Pulmonary Diffuse Airspace Opacities Diagnosis from Chest X-Ray Images Using Deep Convolutional Neural Networks Fine-Tuned by Whale Optimizer

Xusheng Wang¹ · Cunqi Gong² · Mohammad Khishe³ · Mokhtar Mohammadi⁴ ·
Tarik A. Rashid⁵

Accepted: 14 November 2021 / Published online: 1 December 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The early diagnosis and the accurate separation of COVID-19 from non-COVID-19 cases based on pulmonary diffuse airspace opacities is one of the challenges facing researchers. Recently, researchers try to exploit the Deep Learning (DL) method's capability to assist clinicians and radiologists in diagnosing positive COVID-19 cases from chest X-ray images. In this approach, DL models, especially Deep Convolutional Neural Networks (DCNN), propose real-time, automated effective models to detect COVID-19 cases. However, conventional DCNNs usually use Gradient Descent-based approaches for training fully connected layers. Although GD-based Training (GBT) methods are easy to implement and fast in the process, they demand numerous manual parameter tuning to make them optimal. Besides, the GBT's procedure is inherently sequential, thereby parallelizing them with Graphics Processing Units is very difficult. Therefore, for the sake of having a real-time COVID-19 detector with parallel implementation capability, this paper proposes the use of the Whale Optimization Algorithm for training fully connected layers. The designed detector is then benchmarked on a verified dataset called COVID-Xray-5k, and the results are verified by a comparative study with classic DCNN, DUICM, and Matched Subspace classifier with Adaptive Dictionaries. The results show that the proposed model with an average accuracy of 99.06% provides 1.87% better performance than the best comparison model. The paper also considers the concept of Class Activation Map to detect the regions potentially infected by the virus. This was found to correlate with clinical results, as confirmed by experts. Although results are auspicious, further investigation is needed on a larger dataset of COVID-19 images to have a more comprehensive evaluation of accuracy rates.

Keywords COVID-19 · Whale optimization algorithm · Deep convolutional neural networks · Chest X-rays

✉ Cunqi Gong
gongcq98@163.com

Extended author information available on the last page of the article

1 Introduction

Due to the lack of access to treatment and vaccines for coronavirus, its early detection has become a challenge for scientists [1–3]. The polymerase Chain Reaction (PCR) test has been introduced as one of the primary methods for detecting COVID-19 [4–6]. However, PCR test is a laborious [7], very time-consuming, and complicated process with kits currently in short supply [8, 9]. On the other hand, X-ray images are extensively accessible [10, 11], and scans are comparatively low cost [12–14].

The necessity of designing an accurate [15, 16], real-time detector has become more prominent [17–20]. Considering DL's outstanding capability in these cases [21–24], we propose to employ DCNN as a COVID-19 detector. A few research studies have been carried out since the beginning of the year 2020 that attempt to develop methods to identify patients affected by the epidemic by DCNN [25–27]. Although DL's salient features enable it to solve various learning tasks [28–31], training is difficult [32–34]. Some examples of successful methods for training DL are GD [35, 36], Conjugate Gradient (CG) [37, 38], Arbitrary Lagrangian–Eulerian [39, 40], Hessian-Free Optimization (HFO) algorithm [41, 42], and Krylov Subspace Descent (KSD) [42].

Although GD-based Training (GBT) methods are easy to implement and fast in the process [43, 44], they demand numerous manual parameter tuning to make them optimal [45, 46]. Besides, the GBT's procedure is inherently sequential, thereby parallelizing them with Graphics Processing Units (GPU) is very difficult [47, 48]. On the other hand, although GBT is stable for training, it is almost slow [49]; thereby, it needs multiple CPUs and a vast number of RAMs resource [50, 51].

The Deep auto-encoders have used HFO to train the weights [41], which is more efficient in pre-training and fine-tuning deep auto-encoders than the model proposed by Hinton and Salakhutdinov [52]. On the other side, KSD is simpler and more robust than HFO; besides, it is proven that KSD presents better classification performance and optimization speed than HFO. However, KSD requires more memory than HFO [53].

Heuristic or metaheuristic methods have recently been considered to optimize various parameters and problems [54–56]. Nevertheless, the research study on metaheuristics to optimize DL approaches is rarely carried out [57, 58]. The fusion of the Genetic Algorithm (GA) and DCNN, proposed in [59], was the first study that initiates this optimization model using metaheuristic algorithms. Their approach chooses the DCNN characteristic by the process of recombination and mutation on GA, in which the model of DCNN is considered as a chromosome in GA. In addition, in the recombination phase, just the threshold and weight values of C_1 (first convolution layer) and C_3 (third convolution layer) are modified in the DCNN model. Tuning a DCNN using GA and bezier curve for robot path optimization in dynamic environments is another research work, which suffers from high computational complexity [60].

Fine-tuning DCNN parameters using harmony search (HS) and some of its modifications in handwritten digit and fingerprint recognition was proposed by Rosa et al. [61].

Reference [62] proposed a Progressive Unsupervised Learning (PUL) approach to transfer pre-trained deep DCNN. This method is easy to implement and can be viewed as an effective baseline for unsupervised feature learning. Since the clustering results can be very noisy, this method adds a selection operation between the clustering and fine-tuning phases.

An automatic DCNN architecture design method using genetic algorithms is proposed in [63] to optimize the image classification problems. The proposed algorithm's main

feature is related to its automatic characteristic: users do not need any knowledge about DCNN's structure. However, this method's main drawback is that the GA's chromosomes become too large in large DCNNs, which slows down the algorithm.

The Social-spider optimization algorithm has optimized a deep ANFIS network to predict the biochar yield; however, this combination confronts the ill-conditioning problem [64].

It is worth mentioning that there are other algorithms and techniques tried to improve the performance of deep networks in different fields of study, including the multi-objective transportation [65], Improved feature selection technique [66], a firefly-based algorithm with Levy distribution [67], discrete transforms with selective coefficients [68], signcryption technique [69], equivalent transfer function [70], the enhanced whale optimization algorithm [71]. However, all mentioned methods suffer from high computational complexity and unreliability dealing with image processing problems with high-dimension search space.

