Automatic Detection of Covid-19 in CT Images Using an Optimized Deep Neural Network Classifier

[1]G. Prabakaran, [2]Dr. K. Jayanthi

[1]Assistant Professor/Programmer, Department of Computer and Information Science, Annamalai University.
[2]Assistant Professor, Department of Computer Application, Government Arts College, C-Mutlur, Chidambaram.

Email: [1]tvgprabu@gmail.com, [2] jayanthirab@gmail.com

Abstract: An exact and quick finding of Covid-19 patients plays a significant part in the initial period of medicinal treatment and prevention. Automatic recognition of COVID-19 cases using CT images may help lessen the effect of this infection on human civilization. In recent years, computer vision is the major solution for diagnosing the covid-19 disease by using CT images. Besides, many researchers had presented efficient artificial intelligence techniques for identifying Covid-19 disease. Nevertheless, the accuracy and time consumption of the model further to be improved. Thus, in this work, we propose a Covid-19 disease classification using a hybrid deep neural network with handcrafted features. The proposed approach consists of three stages namely, pre-processing, handcrafted feature extraction, and classification. Initially, the images are given to the pre-processing stage to remove the noise present in the input images. Then, we extract the hand-crafted features (Gray-Level Co-Occurrence Matrix (GLCM)) from each image. After that, the pre-processed image is given to the input of the optimized deep neural network classifier to classify an image as normal or abnormal. The proposed optimized deep neural network is a combination of a convolution neural network (CNN) and adaptive artificial jelly optimization (A_2JO) algorithm. To enhance the performance of classifier accuracy, the extracted handcrafted features are fused with the hybrid deep neural network. The performance of the proposed approach was analyzed based on different metrics and performance compared with other techniques.
Keywords: COVID-19, artificial jelly optimization, convolutional neural network, hybrid deep learning network, and handcrafted features

1. Introduction

The novel coronavirus generally called Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-COV-2) is responsible for creating COVID-19 disease in the human body. General symptoms of COVID-19 are fever, cough and respiratory system disease, and pneumonia also major symptoms [1]. Normally, pneumonia is named as the infection which causes inflammation of air sacs that happens in the lungs for oxygen transfer. Another path of pneumonia infection is bacteria, fungi, and different viruses. The reason behind the severity of chronic diseases such as aging people, smoking, weak immune system or impaired immune system, asthma, or bronchitis [2-4]. The infected people are treated related to the infected part of the human body. Moreover, antibiotics, fever reducers, pain relievers, and cough medicine are provided by a doctor to patients related to their symptoms [5].

If the person is badly hurt, they must be taken to the hospital and receive care in the intensive-care unit (ICU), maybe with the aid of a ventilator. Due to COVID-19's intensity and swifter spread, it has evolved into a pandemic [6]. The fact that more people are affected each day as a result of the necessity to supply mechanical ventilators for severely ill patients admitted to intensive care units (ICU) has a greater impact on the healthcare sector [7,8]. Therefore, the number of Inpatient beds also needs to be dramatically raised. In the case mentioned above, a correct assessment is essential for effective treatment, which relieves the burden on health care.

With impressive success in many medical fields, AI technology has progressed rapidly in recent years as a diagnostic support technology. Deep learning as an AI method has exhibited remarkable clinical value in the use of CT scans to support the investigation of lung illnesses. Deep learning can automatically identify features linked to clinical outcomes from CT images because of its strong feature-learning capabilities [9]. Recent research has demonstrated that radiologists and clinicians can treat patients who have COVID-19 by using CT scanning to build an AI system to detect this virus. Different types of deep learning approaches [10]are developed by authors for the automatic prediction of COVID 19 such as DNN, RNN, DBN, and CNN. Among them, CNN is achieving efficient outcomes in terms of accuracy. In the CNN, optimal parameters are selected by using the optimization algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimization (GWO), and so on.

The main contribution of the research is presented as follows,

- This paper proposes a Covid-19 disease classification using an optimized deep neural network with handcrafted features. The proposed approach consists of three stages namely, pre-processing, handcrafted feature extraction, and classification.
- Initially, the images are given to the pre-processing stage to remove the noise present in the input images. Then, we extract the hand-crafted features (GLCM)) from each image. After that, the pre-processed image is given to the input of the optimized deep neural network classifier to classify an image as normal or abnormal.
- The proposed optimized deep neural network is a combination of a CNN and A_2JO algorithm. To enhance the performance of classifier accuracy, the extracted handcrafted features are fused with the optimized deep neural network.

The remaining portion of the article is pre-panned as follows; section 2 gives the related work of the conventional COVID-19 prediction. The proposed methodology explanation is given in section 3. The outcomes of automatic COVID-19 predictions are presented in section 4. The summary of the paper is presented in section 5.

2. Related works

A different deep learning approach is developed for COVID-19 prediction by using CT images. Few works are reviewed in this section.

Dheyaaahmedibrahim*et al.*, [11] have presented a hybrid deep-learning model to identify infected individuals with COVID-19 based on their lung CT images. Additionally, this research developed a system to generate a reliable COVID-19 prediction network using different layers initiating with the segmentation of the lung CT scan image and ending with disease prediction. The first phase of the architecture initiates with a projected method or lung segmentation which depends on a no-threshold histogram-related image segmentation technique. Afterward, the grab cut technique was utilized for enhancing segmentation in the post-segmentation technique and reducing under and over-segmentation issues.

Muhammad Umer*et al.*, [12] have presented a CNN which extracts the features from the X-ray image for the prediction of COVID-19. To identify the edges, three filters are consumed to achieve the desired segmented target with the infected area of the X-ray. To achieve the smaller size of the training database, the karas image data generator class was utilized to create ten thousand augmented images.

Mohit Kumar *et al.*, [13] have presented a novel technique named a hybrid convolutional neural network (HFCNN) which combined Recurrent Neural Network (RNN) and Convolutional neural network (CNN). This method was utilized for a finding of COVID-19 with the consideration of chest X-ray images. The transfer learning technique, defined as slope-weighted activation class planning was utilized to display images responsible for considering decisions. The study was contrasted with remaining CNN such as DenseNet, VGG-19, SqueezeNet, ShuffleNet, and Inception V3.

Aram Ter-Sarkisov*et al.*, [14] have presented a COVID CT mask net model which identifies COVID-19 in chest CT scans. This model operates in two phases, in the initial phase, Mask R CNN is learned to detect and localize two kinds of lesions in images. In the next stage, these detections are fused to detect the complete input image. To manage the solution to COVID-19, the data split was computed from the dataset of chest CT scans.

HamzehAsgharnezhad*et al.*, [15] have presented a Deep Neural Network for detecting COVID-19 in medical images. In this technique, three uncertainty quantification methods are consumed for COVID-19 prediction. This method was a theory of uncertainty estimates. Based on this theory, the network about chest X-ray images outperforms networks about general image database like ImageNet. The generalization power of this network is always questionable because small datasets were utilized.

3. Proposed System Model

In recent years, COVID-19 leads to severe pneumonia and it is computed to generate a high impact on the healthcare system. The specific requirement is an early prediction of COVID-19 from the CT images. To empower the early prediction of COVID-19, the automatic efficient method is developed in this paper. The complete architecture of the proposed method is given in figure 1.

