



Article

Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm

Mohamed Abd Elaziz ^{1,2,*} , Abdelghani Dahou ³, Naser A. Alsaleh ⁴ , Ammar H. Elsheikh ^{5,*} , Amal I. Saba ⁶ and Mahmoud Ahmadein ^{4,5} 

¹ Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

² Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates

³ Mathematics and Computer Science Department, University of Ahmed DRAIA, Adrar 01000, Algeria; dahou.abdghani@univ-adrar.edu.dz

⁴ Mechanical Engineering Department, Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia; naalsaleh@imamu.edu.sa (N.A.A.); m.ahmadein@f-eng.tanta.edu.eg (M.A.)

⁵ Department of Production Engineering and Mechanical Design, Faculty of Engineering, Tanta University, Tanta 31527, Egypt

⁶ Department of Histology, Faculty of Medicine, Tanta University, Tanta 31527, Egypt; amal.saba@med.tanta.edu.eg

* Correspondence: abd_el_aziz_m@yahoo.com (M.A.E.); ammar_elsheikh@f-eng.tanta.edu.eg (A.H.E.)

Abstract: Currently, the world is still facing a COVID-19 (coronavirus disease 2019) classified as a highly infectious disease due to its rapid spreading. The shortage of X-ray machines may lead to critical situations and delay the diagnosis results, increasing the number of deaths. Therefore, the exploitation of deep learning (DL) and optimization algorithms can be advantageous in early diagnosis and COVID-19 detection. In this paper, we propose a framework for COVID-19 images classification using hybridization of DL and swarm-based algorithms. The MobileNetV3 is used as a backbone feature extraction to learn and extract relevant image representations as a DL model. As a swarm-based algorithm, the Aquila Optimizer (Aqu) is used as a feature selector to reduce the dimensionality of the image representations and improve the classification accuracy using only the most essential selected features. To validate the proposed framework, two datasets with X-ray and CT COVID-19 images are used. The obtained results from the experiments show a good performance of the proposed framework in terms of classification accuracy and dimensionality reduction during the feature extraction and selection phases. The Aqu feature selection algorithm achieves accuracy better than other methods in terms of performance metrics.

Keywords: feature selection; metaheuristic; atomic orbital search; dynamic opposite-based learning



Citation: Abd Elaziz, M.; Dahou, A.; Alsaleh, N.A.; Elsheikh, A.H.; Saba, A.I.; Ahmadein, M. Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm. *Entropy* **2021**, *23*, 1383. <https://doi.org/10.3390/e23111383>

Academic Editor: Udo Von Toussaint

Received: 13 September 2021

Accepted: 13 October 2021

Published: 22 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In December 2019, COVID-19 was declared as a new coronavirus which resulted in an explosive outbreak in China [1]. Due to its highly contagious characteristics, it swept over more than 220 countries with more than 200 million confirmed cases and more than 4.3 million deaths. This pandemic has the second rank among all documented pandemics based on the number of deaths after the 1918 flu pandemic [2]. More than 40% of these confirmed cases and deaths are reported in only three countries: namely the United States, Brazil, and India, as shown in Figure 1. The symptoms of this disease are fever, dry cough, loss of smell and taste, dyspnea, fatigue, and malaise [3]. It may produce acute complications for persons who suffer from other chronic diseases such as hypertension, respiratory system diseases, autoimmune diseases, diabetes, and cardiovascular diseases.

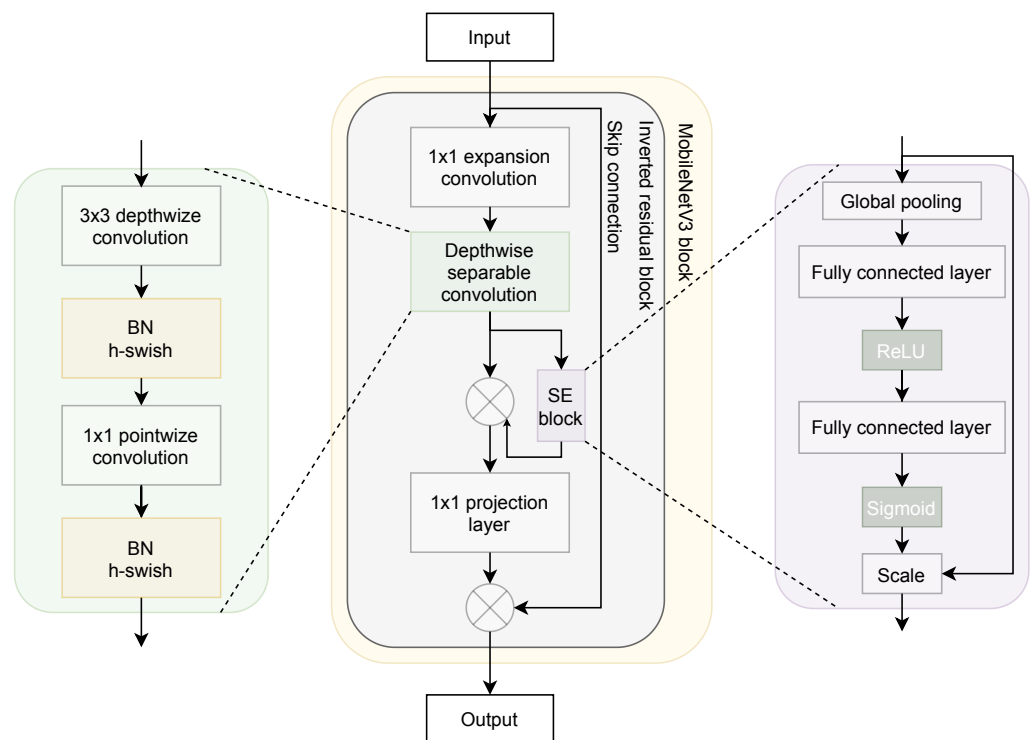


Figure 1. The structure of MobileNetV3 blocks and components.

Diagnosis of COVID-19 infection using X-ray imaging of the chest has been reported as an accurate diagnosis technique [4]. The conventional human-based detecting technique that depends on the technical experience of a physician or radiologist is inefficient, inaccurate, time consuming, limited, and outdated [5]. The implementation of this technique is subjected to human errors, resulting in the misdiagnosing of the disease. This problem is exacerbated in remote regions where there is a lack of expert physicians. The development of advanced artificial intelligence techniques (AI) allows medical researchers and scientists to develop advanced tools, software, and instruments that can help medical radiologists overcome the problems related to human-based detecting techniques [6]. The last two years have seen a surge in the applications of AI in the diagnosis and forecasting of COVID-19 [7–12]. Many approaches have been developed to detect and differentiate between COVID-19 disease and conventional viral pneumonia using chest X-ray and CT images [13]. Zhao et al. [14] investigated the relationship between COVID-19 pneumonia and CT images of the chest. The results revealed typical features observed in the examined images of COVID-19 cases; this finding allows researchers to apply AI in the image processing of chest X-rays and CT of COVID-19 cases. Bernheim et al. [15] reported that the CT images of the infected COVID-19 cases are characterized by the existence of typical hallmarks such as consolidative opacities, ground-glass opacities, and crazy-paving patterns. Pezzano et al. [16] developed a convolutional neural network (CNN) to detect ground-glass opacities in the CT images of COVID-19 infected cases. Yasin et al. [17] correlated the disease severity to patients' sex and age based on X-ray images.