Considering the aforementioned limitations and shortcomings, our proposed approach includes training a DCNN network from a draft on the *COVIDetectionNet* dataset [72] to learn to classify infected and normal X-ray images. Subsequently, the last fully-connected layer of the pre-trained DCNN will be replaced by a new fully-connected layer, which is fine-tuned with WOA [73]. In this regard, a particular network representation will be introduced by searching agents. However, in the residual layers of the pre-trained DCNN, the other weights are preserved, which causes training a linear model with the features produced in the prior layer. The proposed scheme is as follows:

- Exert the *COVIDetectionNet* dataset to the convolutional DCNN and load its pre-trained weights.
- Substitute the final fully-connected layer of the pre-trained DCNN with the WOA-trained fully-connected layer.
- Preserve the remaining layers' weights.
- Finally, retrain the whole model using WOA

For the rest of this research paper, the organization is as follows. In Sect. 2, we review some background materials. Section 3 introduces the proposed scheme. Section 4 presents the simulation and discussion results, and finally, conclusions are presented in Sect. 5.

2 Background and Materials

The background knowledge, including the WOA algorithm, the DCCN, and the COVID-X-ray dataset, will be represented in this section.

2.1 Whale Optimization Algorithm

Generally, WOA is a novel Swarm Intelligence (SI) optimization algorithm that mathematically formulates humpback whales' hunting manner [73]. The distinct difference between WOA and other benchmark optimization algorithms is the updating model to improve the candidate solutions. This mathematical model contains searching and attacking prey called *bubble-net feeding*. The bubble-net feeding behavior is shown in Fig. 1.

Fig. 1 The bubble-net feeding mechanism of hunting.

This figure shows that intelligent hunting is carried out by generating a trap by moving in a spiral “9-shape” way around the prey and generating bubbles along the spiral path. The *encircling* is another hunting mechanism of humpback whales. Firstly, WOA initiates the hunting process by encircling the prey using the *bubble-net feeding* mechanism. The mathematical model of WOA is as follows [73]:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}^*(t) - \mathbf{A}\mathbf{D} & r < 0.5 \\ \mathbf{D}'\mathbf{M}\cos(2\pi t) + \mathbf{X}^*(t) & r > 0.5 \end{cases} \quad (1)$$

$$\mathbf{D} = |\mathbf{C}\mathbf{X}^*(t) - \mathbf{X}(t)| \quad (2)$$

$$\mathbf{D}' = |\mathbf{X}^*(t) - \mathbf{X}(t)| \quad (3)$$

$$\mathbf{A} = 2\mathbf{Q}\mathbf{R} - \mathbf{Q} \quad (4)$$

$$\mathbf{C} = 2\mathbf{R} \quad (5)$$

$$\mathbf{M} = e^{kl} \quad (6)$$

where $\mathbf{X}$ is a vector presenting the positions of whales during iterations and $\mathbf{X}^*$ is a vector indicating locations of the prey, r represents a random number inside $[0,1]$, indicating the distance of the i th searching agent and the prey, k is a constant number determining the bubble-net spiral shape, l is a random number inside $[-1, 1]$, t indicating the current iteration, $\mathbf{Q}$ is a vector which linearly decreases from two to zero during iterations, and $\mathbf{R}$ is a random vector inside $[0, 1]$. While $r < 0.5$, Eq. (1) simulates the encircling behavior; otherwise, the equation intends to model the *bubble-net feeding* mechanism. Therefore, r randomly exchanges the model behavior between these two main phases.

2.2 Convolution Neural Network

Generally, DCNN is a conventional Multi-Layer Perceptron (MLP) based on three concepts: connection weights sharing, local receptive fields, and temporal/spatial sub-sampling. These concepts can be arranged into two classes of layers, including subsampling layers and convolution layers. As shown in Fig. 2, the processing layers include three convolution layers C1, C3, and C5, which are located one between layers S2 and S4, and final output layer F6. These sub-sampling and convolution layers are organized as feature maps.

Neurons in the convolution layer are linked to a local receptive field in the prior layer. Consequently, neurons with identical feature maps (FMs) receive data from various input regions until the input is completely skimmed. However, the same weights are shared.

In the sub-sampling layer, the FMs are spatially down-sampled by a factor of 2. As an illustration, in layer C3, the FM of size 10×10 is sub-sampled to conforming FM of size 5×5 in the next layer, S4. The classification process is the final layer (F6).

Each FMs are the outcome of a convolution from the previous layer's maps by their corresponding kernel and a linear filter in this structure. The weights w^k and adding bias b_k generate the k_{th} (FM) FM_{ij}^k using the tanh function as Eq. (7).

$$FM_{ij}^k = \tanh((W^k \times x)_{ij} + b_k) \quad (7)$$

By reducing the resolution of FMs, the sub-sampling layer lead to spatial invariance, in which each pooled FM refers to one FM of the prior layer. The sub-sampling function is defined as Eq. (8).

$$\alpha_j = \tanh(\beta \sum_{N \times N} \alpha_i^{n \times n} + b) \quad (8)$$

where $\alpha_i^{n \times n}$ are the inputs, β and b are trainable scalar and bias, respectively. After a diverse convolution and sub-sampling layer, the last layer is a fully connected structure that carries out the classification task. There is one neuron for each output class. Thereby, in the case of the Covid-19 dataset, this layer contains two neurons for their classes.

Fig. 2 The architecture of LeNet-5 DCNN

Table 1 The categories of images per class in the COVID dataset

Category	COVID-19	Normal
Training set	84 (420 after augmentation)	2000
Test set	100	3000

Fig. 3 Six stochastic sample images from the *COVID-X-ray-5k* dataset

2.3 COVID-X-Ray Dataset

In this paper, a dataset named *COVID X-ray-5k* dataset, including 2084 training and 3100 test images, was utilized [74]. In this dataset, considering radiologist advice, only anterior–posterior Covid-19 X-ray images are utilized because the lateral images are not applicable for detection purposes. Expert radiologists evaluated those images, and those that did not have clear signs of COVID-19 were removed. In this way, 19 images out of the 203 images were removed, and 184 images remained, indicating clear signs of COVID-19. With this method, the community a more clearly labeled dataset was introduced. Of these 184 photos, 100 images are considered for the test set, and 84 images are intended for the training set. For the sake of increasing the number of COVID-19 samples to 420, data augmentation is applied. Since the number of non-COVID images was minimal in the *covid-chestxray-dataset* [75], the supplementary *ChexPert* dataset [76] was employed. This dataset includes 224,316 chest X-ray images of 65,240 patients. In this dataset, 2000 and 3000 non-COVID images were chosen for the training set and test set, respectively. The final number of images related to various classes is reported in Table 1. Figure 3 indicates six stochastic sample images from the *COVID-X-ray-5k* dataset, including two COVID-19 and four standard samples.

3 Methodology

3.1 Presentation of Searching Agent

Generally, there are two main issues in tuning a deep network using a meta-heuristic optimization algorithm. First, the structure's parameters must be represented by the meta-heuristic algorithm's searching agents (candid solution). Next, the fitness function must be defined based on the interest of the considered problem [77].