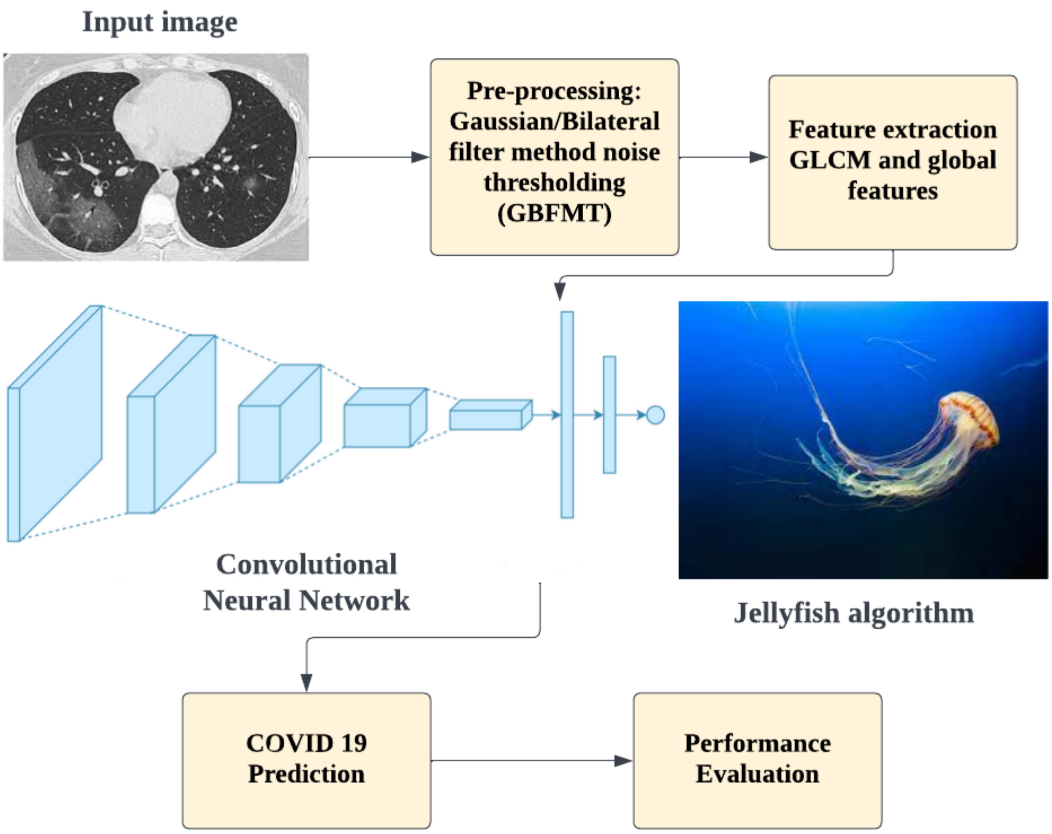


Fig 1: Complete architecture of the proposed method

Initially, the dataset is gathered for validation purposes. The collected database may be containing noise which reduces the prediction accuracy. So, the pre-processing is achieved by using GBFMT. After that, the required features are extracted by using different features such as GLCM and global features. The extracted features are utilized to predict COVID-19 from the images. The extracted features are sent to the optimized deep learning approach. This approach is a combination of CNN and A_2JO. In the jellyfish algorithm, the local optima problems are solved by using the levy flight approach. The complete explanation of the proposed approach is presented in the below section.

3.1. Pre-processing: Gaussian/Bilateral filter method noise thresholding (GBFMT)

The wavelet thresholding and spatial domain filtering are implemented in the Bayesian domain. This technique is introduced to reduce noise from the input images [16]. Method noise is the variation of the denoised image attained from a specific denoising algorithm and the original image.

$$Method\ noise = b - I_s \qquad (1)$$

Here, b is defined as the original image and I_s is defined as a denoising algorithm. Gaussian white additive noise with known zero mean and variance parameters is presumed to achieve the corrupted image. The gaussian noise variance is varied for various noise levels. This noisy image is filtered by using a bilateral or gaussian filter. In the gaussian filter, the image texture is smoothed with the best flattening of harmonics. In a bilateral filter, noise is averaged with the retention of transitions or sharp edges. After this, method noise is attained by using spatial filtering. It consists of the noise that is retained after filtering and image edge data that is suppressed a noise.

$$MN = N + D \qquad (2)$$

Additionally, noise is split into Bayes thresholding and wavelet coefficients are applied. After that, the final image is achieved by adding the outcome of the gaussian filter or bilateral filter with the threshold coefficients. The filtered image is utilized to extract the features from the filtered image.

3.2. Hand crafted Feature extraction

Feature extraction is an important operation for classifying COVID-19. These classes may be insensitive to unrelated changes in the input data. This technique computes the parameter sets which manage the shape of an image precisely and uniquely. In this stage, each image is signified by a feature vector and its identity. Its objective is to reduce input data by managing essential features which differentiate a single input sample from another sample. The features are utilized in COVID-19 prediction.

3.2.1. GLCM features

The statistical method, GLCM utilized the textures which concern the spatial relationship between pixels. The GLCM functions typify the image texture by calculating pixel pairs with specific parameters and in specified spatial variation [17] image transpire through creating a GLCM feature and through extorting statistic measures as of this matrix. The GLCM defines the spatial relationship pixels at each intensity through viewing variations of gray levels V and U at a desired distance D at a defined angleθ.

Contrast: It is a gauge of the intensity contrast among a pixel with its neighbours in the complete image. The co-occurrence matrix of the coordinates in the gray level pixels is presented as follows,

$$Contrast = \sum_{U,V} |U - V|^2 P(U,V) \qquad (3)$$

Here, $P(U,V)$ is defined as the element in the co-occurrence matrix, U and V is defined as gray level pixels.

Correlation: It is a gauge of the pixels correlated with their neighbor in the complete image.

$$Correlation = \sum_{U,V} \frac{(\mu - \mu_U)(V - \mu_V)P(U,V)}{\sigma_U \sigma_V} \qquad (4)$$

Here, $\sigma_U \sigma_V$ is defined as a standard deviation of $U$and V, μ_V is defined as the mean of U, μ_U is defined as the mean of V.

Energy: It is a gauge of data. The energy may be a negative gauge that is to be reduced and a positive gauge to be increased. It is a gauge of a complete of squares of related variables of GLCM. The energy is calculated by using the below equation,

$$Energy = \sum_{U,V} P(U,V)^2 \qquad (5)$$

Homogeneity: It is the gauge of the intimacy of the dissemination of variables in GLCM to the diagonal. This is calculated by using the below formulation.

$$Homogeneity = \sum_{U,V} \frac{P(U,V)}{1 + |I - J|} \qquad (6)$$

3.2.2. Variance, mean and standard deviation

These features define a complete image. These features are attractive due to their generated compact image definition. Here, each image resembles a point in a huge dimension space.

Mean: It is a feature set that defines the arithmetic average of the specific set. The mean operation is computed as follows,

$$Mean\ \mu = \frac{\sum PI}{n} \qquad (7)$$

Here, n is defined as the total number of pixels, and PI is defined as pixels.

Standard deviation: It is defined as a gauge and used to calculate the variation of a data parameter or sum of dispersion of data value pairs. It is calculated by using below equation,

$$SD = \sqrt{\frac{\sum_{I=1}^{N}\left(Pi - \overline{P}\right)^2}{n-1}} \qquad (8)$$

Here, n is defined as the total number of pixels, and PI is defined as pixels.

Variance: It is a dimension of the spread pixel parameters in the feature set. The variance gauges how far each piece of information is from the mean. It is computed as follows,

$$Variance = \sum \frac{\left(Pi - \overline{P}\right)^2}{n-1} \qquad (9)$$

3.3. Optimized deep neural network

The optimized deep neural network is designed for COVID-19 prediction. This neural network is a combination of a convolutional neural network and A_2JO. In the convolutional neural network, the optimal parameter is selected by using A_2JO. This adaptive technology is a jellyfish algorithm and levy flight function process. This levy light function is utilized to enhance the convergence analysis of the jellyfish algorithm. The complete process of convolutional neural network, jellyfish algorithm, and levy flight operation are presented in the below sections.

3.3.1. Convolutional neural network

CNN is defined as the deep learning method it is classified as a multilayer perceptron's ANN. Additionally, this method is utilized combined classification and regression problems. The CNN architecture is designed with three kinds of a layer such as a convolutional layer, a pooling layer, and a fully connected layer. In the convolutional layer, it is equal to a neural network every neuron is attached to a cluster of neurons in the last connected layer. The managing of attaching the neurons is with a variable defined as kernel function [18]. In any convolutional layer, it is several kernels that act as filters to feature extract of the input signals. There is a pooling layer that is responsible to input down samples to reduce overfitting and decrease the space dimension. In the final stage, a fully connected neural network is designed it is utilized to regression or classify in supervised learning.

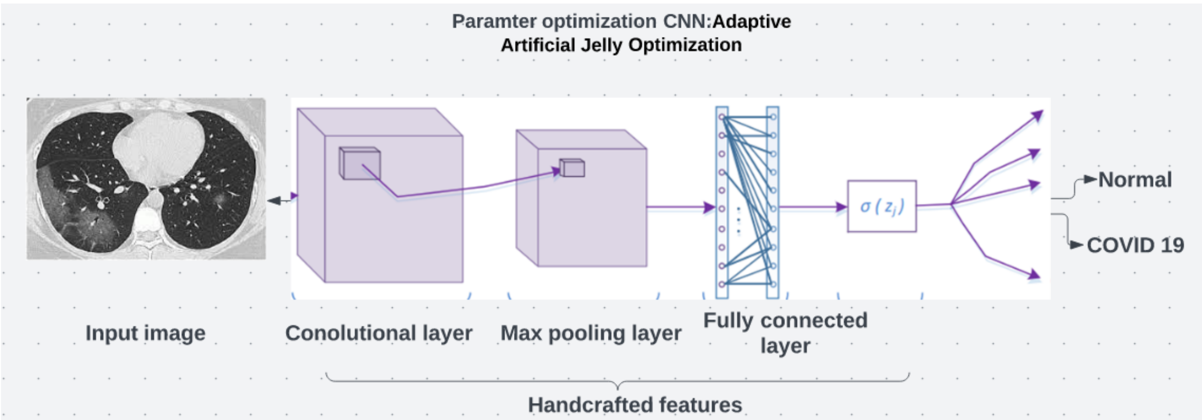


Fig 2: Convolutional Neural Network Architecture

To design a CNN architecture, some important theory is considered.

Kernel: The kernel is designed by using matrices and operates as a filter in a convolutional layer. It is applied to a tiny section of the information matrix and generates a convolution matrix. They are utilized to sharpen the features of the input data. The kernel is a size of $k * k$ and it is applied in the input data of $n * n$, the outcome will be a dimension of

$$((n-k)/(S+1)*((n-k)/(s+1) \qquad (10)$$

Here, S is defined as the input matrix of the moving kernel.