The most common used AI reported in the literature to diagnose COVID-19 infections based on CT or X-ray images is CNN models such as VGG-16, VGG-19, Xception, AlexNet, ResNet50V2, CoroNet, LeNet-5, ResNet18, and ResNet 50 [18,19]. The integration between machine learning methods and the so-called metaheuristic (MH) optimization techniques [20,21] has also been reported in the literature as a correct approach with reasonable computational cost. Canayaz [22] developed a hybrid deep neural network-integrated with metaheuristic optimizers to diagnose COVID-19 infections. A dataset contains three groups of X-ray images, namely normal, pneumonia, and COVID-19, and was used to train the model. The images were preprocessed using the contrast-enhancing technique.

Features were extracted using deep learning models, namely GoogleNet, VGG19, AlexNet, and ResNet. The best features were selected using two metaheuristic optimizers, namely grey wolf optimizer and particle swarm optimizer. Then, the features were classified using a support vector machine.

An advanced hybrid classification approach consists of a CNN model and the marine predators optimizer, and the fractional-order algorithm has been developed to detect the infection of COVID-19 based on X-ray images [23]. CNN was used to extract features from the images, while the marine predators optimizer integrated with a fractional-order algorithm was used to select the essential features. The results obtained by the proposed approach were compared with those obtained by other metaheuristic optimizers such as Henry gas solubility optimizer, slime mold algorithm optimizer, whale optimization optimizer, particle swarm optimizer, sine cosine algorithm, genetic algorithm, grey wolf optimizer, Harris hawks optimizer, and standalone marine predators optimizer. The proposed approach results revealed its excellent performance compared to the other algorithms in terms of high detection accuracy and low computational cost.

A hybrid metaheuristic optimizing the approach, in which the marine predators optimizer is incorporated with the moth-flame optimizer, was used for image segmentation of COVID-19 cases [24]. The latter optimizer was used as a subroutine in the former optimizer to avoid trapping into local optima. The proposed approach outperformed other advanced optimizers such as the Harris hawks optimizer, grey wolf optimizer, particle swarm optimization, grasshopper algorithm, cuckoo search optimizer, spherical search optimizer, moth-flame optimizer, and standalone marine predators optimizer. An improved cuckoo search optimizer using a fractional calculus algorithm was used to classify X-ray images of COVID-19 cases into normal, COVID-19, and pneumonia patients [25]. Four heavy-tailed distributions, namely Cauchy distribution, Mittag-Leffler distribution, Weibull distribution, and Pareto distribution, were utilized to strengthen the model performance. The proposed model showed its superiority over other feature selection techniques such as the genetic algorithm, Henry gas solubility optimizer, Harris hawks optimizer, salp swarm optimizer, whale algorithm, and grey wolf optimizer.

Another machine learning/metaheuristic optimizing approach was proposed to detect COVID-19 infections based on X-ray images [26]. The important features were extracted from the processed images using a machine learning algorithm called fractional exponent moments. The computational process was accelerated using a multicore computational scheme. Then, a hybrid manta-ray foraging/differential evolution optimizer was used to select the important features. The selection of essential features and excluding irrelevant features help accelerate the classification process, which is accomplished using the k-nearest neighbors technique.

However, most of the presented COVID-19 image classification methods have limitations that affect classification accuracy. It has been noticed that these limitations result from either the strategy used to extract the features or the approach used to reduce the number of selected features. Therefore, this motivated us to present an alternative COVID-19 image classification method.

Within this study, we developed an alternative COVID-19 image classification technique that combined the advantages of MobileNetV3 and a new MH technique named Aquila Optimizer (Aqu) [27]. The MobileNetV3 is used to extract the features from the tested images and then, using the binary version of Aquila Optimizer (Aqu) as a feature selection (FS) method, to determine the relevant features. Aqu algorithm has established its performance in several applications, including oil production forecasting [28], and global optimization [29].

The main contributions of this paper are summarized as follows:

1. Develop a COVID-19 cases detection framework by incorporating MobileNetV3 and Aquila Optimizer as feature extraction and selection algorithms, respectively.

2. Propose a new feature selection using the binary version of Aquila Optimizer, in addition, using MobileNetV3 to learn and extract the image embedding from the COVID-19 images.
3. Evaluate the performance of the developed method using two datasets with X-ray and CT images of COVID-19.
4. Compare the efficiency of the developed approach with other methods.

The structure of the remaining parts of this study is as follows: Section 2 introduces the background of MobileNetV3 and Aquila Optimizer algorithm. Section 3 presents the stages of the developed method. The comparison results are given in Section 4. Finally, we introduce the conclusion and future work of the current study in Section 5.

2. Background

2.1. MobileNetV3

Convolutional neural network architectures have been proposed recently to tackle many different problems and improve their performance in terms of speed and size. Efficient convolutional neural networks implementing the depthwise convolution structure such as NASNet [30], MobileNets [31,32], EfficientNet [33], MnasNet [34], and ShuffleNets [35] are considered as a key technique in many computer vision applications [36–39] known by fast training process. The depthwise convolutional kernel is a learnable parameter applied to each input channel separately from the training images to extract spatial information. Moreover, depthwise convolutional kernels are shared across all input channels, increasing model efficiency and reducing computation cost. However, the depthwise convolutional kernel size can be difficult to learn, thus increasing the complexity of the training process of the depthwise convolutions. In the upcoming paragraphs, we briefly discuss the recently proposed MobileNetV3 [32] architecture.

Previously developed MobileNetV1 and MobileNetV2 were improved with a new version called MobileNetV3, proposed by Howard et al. [32] using network architecture search (NAS). The used NAS technique called NetAdapt algorithm was used to search for the best kernel size and find the optimized MobileNet architecture to fulfill the low-resourced hardware platforms in terms of size, performance, and latency. The MobileNetV3 introduced several building components and blocks inspired by the previous versions, as shown in Figure 1. In addition, MobileNetV3 possesses a new nonlinearity called hard swish ($h-swish$), which is a modified version of the sigmoid function introduced in [40]. The $h-swish$ nonlinearity is defined as in Equation (1), which is employed to minimize the number of training parameters and reduce the model complexity and size.

$$h-swish(x) = x \cdot \sigma(x) \quad (1)$$

$$\sigma(x) = \frac{ReLU6(x+3)}{6} \quad (2)$$

where $\sigma(x)$ represents the piece-wise linear hard analog function.

As shown in Figure 1, the MobileNetV3 block contains a core building block called the inverted residual block, which includes a depthwise separable convolution block and a squeeze-and-excitation block [34]. The inverted residual block is inspired from the bottleneck blocks [41], where it uses an inverted residual connection to connect the input and output features on the same channels and improve the features representations with low memory usage. The depthwise separable convolutional contains a depthwise convolutional kernel applied to each channel and a 1×1 pointwise convolutional kernel with batch normalization layer (BN) and the $ReLU$ or $h-swish$ activation functions. The depthwise separable convolutional is used to alter the traditional convolution block and reduce the model capacity. The squeeze-and-excitation (SE) block is used to pay more attention to the relevant features on each channel during training.