The presentation of network parameters is a distinct phase in tuning a DCNN using the WOA algorithm. Thereby, important parameters of the DCNN, i.e., weights and biases of the fully connected layer, should be determined to provide the best detection accuracy [78].

To sum up, the WOA algorithm optimizes the last layer's values of weights and bias used to calculate the loss function as a fitness function. The weight and bias values in the last layer are used as searching agents in the WOA algorithm.

Generally speaking, three schemes are used to present weights and biases of a DCNN as candid solutions of the meta-heuristic algorithm: vector-based, matrix-based, and binary state. Considering the fact that the WOA needs the parameters in a vector-based model, in this paper, the candid solution is shown as Eq. 9 and Fig. 4.

$$\text{Candid solutions} = [W_{11}, W_{12}, \dots, W_{nh}, b_1, \dots, b_h, M_{11}, \dots, M_{hm}] \quad (9)$$

where n is the number of the input nodes, W_{ij} indicates the connection weight between the i_{th} input node and the j_{th} hidden neuron, b_j is the j_{th} hidden neuron's bias, and M_{jo} shows the connection weight from the j_{th} hidden neuron to the o_{th} output neuron. As previously stated, the proposed architecture is a simple LeNet-5 structure. In this section, two

Fig. 4 Assigning the DCNN's parameters as the candid solution (searching agents) of WOA

structures, namely i-6c-2 s-12c-2 s and i-8c-2 s-16c-2 s, are used where C and S are convolution and sub-sampling layers, respectively [79, 80]. The kernel size of all convolution layers is 5×5 , and the scale of sub-sampling is down-sampled by a factor of 2.

3.2 Loss Function

The WOA algorithm trains DCNN (DCNN-WOA) to obtain the best accuracy and minimize evaluated classification error and network complexity in the proposed meta-heuristic method. This objective can be computed by the loss function of the metaheuristic searching agent or the Mean Square Error (MSE) classification procedure. However, the loss function used in this method is as follows:

$$y = \frac{1}{2} \sqrt{\frac{\sum_{i=0}^N (o - u)^2}{N}} \quad (10)$$

where o shows the supposed output, u indicates the desired output, and N shows the number of training samples. Two termination criteria: reaching maximum iteration or predefined loss function, are utilized by the proposed WOA algorithm. Consequently, the pseudo-code of DCNN-WOA is shown in Fig. 5.

4 Simulation Results and Discussion

As stated before, the principal goal is to improve the classification accuracy of classic DCNN by using the WOA algorithm. According to references [25, 73] and also in order to have a fair comparison, the population is set to 10, and the maximum iteration is set to 10. The parameter of DCNN, i.e., the learning rate α and the batch size, are equal to 1 and 100, respectively. Also, the number of epochs is considered between 1 and 10 for every evaluation. The evaluation was carried out in MATLAB-R2019a, on a PC with processor Intel Core i7-4500u, 16 GB RAM running memory, in Windows 10, with five separate runtimes.

As proven in the literature [81, 82], the accuracy rate does not represent enough information about the detector's performance. Thereby, Receiver Operating Characteristic (ROC) curves were used to exploit the classifier on all the samples in the test datasets, devoting an evaluated probability of images P_T for each sample. Next, a threshold value $T \in [0, 1]$ was introduced, and for each threshold value, the detection rate was calculated. Thereby, the computed values were plotted as a ROC curve. Generally speaking, the area under the ROC curve (AUC) shows the probability of correct detection.

Figure 6 indicates the calculated ROC curves for the detection of COVID-19 samples using DCNN-WOA and conventional DCNN. This comparison was carried out because the test dataset, the initial conditions, and the primary convolutional network (LeNet-5 DCNN) are entirely identical. Therefore, the effectiveness of the WOA algorithm on classic LeNet-5 DCNN can be fairly compared. The ROC curves show that DCNN-WOA significantly outperforms LeNet-5 DCNN on the test dataset.

The training was carried out ten times so that the training times varied between 5 and 10 min, and the designed DCNN-WOA had the detection accuracy of between 98.11% and 99.38% on the COVID-19 validation set.

Because of the extensive range of different results, the ten trained DCNN-WOA are ensemble by weighted averaging, using the validation accuracy as the weights. The

Fig. 5 The Pseudo-code for DCNN-WOA model**Pseudo-Code DCNN-WOA****Result:** *accuracy, time*

Initialization of the DCNN structure

Calculation process: *weights (w), biases (b), loss function $f(x)$;***Initialize** the whales population X_i ($i = 1, 2, \dots, n$)

Calculate the loss function of each candid solution

 X^* = the best candid solution (The combination of weights (w), biases (b))**While** ($t < \text{max. iterations}$) **for** each candid solution Update a , A , C , l , and r **if**₁ ($r < 0.5$) **if**₂ ($|A| < 1$)

Update the position of the current candid solution by the Eq. (1a)

else if₂ ($|A| \geq 1$)

Choose a random candid solution

Update the position of the current candid solution by the Eq. (6)

end if₂ **elseif**₁ ($r \geq 0.5$)

Update the position of the current candid solution by the Eq. (1b)

end **end if**₁ **for**

Check the candid solutions move beyond the boundary and modify them

Calculate the loss function of each candid solution

 Update X^* **if** there is a better solution $t = t + 1$ **end while** **return** X^*

combined ensemble DCNN-WOA achieves a validation accuracy of 98.74%, while LeNet-5 DCNN acquires the detection accuracy of between 84.58% and 93.31%, and the resulting ensemble obtained the detection accuracy of 88.94% on the COVID-19 validation dataset.

To further evaluate the performance of DCNN-WOA in the detection of COVID-19 samples from uninfected ones, newly proposed benchmark models include conventional DCNN [83], DUICM [84], and matched subspace classifier with adaptive dictionaries [85] are utilized. Figures 7 and 8 show the outcome ROC and precision-recall curve for $i-6c-2s-12c-2s$ and $i-8c-2s-16c-2s$ structures, respectively.

As can be seen from these figures, the DCNN-WOA detector indicates outstanding COVID-19 detection results compared with other benchmark models. For the sake of comparison, the proposed DCNN-WOA provides over 98.25% correct COVID-19

Fig. 6 ROC curves for DCNN-WOA and classic DCNN

Fig. 7 ROC and Precision-recall curves for *i-6c-2s-12c-2s* models

sample detection for less than a 1.75% false alarm detection rate, which shows the WOA algorithm's capability to increase the performance of the DCNN model.