The output is described in the matrix t with the specified dimension $n*n$ that is entry of I, J is computed as follows,

$$t^{I,J} = \sum_{n=1}^{k}\sum_{m=1}^{k}\left(m_{K*K} \diamond z_{k*k}^{I,J}\right)^{n,m} \qquad (11)$$

Here, $z_{k*k}^{I,J}$ is defined as the input data with a submatrix of z with the dimension of $k*k$, $\diamond$ is defined as the Hadamard matrix product of element-wise and m_{K*K} is defined as a kernel matrix with a specified size of $k*k$. This output matrix is presented as follows,

$$z_{k*k}^{I,J} = \begin{bmatrix} z_{I-Hm\ J-H} & \cdots & z_{I-Hm\ J+H} \\ \cdots & \cdots & \cdots \\ \cdots & Z_{I,J} & \cdots \\ \cdots & \cdots & \cdots \\ z_{I+Hm\ J-H} & \cdots & z_{I+Hm\ J+H} \end{bmatrix} \qquad (12)$$

Here, K is taken as an odd number and it is a kernel size and $H = \frac{K-1}{2}$. Moreover, it is not obligatory. In this research, the above equation, K is considered an odd number.

ReLu: This is called an activation function for every neuron which is utilized in the convolutional layer output. It is formulated as follows,

$$r^{I,J} = MAX(0, t^{I,J}) \qquad (13)$$

Here, $t^{I,J}$ is defined as the input that is considered as output layer of the convolution layer, $r^{I,J}$ is defined as the output of the ReLu function. Here, a lot of operations are utilized in the activation function, for example, sigmoid and tangent. Moreover, the most advantage of using ReLu contrasted with the others and it is a simple computation [19].

Max pooling: This layer is utilized to down sample the data from the last layer. The aim of utilizing a down sampling layer is the large dimension of the space. The high space is a complicated and higher computational cost. So, it is wise to reduce the dimension of the matrices. The method is keeping the strongest one and omits the weaker neurons by utilizing this max-pooling layer.

Softmax layer: This is a final layer of a conventional CNN that must be FCNN, an activation function at the last of this layer is required. This operation can vary related to the aim of the CNN design which is a classification of COVID-19. The benefit of utilizing this function is to assist the network and converge to the specified class more accurately and smoothly.

$$\sigma(J) = \frac{Exp(Z_J)}{\sum_{P=1}^{P} Exp(Z_P)} \qquad (14)$$

Here, P is defined as the count of neurons in the final layer of FCNN, Z_J is defined as the output of the jth neuron of FCNN, $\sigma(J)$ is defined as the output of this Softmax layer. The main aim is to classify COVID-19 from its features. To achieve the COVID-19 prediction, the CNN is utilized from the benchmark input images.In the CNN, the initial structure is designed with a convolutional layer, a max pooling layer, and FCNN. The input features are sent to a convolutional layer with kernels. In this research, the kernel size is considered as off. The outputs from convolved kernels are routed by using a ReLu activation function. The upcoming layer is a max pooling layer with pooling operation. This layer is responsible to extract the efficient features. The final layer is an FCNN with a Softmax function that is utilized to classify the COVID-19 data. In CNN, the optimal parameters are selected by using the A_2JO algorithm. A detailed explanation of this algorithm is presented inthe below section.

3.3.2. Adaptive Artificial Jelly Optimization

This optimizer is defined as a metaheuristic algorithm and created by using a jellyfish performance in the ocean. Additionally, the spark characteristics of jellyfish consist of their motion to the ocean current or swimming in a swarm. Time control management is consumed for switching between these movements [20]. This algorithm is developed based on three conditions,

- ❖ **Condition 1:** The jellyfish follow the ocean current during the travel process in the ocean. After that, managing the switching among the time control mechanism and types of motion is utilized.

- **Condition 2:** It is validated which jellyfish get attracted to the specified location where the required food is presented with a huge amount.
- **Condition 3:** The efficient CNN parameter computes the specific location where a huge amount of food is present.

First, food is searched for by jellyfish in the ocean. The jellyfish moves to a high quantity of food presented in the ocean location. So, the high quantity of food place is considered as the jellyfish location. The CNN parameter manifests various locations where the jellyfish visited, where the quantity of food is present in the specified location. The proportion of food is contrasted with every iteration. At last, the global optimal position is computed where the highest quantity and optimal quantity of food is available.

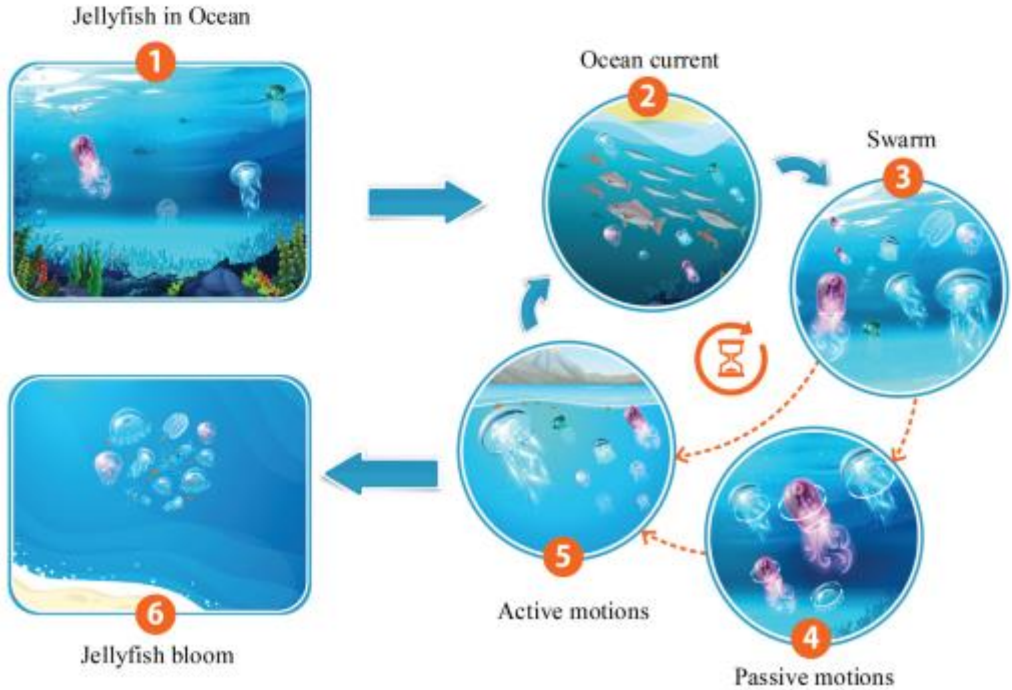


Fig 3: General Behaviour of jellyfish algorithm

Algorithm 1: Pseudocode of the A_2JO

Initialization by CNN parameters
Evaluate the fitness function (classification error)
select the current best fitness (best jellyfish location)
For $\boldsymbol{I} = \mathbf{1}{:}\boldsymbol{N}$ pop do
Compute the time control
If $\boldsymbol{T_C} >= \mathbf{0.2}$
Jellyfish follow ocean current
Else
Jellyfish move inside the swarm
If $\boldsymbol{RAND}\ (\mathbf{0},\mathbf{1}) > (\mathbf{1} - \boldsymbol{T_C})$
Jellyfish choose passive motions
Else
Jellyfish choose active motion
Empower convergence by levy flight operation
End if
End if
End for
End

Step 1: Initialization Process

The initialization of the position of jellyfish is considered as the random variable of CNN parameters. Additionally, the high quantity of food location is considered as the present best location. In the jellyfish algorithm, two motions are performed such as passive or active motion. At first, active motion is managed by jellyfish when the swarm is just achieved. Afterward, passive motion is considered. Jellyfish movement around their way is defined as active motion and its position related to every location is updated by using the below equation,

$$\alpha_l^{T+1} = \alpha_l^T + RAND(0,1) \times RAND(0,1) \times \mu \qquad Here, l = 1,2, \dots 5 \qquad (15)$$

Here, μ is defined as the mean location of the whole jellyfish. α_l^T is defined as the hyperparameter of CNN. The five hyperparameters are considered in this research to reduce the classification error rate. This parameter is presented in table 1.

Table 1: CNN Hyperparameters and limits

S. No	Parameter	Upper bound	Lower bound
1	Gradient threshold	10	0.1
2	Gradient threshold method	3	1
3	L2 regularization	0.01	0.00001
4	Solver	3	1
5	Learning rate	0.01	0.00001

Step 2: Fitness function

In this research, the CNN parameter is selected with the assistance of the A_2JOalgorithm. For that reason, the efficient fitness function is considered a global classification error of the population. This formulation is presented as follows,

$$F(Hm) = \sum_J^n E_J(Hm) \cdot \left(\sum\nolimits_I^{\mu+\lambda} E_J(Hi)\right)^K, Hm, Hi \epsilon P_{\mu+\lambda}^T \qquad (16)$$

Here,$E_J(Hi)$is defined as the classification error, K is defined as a penalty exponent, n is defined as the dataset size, Hm is defined as a hyperparameter related to the CNN of the current interim population. The penalty exponent is related to the classification error of the complete population. Based on the fitness function, the parameter of CNN is selected which reduces the classification error in COVID-19 prediction.