2.2. Aquila Optimizer (Aqu)

Aquila Optimizer (Aqu) [27] is a new population-based optimizer that is classified as a metaheuristic optimization technique. The mathematical formulation of this optimizer is presented in this section. The social behavior of Aquila inspires the Aqu algorithm during the hunting process of its prey. Like other population-based metaheuristic techniques, Aqu starts with N agents with an X initial population. This initialization process is executed using the following formula.

$$X_{ij} = r_1 \times (UB_j - LB_j) + LB_j, \quad i = 1, 2, \dots, N \quad j = 1, 2, \dots, Dim \quad (3)$$

where LB_j and UB_j are the lower and upper bounds of the exploration domain. $r_1 \in [0, 1]$ is a randomly generated parameter, and Dim is the population size.

Once the population is initialized, the algorithm executes exploitation and exploration processes until the optimal solution is obtained. There are two main implemented strategies during exploitation and exploration processes [27].

The first strategy is implemented to execute the exploration process considering the average agents (X_M) and the best agent X_b . This strategy is mathematically formulated as follows:

$$X_i(t+1) = X_b(t) \times \left(\frac{1-t}{T}\right) + (X_M(t) - X_b(t) * rand), \quad (4)$$

$$X_M(t) = \frac{1}{N} \sum_{i=1}^N X(t), \quad \forall j = 1, 2, \dots, Dim \quad (5)$$

where T is the total number of iterations, while the search process is controlled using $\left(\frac{1-t}{T}\right)$.

In the second strategy, the exploration of the agents is updated based on the Levy flight ($Levy(D)$) distribution and X_b . This strategy is mathematically formulated as follows:

$$X_i(t+1) = X_b(t) \times Levy(D) + X_R(t) + (y - x) * rand, \quad (6)$$

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \quad (7)$$

where $\beta = 1.5$ and $s = 0.01$, while v and u are randomly generated parameters. In Equation (6), X_R is a randomly selected agent. Moreover, x and y are used to follow the spiral tracking shape, and they are mathematically formulated as follows:

$$y = r \times \cos(\theta), \quad x = r \times \sin(\theta) \quad (8)$$

$$r = r_1 + U \times D_1, \quad \theta = -\omega \times D_1 + \theta_1, \quad \theta_1 = \frac{3 \times \pi}{2} \quad (9)$$

where $U = 0.00565$ and $\omega = 0.005$. $r_1 \in [0, 20]$ is a randomly generated parameter.

In [27], the first strategy is utilized to update the agents during the exploitation process based on X_M and X_b , and it is mathematically formulated as follows:

$$X_i(t+1) = (X_b(t) - X_M(t)) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta, \quad (10)$$

where δ and α denote the adjustment parameters of exploitation process. $rand \in [0, 1]$ is a randomly generated parameter.

In the second strategy, the agent is updated during the exploitation process using the quality function QF , and X_b , $Levy$. This strategy is mathematically formulated as follows:

$$X_i(t+1) = QF \times X_b(t) - (G_1 \times X(t) \times rand) - G_2 \times Levy(D) + rand \times G_1, \quad (11)$$

$$QF(t) = t^{\frac{2 \times \text{rand}() - 1}{(1-T)^2}} \quad (12)$$

Furthermore, G_1 specifies the employed motions during tracking the best solution, and it is given as:

$$G_1 = 2 \times \text{rand}() - 1, G_2 = 2 \times \left(1 - \frac{t}{T}\right) \quad (13)$$

rand is a function that generates random values, and G_2 specifies decreased values from 2 to 0, and it is given as:

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \quad (14)$$

3. Proposed Framework

In this section, the general framework of the developed COVID-19 image classification method is described.

3.1. MobileNetV3 for Feature Extraction

The fine-tuning process of MobileNetV3 and feature extraction phase are described in this section. The main objective is to extract relevant image embeddings relying on a pretrained model on different COVID-19 image datasets. Meanwhile, the extracted image embedding in this phase is fed into the feature selection phase, discussed in the next section. Compared to previous studies, the feature selection phase employs a new swarm optimization technique to enhance the recognition accuracy, select only essential features, and reduce the features representation space of the overall proposed framework.

As described in Section 2.1, efficient convolutional neural networks such as MobileNetV3 [32] act as suitable models to perform image recognition where they can act as a core component in the feature extraction phase. We used a pretrained model of MobileNetV3 trained on the ImageNet dataset to avoid training the model from scratch and speed up the learning process. More specifically, the MobileNetV3-Large pretrained model was used in our experiments and adapted to the COVID-19 recognition task via transfer learning and fine-tuning. We follow the standard procedure to fine-tune the MobileNetV3 model and extract the relevant image embeddings. First, we change the top two output layers of the MobileNetV3 model used for image classification with a 1×1 point-wise convolution to extract images features. The 1×1 point-wise convolution can act as a multilayer perceptron (MLP) to perform image classification or dimensionality reduction by integrating different nonlinearity operations. In addition, other 1×1 point-wise convolutions have been added on the top of the model for fine-tuning the model's weights on different datasets based on the classification task. Second, after fine-tuning the model, we flatten the output of the 1×1 point-wise convolution used for feature extraction to generate image embeddings with the size of 128 for each image in the dataset. Lastly, the extracted image embeddings are fed into the feature selection phase.

Figure 2 shows the architecture of the modified MobileNetV3 for COVID-19 images feature extraction. The feature extraction phase was performed after fine-tuning the model for 100 epochs during ten randomly initialized runs where we used the model resulting in the highest classification accuracy on each dataset. A batch of size 32 and a stochastic gradient descent approach named RMSprop were used to fine-tune the model with a learning rate set to 1×10^{-4} . Data augmentation was employed during the data preprocessing phase to overcome overfitting and improve the model's generalization. The data augmentation transformation such as random crop, random horizontal flip, color jitter, and random vertical flip was used alongside original image resizing to shape 224×224 .

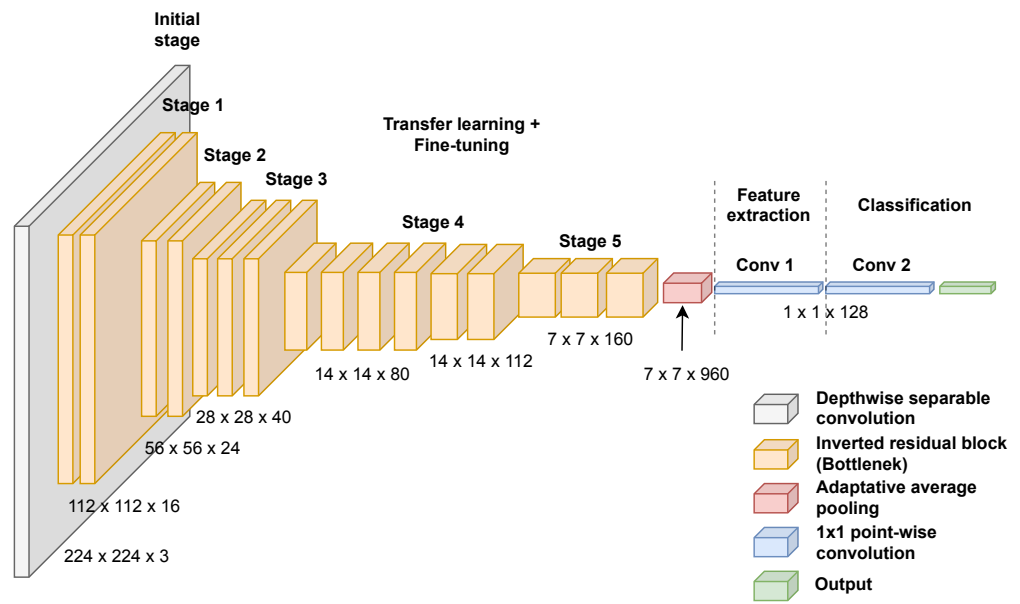


Figure 2. The architecture of MobileNetV3 used for feature extraction.

