RESEARCH **Open Access**

CNN-IKOA: convolutional neural network with improved Kepler optimization algorithm for image segmentation: experimental validation and numerical exploration

Mohamed Abdel-Basset[1], Reda Mohamed[1], Ibrahim Alrashdi[2*], Karam M. Sallam[3,4*] and Ibrahim A. Hameed[5*]

*Correspondence:
irrashdi@ju.edu.sa;
KSallam@sharjah.ac.ae; karam.sallam@canberra.edu.au;
ibib@ntnu.no

[1] Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt
[2] Department of Computer Science, College of Computer and Information Sciences, Jouf University, 2014 Sakaka, Saudi Arabia
[3] Department of Computer Science, University of Sharjah, Sharjah, United Arab Emirates
[4] School of IT and Systems, Faculty of Science and Technology, University of Canberra, Canberra 2601, Australia
[5] Department of ICT and Natural Sciences, Norwegian University of Science and Technology (NTNU), 7491 Ålesund, Norway

Abstract

Chest diseases, especially COVID-19, have quickly spread throughout the world and caused many deaths. Finding a rapid and accurate diagnostic tool was indispensable to combating these diseases. Therefore, scientists have thought of combining chest X-ray (CXR) images with deep learning techniques to rapidly detect people infected with COVID-19 or any other chest disease. Image segmentation as a preprocessing step has an essential role in improving the performance of these deep learning techniques, as it could separate the most relevant features to better train these techniques. Therefore, several approaches were proposed to tackle the image segmentation problem accurately. Among these methods, the multilevel thresholding-based image segmentation methods won significant interest due to their simplicity, accuracy, and relatively low storage requirements. However, with increasing threshold levels, the traditional methods have failed to achieve accurate segmented features in a reasonable amount of time. Therefore, researchers have recently used metaheuristic algorithms to tackle this problem, but the existing algorithms still suffer from slow convergence speed and stagnation into local minima as the number of threshold levels increases. Therefore, this study presents an alternative image segmentation technique based on an enhanced version of the Kepler optimization algorithm (KOA), namely IKOA, to better segment the CXR images at small, medium, and high threshold levels. Ten CXR images are used to assess the performance of IKOA at ten threshold levels (T-5, T-7, T-8, T-10, T-12, T-15, T-18, T-20, T-25, and T-30). To observe its effectiveness, it is compared to several metaheuristic algorithms in terms of several performance indicators. The experimental outcomes disclose the superiority of IKOA over all the compared algorithms. Furthermore, the IKOA-based segmented CXR images at eight different threshold levels are used to train a newly proposed CNN model called CNN-IKOA to find out the effectiveness of the segmentation step. Five performance indicators, namely overall accuracy, precision, recall, F1-score, and specificity, are used to disclose the CNN-IKOA's effectiveness. CNN-IKOA, according to the experimental outcomes, could achieve outstanding outcomes for the images segmented at T-12, where it could reach 94.88% for overall accuracy, 96.57% for specificity, 95.40% for precision, and 95.40% for recall.

Keywords: Optimization, AI, Convolutional neural network, Chest diseases, X-ray, Global health, Artificial intelligence

Introduction

The Coronavirus disease (COVID-19), which was discovered in December 2019, has infected several people all over the world and has caused the deaths of a significant number of those people [1, 2]. To prevent the outbreak of this pandemic, in the beginning, scientists have paid attention to the reverse transcriptase-polymerase chain reaction (RT-PCR) as the most prevalent tool for COVID-19 diagnosis. Although RTPCR has a high accuracy for diagnosing this disease, it is expensive, sluggish, and in high demand [3]. As a result, computed tomography (CT) and X-ray images as important alternative tools were used for the early and speedy detection of COVID-19 [4]. The chest X-ray (CXR) images are cheaper and faster, but the detection of COVID-19 manually from the lungs in those images is hard to achieve and might cause some wrong diagnoses [3]. Therefore, deep learning (DL) techniques have been extensively used to aid in detecting COVID-19 infection from the CXR images accurately and quickly [5, 6]. In general, deep learning and machine learning have been applied to detecting several diseases, such as breast cancer [7], heart diseases [8], skin diseases [9], and plant diseases [10].

Chest X-ray image segmentation problem (CXIS) is an essential part of image processing and computer vision for easy analysis and interpretation [11]. The multilevel threshold image segmentation method is often considered to be the most frequent and effective method for image segmentation owing to its simplicity, precision, and relatively low storage requirements [12]. The CXIS is an essential preparation step for the deep learning models to strengthen their performance for accurately detecting COVID-19 infection. However, the multilevel thresholding-based image segmentation problem is considered a complicated problem, especially with increasing the threshold level, and could not be accurately solved using traditional techniques such as Kapur's entropy and Otsu's method [13]. Therefore, over the last few decades, researchers have used metaheuristic algorithms for accurately tackling this problem in a reasonable amount of time. The reason for using those algorithms to solve this problem is that they have great results for several difficult discrete and continuous optimization problems, including feature selection [7, 14], the parameter estimation problem [15, 16], hyperparameter tuning [14, 17], and 0–1 Knapsack problems [18].

As aforementioned the metaheuristic optimization techniques have an important role in accurately tackling the CXIS problem to separate the homogenous regions for improving the classification accuracy of the deep learning models. Although several studies have been presented in the literature for segmenting COVID-19 images using metaheuristics, they have some drawbacks that do not qualify them as the best alternatives for tackling this problem. Those drawbacks are summarized in the following list:

- Falling into the local minima problem
- Slow convergence speed
- Expensive computational costs
- The majority of them have not been investigated for high threshold levels.

Therefore, this study investigates the performance of a newly proposed metaheuristic algorithm, known as the Kepler optimization algorithm (KOA), for segmenting the CXR images. However, the performance of the classical KOA suffers from slow convergence speed which makes it consume several function evaluations for achieving superior outcomes. Therefore, it is improved in this study using a novel improvement mechanism to propose a new variant, namely IKOA, with better exploration and exploitation operators. Both KOA and IKOA are assessed using ten CXR images at ten threshold levels (T-5, T-7, T-8, T-10, T-12, T-15, T-18, T-20, T-25, T-30), and compared to several metaheuristic algorithms in terms of several performance indicators and Wilcoxon rank-sum (WRS) test. The experimental outcomes show the superiority of IKOA over all the compared algorithms. In addition, the IKOA-based segmented CXR images at eight different threshold levels (T-3, T-4, T-5, T-7, T-8, T-10, T-12, and T-15) are used to train a newly proposed CNN model called CNN-IKOA to show the effectiveness of the image segmentation step for improving the performance of a deep learning model. This model is separately trained using the segmented CXR images at various threshold levels, and the original images and the obtained outcomes are compared in terms of five performance indicators, namely recall, overall accuracy, F1-score, precision, and specificity. According to the experimental outcomes, CNN-IKOA could perform better under the images segmented at T-12 with an overall accuracy of 94.88%, a specificity of 96.57%, a precision of 95.40%, and a recall of 95.40%. From that, it is concluded that the metaheuristic algorithms could aid in improving the classification accuracy of deep learning models for not only COVID-19 infection but also for any image classification problem. Briefly, the main contributions of this study are summarized as follows:

- Adapting the classical KOA for segmenting the COVID-19 X-ray images.
- Improving KOA using a novel improvement strategy to present a boosted variant, namely IKOA.
- Assessing KOA and IKOA using nine COVID-19 X-ray images, and comparing them to several metaheuristic algorithms under several performance indicators and the WRS test.
- The experimental findings reveal the effectiveness of IKOA over all the compared algorithms.
- Observing the performance of a newly proposed deep learning model based on CNN under the segmented images to elaborate the importance of the metaheuristic algorithms for the image segmentation problem.
- The experimental findings show that the performance of this model with the segmented images is better than that with the original images.

The remainder of this study is structured as follows: Sect. "Kepler optimization algorithm (KOA)" describes the Kepler optimization algorithm, Sect. "Convolutional neural network (CNN)" overviews the convolutional neural network (CNN), Sect. "The proposed work" introduces the proposed KOA and IKOA, in addition to the proposed deep learning model, Sect. "Results and discussion" reports results and discusses them, and Sect. "Conclusion and future work" discusses the conclusion and future prospects.

Literature review

Deep learning techniques for COVID-19 detection

Over the last few years, several machine learning and deep learning techniques have been presented to better diagnose the COVID-19 infection. Some of these techniques are reviewed in the rest of this section. In [6], a hybrid deep learning model based on integrating both convolutional neural networks (CNN) and gated recurrent unit (GRU) was presented for detecting COVID-19 from the CXR images, where CNN was utilized for extracting features, and GRU was employed as a classifier; this model was called GRU-CNN. This model achieved 96% for precision, 96% for recall, and 95% for F1-score. Sun [19] improved a metaheuristic algorithm known as the biogeography-based optimization to identify the CNN's hyperparameters for accurately detecting the COVID-19 infection from the CXR images. Wang [20] tailored a deep CNN for detecting this epidemic. Islam [21] combined long short-term memory (LSTM) with CNN for automatically identifying the COVID-19 infection from the CXR images; this model was called CNN-LSTM. This model could achieve outstanding results up to 99.4% for accuracy metric, 99.9% for AUC, 99.3% for specificity, and 98.9% for F1-score.

In [22], a CNN model based on incorporating both dilated convolution and the residual network was developed for early detection of the COVID-19 infection from the CXR images. Hussein [23] proposed two new lightweight diagnostic models based on CNN for the early and automatic diagnosis of COVID-19 individuals in CXR images. The first model was developed for the purpose of binary classification, whereas the second model was developed for multiclass classification. Also, Gupta [24] proposed a hybrid CNN for accurately detecting COVID-19 images from chest X-ray images. There are several other recently proposed deep learning models for accurately detecting the COVID-19 infection from the CXR images such as the cutting-edge CNN [25], Homomorphic Transformation and VGG-inspired deep CNN [26], multi-scale CNN [27], MobileNet-based CNN [28], and several else [29].

