Applied Intelligence (2023) 53:18630–18652
https://doi.org/10.1007/s10489-022-04446-8

Feature selection of pre-trained shallow CNN using the QLESCA optimizer: COVID-19 detection as a case study

Qusay Shihab Hamad[1,3] · Hussein Samma[2] · Shahrel Azmin Suandi[1]

Accepted: 29 December 2022 / Published online: 6 February 2023

Abstract
According to the World Health Organization, millions of infections and a lot of deaths have been recorded worldwide since the emergence of the coronavirus disease (COVID-19). Since 2020, a lot of computer science researchers have used convolutional neural networks (CNNs) to develop interesting frameworks to detect this disease. However, poor feature extraction from the chest X-ray images and the high computational cost of the available models introduce difficulties for an accurate and fast COVID-19 detection framework. Moreover, poor feature extraction has caused the issue of 'the curse of dimensionality', which will negatively affect the performance of the model. Feature selection is typically considered as a preprocessing mechanism to find an optimal subset of features from a given set of all features in the data mining process. Thus, the major purpose of this study is to offer an accurate and efficient approach for extracting COVID-19 features from chest X-rays that is also less computationally expensive than earlier approaches. To achieve the specified goal, we design a mechanism for feature extraction based on shallow conventional neural network (SCNN) and used an effective method for selecting features by utilizing the newly developed optimization algorithm, Q-Learning Embedded Sine Cosine Algorithm (QLESCA). Support vector machines (SVMs) are used as a classifier. Five publicly available chest X-ray image datasets, consisting of 4848 COVID-19 images and 8669 non-COVID-19 images, are used to train and evaluate the proposed model. The performance of the QLESCA is evaluated against nine recent optimization algorithms. The proposed method is able to achieve the highest accuracy of 97.8086% while reducing the number of features from 100 to 38. Experiments prove that the accuracy of the model improves with the usage of the QLESCA as the dimensionality reduction technique by selecting relevant features.

Keywords Features selection · Shallow convolutional neural networks · Swarm intelligence · SVM · Features extraction · Q-learning embedded sine cosine algorithm (QLESCA)

1 Introduction

Medical diagnosis is one of the more recent fields where computer science is very beneficial, as faster detection may effectively benefit from Computational Intelligence models [1, 2]. In addition, the fast spread of the COVID-19 outbreak has boosted the demand for specialized knowledge and stoked interest in the development of automated detection systems that rely on artificial intelligence (AI) techniques [3–5]. AI techniques can aid in obtaining reliable findings and are beneficial for removing obstacles such as the absence of readily available real-time test kits and a waiting period for test results. Scans of chest X-rays and chest computed tomography (CT) reveal the chief signs of COVID-19, even when the symptoms are minor [6–9]. This information can be used to circumvent the limitations of certain technologies, such as the lack of diagnostic kits. Even if CT scans are superior, X-rays are still useful since they are less costly, quicker, and more

✉ Shahrel Azmin Suandi
shahrel@usm.my

Qusay Shihab Hamad
qusay@student.usm.my; engitq@yahoo.com

Hussein Samma
hussein.samma@yahoo.com

[1] Intelligent Biometric Group, School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

[2] SDAIA-KFUPM Joint Research Center for Artificial Intelligence (JRC-AI), King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

[3] University of Information Technology and Communications (UOITC), Baghdad, Iraq

generally utilized. Even in rural locations, imaging technologies are available at the majority of health clinics and laboratories that utilize X-ray pictures. In the absence of typical symptoms such as fever, chest X-rays have a reasonable ability to identify the disease [10–12].

The feature selection process is critical for improving the model's evaluation results. Due to the high dimensionality of the data, certain undesirable situations occur in the applied model when the number of features is increased. These include: increased training time, and model overfitting [13]. There are three primary benefits of performing feature selection: decreased training time (fewer features means that the algorithm can learn more quickly), increased accuracy (less misleading data means that the model is more efficient), and less over-fitting (higher probability of successful classification) [14].

The extraction of features is a crucial stage in classification, as the extracted features provide useful image information. Deep Neural Networks are extraordinarily capable of extracting the essential elements from a vast dataset for image feature extraction. Consequently, these are often utilized in computer vision algorithms [15]. Unlike standard machine learning algorithms, which rely on manually extracted features, CNNs can learn and represent complicated characteristics automatically. The main goal at this point is to figure out how to make sense of the raw data, which will help improve the overall accuracy of the model [16]. By selecting effective features through the feature selection process, such undesirable situations can be avoided.

The capacity to collect vast quantities of data in the modern day is a double-edged sword. On the one hand, it allows for a more rigorous analysis of characteristics, but on the other, it becomes increasingly difficult to store and interpret such enormous volumes of data. Thus, the importance of dimensionality reduction strategies increases since they eliminate unnecessary characteristics without harming the learner's performance. Using an exhaustive search to pick feature subsets is an NP-hard task. Consequently, a number of intelligent optimization frameworks have been presented in the literature in order to choose an optimal subset at significantly reduced computational expense. The use of metaheuristics [13] to solve optimization issues has grown in popularity due to its capacity to efficiently search for a global optimum. Tabularized in Table 1 is a detailed literature overview of the prevalent metaheuristics employed in the feature selection field.

Every species employs a unique hunting and/or migratory strategy. Over millions of years, organisms have evolved into their current form, where their feeding habits are ideal, resulting in a stable population. Consequently, the study of their actions in nature has led to the development of optimization algorithms that attempt to quantitatively imitate their strategies. Utilizing much fewer resources, the effectiveness of such formulations in a broad variety of issues has led to their prominence as a study subject, where innovative techniques have been developed recently. In several fields, conventional optimization methods such as the Genetic Algorithm (GA) [17] and Particle Swarm Optimization (PSO) [18] have been widely implemented.

Nevertheless, the No Free Lunch Theorem [19] makes it clear that these traditional algorithms cannot offer the best answers to all the optimization issues that researchers must deal with. Thus, many metaheuristic optimization algorithms have been successfully implemented in a wide variety of applications, and a large number of researchers have proposed numerous algorithms, including monarch butterfly optimization (MBO) [20], the slime mould algorithm (SMA) [21], moth search algorithm (MSA) [22], hunger games search (HGS) [23], Runge Kutta method (RUN) [24], colony predation algorithm (CPA) [25], and Harris hawks optimization (HHO) [26].

The selected features that were selected via each agent of the optimization algorithm are passed to the fitness function, where a fitness value is obtained for each agent. The algorithm seeks features that provide a better value than the fitted value in each iteration. At the end of the algorithm's running time, the features with the highest fitness values are chosen in the evaluation process of the model [13].

However, numerous methods for COVID-19 detection have been proposed, and existing chest X-ray-based methods for COVID-19 diagnosis have three major limitations. Firstly, the complex strategy used to extract and select the features requires more computing resources, which is a demanding task. Secondly, existing deep learning-based methods require a higher number of training parameters, which not only leads to a computation burden in the classification but also leads to over-fitting problems because of the limited availability of COVID-19 images. Finally, due to the huge number of model parameters and features extracted, it makes it difficult to install on a device with limited resources (processing unit, memory, etc.). Because of this, we came up with a simple and fast way to find COVID-19 using shallow CNNs and a swarm optimization algorithm.

Due to the high dimension of the features, we require an effective optimization algorithm with the capacity to handle high dimensions. The Q-Learning Embedded Sine Cosine Algorithm (QLESCA) is one of the most recent optimization algorithms proposed to deal with high-dimensional problems; thus, the QLESCA is used for feature selection in this study. Our team developed the QLESCA [44], which makes the classical SCA better so it does not get stuck in local optima and gives the algorithm more freedom to switch between exploration and exploitation phases based on the search cases.

The aim of our work is to provide a computer-based diagnostic model for COVID-19 that takes chest X-ray images as the input. We initially tried to detect COVID-19 by utilizing a

Table 1 Set of metaheuristics algorithms employed in the feature selection field

No.	Reference	Optimization algorithm	Year
1.	[27]	Binary butterfly optimization approaches	2019
2.	[28]	binary grey wolf algorithm	2020
3.	[29]	Improved Binary Grey Wolf Optimizer	2020
4.	[30]	Enhanced Crow Search Algorithm	2020
5.	[31]	An efficient binary social spider algorithm	2020
6.	[32]	multi population-based particle swarm optimization	2021
7.	[33]	Dynamic Salp swarm algorithm	2021
8.	[34]	binary butterfly optimization algorithm	2021
9.	[35]	genetic algorithms, particle swarm optimization, and artificial bee colony	2022
10.	[36]	grey wolf optimization algorithm	2022
11.	[37]	Quantum Whale Optimization Algorithm	2022
12.	[38]	butterfly optimization algorithm	2022
13.	[39]	group teaching optimization algorithm	2022
14.	[40]	dynamic stochastic search algorithm	2022
15.	[41]	Dynamic Butterfly algorithm	2022
16.	[42]	Binary Harris Hawks Optimization	2022
17.	[43]	Enhanced Gaussian bare-bones grasshopper optimization	2023

pre-trained convolutional neural network (CNN) model called VGG19. Though it provides satisfactory results, there are certain drawbacks to using DL architecture. The use of DL models involves a huge computational overhead and requires an enormous number of training samples and a large amount of training data that are not available for new pandemics like COVID-19. We therefore used a shallow CNN (SCNN) model instead of the full VGG19 to extract dominant features from the data before further processing. However, some of these features may be redundant and unessential. The features which we obtained were optimized using several feature optimization algorithms. For further dimensional reduction of the feature space, we used our novel optimization algorithm, the QLESCA. The highlights of this work are:

1. Combination of pre-trained CNNs and a metaheuristic-based feature selection algorithm to produce a CAD system to detect COVID-19 in chest X-ray images.
2. A new lightweight model (SCNN-SVM) was proposed for COVID-19 diagnosis that can overcome the overfitting caused by deep neural network training on a small dataset.
3. To reduce the dimensionality of the feature set obtained from the SCNN-SVM model, an improved version of the SCA, called the QLESCA, has been used due to the superior ability of the QLESCA to solve high-dimensional problems.
4. So far as we know, this is the first study to use a shallow CNN-based feature extractor to pull important features out of datasets.
5. A huge number of COVID-19 samples with 4848 images from five public datasets (two of them published in 2022) and detailed comparisons with nine optimization algorithms are used to test the quality of the new approach.
6. This research developed a fitness equation that strikes a balance between the number of features chosen and how accurate the model is.

The following is the organization of this paper. The related works are clarified in Section 2. The theoretical preliminaries needed in the proposed model are clarified in Section 3. Section 4 presents the proposed framework. Experimental results are clarified in Section 5. Section 6 discusses the conclusion of the research findings and some future work.

2 Related work

In recent years, the application of artificial intelligence in the automatic diagnosis of medical images has yielded promising results. Deep convolutional neural networks have been used in previous studies to classify chest X-ray and CT images and to successfully diagnose common chest diseases [1, 45–47]. In the current outbreak of COVID-19, researchers are attempting to alleviate the epidemic through their research. Numerous researchers are conducting studies in this area. In general, the literature on COVID-19 diagnoses can be divided into two categories based on: 1) different artificial intelligence techniques, and 2) metaheuristic optimization algorithms.

2.1 COVID-19 diagnoses using different artificial intelligence techniques

This section describes research being done to combat COVID-19 using a variety of techniques. Abdullah et al. [48] utilized K-means clustering in their fight against COVID-19. This was accomplished by determining a province's proximity or similarity based on confirmed cases, recovered cases, and death cases. The case study was Indonesia. Abir et al. [49] proposed a new method to detect presymptomatic COVID-19 infection using the resting heart rate derived from wearable devices, such as smartwatches or fitness trackers. Based on the reported findings, the diagnostic tool's viability as an early-stage COVID-19 detection method was demonstrated. Ahmad et al. [50] proposed a new CNN model for COVID-19 patient identification. The proposed CNN consists of three convolution layers (16, 128, and 256 filters), two max-pooling layers, a flattening layer, and three fully connected layers (FCLs) (120, 60, and 10 neurons). Due to the fact that a small dataset was used to train and evaluate their model, image augmentation techniques have been implemented to address the issue of data scarcity. For two classes (COVID-19 and non-COVID-19), the proposed model achieves a 97.68% accuracy rate. Akter et al. [51] developed an innovative deep learning model called AD-CovNet for predicting the mortality of Alzheimer's patients infected with COVID-19. There were three datasets used to evaluate the proposed model. This proposed model had an accuracy of 97% for datasets 1 and 2 and 86% for dataset 3. Polsinelli et al. [52] proposed a light CNN for detecting COVID-19 from computer tomography (CT) imaging of the chest. They used Bayesian optimization to select the optimal hyperparameters for SqueezeNet. They selected three hyperparameters in their work: initial learning rate, momentum, and L2-Regularization. The proposed model achieved an accuracy of 85.03%.