3.2. Developed Aqu FS Algorithm

To apply the Aqu algorithm as an FS method, its binary version is developed as given in Figure 3. The main target of this conversion is to prepare the Aqu algorithm for working with the discrete problem since its original version is implemented to work with real-valued problems only. There are two stages of Aqu as FS technique, and the details of each stage are given in the following sections.

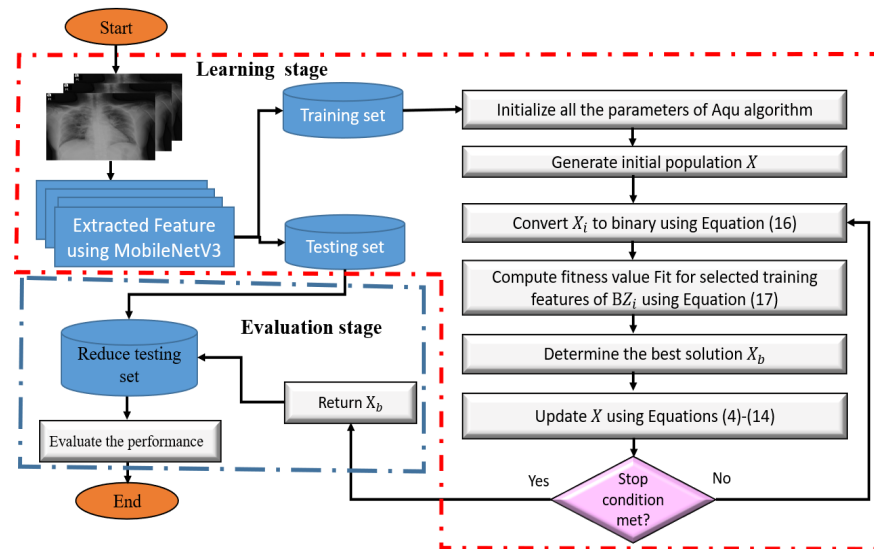


Figure 3. Steps of Aqu for FS problem.

3.2.1. First Stage: Learning of Model

This stage aims to use the training set to learn the model to select the most relevant features, and in this study, we used 70% from the dataset as a training set. The first process in this stage is to set the initial value for the population X , which contains N agents. This process is defined as:

$$X_i = rand * (U - L) + L, i = 1, 2, \dots, N, j = 1, 2, \dots, N_F \quad (15)$$

where N_F represents the number of features, whereas U and L are the limits of search domain.

The next step is to obtain the binary form for each X_i , and this was produced using Equation (16).

$$BX_{ij} = \begin{cases} 1 & \text{if } X_{ij} > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Then, the fitness value Fit_i of each X_i is evaluated using the following formula:

$$Fit_i = \lambda \times \gamma_i + (1 - \lambda) \times \left(\frac{|BX_i|}{N_F} \right), \quad (17)$$

In Equation (17), $|BX_i|$ stands for the number of features (i.e., the ones in BX_i), whereas γ_i represents the classification error using the KNN classifier that used the reduced training set based on BX_i . In addition, λ is a weight value used to balance between the two objectives in Equation (17) (i.e., minimizing the selecting features and reducing the error of classification).

After that, the agent with the best fitness value (Fit_b) is considered the best agent X_b . The X_b agent is used to update the other agents according to the operators of the Aqu algorithm as discussed in Equations (4)–(14).

The next step is to check if the terminal conditions are met, then X_b is returned; otherwise, updating the solutions is conducted again.

3.2.2. Second Stage: Evaluation of the Selected Features

Within this stage, the relevant features in the best solution X_b are used to reduce the testing set used as input to the KNN classifier. Later, we compute the performance of the predicted output using various performance measures.

4. Experimental Results

4.1. Dataset Description

This section describes the datasets used in the COVID-19 detection task and the distribution of their corresponding samples. The datasets include two types of images: X-ray and CT scan images (computed tomography scan), where Figure 4 shows examples from each dataset. Our experiments used three different datasets to train and fine-tune the feature extraction model, namely the COVID-CT dataset (dataset1), the COVID-XRay-6432 dataset (dataset2), and the COVID-19 radiography dataset (dataset3). We keep the same data split after extracting image embeddings from each dataset which are fed to a feature selection and classification phase. In the following section, a detailed description of each dataset is given.

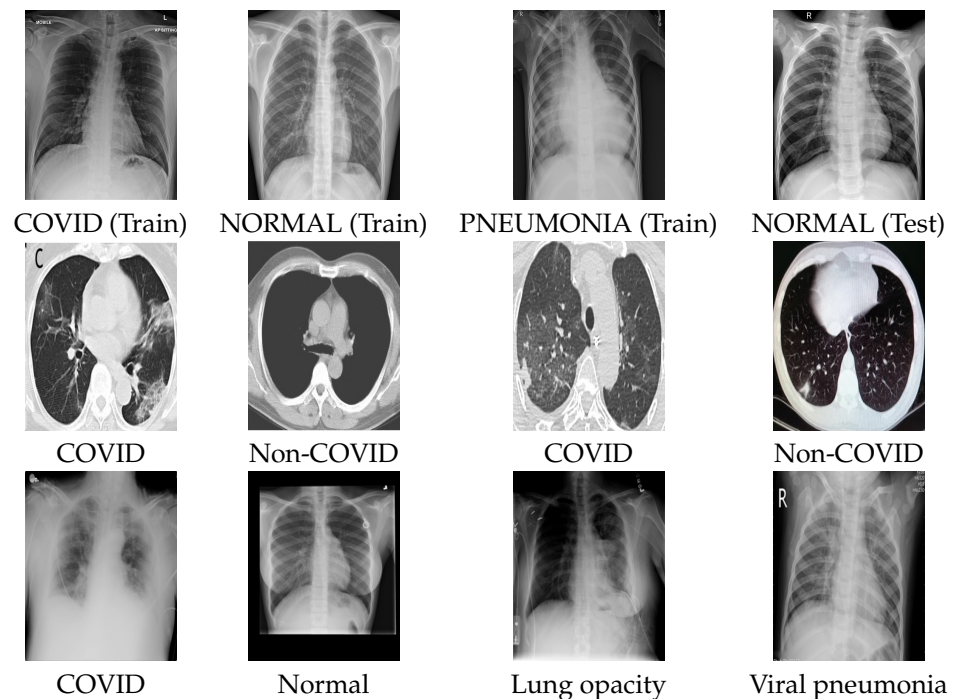


Figure 4. 1st row: COVID-XRay-6432 dataset samples, 2nd row: COVID-CT dataset samples, and 3rd row: COVID-19 radiography dataset samples.