Metaheuristic algorithms for segmenting CXR images

Several metaheuristic algorithms have been recently proposed for tackling the CXIS problem over the last few years. For instance, the Es-MFO algorithm, developed by Sahoo [11], is an enhancement of the moth flame optimization algorithm that uses a non-linear self-adaptive parameter and a Fibonacci search approach. This variant was applied to extract the pertinent characteristics from CT images to more precisely categorize cases of COVID-19 infection. The whale optimization algorithm (WOA) was improved in [30] to present a new robust variant, namely IWOA, for solving the CXIS problem. IWOA was improved based on three folds: The first fold includes replacing the classical WOA's exploration equation with an equation responsible for randomly re-initializing the solutions within the search space; the second fold is based on changing the constant (b) and coefficient parameter (A) to boost the exploration operator in the early stages to prevent getting stuck into local minima and the exploitation operator in the latter stages to accelerate the convergence speed; and the last fold is based on using the population reduction mechanism to gradually minimize the population diversity that might aid in accelerating the convergence speed. IWOA was applied for segmenting several CXR images and compared to several rival optimizers. According to the experimental findings, it could achieve outstanding outcomes. Also, Han et al. [31] improved the performance of the multi-verse optimizer (MVO) based on the diffusion mechanism (DM) and Rosenbrock method (RM) to accurately tackle the CXIS problem. Both RM and DM were used to aid in enhancing the convergence speed and preventing stagnation into local minima.

In [32], the salp swarm algorithm (SSA) was integrated with the reptile search algorithm (RSA) to design a new version termed RSA-SSA. This version was employed for accurately tackle the CXIS problem. Su et al. [33] improved the artificial bee colony algorithm (ABC) using two mechanisms, namely vertical search and horizontal search, to develop a new robust variant for accurately solving the CXIS problem. Nama [34] integrated the slime mould algorithm (SMA) with the quasi-reflection-based learning (QRBL) and the quasi-reflection-based jumping (QRBJ) mechanisms to propose a new robust algorithm, namely QRSMA, to accurately tackle the CXIS problem. In [35], QGBWOA, an enhanced variant of WOA, was proposed. QGBWOA first employed the quasi-opposition-based learning strategy to accelerate convergence towards the near-optimal solution and then incorporated the Gaussian barebone strategy to promote population diversity. QGBWOA was applied for accurately tackling the CXIS problem. CDHGS is an enhanced variant of the hunger games search (HGS) that was introduced in [36]. CDHGS augments HGS with the dimension learning-based hunting (DLH) mechanism and crisscross optimizer (CSO). First, CSO enables individuals to share information, which accelerates convergence, while DLH is used to alleviate the algorithm's local optimum problem.

In [37], the ant colony optimization (ACO) was improved using two different mechanisms, namely directional mutation (DM) and directional crossover (DX), to present a new variant called XMACO. The DM mechanism is responsible for improving population diversity to prevent getting stuck in local optima, while the DX mechanism strives to exploit the regions around the existing individuals to accelerate convergence speed. XMACO was employed for segmenting some CXR images, and its outcomes were compared to those of several competitors to observe its efficacy. The experimental outcomes revealed the superiority of XMACO over all the compared algorithms. Zhao et al. [38] improved the classical crow search algorithm by information exchange mutation (IEM) and variable neighborhood descent (VND) mechanisms; this improved variant is named VMCSA. The IEM mechanism is responsible for improving the exploration operator to avoid stagnation into local optima, while the VND mechanism is used to improve the exploitation operator to accelerate the convergence speed. VMCSA was first assessed using CEC2014 and CEC2021 to reveal its ability to explore and exploit the search space. Then, it was applied to segment the CXR images to oberve its ability to find the optimal threshold values which could segment those images more accurately. This algorithm, according to the experimental outcomes, was better than all the compared algorithms.

Liu et al. [39] improved the ACO using two different mutation strategies, namely the greedy Levy (GL) and Cauchy; this variant was named CLACO. The GL mutation is used to boost the ACO's ability to avoid stagnation into local optima, while the Cauchy mutation is utilized to boost the exploitation operator to aid in accelerating the convergence speed. CLACO was used to solve the CXIS problem and could achieve outstanding outcomes for this problem when compared to some rival optimizers. In [40], the Harris Hawks optimization algorithm under the Otsu method was adapted for accurately segmenting the CT images. This algorithm was compared to several existing techniques to observe its efficacy for several performance indicators. The experimental results disclosed that it was able to find the optimal threshold values that could extract the most relevant features from the CT images.

Kepler optimization algorithm (KOA)

Recently, the Kepler optimization algorithm (KOA) was introduced as a new metaheuristic method to address the challenges of continuous optimization. The algorithm's design was motivated by Kepler's theories on the motion of the planets. According to these rules, the orbit of planets around the sun is subject to influence from four sources, which are represented in a planet's gravitational pull, location, mass, and orbital speed. In KOA, planets far from the sun are responsible for exploring the search space, while those closer have to settle for exploiting the promising zones for accelerating the convergence speed. The steps of KOA are presented in Algorithm 1. The KOA mathematical model is explored in depth next:

- Initialization step

At the beginning of the optimization process, the KOA will disperse N planets in the search space of the optimization process, and each planet will be composed of *d* dimensions. Following is a mathematical formulation of the formula that is used to randomly distribute those planets:

$$\overrightarrow{X}_i = \overrightarrow{X}_L + \overrightarrow{r} \cdot \left(\overrightarrow{X}_U - \overrightarrow{X}_L\right) \tag{1}$$

where $\overrightarrow{X}_i$ represents the *ith* solution/planet; $\overrightarrow{X}_U$ and $\overrightarrow{X}_L$ represent the search boundary of the tackled optimization problem; $\overrightarrow{r}$ is a uniform distribution-based randomly generated vector. In KOA, the orbital eccentricity (e) of each planet is randomly assigned as clarified in (2), and the orbital period (T) is randomly generated according to the normal distribution as defined in (3).

$$e_i = r, i = 1, \ldots, N \tag{2}$$

$$T_i = |rn|, i = 1, \ldots, N \tag{3}$$

where r is a uniform distribution-based random number in $[0, 1]$, and rn is a normal distribution-based random number.

- Defining the gravitational force

Gravity, the most fundamental force in the universe, manages the orbits of the planets around the Sun. Each planet has its own unique level of gravity that is proportional to its mass. The pull of the Sun has an effect on a planet's velocity. Planets' orbital velocities increase as they draw nearer to the Sun and decrease as they move further from the star. The force of pull between the Sun $\left(\overrightarrow{X}_S\right)$ and any planet $\left(\overrightarrow{X}_i\right)$ can be described by the universal law of gravity, as presented in the following formula:

$$\boldsymbol{F}_{\boldsymbol{g}_i}(t) = e_i \times \mu(t) \times \frac{\overline{M}_s \times \overline{m}_i}{\overline{R}^2_i + \varepsilon} + r_1, \tag{4}$$

where r_1 is a uniform distribution-based random number in [0, 1], and ε has a small value to prevent division by 0. $\overline{M}_s$ and $\overline{m}_i$ represents the normalized values of M_s and m_i, where M_s and m_i represent the masses of $\overrightarrow{X}_S$ and $\overrightarrow{X}_i$, which are computed using (7) and (8), respectively; µ is a constant representing the universal gravitational constant; and $\overline{R}_i$ is the normalized value of R_i, where R_i is the Euclidean distance between $\overrightarrow{X}_S$ and $\overrightarrow{X}_i$ and is computed using (5):

$$R_i(t) = \|\overrightarrow{X}_S(t) - \overrightarrow{X}_i(t)\|_2 = \sqrt{\sum_{i=1}^{N} \left(\overrightarrow{X}_S(t) - \overrightarrow{X}_i(t)\right)^2} \tag{5}$$

$$\overline{R}_i = \frac{R_i(t) - \min(R(t))}{\max(\mathrm{R(t)}) - \min(R(t))} \tag{6}$$

$$M_s = \frac{\mathit{fit}_s(t) - \mathit{worst}(t)}{\sum_{k=1}^{N}\left(\mathit{fit}_k(t) - \mathit{worst}(t)\right)}, \tag{7}$$

$$m_i = r_2 \frac{\mathit{fit}_i(t) - \mathit{worst}(t)}{\sum_{k=1}^{N}\left(\mathit{fit}_k(t) - \mathit{worst}(t)\right)} \tag{8}$$

where

$$\mathit{fit}_s(t) = \mathit{best}(t) = {}_{k\in\{1,2,\ldots,N\}}{}^{min}\mathit{fit}_k(t) \tag{9}$$

$$\mathit{worst}(t) = {}_{k\in\{1,2,\ldots,N\}}{}^{\max}\mathit{fit}_k(t) \tag{10}$$

where r_2 is a uniform distribution-based random number in [0, 1]. $\mu(t)$ is computed using the following equation:

$$\mu(t) = \mu_0 \times \exp\left(-\gamma \frac{t}{T_{max}}\right), \tag{11}$$

where γ is a constant; μ_0 is a predefined value (Recommended 0.1); and T_{max} and t are the maximum function evaluation and current function evaluation, respectively.

- Calculating an object's velocity

The distance a planet is from the sun is the primary factor in determining its speed. The closer a planet is to the Sun, the faster it orbits, and the slower it orbits when it is farther away from the Sun. When a planet or other object gets close to the Sun, the Sun's gravity becomes significantly greater; thus, the planet seeks to accelerate up so that it does not get sucked in by the Sun. This behavior is modeled mathematically, as shown in [41]:

$$V_i(t) = \begin{cases} l \times \left(2r_4 \overrightarrow{X}_i - \overrightarrow{X}_b\right) + \ddot{i} \times \left(\overrightarrow{X}_a - \overrightarrow{X}_b\right) + (1 - R_{i-norm}(t)) \times F \times \overrightarrow{U}_1 \times \left(\overrightarrow{X}_U - \overrightarrow{X}_L\right), & if R_{i-norm}(t) \leq 0.5 \\ r_4 \times L \times \left(\overrightarrow{X}_a - \overrightarrow{X}_i\right) + (1 - R_{i-norm}(t)) \times F \times U_2 \times \left(r_3 \overrightarrow{X}_U - \overrightarrow{X}_L\right) & Else \end{cases}, \tag{12}$$

$$l = \overrightarrow{U} \times \mathcal{M} \times \mathcal{L}, \tag{13}$$

$$\mathcal{L} = \left[\mu(t) \times (M_S + m_i)\left(\frac{2}{R_i(t) + \varepsilon} - \frac{1}{a_{i(t)} + \varepsilon}\right)\right]^{\frac{1}{2}} \tag{14}$$

$$\mathcal{M} = (r_3 \times (1 - r_4) + r_4), \tag{15}$$

$$\overrightarrow{U} = \begin{cases} 0 & \overrightarrow{r_5} \leq \overrightarrow{r_6} \\ 1 & Else \end{cases}, \tag{16}$$

$$\mathcal{F} = \begin{cases} 1, & if \ r_4 \leq 0.5 \\ -1, & Else \end{cases}, \tag{17}$$

$$\ddot{l} = \left(1 - \overrightarrow{U}\right) \times \overrightarrow{\mathcal{M}} \times \mathcal{L} \tag{18}$$

$$\overrightarrow{\mathcal{M}} = \left(r_3 \times \left(1 - \overrightarrow{r_5}\right) + \overrightarrow{r_5}\right), \tag{19}$$

$$\overrightarrow{U}_1 = \begin{cases} 0 & \overrightarrow{r_5} \leq r_4 \\ 1 & Else \end{cases}, \tag{20}$$

$$U_2 = \begin{cases} 0 & r_3 \leq r_4 \\ 1 & Else \end{cases}, \tag{21}$$

where $\overrightarrow{V_i}(t)$ refers to the velocity of the *ith* object; r_3 and r_4 are two numerical values chosen at random between 0 and 1 according to the uniform distribution; and $\overrightarrow{r_5}$ and $\overrightarrow{r_6}$ represent two vectors assigned numerical values generated at random between 0 and 1 according to the uniform distribution;$\overrightarrow{X}_a$ and $\overrightarrow{X}_b$ represent two planets/solutions chosen randomly from the current solutions; $\mathcal{F}$ is a controlling factor to reflect the search direction, this factor selects randomly 1 or -1; a_i is computed as follows:

$$a_i(t) = r_3 \times \left[T_i^2 \times \frac{\mu(t) \times (M_s + m_i)}{4\pi^2}\right]^{\frac{1}{3}}, \tag{22}$$

- Escaping from the local optimum

The majority of the planets in the solar system revolve on their own axes and orbit the Sun in an anticlockwise direction, whereas the other planets orbit the Sun in a clockwise direction. This behavior is utilized by KOA in order to escape from local optimal zones. This is accomplished by changing the direction of the search at predetermined intervals with the assistance of a flag designated as $\mathcal{F}$. Because of this, the agents have a greater possibility of searching the entire search space effectively.