Generally, the precision-recall plot shows the tradeoff between recall and precision for various threshold levels [86]. A high area under the precision-recall curve represents both high precision and recall, where high precision indicates a low false-positive rate, and high recall indicates a low false-negative rate [86, 87]. As can be observed from the curves in Figs. 7 and 8, DCNN-WOA has a higher area under the precision-recall curves. Therefore, it indicates a lower false positive and false negative rate than other benchmark detectors.

Fig. 8 ROC and Precision-recall curves for i-8c-2s-16c-2s models

The obtain accuracy and computational time results for the i-6c-2s-12c-2s structure are summarized in Tables 2 and 3 in order. Also, the obtain accuracy and computational time results for the i-8c-2s-16c-2s network are summarized in Tables 4 and 5, in order. The original DCNN is evaluated only once for each epoch because the accuracy is not changed if the evaluation is repeated with the same experimental condition. In general, the tests conducted showed that the higher the epoch value, the better is the accuracy. For example, in the first epoch, compared to DCNN (84.11), the accuracy increased to 4.01 for MSAD (88.12), 5.18 to DCNN-WOA (89.29), and 2.04 for DUICM (86.15). While in the fifth epoch, compared to DCNN (93.47), the increase of accuracy is 1.94 for MSAD (95.41), 4.09 for DCNN-WOA (97.56), and 0.50 for DUICM (93.97). In the case of 10 epochs, as reported in Table 2, the increase in accuracy compared to DCNN (97.18) is only 0.04 for MSAD (97.22), 2.67 for DCNN-WOA (99.85), and 0.28 for DUICM (97.46).

The simulation results indicate that DCNN-WOA represents the best accuracy for all epochs. Accuracy improvement of DCNN-WOA, compared to the original DCNN, varies for each epoch, with a range of values between 2.67 (the tenth epoch) up to 5.88 (the first epoch). Considering the stochastic nature of WOA and stochastically choosing

Table 2 Accuracy and STD for i-2s-6c-2s-12c structure

Epoch	DCNNWOA		MSAD		DUICM		DCNN	
	Accuracy (%)	STD	Accuracy (%)	STD	Accuracy (%)	STD	Accuracy (%)	STD
1	89.99	N/A	88.12	0.41	86.15	0.33	84.11	0.75
2	93.14	N/A	92.11	0.37	90.22	0.21	89.22	0.31
3	96.22	N/A	93.12	0.31	92.01	0.32	91.11	0.38
4	97.45	N/A	94.62	0.24	92.99	0.38	92.47	0.11
5	97.56	N/A	95.41	0.18	93.97	0.14	93.47	0.29
6	98.88	N/A	95.92	0.17	95.11	0.22	94.02	0.39
7	98.95	N/A	96.11	0.16	96.01	0.21	95.17	0.19
8	98.97	N/A	96.77	0.12	97.11	0.13	96.58	0.21
9	99.37	N/A	96.99	0.09	97.44	0.14	96.76	0.09
10	99.85	N/A	97.22	0.15	97.46	0.11	97.18	0.11

Table 3 Computation time and STD for i-2s-6c-2s-12c structure

Epoch	DCNNWOA		MSAD		DUICM		DCNN	
	Time (s)	STD	Time (s)	STD	Time (s)	STD	Time (s)	STD
1	99.11	N/A	102.23	1.01	112.08	0.89	102.11	0.77
2	202.14	N/A	236.03	8.54	267.11	1.54	268.47	4.33
3	307.68	N/A	347.41	2.32	399.96	2.25	299.17	0.53
4	362.22	N/A	421.45	2.01	547.27	2.16	443.49	0.65
5	461.11	N/A	582.47	3.96	521.27	1.37	601.77	3.09
6	586.31	N/A	692.75	1.23	636.75	5.96	721.22	2.01
7	657.42	N/A	797.02	1.02	739.67	4.01	836.75	2.25
8	754.11	N/A	854.43	1.74	828.73	4.36	1007.9	1.53
9	849.57	N/A	964.12	2.01	998.89	6.02	1125.57	5.07
10	934.54	N/A	1112.36	1.97	1223.35	1.47	1133.44	4.25

Table 4 Accuracy and STD for i-2s-8c-2s-16c structure

Epoch	DCNNWOA		MSAD		DUICM		DCNN	
	Accuracy (%)	STD	Accuracy (%)	STD	Accuracy (%)	STD	Accuracy (%)	STD
1	89.32	N/A	88.17	0.32	87.25	0.31	86.22	0.65
2	92.56	N/A	91.99	0.33	87.99	0.14	87.12	0.32
3	93.22	N/A	94.12	0.15	90.22	0.22	90.56	0.39
4	95.11	N/A	94.52	0.25	92.25	0.41	92.72	0.11
5	95.75	N/A	95.33	0.21	94.11	0.14	93.59	0.29
6	95.73	N/A	96.01	0.11	95.05	0.22	94.18	0.39
7	96.50	N/A	96.57	0.11	96.32	0.21	95.15	0.19
8	96.86	N/A	96.89	0.12	96.48	0.12	95.78	0.22
9	97.86	N/A	97.01	0.05	96.78	0.16	96.53	0.09
10	98.27	N/A	97.27	0.12	96.92	0.11	96.89	0.05

the connection weight of the fully connected layer, the computation time for the DCNN-WOA, compared to the classic DCNN, is in the range of 0.9706 times (for the first epoch: 99.11/102.11) up to 0.8245 times (for the tenth epoch: 934.54/1133.44). It is evident that as the number of epochs increases, the time efficiency of the WOA is more prominent because the stochastic nature of the WOA algorithm leads to decreasing the complexity of the search space. It should be pointed out that the results of the i-8c-2s-16c-2s structure indicated in Tables 4 and 5 approve the prior conclusion for the i-8c-2s-16c-2s network. Consequently, WOA can improve the performance of DCNN with the i-8c-2s-16c-2s structure as well as the i-6c-2s-12c-2s structure.

From the viewpoint of data science experts, the best result could be indicated in terms of the confusion matrix, overall accuracy, precision, recall, ROC curve, etc. However, these optimal results might not be sufficient for medical specialists and radiologists if the results cannot be interpreted. Identifying the Region of Interest (ROI) leading to the network's decision-making will enhance medical experts' understanding [88].