Ahuja et al. [53] analyzed the performance of four pre-trained architectures for COVID-19 detection in CT scan images; these models included ResNet18, ResNet50, ResNet101, and SqueezeNet. The reported result confirms that the ResNet18 model offered superior classification accuracy compared to competing models. COVIDetectioNet is a diagnosis system proposed by Turkoglu [54]; this model comprises three fundamental stages: feature extraction, feature selection, and classification. In the initial phase, convolution and FCLs of the AlexNet architecture were used to extract deep features. In the second stage, the Relief algorithm was used to select the most efficient features. In the third phase, a SVM was used to classify these exceptional features. Using samples of chest X-ray scans, Jain et al. [55] compared three pre-trained CNN models for COVID-19 detection. These were the Inception V3, Xception, and ResNeXt models. According to reported results, the Xception model provides the highest detection accuracy. Punn and Agarwal [56] investigated five CNN models for COVID-19 detection from chest X-ray images. ResNet, Inception-v3, Inception ResNet-v4, DenseNet169, and NASNetLarge were the respective models. Both binary classification (normal and COVID-19) and multi-class classification (COVID-19, pneumonia, and normal) were investigated. In comparison to other models, NASNetLarge demonstrated superior performance based on the reported results. Mukherjee et al. [57] proposed a new CNN model with the ability to train and test using CT and chest X-ray images collectively. This model achieved an accuracy of 96.2% overall. Li et al. [58] proposed a COVID-19 detection model using CT images from a small dataset. A stacked autoencoder detector model was proposed to significantly improve the detection model's performance. The results of the experiment showed that the proposed model worked well with a small dataset and an average accuracy of 94.7%.

2.2 COVID-19 detection using metaheuristic algorithms

This section describes research utilizing metaheuristic optimization algorithms for feature selection to combat COVID-19. Two types of approaches are presented: the first approach evaluates the performance of an optimization algorithm for selecting efficient features, while the second presents a model for extracting features and using an optimization algorithm to select a subset of extracted features. This second approach is thought to be a new area of research, and it is important in this manuscript because it shows the proposed architecture.

2.2.1 Handcrafted features

Too et al. [59] presented an enhancement to the Binary Dragonfly Algorithm (BDA) for feature selection. The proposed algorithm is called the Hyper Learning Binary Dragonfly Algorithm (HLBDA), which enhances the BDA by using a hyper learning strategy. The proposed algorithm was evaluated using datasets from the University of California, Irvine (UCI) repository and Arizona State University. Also, the proposed method was applied to the COVID-19 dataset. The COVID-19 dataset comprised 15 features. The HLBDA aims to forecast death and recovery conditions. Piri et al. [60] used a discrete gorilla troop optimizer (DAGTO) for medical sector feature selection. Based on the number and type of objective functions for identifying relevant features, four variants of the proposed method were proposed in this study: (1) single objective, (2) bi-objective (wrapper), (3) bi-objective (filter wrapper hybrid), and (4) tri-objective (filter wrapper hybrid). To evaluate these four variants, ten medical datasets were utilized. The integration of both filter and wrapper approaches yielded the smallest number of features and the highest recognition accuracy, according to the evaluation results. Additionally, to demonstrate the superiority of the proposed method, the COVID-19 dataset was utilized to predict the health of COVID-19 patients. An enhancement to the whale optimization algorithm (WOA) was proposed by Shahraki et al. [61] to tackle the feature selection problem. The performance of the proposed algorithm was benchmarked on the COVID-19 dataset based on the reported results. The results showed that the E-WOA can find the optimum subset of features in the COVID-19 dataset. The Simulated Annealing (SA) and Generalized Normal Distribution Optimizer (GNDO) were combined to create the Binary Simulated Normal Distribution Optimizer (BSNDO) by Ahmed et al. [62]. The BSNDO algorithm was tested using 18 well-known UCI datasets. Additionally, it was utilized for feature selection for classification purposes on a COVID-19 dataset. According to the results that were reported, the suggested method worked well as a feature selection method.

2.2.2 Features extracted based on CNNs

In the literature, there is a collection of research efforts aimed at resolving the problem of selecting features extracted via CNNs using metaheuristic optimization algorithms. Another motivation for this paper was the scarcity of research on feature extraction using CNN models, especially regarding the COVID-19 problem in the literature. To address this, a variety of studies are presented in this section to address a variety of issues, but all of them share a common thread: they all use optimization algorithms to select features extracted via CNNs. In a lot of studies, different metaheuristic optimization techniques are used. The point at which the features of CNNs were taken from each study is different.

Canayaz [13] proposed a model for COVID-19 detection in chest X-ray images. This approach utilized a dataset comprising three classes: COVID-19, normal, and pneumonia, each of which contained 364 images. Four deep learning models were used to extract features from the dataset: AlexNet, VGG19, GoogleNet, and ResNet. Then, two metaheuristic algorithms, binary PSO (BPSO) and binary GWO (BGWO) were used to select the best possible features. Finally, they were classified using a SVM after combining the features obtained during the feature selection phase. The proposed approach achieved an overall accuracy of 99.38% based on the reported results. Fatani et al. [16] proposed a model for an intrusion detection system for the Internet of Things. They developed a CNN-based feature extraction mechanism. Following that, they presented an alternative approach to feature selection utilizing the Aquila optimizer (AQU). Four publicly available benchmark datasets were used to evaluate the developed approach: BoT-IoT, NSL-KDD, CIC2017, and KDD99. They use a light feature extraction approach based on a CNN to extract features from the datasets used in this study. The proposed CNN consists of two convolutional layers, two pooling layers, and four FCLs. Each Conv has 64 filters and the kernel size is three. Max-pooling is used in the first Conv, while average-pooling is used in the second. FCL1, FCL2, and FCL3 are fully connected layers with a total of 128, 128, and 64 neurons. FCL1, FCL2, and FCL3 serve as feature extraction layers that output the learned features from the raw input, while FCL4 serves as the final FCL that outputs the classification predictions. The K-nearest neighbour (KNN) classifier is then used to determine the efficiency of the determined feature. The results demonstrate that the proposed approach possesses superior performance.

Abd Elaziz et al. [63] have developed a framework for detecting COVID-19 cases using X-ray and CT images. This model was evaluated using three datasets. When MobileNetV3 was used to extract features, they replaced the top two output layers of the MobileNetV3 model used for image classification with a 1 × 1 point-wise convolution of size 128. The AQU optimizer was utilized to select a subset of extracted features. Sahlol et al. [14] proposed a classification scheme for white blood cell leukaemia based on the statistically enhanced Salp Swarm Algorithm (SESSA). They extracted features using VGG19. Given that the input image has the shape (224, 224, 3), the final layer produced by VGG19 has the shape (7, 7, 512). This indicates that VGG19 returns a feature vector containing 7 × 7 × 512 = 25,088 features. The SESSA then works to eliminate unnecessary and noisy features. SESSA optimization selected only 1087 features out of 25,088 extracted with VGG19, while simultaneously improving in accuracy. The determined features were fed into SVM for classification.

A two-stage pipeline composed of feature extraction followed by feature selection (FS) for the detection of COVID-19 from CT scan images has been proposed by Bandyopadhyay et al. [64]. For feature extraction, the DenseNet201 architecture was utilized. In this research, they decided to extract the features from a global pooling layer which produces a 1D vector of dimension 1920. This vector represents the extracted features which are used in the next stage of FS. To eliminate the non-informative and redundant features, they proposed a hybrid Harris hawks optimization (HHO) algorithm combined with Simulated Annealing (SA), and chaotic initialization was employed. This hybrid algorithm is called the CHHO+SA. A KNN classifier has been employed for calculating the accuracy of the proposed method. Based on the reported results, the proposed algorithm was able to reduce the number of features chosen by about 75%.

The aforementioned papers are summarized in Table 2, which is included below. This table details the number of features used and the CNN part from which they were extracted.

3 Theoretical preliminaries

The purpose of this section is to expose readers to some fundamental principles that will serve as a foundation for the rest of the article.

Table 2 Brief description of the metaheuristic optimization algorithms used to select features extracted via CNN

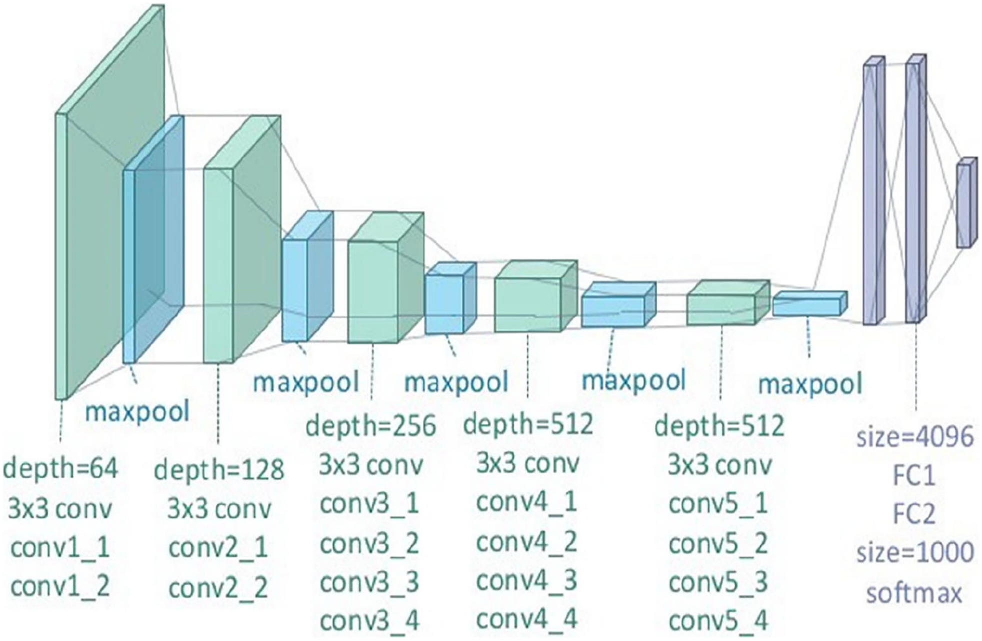
Reference	Optimizer	CNN	Number of features	Features were obtained from	Classifier type
[13]	BPSO	Alexnet VGG19	1000	FC8 FC8	SVM
		GoogleNet		loss3-classifier	
		ResNet		FC1000	
[13]	BGWO	Alexnet VGG19	1000	FC8 FC8	SVM
		GoogleNet		loss3-classifier	
		ResNet		FC1000	
[16]	AQU	CNN from Scratch	64	FC3	KNN
[63]	AQU	MobileNetV3	128	1×1 point-wise convolution	KNN
[14]	SESSA	VGG19	25,088	The last convolutional layer at VGG19	SVM
[64]	CHHO+SA	DenseNet201	1920	global pooling layer	KNN

3.1 VGG19

VGG19 is a CNN model developed at the University of Oxford by K. Simonyan and A. Zisserman [65]. On more than 14 million image datasets comprising 1000 classes in ImageNet, the model achieved 92.7% top-5 test accuracy. This model outperformed AlexNet by employing 3 × 3 core size filters instead of large core size filters. In the first layer, the input size is 224 × 224. VGG19 consists of a 19-layer network with 16 convolutional layers, five max-pooling layers, and three FCLs for feature extraction: "FC6", "FC7", and "FC8". Each "FC6" and "FC7" consists of 4096 neurons, whereas "FC8" has only 1000 neurons [13, 66]. The basic architecture of the VGG19 model is illustrated in Fig. 1.

VGG19 was selected in this study to conduct our experiments in the process of detection of COVID-19 based on its superior performance over many pre-trained CNN models for COVID-19 detection based on the literature [67, 68]. Seven different CNN models (VGG19, DenseNet121, InceptionV3, ResNetV2, Inception-ResNet-V2, Xception, MobileNetV2) were used by Hemdan et al. [67] to detect COVID-19 from chest X-ray images. These models were validated on 50 X-ray images (25 Normal and 25 COVID-19 cases). The VGG19 and DenseNet201models achieved the best performance. Balaha et al. [68] developed a technique for COVID-19 detection in chest CT images. To optimize the CNN hyperparameters, they used HHO. They compared nine pre-trained CNNs in this study. These models were ResNet50, ResNet101, VGG16, VGG19, Xception, MobileNetV1, MobileNetV2, DenseNet121, and DenseNet169. HHO was tasked with the responsibility of determining the optimal hyperparameters. The dropout ratio, learning ratio, and batch size were all used as hyperparameters. VGG-19

Fig. 1 VGG19 architecture



reported the best value using the Stochastic Gradient Descent (SGD) parameters optimizer, 32 batch size, 56% dropout ratio, and 80% learning ratio.

3.2 Q-learning embedded sine cosine algorithm

A recent population-based optimizer is the QLESCA [44]. The QLESCA is an innovative variant of the Sine Cosine Algorithm (SCA) [69]. Because the conventional SCA has a number of flaws, including stagnation at local optimums, a slow convergence curve, and an inefficient balance between exploration and exploitation search modes, the QLESCA proposes intelligently controlling SCA parameters using an embedded Q-learning algorithm at runtime to mitigate these limitations. Each QLESCA agent possesses its own Q-table and evolves independently. The Q-table contains nine distinct states based on population density and distance from the leader of the population. Therefore, Q-table generates nine distinct actions (Action 1 to Action 9) to regulate QLESCA parameters, including r_1 and r_3. These QLESCA parameters are responsible for switching between adaptive exploration and exploitation and vice versa. A well-performing agent receives a reward for each action, while a poorly performing agent receives a penalty. This algorithm, like other population-based metaheuristics, starts with a set of random solutions, and then each solution updates its position based on the simple mathematical functions of sine and cosine as in Eq. (1).

$$X_i^t = \begin{cases} X\_best_i + r_1\, sin(r_2)(r_3 P - X\_best_i), & r_4 < 0.5 \\ X\_best_i + r_1\, cos(r_2)(r_3 P - X\_best_i), & r_4 \geq 0.5 \end{cases} \tag{1}$$

where X_best_i is the best position discovered by the current solution in the *i-th* dimension at the *t-th* iteration, | | indicates the absolute value, P_i^t represents the position of the best solution discovered among all solutions ever discovered by all agents and is called the destination solution, r_1 and r_3 are calculated through Eqs. (2) and (3), r_2 is a random variable in the range [0, 2π], and r_4 is a random number between [0, 1] and used to switch between sine and cosine equally.

