Multi-texture features and optimized DeepNet for COVID-19 detection using chest x-ray images

Anandbabu Gopatoti1,2 | Vijayalakshmi P1

1Department of Electronics and Communication Engineering, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India
2Anna University, Chennai, Tamil Nadu, India

Correspondence
Anandbabu Gopatoti, Department of Electronics and Communication Engineering, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India. Email: anandbabu.gopathoti@gmail.com

Summary
The corona virus disease 2019 (COVID-19) pandemic has a severe influence on population health all over the world. Various methods are developed for detecting the COVID-19, but the process of diagnosing this problem from radiology and radiography images is one of the effective procedures for diagnosing the affected patients. Therefore, a robust and effective multi-local texture features (MLTF)-based feature extraction approach and Improved Weed Sea-based DeepNet (IWS-based DeepNet) approach is proposed for detecting the COVID-19 at an earlier stage. The developed IWS-based DeepNet is developed for detecting COVID-19 to optimize the structure of the Deep Convolutional Neural Network (Deep CNN). The IWS is devised by incorporating the Improved Invasive Weed Optimization (IIWO) and Sea Lion Optimization (SLnO), respectively. The noises present in the input chest x-ray (CXR) image are discarded using Region of Interest (RoI) extraction by adaptive thresholding technique. For feature extraction, the proposed MLFT is newly developed by considering various texture features for extracting the best features. Finally, the COVID-19 detection is performed using the proposed IWS-based DeepNet. Furthermore, the proposed technique achieved effective performance in terms of True Positive Rate (TPR), True Negative Rate (TNR), and accuracy with the maximum values of 0.933%, 0.890%, and 0.919%.

KEYWORDS
COVID-19, deep convolutional neural network (deep CNN), deep learning, improved invasive weed optimization (IIWO), sea lion optimization (SLnO)

1 INTRODUCTION

Coronavirus is an outbreak that transmits very quickly from person to person. For this case, it has very overwhelming effects in numerous fields across the world.1 Hence, it is very important to identify the COVID-19 disease2 as fast as possible for controlling the virus spread. Moreover, the COVID-19 outbreak is a respiratory infection that is caused by serious acute respiratory syndrome coronavirus 2 (SARS-CoV-2).3 It was detected in November 2019 in Wuhan city. In the first month of 2020, COVID-19 expands all over the world through transmission from human to human.4 Coronavirus affects animals, but the virus can be transmitted to humans due to its zoonotic nature.5 The widespread occurrence of COVID-19 has been of greater trouble to the world health organization since no effecual treatment has been found.6 The biological composition of COVID-19 consists of positive-sense single-stranded RNA-type, and it is more complicated to diagnose the COVID-19 due to its mutating power.7 Moreover, the COVID-19 disease may cause severe problems for the patients having difficulty in breathing, sudden cardiac arrest, multiorgan failure, and pneumonia.8 Medical professionals across the world are undergoing demanding research to discover a robust treatment for diagnosing COVID-19 disease.9
The distinctive characteristics of COVID-19 disease are fatigue, fever, muscle pain, cough, breathing problem, and headache. The virus can aggravate the demise of individuals with weakened immune systems. The transmission of COVID-19 disease is from one person to another person by having direct interaction between people. In general, healthy people can be affected by COVID-19 disease through mucous contact, breath contact, and hand contact with the persons having COVID-19. Moreover, the real-time reverse transcription-polymerase chain reaction (RT-PCR) is the most widespread test approach presently utilized for diagnosing COVID-19. In addition, Chest radiological imaging techniques, namely computed tomography (CT) and x-ray have fundamental roles in diagnosing and treating the COVID-19 diseases in the earlier stage. CXR images have greater potential in monitoring and examining various types of lung diseases, like infiltration, pneumonia, atelectasis, hernia, and tuberculosis. Consequently, CXR images may also be utilized in the COVID-19 detection. A chest radiology image-driven detection model has numerous benefits over various conventional techniques due to its less cost, less radiation dose, capability to analyze several cases at the same time, and ease of access. It can be very supportive for hospitals with no or inadequate amount of testing resources and testing kits.

The automatic assessment of CXR can be carried out using various deep learning-enabled methods, which may speed up the analysis. Deep learning is generally the most populous research topic of artificial intelligence (AI), which promotes the construction of end-to-end architectures for achieving consequences using input data, without the requirement of extracting the features in a manual mode. In addition, deep learning approaches have been accomplished in treating various issues like breast cancer classification, brain tumor identification, skin cancer detection, arrhythmia detection, lung cancer classification, fundus image segmentation, and pneumonia prediction using CXR images. In Reference 9, a deep learning approach was introduced for diagnosing the COVID-19 automatically. The introduced approach consists of an end-to-end architecture model without using any kind of feature extraction methods, and it requires unprocessed CXR images to perform the analysis. In References 22,23, the CNN model was devised for diagnosing the COVID-19 affected individuals utilizing chest CT imaging. The gated bidirectional CNNs (GCNN) was employed for detecting COVID-19. Based on in-depth investigation, it is found that deep learning methods achieve better outcomes in classifying the COVID-19 disease from lung CT imaging. Recovery from the virus causing COVID-19 is very much dependent on the immunity of the affected host. The virus prevention includes following personal hygiene and social distancing.

The major goal of this research article is to devise an efficient MLTF + IWS-enabled DeepNet method for detecting the COVID-19 disease in people. Initially, the CXR image collected from the dataset is given as an input to the preprocessing component to get rid of the noises. Here, the preprocessing is performed by the RoI extraction based adaptive thresholding technique. Then, the lung lobe segmentation process is performed using SegNet model. Based on the interested lobe regions, the proposed MLTF is formed based on the effective texture features, like Local optimal oriented pattern (LOOP), local gradient hexa pattern (LGHP), local binary pattern (LBP), local directional pattern (LDP), local ternary pattern (LTP), local directional ternary pattern (LDTP), local gradient pattern (LGP), and harmonic mean local gradient pattern (HLP). Finally, COVID-19 detection is performed using proposed IWS-based DeepNet. Here, Deep CNN classifier is accomplished to detect COVID-19, where the structural optimization of Deep CNN classifier is done using IWS algorithm.

The key objective of this article is illustrated as follows,

- **Proposed MLTF from interested lobe region**: The proposed MLTF is formed from the interested lobe regions by considering the texture features, such as LOOP, LGHP, LBP, LDP, LTP, LDTP, LGP, and HLP in order to improve the detection process.

- **Proposed IWS-based DeepNet**: An effective COVID-19 detection technique is devised using IWS-based DeepNet. Here, COVID-19 detection is performed using Deep CNN classifier. The developed optimization algorithm, named IWS is exploited for optimizing the structural layers of Deep CNN classifier. Moreover, the IWS algorithm is designed by the integration of IIWO and SLnO.