- Updating objects' positions

The following formula can be used to estimate the new positions of the planets in KOA:

$$\overrightarrow{X}_i(t+1) = \overrightarrow{X}_i(t) + \mathcal{F} \times \overrightarrow{V}_i(t) + \left(\boldsymbol{F}_{\boldsymbol{g}_i}(t) + |r|\right) \times \overrightarrow{U} \times \left(\overrightarrow{X}_S(t) - \overrightarrow{X}_i(t)\right), \quad (23)$$

- Updating distance with the sun

The naturally occurring change in the distance to the Sun and the planets is recreated as part of an effort to enhance the KOA's exploration and exploitation capabilities. When planets are in close proximity to the Sun, KOA will give preference to the exploitation operator, whereas when the Sun is farther distant, KOA will give preference to the exploration operator. To apply this idea, in KOA, a time-dependent variation in the value of the controlling parameter *h* is used. When this value is large, KOA activates the exploration operator to broaden the search space for a better solution, while when it is small, KOA activates the exploitation operator to extract the most value from the regions close to the best solution obtained even now. The mathematical model for this principle can be expressed in terms of the following equations:

$$\begin{aligned}\overrightarrow{X}_i(t+1) =& \overrightarrow{X}_i(t) \times \overrightarrow{U}_1 + \left(1 - \overrightarrow{U}_1\right) \\ & \times \left(\frac{\overrightarrow{X}_i(t) + \overrightarrow{X}_S + \overrightarrow{X}_a(t)}{3.0} + h \times \left(\frac{\overrightarrow{X}_i(t) + \overrightarrow{X}_S + \overrightarrow{X}_a(t)}{3.0} - \overrightarrow{X}_b(t)\right)\right),\end{aligned} \quad (24)$$

$$h = \frac{1}{e^{\eta r}} \quad (25)$$

$$\eta = (a_2 - 1) \times r_4 + 1, \quad (26)$$

$$a_2 = -1 - \left(\frac{t\% \frac{T_{max}}{TC}}{\frac{T_{max}}{TC}}\right) \quad (27)$$

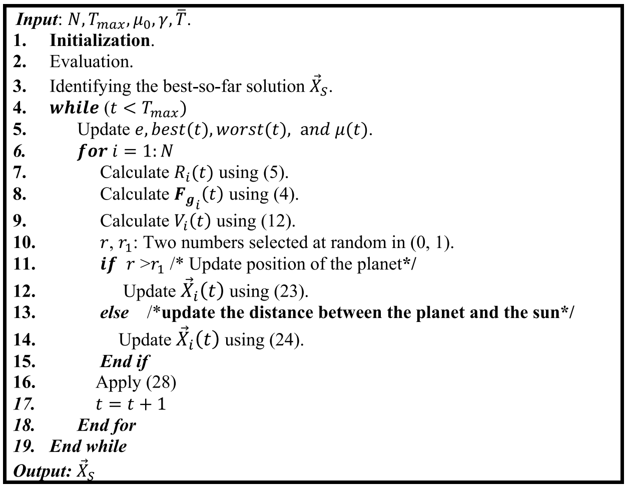
where *r* is a value chosen at random based on the normal distribution, where *TC* refers to the cycle's number, and % indicates the remainder operator.

- Elitism

This step is used to ensure that the planets are always in their current local-best positions, as defined by the accompanying mathematical formula:

$$\overrightarrow{X}_i(t+1) = \begin{cases} \overrightarrow{X}_i(t+1), & iff(\overrightarrow{X}_i(t+1) \le f(\overrightarrow{X}_i(t)) \\ \overrightarrow{X}_i(t) & Else \end{cases}, \quad (28)$$

Algorithm 1 The steps of KOA.



```
Input: $N, T_{max}, \mu_0, \gamma, \bar{T}$.
Initialization.
Evaluation.
Identifying the best-so-far solution $\vec{X}_S$.
while $(t < T_{max})$
    Update $e, best(t), worst(t),$ and $\mu(t)$.
    for $i = 1{:}N$
        Calculate $R_i(t)$ using (5).
        Calculate $\boldsymbol{F}_{g_i}(t)$ using (4).
        Calculate $V_i(t)$ using (12).
        $r, r_1$: Two numbers selected at random in (0, 1).
        if $r > r_1$ /* Update position of the planet*/
            Update $\vec{X}_i(t)$ using (23).
        else /*update the distance between the planet and the sun*/
            Update $\vec{X}_i(t)$ using (24).
        End if
        Apply (28)
        $t = t + 1$
    End for
End while
Output: $\vec{X}_S$
```

Convolutional neural network (CNN)

CNN is a well-common deep learning model that is based on mimicking the visual perception mechanism of human beings [42]. Typically, the CNN architecture is composed of three building blocks: pooling layer, Convolutional layer, and fully connected (FC) layer, as depicted in Fig. 1 [42, 43]. This layer is considered the main component for any CNN model, where it includes a set of filters, also referred to as convolutional kernels. Those kernels are used to generate the feature maps from the input tensors. This layer includes two hyperparameters, namely kernel size and number of filters, which have to be accurately estimated to maximize the performance of CNN. The convolutional layer is followed by a pooling layer to decrease the large size of the generated feature maps to decrease the number of trainable parameters. There are several pooling techniques that could be used in the pooling layer, some of which are min pooling, gated pooling, max pooling, average pooling, and tree pooling [44]. The output feature maps of the last convolution or pooling layer are typically transformed into a vector, and linked to one or more FC layers, also referred to

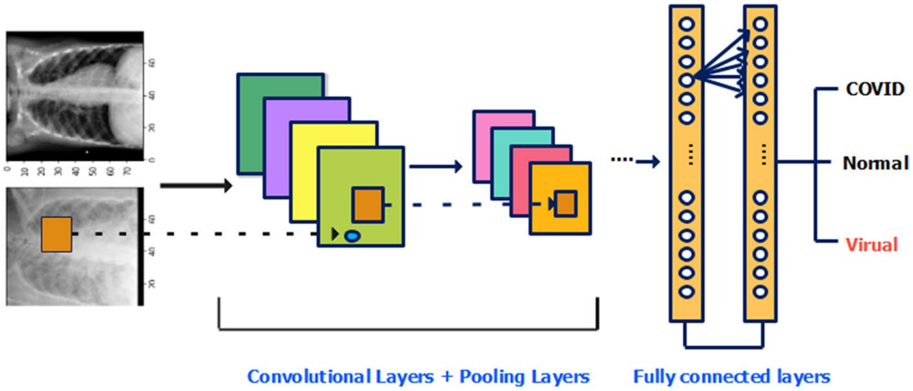


Fig. 1 The structure of a CNN model

as dense layers, for estimating the final output. The last FC layer is typically composed of a number of output nodes equal to the number of classes.

The proposed work

This section discusses the main steps for adapting the metaheuristic algorithms to tackle the CXIS problem. These steps are initialization, objective function, the pseudocode of the proposed KOA, the proposed improvement mechanism, and the pseudocode of the proposed IKOA. As aforementioned, image segmentation significantly affects the performance of the deep learning models for better-classifying images. In brief, some of the advantages of image segmentation for deep learning are described in the following list:

- It can alleviate the noise and irrelevant details in the images, making them easier to process and analyze by the deep learning models.
- It can highlight the important features and regions of interest in the images that can aid the deep learning models in learning more effectively and accurately.

Therefore, at the end of this section, we design a new deep-learning model based on CNN to check the effectiveness of the CXR images segmented by the proposed IKOA at different threshold levels.

Initialization

The classical KOA begins with generating two-dimensional matrix of $N \times d$, where N represents the population size, and d represents the dimension size or threshold level. This matrix is randomly initialized within the lower bound and upper bound of the pixels in a grey image, as mathematically defined in the following formula:

$$\overrightarrow{X}_i = \overrightarrow{X}_L + \overrightarrow{r} \cdot \left(\overrightarrow{X}_U - \overrightarrow{X}_L\right) \tag{29}$$

where $\overrightarrow{X}_L$ and $\overrightarrow{X}_L$ includes values of 0 and 255, respectively; those values represent the smallest and highest intensity level for each pixel in a grey image. The intensity level for each pixel must include an integer ranging between 0 and 255, while the classical KOA generates continuous solutions, which are irrelevant to this problem. Therefore, those solutions are mapped into integers by truncating the fractional part separated by a decimal dot. Then, each solution is ordered ascendingly and evaluated using Otsu's method discussed in the next section.