The proposed QLESCA incorporates Q-learning into SCA. Basically, the Q-learning technique is embedded in order to control the values of the SCA parameters, namely r_1 and r_3. r_1 controls the amount of jump, and r_3 is responsible for the destination's contribution level (*P*).

Under the control of Q-learning, the r_1 variable will be given a random value that belongs to one of three scales, namely Low (from 0 to 0.666), Medium (from 0.667 to 1.332), and High (from 1.333 to 2). So, when r_1 is low, the SCA algorithm will be in the exploitation mode. On the other hand, it performs search exploration when r_1 is High. However, on a Medium scale, it will work in two scenarios. If the randomly generated value of r_1 (from 0.667 to 0.999) works in exploitation mode, while r_1 values in the range from 1 to 1.332, will work in the exploration phase.

For the r_3 parameter, it is also like r_1 being in the range from 0 to 2 with three intervals: Low (from 0 to 0.666), Medium (from 0.667 to 1.332), and High (from 1.333 to 2). The architecture of the Q-table has nine actions, including (r_1 = *L*, r_3 = *L*), (r_1 = *L*, r_3 = *M*), (r_1 = *L*, r_3 = *H*), (r_1 = *M*, r_3 = *L*),(r_1 = *M*, r_3 = *M*), (r_1 = *M*, r_3 = *H*), (r_1 = *H*, r_3 = *L*), (r_1 = *H*, r_3 = *M*), and (r_1 = *H*, r_3 = *H*).

$$r_1 = r_1\_min + (r_1\_max - r_1\_min)U(0,1) \tag{2}$$

$$r_3 = r_3\_min + (r_3\_max - r_3\_min)U(0,1) \tag{3}$$

Two indicators were used to measure population status and individual agent location with respect to destination (*P*). These indicators are population density and distance. It should be noted that the range of these indicators will be in the range of [0,1] and this is further categorized into three ranges, which are Low (from 0 to 0.333), Medium (from 0.334 to 0.666), and High (from 0.667 to 1). Table 3 shows the combination of various ranges with their respective action and state.

4 Proposed framework

This section describes the overall structure of the developed COVID-19 detection method. As shown in Fig. 2, the proposed method has three phases: feature extraction, feature selection, and testing. During the feature extraction phase, SCNN is used to pull out the features, which are then split into the training set and the testing set. In the feature selection phase, the QLESCA will work on the training set to determine the subset of features that has the highest fitness value. This phase is repeated until it reaches the maximum number of iterations, and then the best features detected pass to the testing phase. In the testing phase, the detected features are evaluated through a testing set. The flow chart of the proposed feature selection method is illustrated in Fig. 3.

4.1 Feature extraction phase

In this phase, a shallow CNN model is used for feature extraction. SCNN has been previously stated in our study [70], and

Table 3 Indicator of actions and states

Action State	1	2	3	4	5	6	7	8	9
Den	H	H	H	M	M	M	L	L	L
Dis	H	M	L	H	M	L	H	M	L

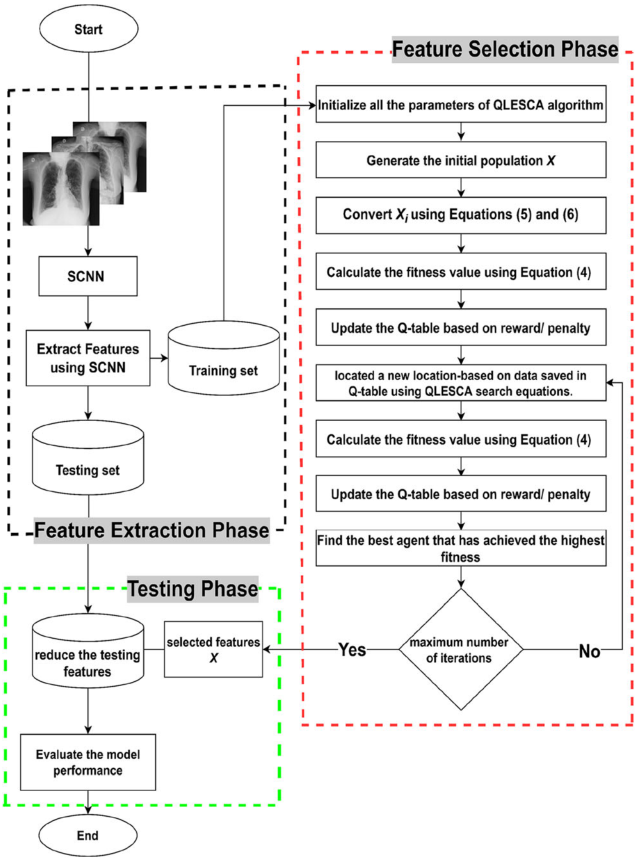

Start
Feature Selection Phase
Initialize all the parameters of QLESCA algorithm
Generate the initial population X
Convert X_i using Equations (5) and (6)
Calculate the fitness value using Equation (4)
Update the Q-table based on reward/ penalty
located a new location-based on data saved in Q-table using QLESCA search equations.
Calculate the fitness value using Equation (4)
Update the Q-table based on reward/ penalty
Find the best agent that has achieved the highest fitness
maximum number of iterations
Yes
No
SCNN
Extract Features using SCNN
Training set
Testing set
Feature Extraction Phase
Testing Phase
reduce the testing features
selected features X
Evaluate the model performance
End

Fig. 2 Steps of the QLESCA for the feature selection problem

the shallow CNN (SCNN) model had the highest accuracy compared with the deep CNN model. The SCNN model has the smallest number of layers (the first two VGG19 layers). It is computationally efficient because it has a limited number of layers, which means that it has a small number of parameters. As a result, when resources are limited, the shallow CNN architecture is a better fit. [71]. Also, as demonstrated in [72, 73], deeper models performed slightly better with larger datasets, but shallow CNNs performed better with smaller datasets. As a result, the SCNN model was chosen in this research in order to apply the QLESCA and thus improve its accuracy and overall performance. It is worth mentioning here that we used a pre-trained model of VGG19 trained on the ImageNet dataset to avoid training the model from scratch and speed up the learning process.

As can be seen in the original VGG19, it has FCLs followed by a SoftMax classifier that can classify 1000 different objects. Typically, the fully connected layer is utilized to further purify the features recovered by the convolutional layer, and so it plays a critical role in mapping the distributed feature to the sample space representation [74]. Here in the shallow architecture, to classify the features that were extracted via SCNN layers, a fully connected layer should be added. Since the output of the SCNN's final layer is a three-dimensional matrix, flattening it entails unrolling all of its values into a vector. Prior to submission to a fully connected layer, a flattened layer is required. Additionally, the majority of SCNN models in the literature use a SVM as the classifier due to its efficiency and minimal hyperparameter tuning requirements, which make the model less prone to overfitting and the experiment highly reproducible [75]. As a result, the proposed model includes a SCNN, FL, and FCL for feature extraction and SVM for classification, as can be seen in Fig. 4.

4.2 Feature selection phase

In this phase, QLESCA is used to select a subset of features from the training set. This section contains three parts: 1) the formulation of the fitness equation 2) the QLESCA for feature selection; and 3) the computational complexity of the proposed methodology. In the first part, the designed fitness equation is introduced, while the second part will clarify the steps of the feature selection algorithm, and the last part presents the complexity.

4.2.1 Formulation of fitness equation

In optimization algorithms, fitness evaluation is very important and reflects the objective of the problem that the optimization algorithm intends to solve. It is considered a vital component in order to design the correct fitness equation. So, in this subsection, we will give a simple explanation of the proposed fitness equation.

The QLESCA trains the SVM using a subset of features from the FCL and 20 images (10 COVID-19 and 10 normal). In other words, the QLESCA looks for two important items:

1. Significant features that aid the SVM to achieve higher classification accuracy. The QLESCA wants to use as few features as possible while still achieving the best possible classification accuracy.
2. Selecting 20 images (10 COVID-19 and 10 normal) to feed into the SCNN.

Figure 5 shows the concept of calculating the fitness equation in the proposed model. It depends on two factors: classification accuracy and the number of selected features. In other words, the best fitness is one that has high accuracy with a small number of selected features. Eq.)4(shows the fitness function used to evaluate each QLESCA search agent.

$$\text{Fitness, } F = \boldsymbol{K}*(\alpha*\boldsymbol{A}-\beta*(\boldsymbol{S}\_Features/\boldsymbol{T}\_Features)) \quad (4)$$

where $\boldsymbol{A}$ denotes accuracy and $\boldsymbol{S}_{\_Features}$ denotes the number of selected features. The parameters $\alpha \in [0,1]$ and $\beta = (1 - \alpha)$ correspond to the importance of classification accuracy and the number of selected features, respectively, on the fitness function [59, 76]. $\boldsymbol{T}_{\_Features}$ is the total number of features in the FCL (total number of features that are extracted via the SCNN).

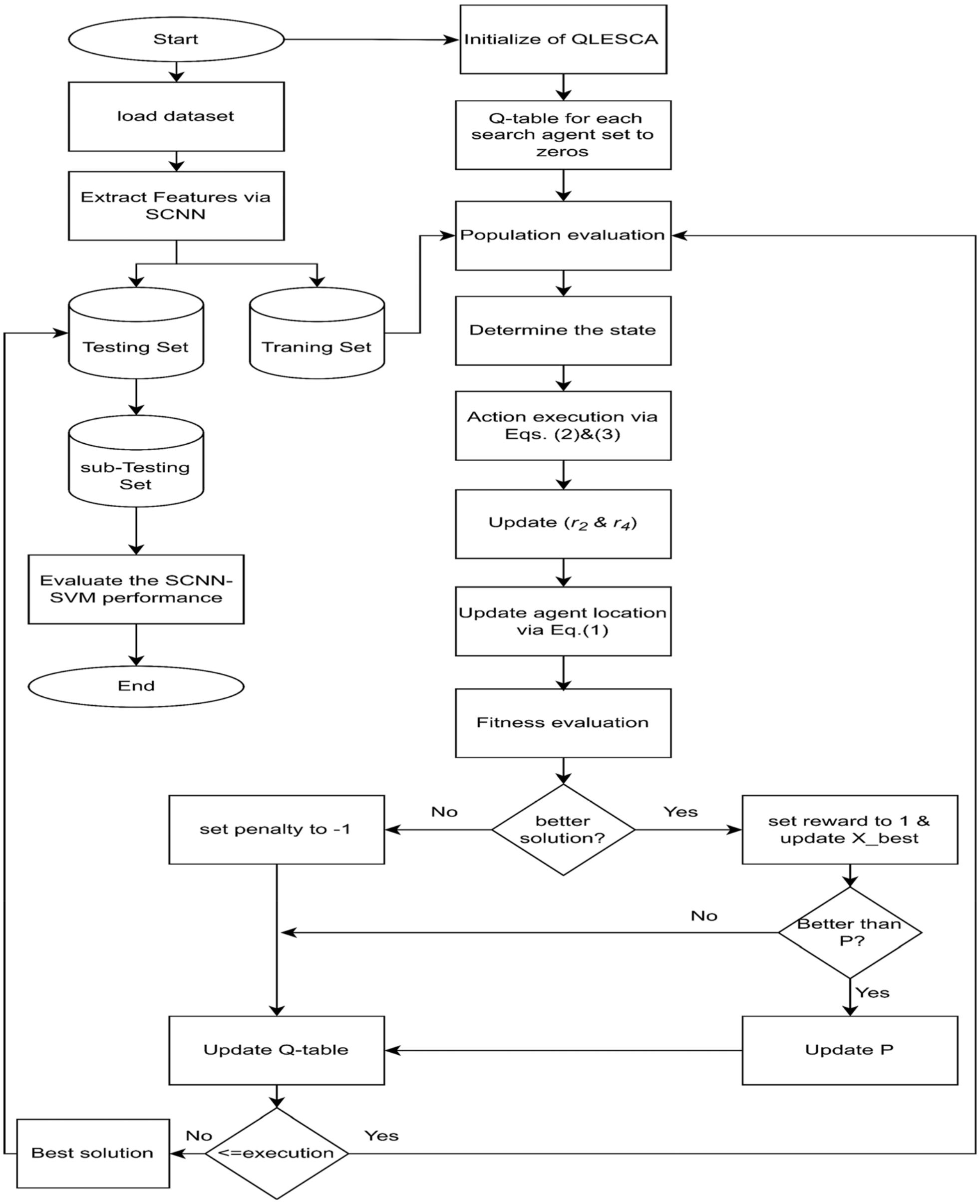


Fig. 3 Flowchart of the proposed method

Finally, the fitness result is multiplied by $K = -1$ to convert it to a negative value, as the optimization algorithm seeks the minimum value.

The position vector X of every search agent is a D-dimensional vector that represents the QLESCA search agent's position. As a consequence, each vector X denotes features extracted via the SCNN. In this study, the FCL is used as a feature extraction layer to output the learned features from the raw input, which contains a total of 100 neurons (100 features). Additionally, the QLESCA must choose only 20 images (10 COVID-19 and 10 normal). Each search agent has 120 variables (i.e., x_1 to x_{120}), which are listed in Table 4 with their respective ranges. Since each dimension of vector X has a different definition and a different range of values, Fig. 6 illustrates the vector X of every search agent.