Table 5 Computation time and STD for i-2s-8c-2s-16c structure

Epoch	DCNNWOA		MSAD		DUICM		DCNN	
	Time (s)	STD	Time (s)	STD	Time (s)	STD	Time (s)	STD
1	95.33	N/A	99.48	1.17	122.28	0.89	171.54	0.78
2	247.14	N/A	187.41	9.78	262.18	1.47	362.15	6.01
3	300.04	N/A	342.19	2.01	399.63	2.22	399.10	0.54
4	352.04	N/A	425.81	2.00	524.35	2.14	532.12	0.62
5	422.33	N/A	523.15	3.98	517.23	1.36	701.91	3.01
6	545.87	N/A	699.41	1.33	680.31	5.86	811.56	2.01
7	625.07	N/A	795.15	1.12	764.19	4.01	968.18	2.02
8	687.01	N/A	900.61	1.67	874.74	4.48	1105.2	1.22
9	799.12	N/A	1002.11	2.34	972.87	5.78	1245.54	5.02
10	932.35	N/A	1112.56	1.97	1223.44	1.47	1473.99	4.74

In this section, the results provided by designed networks for the COVID X-ray-5k dataset were investigated. The Class Activation Mapping (CAM) results were displayed for the COVID X-ray-5k dataset to localize the areas suspicious of the Covid-19 virus. To emphasize the discriminative regions, the probability predicted by the DCNN model for each image class gets mapped back to the last convolutional layer of the corresponding model that is particular to each class. The CAM for a determined image class is the outcome of the activation map of the Rectified Linear Unit (ReLU) layer following the last convolutional layer. It is identified by how much each activation mapping contributes to the final grade of that particular class. The novelty of CAM is the total average pooling layer applied after the last convolutional layer based on the spatial location to produce the connection weights. Thereby, it permits identifying the desired regions within an X-ray image that differentiates the class specificity preceding the Softmax layer, which leads to better predictions.

Demonstration using CAM for DCNN models allows the medical specialists and radiology experts to localize the areas suspicious of the Covid-19 virus indicating in Figs. 9 and 10.

Figures 9 and 10 indicate the results for Covid-19 detection in X-ray images. Figure 9 shows the outcomes for the case marked as ‘Covid-19’ by the radiologist, and the DCNN-WOA model predicts the same and indicates the discriminative area for its decision.

Figure 10 shows the outcomes for a ‘normal’ case in X-ray images, and different regions are emphasized by both comparing models for their prediction of the ‘normal’ subset.

Fig. 9 ROI for positive Covid-19 cases using ACM

Fig. 10 ROI for Normal cases using ACM

Now, medical specialists and radiology experts can choose the network architecture based on these decisions. This kind of CAD visualization would provide a useful second opinion to the medical specialists and radiology experts and also improve their understanding of deep learning models.

5 Conclusion

In this paper, the WOA was proposed to design an accurate DCNN model for positive Covid-19 X-ray detection. The designed detector was benchmarked on the COVID-Xray-5k dataset, and the results were evaluated by a comparative study with classic DCNN, DUICM, and MSAD. The results indicated that the designed detector could present very competitive results compared to these benchmark models. The concept of Class Activation Map (CAM) was also applied to detect the virus's regions potentially infected. It was found to correlate with clinical results, as confirmed by experts. A few research directions can be proposed for future work with the DCNN-WOA, such as underwater sonar target detection and classification. Also, changing WOA to tackle multi-objective optimization problems can be recommended as a potential contribution. The investigation of the chaotic maps' effectiveness to improve the performance of the DCNN-WOA can be another research direction. Although the results were promising, further investigation is needed on a larger dataset of COVID-19 images to have a more comprehensive evaluation of accuracy rates.

Funding Not Applicable.

Availability of Data and Material The resource images can be downloaded using the following link and references [72]. <https://github.com/ieee8023/covid-chestxray-dataset>, 2020.

Code Availability The source code of the models can be available by request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Li, X., Dong, Z.-Q., Yu, P., Wang, L.-P., Niu, X.-D., Yamaguchi, H., et al. (2021). Effect of self-assembly on fluorescence in magnetic multiphase flows and its application on the novel detection for COVID-19. *Physics of Fluids*, 33, 42004.

2. Liu, Y., Lv, X., & Tang, Z. (2021). The impact of mortality salience on quantified self behavior during the COVID-19 pandemic. *Personality and Individual Differences*, 180, 110972.
3. Abdel-Basset, M., Mohamed, R., Elhoseny, M., Chakraborty, R. K., & Ryan, M. (2020). A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy. *IEEE Access*, 8, 79521–79540.
4. Pang, X., Gong, K., Zhang, X., Wu, S., Cui, Y., & Qian, B.-Z. (2019). Osteopontin as a multifaceted driver of bone metastasis and drug resistance. *Pharmacological Research*, 144, 235–244.
5. Elhoseny, M., Shankar, K., & Uthayakumar, J. (2019). Intelligent diagnostic prediction and classification system for chronic kidney disease. *Science and Reports*, 9, 1–14.
6. Thakur, S., Singh, A. K., Ghrera, S. P., & Elhoseny, M. (2019). Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications. *Multimedia Tools and Applications*, 78, 3457–3470.
7. Abdel-Basset, M., El-Hoseny, M., Gamal, A., & Smarandache, F. (2019). A novel model for evaluation Hospital medical care systems based on plithogenic sets. *Artificial Intelligence in Medicine*, 100, 101710.
8. Abdel-Basset, M., Mohamed, M., Elhoseny, M., Chiclana, F., & Zaid, A.E.-N.H. (2019). Cosine similarity measures of bipolar neutrosophic set for diagnosis of bipolar disorder diseases. *Artificial Intelligence in Medicine*, 101, 101735.
9. Libo, Z., Tian, H., Chunyun, G., & Elhoseny, M. (2019). Real-time detection of cole diseases and insect pests in wireless sensor networks. *Journal of Intelligent & Fuzzy Systems*, 37, 3513–3524.
10. Zuo, C., Chen, Q., Tian, L., Waller, L., & Asundi, A. (2015). Transport of intensity phase retrieval and computational imaging for partially coherent fields: The phase space perspective. *Optics and Lasers in Engineering*, 71, 20–32.
11. Zuo, C., Sun, J., Li, J., Zhang, J., Asundi, A., & Chen, Q. (2017). High-resolution transport-of-intensity quantitative phase microscopy with annular illumination. *Science and Reports*, 7, 1–22.
12. Elhoseny, M., Bian, G.-B., Lakshmanaprabu, S. K., Shankar, K., Singh, A. K., & Wu, W. (2019). Effective features to classify ovarian cancer data in internet of medical things. *Computer Networks*, 159, 147–156.
13. Elhoseny, M., & Shankar, K. (2019). Optimal bilateral filter and convolutional neural network based denoising method of medical image measurements. *Measurement*, 143, 125–135.
14. Geetha, K., Anitha, V., Elhoseny, M., Kathiresan, S., Shamsolmoali, P., & Selim, M. M. (2021). An evolutionary lion optimization algorithm-based image compression technique for biomedical applications. *Expert Systems*, 38, e12508.
15. Uthayakumar, J., Elhoseny, M., & Shankar, K. (2020). Highly reliable and low-complexity image compression scheme using neighborhood correlation sequence algorithm in WSN. *IEEE Transactions on Reliability*, 69, 1398–1423.
16. Pan, D., Xia, X.-X., Zhou, H., Jin, S.-Q., Lu, Y.-Y., Liu, H., et al. (2020). COCO enhances the efficiency of photoreceptor precursor differentiation in early human embryonic stem cell-derived retinal organoids. *Stem Cell Research & Therapy*, 11, 1–12.
17. Wu, C., Khishe, M., Mohammadi, M., Karim, S. H. T., Rashid, T. A. (2021). Evolving deep convolutional neural network by hybrid sine–cosine and extreme learning machine for real-time COVID19 diagnosis from X-ray images. *Soft Computing* 1–20.
18. Elhoseny, M. (2020). Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. *Circuits, Systems, and Signal Processing*, 39, 611–630.
19. Shankar, K., Elhoseny, M., Lakshmanaprabu, S.K., Ilayaraja, M., Vidhyavathi, R., Elsoud, M.A., & Alkhambashi, M. (2020). Optimal feature level fusion based ANFIS classifier for brain MRI image classification. *Concurrency and Computation: Practice and Experience*, 32.
20. Niu, Z., Zhang, B., Wang, J., Liu, K., Chen, Z., Yang, K., et al. (2020). The research on 220GHz multi-carrier high-speed communication system. *China Communications*, 17, 131–139.
21. Qin, C., Jin, Y., Tao, J., Xiao, D., Yu, H., Liu, C., et al. (2021). DTCNNMI: A deep twin convolutional neural networks with multi-domain inputs for strongly noisy diesel engine misfire detection. *Measurement*, 180, 109548.
22. Cai, K., Chen, H., Ai, W., Miao, X., Lin, Q., & Feng, Q. (2021). Feedback convolutional network for intelligent data fusion based on near-infrared collaborative IoT technology. *IEEE Transactions on Industrial Informatics*, 18, 1200–1209.
23. Mohanty, S. N., Lydia, E. L., Elhoseny, M., Al Otaibi, M. M. G., & Shankar, K. (2020). Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks. *Physical Communication*, 40, 101097.