Step 3: Passive motion or active motion

The remaining jellyfish (M)moves to passive motive contrast with the jellyfish(L) who doing active motion. Passive motion is managed to compute the position of motion by choosing a vector and random position from the jellyfish. The quantity of food at the jellyfish M location is higher than the quantity of food at the jellyfish L location. So, L jellyfish moves to the M and$L$ contain high food means, it will move away from its food location to compute the remaining best location. so, every jellyfish can achieve a high amount of food by transferring in a specific path. Thus, the path in which to compute the global optimal position iteration is achieved in different directions of motion [21]. Based on that, the optimal position is upgraded.

$$Direction = \begin{cases} \alpha_M^{(T)} - \alpha_L^{(T)} & if\ F(\alpha_L) \geq F(\alpha_M) \\ \alpha_L(T) - \alpha_M(T) & if\ F(\alpha_L) < F(\alpha_M) \end{cases} \qquad (17)$$

Here, F is defined as the fitness function of classification error minimization of location α. To achieve successive iteration, the optimal position of jellyfish is achieved by using the below equation,

$$\alpha_L^{T+1} = \alpha_L^T + RAND(0,1) \times Direction \qquad (18)$$

Step 4: A time control mechanism

The time control technique is used for validating the type of motion over time. It assists to manage passive and active motion and evaluate the motion of jellyfish if they are pointing to an ocean current. The time control function and constant variable are presented in the time control method. It contains a constant parameter which is equivalent to 0.5 as it is the mean parameter of 1 and 0. The time control function is a random variable that varies from 0 to 1 and it is computed based on the below formulation.

$$T_C = \left|2 \times \left(1 - \frac{T}{MAX\ iteration}\right) \times (2 \times RAND(0,1) - 1)\right| \qquad (19)$$

Here, $MAX\ iteration$ is a maximum number of iterations and T_C is defined as time-based iterations. Jellyfish move to the position where a high amount of food is presented, leading to the creation of a swarm. Inside the swarm, jellyfish move to another ocean current and remain jellyfish are created because of the variation of time. This change empowers variation of wind direction and temperature.

If $T_C \geq 0.5$, jellyfish point to an ocean current. It is computed by the below equation,

$$Current = \alpha^* - \beta \times RAND(0,1) \times \mu \qquad (20)$$

Here, μ is defined as the mean location of all jellyfish, and β is defined as the distribution coefficient. If $RAND(0,1) > (1 - T_C)$ it moves to passive motion.Similarly, if $RAND(0,1) < (1 - T_C)$ it moves to active motion.If $RAND(0,1) < (1 - T_C)$ jellyfish start to travel inside the swarm, wherein they follow the passive or active motion.

Step 5: Levy flight based empower convergence

Empowering the convergence speed to a high parameter and creating the algorithm is independent at local optimal conditions. This local optimum condition may be decreasing the performance of the jellyfish algorithm [21,22]. To reduce these problems, the levy flight is considered this avoid premature convergence and assists to provide enormous divergence in the initial population. The levy flight operation is presented below equation,

$$l(S,\gamma,\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} exp\left[-\frac{\gamma}{2(S-\mu)}\right]\frac{1}{(S-\mu)^{3/2}} & if\ 0 < \mu < s < \infty \\ 0 & if\ s \leq 0 \end{cases} \qquad (21)$$

Here, $\gamma > 0$ is defined as the scale parameter, and μ is defined as the shift parameter or location parameter. Based on the levy flight operation, the convergence is enhanced by reducing the effect of local optima issues.

Step 6: Termination Condition

The final boundary condition is validated and the food quantity at the new position is computed. In every iteration, the jellyfish location is upgraded and continued to achieve the maximum iteration. Finally, stops the operation and saves the optimal parameter of CNN.

4. Performance Evaluation

The proposed method is analyzed and validated in this section. This methodology is developed for COVID-19 prediction from the CT images. To justify the performance of the proposed technique, it is analyzed with statistical measurements such as F_Measure, precision, recall, sensitivity, specificity, and accuracy. To validate the proposed methodology, it is compared with the traditional approaches such as CNN, RNN, DBN, and DNN. The proposed method is analyzed with three phases pre-processing, feature extraction, and classification. The proposed method is implemented in Python and performances are evaluated. The proposed model parameters are presented in table 1. To validate the performance of the proposed method, the UCSD-A14H dataset is used [23] which consists of 349 COVID-19 images and 397 Non COVID 19 images. From the database, 70% of the data is consumed for the training phase, and 30% of the data is consumed for the testing phase. The sample input image of COVID-19 and non-COVID-19 is illustrated in figure 3.

Table 1: Proposed OCNN Model parameters

S. No	Description	Parameters
1	Momentum	0.9
2	Learn rate drop period	5
3	Learn rate drop factor	0.2
4	Initial learn rate	0.05
5	Maximum epochs	15
6	Mini batch size	500
7	Number of search agent	50
8	Number of iterations	100
9	Upper bound	10
10	Lower bound	-10

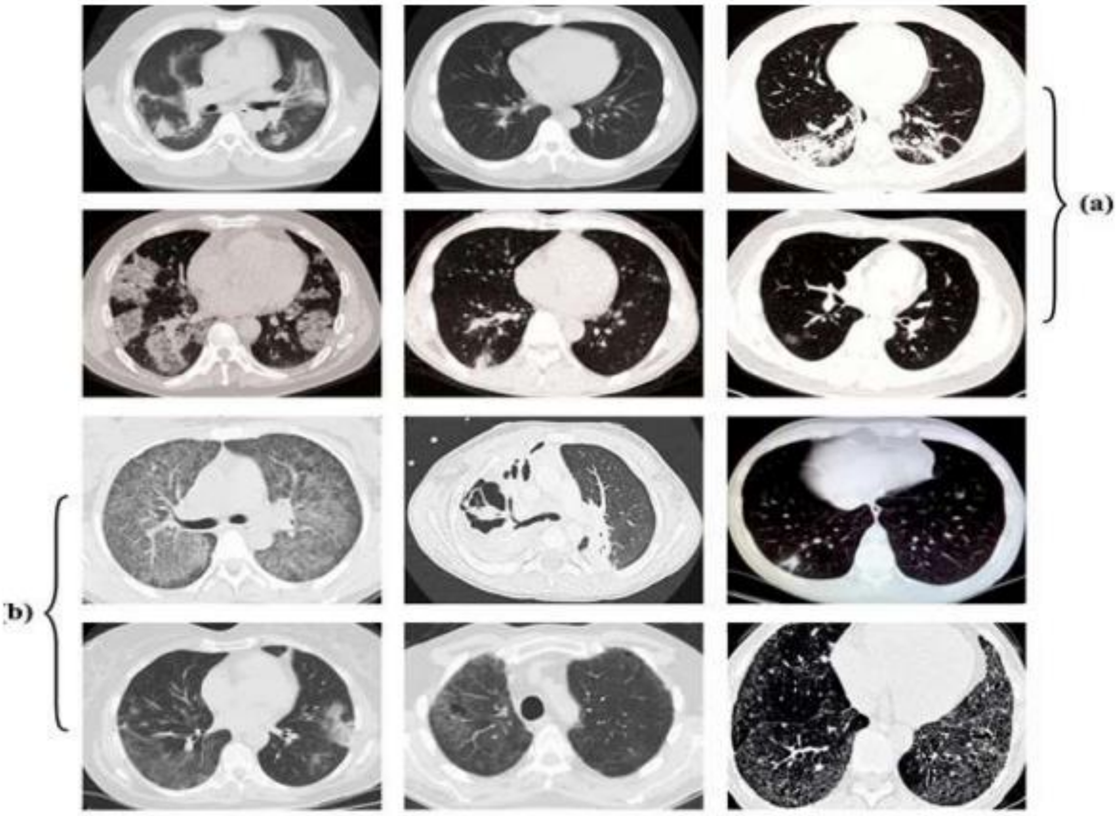
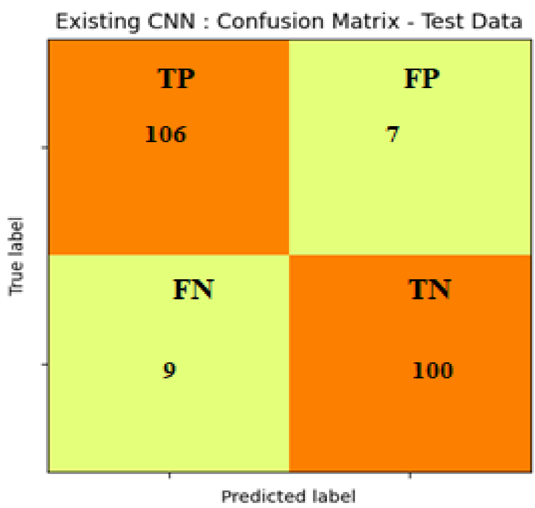
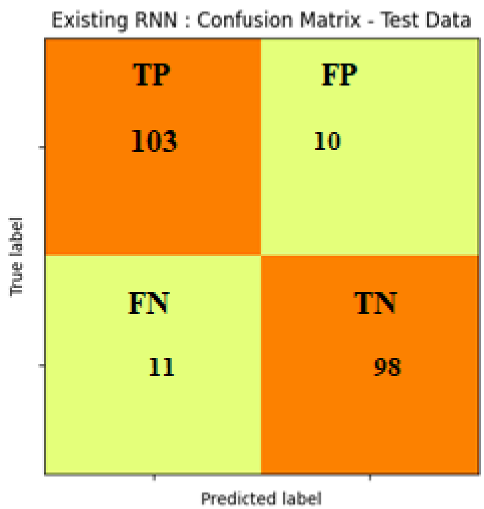