Objective function: Otsu's method

Otsu [45] proposed the conventional Otsu method as a variance-based strategy for finding threshold values that maximize the between-class variance, or, opposite, minimize the intra-class intensity variance, in order to find the homogeneous regions from an image. This method is mathematically described as follows:

$$f\left(h_0, h_1, h_2, \ldots\ldots.., h_{d+1}\right) = {\sigma_0}^2 + {\sigma_1}^2 + {\sigma_2}^2 + \ldots\ldots.. + {\sigma_d}^2 \tag{30}$$

where

$$\sigma_0{}^2 = \omega_0(\mu_0 - \mu_{d+1})^2, \omega_0 = \sum\nolimits_{i=h_0}^{h_1-1} p_i, \mu_0 = \sum\nolimits_{i=h_0}^{h_1-1} \frac{ip_i}{\omega_0} \tag{31}$$

$$\sigma_1{}^2 = \omega_1(\mu_1 - \mu_{d+1})^2, \omega_1 = \sum\nolimits_{i=h_1}^{h_2-1} p_i, \mu_1 = \sum\nolimits_{i=h_1}^{h_2-1} \frac{ip_i}{\omega_1} \tag{32}$$

$$\sigma_2{}^2 = \omega_2(\mu_2 - \mu_{d+1})^2, \omega_2 = \sum\nolimits_{i=h_2}^{h_3-1} p_i, \mu_2 = \sum\nolimits_{i=h_2}^{h_3-1} \frac{ip_i}{\omega_2} \tag{33}$$

$$\sigma_d{}^2 = \omega_d(\mu_d - \mu_{d+1})^2, \omega_d = \sum\nolimits_{i=h_d}^{h_{d+1}} p_i, \mu_d = \sum\nolimits_{i=h_d}^{h_{d+1}} \frac{ip_i}{\omega_d} \tag{34}$$

$$p_i = \frac{P_i}{W} \tag{35}$$

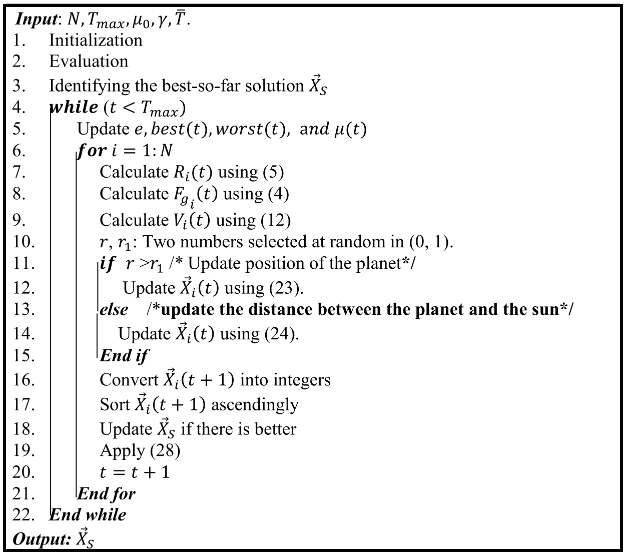
$$\mu_{d+1} = \sum\nolimits_{i=0}^{d+1} ip_i \tag{36}$$

where d represents the threshold level; $h_0, h_1, h_2, ..., h_{d+1}$ represent the estimated threshold values in ascending order, which are responsible for separating $d + 1$ homogenous regions from an image; $\sigma_0{}^2, \sigma_1{}^2, \sigma_2{}^2, \ldots\ldots.., and \sigma_T{}^2$ stands for the variances of the $d + 1$ homogenous regions; $\omega_0, \omega_1, \omega_2, \ldots\ldots.., \omega_T$ represent the probabilities of the homogenous regions; $\mu_0, \mu_1, \mu_2, \ldots\ldots.., \mu_T$ represent the means of the homogenous regions; h_0 and h_{d+1} in (30) include the minimum and maximum intensity levels in the greyscale, these levels are 0 and 255, respectively. Based on that, $[h_1, h_2, \ldots\ldots, h_T]$ are the threshold values that need to be accurately estimated for separating the homogenous regions in the given image. P_i represent the number of pixels with the intensity level i. W is the total number of pixels in the given image.

COVID-19 X-ray image segmentation using KOA

The CXIS problem is discrete and therefore cannot be directly solved by the classical KOA. Therefore, in this section, we expand on how to apply the classical KOA to address this problem. KOA begins by randomly dispersing $\boldsymbol{N}$ solutions between the maximum and minimum intensities for each pixel using (29). Those initial solutions are converted into integers and sorted ascendingly to become relevant to this problem. Then, they are evaluated using (30) and compared to identify the best-so-far solution. Finally, the optimization process of KOA is executed to update those solutions to search for better solutions. This process is continued until the maximum number of function evaluations is satisfied. In brief, algorithm 2 presents the pseudocode for the proposed KOA used to segment the CXR images.

Algorithm 2 KOA for CXR image segmentation



```
Input: $N, T_{max}, \mu_0, \gamma, \bar{T}$.
Initialization
Evaluation
Identifying the best-so-far solution $\vec{X}_S$
while $(t < T_{max})$
    Update $e, best(t), worst(t),$ and $\mu(t)$
    for $i = 1{:}N$
        Calculate $R_i(t)$ using (5)
        Calculate $F_{g_i}(t)$ using (4)
        Calculate $V_i(t)$ using (12)
        $r, r_1$: Two numbers selected at random in (0, 1).
        if $r > r_1$ /* Update position of the planet*/
            Update $\vec{X}_i(t)$ using (23).
        else /*update the distance between the planet and the sun*/
            Update $\vec{X}_i(t)$ using (24).
        End if
        Convert $\vec{X}_i(t+1)$ into integers
        Sort $\vec{X}_i(t+1)$ ascendingly
        Update $\vec{X}_S$ if there is better
        Apply (28)
        $t = t + 1$
    End for
End while
Output: $\vec{X}_S$
```

Exploration and exploitation improvement mechanism

To further enhance the exploitation and exploration operators of KOA, it is integrated with a new updating mechanism, namely the exploration and exploitation improvement (EEI) mechanism, to explore the regions around the mean solution of the current solution and a solution picked randomly from the current population. This could improve the KOA's exploration operator. Regarding the exploitation operator, this scheme sometimes strives to update this mean solution toward the best solution obtained so far for exploiting the regions around it in the hope of accelerating the convergence speed. This scheme is mathematically described in the following equation:

$$\overrightarrow{X}_i(t+1) = \frac{\left(\overrightarrow{X}_i(t) + \overrightarrow{X}_b(t)\right)}{2.0} + r * \overrightarrow{v}_1 + r_1 * \left(\overrightarrow{X}^* - \overrightarrow{X}_i(t)\right) \tag{37}$$

where $\boldsymbol{r}$ and $\boldsymbol{r}_1$ stand for two random numbers in [0, 1]. $\overrightarrow{\boldsymbol{v}}_1$ is given by the following equation:

$$\overrightarrow{v}_1 = \begin{cases} \overrightarrow{X}_a(t) - \overrightarrow{X}_i(t) & f\left(\overrightarrow{X}_a(t)\right) > f(\overrightarrow{X}_i(t)) \\ \overrightarrow{X}_i(t) - \overrightarrow{X}_a(t) & otherwise \end{cases}, \tag{38}$$

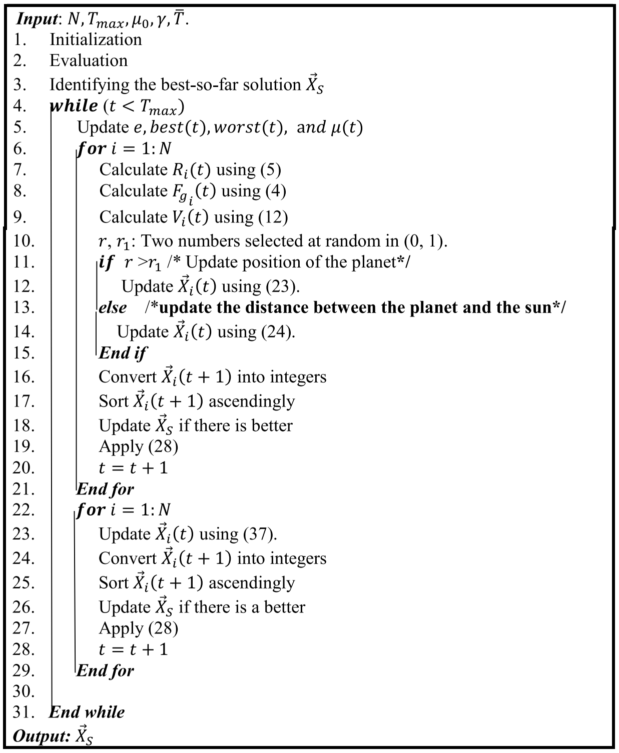
where $\boldsymbol{f}(\cdot)$ represents the objective function, and $\overrightarrow{\boldsymbol{X}}_{\boldsymbol{a}}(\boldsymbol{t})$ represents a solution picked randomly from the current solutions.

This mechanism is integrated with the classical KOA to present an improved variant of KOA, namely IKOA. This variant has the following advantages that make it a strong alternative for tackling the CXIS problem:

- Easy to implement
- Consuming low computational cost
- Having a high convergence speed
- Having a high ability to avoid stagnation into local minima due to the EEI mechanism

Meanwhile, the main disadvantage of the proposed KOA and IKOA is that they have three control parameters ($\boldsymbol{\mu}_0, \boldsymbol{\gamma}, \overline{\boldsymbol{T}}$) that need a lot of effort to be accurately estimated before starting the optimization process for maximizing their performance. Finally, Algorithm 3 presents the pseudocode for the proposed IKOA used to segment the CXR images.

Algorithm 3 IKOA for CXR image segmentation



```
Input: $N, T_{max}, \mu_0, \gamma, \bar{T}$.
Initialization
Evaluation
Identifying the best-so-far solution $\vec{X}_S$
while ($t < T_{max}$)
    Update $e, best(t), worst(t),$ and $\mu(t)$
    for $i = 1{:}N$
        Calculate $R_i(t)$ using (5)
        Calculate $F_{g_i}(t)$ using (4)
        Calculate $V_i(t)$ using (12)
        $r, r_1$: Two numbers selected at random in (0, 1).
        if $r > r_1$ /* Update position of the planet*/
            Update $\vec{X}_i(t)$ using (23).
        else /*update the distance between the planet and the sun*/
            Update $\vec{X}_i(t)$ using (24).
        End if
        Convert $\vec{X}_i(t + 1)$ into integers
        Sort $\vec{X}_i(t + 1)$ ascendingly
        Update $\vec{X}_S$ if there is better
        Apply (28)
        $t = t + 1$
    End for
    for $i = 1{:}N$
        Update $\vec{X}_i(t)$ using (37).
        Convert $\vec{X}_i(t + 1)$ into integers
        Sort $\vec{X}_i(t + 1)$ ascendingly
        Update $\vec{X}_S$ if there is a better
        Apply (28)
        $t = t + 1$
    End for

End while
Output: $\vec{X}_S$
```

The proposed deep learning model: CNN-IKOA

Image segmentation can be utilized as a preprocessing step for enhancing the performance of deep learning models that work with images, such as scene understanding, object detection, or medical image analysis. Therefore, in this study, we employ IKOA to perform this step for the following purposes:

- Observing whether the image segmentation could improve the classification accuracy of deep learning or not.