1. COVID-CT dataset: This dataset was collected from two sources, including research papers (for training) and original CT scans donated by hospitals (for testing). For the research papers, the authors [42] collected 760 preprints from two databases including medRxiv <https://www.medrxiv.org/> (accessed on 12 October 2021) and bioRxiv <https://www.biorxiv.org/> (accessed on 12 October 2021). The preprints were collected from papers posted from 19 January to 25 March 2020. In total, 349 CT images labeled as positive were collected from 216 patient cases for COVID-19. In addition, the authors collected 397 negative CT images (Non-Covid19) to build their dataset for a binary classification task from sources including MedPix <https://medpix.nlm.nih.gov/home> (accessed on 12 October 2021) database, the LUNA7 <https://luna16.grand-challenge.org/> (accessed on 12 October 2021) dataset, the Radiopaedia <https://radiopaedia.org/articles/covid-19-3> (accessed on 12 October 2021) website, and PubMed <https://www.ncbi.nlm.nih.gov/pmc/> (accessed on 12 October 2021) Central (PMC). Table 1 lists the number of positive and negative Covid-19 CT images used in our experiments.
2. COVID-XRay-6432 dataset: The dataset is publicly available on Kaggle <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia> (accessed on 12 October 2021) and was gathered from various public resources. The dataset includes 6432 X-ray COVID-19 images distributed on three classes which are COVID-19, PNEUMONIA, and NORMAL (Non-COVID). The training set comprises 80% of the dataset, and the test set comprises 20% of the dataset. In our experiments, 15% of the training sample is used in the validation set and fine-tuning. Table 2 lists the number of samples in each class.
3. COVID-19 radiography dataset: The dataset was collected by a team of researchers from different countries and universities, including Qatar, Bangladesh, Pakistan, and Malaysia, collaborating with medical doctors. The dataset is freely available and frequently updated on Kaggle <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed on 12 October 2021). The dataset consists of 21,165 chest X-ray (CXR) COVID-19 images distributed on four categories which are COVID19, lung opacity, viral pneumonia, and NORMAL (Non-COVID). In our

experiment, we randomly split the data into 70%, 10%, and 20% for training, validation, and testing sets, respectively. Table 3 lists the number of samples in each class after splitting the data.

Table 1. COVID-CT dataset samples distribution.

	Class	Train	Validation	Test
# patients	COVID	130	32	54
	Non-COVID	105	24	42
# images	COVID	191	60	98
	Non-COVID	234	58	105

Table 2. COVID-XRay-6432 dataset samples distribution.

	Class	Train	Test
# images	COVID	460	116
	Non-COVID	1266	317
	PNEUMONIA	3418	855

Table 3. COVID-19 radiography dataset samples distribution.

	Train	Validation	Test
# images	15,238	1694	4233

4.2. Performance Metrics

To assess the accuracy of the developed model, some statistical parameters were computed, such as the mean of best values, the mean of the worst values (Max), standard deviation, and computational time elapsed during the selection of features. Then, statistical measures were computed during the classification phase. The mathematical form of these measures are given as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (19)$$

$$Specificity = \frac{TN}{TN + FP} \quad (20)$$

$$F_{Score} = 2 \times \frac{Specificity \times Sensitivity}{Specificity + Sensitivity} \quad (21)$$

where “TP” is the abbreviation of true positives and represents the positive COVID-19 images labeled using the proposed classifier correctly. “TN” stands for the true negative samples and represents the negative COVID-19 images that were labeled using the proposed classifier correctly. “FP” is the abbreviation of false positives and represents the positive COVID-19 images labeled using the proposed classifier incorrectly, while “FN” is the abbreviation of false negatives and represents the negative COVID-19 images that were labeled using the proposed classifier incorrectly.

- Best accuracy:

$$Best_{acc} = \max_{1 \leq i \leq r} Accuracy \quad (22)$$

where r denotes the run numbers. Fit_i represents a fitness function value.

To validate the performance of Aqu as an FS method, its results were compared with other well-known FS methods based on MH techniques. For example, whale optimization algorithm (WOA) [43], moth-flame optimization (MFO) [44,45], firefly algorithm (FFA) [46], bat algorithm (BAT) [47], hunger games search (HGS) [48], transient search optimization (TSO) [49], and Aquila Optimizer (Aqu) [27]. In this paper, the parameters of these FS methods are assigned based on the original implementation of each method. However, the common parameters, such as the number of iterations and population size, are set to 20 and 15, respectively. In addition, each FS method conducted 25 runs for a fair comparison between them. All DL training and feature extraction phases were conducted on a GPU (Graphics processing unit) of type GTX1080 from Nvidia, while the feature selection phase has experimented on the Google collaborative platform. For a proper validation of the framework, other DL models such as DenseNet, VGG19, and EfficientNet were exploited as backbone feature extraction methods using their standard architecture and parameters.

4.3. Results and Discussion

In this subsection, the performance of the developed model is evaluated using two datasets as given in Tables 4 and 5 and Figures 5 and 6. In general, it can be noticed from Table 4 that the developed method can improve the performance of classification accuracy among the two tested datasets. For example, to analyze the performance of the developed Aqu over Dataset1, the following points can be observed: firstly, the accuracy of Aqu is better than other methods, which nearly have a difference between the best second algorithm (i.e., HGS) with 1.2%. Secondly, Aqu has a higher Recall, Precision, and F1 score overall than the comparative FS methods such as HGS and BAT, which allocated the second and third ranks, respectively. Similar to dataset1, the efficiency of Aqu using dataset2 is the best in terms of classification accuracy, followed by BAT and HGS. In addition, the recall value of HGS and MFO is better than all other methods (i.e., WOA, FFA, BAT, and TSO) except for the developed Aqu method, whereas the precision value of WOA and TSO allocate the second rank after the Aqu method, which allocates the first rank. In order to analyze the results of the F1-score obtained using the second dataset, it can be observed that the HGS is better than other algorithms, which allocate the second rank after the Aqu algorithm.

Table 4. Comparison between Aqu and other methods in terms of accuracy, recall, precision, and F1-score.

	Dataset1				Dataset2			
	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score
MFO	0.769	0.769	0.771	0.767	0.957	0.956	0.928	0.942
WOA	0.761	0.761	0.764	0.760	0.925	0.920	0.967	0.943
FFA	0.769	0.769	0.771	0.767	0.958	0.812	0.873	0.841
BAT	0.771	0.771	0.775	0.769	0.963	0.944	0.802	0.867
HGS	0.773	0.773	0.776	0.772	0.963	0.958	0.959	0.959
TSO	0.766	0.766	0.770	0.764	0.958	0.827	0.967	0.892
Aqu	0.783	0.783	0.785	0.782	0.974	0.974	0.974	0.974

Table 5. Comparison between Aqu and other methods in terms of CPU time(s) and number of selected features.