For the first 100 variables, the QLESCA searches for a range from 0 to 1 because it works on generating random

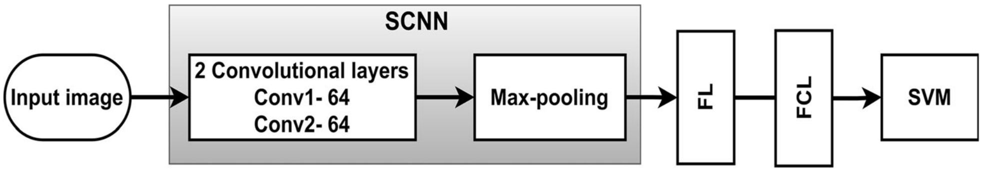


Fig. 4 Feature extraction using a SCNN

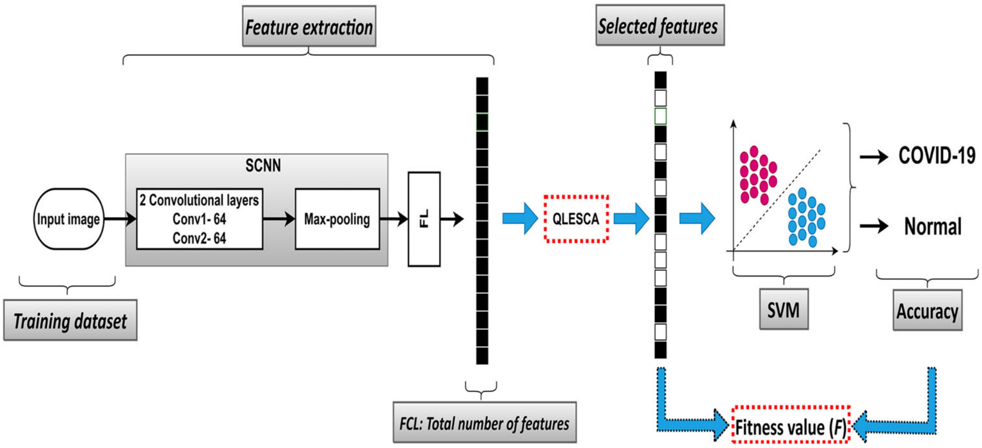


Fig. 5 Calculation of fitness evaluation in the proposed feature selection model

numbers, with each number generated if less than 0.5 shifted to 0 and otherwise shifted to 1, and this was produced using Eq. (5). For example, if x_{12} = 0, it means that feature number 12 is turned off, and if it equals 1, it is turned on. The same is true for the other values (from x_1 to x_{100}). For the second part (from x_{101} to x_{120}), the number is rounded to the nearest integer number, and this was produced using Eq. (6). For example, if x_{113} = 98.7, it means that the selected image is 99. The same is true for the other values in this range.

$$X_i = \begin{cases} 1 & if\ X_i \geq 0.5 \\ 0 & otherwise \end{cases} \qquad i = \{1, \ldots, 100\} \tag{5}$$

$$X_i = round(X_i) \quad i = \{101, \ldots, 120\} \tag{6}$$

Table 4 The encoding of vector X for every QLESCA search agent

Variables	Hyperparameters	Search range
X_1	Feature number 1 in the fully connected layer	[0,1]
X_2	Feature number 2 in the fully connected layer	[0,1]
X_3	Feature number 3 in the fully connected layer	[0,1]
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
X_{98}	Feature number 98 in the fully connected layer	[0,1]
X_{99}	Feature number 99 in the fully connected layer	[0,1]
X_{100}	Feature number 100 in the fully connected layer	[0,1]
X_{101}	COVID-19 image number	[0,403]
X_{102}	COVID-19 image number	[0, 403]
.	.	.
.	.	.
.	.	.
X_{109}	COVID-19 image number	[0, 403]
X_{110}	COVID-19 image number	[0, 403]
X_{111}	Normal image number	[0, 1108]
X_{112}	Normal image number	[0, 1108]
.	.	.
.	.	.
.	.	.
X_{119}	Normal image number	[0, 1108]
X_{120}	Normal image number	[0, 1108]

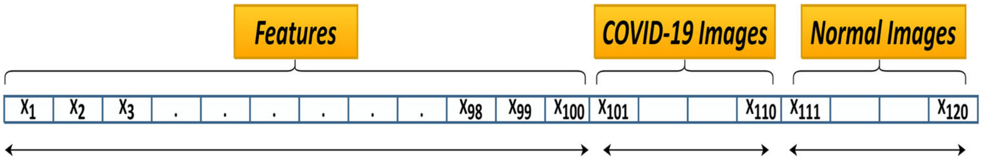


Fig. 6 Structure of vector *X* for every QLESCA search agent

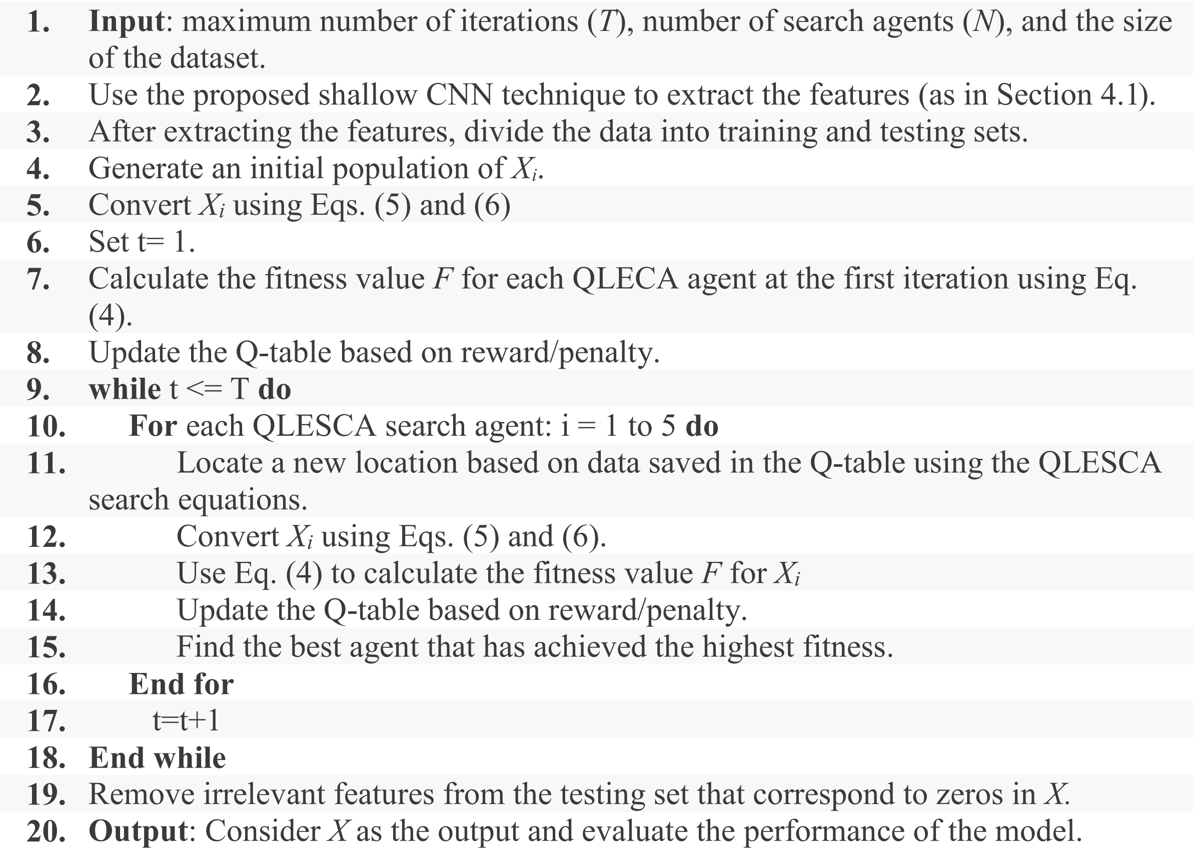
4.2.2 QLESCA steps for feature selection

The principal steps of feature selection are outlined in Algorithm 1, which provides a detailed description of the method presented for selecting features. As shown, the QLESCA is initialized by setting the maximum number of iterations (T), the number of search agents (N), and the dataset used by this model. Then the QLESCA operates on the training set by generating the first random X, converting it by using Eqs. (5) and (6) and calculating the fitness using Eq. (4). The Q-table is updated based on the fitness results for each agent, and the update process is based on reward and penalty. The search agent will receive a reward if its current result is superior to the previous result and a penalty if its result is weaker compared to the previous result. This process is repeated until the maximum number of iterations is reached. At that point, the algorithm ends, and the best values in X represent the best-selected features that are sent to the testing phase to be tested on the testing set.

4.2.3 Computational complexity of the proposed methodology

The computational complexity of the QLESCA-SCNN-SVM approach is estimated as a performance indicator that mainly depends on three steps: (1) position update using Q-Learning, (2) selecting the best features using the QLESCA, and (3) SVM classifier training time. Therefore, the complexity can be mathematically represented as $O(C_{QL} + C_{QLESCA-SCNN} + C_{SVM})$, where O denotes the worst-case time complexity, and C_{QL}, $C_{QLESCA\text{-}SCNN}$, and C_{SVM} indicate the complexity of Q-Learning implementation while modifying the location of each QLESCA search agent, the QLESCA-SCNN feature selection method, and the execution time of the SVM classifier in the training phase, respectively. Determining the computational complexity of many metaheuristic algorithms typically involves the analyses of three components [40]:

Algorithm 1 Proposed feature selection for COVID-19 detection



```
Input: maximum number of iterations ($T$), number of search agents ($N$), and the size of the dataset.
Use the proposed shallow CNN technique to extract the features (as in Section 4.1).
After extracting the features, divide the data into training and testing sets.
Generate an initial population of $X_i$.
Convert $X_i$ using Eqs. (5) and (6)
Set t= 1.
Calculate the fitness value $F$ for each QLECA agent at the first iteration using Eq. (4).
Update the Q-table based on reward/penalty.
while t <= T do
    For each QLESCA search agent: i = 1 to 5 do
        Locate a new location based on data saved in the Q-table using the QLESCA search equations.
        Convert $X_i$ using Eqs. (5) and (6).
        Use Eq. (4) to calculate the fitness value $F$ for $X_i$
        Update the Q-table based on reward/penalty.
        Find the best agent that has achieved the highest fitness.
    End for
    t=t+1
End while
Remove irrelevant features from the testing set that correspond to zeros in $X$.
Output: Consider $X$ as the output and evaluate the performance of the model.
```

The computational complexity of initializing the population is bounded by $O(C_{QL}.N.D)$.

The computational complexity of evaluating the fitness values of the initial population is bounded by $O(N.C_{QLESCA}-SCNN.C_{SVM})$.

- The computational complexity of the main loop is bounded by

$$O(C_{QL}.N.T_{max}.D + N.T_{max}.C_{QLESCA-SCNN}.C_{SVM}).$$

Here, the overall complexity is represented in terms of the number of iterations (T_{max}), population size (N), feature size (D), and training time of the classifier (C_{SVM}). To specify the total computational complexity for the proposed approach by considering only the most complex terms, the computational complexity of the main loop can be considered as the total computational complexity.

4.3 Testing phase

In this phase, as can be seen in Fig. 7, the best selected features (X) that are discovered during the feature extraction phase are used to evaluate the testing set, and the features of the testing set are reduced based on the selected features (X). Then several performance measures are employed to compute the quality of the proposed approach to COVID-19 detection. All testing results are presented in the experimental results section.

5 Experimental results

This section evaluates the quality of the proposed feature extraction method for COVID-19 detection.

5.1 Dataset description

In this study, to train, test, and compare our proposed model to others, a dataset of 2159 chest X-ray images [77] were used. The dataset was divided into two groups: 576 COVID-19 samples and 1583 healthy (Normal) samples. This is referred to as Dataset 1. To conduct the evaluation, Dataset 1 was randomly divided into 70% for training and 30% for testing, as indicated in Table 5. After applying the proposed metaheuristic optimization-CNN model to Dataset 1 and determining which CNN architecture is the best, this discovered CNN model was tested on another four datasets referred to as Dataset 2, Dataset 3, Dataset 4, and Dataset 5 in order to determine the strength and stability of the proposed model. Dataset 2 contained 2183 images, 600 of which were COVID-19 [78] and 1583 were Normal [79]. Dataset 3 [80] contained 4551 images, 1281 of which were COVID-19 and 3270 were Normal. Dataset 4 [81] is a recent dataset published on 15 June 2022, containing 3428 images, 1626 of which are COVID-19 and 1802 are Normal. Dataset 5 [82] is a recent dataset published on 3 October 2022, containing 1196 images, 765 of which are COVID-19 and 431 are Normal. Figure 8 displays a few COVID-19 samples as well as healthy chest X-ray images.

5.2 Evaluation metrics

The specifications of the machine utilized in this study are listed in Table 6 below. Accuracy, precision, specificity, sensitivity (Recall), and F1-score were the five metrics used to evaluate models [87, 88]. These metrics were calculated using Eqs. (7) to (11).