24. Elhoseny H., Elhoseny, M., Riad A. M., Hassanien, A. E. (2018). A framework for big data analysis in smart cities. In *The International Conference on Advanced Machine Learning Technologies and Applications*, Springer; (pp. 405–14).
25. Khishe, M., Caraffini, F., & Kuhn, S. (2021). Evolving deep learning convolutional neural networks for early COVID-19 detection in chest X-ray images. *Mathematics*, 9, 1002.
26. Hu, T., Khishe, M., Mohammadi, M., Parvizi, G.-R., Karim, S. H. T., & Rashid, T. A. (2021). Real-time COVID-19 diagnosis from X-Ray images using deep CNN and extreme learning machines stabilized by chimp optimization algorithm. *Biomedical Signal Processing and Control*, 68, 102764.
27. Krishnaraj, N., Elhoseny, M., Lydia, E. L., Shankar, K., & ALDabbas, O. (2021). An efficient radix trie-based semantic visual indexing model for large-scale image retrieval in cloud environment. *Software Practice and Experience*, 51, 489–502.
28. Xu, X., Cao, D., Zhou, Y., & Gao, J. (2020). Application of neural network algorithm in fault diagnosis of mechanical intelligence. *Mechanical Systems and Signal Processing*, 141, 106625.
29. Saračević, M., Adamović, S., Maček, N., Elhoseny, M., & Sarhan, S. (2020). Cryptographic keys exchange model for smart city applications. *IET Intelligent Transport Systems*, 14, 1456–1464.
30. Krishnaraj, N., Elhoseny, M., Thenmozhi, M., Selim, M. M., & Shankar, K. (2020). Deep learning model for real-time image compression in Internet of Underwater Things (IoUT). *Journal of Real-Time Image Processing*, 17, 2097–2111.
31. Tang, Y., & Elhoseny, M. (2019). Computer network security evaluation simulation model based on neural network. *J Intell Fuzzy Syst*, 37, 3197–3204.
32. Elhoseny, M., Yuan, X., El-Minir, H. K., Riad, A. M. (2014). Extending self-organizing network availability using genetic algorithm. In *Fifth International Conference on Computing, Communications and Networking Technologies*, IEEE, (pp. 1–6).
33. Tharwat, A., Mahdi, H., Elhoseny, M., & Hassanien, A. E. (2018). Recognizing human activity in mobile crowdsensing environment using optimized k-NN algorithm. *Expert Systems with Applications*, 107, 32–44.
34. Jiang, D., Chen, F.-X., Zhou, H., Lu, Y.-Y., Tan, H., Yu, S.-J., et al. (2020). Bioenergetic crosstalk between mesenchymal stem cells and various ocular cells through the intercellular trafficking of mitochondria. *Theranostics*, 10, 7260.
35. Ali, M., Jung, L. T., Abdel-Aty, A.-H., Abubakar, M. Y., Elhoseny, M., & Ali, I. (2020). Semantic-k-NN algorithm: An enhanced version of traditional k-NN algorithm. *Expert Systems with Applications*, 151, 113374.
36. Zhu, L., Kong, L., & Zhang, C. (2020). Numerical study on hysteretic behaviour of horizontal-connection and energy-dissipation structures developed for prefabricated shear walls. *Applied Sciences*, 10, 1240.
37. Le, Q. V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Ng, A. Y. (2011). On optimization methods for deep learning. In *Proc. 28th Int. Conf. Mach. Learn. ICML 2011*.
38. Zaher, M., Shehab, A., Elhoseny, M., & Farahat, F. F. (2020). Unsupervised model for detecting plagiarism in internet-based handwritten Arabic documents. *Journal of Organizational and End User Computing*, 32, 42–66.
39. Zhang, C., Abedini, M., & Mehrmashhadi, J. (2020). Development of pressure-impulse models and residual capacity assessment of RC columns using high fidelity Arbitrary Lagrangian-Eulerian simulation. *Engineering Structures*, 224, 111219.
40. Liu, Y., Zhang, B., Feng, Y., Lv, X., Ji, D., Niu, Z., et al. (2020). Development of 340-GHz Transceiver Front End Based on GaAs Monolithic Integration Technology for THz Active Imaging Array. *Applied Sciences*, 10, 7924.
41. Martens, J. (2010). Deep learning via Hessian-free optimization. In *ICML 2010 - Proceedings*, 27th Int. Conf. Mach. Learn.
42. Puri, V., Jha, S., Kumar, R., Priyadarshini, I., Abdel-Basset, M., Elhoseny, M., et al. (2019). A hybrid artificial intelligence and internet of things model for generation of renewable resource of energy. *IEEE Access*, 7, 111181–111191.
43. Cao, B., Zhao, J., Yang, P., Yang, P., Liu, X., Qi, J., et al. (2019). Multi-objective feature selection for microarray data via distributed parallel algorithms. *Future Generation Computer Systems*, 100, 952–981.
44. Dorri, A., Kanhere, S. S., & Jurdak, R. (2019). MOF-BC: A memory optimized and flexible block-chain for large scale networks. *Future Generation Computer Systems*, 92, 357–373.
45. Yuan, X., Li, D., Mohapatra, D., & Elhoseny, M. (2018). Automatic removal of complex shadows from indoor videos using transfer learning and dynamic thresholding. *Computers & Electrical Engineering*, 70, 813–825.