The arrangement of the research article is designed as follows: the various existing COVID-19 detection methods are reviewed in Section 2, the proposed technique for COVID-19 detection is portrayed in Section 3, the implementation outcomes are elucidated in Section 4, and Section 5 describes the conclusion.

2 | LITERATURE SURVEY

This section describes the different existing COVID-19 detection approaches with its advantages and drawbacks. Minaei et al. designed a deep learning framework to predict COVID-19 from CXR images. This method had lesser computational complexity and memory utilization. However, a large quantity of clean labeled COVID-19 images is required for improving the detection accuracy. Shankar and Perusals developed a robust hand-crafted deep learning features driven fusion method to classify the COVID-19. Here, the fusion model was employed for fusing the handcrafted features with the support of deep learning features and LBP. The training time of this method was very low. Meanwhile, this technique does not utilize other effective classification techniques for better performance. Wang et al. introduced a deep CNN technique for identifying the COVID-19 cases from CXR images. This approach effectively improved the screening procedure for making precise and accurate analysis. Consequently, this technique failed to determine the risk status of patients, and also failed to evaluate the hospitalization period for managing the patient population.
Waheed et al.\textsuperscript{27} introduced an auxiliary classifier generative adversarial network (ACGAN) for identifying COVID-19. Here, a CNN-driven approach was employed for the detection process. This method achieved improved performance rate, but this technique failed to improve the synthetic CXR image quality.

Demi\textsuperscript{8} introduced a deep long short term memory (deep LSTM) approach for detecting COVID-19 cases automatically. Besides, the marker-controlled watershed and the Sobel gradient segmentation processes were applied to raw images to improve the preprocessing. This approach effectively reduced the computational time. Moreover, this approach failed to evaluate the performance of the technique using more consistent and challenging dataset. Togacar et al.\textsuperscript{9} developed a support vector machine (SVM) for COVID-19 detection. This technique minimized the image interferences in dataset and hence offered proficient features with stacking method. Meanwhile, this approach failed to work with various sophisticated deep learning methods with correct depth into samples for attaining better outcomes. Singh et al.\textsuperscript{28} designed a hybrid social group optimization (HSGO) to identify COVID-19 disease. The features extracted from images of CXR and the appropriate features were chosen by hybrid social group optimization algorithm. This method had minimum computational cost and attained superior outcomes on the equipped dataset. On the other hand, this technique failed to consider more number of training data. Ismael and Şengür\textsuperscript{18} developed a CNN technique for detecting COVID-19 by considering CXR images. Here, the SVM classifier was utilized with different kernel functions, such as quadratic, linear, cubic, and Gaussian for classifying the deep features. This method obtained higher classification accuracy. However, the major challenge lies in developing graphical user interface (GUI) to support radiologists in detecting COVID-19.

3 \hspace{1em} PROPOSED MLTF + IWS-BASED DEEPNET METHOD FOR COVID-19 DETECTION

This section elucidates the proposed MLTF + IWS-based DeepNet method to detect COVID-19. The major processes performed for the developed COVID-19 detection using CXR images includes four stages, namely preprocessing, lung lobe segmentation, feature extraction, and COVID-19 detection. At first, the input CXR image is given to the preprocessing phase. In preprocessing stage, Region of Interest (RoI) extraction is done using adaptive thresholding mechanism\textsuperscript{29} in order to eliminate the noises and the redundant pixels from the input image. After preprocessing, then lung lobe segmentation is carried out based on SegNet.\textsuperscript{30} After the segmentation of lung lobes, feature extraction is performed using proposed MLTF, which is developed newly by combining various texture features, like LOOP\textsuperscript{31} LGHP\textsuperscript{32} LBP\textsuperscript{31} LDP\textsuperscript{31} LTP\textsuperscript{33} LDTP\textsuperscript{34} LGP\textsuperscript{35} and HLGP\textsuperscript{36} from the interested lobe regions. Finally, IWS-based DeepNet is developed for detecting COVID-19 by optimizing the structure of deep convolutional neural network (Deep CNN).\textsuperscript{37} For structure optimization, IWS algorithm is used, which is newly designed by integrating IIWO,\textsuperscript{38} and SLnO.\textsuperscript{39} The schematic diagram of COVID-19 detection process based on proposed MLTF and optimized IWS-based DeepNet using CXR images is depicted in Figure 1.

3.1 \hspace{1em} Input image acquisition

The input image considered for the COVID-19 detection is CXR image, which is acquired from DeepCovid dataset. Consider dataset $D$ has $\nu$ number of input CXR images as given in Equation (1).

$$D = \{E_1, E_2, \ldots, E_z, \ldots, E_\nu\}.$$  \hspace{1em} (1)

where $D$ is the dataset, $E$ implies input CXR image, $\nu$ specifies the total images, $E_z$ represents $z$th CXR image. The input CXR image is acquired from the database and is presented to the preprocessing module for discarding falsifications from the input image.

3.2 \hspace{1em} Preprocessing by RoI extraction based adaptive thresholding technique

The noises present in the image degrade the visual quality and produce inaccurate results. Hence, preprocessing is done to discard the falsifications or the noises in the image. Here, the input CXR image $E_z$ is given to the preprocessing module such that the input image is preprocessed by RoI extraction based adaptive thresholding technique. The adaptive thresholding strategy removes the noises in the image, which separates the RoI from the input CXR image.

Adaptive thresholding technique: The input CXR image is given to preprocessing module where the image is preprocessed using adaptive thresholding strategy\textsuperscript{29} for eliminating the noises in the image. Image thresholding provides the image segments on the basis of specific image characteristics, while the image characteristics can be the pixels with their intensity levels. In the thresholding process, the image is represented in a binary configuration by classifying the image as dark or light. Here, adaptive thresholding is utilized to overcome the variations produced as result of illumination. This strategy utilizes varying threshold values for every pixels present in the image and it provides effectiveness toward illumination variations. In the adaptive thresholding procedure, the integral image is computed initially, and thereafter the average value is measured for the
\[ \delta \times \delta \] window of every pixel in the integral image. If the pixel value is less than the average of \( \delta \times \delta \) neighbors, then the pixel value is fixed as black, or else the pixel value is fixed as white. The major advantages of thresholding strategy are that it offers uncomplicated, accurate, and clean results. The outcome of adaptive thresholding strategy is the noise free preprocessed CXR image, which is signified as \( P_c \), and is fed to the lung lobe segmentation phase.

### 3.3 Lung lobe segmentation using SegNet

The preprocessed image is presented to the segmentation phase, which is done using SegNet for segmenting the lobes from the preprocessed CXR image. In the COVID-19 detection, SegNet model is employed for COVID-19 detection considering each CXR image segments.