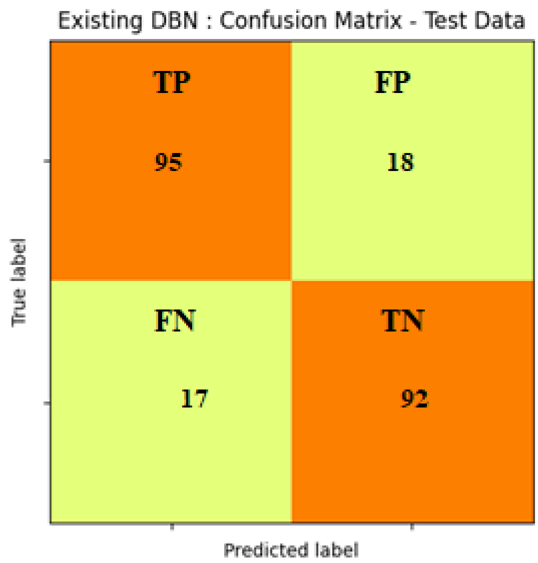
Fig 3: Sample Images a) COVID b) non-COVID



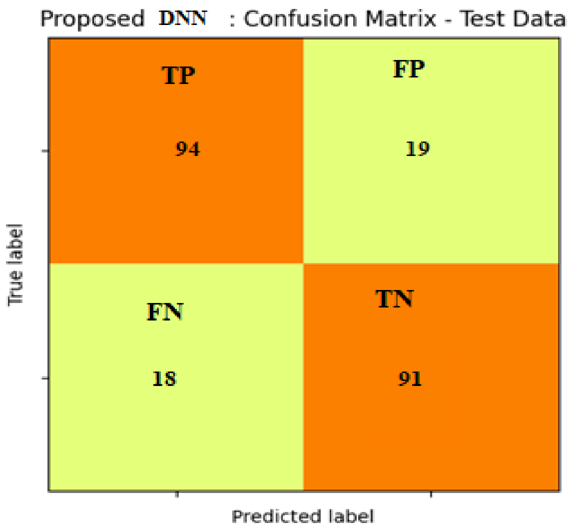
(a)



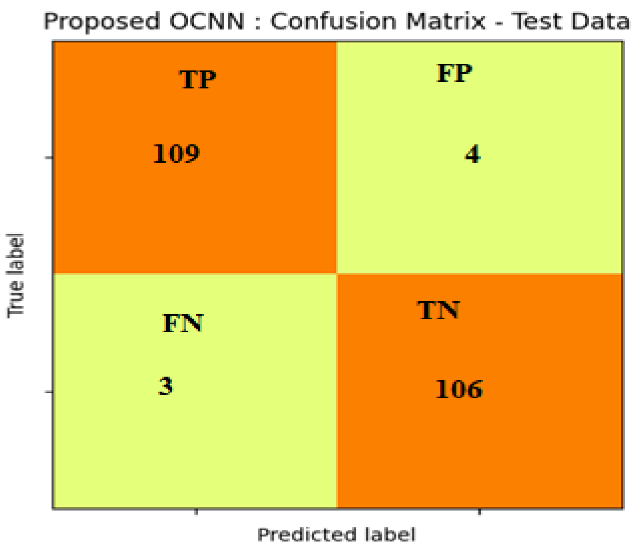
(b)



(c)



(d)



(e)

Fig 4: Confusion matrix of (a) CNN, (b) RNN, (c) DBN, (d) DNN, and (e) Proposed OCNN model