- Showing the effectiveness of the proposed IKOA for aiding deep learning techniques in classifying the COVID-19 infection more effectively.

In a more sense, the proposed IKOA is first employed to segment the CXR images at different threshold levels. The segmented CXR images at each threshold level are used to train and test a newly proposed CNN model, namely CNN-IKOA. This model is based on three convolutional layers that are used to extract more complex and abstract features from the segmented images fed into it. Each convolutional layer employs a number of filters in order to extract information from the input images, such as edges and corners. The number and size of filters in each layer are considered hyperparameters that are tuned in the experiments section to maximize the performance of the proposed model. These filters are responsible for the extraction of feature maps, which provide information regarding the location and existence of specific patterns within the input images. The proposed CNN-IKOA stacks three convolutional layers, which allow it to learn to detect higher-level features that are made of lower-level features, thereby detecting COVID-19 infection more accurately. In CNN-IKOA, each convolutional layer is followed by a max pooling layer to alleviate the spatial size of the representation to reduce the number of parameters and computational costs required by the model. The ReLU activation function is utilized with the convolutional layers to prevent the vanishing gradient problem. The feature map from the last max pooling layer is converted into a one-dimensional matrix using the flattening layer. This matrix is input to an FC-connected layer, which is followed by a dropout layer to avoid the overfitting problem. The output from the dropout layer is input to the output layer for detecting if the input chest image is infected with COVID-19 or not based on the softmax activation function. This function computes the probability for each possible class according to (39) and the class that has the highest probability is considered.

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}}, \tag{39}$$

where C represents the number of possible classes, x is the input vector to the softmax function, and x_i represents ith element of the input vector. The structure of the proposed

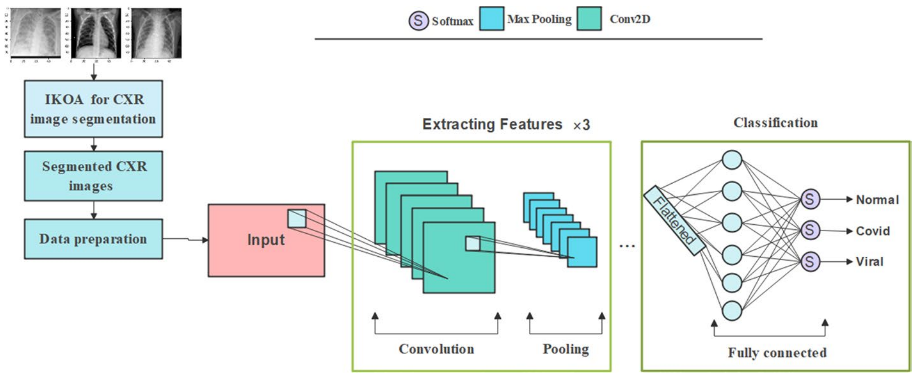


Fig. 2 The proposed deep learning model: IKOA-CNN

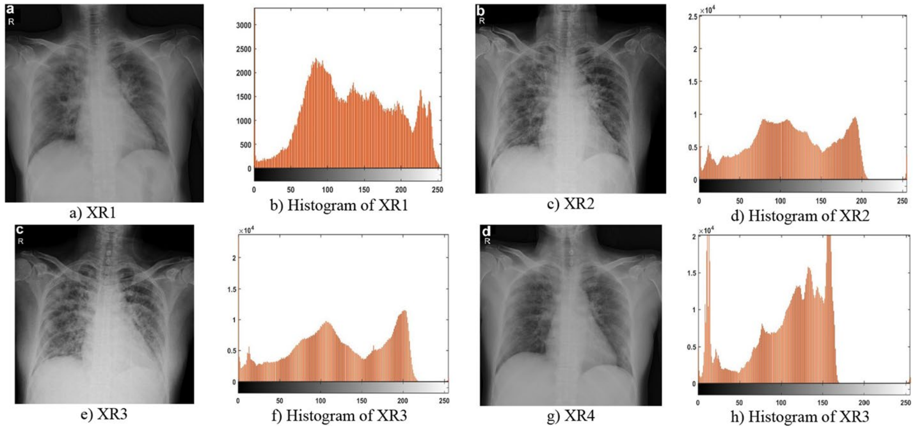
CNN-IKOA is depicted in Fig. 2. The proposed CNN-IKOA has eleven hyperparameters that have to be accurately estimated to improve its performance when applied to detect COVID-19 infection. Those hyperparameters are summarized in the following list:

- Since three convolutional layers are used and each layer uses a number and size of filters, six different parameters need to be accurately estimated.
- The pooling layer includes a hyperparameter known as pooling size that needs to be accurately estimated. Since CNN-IKOA uses three pooling layers, three additional hyperparameters need to be tuned.
- The probability of the dropout layer is considered an additional hyperparameter that also needs to be accurately estimated.
- The last hyperparameter that needs to be tuned is the number of nodes in the FC layer.

The best values for those hyperparameters are discussed in detail in the experiments section. The main disadvantage of the proposed CNN-IKOA is that they have fourteen hyperparameters that need a lot of computation to be accurately estimated before starting the classification process to maximize its performance.

Results and discussion

In this section, first, the proposed algorithms (KOA and IKOA) are assessed by applying them to segmenting ten COVID-19 X-ray images for ten threshold levels (T-5, T-7, T-8, T-10, T-12, T-15, T-18, T-20, T-25, and T-30). These images are taken from [46] based on their various histograms that enable observing the stability of the proposed algorithms. Figure 3 presents some of those images with their histogram. The performance of KOA and IKOA for considered threshold levels over the used images are compared to that of seven well-known optimization techniques for several performance indicators, such as average fitness value (AFV), Friedman mean rank (F-rank), Convergence curve, Computational cost, PSNR [47], and FSIM [47]. Those compared algorithms are named Modified SSA (MSSA) [48], Constriction coefficient based gravitational search algorithm and particle swarm optimization (CPSOGSA) [49], SSA [48], Nutcracker optimization



a) XR1

b) Histogram of XR1

c) XR2

d) Histogram of XR2

e) XR3

f) Histogram of XR3

g) XR4

h) Histogram of XR3

Fig. 3 Some of the CXR images with their histogram

algorithm (NOA) [50], Equilibrium optimizer (EO) [51], Teaching–learning–based optimization (TLBO) [52], and Differential evolution [53]. The controlling parameters of those algorithms are set as suggested in the cited references, with the exception of the maximum number of function evaluations and population size, which are set to 25 and 30, respectively, to guarantee a fair comparison. Those algorithms are run on a device with 32GB of RAM, an Intel® CoreTM i3-2330M CPU at 2.20 GHz, and a 64-bit operating system, and are implemented in MATLAB2019a.

Second, the proposed IKOA is applied to segment the CXR dataset for eight threshold levels (T-3, T-4, T-5, T-7, T-8, T-10, T-12, and T-15). This dataset contains 536 COVID-19 images, 619 images of viral pneumonia, and 668 normal images [54]. The COVID-19 cases included in this dataset have ages ranging from 18 to 75 years old. Afterwards, the segmented images are divided into training and testing datasets with a probability of 80% for the training dataset and 20% for the testing dataset. The proposed CNN-IKOA is trained using the training dataset and validated using the testing dataset to observe its performance for classifying unknown images. The effectiveness of CNN-IKOA under various threshold levels is observed using several performance indicators, such as recall, accuracy, F1-score, precision, and specificity, which are mathematically described as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{40}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{41}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{42}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{43}$$

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recal}} \tag{44}$$

where TP, FP, TN, and FN refer to true positive, false positive, true negative, respectively, and false negative, respectively.

Hyperparameters tuning

As aforementioned, CNN has some hyperparameters that need to be accurately identified to maximize its classification accuracy. Those parameters are the number of filters and kernel size in each convolutional layer, and the pool size in the pooling layer. The proposed CNN-IKOA has three pooling layers and three convolutional layers, so nine parameters need to be estimated. However, the pooling size has been recommended to be set to 2 in several studies in the literature. Therefore, this recommendation is considered in our proposed model. Based on that, the proposed model has only six unknown parameters. To estimate the most effective values for these parameters, several

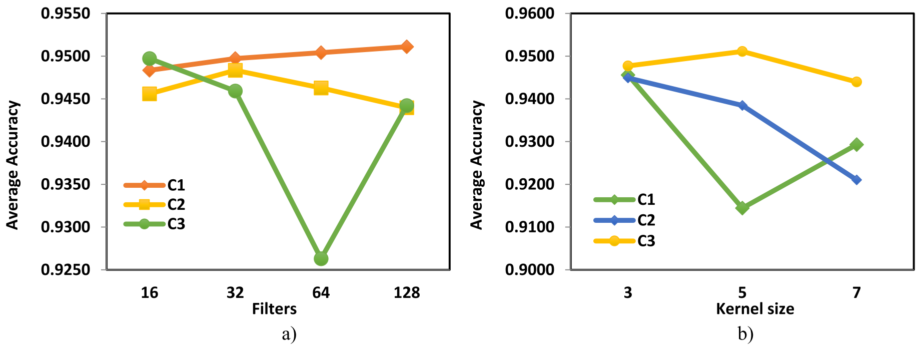


Fig. 4 Tuning the number of filters and kernel size of the proposed CNN-IKOA

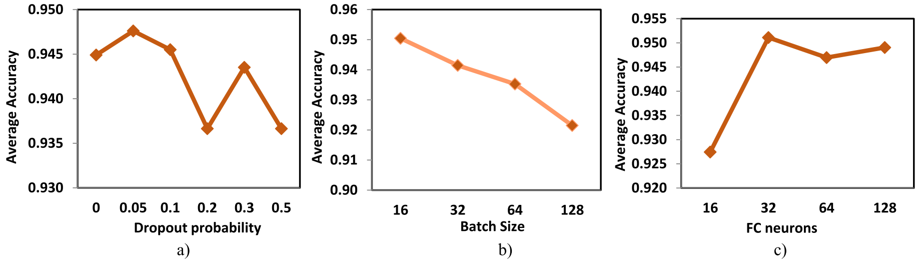


Fig. 5 Tuning the dropout probability, batch size, and FC neurons of the proposed CNN-IKOA

experiments are done using various values for each parameter, and the classification accuracy of CNN-IKOA under each value is calculated and depicted in Fig. 4. This figure discloses that CNN-IKOA could reach strong performance when, respectively, setting the number of filters and kernel size for the first convolutional layer (C1) to 64 and 3, the second convolutional layer (C2) to 32 and 3, and the third convolutional layer (C3) to 16 and 5. Regarding the other hyperparameters, CNN-IKOA could perform better when setting the batch size to 16, the number of nodes in the FC layer to 32, and the dropout probability for the FC layer to 0.05, as reported in Fig. 5.