**SegNet**: The SegNet model consists of a network of encoder and decoder with a pixel-wise layer. Hence, the SegNet model generates lobe segments from the preprocessed CXR image. The architectural representation of SegNet is presented in Figure 2. Here, encoder has convolutional layers for performing segmentation. The training procedure is initialized based on the weights trained for the detection. Every layer of encoder has its subsequent decoder layer. The output produced by the decoder is subjected to multiclass soft-max classifier for producing class probabilities for all the values of pixels. In the encoder network, each encoder convolves with the filter bank to form a cluster of feature maps. After that, an element-wise rectified linear nonlinearity (ReLU) is adapted in such a way that the output is made to sub sample. ReLU is utilized to ensure the efficiency of the process and it works more quickly while handling large scale networks. Moreover, a POOL layer effectively reduces the difficulty, and there is no bias or weights to be trained as the input gets processed by combining the local regions of filter. Moreover, an appropriate decoder is adapted to up sample the feature maps with maximal pooling indices so that the feature maps are convolved with trained decoder filter bank for creating dense feature maps. Moreover, the output obtained by the final decoder is presented to the trainable soft-max classifier, which classifies every pixel in a self-regulating way. The soft-max classifier output is represented as \( M \), where \( M \) signifies overall classes. Consequently, the predicted segmentation is equivalent to the class with maximum probability for each pixels. Hence, the SegNet has the capability to produce the segments.
FIGURE 2  SegNet architecture

Let the segmented lobes of input image $S_c$ is expressed as,

$$S_c = \{ s_1, s_2, \ldots, s_a, \ldots, s_𝓁 \}, \tag{2}$$

where $𝓁$ denotes the overall image segments, and $s_a$ denotes the $a$th image segment. The lobe segmentation output is denoted as $S_c$, which is presented as an input to the feature extraction module.

3.4  |  Proposed MLTF from interested lobe regions

The lobe segmentation output $S_c$ is given as an input to the feature extraction phase for obtaining the proposed MLTF from interested lobe regions. The proposed MLTF can be generated by extracting the significant texture features, such as LOOP, LGHP, LBP, LDP, LTP, LDTP, LGP, and HLGP from the interested lobe regions. Figure 3 depicts the illustration of proposed MLTF.

The various effective texture features, such as LOOP, LGHP, LBP, LDP, LTP, LDTP, LGP, and HLGP are explained below as follows.

3.4.1  |  LOOP

This feature illustrates the nonlinear incorporation of the LDP and LBP, respectively. Let us assume the image intensity as $R_z$ at pixel $(P_r, Q_r)$ and the pixel intensity in $3 \times 3$ neighborhood is represented as $R_z(z = 0, 1, \ldots, 7).$ The LOOP value of pixel $(P_r, Q_r)$ is calculated as,

$$f_1 (P_r, Q_r) = \sum_{z=1}^{7} G(R_z - R_{P_r})^{2s_z}. \tag{3}$$

Here,

$$G(P) = \begin{cases} 
1; & P \geq 0 \\
0; & \text{otherwise}
\end{cases} \tag{4}$$

The encoding of rotation invariance to main formulation is done by LOOP feature. Here, $f_1$ represents the LOOP feature.
3.4.2 | LGHP

LGHP feature[^1] finds the modifications in the original pixels of higher order derivative space. Here, the four first order directional derivatives $G_{\eta E}^1$ of image $T(W)$ with $R \times S$ dimension in directions $\eta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ at an arbitrary reference point $W_0$ at distance $E$ is calculated as follows,

$$G_{\eta E}^1(W_0) = T(W_0) - T\left( W_{\left(\frac{\eta}{45}+1\right)E} \right).$$

(5)

where reference pixel in direction $0^\circ$ at a particular distance is signified as $W_{\left(\frac{\eta}{45}+1\right)}$ that is $W_{1,1}$, and $W_0$ specifies the pixel value.

---

[^1]: Reference to the LGHP feature.
The second order $f_{2\nu,E}^2(\cdot)$ at distance $E$ is calculated by performing the encoding operation using encoding function for the pair wise derivatives $C(\ldots)$ and then the patterns encoded are concatenated. In addition, $f_{2\nu,E}^2(\cdot)$ is illustrated as,

$$f_{2\nu,E}^2(W_0) = \left\{ C(G_{\nu,E}^1(W_0)G_{\nu,E}^1(W_0)), C(G_{\nu,E}^1(W_1)G_{\nu,E}^1(W_1)), \ldots, C(G_{\nu,E}^1(W_{8,E})G_{\nu,E}^1(W_{8,E})) \right\},$$  

(6)

where $\eta, \epsilon = 0^\circ, 45^\circ, 90^\circ, 135^\circ$.

The encoding function $C(\ldots)$ in the derivative space at a point $E$ is formulated as,

$$C(G_{\nu,E}^1(W), G_{\nu,E}^1(W)) = \begin{cases} 1 & \text{if } G_{\nu,E}^1(W) > G_{\nu,E}^1(W), \\ 0 & \text{otherwise}. \end{cases}$$  

(7)

Finally, the second order LGHP is formulated as,

$$f_{2\nu}^2(W_0) = \left\{ \text{LGHP}_{E}^2(W_0) | E = 1, 2, 3 \ldots \pi \right\},$$  

(8)

The six binary patterns of nine bits are encoded by LGHP each at a specific distance for various $(\eta, \epsilon)$ pairs. These six patterns are transformed to its corresponding decimal values for generating six LGHP matrices. Finally, the LGHP feature is denoted as $f_2$.

### 3.4.3 LBP

LBP\(^{31}\) is a most common descriptor feature, which is used for capturing the local intensity variations of image and LBP feature has better discriminative character. Let $R_i$ be the image intensity at pixel $(P_y, Q_y)$ and the $3 \times 3$ neighborhood pixel intensity is specified as $R_i (z = 0, 1, \ldots, 7)$. Besides, the LBP value of pixel $(P_y, Q_y)$ is given as,

$$f_3 (P_y, Q_y) = \sum_{z=1}^{7} G(R_z - R_y) \cdot 2^{z},$$  

(9)

where

$$G(P) = \begin{cases} 1 & \text{if } P \geq 0, \\ 0 & \text{otherwise}. \end{cases}$$  

(10)

Here, $f_3$ represents the LOOP feature.