	Dataset1		Dataset2	
	CPU Time (s)	#FS	CPU Time (s)	#FS
MFO	3.481	278.5	26.654	243.5
WOA	3.260	248.5	26.572	234
FFA	4.294	260.5	27.009	250
BAT	3.602	256.5	31.179	250.5
HGS	4.115	281	31.028	162.5
TSO	3.197	141.5	24.013	142
Aqu	3.123	130	25.737	140

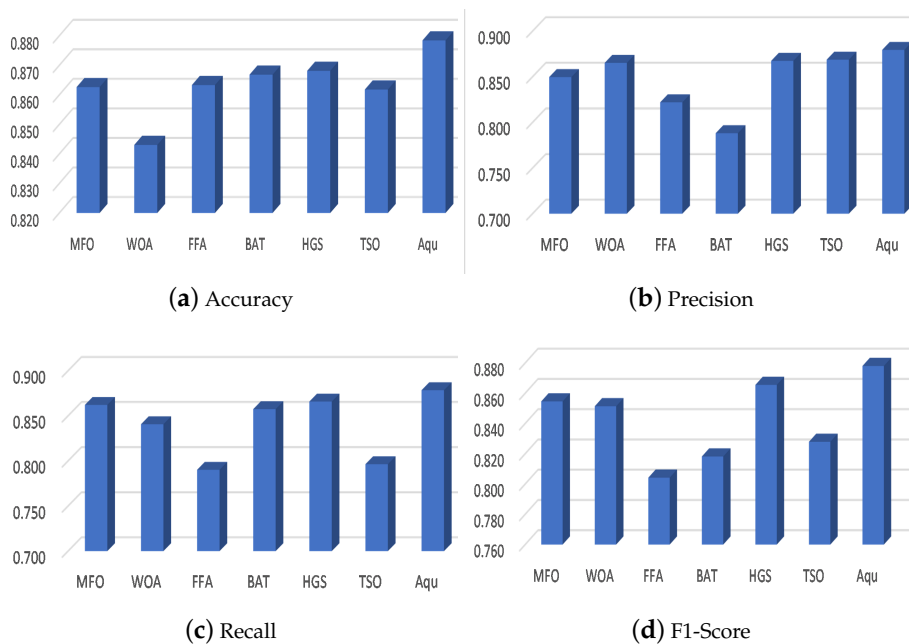


Figure 5. Average of the competitive algorithms in terms of (a) Accuracy, (b) Precision (c) Recall, and (d) F1-score.

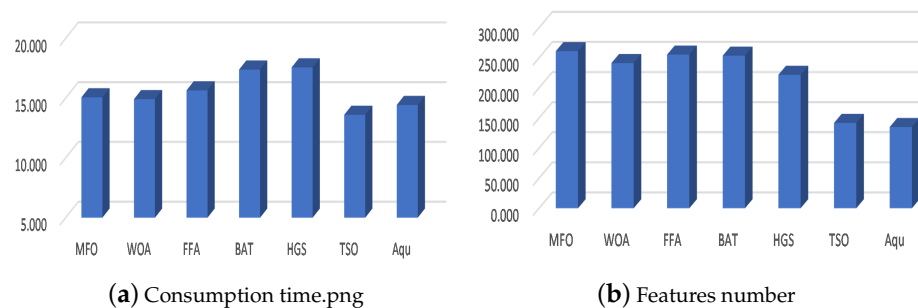


Figure 6. Average of each algorithm among two datasets in terms of (a) CPU time(s) and (b) number of selected features.

Figure 5 depicts the average of each algorithm in terms of accuracy, recall, precision, and F1-score. It can be seen from this figure that the average of the Aqu algorithm over the two datasets is better than other methods in terms of performance measures.

Moreover, the time computational of the FS methods is computed to justify their time complexity as given in Table 5. We can notice from the CPU time(s) values that the

developed Aqu algorithm has the shortest time in dataset1. However, the CPU time(s) of the Aqu algorithm over datasets is the second-best one that followed the TSO algorithm. Meanwhile, the efficiency of Aqu to reduce the number of features is observed from the number of selected features (i.e., the #FS column). Aqu has the smallest number of features, 130 and 140 at Dataset1 and Dataset2, respectively. In addition, from Figure 6 which shows the average over the two datasets in terms of CPU time(s) and #FS, the high superiority of Aqu over other methods can be seen.

4.4. Comparison with Other CNN Types

In this section, the performance of the developed method that combines the MobileNetV3 and Aqu is compared with the other three CNN types networks. These network include VGG19 [50] (Visual Geometry Group), DenseNet [51], and EfficientNet [33]. The main aim of this comparison is to assess the ability of MobileNetV3 to extract the relevant features.

The comparison results between the MobileNetV3 and other CNN types are given in Table 6. From these results, it can be seen that the MobileNetV3 can provide better performance than other CNNs followed by DenseNet that has a high ability to extract relevant features better than the other two networks (i.e., VGG and EfficientNet). The same observation can be noticed in Figure 7, which depicts the average of the accuracy in all the tested FS methods using the feature extracted from each CNN type. In addition, the performance of Aqu based on MobileNetV3 in terms of accuracy among the two tested datasets is given in Figure 8. From these averages, it can be noticed that the developed method provides better results than others. In addition, the ability of Aqu to increase the accuracy classification is better than other FS methods when using different CNN types.

Table 6. Comparison with other CNN types.

	Dataset1				Dataset2			
	VGG	DenseNet	EfficientNet	MobileNetV3	VGG	DenseNet	EfficientNet	MobileNetV3
MFO	0.667	0.766	0.757	0.769	0.939	0.947	0.938	0.957
WOA	0.672	0.751	0.742	0.761	0.937	0.936	0.936	0.925
FFA	0.670	0.784	0.742	0.769	0.960	0.934	0.959	0.958
BAT	0.667	0.756	0.761	0.771	0.935	0.959	0.935	0.963
HGS	0.672	0.764	0.742	0.773	0.944	0.935	0.945	0.963
TSO	0.665	0.785	0.725	0.766	0.961	0.934	0.961	0.958
Aqu	0.676	0.777	0.751	0.783	0.972	0.973	0.971	0.974

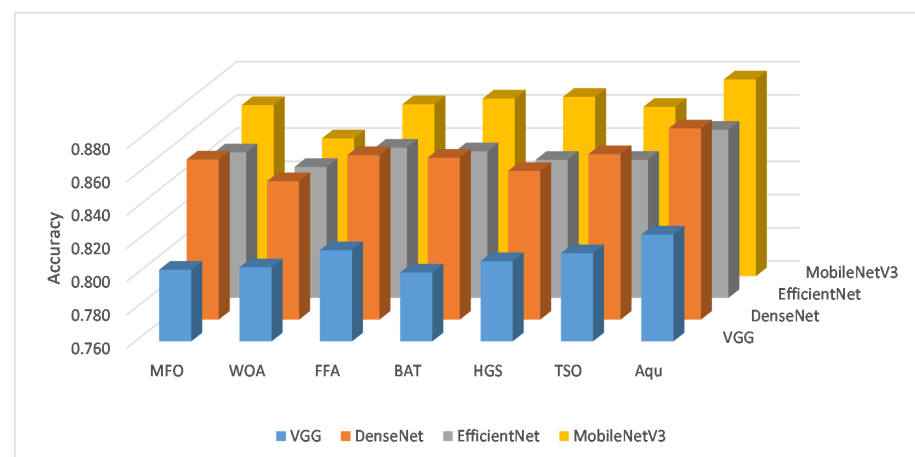


Figure 7. Average of each CNN type overall the FS methods.