Experiment 1: comparison between IKOA and rival optimizers

In this section, IKOA and KOA are compared to seven rival optimizers in terms of AFV, PSNR, and FSIM for all CXR images at each threshold level to test and verify the effectiveness of their optimization process in addition to the quality of their segmented images. All algorithms are independently executed 20 times under the same number of function evaluations and population size to achieve a fair comparison.

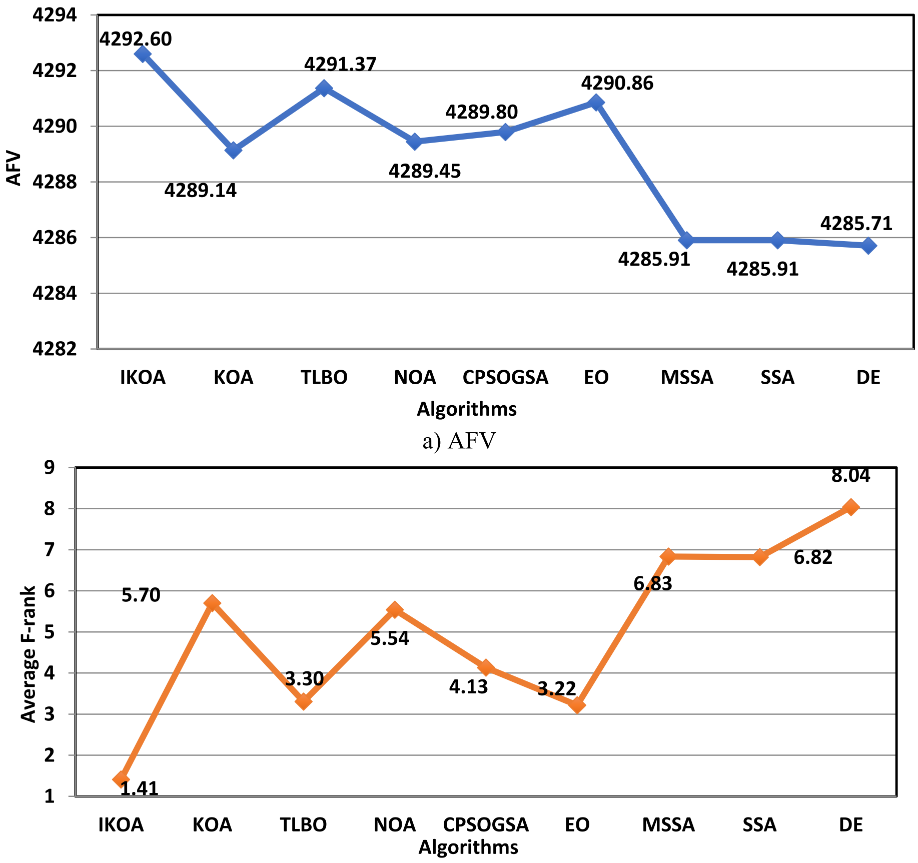
Comparison in terms of fitness value

Each algorithm is run 20 times independently on each CXR image at each threshold level. Then, The AFV and average F-rank for each threshold level on all CXR images are calculated and provided in Table 1. This table demonstrates the superior performance of IKOA in comparison to all compared algorithms, as it was able to outperform the traditional KOA as well as all competing algorithms at every threshold level. TLBO is regarded as the second-best algorithm due to its ability to outperform all other algorithms for the majority of threshold levels, while DE is considered the worst algorithm.

Table 1 Comparison in terms of AFV and F-rank

T	Metrics	IKOA	KOA	TLBO	NOA	CPSOGSA	EO	MSSA	SSA	DE
T-5	Avg	**4229.7638**	4226.5791	4229.3189	4228.6160	4224.8948	4226.9364	4216.2340	4216.2536	4223.3557
	F-rank	**1.4969**	5.8063	3.2125	4.1313	4.5875	3.4219	7.7313	7.5625	7.0500
T-7	Avg	**4268.1657**	4263.7601	4267.2701	4265.2285	4264.7933	4265.6541	4257.4516	4256.6403	4259.1028
	F-rank	**1.3469**	5.6625	3.0875	4.7500	3.9906	3.7688	7.3188	7.6563	7.4188
T-8	Avg	**4279.2207**	4274.5392	4277.9336	4275.2364	4274.2339	4276.5149	4268.1974	4268.7154	4270.0817
	F-rank	**1.4063**	5.2813	2.9063	4.8375	4.7500	3.5875	7.5313	7.4563	7.2438
T-10	Avg	**4292.2843**	4287.5362	4290.7073	4288.0166	4288.3161	4289.7509	4283.0549	4283.5733	4282.7418
	F-rank	**1.3375**	5.3500	2.9563	5.0125	4.4875	3.6000	7.2875	7.2000	7.7688
T-12	Avg	**4299.6966**	4295.3237	4297.9845	4295.3644	4296.4727	4297.7104	4292.8860	4292.3665	4291.2266
	F-rank	**1.3688**	5.5375	2.9938	5.2625	4.4250	3.3625	6.8813	7.2000	7.9688
T-15	Avg	**4305.9767**	4302.4406	4304.5231	4302.0356	4303.5794	4304.5416	4300.7905	4300.8014	4298.9576
	F-rank	**1.5500**	5.4313	3.2188	5.8125	4.0438	3.0688	6.9250	6.7813	8.1688
T-18	Avg	**4309.7492**	4306.7939	4308.3379	4306.1951	4307.9529	4308.5329	4305.9748	4305.9825	4303.6955
	F-rank	**1.4313**	5.4813	3.2688	6.2563	3.7875	3.0563	6.6688	6.5125	8.5375
T-20	Avg	**4311.4020**	4308.4108	4310.0091	4308.4994	4309.8222	4310.2731	4308.2546	4308.4192	4305.8823
	F-rank	**1.3375**	6.0938	3.3938	5.9438	3.8938	3.1250	6.3313	6.2250	8.6563
T-25	Avg	**4314.0941**	4312.0124	4312.9723	4311.6292	4313.0574	4313.4596	4312.1010	4312.1116	4309.8257
	F-rank	**1.4125**	6.0875	3.8438	6.6625	3.6500	2.7125	5.9000	5.9688	8.7625
T-30	Avg	**4315.6474**	4313.9702	4314.6669	4313.6777	4314.8369	4315.2044	4314.1508	4314.2212	4312.2543
	F-rank	**1.4188**	6.2688	4.1625	6.7375	3.7250	2.4813	5.7625	5.6375	8.8063

Bold value refers to the best finding



a) AFV

b) Average F-rank

Fig. 6 Comparison in terms of the average of AFV and F-rank

Figure 6 is presented to demonstrate the average of the AFV and F-rank values reported in Table 1 for each method. Based on this figure, we can deduce that IKOA performs the best among all of the other algorithms, followed by EO, while DE is the method with the worst results.

Evaluation of the segmented image quality

After demonstrating that IKOA is superior for the fitness values, this section is offered to examine the segmented images' quality in comparison to the source images using two indicators, namely PSNR and FSIM. The FSIM metric measures the feature similarity of the images, while the PSNR computes the error percentage between the original and segmented images. All algorithms are independently executed 20 runs, and the best solution returned after completing the optimization process is used to generate the segmented image. This image is then compared to the original image based on both PSNR and SSIM. Table 2 reports the average PSNR and FSIM on all test images segmented at each threshold level. According to this table, IKOA is capable of producing better FSIM values for all threshold levels, except for T-18 and T-20, where EO could achieve better FSIM values for them. This table also shows the superiority of IKOA for the PSNR metric at all threshold levels. In order to provide a comprehensive

Table 2 Comparison in terms of PSNR and FSIM metrics

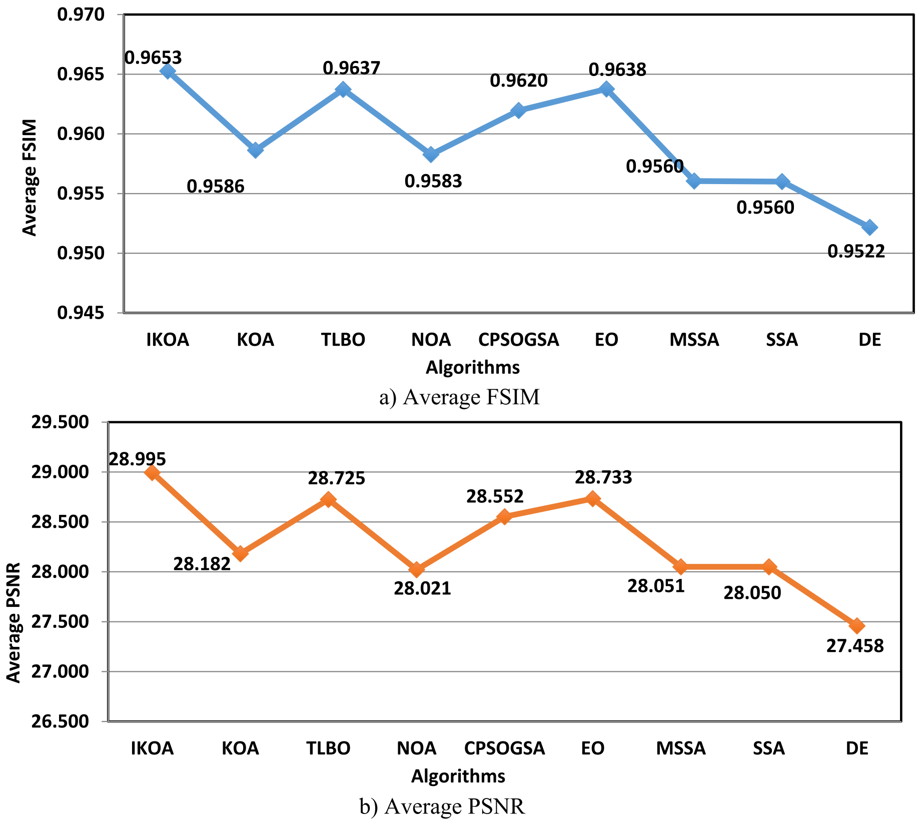
T	IKOA	KOA	TLBO	NOA	CPSOGSA	EO	MSSA	SSA	DE
FSIM metric									
T-5	**0.9019**	0.8978	0.9017	0.9008	0.8941	0.8973	0.8852	0.8849	0.8941
T-7	0.9347	0.9282	**0.9350**	0.9314	0.9308	0.9310	0.9194	0.9184	0.9218
T-8	**0.9463**	0.9392	0.9445	0.9399	0.9400	0.9431	0.9301	0.9299	0.9334
T-10	**0.9620**	0.9527	0.9596	0.9545	0.9572	0.9593	0.9475	0.9476	0.9445
T-12	**0.9715**	0.9620	0.9685	0.9622	0.9680	0.9706	0.9591	0.9587	0.9536
T-15	**0.9793**	0.9710	0.9770	0.9693	0.9762	0.9788	0.9718	0.9728	0.9652
T-18	0.9839	0.9779	0.9830	0.9752	0.9833	**0.9844**	0.9808	0.9804	0.9709
T-20	0.9866	0.9799	0.9851	0.9790	0.9860	**0.9878**	0.9842	0.9844	0.9738
T-25	**0.9920**	0.9869	0.9901	0.9826	0.9906	0.9911	0.9894	0.9900	0.9800
T-30	**0.9944**	0.9906	0.9928	0.9879	0.9933	0.9942	0.9930	0.9930	0.9843
PSNR metric									
T-5	**21.8829**	21.7525	21.8384	21.8396	21.5796	21.6041	21.1349	21.1479	21.5699
T-7	**24.3285**	23.9740	24.2983	24.0920	24.0531	24.0755	23.5550	23.4417	23.6319
T-8	**25.2566**	24.8445	25.1949	24.8890	24.8869	25.0859	24.3995	24.3853	24.4867
T-10	**26.9283**	26.2704	26.7954	26.4204	26.5411	26.7188	25.9164	25.9121	25.7407
T-12	**28.2530**	27.4573	28.0730	27.5082	27.8740	28.0575	27.2579	27.2046	26.8615
T-15	**29.8682**	28.9285	29.6784	28.8450	29.4137	29.6845	28.9024	28.9886	28.3238
T-18	**31.3447**	30.3176	31.0655	29.9693	30.9277	31.1303	30.4275	30.3859	29.3979
T-20	**32.2371**	31.0429	31.7904	30.7317	31.7590	31.9759	31.2544	31.2630	30.0513
T-25	**34.1783**	32.9337	33.5885	32.1561	33.5047	33.7189	33.0934	33.1385	31.6174
T-30	**35.6710**	34.2938	34.9265	33.7634	34.9790	35.2779	34.5685	34.6315	32.8990