### 3.4.4 LDP

LDP feature\(^{31}\) has a low discriminative power and hence LDP feature is used for COVID-19 detection process. LDP feature is slightly liable to noises when compared with other conventional LBP operators.

$$f_{4k} (P_y, Q_y) = \sum_{z=0}^{7} G(B_z - B_0) \cdot 2^{z}$$  

(11)

where $k$ defines the directional bit responses, $(P_y, Q_y)$ denotes the location of the pixel, $B_z$ signifies Kirsch masks, and $B_0$ implies center directional response.

$$G(P) = \begin{cases} 1 & \text{if } P \geq 0, \\ 0 & \text{otherwise}. \end{cases}$$  

(12)

The LDP feature output is denoted as $f_4$. 
3.4.5 | LTP

LTP-based feature\(^{33}\) is a generalization of LBP patch wise texture feature extractor. The LBP feature is formulated as,

\[
f_{S,A,B}(T_w) = \sum_{z=0}^{B-1} G(T_z - T_w) 2^z,
\]

(13)

where \(T_z\) signifies neighboring pixel, \(T_w\) denotes gray scale value of centering pixel, \(A\) specifies neighborhood radius, \(B\) denotes the overall neighbors. Here, function \(G(T)\) is expressed as,

\[
G(T) = \begin{cases} 
1; T \geq 0, \\
0; Otherwise. 
\end{cases}
\]

(14)

The LTP operator is computed based on the modification of step function and the equation is given as,

\[
G(T) = \begin{cases} 
1; T \geq \theta, \\
0; -\theta < T < \theta, \\
-1; T < -\theta, 
\end{cases}
\]

(15)

where \(\theta\) represents the values of threshold to identify sharpness and the output of LTP feature is denoted as \(f_5\).

3.4.6 | LDTP

LDTP\(^{34}\) is a foremost texture function for extracting texture information from primary axis in every neighborhood. This LDTP feature contains principal information and hence it is used for COVID-19 detection with improved performance.

The eight values of absolute edge response \(L_o\) is calculated based on Kirsch masks for obtaining LDTP feature and the equation is expressed as,

\[
L_o = |U * M_o|.
\]

(16)

where \(U\) denotes the image, \(M_o\) signifies oth Kirsch mask and \(*\) is the convolution operation. The output of LDTP feature is denoted as \(f_6\).

3.4.7 | LGP

The center region of pixel is characterized by two different two-bit binary patterns known as LGP\(^{35}\). In general, LGP exploits gradient values and eight pixel values. The gradient values and center pixel are computed using the correct dissimilarity among the center intensity and closer value of pixel. Furthermore, threshold value is calculated on the basis of average gradient values considering adjoining pixels and thereafter the kernel is used for calculating the closer pixel value for the central region of pixel. Besides, the value of adjacent pixel is "1" if the pixel gradient is greater than the rate of threshold; else it is "0". Moreover, the LGP feature is expressed as,

\[
f_7(P_y, Q_y) = \sum_{z=0}^{B-1} G(B_z - B_k) \cdot 2^z,
\]

(17)

where \(B_k\) denotes the gradient value of central pixel, \(z\) signifies total neighbors, and \(B_z\) implies gradient value of neighbors. The adjoining pixel value \(\tau\) is computed as,

\[
B_z = |\mu - \mu^*|,
\]

(18)

where \(\mu^*\) specifies the locally modified threshold. The LGP extracts invariant samples with the gradient dissimilarity and LGP is not inclined due to its local color variations. The dimensions of color histograms is signified as \([3 \times \phi]\), here \(\phi\) denotes bins. The LGP feature output is represented as \(f_7\).
3.4.8 HLGP

HLGP\textsuperscript{36} is a descriptor feature where harmonic mean (HM) is utilized to capture the adaptive threshold for encoding the gradient value. Because of its value of adaptive threshold, HLGP contains the capability to reduce the outlier’s dilemma. Besides, the harmonic mean of the values of gradient \( B_k \) is calculated by the equation given below as,

\[
B_k = \frac{B}{\sum_{i=0}^{B} f_i}.
\] (19)

Afterwards, all the neighborhood gradient values are compared with threshold \( B_k \) and a binary value of 0 is assigned if it is lesser than \( B_k \), otherwise fix the binary value as 1. Thereafter, the binary numbers are concatenated with respect to any direction and finally give a decimal value corresponding to its binary value. The HLGP feature is computed as,

\[
f_8 (P, Q) = \sum_{i=0}^{B-1} G(B_i - B_k) \cdot 2^i.
\] (20)

Here,

\[
G(P) = \begin{cases} 
1; & P \geq 0, \\
0; & otherwise,
\end{cases}
\] (21)

where the difference between the threshold \( B_i \) and its gradient neighbor value \( B_k \) specifies the gradient value of central pixel, \( z \) denotes total neighbors, and \( B_z \) signifies the gradient value of neighbors. The HLGP feature is denoted as \( f_8 \).

Here, all the various texture features, namely LOOP, LGHP, LBP, LDP, LTP, LDTP, LGP, and HLGP are individually processed to obtain the binary values for each texture features. Thereafter, the obtained binary values of each different texture features are made to perform XOR operation, which forms the 8-bit pixel values.

\[
\text{LOOP + LGHP + LBP + LDP + LTP + LDTP + LGP + HLGP},
\] (22)

\[
f_1 \oplus f_2 \oplus f_3 \oplus f_4 \oplus f_5 \oplus f_6 \oplus f_7 \oplus f_8.
\] (23)

After generating 8-bit pixel values, it finally gives a decimal value corresponding to its binary value to replace all the locations of the features in order to form the final proposed MLTF feature, which is represented as \( F_c \). The proposed MLTF obtained is presented as an input to the COVID-19 detection module.

3.5 COVID-19 detection using proposed IWS-based DeepNet

The feature extraction output \( F_c \) is given as an input to the detection phase, where the process is performed using IWS-based DeepNet. Here, the deep CNN\textsuperscript{37} classifier is employed for COVID-19 detection. The structural optimization of deep CNN is carried out using developed IWS algorithm, which is designed by the combination of IIWO\textsuperscript{38} and SLnO\textsuperscript{39}. Moreover, the deep CNN architecture, and the structural optimization of deep CNN is explained below as follows:

3.5.1 Structure of Deep CNN

Deep CNN\textsuperscript{37} is a kind of deep neural network structure such that the working procedure of deep CNN is same as CNN structure but with several hidden layers. The main aim of using deep CNN is that it poses effective self-learning, fault tolerance ability, layer by layer extraction procedure and has the potential to operate with inadequate data. Figure 4 presents the architecture of deep CNN classifier.

Input layer: The proposed MLTF output \( F_c \) obtained from interested lobe regions is presented as an input for COVID-19 detection process.