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

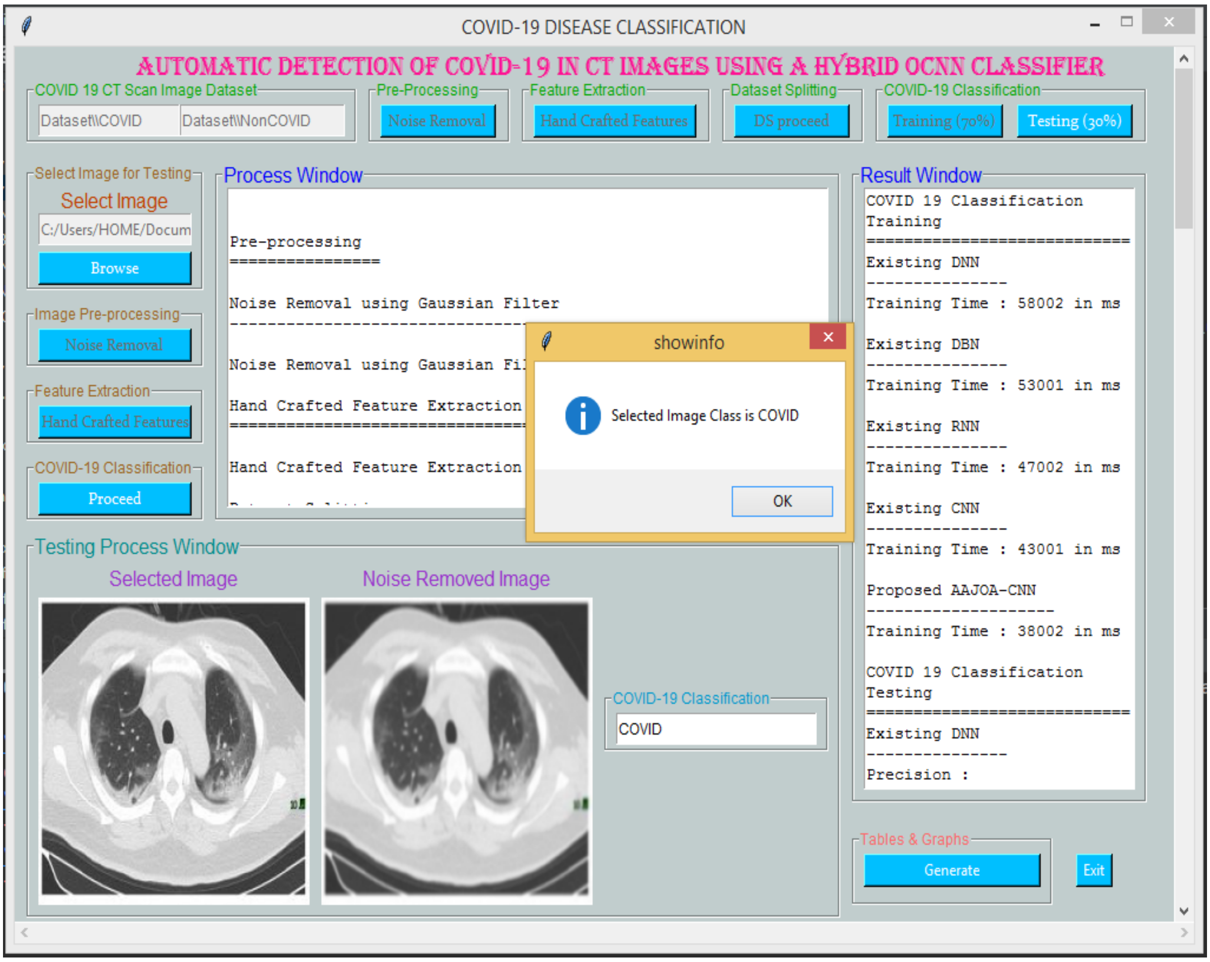


Fig 5: Predicted output

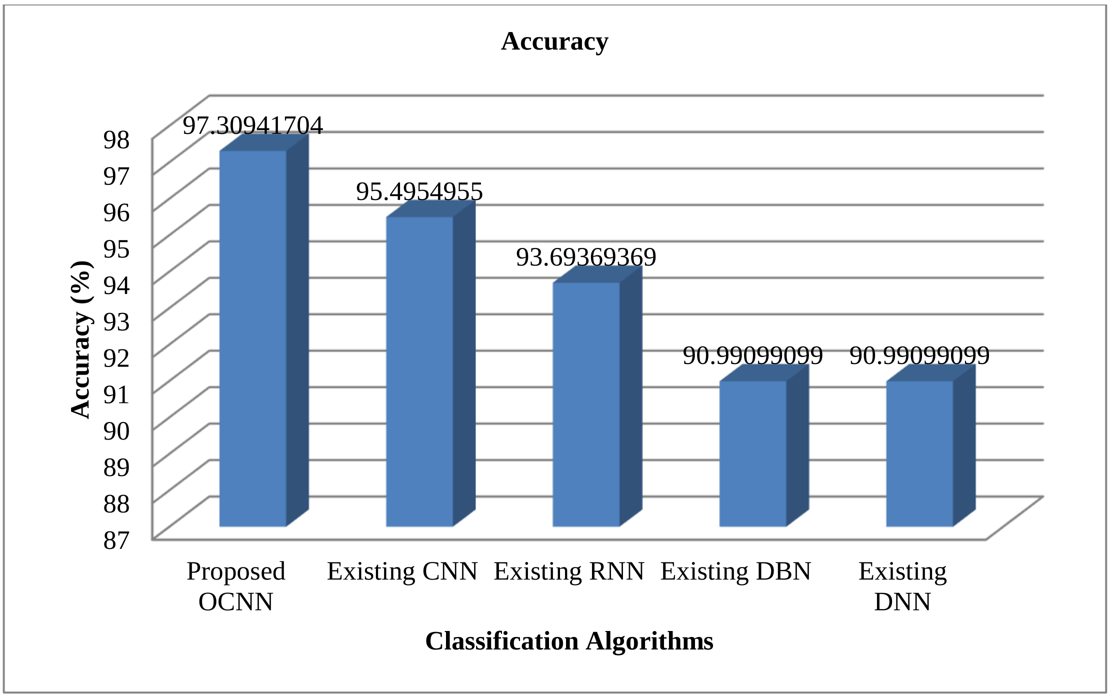


Fig 6: Accuracy

The accuracy measure of comparison and validation analysis of the proposed method is illustrated in figure 6. The proposed method achieves the 97.9% of accuracy in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. The CNN achieves the 95% of accuracy in COVID-19 prediction. The RNN achieves the 93% of accuracy in COVID-19 prediction. The DBN achieves 90% of accuracy in COVID-19 prediction. The DNN achieves 90.4% of accuracy in COVID-19 prediction. Related to the analysis, the proposed approach attained best outcome measure of accuracy when compared with existing techniques.

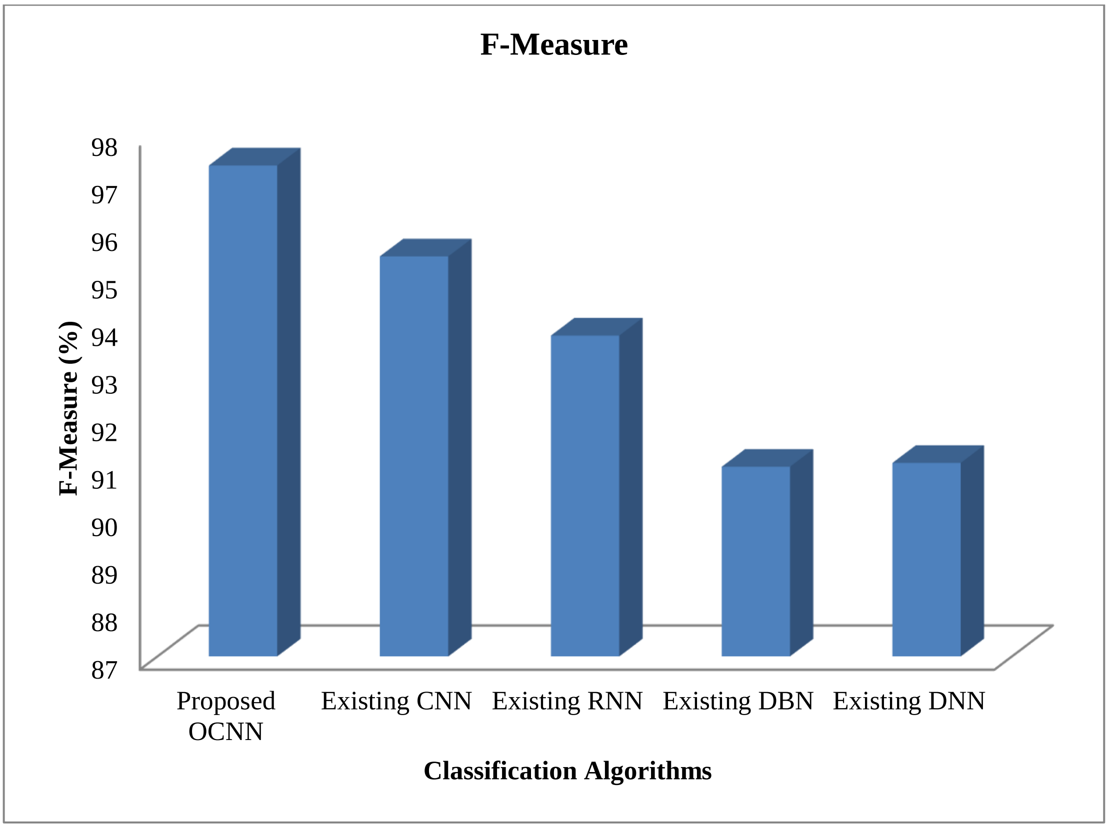


Fig 7: F_Measure

The F-measure of comparison and validation analysis of the proposed method is illustrated in figure 7. The proposed method achieves the 98% of F_Measure in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. The CNN achieves the 95% of F-measure in COVID-19 prediction. The RNN achieves the 93.2% of F-measure in COVID-19 prediction. The DBN achieves 91.45% of the F-measure in COVID-19 prediction. The DNN achieves 91.8% of F_Measure in COVID-19 prediction. Related to the analysis, the proposed approach attained the best outcome measure of F_Measure when compared with existing techniques. The precision measure of comparison and validation analysis of the proposed method is illustrated in figure 8. The proposed method achieves the 97.91% of precision in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. The CNN achieves the 94.91% of precision in COVID-19 prediction. The RNN achieves the 93.2% of precision in COVID-19 prediction. The DBN achieves 91.2% of precision in COVID-19 prediction. The DNN achieves 91.8% of precision in COVID-19 prediction. Related to the analysis, the proposed approach attained the best outcome measure of precision when compared with existing techniques.