Bold value represents the best finding

illustration of the quality of the images segmented by each algorithm, Fig. 7 is presented to compute the average of the PSNR and FSIM values presented in Table 2. Based on this figure, it is concluded that IKOA is the algorithm with the best performance because it was able to provide a value of 0.9653 for FSIM and a value of 28.995 for SSIM. In addition, the data presented in this figure demonstrates that EO is the algorithm with the second-best performance, with average values of 0.9638 and 28.733 for FSIM and PSNR, respectively. As a result, we can conclude that IKOA is an excellent alternative to segmenting the CXR images for the purpose of rapid interpretation to help in the accurate and speedy detection of the COVID-19 infection.

Statistical analysis: WRS test

The WRS test [55] is used to reveal the differences between the outcomes achieved by IKOA and those achieved by each competitor for each threshold image across all CXR images. This statistical test gives a p-value as a result of comparing each pair of algorithms. After that, this value is contrasted to a significance level of 5%; if it is lower, the alternative hypothesis is accepted; otherwise, the null hypothesis is accepted. The average p-value achieved by IKOA and each rival algorithm for each threshold level across all test images is shown in Table 3. This table elaborates that there are considerable differences between the outcomes of IKOA and those of the rival algorithms for all threshold levels. From that, it is concluded that IKOA is noticeably distinct from all of its competitors at all threshold levels.



a) Average FSIM

b) Average PSNR

Fig. 7 Comparison in terms of average FSIM and PSNR

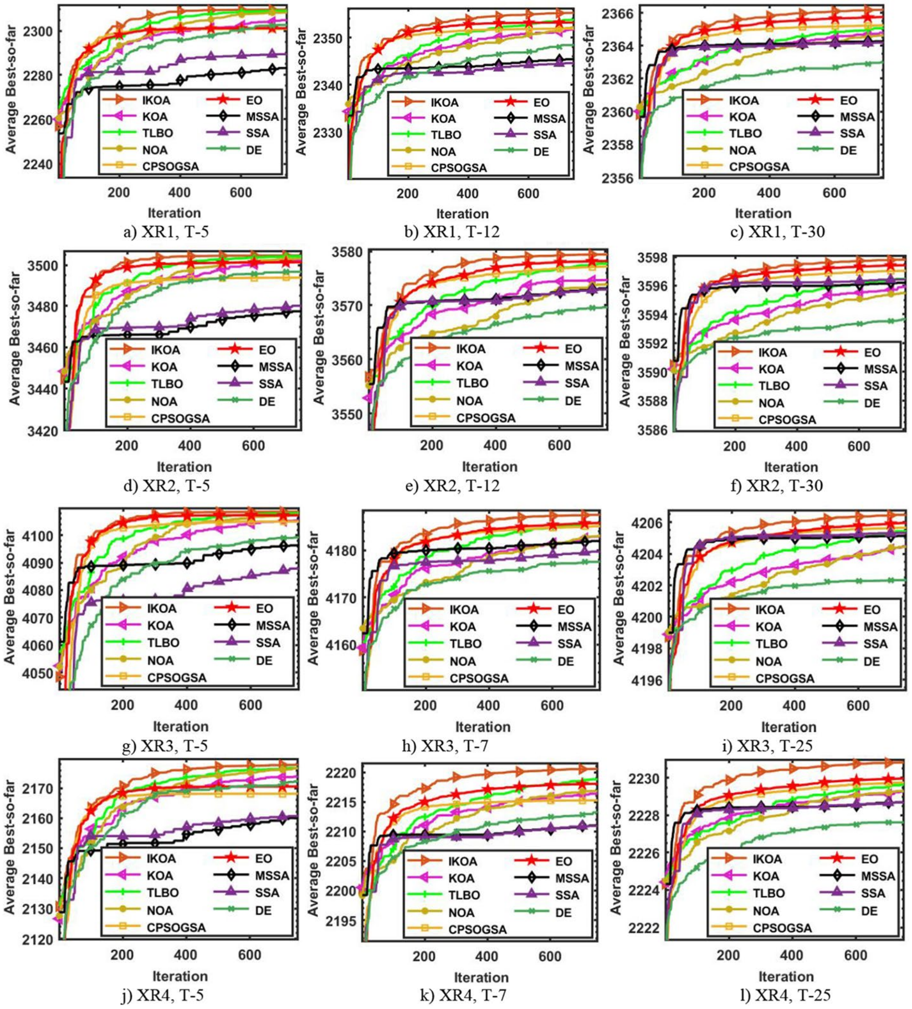
Table 3 Comparison under the Wilcoxon rank-sum test

T	KOA	TLBO	NOA	CPSOGSA	EO	MSSA	SSA	DE
T-5	3.6200E−06	5.2220E−05	8.4343E−06	3.8736E−04	4.1029E−02	2.2191E−07	1.2741E−07	1.3005E−07
T-7	2.1215E−07	8.9050E−04	7.9429E−07	3.7258E−04	1.1743E−04	9.6443E−08	8.3322E−08	1.4566E−07
T-8	2.5814E−06	2.9497E−04	6.0420E−06	1.7131E−04	8.0209E−04	9.1517E−08	8.3131E−08	7.2440E−07
T-10	4.5105E−07	5.4989E−05	2.3633E−06	9.6592E−06	1.9652E−03	7.7866E−08	7.4895E−08	7.4895E−08
T-12	1.4414E−07	2.1812E−04	2.2236E−07	1.3448E−04	1.2130E−03	7.7866E−08	7.7651E−08	6.7956E−08
T-15	3.5561E−07	1.3538E−04	2.7791E−06	2.4001E−03	2.4755E−02	9.0155E−08	1.1547E−07	6.7956E−08
T-18	1.8121E−07	9.4566E−05	1.4353E−07	5.6731E−04	2.6634E−03	7.0712E−08	6.9334E−08	6.7956E−08
T-20	1.4112E−07	1.3532E−05	5.9447E−07	2.3398E−05	1.5431E−03	8.2679E−08	1.0094E−07	6.7956E−08
T-25	1.3809E−07	1.6307E−06	1.0637E−07	1.5352E−04	1.4737E−02	1.1076E−07	1.0870E−07	6.7956E−08
T-30	7.8959E−08	9.1363E−07	1.0340E−07	1.3306E−04	6.2798E−03	1.3264E−07	1.0488E−07	6.7956E−08

Bold values show that there is a difference

Convergence speed analysis

In this section, the rival and compared algorithms' convergence curves for some CXR images segmented at some threshold levels are presented to determine which algorithm is capable of rapidly reaching the best fitness value. All algorithms are independently executed 20 runs, and the average convergence curve within those times is described in Fig. 8. This figure demonstrates that IKOA is significantly faster than all of the rival optimizers at all investigated threshold levels. This figure also illustrates that EO is the



a) XR1, T-5 b) XR1, T-12 c) XR1, T-30

d) XR2, T-5 e) XR2, T-12 f) XR2, T-30

g) XR3, T-5 h) XR3, T-7 i) XR3, T-25

j) XR4, T-5 k) XR4, T-7 l) XR4, T-25

Fig. 8 Comparison in terms of Convergence curve

second-best algorithm, whereas SSA, DE, and MSSA are the methods with the worst convergence curve. Even now, IKOA performs superiorly to the other competitors for convergence speed, PSNR, SSIM, FSIM, fitness value, and F-rank. As a result, we conclude that IKOA is more effective for accurately segmenting the CXR images.

Computational cost analysis

In the previous experiments, we illustrated that IKOA is more effective than all the compared algorithms. In this section, we illustrate the efficiency of IKOA in terms of computational cost compared to rival optimizers. All algorithms are executed 20 independent times, and the average computational cost required by each algorithm is reported in Fig. 9. This figure shows that IKOA consumes the least computational cost, where it needs an average computational cost of 2.695, followed by SSA with a value of 2.808, while TLBO is the worst with a value of 2.933. From this, we can conclude that IKOA is a robust optimizer for accurately and rapidly segmenting the CXR images, as it is more effective and efficient than all compared algorithms.