Convolutional layer: It includes number of convolution kernels for sharing the weights in such a way that each set of weight sharing connection forms the filter, which is then convoluted with input feature. The feature maps are created by performing convolution across the images. In addition,
the interconnection of neurons with trainable weights and the convolution of trained weights with input is done to create the feature maps. The input of convolution layer is given as,

\[ X = \{X_1, X_2, \ldots, X_a, \ldots, X_b\} \quad (24) \]

where \( b \) signifies number of convolution layer, \( X_a \) is \( a \)th convolution layer of deep CNN classifier. Moreover, the output generated by the convolution layer is formulated as,
\[(X_{d}^{e})_{ij} = (Y_{d}^{e})_{ij} + \sum_{h=1}^{l} \sum_{l=1}^{q} \sum_{t_{z}} (C_{h}^{e} \ast (X_{d}^{e-1})_{i+\text{h}+\text{q},j+\text{t}_{z}}). \]  

(25)

where \((X_{d}^{e-1})_{i+\text{h}+\text{q},j+\text{t}_{z}}\) implies the fixed feature map, \(l\) indicates total feature maps from previous convolution layer, and \(\ast\) denotes the convolution operator. \((Y_{d}^{e})_{ij}\) signifies the network weights optimized by proposed IWS.

**Maximum pooling layer:** Pooling layer exploits activation function in order to assist the simplicity and effectiveness, while managing massive number of networks. The output generated at this layer considering the feature maps is formulated as,

\[X_{d}^{o} = \text{fun} (X_{d}^{e-1}).\]  

(26)

where \(\text{fun}\) denotes the activation function, and this layer helps to minimize the difficulty of COVID-19 detection process.

**Dropout layer:** Dropout is a regularization method that prevents overfitting by discarding the value of some units randomly, which introduces a dropout layer.

**Fully connected layer (FC):** The input presented to FC layer is the output generated from convolutional layer, pooling layer and the FC layer is utilized for combining various kinds of local information. The result generated at this fully connected layer is expressed as,

\[N_{d}^{e} = O(B_{d}^{e}) \text{ with } B_{d}^{e} = \sum_{w=1}^{l} \sum_{i=1}^{q} \sum_{j=1}^{t_{z}} (U_{d,w,ij}^{e}) \cdot (X_{d}^{e-1})_{i,j}.\]  

(27)

Moreover, the structural optimization is done to optimize the convolutional layer, activation layer, maximum pooling layer, and drop out layer. The output obtained at this phase is signified as \(J_{s}\).

### 3.5.2 Proposed IWS algorithm for structure optimization of Deep CNN

This section elaborates the structural optimization of Deep CNN\(^{37}\) classifier using proposed IWS algorithm for COVID-19 detection. Here, the IWS is designed by integrating SLnO\(^{39}\) and IIWO.\(^{38}\) The SLnO\(^{39}\) algorithm is motivated by the hunting characteristics of sea lions. In addition, it is also inspired by whiskers of sea lions’ to find the prey. This algorithm avoids the premature convergence and enables better exploration capability so that the optimal location can be identified. Meanwhile, IIWO\(^{38}\) is an intellectual population-based algorithm, which effectively performs the global searches more speedily in such a way that the population diversity is enhanced. Furthermore, the incorporation of SLnO and IIWO improved the overall system performance and achieves global best solution by minimizing the computational problems. The algorithmic steps of developed IWS are explained below.

**Step 1: Initialization.**

The preliminary phase is to initialize the solution \(Q\) which is expressed as,

\[Q = \{Q_{1}, Q_{2}; \ldots, Q_{u}; \ldots, Q_{v}; 1 \leq u \leq v\}.\]  

(28)

where \(v\) signifies the overall solutions and \(Q_{u}\) represents the \(u\)th solution.

**Step2: Solution encoding representation.**

The solution vector determines the best solution and it has the dimension of \(1 \times 4\), representing number of convolution layer, type of activation function, number of maximum pooling layers, and number of dropout layer. Figure 5 depicts the solution encoding of developed IWS algorithm.

**Step 3: Fitness calculation.**

It is used for computing the optimum solution such that the solution with the least error value is regarded as the optimal solution. The fitness is calculated using Equation (29).

\[\xi = \frac{1}{r} \sum_{k=1}^{r} [O_{k} - J_{s}]^{2}.\]  

(29)

where \(\xi\) indicates the fitness measure, \(J_{s}\) signifies classifier output, and \(O_{k}\) denotes the target output.

**Step 4: Detecting and tracking phase.**
Based on SLnO\textsuperscript{19} algorithm, the sea lions use its whiskers for predicting the size, location, and figure of prey. The sea lion move in the route of target prey and is given as,

\begin{equation}
\bar{Q}(h + 1) = \bar{R}(h) - \bar{C} \bar{L},
\end{equation}

\begin{equation}
\bar{Q}(h + 1) = \bar{R}(h) - |2 \bar{N} \bar{R}(h) - \bar{Q}(h)| \bar{L}.
\end{equation}

Assume, $\bar{R}(h) > \bar{Q}(h)$,

\begin{align}
\bar{Q}(h + 1) &= \bar{R}(h) - (2 \bar{N} \bar{R}(h) - \bar{Q}(h)) \bar{L}, \\
\bar{Q}(h + 1) &= \bar{R}(h)(1 - 2 \bar{N} \bar{L}) + \bar{Q}(h) \bar{L},
\end{align}

By incorporating SLnO with IIWO, the global convergence characteristics can be improved, and optimal solution can be achieved. The standard equation of IIWO\textsuperscript{38} is given as,

\begin{align}
\bar{Z}(h + 1) &= \sigma(h) \times \bar{Z}(h) + \left( \bar{Z}_{\text{best}} - \bar{Z}(h) \right), \\
\bar{Z}(h + 1) &= \bar{Z}(h)(\sigma(h) - 1) + \bar{Z}_{\text{best}}, \\
\bar{Z}_{\text{best}} &= \bar{Z}(h + 1) - \bar{Z}(h)(\sigma(h) - 1).
\end{align}

By substituting Equation (36) in (33) at the target prey $\bar{R}(h)$ of SLnO and also by assuming $\bar{Z}(h) = \bar{Q}(h)$, the equation becomes,

\begin{align}
\bar{Q}(h + 1) &= (\bar{Q}(h + 1) - \bar{Q}(h)(\sigma(h) - 1))(1 - 2 \bar{N} \bar{L}) + \bar{Q}(h) \bar{L}, \\
\bar{Q}(h + 1) &= \bar{Q}(h + 1)(1 - 2 \bar{N} \bar{L}) - \bar{Q}(h)(\sigma(h) - 1)(1 - 2 \bar{N} \bar{L}) + \bar{Q}(h) \bar{L}, \\
\bar{Q}(h + 1) - \bar{Q}(h + 1)(1 - 2 \bar{N} \bar{L}) &= \bar{Q}(h)(\bar{L} - (\sigma(h) - 1)(1 - 2 \bar{N} \bar{L})), \\
\bar{Q}(h + 1)(1 - 1 + 2 \bar{N} \bar{L}) &= \bar{Q}(h)(\bar{L} - (\sigma(h) - 1)(1 - 2 \bar{N} \bar{L})), \\
\bar{Q}(h + 1) &= \frac{\bar{Q}(h)(\bar{L} - (\sigma(h) - 1)(1 - 2 \bar{N} \bar{L}))}{2 \bar{N} \bar{L}}, \\
\sigma(h) &= \left( \frac{W - h}{W} \right)^m (\sigma_{\text{final}} - \sigma_{\text{final}}) + \sigma_{\text{final}} \times g(h),
\end{align}