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \times 100\% \quad (7)$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \times 100\% \quad (8)$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \times 100\% \quad (9)$$

$$\text{Sensitivity (Recall)} = \text{TP}/(\text{TP} + \text{FN}) \times 100\% \quad (10)$$

$$\text{F1 Score} = 2\text{TP}/(2\text{TP} + \text{FP} + \text{FN}) \times 100\% \quad (11)$$

Fig. 7 The testing Phase

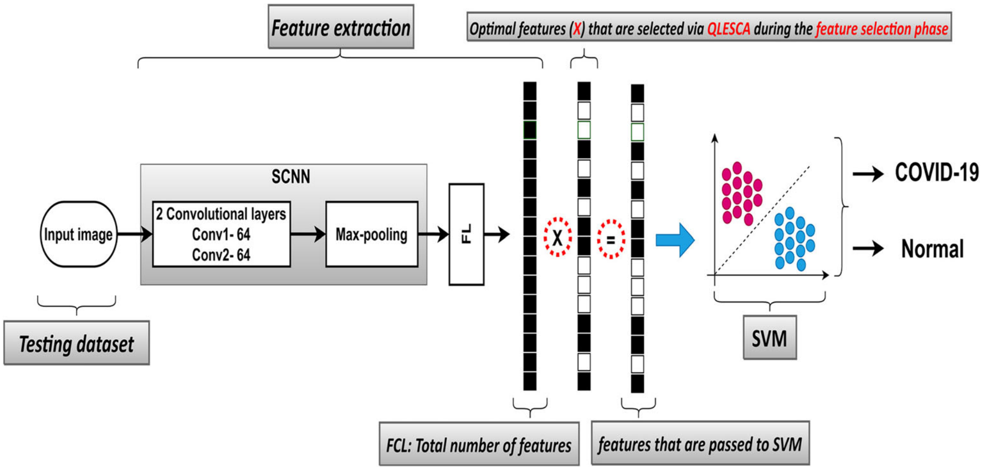

Feature extraction
Optimal features (X) that are selected via QLESCA during the feature selection phase
SCNN
Input image
2 Convolutional layers
Conv1- 64
Conv2- 64
Max-pooling
FL
X
=
COVID-19
Normal
SVM
Testing dataset
FCL: Total number of features
features that are passed to SVM

Table 5 The datasets that were involved in this research

Dataset	Category	Number of images	Training set	Testing set
Dataset 1 [77]	COVID-19	576	403	173
	healthy (Normal)	1583	1108	475
Dataset 2 [78, 79]	COVID-19	600	–	600
	healthy (Normal)	1583	–	1583
Dataset 3 [80]	COVID-19	1281	–	1281
	healthy (Normal)	3270	–	3270
Dataset 4 [81, 83, 84]	COVID-19	1626	–	1626
	healthy (Normal)	1802	–	1802
Dataset 5 [82, 85, 86]	COVID-19	765	–	765
	healthy (Normal)	431	–	431

Here, TP (true positives) denotes correctly predicted COVID-19 cases, FP (false positives) denotes Normal cases classified as COVID-19 by a model, TN (true negatives) denotes Normal cases classified as Normal cases, and FN (false negatives) denotes COVID-19 cases classified as Normal cases.

The performance of the QLESCA as an FS method was tested by comparing its results from the same proposed approach with those of nine recent optimization algorithms. These algorithms are the Sine Cosine Algorithm (SCA) [69], Opposition-Based Sine Cosine Algorithm (OBSCA) [89], Hybridization of SCA with Differential Evolution (DE) termed SCADE [90], Multi-Strategy SCA Algorithm (MSCA) [91], Arithmetic Optimization Algorithm (AOA) [92], Horse Herd Optimization Algorithm: A nature-inspired algorithm for high-dimensional optimization problems (HOA) [93], Farmland Fertility Algorithm (FFA) [94], African Vultures Optimization Algorithm (AVOA) [95], Artificial Gorilla Troops Optimizer (AGO) [96], and Q-learning Embedded Sine Cosine Algorithm (QLESCA) [44].

All of these algorithms were run with the settings shown in Table 7. As can be seen, the initial populations of these algorithms were not identical. While all the algorithms use a large population of search agents (30), only the QLESCA uses a small population of search agents (5) because its concept is based on a micro population. In order to make a fair comparison between these metaheuristic algorithms, the fitness counter parameter was utilized. The purpose of the fitness counter is to provide all algorithms with an equal opportunity to evaluate the objective function. For example, if SCA has 30 agents and this algorithm is executed 100 times, each agent will check the objective function at each iteration. That means 30 * 100 equals 3000 times in total. But in the case of the QLESCA, there are only five agents, which should give it the same opportunity to evaluate the objective function as SCA (3000 times). In this manner, all algorithms will function equally.

5.3 Results and discussion

In this subsection, all the experiment results for the proposed model are presented in order to determine the optimal fitness value. Also, 1) a statistical test analysis was conducted, 2) the QLESCA-SCNN-SVM model was tested using four different datasets, and 3) the limitations of the proposed method were presented. Table 8 displays the best, mean, median, worst, and standard deviation values for all algorithms tested. According to the reported results, the QLESCA outperformed the other algorithms, with a fitness evaluation mean of **−0.9957**. Further analysis was examined by plotting the convergence curve of the QLESCA as compared with other evaluated algorithms, as shown in Fig. 9. The horizontal axis denotes the 10^3 iterations, and the vertical axis represents the best score obtained.

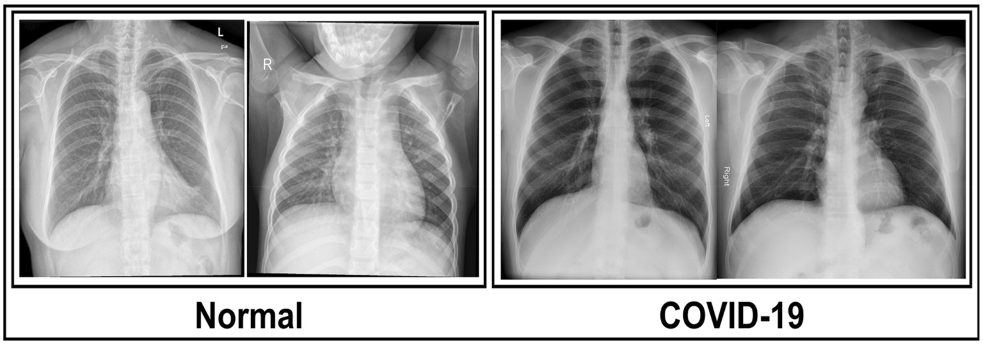


Fig. 8 Various Normal and COVID-19 case chest X-ray images

Table 6 The detailed settings of the utilized system

Hardware					
CPU	Frequency	RAM	GPU	SSD	Hard drive
Intel(R) Core (TM) i5-9400F	2.90 GHz	16 GB	Nvidia GeForce RTX 2070 Super	476 GB	1 TB
Software					
Operating system			Language		
Windows 10			MATLAB R2021a		

Table 7 Algorithms' parameter settings

Method	Population size	Maximum no. of iteration	Other parameters
QLESCA	5	10^3	α = 0.9 and γ = 0.1 All the parameters are set as in [44]
SCA	30	10^3	a = 2. All the parameters are set as in [69]
MSCA	30	10^3	a = 2; μ = 4; F = random in [0.2, 0.8]; P_c = 0.8. All the parameters are set as in [91]
SCADE	30	10^3	a = 2; F = random in [0.2, 0.8]; P_c = 0.8. All the parameters are set as in [91]
OBSCA	30	10^3	a = 2. All the parameters are set as in [91]
AOA	30	10^3	α = 5 and μ = 0.5. All the parameters are set as in [92]
HOA	30	10^3	All the parameters are set as in [93]
AVOA	30	10^3	All the parameters are set as in [94]
FFA	30	10^3	All the parameters are set as in [95]
GTO	30	10^3	All the parameters are set as in [96]

The proposed models were assessed using a variety of machine learning (ML) classification metrics, including accuracy, precision, specificity, sensitivity (recall), and F1-score. Table 9 shows the average experimental results for features that were optimized with the QLESCA, other optimization algorithms, or not at all. The QLESCA attained the highest average of accuracy, sensitivity, and F$_1$-score overall among the comparative optimization algorithms. It received average scores of 97.8086, 94.7399, and 95.9064, respectively.

The confusion matrix for each model is depicted in Fig. 10. This figure provides an overview of how all of the images were classified and where the majority of the misclassification occurred. According to this figure, the QLESCA had the highest COVID-19 detection rate, with 164 images and nine misclassifications. The QLESCA achieved the lowest misclassification of COVID-19 when compared to the other approaches.

Table 8 Results of the comparison between the QLESCA and state-of-the-art algorithms

Algorithm	Best	Mean	Median	Worst	STD
QLESCA	**−0.9999**	**−0.9957**	**−0.9952**	−0.9952	**0.0015**
SCA	−0.9953	−0.9939	−0.9952	−0.9906	0.0022
MSCA	−0.9952	−0.9920	−0.9906	−0.9906	0.0022
OBSCA	−0.9952	−0.9925	−0.9906	−0.9906	0.0024
SCADE	−0.9906	−0.9892	−0.9906	**−0.9860**	0.0022
AOA	−0.9952	−0.9920	−0.9906	−0.9906	0.0022
HOA	−0.9953	−0.9939	−0.9952	−0.9906	0.0022
AVOA	−0.9999	−0.9948	−0.9952	−0.99063	0.0026
FFA	−0.9952	−0.9906	−0.9906	−0.9860	0.0022
GTO	−0.9953	−0.991998	−0.99062	−0.9906	0.0022

Boldface is used to highlight the best result achieved

The number of features chosen by the optimization algorithms is shown in Table 10. As shown, the HOA achieved the fewest number of features, reducing the number from 100 to 24, while the QLESCA, SCA, MSCA, OBSCA, SCADE, AOA, AVOA, FFA, and GTO received 38, 26, 34, 43, 46, 34, 29, 49, and 32 respectively.

Despite the fact that the HOA, AOA, SCA, and MSCA used fewer features than the QLESCA, the QLESCA achieved: a) the best convergence curve during model training. b) a maximum detection accuracy of 97.8086%) the greatest number of correctly positive COVID-19, which is important in medical image processing due to its link to public health.

5.3.1 Statistical test analysis

This section statistically compares the suggested strategy against alternative methods to determine its superiority. The analysis employed the Wilcoxon rank-sum test [97]. The significance level was set to 0.05, which indicates that if the p value was less than 0.05, the result would be accepted (95% confidence level). The Wilcoxon rank-sum test results are presented in Table 11. In this table, the QLESCA optimizer's