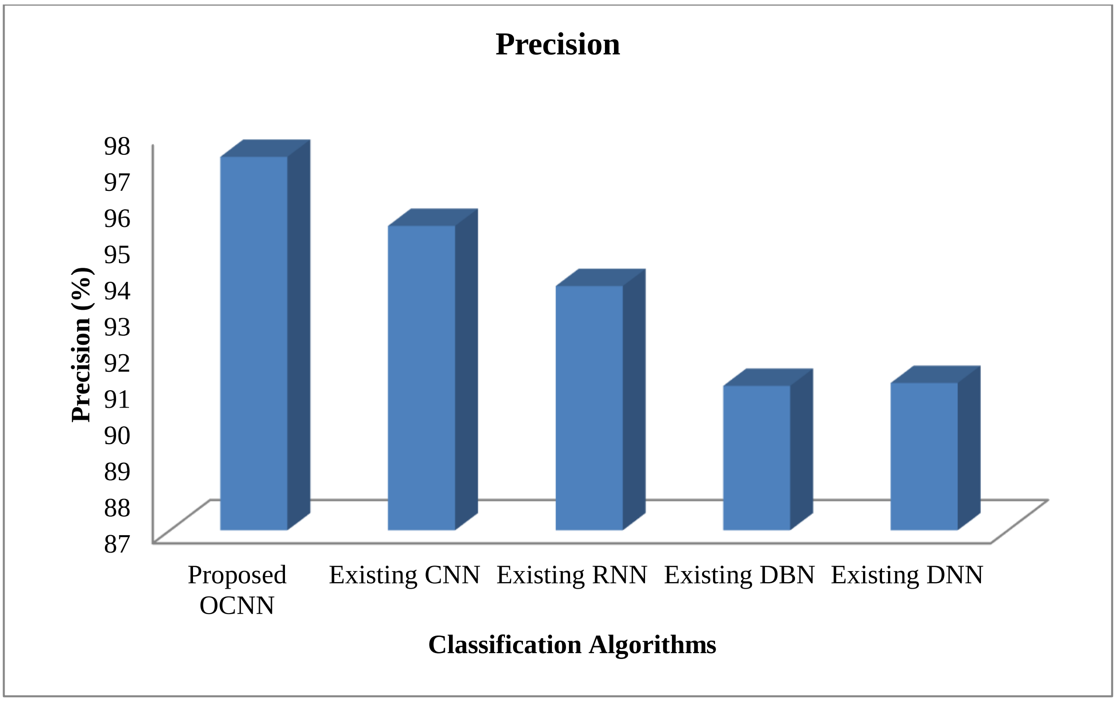


Fig 8: Precision

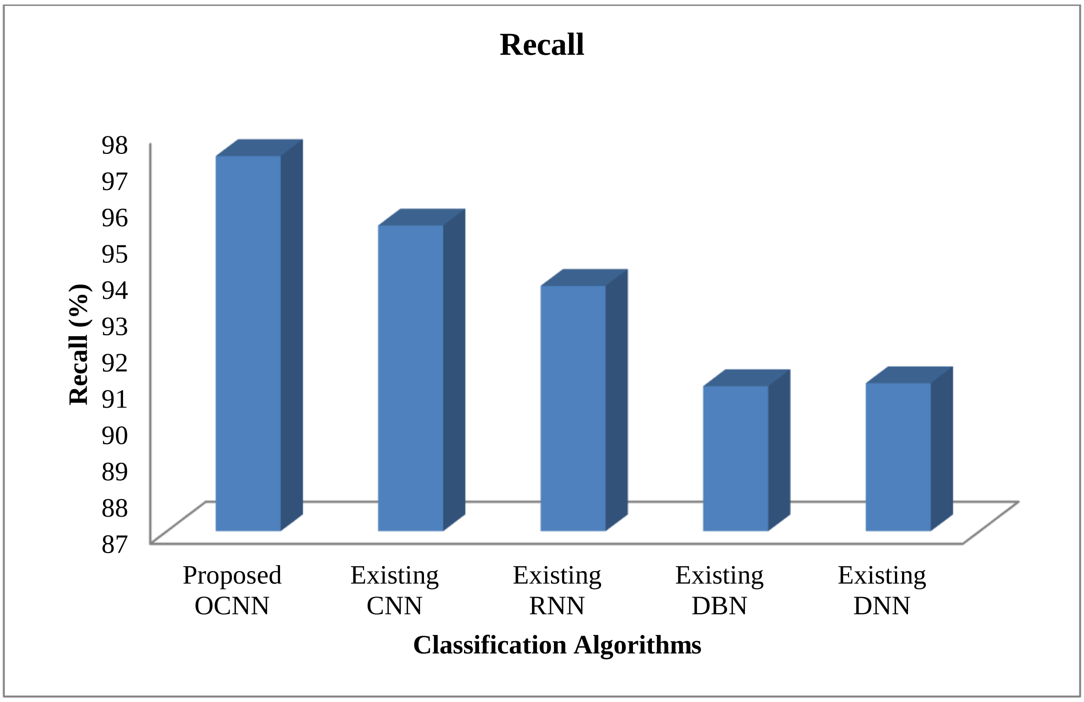


Fig 9: Recall

The sensitivity measure of comparison and validation analysis of the proposed method is illustrated in figure 10. The proposed method achieves the 95.412 % of sensitivity in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. The CNN achieves the 95.412% of sensitivity in COVID-19 prediction. The RNN achieves the 93.75% of sensitivity in COVID-19 prediction. The DBN achieves 90.99% of sensitivity in COVID-19 prediction. The DNN achieves 91.07% of sensitivity in COVID-19 prediction. Related to the analysis, the proposed approach attained the best outcome measure of sensitivity when compared with existing techniques. The recall measure of comparison and validation analysis of the proposed method is illustrated in figure 9. The proposed method achieves the 97.32143% of recall in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. CNN achieves the 95.412% of recall in COVID-19 prediction. The RNN achieves the 93.75% of recall in COVID-19 prediction. The DBN achieves 90.99% of recall in COVID-19 prediction. The DNN achieves 91.07% of recall in COVID-19 prediction. Related to the analysis, the proposed approach attained the best outcome measure of recall when compared with existing techniques.