where $(h + 1)$ implies the next iteration, $\bar{N}$ indicates the random vector within the range $[0,1]$, $\bar{R}(h)$ denotes the position vector of target prey, $\bar{Q}(h)$ denotes the position vector of sea lion, $\bar{C}$ is the distance among target prey and sea lion, $\bar{L}$ represents a parameter that linearly decreases from 2 to 0 for every iterations, $g(h)$ represents chaotic mapping in $h$th iteration, $W$ denotes the maximum iteration, $\sigma(h)$ signifies the standard deviation, $n$ specifies the nonlinear modulation index, $\bar{Z}(h + 1)$ denotes the position of new weed, and $\bar{Z}_{\text{best}}$ signifies the optimal weed present in the overall population.
Step 5: Vocalization phase.

The sea lions interact with each other through vocalizations. When the sea lion finds a prey then, it calls remaining sea lions to surround and harass the prey, which is expressed as,

\[ Q_{\text{leader}} = \left| \left( \bar{T}_1 + \left( 1 + \bar{T}_2 \right) \right) / \bar{T}_2 \right|, \]  

(43)

where \( T_1 \) is the speed of sounds in water, \( T_2 \) signifies the speed of sounds in air, and \( Q_{\text{leader}} \) implies the sound speed of sea lion leader. Accordingly, the sound speed in water is represented as,

\[ \bar{T}_1 = \sin \theta, \]  

(44)

The speed of sounds in air is denoted in Equation (45).

\[ \bar{T}_2 = \sin \psi. \]  

(45)

Step 6: Attacking phase (Exploitation phase).

For modeling the hunting characteristics of sea lions, two different phases, (1) dwindling encircling and (2) circle updating are introduced.

(a) Dwindling encircling.

It is dependent on \( \vec{L} \), that assists the sea lion leader to progress the direction of the track of prey and surround them.

(b) Circle updating.

The sea lions hunt bait ball of fishes starting from edges, which is given as,

\[ \vec{Q}(h + 1) = \vec{R}(h) - \bar{Q}(h) \cdot \cos(2\pi n) + \vec{G}(h), \]  

(46)

where \( | \vec{R}(h) - \bar{Q}(h) | \) denotes the distance among finest solution and search agent, \( | | \) signifies the absolute value, and \( n \) implies the random number between \([-1, 1]\).

Step 7: Searching for prey (exploration phase).

In the phase of exploitation, the optimal search agent updates the sea lion’s position. In phase of exploration, the search agent location is updated in accordance with arbitrarily selected sea lion, which is formulated in Equation (47).

\[ \vec{C} = |2\vec{N} \bar{Q}_{\text{rand}}(h) - \bar{Q}(h)|, \]  

(47)

where \( \bar{Q}_{\text{rand}}(h) \) denotes random sea lion which is selected from present population.

\[ \bar{Q}(h + 1) = \bar{Q}_{\text{rand}}(h) - \vec{C} \cdot \vec{L}. \]  

(48)

Step 8: Evaluation of solution feasibility.

In this step, the solution with best best value of fitness is regarded as finest solution.

Step 9: Termination.

Steps 1 to 8 are repeatedly performed until the finest solution is achieved.

The pseudo code of proposed IWS algorithm is presented in Algorithm 1.

Algorithm 1. Pseudo code of developed IWS

1 Input: Solution set \( Q \), maximal iterations \( h_{\text{max}} \)
2 Output: Optimal solution \( \bar{Q}(h + 1) \)
3 Begin
4 Initialize the solution
5 Select \( \bar{Q}_{\text{rand}}(h) \)
6 Compute the fitness for every search agent by Equation (29)
7 \( \bar{Q}^* \) defines best candidate search agent with optimal fitness
8 If \( h < h_{\text{max}} \)
9 Compute \( \bar{Q}_{\text{leader}} \) using Equation (43)
10 If \( \left( \bar{Q}_{\text{leader}} < 0.25 \right) \)
11 If \( |L| < 1 \)
12 Update the position of current search agent by Equation (46)
13 else
14 Select random search agent \( \vec{Q}_{\text{rand}}(h) \)
15 Update the position of current search agent by Equation (30)
16 End if
17 Update the position of current search agent by Equation (48)
18 End if
19 If search agent does not belong to any \( \vec{Q}_{\text{leader}} \)
20 Go to step 9
21 Else
22 Calculate fitness of search agent using Equation (29)
23 Update \( \vec{Q} \), if best solution exists
24 Return \( \vec{Q} \)
25 End if
26 End if
27 End

4  RESULTS AND DISCUSSION

The experimental analysis of the developed MLTF + IWS-based DeepNet based on the evaluation metrics are described in this section.

4.1  Experimental set-up

The execution setup of the developed MLTF + IWS-based DeepNet is carried out in a PYTHON tool with 4 GB RAM, Windows 10 OS, and Intel I3 processor.

4.2  Dataset description

The dataset utilized for detecting the COVID-19 is DeepCovid dataset.\(^{40}\) This dataset contains 5000 CXR images with their binary labels to detect COVID-19. This dataset remains as a standard to support the research organization. Here, the board-certified radiologist labels the CXR images in class of COVID-19 such that the images with clear indication are utilized for the testing operations. In accordance with board-certified radiologist recommendation, only anterior–posterior images are considered for predicting the Covid-19 disease as lateral images are not appropriate for this function. The anterior–posterior CXR images are evaluated, and the images without any minute radiographic indications of Covid-19 are discarded from dataset.

4.3  Experimental outcomes

The implementation outcomes of developed MLTF + IWS-based DeepNet approach for COVID-19 detection are presented in Figure 6. Figure 6A presents the original CXR image, preprocessed RoI image is shown in Figure 6B, lobe segmented image is specified in Figure 6C. In addition, extracted LOOP feature image is portrayed in Figure 6D, extracted LGHP feature image is shown in Figure 6E,F indicates the extracted LBP feature image, Figure 6G represents the extracted LDP feature image, extracted LTP feature image is presented in Figure 6H. The extracted LGP feature image is portrayed in Figure 6I,J specifies the extracted HLGP feature image and Figure 6K shows the detected output image.