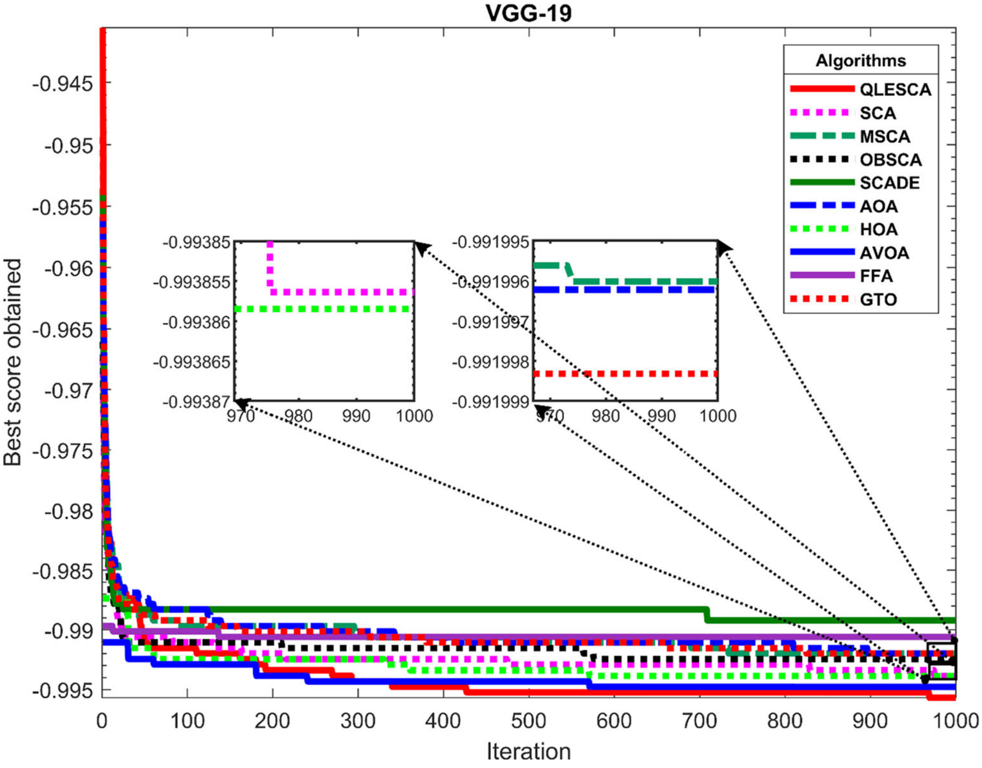


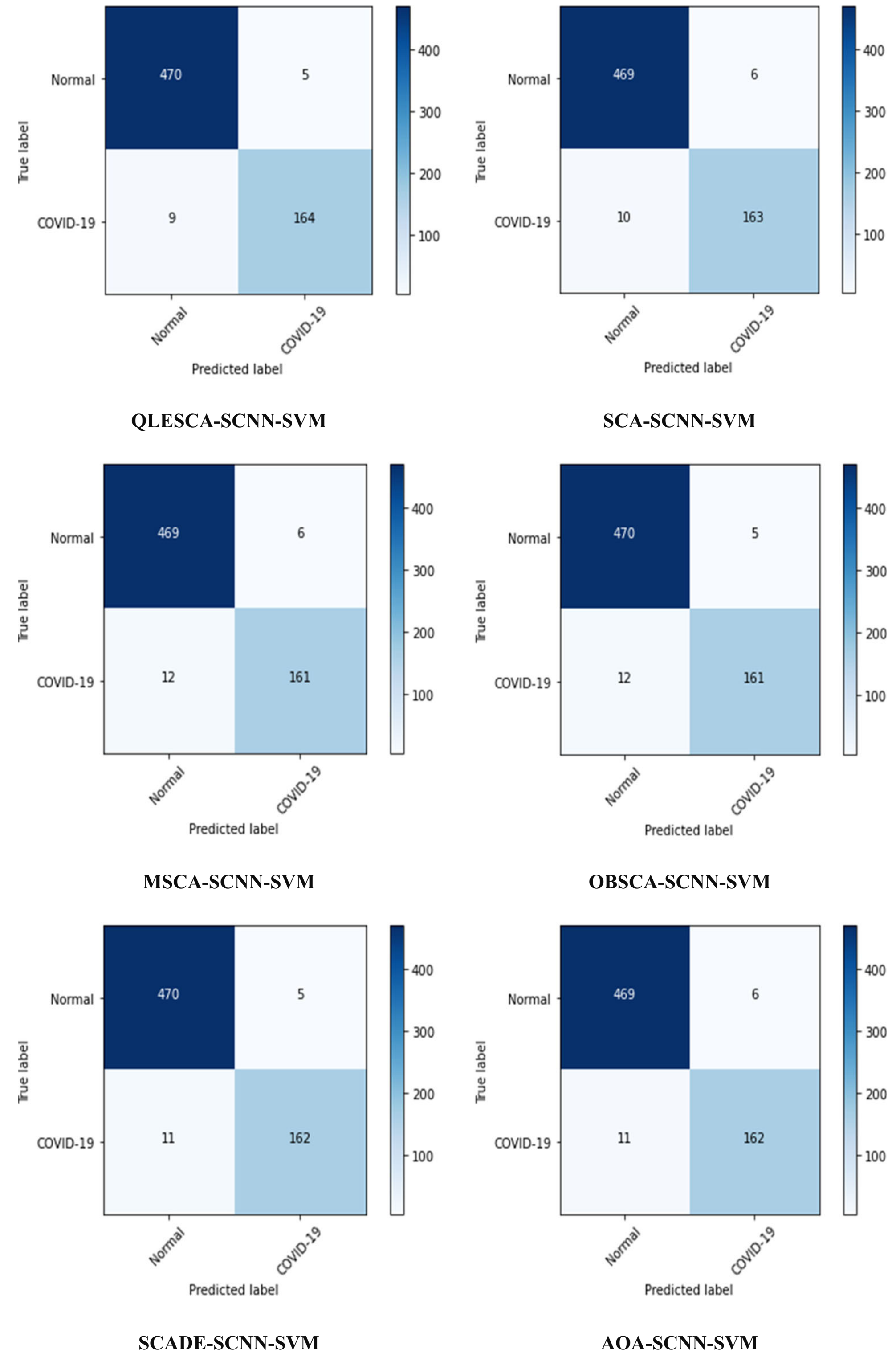
Fig. 9 Average performances of the state-of-the-art algorithms on SCNN-SVM. The QLESCA performs better than the others

performance is compared against various optimization methods, including the SCA, MSCA, OBSCA, SCADE, AOA, HOA, AVOA, FFA, and GTO.

As can be observed, all estimated p values are less than 0.05, except for the SCA and HOA, which do not differ significantly from the QLESCA. The main reason for these results is dependent on the method of calculating the fitness value. According to Eq. (4), fitness has two main components: the accuracy value and the number of selected features. When the number of selected features is high, it reduces the total value of fitness, while it increases it when the search agent selects a smaller number of features. Because (SCA and HOA) agents are unable to find a good solution (best features that increase the accuracy of the model), they tend to set the dimension of the solution (vector X_{1-100}) to zero, which results in the features being turned off, thereby reducing the number of features. So, both the SCA and HOA selected the lowest number of features, 26 and 24, respectively, compared with the 38 features selected by the QLESCA. That effect of p values despite the fact that the SCA and HOA do not

Table 9 Performance comparison between the QLESCA and other approaches

Model	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Specificity (%)	Sensitivity / Recall (%)	F1-score (%)
The proposed approach (QLESCA-SCNN-SVM)	164	470	5	9	97.8086	97.0414	98.9263	94.7399	95.9064
SCA-SCNN-SVM	163	469	6	10	97.4228	96.4497	98.6737	93.9884	95.3216
MSCA-SCNN-SVM	161	469	6	12	97.2840	96.4072	98.7579	93.2370	94.7059
OBSCA-SCNN-SVM	161	470	5	12	97.3765	96.9880	98.9474	93.0636	94.9853
SCADE-SCNN-SVM	162	470	5	11	97.5000	97.0060	98.8842	93.6994	95.2941
AOA-SCNN-SVM	162	469	6	11	97.3457	96.4286	98.6947	93.6416	95.0147
HOA-SCNN-SVM	162	469	6	11	97.3765	96.4286	98.8211	93.4104	95.0147
AVOA-SCNN-SVM	162	469	6	11	97.3765	96.4286	98.8211	93.4104	95.0147
FFA-SCNN-SVM	161	469	6	12	97.25309	96.4072	98.84211	92.89017	94.7059
GTO-SCNN-SVM	160	470	5	13	97.20679	96.9697	98.90526	92.54335	94.6746
SCNN-SVM	153	474	1	20	96.7590	99.3500	99.7890	88.4390	93.5780



QLESCA-SCNN-SVM

SCA-SCNN-SVM

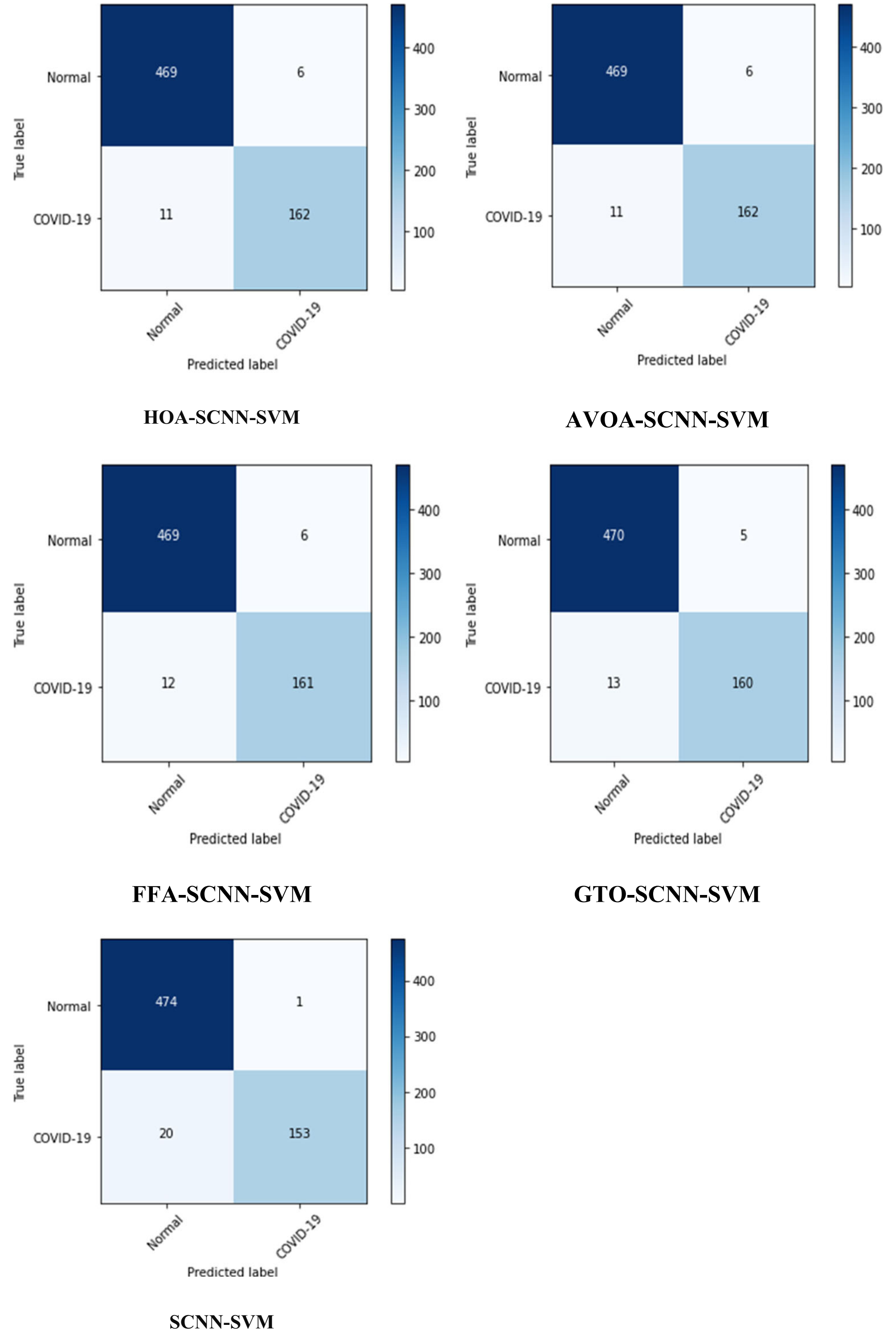
MSCA-SCNN-SVM

OBSCA-SCNN-SVM

SCADE-SCNN-SVM

AOA-SCNN-SVM

Fig. 10 Confusion matrix for all models



HOA-SCNN-SVM

AVOA-SCNN-SVM

FFA-SCNN-SVM

GTO-SCNN-SVM

SCNN-SVM

Fig. 10 (continued)

Table 10 Number of features chosen by the optimization algorithms

Model	Number of Features
The proposed approach (QLESCA-SCNN-SVM)	38
SCA-SCNN-SVM	26
MSCA-SCNN-SVM	34
OBSCA-SCNN-SVM	43
SCADE-SCNN-SVM	46
AOA-SCNN-SVM	34
HOA-SCNN-SVM	**24**
AVOA-SCNN-SVM	29
FFA-SCNN-SVM	49
GTO-SCNN-SVM	32
SCNN-SVM	100

Table 11 Wilcoxon rank-sum of the QLESCA against other optimizers ($P < 0.05$)

Model	p-value
SCA-SCNN-SVM	0.8200
MSCA-SCNN-SVM	0.0060
OBSCA-SCNN-SVM	0.0005
SCADE-SCNN-SVM	0.0002
AOA-SCNN-SVM	0.0130
HOA-SCNN-SVM	0.3840
AVOA-SCNN-SVM	0.0210
FFA-SCNN-SVM	0.0030
GTO-SCNN-SVM	0.0410

significantly differ from the QLESCA, was that these algorithms faced overfitting problems when evaluated on testing data and failed to achieve the highest accuracy results as were achieved by the QLESCA. This proves that the features that are selected by these algorithms are not too relevant and some of them affect the accuracy of the model in the evaluation process.

5.3.2 Additional evaluation

To ensure that the proposed method is stable, the best features selected by the QLESCA that achieved the highest accuracy on Dataset 1 were evaluated on four additional datasets (Dataset 2, Dataset 3, Dataset 4, and Dataset 5). Table 12 summarizes the experimental results of QLESCA-SCNN-SVM on four distinct datasets. The confusion matrix is depicted in Fig. 11. In Dataset 2, the model correctly detected 560 COVID-19 cases with 40 misclassifications and 1563 normal cases with 20 misclassifications. This model correctly identified COVID-19 in Dataset 3 with 1202 images and 79 misclassified images. It also correctly identified normal images with 3227 normal images and 43 misclassified images. In Dataset 4, the model correctly detected 1545 COVID-19 cases with 81 misclassifications and 1781 normal cases with 21 misclassifications. In Dataset 5, the model correctly detected 739 COVID-19 cases with 26 misclassifications and 426 normal cases with five misclassifications.

5.3.3 The limitations of the proposed method

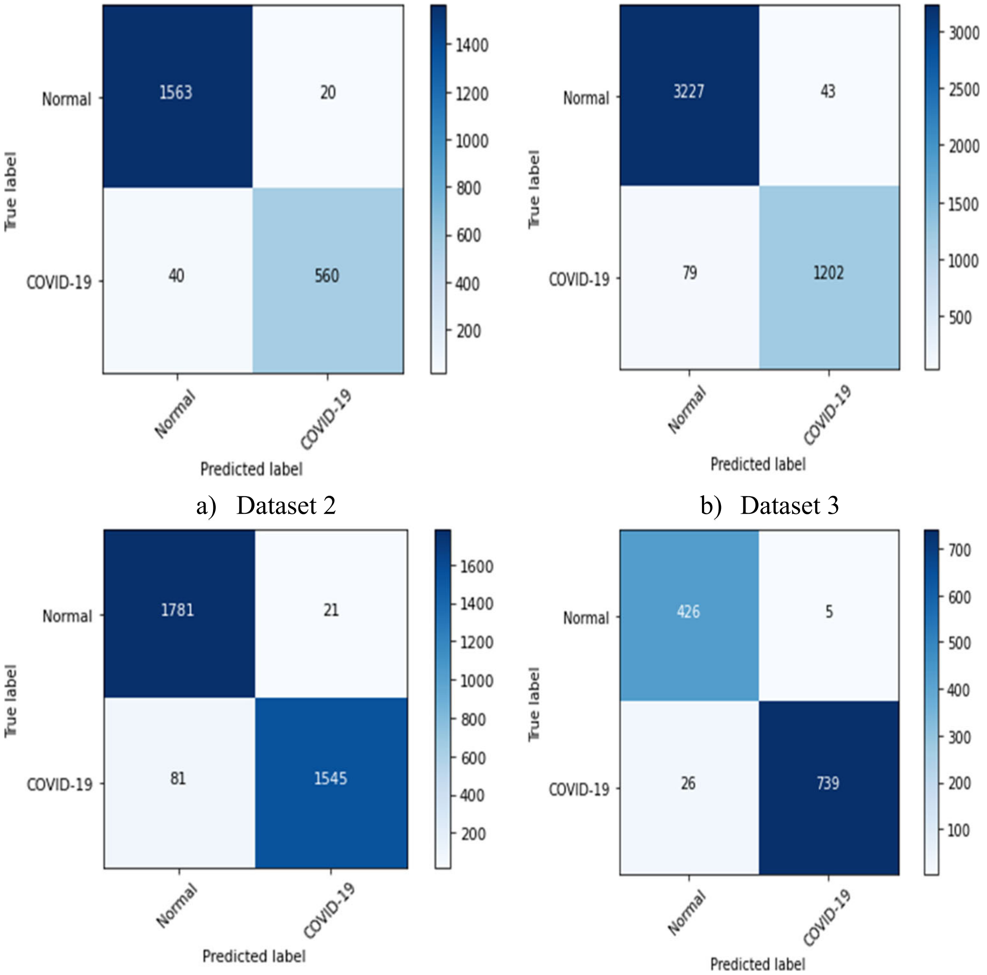
After presenting the results of the proposed model, it would be appropriate to discuss the method's limitations. Due to the high computational cost of fitness evaluation during model training, the proposed approach is still facing the following drawbacks.