46. Alam, Z., Sun, L., Zhang, C., Su, Z., & Samali, B. (2021). Experimental and numerical investigation on the complex behaviour of the localised seismic response in a multi-storey plan-asymmetric structure. *Structure and Infrastructure Engineering*, 17, 86–102.
47. Hurrah, N. N., Parah, S. A., Loan, N. A., Sheikh, J. A., Elhoseny, M., & Muhammad, K. (2019). Dual watermarking framework for privacy protection and content authentication of multimedia. *Future Generation Computer Systems*, 94, 654–673.
48. Shankar, K., & Elhoseny, M. (2019). Trust based cluster head election of secure message transmission in MANET using multi secure protocol with TDES. *JUCS Journal of Universal Computer Science*, 25, 1221–1239.
49. Eassa, A. M., Elhoseny, M., El-Bakry, H. M., & Salama, A. S. (2018). NoSQL injection attack detection in web applications using RESTful service. *Programming and Computer Software*, 44, 435–444.
50. Muhammad, K., Khan, S., Elhoseny, M., Ahmed, S. H., & Baik, S. W. (2019). Efficient fire detection for uncertain surveillance environment. *IEEE Transactions on Industrial Informatics*, 15, 3113–3122.
51. Dutta, A. K., Elhoseny, M., Dahiya, V., & Shankar, K. (2020). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. *Transactions on Emerging Telecommunications Technologies*, 31, e3690.
52. Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527–1554.
53. Murugan, B. S., Elhoseny, M., Shankar, K., & Uthayakumar, J. (2019). Region-based scalable smart system for anomaly detection in pedestrian walkways. *Computers & Electrical Engineering*, 75, 146–160.
54. Khishe, M., & Mosavi, M. R. (2020). Chimp optimization algorithm. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2020.113338>
55. Metawa, N., Elhoseny, M., Hassan, M. K., Hassanien, A. E. (2016). Loan portfolio optimization using genetic algorithm: a case of credit constraints. In 2016 12th International Computer Engineering Conference, IEEE; (pp. 59–64).
56. Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *Journal of Parallel and Distributed Computing*, 142, 36–45.
57. Hosseinabadi, A. A. R., Vahidi, J., Saemi, B., Sangaiah, A. K., & Elhoseny, M. (2019). Extended genetic algorithm for solving open-shop scheduling problem. *Soft Computing*, 23, 5099–5116.
58. Zhang, B., Ji, D., Fang, D., Liang, S., Fan, Y., & Chen, X. (2019). A novel 220-GHz GaN diode on-chip tripler with high driven power. *IEEE Electron Device Letters*, 40, 780–783.
59. You, Z., & Pu, Y. (2015). The genetic convolutional neural network model based on random sample. *International Journal of u-and e-Service, Science and Technology*. <https://doi.org/10.14257/ijunesst.2015.8.11.31>
60. Elhoseny, M., Shehab, A., & Yuan, X. (2017). Optimizing robot path in dynamic environments using genetic algorithm and bezier curve. *Journal of Intelligent & Fuzzy Systems*, 33, 2305–2316.
61. Rosa, G., Papa, J., Marana, A., Scheirer, W., Cox, D. (2015). Fine-tuning convolutional neural networks using Harmony Search. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. https://doi.org/10.1007/978-3-319-25751-8_82.
62. Fan, H., Zheng, L., Yan, C., & Yang, Y. (2018). Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications*. <https://doi.org/10.1145/3243316>
63. Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Lv, J. (2020). Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*. <https://doi.org/10.1109/TCYB.2020.2983860>
64. Ewees, A. A., Abd El Aziz, M., Elhoseny, M. (2017). Social-spider optimization algorithm for improving ANFIS to predict biochar yield. In 2017 8th International Conference on Computing, Communications and Networking Technologies, IEEE; (pp. 1–6).
65. Rizk-Allah, R. M., Hassanien, A. E., & Elhoseny, M. (2018). A multi-objective transportation model under neutrosophic environment. *Computers & Electrical Engineering*, 69, 705–719.
66. El-Hasnony, I. M., Barakat, S. I., Elhoseny, M., & Mostafa, R. R. (2020). Improved feature selection model for big data analytics. *IEEE Access*, 8, 66989–67004.
67. Elhoseny, M. (2020). Intelligent firefly-based algorithm with Levy distribution (FF-L) for multicast routing in vehicular communications. *Expert Systems with Applications*, 140, 112889.