4.4  Performance assessment

This section illustrates the performance assessment of the proposed MLTF + IWS-based DeepNet technique using holdout method. Figure 7 shows the performance assessment of the developed MLTF + IWS-based DeepNet technique in terms of the evaluation measures, namely TPR, TNR, and
The assessment based on TPR is portrayed in Figure 7A. For the training data 70%, the MLTF + IWS-based DeepNet computed a TPR value for the iteration 5 is 0.897, iteration 10 is 0.898, iteration 15 is 0.899, and iteration 20 is 0.901. Figure 7B presents the analysis based on TNR. When considering 80% of training data, the proposed MLTF + IWS-based DeepNet attained a TNR value for the iteration 5, 10, 15, and 20 is 0.865, 0.866, 0.867, and 0.868. The analysis with respect to accuracy is portrayed in Figure 7C. When considering the training data as 60%, the value of accuracy obtained by the developed MLTF + IWS-based DeepNet for the iteration 5 is 0.864, iteration 10 is 0.865, iteration 15 is 0.866, and iteration 20 is 0.867. The performance assessment based on ROC is presented in Figure 7D. When the value of FPR is 2, the developed MLTF + IWS-based DeepNet attained a value of TPR for the iteration 5, 10, 15, and 20 is 0.821, 0.832, 0.843, and 0.854.

4.5 Comparative methods

The various comparative methods exploited for the COVID-19 assessment in comparison with the developed technique are Deep transfer learning,\textsuperscript{25} multimodal covid network III (MMCOVID-NET-III),\textsuperscript{41} Bayesian optimization-based CNN,\textsuperscript{42} Deep CNN,\textsuperscript{26} Gayathri et al.,\textsuperscript{43} Auxiliary GAN,\textsuperscript{27} and Deep LSTM,\textsuperscript{8} randomly initialized CNN (RND-CNN).\textsuperscript{44}
4.6 | Comparative analysis

The comparative assessment of developed technique using holdout method and cross validation is explained in this section.

4.6.1 | Analysis using holdout method

Figure 8 explains the comparative assessment of the developed technique with respect to the metrics, such as TPR, TNR, and accuracy. Figure 8A portrays the assessment in terms of TPR measure. When the training data is 60%, the proposed MLT + IWS-based DeepNet measured a TPR value of 0.890, while the TPR value measured by deep transfer learning, MMCOVID-NET-III, Bayesian optimization-based CNN, Deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, and RND-CNN is 0.722, 0.745, 0.765, 0.790, 0.814, 0.834, 0.851, and 0.875. The analysis using TNR metric is presented in Figure 8B. When the training data is 70%, the value of TNR achieved by the deep transfer learning, MMCOVID-NET-III, Bayesian optimization-based CNN, Deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, RND-CNN, and proposed MLT + IWS-based DeepNet is 0.697, 0.705, 0.714, 0.745, 0.763, 0.780, 0.824, 0.833, and 0.867. Figure 8C illustrates the assessment based on accuracy. For the 80% of training data, the value of accuracy measured by the existing methods, like Deep transfer learning, MMCOVID-NET-III, Bayesian optimization-based CNN, Deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, and RND-CNN is 0.723, 0.733, 0.754, 0.770, 0.799, 0.812, 0.853, and 0.875, whereas the developed MLT + IWS-based DeepNet measured a accuracy value of 0.901. Figure 8D portrays the analysis based on ROC. When considering the FPR value as 2, the proposed MLT + IWS-based DeepNet measured a TPR value of 0.856, while the TPR value measured by deep transfer learning, MMCOVID-NET-III, Bayesian optimization-based CNN, Deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, and RND-CNN is 0.753, 0.760, 0.766, 0.771, 0.799, 0.823, 0.853, and 0.854.
4.6.2 Analysis using cross validation

Figure 9 presents the comparative assessment of the developed technique with respect to the performance measures, like TPR, TNR, and accuracy. Figure 9A depicts the assessment based on TPR. For the K-fold value 5, the TPR value computed by deep transfer learning is 0.752, MMCOVID-NET-III is 0.771, Bayesian optimization-based CNN is 0.792, Deep CNN is 0.813, Gayathri et al. is 0.825, auxiliary GAN is 0.860, deep LSTM is 0.873, RND-CNN is 0.883, and proposed MLTF + IWS-based DeepNet is 0.922. The analysis using TNR measure is presented in Figure 9B. By considering the K-fold value as 6, the TNR value computed by developed MLTF + IWS-based DeepNet is 0.875, while the existing methods, such as deep transfer learning, MMCOVID-NET-III, Bayesian optimization-based CNN, Deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, and RND-CNN measured a TNR value of 0.704, 0.737, 0.741, 0.758, 0.775, 0.808, 0.842, and 0.851. Figure 9C depicts the assessment considering accuracy metric. When considering K-fold value as 7, the accuracy value obtained by the existing methods, like deep transfer learning is 0.734, MMCOVID-NET-III is 0.754, Bayesian optimization-based CNN is 0.763, Deep CNN is 0.780, Gayathri et al. is 0.801, auxiliary GAN is 0.823, deep LSTM is 0.863, and RND-CNN is 0.895, whereas the accuracy value computed by the proposed MLTF + IWS-based DeepNet is 0.913.

4.7 Comparative discussion

This section elaborates the comparative discussion of the developed technique in comparison with the various existing techniques. Table 1 portrays the comparative discussion of the developed technique for the training data 90% and K-fold value 8. For the training data 90%, the value of TPR
FIGURE 9  Analysis of the developed method considering K-fold (A) TPR, (B) TNR, and (C) accuracy

TABLE 1  Comparative discussion of the developed technique

|                              | Deep transfer learning | MMCCOVID-NET-III | Bayesian optimization-based CNN | Deep CNN | Gayathri et al. | Auxiliary GAN | Deep LSTM | RND-CNN | Proposed MLTF + IWS-based DeepNet |
|------------------------------|------------------------|------------------|---------------------------------|----------|-----------------|---------------|-----------|---------|----------------------------------|
| **Hold-out validation**      |                        |                  |                                 |          |                 |               |           |         |                                  |
| TPR                          | 0.769                  | 0.786            | 0.799                           | 0.837    | 0.854           | 0.878         | 0.900     | 0.905   | 0.933                            |
| TNR                          | 0.712                  | 0.724            | 0.754                           | 0.779    | 0.785           | 0.800         | 0.856     | 0.865   | 0.888                            |
| Accuracy                     | 0.745                  | 0.754            | 0.765                           | 0.791    | 0.814           | 0.832         | 0.877     | 0.888   | 0.919                            |
| **Cross validation**         |                        |                  |                                 |          |                 |               |           |         |                                  |
| TPR                          | 0.768                  | 0.791            | 0.804                           | 0.827    | 0.853           | 0.874         | 0.891     | 0.914   | 0.931                            |
| TNR                          | 0.714                  | 0.752            | 0.765                           | 0.770    | 0.799           | 0.818         | 0.855     | 0.875   | 0.890                            |
| Accuracy                     | 0.738                  | 0.765            | 0.770                           | 0.784    | 0.807           | 0.829         | 0.868     | 0.900   | 0.918                            |

