chambolle, A. (2004). An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision, 20, 89-97.

[16] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[17] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

[18] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review, 53, 5455-5516.

[19] Tripathi, S., & Singh, S. K. (2020). Ensembling handcrafted features with deep features: an analytical study for classification of routine colon cancer histopathological nuclei images. Multimedia Tools and Applications, 79(47), 34931-34954.

[20] Koklu, M., Cinar, I., & Taspinar, Y. S. (2022). CNN-based bi-directional and directional long-short term memory network for determination of face mask. Biomedical signal processing and control, 71, 103216.

[21] Mishra, A. K., Das, S. K., Roy, P., & Bandyopadhyay, S. (2020). Identifying COVID19 from chest CT images: a deep convolutional neural networks based approach. Journal of Healthcare Engineering, 2020.

[22] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[23] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[24] Ahmad, F., Farooq, A., & Ghani, M. U. (2021). Deep ensemble model for classification of novel coronavirus in chest X-ray images. Computational intelligence and neuroscience, 2021.

[25] Ghosh, S., Bandyopadhyay, A., Sahay, S., Ghosh, R., Kundu, I., & Santosh, K. C. (2021). Colorectal histology tumor detection using ensemble deep neural network. Engineering Applications of Artificial Intelligence, 100, 104202.

[26] Talo, M. (2019). Automated classification of histopathology images using transfer learning. Artificial intelligence in medicine, 101, 101743.

[27] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[28] Krishnamurthy, S., Srinivasan, K., Qaisar, S. M., Vincent, P. D. R., & Chang, C. Y. (2021). Evaluating deep neural network architectures with transfer learning for pneumonitis diagnosis. Computational and Mathematical Methods in Medicine, 2021, 1-12.

[29] Srinivasu, P. N., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., & Kang, J. J. (2021). Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. Sensors, 21(8), 2852.