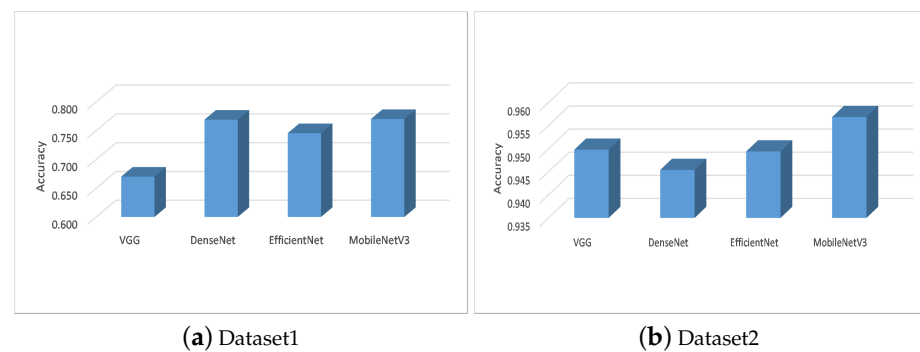


Figure 8. Average of each CNN type overall the FS methods.

4.5. Influence of the Size of COVID19 Dataset

In this section, the influence of using a large number of images on the performance of the developed method is assessed using a third dataset (i.e., COVID-19 radiography) described in Section 4.1.

Table 7 shows the average of the results in terms of performance measures for each FS algorithm that depends on the features extracted using MobileNetV3. From these results, one can reach the following observations: firstly, Aqu has a high ability to enhance the classification accuracy; in addition, it can reduce the number of features required to increase the classification accuracy. However, the Aqu allocates the second rank after TSO in CPU time (s) required to determine the relevant features.

Table 7. Performance of FS methods using COVID-19 radiography dataset.

	Accuracy	Recall	Precision	F1-Score	CPU Time (s)	#FS
MFO	0.889	0.897	0.840	0.868	15.347	61
WOA	0.886	0.885	0.828	0.855	15.593	58.5
FFA	0.910	0.885	0.828	0.855	15.632	56
BAT	0.887	0.909	0.852	0.880	18.185	57
HGS	0.894	0.884	0.827	0.855	18.306	69
TSO	0.910	0.884	0.827	0.855	15.179	63
Aqu	0.924	0.924	0.866	0.894	15.290	57

5. Conclusions

This study developed a framework to detect the COVID-19 cases from X-ray and CT images using three datasets with a considerable amount of samples. The proposed framework depends on the combination of the MobileNetV3 DL model and metaheuristic (MH) techniques. Furthermore, three other DL networks were included in our experiments, namely VGG19, DenseNet, and EfficientNet. For instance, MobileNetV3 was used to extract the features from all existing images in each dataset. By contrast, a new MH technique named Aquila Optimizer (Aqu) was proposed for feature selection (FS) by converting it to binary. The extracted image embeddings from each DL network were fed to the FS algorithm for feature space reduction and classification performance improvement. To justify the performance of the developed method, three datasets are used with different characteristics since they represent X-ray and CT COVID-19 images collected from different sources. The comparison results illustrated the high performance of the developed method based on the Aqu method over the other competitive methods.

Besides the promising results, the developed method can be extended to other applications such as agriculture, remote sensing, galaxy classification, and other image classification tasks.

Author Contributions: M.A.E.: Conceptualization, supervision, methodology, formal analysis, resources, data curation, and writing—original draft preparation. A.D.: Conceptualization, supervision, methodology, formal analysis, resources, data curation, and writing—original draft preparation. A.H.E.: Conceptualization, supervision, methodology, formal analysis, resources, data curation, and writing—original draft preparation. A.I.S.: Supervision and writing—review and editing, methodology, formal analysis, resources, and data curation. M.A.: supervision and writing—review and editing, methodology, formal analysis, resources, and data curation. N.A.A.: supervision and writing—review and editing, methodology, formal analysis, resources, and data curation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data can be obtained from <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>, <https://www.medrxiv.org/>, <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed on 12 October 2021).

Acknowledgments: This research was supported by the Deanship of Scientific Research, Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia, Grant No. (21-13-18-032).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, D.; Xu, W.; Lei, Z.; Huang, Z.; Liu, J.; Gao, Z.; Peng, L. Recurrence of positive SARS-CoV-2 RNA in COVID-19: A case report. *Int. J. Infect. Dis.* **2020**, *93*, 297–299. [[CrossRef](#)] [[PubMed](#)]
- Liu, Y.C.; Kuo, R.L.; Shih, S.R. COVID-19: The first documented coronavirus pandemic in history. *Biomed. J.* **2020**, *43*, 328–333. [[CrossRef](#)] [[PubMed](#)]
- Huang, C.; Wang, Y.; Li, X.; Ren, L.; Zhao, J.; Hu, Y.; Zhang, L.; Fan, G.; Xu, J.; Gu, X.; et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* **2020**, *395*, 497–506. [[CrossRef](#)]
- Serena Low, W.C.; Chuah, J.H.; Tee, C.A.T.; Anis, S.; Shoaib, M.A.; Faisal, A.; Khalil, A.; Lai, K.W. An Overview of Deep Learning Techniques on Chest X-Ray and CT Scan Identification of COVID-19. *Comput. Math. Methods Med.* **2021**, *2021*, 5528144. [[CrossRef](#)] [[PubMed](#)]
- Brady, A.P. Error and discrepancy in radiology: Inevitable or avoidable? *Insights Imaging* **2017**, *8*, 171–182. [[CrossRef](#)]
- Briganti, G.; Le Moine, O. Artificial intelligence in medicine: Today and tomorrow. *Front. Med.* **2020**, *7*, 27. [[CrossRef](#)] [[PubMed](#)]
- Kumar, L.K.; Alphonse, P. Automatic Diagnosis of COVID-19 Disease using Deep Convolutional Neural Network with Multi-Feature Channel from Respiratory Sound Data: Cough, Voice, and Breath. *Alex. Eng. J.* **2021**. [[CrossRef](#)]
- Al-Qaness, M.A.; Saba, A.I.; Elsheikh, A.H.; Abd Elaziz, M.; Ibrahim, R.A.; Lu, S.; Hemedan, A.A.; Shanmugan, S.; Ewees, A.A. Efficient artificial intelligence forecasting models for COVID-19 outbreak in Russia and Brazil. *Process Saf. Environ. Prot.* **2021**, *149*, 399–409. [[CrossRef](#)] [[PubMed](#)]
- Göreke, V.; Sarı, V.; Kockanat, S. A novel classifier architecture based on deep neural network for COVID-19 detection using laboratory findings. *Appl. Soft Comput.* **2021**, *106*, 107329. [[CrossRef](#)] [[PubMed](#)]
- Elsheikh, A.H.; Saba, A.I.; Abd Elaziz, M.; Lu, S.; Shanmugan, S.; Muthuramalingam, T.; Kumar, R.; Mosleh, A.O.; Essa, F.; Shehabeldeen, T.A. Deep learning-based forecasting model for COVID-19 outbreak in Saudi Arabia. *Process Saf. Environ. Prot.* **2021**, *149*, 223–233. [[CrossRef](#)] [[PubMed](#)]
- Saba, A.I.; Elsheikh, A.H. Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks. *Process Saf. Environ. Prot.* **2020**, *141*, 1–8. [[CrossRef](#)]
- Mohammadi, F.; Pourzamani, H.; Karimi, H.; Mohammadi, M.; Mohammadi, M.; Ardalan, N.; Khoshravesh, R.; Pooresmaeil, H.; Shahabi, S.; Sabahi, M.; et al. Artificial neural network and logistic regression modelling to characterize COVID-19 infected patients in local areas of Iran. *Biomed. J.* **2021**. [[CrossRef](#)] [[PubMed](#)]
- Albahli, S. A deep neural network to distinguish covid-19 from other chest diseases using x-ray images. *Curr. Med. Imaging* **2021**, *17*, 109–119. [[CrossRef](#)]
- Zhao, W.; Zhong, Z.; Xie, X.; Yu, Q.; Liu, J. Relation between chest CT findings and clinical conditions of coronavirus disease (COVID-19) pneumonia: A multicenter study. *Am. J. Roentgenol.* **2020**, *214*, 1072–1077. [[CrossRef](#)] [[PubMed](#)]
- Bernheim, A.; Mei, X.; Huang, M.; Yang, Y.; Fayad, Z.A.; Zhang, N.; Diao, K.; Lin, B.; Zhu, X.; Li, K.; et al. Chest CT findings in coronavirus disease-19 (COVID-19): Relationship to duration of infection. *Radiology* **2020**, 200463. [[CrossRef](#)] [[PubMed](#)]
- Pezzano, G.; Diaz, O.; Ripoll, V.R.; Radeva, P. CoLe-CNN+: Context learning-Convolutional neural network for COVID-19-Ground-Glass-Opacities detection and segmentation. *Comput. Biol. Med.* **2021**, *136*, 104689. [[CrossRef](#)] [[PubMed](#)]
- Yasin, R.; Gouda, W. Chest X-ray findings monitoring COVID-19 disease course and severity. *Egypt. J. Radiol. Nucl. Med.* **2020**, *51*, 1–18. [[CrossRef](#)]
- Wang, L.; Lin, Z.Q.; Wong, A. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest X-ray images. *Sci. Rep.* **2020**, *10*, 1–12.