computed by Deep transfer learning is 0.769, MMCCOVID-NET-III is 0.786, Bayesian optimization-based CNN is 0.799, Deep CNN is 0.837, Gayathri et al. is 0.854, auxiliary GAN is 0.878, deep LSTM is 0.900, RND-CNN is 0.905, and proposed MLTF + IWS-based DeepNet is 0.933. For the K-fold value 8, the developed MLTF + IWS-based DeepNet measured a TNR value of 0.890, whereas the value of TNR measured by the existing methods, namely Deep transfer learning, MMCCOVID-NET-III, Bayesian optimization-based CNN, deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, and
RND-CNN is 0.714, 0.752, 0.765, 0.770, 0.799, 0.818, 0.855, and 0.875. By considering the 90% of training data, the value of accuracy measured by Deep transfer learning, MMC-COVID-NET-III, Bayesian optimization-based CNN, deep CNN, Gayathri et al., auxiliary GAN, deep LSTM, RND-CNN, and proposed MLTF + IWS-based DeepNet is 0.745, 0.754, 0.765, 0.791, 0.814, 0.832, 0.877, 0.888, and 0.919. Thus, from Table 1, it is clearly illustrated that the developed technique achieved a maximal TPR of 0.933 using training data, maximal TNR of 0.890 using K-fold, and maximal accuracy of 0.919 using training data.

5 | CONCLUSION

This research presents an effective technique to detect COVID-19 disease using MLTF + IWS-based DeepNet based on the CXR images. The performance of the developed method is analyzed using the performance metrics, namely TPR, TNR, and accuracy. The proposed technique attained better performance with the highest TPR of 0.933%, highest TNR of 0.890%, and highest accuracy of 0.919%. The proposed technique offers efficient and acceptable detection performance for the COVID-19 cases. Anyhow, the detection accuracy of the proposed method needs further improvement.

In the future, more details, such as vaccination plan and the effective deep learning classifiers will be considered to enhance the detection process.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in DeepCovid Dataset at https://github.com/shervinmin/DeepCovid, Reference 28.

ORCID
Anandbabu Gopatoti https://orcid.org/0000-0001-6511-3055
Vijayalakshmi P https://orcid.org/0000-0002-1937-9538

REFERENCES
1. Fan Y, Zhao K, Shi ZL, Zhou P. Bat coronaviruses in China. Viruses. 2019;11(3):210.
2. Romeo L, Frontoni E. A unified hierarchical XGBoost model for classifying priorities for COVID-19 vaccination campaign. Pattern Recognit. 2022;121:108197.
3. Rothe C, Schunk M, Sothmann P, et al. Transmission of 2019-nCoV infection from an asymptomatic contact in Germany. New Engl J Med. 2020;382(10):970-971.
4. Qiao Z, Bae A, Glass LM, Xiao C, Sun J. FLANNEL: focal loss based neural network ensemble for COVID-19 detection. arXiv preprint arXiv:2010.16039, 2020.
5. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med. 2020;121:103792.
23. Yu Y, Lin H, Meng J, Wei X, Guo H, Zhao Z. Deep transfer learning for modality classification of medical images. *Information*. 2017;8(3):91.

24. Zeng X, Ouyang W, Yang B, Yan J, Wang X. Gated bi-directional cnn for object detection. Proceedings of European Conference on Computer Vision; October 2016:354-369.

25. Minaei S, Kafieh R, Sonka M, Yazdani S, Soufi GJ. Deep-covid: predicting covid-19 from chest x-ray images using deep transfer learning. *Med Image Anal*. 2020;65:101794.

26. Wang L, Lin ZQ, Wong A. Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Sci Rep. 2020;10(1):1-12.

27. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR. Covid GAN: data augmentation using auxiliary classifier GAN for improved covid-19 detection. *IEEE Access*. 2020;8:91916-91923.

28. Singh AK, Kumar A, Mahmud M, Kaiser MS, Kishore A. COVID-19 infection detection from chest X-ray images using hybrid social group optimization and support vector classifier. *Cognit Comput*. 2021;1-13. https://doi.org/10.1007/s12559-021-09848-3

29. Bradley D, Roth G. Adaptive thresholding using the integral image. *IEEE Trans Pattern Anal Mach Intell*. 2017;39(12):2481-2495.

30. Chakraborty T, McCane B, Mills S, Pal U. LOOP descriptor: local optimal-oriented pattern. *IEEE Signal Process. Lett*. 2018;25(5):635-639. https://doi.org/10.1109/LSP.2018.2817176

31. Freitas PG, Akamine WY, Farias MC. No-reference image quality assessment based on statistics of local ternary pattern. Proceedings of the 8th International Conference on Quality of Multimedia Experience (QoMEx); 2016:1-6.

32. Jun B, Kim D. Robust face detection using local gradient patterns and evidence accumulation. *Pattern Recognit*. 2012;45(9):3304-3316.

33. Habiba U, Howlader MR, Faisal RH, Rahman MM. Hlgp: a modified local gradient pattern for image classification. Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE); February 2019:1-6.

34. Tu F, Yin S, Ouyang P, Tang S, Liu L, Wei S. Deep convolutional neural network architecture with reconfigurable computation patterns. *IEEE Trans Very Large Scale Integr (VLSI) Syst*. 2017;25(8):2220-2233.

35. Misaghi M, Yaghoobi M. Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller. *J Comput Des Eng*. 2019;6(3):284-295.

36. Padmapriya T, Kalaiselvi T, Priyadharshini V. Multimodal covid network: multimodal bespoke convolutional neural network architectures for COVID-19 detection from chest X-ray’s and computerized tomography scans. *Int J Imaging Syst Technol*. 2022;32:704-716.

37. Gayathri JL, Abraham B, Sujarani MS, Nair MS. A computer-aided diagnosis system for the classification of COVID-19 and non-COVID-19 pneumonia on chest X-ray images by integrating CNN with sparse autoencoder and feed forward neural network. *Comput Biol Med*. 2022;141:1-10.

38. Atitallah SB, Driss M, Boulla L, Ghézala HB. Randomly initialized convolutional neural network for the recognition of COVID-19 using X-ray images. *Int J Imaging Syst Technol*. 2021;32:55-73.

How to cite this article: Gopatoti A, P V. Multi-texture features and optimized DeepNet for COVID-19 detection using chest x-ray images. *Concurrency Computat Pract Exper*. 2022;34(22):e7157. doi: 10.1002/cpe.7157