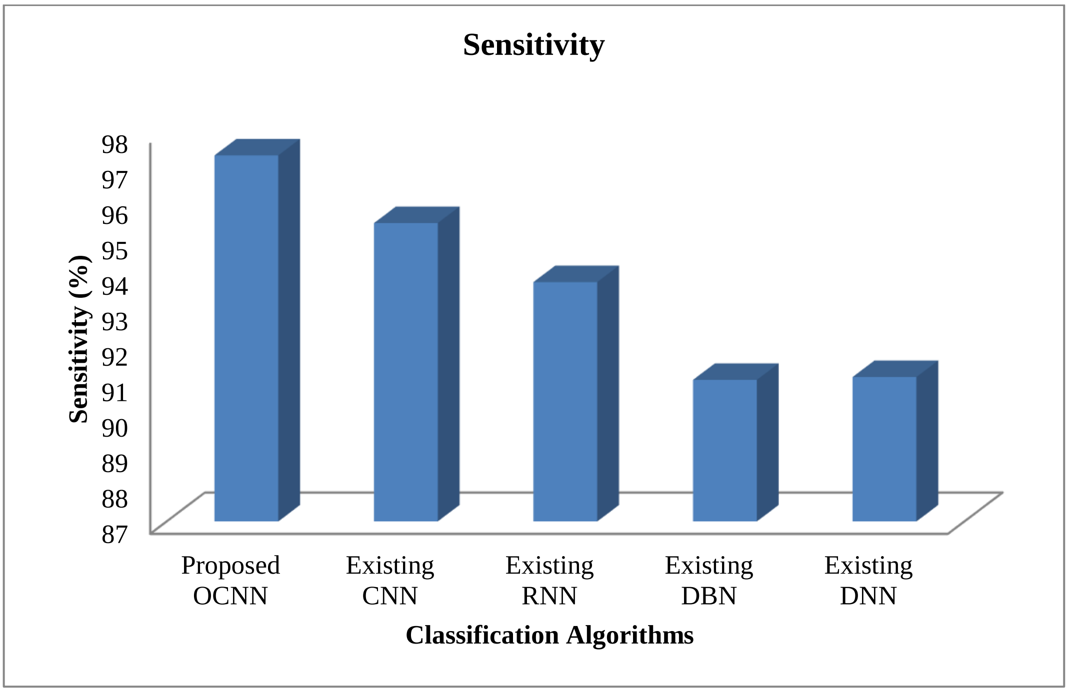


Fig 10: Sensitivity

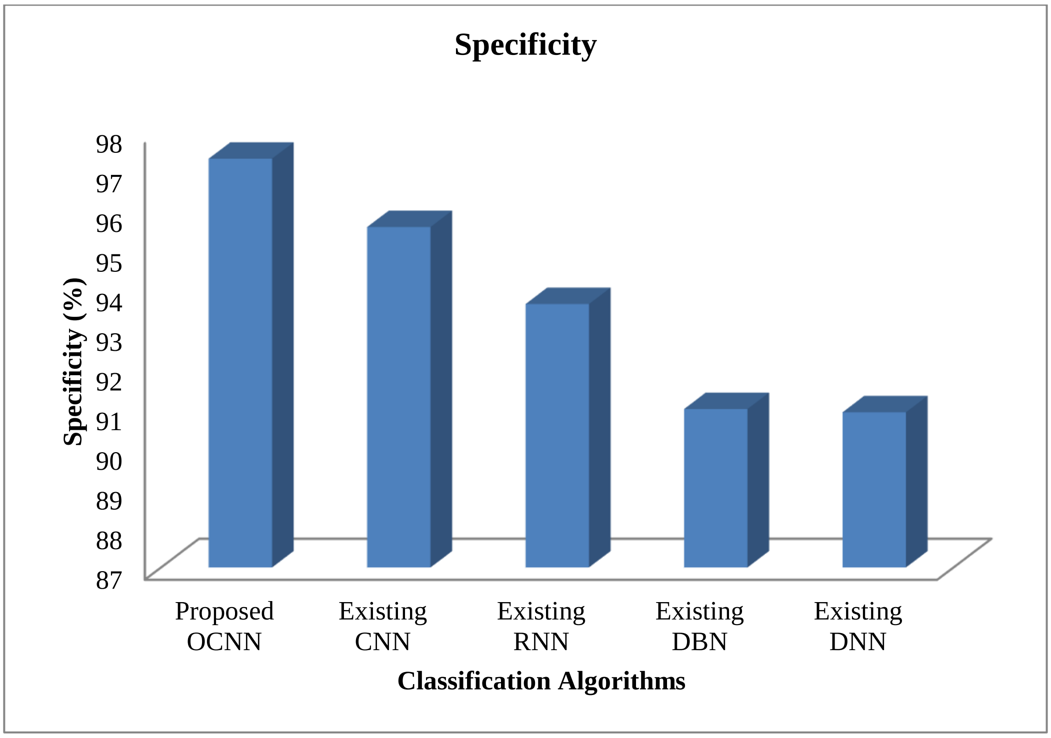


Fig 11: Specificity

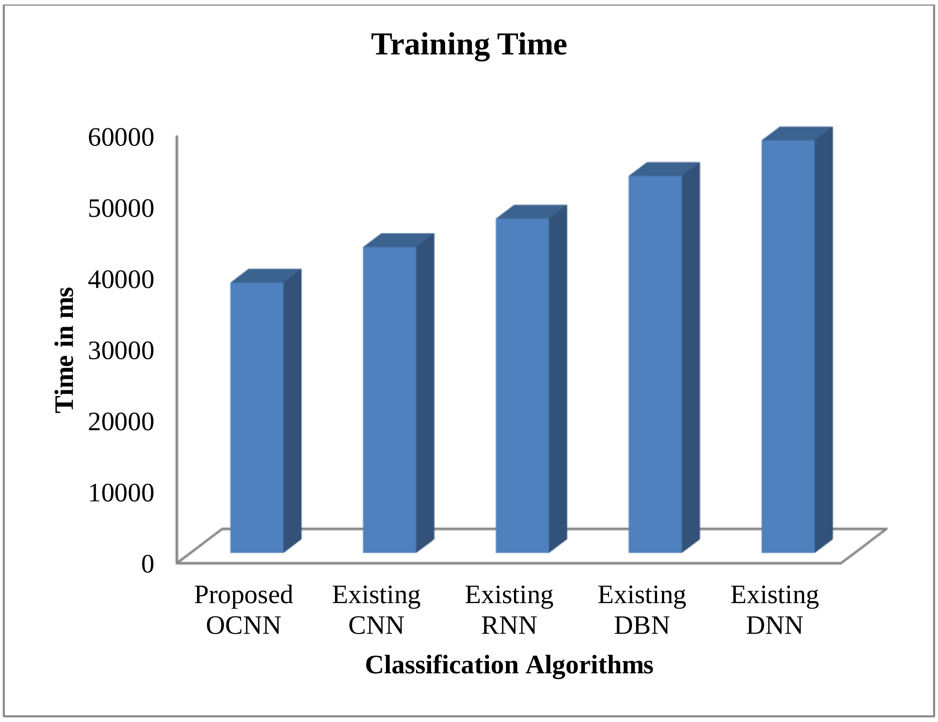


Fig 12: Training time

Table 2: Comparison Table

S. No	References	Method	Accuracy
1	Dheyaaahmedibrahim et al., [11]	a hybrid deep-learning model	95.14
2	Muhammad Umer et al., [12]	CNN model	96.12
3	Mohit Kumar et al., [13]	hybrid convolutional neural network (HFCNN)	94.15
4	Aram Ter-Sarkisov et al., [14]	machine learning	92.15
5	HamzehAsgharnezhad et al., [15]	deep neural networks	95.12
6	Existing	hop field neural network (HFNN)	96.88
7	Proposed	OptimizedCNN	97.30

The specificity measure of comparison and validation analysis of the proposed method is illustrated in figure 11. The proposed method achieves the 97.2972% of specificity in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. The CNN achieves the 95.575 % of specificity in COVID-19 prediction. The RNN achieves the 93.63 % of specificity in COVID-19 prediction. The DBN achieves 90.99 % of specificity in COVID-19 prediction. The DNN achieves 90.909% of specificity in COVID-19 prediction. Related to the analysis, the proposed approach attained the best outcome measure of sensitivity when compared with existing techniques. The training time measure of comparison and validation analysis of the proposed method is illustrated in figure 12. The proposed method achieves the 97.2972% of training time in COVID-19 prediction. It is compared with traditional approaches such as CNN, RNN, DBN, and DNN. The CNN achieves the 95.575 % of training time in COVID-19 prediction. The RNN achieves the 93.63 % of training time in COVID-19 prediction. The DBN achieves 90.99 % of training time in COVID-19 prediction. The DNN achieves 90.909% of training time in COVID-19 prediction. Related to the analysis, the proposed approach attained the best outcome measure of sensitivity when compared with existing techniques.

5. Conclusion

This paper has developed a Covid-19 disease classification using an optimized deep neural network with handcrafted features. The proposed approach consists of three stages namely, pre-processing, handcrafted feature extraction, and classification. Initially, the images have been given to the pre-processing stage to remove the noise present in the input images. Then, extract the hand-crafted features (GLCM) from each image. After that, the pre-processed image is given to the input of the optimized deep neural network classifier to classify an image as normal or abnormal. The proposed optimized deep neural network has been a combination of a CNN and A_2JO algorithm. To enhance the performance of classifier accuracy, the extracted handcrafted features are fused with the optimized deep neural network. The proposed method has achieved 97.30% of accuracy during COVID-19 prediction. In the future, real-time applications are considered for the evaluation of COVID-19 prediction.

References

[1] Mandal, Manotosh, Soovoojeet Jana, Swapan Kumar Nandi, AnupamKhatua, Sayani Adak, and T. K. Kar. "A model based study on the dynamics of COVID-19: Prediction and control." Chaos, Solitons & Fractals 136 (2020): 109889.

[2] Achterberg, Massimo A., Bastian Prasse, Long Ma, StojanTrajanovski, Maksim Kitsak, and Piet Van Mieghem. "Comparing the accuracy of several network-based COVID-19 prediction algorithms." International journal of forecasting (2020).

[3] Santosh, K. C., and Amit Joshi, eds. COVID-19: prediction, decision-making, and its impacts. Berlin: Springer, 2021.

[4] Magesh, S., V. R. Niveditha, P. S. Rajakumar, and L. Natrayan. "Pervasive computing in the context of COVID-19 prediction with AI-based algorithms." International Journal of Pervasive Computing and Communications 16, no. 5 (2020): 477-487.

[5] Song, Jialu, HujinXie, BingbingGao, YongminZhong, ChengfanGu, and Kup-Sze Choi. "Maximum likelihood-based extended Kalman filter for COVID-19 prediction." Chaos, Solitons & Fractals 146 (2021): 110922.

[6] Sinha, Adwitiya, and MeghaRathi. "COVID-19 prediction using AI analytics for South Korea." Applied Intelligence (2021): 1-19.

[7] Mahalle, Parikshit N., Nilesh P. Sable, Namita P. Mahalle, and Gitanjali R. Shinde. "Data analytics: Covid-19 prediction using multimodal data." Intelligent systems and methods to combat Covid-19 (2020): 1-10.

[8] Ghany, Kareem Kamal A., Hossam M. Zawbaa, and Heba M. Sabri. "COVID-19 prediction using LSTM algorithm: GCC case study." Informatics in Medicine Unlocked 23 (2021): 100566.

[9] Villavicencio, CharlynNayve, Julio JerisonEscuderoMacrohon, Xavier Alphonse Inbaraj, Jyh-HorngJeng, and Jer-Guang Hsieh. "Covid-19 prediction applying supervised machine learning algorithms with comparative analysis using weka." Algorithms 14, no. 7 (2021): 201.

[10] Singh, Vijander, Ramesh Chandra Poonia, Sandeep Kumar, Pranav Dass, Pankaj Agarwal, Vaibhav Bhatnagar, and Linesh Raja. "Prediction of COVID-19 corona virus pandemic based on time series data using Support Vector Machine." Journal of Discrete Mathematical Sciences and Cryptography 23, no. 8 (2020): 1583-1597.

[11] Ibrahim, Dheyaa Ahmed, DilovanAsaadZebari, Hussam J. Mohammed, and Mazin Abed Mohammed. "Effective hybrid deep learning model for COVID-19 patterns identification using CT images." Expert Systems 39, no. 10 (2022): e13010.

[12] Umer, Muhammad, Imran Ashraf, SaleemUllah, ArifMehmood, and Gyu Sang Choi. "COVINet: a convolutional neural network approach for predicting COVID-19 from chest X-ray images." Journal of Ambient Intelligence and Humanized Computing (2022): 1-13.

[13] Kumar, Mohit, DhairyataShakya, Vinod Kurup, and WanichSuksatan. "COVID-19 prediction through X-ray images using transfer learning-based hybrid deep learning approach." Materials Today: Proceedings 51 (2022): 2520-2524.

[14] Ter-Sarkisov, Aram. "Covid-ct-mask-net: Prediction of covid-19 from ct scans using regional features." Applied Intelligence (2022): 1-12.
[15] Asgharnezhad, Hamzeh, AfsharShamsi, RoohallahAlizadehsani, Abbas Khosravi, SaeidNahavandi, Zahra Alizadeh Sani, Dipti Srinivasan, and Sheikh Mohammed Shariful Islam. "Objective evaluation of deep uncertainty predictions for covid-19 detection." Scientific Reports 12, no. 1 (2022): 1-11.
[16] Goyal, Bhawna, Anupma Gupta, Ayush Dogra, and DeepikaKoundal. "An adaptive bitonic filtering based edge fusion algorithm for Gaussian denoising." International Journal of Cognitive Computing in Engineering 3 (2022): 90-97.
[17] Jerith, G. Gifta, and P. Nirmal Kumar. "Recognition of Glaucoma by means of gray wolf optimized neural network." Multimedia Tools and Applications 79, no. 15-16 (2020): 10341-10361.
[18] Rahimilarki, Reihane, ZhiweiGao, NanlinJin, and Aihua Zhang. "Convolutional neural network fault classification based on time-series analysis for benchmark wind turbine machine." Renewable Energy 185 (2022): 916-931.
[19] Sarıgül, Mehmet, BuseMelisOzyildirim, and MutluAvci. "Differential convolutional neural network." Neural Networks 116 (2019): 279-287.
[20] Siddiqui, Nimra Idris, AfrozAlam, LayebaQuayyoom, AdilSarwar, Mohd Tariq, Hani Vahedi, Shafiq Ahmad, and Adamali Shah Noor Mohamed. "Artificial jellyfish search algorithm-based selective harmonic elimination in a cascaded H-bridge multilevel inverter." Electronics 10, no. 19 (2021): 2402.
[21] Haklı, Hüseyin, and Harun Uğuz. "A novel particle swarm optimization algorithm with Levy flight." Applied Soft Computing 23 (2014): 333-345.
[22] Bochinski, Erik, Tobias Senst, and Thomas Sikora. "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms." In 2017 IEEE international conference on image processing (ICIP), pp. 3924-3928. IEEE, 2017.