68. Lydia, E. L., Raj, J. S., Pandi Selvam, R., Elhoseny, M., & Shankar, K. (2021). Application of discrete transforms with selective coefficients for blind image watermarking. *Transactions on Emerging Telecommunications Technologies*, 32, e3771.
69. Elhoseny, M., & Shankar, K. (2019). Reliable data transmission model for mobile ad hoc network using signcryption technique. *IEEE Transactions on Reliability*, 69, 1077–1086.
70. Lakshmanaprabu, S. K., Elhoseny, M., & Shankar, K. (2019). Optimal tuning of decentralized fractional order PID controllers for TITO process using equivalent transfer function. *Cognitive Systems Research*, 58, 292–303.
71. Valayapalayam Kittusamy, S. R., Elhoseny, M., Kathiresan, S. (2019). An enhanced whale optimization algorithm for vehicular communication networks. *International Journal of Communication Systems* e3953.
72. Turkoglu, M. (2021). COVIDetectionNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble. *Applied Intelligence*, 51, 1213–1226.
73. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
74. Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., & Jamalipour, S. G. (2020). Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Medical Image Analysis*. <https://doi.org/10.1016/j.media.2020.101794>
75. Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., Ghassemi, M. (2020). Covid-19 image data collection: Prospective predictions are the future. *ArXiv Prepr ArXiv200611988*.
76. Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., et al. (2019). CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. 33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019. <https://doi.org/10.1609/aaai.v33i01.3301590>.
77. Elsayed, W., Elhoseny, M., Sabbeh, S., & Riad, A. (2018). Self-maintenance model for wireless sensor networks. *Computers & Electrical Engineering*, 70, 799–812.
78. Gaber, T., Abdelwahab, S., Elhoseny, M., & Hassanien, A. E. (2018). Trust-based secure clustering in WSN-based intelligent transportation systems. *Computer Networks*, 146, 151–158.
79. Zhang, B., Niu, Z., Wang, J., Ji, D., Zhou, T., Liu, Y., et al. (2020). Four-hundred gigahertz broadband multi-branch waveguide coupler. *IET Microwaves, Antennas and Propagation*, 14, 1175–1179.
80. Niu, Z. Q., Zhang, B., Li, D. T., Ji, D. F., Liu, Y., Feng, Y. N., et al. (2021). A mechanical reliability study of 3dB waveguide hybrid couplers in the submillimeter and terahertz band. *Frontiers of Information Technology & Electronic Engineering*, 22, 1104–1113. <https://doi.org/10.1631/FITEE.2000229>
81. Wang, Y., Yuan, L. P., Khishe, M., Moridi, A., & Mohammadzade, F. (2020). Training RBF NN using sine-cosine algorithm for sonar target classification. *Archives of Acoustics*. <https://doi.org/10.24425/aoa.2020.135281>
82. Yang, M., Li, C., Zhang, Y., Jia, D., Zhang, X., Hou, Y., et al. (2017). Maximum undeformed equivalent chip thickness for ductile-brittle transition of zirconia ceramics under different lubrication conditions. *International Journal of Machine Tools and Manufacture*, 122, 55–65.
83. Berg, H., Hjelmervik, K. T. (2019). Classification of anti-submarine warfare sonar targets using a deep neural network. *Ocean. 2018 MTS/IEEE Charleston, Ocean 2018*. <https://doi.org/10.1109/OCEANS.2018.8604847>.
84. Aridoss, M., Dhasarathan, C., Dumka, A., & Loganathan, J. (2020). DUICM deep underwater image classification model using convolutional neural networks. *International Journal of Grid and High Performance Computing*, 12, 88–100.
85. Hall, J. J., Azimi-Sadjadi, M. R., Kargl, S. G., Zhao, Y., & Williams, K. L. (2019). Underwater unexploded ordnance (UXO) classification using a matched subspace classifier with adaptive dictionaries. *IEEE Journal of Oceanic Engineering*. <https://doi.org/10.1109/JOE.2018.2835538>
86. Liu, M., Li, C., Cao, C., Wang, L., Li, X., Che, J., et al. (2021). Walnut fruit processing equipment: academic insights and perspectives. *Food Engineering Reviews*, 13, 1–36.
87. Wang, Y., Li, C., Zhang, Y., Yang, M., Li, B., Jia, D., et al. (2016). Experimental evaluation of the lubrication properties of the wheel/workpiece interface in minimum quantity lubrication (MQL) grinding using different types of vegetable oils. *Journal of Cleaner Production*, 127, 487–499.
88. Mahmud, T., Rahman, M. A., & Fattah, S. A. (2020). CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Computers in Biology and Medicine*. <https://doi.org/10.1016/j.compbiomed.2020.103869>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xusheng Wang was born in Shanxi, P.R. China, in 1988. He received the Ph.D. degree from University of Paris Sud, France. Now, he works in Xi'an University of Technology. His research interest include computer network computational intelligence, and integrated circuit design.

Cunqi Gong, male, 42, his research interest is clinical biochemistry. Department of Clinical Laboratory, Jining No. 1 People's Hospital, 6 Jiankang Road, Jining, Shandong 272,000, P.R. China.

Mohammad Khishe was born on 1985 in Nahavand, Iran. He received the B.Sc. degree in Maritime Electronic and Communication Engineering from Imam Khomeini Maritime Sciences University, Nowshahr, Iran in 2007 and also, M.Sc. and Ph. D. degrees in Electronic Engineering from Islamic Azad University, Qazvin branch and Iran University of Science and Technology, Tehran in 2012 and 2017, respectively. Since 2011, he is a faculty member of Maritime Electrical and Electronic Engineering at Imam Khomeini Maritime Sciences University, Nowshahr, Iran. His research interests are Digital Signal Processing, Artificial Neural Networks, Meta-heuristic Algorithms, Sonar and Radar Signal Processing, and FPGA design.

Mokhtar Mohammadi received the B.S. degree in computer engineering from Shahed University, Tehran, Iran, in 2003, the M.S. degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 2012, and the Ph.D. degree in computer engineering from Shahrood University of Technology, Shahrood, Iran, in 2018. His current research interests include signal processing, time–frequency analysis, and machine learning. He is currently with the Department of Information Technology, Lebanese French University-Erbil, Iraq.

Professor Dr. Tarik A. Rashid received his Ph.D. in Computer Science and Informatics from College of Engineering, Mathematical and Physical Sciences, University College Dublin (UCD) in 2001–2006. He pursued his Post-Doctoral Fellow at the Computer Science and Informatics School, College of Engineering, Mathematical and Physical Sciences, University College Dublin (UCD) from 2006–2007. He joined the University of Kurdistan Hewlêr (UKH) in 2017.

Authors and Affiliations

Xusheng Wang¹ · Cunqi Gong² · Mohammad Khishe³ · Mokhtar Mohammadi⁴ · Tarik A. Rashid⁵

¹ Xi'an University of Technology, Xi'an 710048, Shaanxi, China

² Department of Clinical Laboratory, Jining No.1 People's Hospital, Jining 272011, Shandong, China

³ Department of Electronic Engineering, Imam Khomeini Marine Science University, Nowshahr, Iran

⁴ Department of Information Technology, College of Engineering and Computer Science, Lebanese French University, Erbil, Kurdistan Region, Iraq

⁵ Computer Science and Engineering Department, Science and Engineering School, University of Kurdistan Hewler, Erbil, KRG, Iraq