1. The model was trained with only a maximum of 1000 fitness evaluations, and a larger number of iterations will result in a huge training time.
2. The above constraint in the maximum number of fitness evaluations prevents the Q-table of the QLESCA having sufficient time to be built/trained well.
3. The proposed approach has used a limited number of training images (20 images) for training the SVM which effects the performances of the classifier, and it is better to have a large number of training images such as 100 images per class.

Table 12 The experimental results of QLESCA-SCNN-SVM achieved on other datasets

	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Specificity (%)	Sensitivity / Recall (%)	F1-score (%)
Dataset 2	560	1563	20	40	97.2515	96.5517	98.7366	93.3333	94.9153
Dataset 3	1202	3227	43	79	97.3193	96.5462	98.6850	93.8329	95.1702
Dataset 4	1545	1781	21	81	97.0245	98.6590	98.8346	95.0185	96.8045
Dataset 5	739	426	5	26	97.4080	99.3280	98.8399	96.6013	97.9457

Fig. 11 Confusion matrix for the selected features via the QLESCA on other datasets



a) Dataset 2

b) Dataset 3

c) Dataset 4

d) Dataset 5

6 Conclusions and future works

This study developed a framework to detect COVID-19 cases from chest X-ray images. The proposed framework depends on a combination of the shallow VGG19 model and a metaheuristic optimization algorithm. A shallow VGG19 was used to extract the features. By contrast, a recent optimization algorithm named Q-Learning Embedded Sine Cosine Algorithm (QLESCA) was used for feature selection. An efficient fitness equation has been proposed that ensures the balance between the overall accuracy of the model and the number of selected features. The QLESCA was compared to nine recent optimization algorithms, and the results of the comparison demonstrated the superior performance of the proposed method based on the QLESCA over the other comparative optimization algorithms, with an accuracy of 97.8086% and a reduction in the number of features from 100 to 38. For additional evaluation, four recent datasets were used to evaluate the selected features by the QLESCA, and both of them achieved a high degree of accuracy of greater than 97%.

In future work, we intend to: 1) test our model on other medical image datasets such as breast cancer, skin cancer, and so on; 2) due to the model's lighter weight, we intend to deploy it on IoT devices, mobile phones, and drones that farmers can use to detect diseased crops early and prevent the disease's spread; and 3) the same methodology can be used to investigate other types of pre-trained CNNs such as GoogleNet, ResNet, DenseNet, and so on.

Acknowledgments This research is supported by the Malaysia Ministry of Higher Education (MOHE) Fundamental Research Grant Scheme (FRGS), no. FRGS/1/2019/ICT02/USM/03/3.

The authors would like to thank Professor Dr. Huiling Chen, College of Computer Science and Artificial Intelligence, Wenzhou University, China for sharing the MSCA, OBSCA, SCADE codes that were used in the comparison experiments.

References

1. Woźniak M, Siłka J, Wieczorek M (2021) Deep neural network correlation learning mechanism for CT brain tumor detection. Neural Comput. Appl. https://doi.org/10.1007/s00521-021-05841-x
2. Garg S, Kumar S, Muhuri PK (2022) A novel approach for COVID-19 infection forecasting based on multi-source deep transfer learning. Comput Biol Med 149:105915. https://doi.org/10.1016/j.compbiomed.2022.105915
3. Al-antari MA, Hua C-H, Bang J, Lee S (2021) Fast deep learning computer-aided diagnosis of COVID-19 based on digital chest x-ray images. Appl Intell 51(5):2890–2907. https://doi.org/10.1007/s10489-020-02076-6
4. Singh KK, Kumar S, Dixit P, Bajpai MK (2021) Kalman filter based short term prediction model for COVID-19 spread. Appl Intell 51(5):2714–2726. https://doi.org/10.1007/s10489-020-01948-1
5. Shuja J, Alanazi E, Alasmary W, Alashaikh A (2021) COVID-19 open source data sets: a comprehensive survey. Appl Intell 51(3): 1296–1325. https://doi.org/10.1007/s10489-020-01862-6
6. Zebin T, Rezvy S (2021) COVID-19 detection and disease progression visualization: deep learning on chest X-rays for classification and coarse localization. Appl Intell 51(2):1010–1021. https://doi.org/10.1007/s10489-020-01867-1
7. Samson ABP, Annavarapu CSR (2021) Deep learning-based improved snapshot ensemble technique for COVID-19 chest X-ray classification. Appl Intell 51(5):3104–3120. https://doi.org/10.1007/s10489-021-02199-4
8. Ilhan HO, Serbes G, Aydin N (2022) Decision and feature level fusion of deep features extracted from public COVID-19 data-sets. Appl Intell 52(8):8551–8571. https://doi.org/10.1007/s10489-021-02945-8
9. Ter-Sarkisov A (2022) COVID-CT-mask-net: prediction of COVID-19 from CT scans using regional features. Appl Intell 52(9):9664–9675. https://doi.org/10.1007/s10489-021-02731-6
10. Chakraborty M, Dhavale SV, Ingole J (2021) Corona-Nidaan: lightweight deep convolutional neural network for chest X-ray based COVID-19 infection detection. Appl Intell 51(5):3026–3043. https://doi.org/10.1007/s10489-020-01978-9
11. Sen S, Saha S, Chatterjee S, Mirjalili S, Sarkar R (2021) A bi-stage feature selection approach for COVID-19 prediction using chest CT images. Appl Intell 51(12):8985–9000. https://doi.org/10.1007/s10489-021-02292-8
12. Choudhary T, Gujar S, Goswami A, Mishra V, Badal T (2022) “Deep learning-based important weights-only transfer learning approach for COVID-19 CT-scan classification,” Appl. Intell., https://doi.org/10.1007/s10489-022-03893-7
13. Canayaz M (2021) MH-COVIDNet: diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images. Biomed Signal Proc Contr 64:102257. https://doi.org/10.1016/j.bspc.2020.102257
14. Sahlol AT, Kollmannsberger P, Ewees AA (2020) Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features. Sci. Rep. 10(1):2536. https://doi.org/10.1038/s41598-020-59215-9
15. Shah FM et al (2021) A Comprehensive Survey of COVID-19 Detection Using Medical Images. SN Comput. Sci. 2(6):434. https://doi.org/10.1007/s42979-021-00823-1
16. Fatani A, Dahou A, Al-qaness MAA, Lu S, Abd Elaziz MA (2021) Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System. *Sensors* 22(1):140. https://doi.org/10.3390/s22010140
17. Holland JH, “Genetic Algorithms,” *Sci. Am.*, vol. 267, no. 1, pp. 66–73, (1992), [Online]. Available: http://www.jstor.org/stable/24939139
18. Kennedy J, Eberhart R (1995) “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968
19. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. https://doi.org/10.1109/4235.585893
20. Wang G-G, Deb S, Cui Z (2019) Monarch butterfly optimization. Neural Comput Appl 31(7):1995–2014. https://doi.org/10.1007/s00521-015-1923-y
21. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst 111:300–323. https://doi.org/10.1016/j.future.2020.03.055
22. Wang G-G (2018) Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Comput 10(2):151–164. https://doi.org/10.1007/s12293-016-0212-3
23. Yang Y, Chen H, Heidari AA, Gandomi AH (2021) Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. Expert Syst Appl 177:114864. https://doi.org/10.1016/j.eswa.2021.114864
24. Ahmadianfar I, Heidari AA, Gandomi AH, Chu X, Chen H (2021) RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. Expert Syst Appl 181:115079. https://doi.org/10.1016/j.eswa.2021.115079
25. Tu J, Chen H, Wang M, Gandomi AH (2021) The Colony predation algorithm. J Bionic Eng 18(3):674–710. https://doi.org/10.1007/s42235-021-0050-y
26. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872. https://doi.org/10.1016/j.future.2019.02.028
27. Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. Expert Syst Appl 116:147–160. https://doi.org/10.1016/j.eswa.2018.08.051
28. Abdel-Basset M, El-Shahat D, El-henawy I, de Albuquerque VHC, Mirjalili S (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. Expert Syst Appl 139:112824. https://doi.org/10.1016/j.eswa.2019.112824
29. Hu P, Pan J-S, Chu S-C (2020) Improved binary Grey wolf optimizer and its application for feature selection. Knowledge-Based Syst. 195:105746. https://doi.org/10.1016/j.knosys.2020.105746
30. Ouadfel S, Abd Elaziz M (2020) Enhanced crow search algorithm for feature selection. Expert Syst Appl 159:113572. https://doi.org/10.1016/j.eswa.2020.113572
31. Baş E, Ülker E (2020) An efficient binary social spider algorithm for feature selection problem. Expert Syst Appl 146:113185. https://doi.org/10.1016/j.eswa.2020.113185
32. Kılıç F, Kaya Y, Yildirim S (2021) A novel multi population based particle swarm optimization for feature selection. Knowledge-Based Syst 219:106894. https://doi.org/10.1016/j.knosys.2021.106894
33. Tubishat M, Ja'afar S, Alswaitti M, Mirjalili S, Idris N, Ismail MA, Omar MS (2021) Dynamic Salp swarm algorithm for feature selection. Expert Syst Appl 164:113873. https://doi.org/10.1016/j.eswa.2020.113873
34. Sadeghian Z, Akbari E, Nematzadeh H (2021) A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. Eng Appl Artif Intell 97:104079. https://doi.org/10.1016/j.engappai.2020.104079
35. Başaran E (2022) A new brain tumor diagnostic model: selection of textural feature extraction algorithms and convolution neural network features with optimization algorithms. Comput Biol Med 148:105857. https://doi.org/10.1016/j.compbiomed.2022.105857