19. Das, A.K.; Ghosh, S.; Thunder, S.; Dutta, R.; Agarwal, S.; Chakrabarti, A. Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network. *Pattern Anal. Appl.* **2021**, 1–14. [[CrossRef](#)]
20. Oliva, D.; Abd Elaziz, M.; Elsheikh, A.H.; Ewees, A.A. A review on meta-heuristics methods for estimating parameters of solar cells. *J. Power Sources* **2019**, *435*, 126683. [[CrossRef](#)]
21. Abd Elaziz, M.; Elsheikh, A.H.; Oliva, D.; Abualigah, L.; Lu, S.; Ewees, A.A. Advanced Metaheuristic Techniques for Mechanical Design Problems. *Arch. Comput. Methods Eng.* **2021**, 1–22. [[CrossRef](#)]
22. Canayaz, M. MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images. *Biomed. Signal Process. Control* **2021**, *64*, 102257. [[CrossRef](#)]
23. Sahlol, A.T.; Yousri, D.; Ewees, A.A.; Al-Qaness, M.A.; Damasevicius, R.; Abd Elaziz, M. COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Sci. Rep.* **2020**, *10*, 1–15. [[CrossRef](#)] [[PubMed](#)]
24. Abd Elaziz, M.; Ewees, A.A.; Yousri, D.; Alwerfali, H.S.N.; Awad, Q.A.; Lu, S.; Al-Qaness, M.A. An improved Marine Predators algorithm with fuzzy entropy for multi-level thresholding: Real world example of COVID-19 CT image segmentation. *IEEE Access* **2020**, *8*, 125306–125330. [[CrossRef](#)] [[PubMed](#)]
25. Yousri, D.; Abd Elaziz, M.; Abualigah, L.; Oliva, D.; Al-Qaness, M.A.; Ewees, A.A. COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. *Appl. Soft Comput.* **2021**, *101*, 107052. [[CrossRef](#)] [[PubMed](#)]
26. Elaziz, M.A.; Hosny, K.M.; Salah, A.; Darwish, M.M.; Lu, S.; Sahlol, A.T. New machine learning method for image-based diagnosis of COVID-19. *PLoS ONE* **2020**, *15*, e0235187. [[CrossRef](#)]
27. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-qanes, M.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization Algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [[CrossRef](#)]
28. AlRassas, A.M.; Al-qaness, M.A.; Ewees, A.A.; Ren, S.; Abd Elaziz, M.; Damaševičius, R.; Krilavičius, T. Optimized ANFIS model using Aquila Optimizer for oil production forecasting. *Processes* **2021**, *9*, 1194. [[CrossRef](#)]
29. Wang, S.; Jia, H.; Liu, Q.; Zheng, R. An improved hybrid Aquila Optimizer and Harris Hawks Optimization for global optimization. *Math. Biosci. Eng.* **2021**, *18*, 7076–7109. [[CrossRef](#)]
30. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8697–8710.
31. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 2 September–27 October 2019; pp. 1314–1324.
33. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
34. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2820–2828.
35. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
36. Tran, D.; Wang, H.; Torresani, L.; Feiszli, M. Video classification with channel-separated convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 5552–5561.
37. Ji, J.; Krishna, R.; Fei-Fei, L.; Niebles, J.C. Action genome: Actions as compositions of spatio-temporal scene graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10236–10247.
38. Liu, J.; Inkawhich, N.; Nina, O.; Timofte, R. NTIRE 2021 multi-modal aerial view object classification challenge. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 588–595.
39. Ignatov, A.; Romero, A.; Kim, H.; Timofte, R. Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: Report. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2535–2544.
40. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Yang, X.; He, X.; Zhao, J.; Zhang, Y.; Zhang, S.; Xie, P. COVID-CT-dataset: A CT scan dataset about COVID-19. *arXiv* **2020**, arXiv:2003.13865.
43. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
44. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
45. Abd Elaziz, M.; Ewees, A.A.; Ibrahim, R.A.; Lu, S. Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Math. Comput. Simul.* **2020**, *168*, 48–75. [[CrossRef](#)]

46. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36–50. [[CrossRef](#)]
47. Nakamura, R.Y.; Pereira, L.A.; Costa, K.A.; Rodrigues, D.; Papa, J.P.; Yang, X.S. BBA: A binary bat algorithm for feature selection. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012; pp. 291–297.
48. AbuShanab, W.S.; Abd Elaziz, M.; Ghandourah, E.I.; Moustafa, E.B.; Elsheikh, A.H. A new fine-tuned random vector functional link model using Hunger games search optimizer for modeling friction stir welding process of polymeric materials. *J. Mater. Res. Technol.* **2021**, *14*, 1482–1493. [[CrossRef](#)]
49. Qais, M.H.; Hasanien, H.M.; Alghuwainem, S. Transient search optimization: A new meta-heuristic optimization algorithm. *Appl. Intell.* **2020**, *50*, 3926–3941. [[CrossRef](#)]
50. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
51. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.