36. Wang J, Lin D, Zhang Y, Huang S (2022) An adaptively balanced grey wolf optimization algorithm for feature selection on high-dimensional classification. Eng Appl Artif Intell 114:105088. https://doi.org/10.1016/j.engappai.2022.105088
37. Kaur B, Rathi S, Agrawal RK (2022) Enhanced depression detection from speech using quantum whale optimization algorithm for feature selection. Comput Biol Med 150:106122. https://doi.org/10.1016/j.compbiomed.2022.106122
38. Long W, Xu M, Jiao J, Wu T, Tang M, Cai S (2022) A velocity-based butterfly optimization algorithm for high-dimensional optimization and feature selection. Expert Syst Appl 201:117217. https://doi.org/10.1016/j.eswa.2022.117217
39. Khosravi H, Amiri B, Yazdanjue N, Babaiyan V (2022) An improved group teaching optimization algorithm based on local search and chaotic map for feature selection in high-dimensional data. Expert Syst Appl 204:117493. https://doi.org/10.1016/j.eswa.2022.117493
40. Liu Q, Liu M, Wang F, Xiao W (2022) A dynamic stochastic search algorithm for high-dimensional optimization problems and its application to feature selection. Knowledge-Based Syst. 244:108517. https://doi.org/10.1016/j.knosys.2022.108517
41. Tiwari A, Chaturvedi A (2022) A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification. Expert Syst Appl 196:116621. https://doi.org/10.1016/j.eswa.2022.116621
42. Yedukondalu J, Sharma LD (2022) "Cognitive load detection using circulant singular spectrum analysis and Binary Harris Hawks Optimization based feature selection," *Biomed. Signal Process. Control*, p. 104006, https://doi.org/10.1016/j.bspc.2022.104006
43. Xu Z, Heidari AA, Kuang F, Khalil A, Mafarja M, Zhang S, Chen H, Pan Z (2023) Enhanced Gaussian bare-bones grasshopper optimization: mitigating the performance concerns for feature selection. Expert Syst Appl 212:118642. https://doi.org/10.1016/j.eswa.2022.118642
44. Hamad QS, Samma H, Suandi SA, Mohamad-Saleh J (2022) Q-learning embedded sine cosine algorithm (QLESCA). Expert Syst Appl 193:116417. https://doi.org/10.1016/j.eswa.2021.116417
45. Dash S et al (2022) Guidance Image-Based Enhanced Matched Filter with Modified Thresholding for Blood Vessel Extraction. *Symmetry (Basel)* 14(2):194. https://doi.org/10.3390/sym14020194
46. Wieczorek M, Silka J, Wozniak M, Garg S, Hassan MM (2022) Lightweight convolutional neural network model for human face detection in risk situations. IEEE Trans Ind Informatics 18(7):4820–4829. https://doi.org/10.1109/TII.2021.3129629
47. Middleton S, Dimbath E, Pant A, George SM, Maddipati V, Peach MS, Yang K, Ju AW, Vahdati A (2022) Towards a multi-scale computer modeling workflow for simulation of pulmonary ventilation in advanced COVID-19. Comput Biol Med 145:105513. https://doi.org/10.1016/j.compbiomed.2022.105513
48. Abdullah D, Susilo S, Ahmar AS, Rusli R, Hidayat R (2022) The application of K-means clustering for province clustering in Indonesia of the risk of the COVID-19 pandemic based on COVID-19 data. Qual Quant 56(3):1283–1291. https://doi.org/10.1007/s11135-021-01176-w
49. Abir FF, Alyafei K, Chowdhury MEH, Khandakar A, Ahmed R, Hossain MM, Mahmud S, Rahman A, Abbas TO, Zughaier SM, Naji KK (2022) PCovNet: a presymptomatic COVID-19 detection framework using deep learning model using wearables data. Comput Biol Med 147:105682. https://doi.org/10.1016/j.compbiomed.2022.105682
50. Ahmad M, Sadiq S, Eshmawi A'A, Alluhaidan AS, Umer M, Ullah S, Nappi M (2022) Industry 4.0 technologies and their applications in fighting COVID-19 pandemic using deep learning techniques. Comput Biol Med 145:105418. https://doi.org/10.1016/j.compbiomed.2022.105418
51. Akter S, das D, Haque RU, Quadery Tonmoy MI, Hasan MR, Mahjabeen S, Ahmed M (2022) AD-CovNet: an exploratory analysis using a hybrid deep learning model to handle data imbalance, predict fatality, and risk factors in Alzheimer's patients with COVID-19. Comput Biol Med 146:105657. https://doi.org/10.1016/j.compbiomed.2022.105657
52. Polsinelli M, Cinque L, Placidi G (2020) A light CNN for detecting COVID-19 from CT scans of the chest. Pattern Recogn Lett 140:95–100. https://doi.org/10.1016/j.patrec.2020.10.001
53. Ahuja S, Panigrahi BK, Dey N, Rajinikanth V, Gandhi TK (2021) Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices. Appl Intell 51(1):571–585. https://doi.org/10.1007/s10489-020-01826-w
54. Turkoglu M (2021) COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble. Appl Intell 51(3):1213–1226. https://doi.org/10.1007/s10489-020-01888-w
55. Jain R, Gupta M, Taneja S, Hemanth DJ (2021) Deep learning based detection and analysis of COVID-19 on chest X-ray images. Appl Intell 51(3):1690–1700. https://doi.org/10.1007/s10489-020-01902-1
56. Punn NS, Agarwal S (2021) Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks. Appl Intell 51(5):2689–2702. https://doi.org/10.1007/s10489-020-01900-3
57. Mukherjee H, Ghosh S, Dhar A, Obaidullah SM, Santosh KC, Roy K (2021) Deep neural network to detect COVID-19: one architecture for both CT scans and chest X-rays. Appl Intell 51(5):2777–2789. https://doi.org/10.1007/s10489-020-01943-6
58. Li D, Fu Z, Xu J (2021) Stacked-autoencoder-based model for COVID-19 diagnosis on CT images. Appl Intell 51(5):2805–2817. https://doi.org/10.1007/s10489-020-02002-w
59. Too J, Mirjalili S (2021) A hyper learning binary dragonfly algorithm for feature selection: a COVID-19 case study. Knowledge-Based Syst. 212:106553. https://doi.org/10.1016/j.knosys.2020.106553
60. Piri J et al (2022) Feature Selection Using Artificial Gorilla Troop Optimization for Biomedical Data: A Case Analysis with COVID-19 Data. Mathematics 10(15):2742. https://doi.org/10.3390/math10152742
61. Nadimi-Shahraki MH, Zamani H, Mirjalili S (2022) Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study. Comput Biol Med 148:105858. https://doi.org/10.1016/j.compbiomed.2022.105858
62. Ahmed S, Sheikh KH, Mirjalili S, Sarkar R (2022) Binary simulated Normal distribution optimizer for feature selection: theory and application in COVID-19 datasets. Expert Syst Appl 200:116834. https://doi.org/10.1016/j.eswa.2022.116834
63. Abd Elaziz M, Dahou A, Alsaleh NA, Elsheikh AH, Saba AI, Ahmadein M (2021) Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm. Entropy 23(11):1383. https://doi.org/10.3390/e23111383
64. Bandyopadhyay R, Basu A, Cuevas E, Sarkar R (2021) Harris hawks optimisation with simulated annealing as a deep feature selection method for screening of COVID-19 CT-scans. Appl Soft Comput 111:107698. https://doi.org/10.1016/j.asoc.2021.107698
65. Simonyan K, Zisserman A (2014) "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*
66. Darwish A, Ezzat D, Hassanien AE (2020) An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. Swarm Evol Comput 52:100616. https://doi.org/10.1016/j.swevo.2019.100616
67. Hemdan EED, Shouman MA, Karar ME (2020) "COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images," *arXiv*, Accessed: Nov. 20, 2020. [Online]. Available: http://arxiv.org/abs/2003.11055

68. Balaha HM, El-Gendy EM, Saafan MM (2021) CovH2SD: a COVID-19 detection approach based on Harris hawks optimization and stacked deep learning. Expert Syst Appl 186:115805. https://doi.org/10.1016/j.eswa.2021.115805
69. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowledge-Based Syst. 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
70. Hamad QS, Samma H, Suandi SA, Saleh JM (2022) "Study of VGG-19 Depth in Transfer Learning for COVID-19 X-Ray Image Classification," penang- Malaysia: Lecture Notes in Electrical Engineering - Springer, pp. 930–935
71. Mukherjee H, Ghosh S, Dhar A, Obaidullah SM, Santosh KC, Roy K (2021) "Shallow convolutional neural network for COVID-19 outbreak screening using chest X-rays," Cognit. Comput, https://doi.org/10.1007/s12559-020-09775-9
72. Schindler A, Lidy T, and Rauber A (2016) "Comparing shallow versus deep neural network architectures for automatic music genre classification," in *9th Forum Media Technology (FMT2016)*, vol. 1734, pp. 17–21, [Online]. Available: https://pub-inf.tuwien.ac.at/showentry.php?ID=256008
73. Li Y, Nie J, Chao X (2020) Do we really need deep CNN for plant diseases identification? *Comput. Electron. Agric* 178(August): 105803. https://doi.org/10.1016/j.compag.2020.105803
74. Wang L, Chen A, Zhang Y, Wang X, Zhang Y, Shen Q, Xue Y (2020) AK-DL: A Shallow Neural Network Model for Diagnosing Actinic Keratosis with Better Performance than Deep Neural Networks. Diagnostics 10(4):217. https://doi.org/10.3390/diagnostics10040217
75. Impedovo D, Dentamaro V, Abbattista G, Gattulli V, Pirlo G (2021) A comparative study of shallow learning and deep transfer learning techniques for accurate fingerprints vitality detection. Pattern Recogn Lett 151:11–18. https://doi.org/10.1016/j.patrec.2021.07.025
76. Luo Z, Jin S, Li Z, Huang H, Xiao L, Chen H, Heidari AA, Hu J, Chen C, Chen P, Hu Z (2022) Hierarchical Harris hawks optimization for epileptic seizure classification. Comput Biol Med 145: 105397. https://doi.org/10.1016/j.compbiomed.2022.105397
77. PATEL P (2021) "kaggle (Covid-19 & Normal)," . https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia ()
78. Cohen JP, Morrison P, Dao L, Roth K, Duong TQ, Ghassemi M (2020) "COVID-19 Image Data Collection: Prospective Predictions Are the Future", Accessed: Jun. 02, 2021. [Online]. Available: http://arxiv.org/abs/2006.11988
79. Mooney P, "Chest X-ray images (Normal) | Kaggle", (2019) https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia (accessed Jun. 02, 2021)
80. Rahman T, Khandakar A, Kadir MA, Islam KR, Islam KF, Mazhar R, Hamid T, Islam MT, Kashem S, Mahbub ZB, Ayari MA, Chowdhury MEH (2020) Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access 8:191586–191601. https://doi.org/10.1109/ACCESS.2020.3031384
81. Kumar S, "Covid19-pneumonia-Normal chest X-ray images", (2022) https://data.mendeley.com/datasets/dvntn9yhd2 (accessed Oct. 05, 2022)
82. Danilov V, Proutski A, Kirpich A, Litmanovich D, Gankin Y (2022) "Dataset for COVID-19 segmentation and severity scoring," https://data.mendeley.com/datasets/36fjrg9s69 (accessed Oct. 06, 2022)
83. Kumar S, Shastri S, Mahajan S, Singh K, Gupta S, Rani R, Mohan N, Mansotra V (2022) A lightweight deep neural network model for detection of COVID-19 using X-ray images. Int J Imaging Syst Technol 32(5):1464–1480. https://doi.org/10.1002/ima.22770
84. Shastri S, Kansal I, Kumar S, Singh K, Popli R, Mansotra V (2022) CheXImageNet: a novel architecture for accurate classification of Covid-19 with chest x-ray digital images using deep convolutional neural networks. Health Technol (Berl) 12(1):193–204. https://doi.org/10.1007/s12553-021-00630-x
85. Danilov VV et al (2022) Automatic scoring of COVID-19 severity in X-ray imaging based on a novel deep learning workflow. Sci. Rep. 12(1):12791. https://doi.org/10.1038/s41598-022-15013-z
86. Danilov VV, Proutski A, Karpovsky A, Kirpich A, Litmanovich D, Nefaridze D, Talalov O, Semyonov S, Koniukhovskii V, Shvartc V, Gankin Y (2022) Indirect supervision applied to COVID-19 and pneumonia classification. Informatics Med Unlocked 28:100835. https://doi.org/10.1016/j.imu.2021.100835
87. Toğaçar M, Ergen B, Cömert Z (2020) COVID-19 detection using deep learning models to exploit social mimic optimization and structured chest X-ray images using fuzzy color and stacking approaches. Comput Biol Med 121:103805. https://doi.org/10.1016/j.compbiomed.2020.103805
88. Jin W, Dong S, Dong C, Ye X (2021) Hybrid ensemble model for differential diagnosis between COVID-19 and common viral pneumonia by chest X-ray radiograph. Comput Biol Med 131:104252. https://doi.org/10.1016/j.compbiomed.2021.104252
89. Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500. https://doi.org/10.1016/j.eswa.2017.07.043
90. Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl Soft Comput 62:1019–1043. https://doi.org/10.1016/j.asoc.2017.09.039
91. Chen H, Wang M, Zhao X (2020) A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. Appl Math Comput 369:124872. https://doi.org/10.1016/j.amc.2019.124872
92. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609. https://doi.org/10.1016/j.cma.2020.113609
93. MiarNaeimi F, Azizyan G, Rashki M (2021) Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems. Knowledge-Based Syst. 213:106711. https://doi.org/10.1016/j.knosys.2020.106711
94. Shayanfar H, Gharehchopogh FS (2018) Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. Appl Soft Comput 71:728–746. https://doi.org/10.1016/j.asoc.2018.07.033
95. Abdollahzadeh B, Gharehchopogh FS, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 158: 107408. https://doi.org/10.1016/j.cie.2021.107408
96. Abdollahzadeh B, Soleimanian Gharehchopogh F, Mirjalili S (2021) Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int J Intell Syst 36(10):5887–5958. https://doi.org/10.1002/int.22535
97. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci (Ny) 180(10):2044–2064. https://doi.org/10.1016/j.ins.2009.12.010

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

Qusay Shihab Hamad received his bachelor's and master's degrees in computer engineering from the University of Technology, Baghdad, Iraq in 2011 and 2014, respectively. He is currently doing his Ph.D. in computational intelligence at the School of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM). He is also a lecturer at the University of Information Technology and Communications (UOITC) in Baghdad, Iraq. His research interests include computational intelligence, computer vision, deep learning, and swarm intelligence.

Hussein Samma received his Ph.D. degree in computer vision and machine learning from Universiti Sains Malaysia (2016), M.Eng. degree in computer engineering from Jordan University of Science and Technology (2009) B.Eng. degree in computer engineering from Yarmouk University, Jordan (2006). His research interest in computer vision, deep learning, swarm intelligence, and soft biometrics. He is a member of IEEE computational intelligence society, and he has served as a reviewer for several well-known international journals such as machine vision and applications. He is currently an associate professor at the SDAIA-KFUPM Joint Research Center for Artificial Intelligence (JRC-AI) King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

Shahrel Azmin Suandi (Senior Member, IEEE) received the B.Eng. degree in electronic engineering and the M.Eng. and D.Eng. degrees in information science from the Kyushu Institute of Technology, Fukuoka, Japan, in 1995, 2003, and 2006, respectively. He joined the Industries, such as Sony Video (M) Sdn. Bhd., and Technology Park Malaysia Corporation Sdn. Bhd., for a period of six years, as an Engineer. He is currently a Professor and the Dean of the School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia. His current research interests include face-based biometrics, real-time object detection and tracking, and pattern classification using deep learning. He served as a Reviewer for several international conferences and journals, including IET Biometrics, IET Computer Vision, Multimedia Tools and Applications, Neural Computing and Applications, Journal of Electronic Imaging, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE ACCESS, and so on.