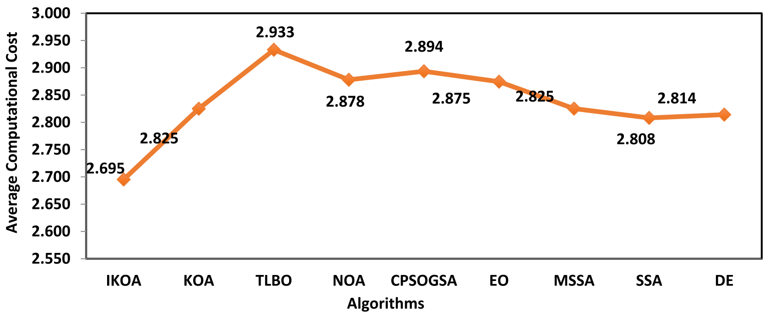


Fig. 9 Comparison in terms of computational cost

Table 4 Comparison of CNN-IKOA's outcomes under various threshold levels

T	Accuracy	Specificity	Precision (M)	Precision (W)	Recall (M)	Recall (W)	F1-score (M)	F1-score (W)
SI	93.26	95.76	93.40	93.40	93.20	93.20	93.40	93.20
T-3	92.91	95.42	93.40	93.20	92.80	93.00	93.00	93.00
T-4	94.56	96.04	94.80	94.80	94.60	94.80	94.80	94.40
T-5	94.83	94.51	96.47	94.80	94.60	94.60	94.60	94.80
T-7	94.23	93.63	95.75	94.00	93.80	93.80	93.80	93.80
T-8	94.32	96.25	94.40	94.40	94.40	94.20	94.40	94.20
T-10	94.44	96.28	94.80	94.60	94.80	94.60	94.80	94.60
T-12	94.88	**96.57**	**95.60**	**95.40**	**95.80**	95.40	**95.60**	**95.40**
T-15	**94.94**	96.54	**95.60**	**95.40**	95.60	**95.60**	95.40	95.20

Experiment 2: segmented COVID-19 images classification

In this section, the proposed CNN-IKOA's performance is observed to show its classification accuracy under the CXR images segmented by IKOA at different threshold levels and the source images (SI).

Performance evaluation of CNN-IKOA under various threshold levels

This CNN-IKOA model is executed five independent times for training under segmented images at each threshold level, and the average values for eight performance metrics, including accuracy, macro precision (precision (M)), weighted precision (precision (W)), macro recall (recall (M)), weighted recall (recall (W)), macro F1-score (F1-score (M)), weighted F1-score (F1-score (W)), and specificity, are calculated and presented in Table 4. Inspecting this table reveals that CNN-IKOA with the images segmented at 12-T and 15-T could achieve an overall classification accuracy of 94.88% and 94.94%, respectively, which are significantly better than the performance of CNN with the original images, which could achieve a classification accuracy of 93.26%. Also, this table shows that CNN-IKOA at 12-T and 15-T is competitive with each other and superior to its performance at the other threshold levels.

Tables 5 and 6 show the classification confusion matrices of the proposed CNN-IKOA for segmented images at different threshold levels and for the original images. This table demonstrates that the CNN-IKOA at T-12 could reach outstanding precision and recall,

Table 5 Confusion matrix of CNN under the source images

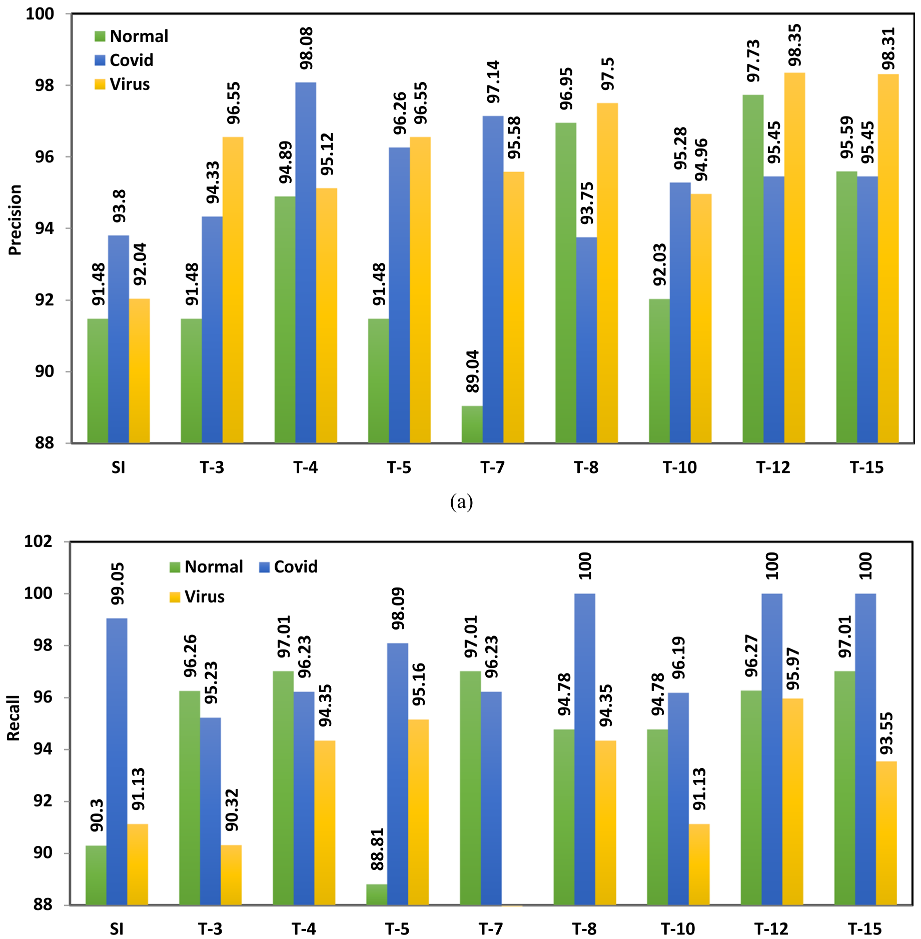
Predicted classes					
		Normal	**COVID**	**Virus**	**Recall (%)**
Actual classes	Normal	121	5	8	90.30
	Covid	1	104	0	99.05
	Virus	7	4	113	91.13
Precision (%)		91.48	93.80	92.04	–

Table 6 Confusion matrix of CNN-IKOA at various threshold levels

Predicted classes									
	T-3					**T-4**			
		Normal	**COVID**	**Virus**	**Recall (%)**	**Normal**	**COVID**	**Virus**	**Recall (%)**
Actual classes	Normal	129	4	1	96.26	**130**	**1**	**3**	97.01
	Covid	2	100	3	95.23	**1**	**102**	**3**	96.23
	Virus	10	2	112	90.32	**6**	**1**	**117**	94.35
Precision (%)		91.48	94.33	96.55	–	94.89	98.08	95.12	–
	T-5					**T-7**			
		Normal	**COVID**	**Virus**	**Recall (%)**	**Normal**	**COVID**	**Virus**	**Recall (%)**
Actual classes	Normal	119	2	13	88.81	130	1	3	97.01
	Covid	1	103	1	98.09	2	102	1	96.23
	Virus	4	2	118	95.16	14	2	108	87.10
Precision (%)		91.48	96.26	96.55	–	89.04	**97.14**	95.58	–
	T-8					**T-10**			
		Normal	**COVID**	**Virus**	**Recall (%)**	**Normal**	**COVID**	**Virus**	**Recall (%)**
Actual classes	Normal	127	4	3	94.78	127	2	5	94.78
	Covid	0	105	0	**100**	3	101	1	96.19
	Virus	4	3	117	94.35	8	3	113	91.13
Precision (%)		96.95	93.75	97.50	–	92.03	95.28	94.96	–
	T-12					**T-15**			
		Normal	**COVID**	**Virus**	**Recall (%)**	**Normal**	**COVID**	**Virus**	**Recall (%)**
Actual classes	Normal	129	3	2	96.27	130	3	1	97.01
	Covid	0	105	0	**100**	**0**	**105**	**0**	**100**
	Virus	3	2	119	**95.97**	6	2	116	93.55
Precision (%)		**97.73**	95.45	**98.35**	–	95.59	95.45	98.31	–

The bold values highlight the best precision and recall values obtained for three classes (Normal, COVID, Virus)

where it could achieve recall values of 100% for the COVID class, 96.27% for the normal class, and 95.97% for the virus class, and could reach precision values of 97.73% for the normal class, 95.45% for the COVID class, and 98.35% for the virus class. From that, it is clear that the performance of CNN-IKOA at T-12 is better than its performance with the source images and the other threshold levels. The precision and recall estimated by the proposed model for each possible class before and after performing the segmentation step are presented in Fig. 10. This figure shows that CNN-IKOA under T-12 could achieve better precision for both viral and normal classes with values of 97.73% and



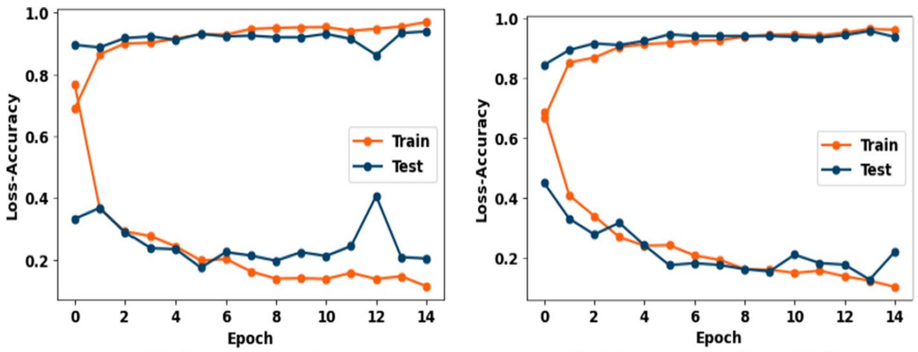
(a)

(b)

Fig. 10 Precision and recall obtained by CNN-IKOA for each possible class before and after the segmentation step

98.35%, respectively, while its performance under T-12 could achieve better precision for the COVID class with a value of 98.08%. This figure also shows that the performance of CNN-IKOA without performing the segmentation step is worse than that under the majority of threshold levels. Regarding the recall metric, CNN-IKOA under T-12, T-8, and T-15 has the same recall for the COVID class with a value of 100, which is better than all the precision values obtained under the other threshold levels and the source images. The recall values for the other classes are maximized by the proposed model under T-12 and T-15, as reported in Fig. 10(b).

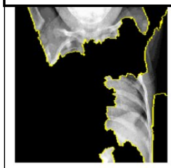
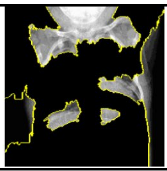
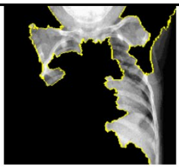
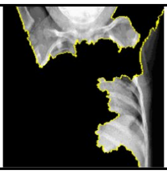
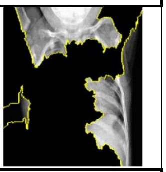
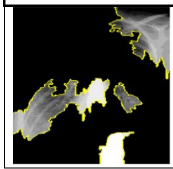
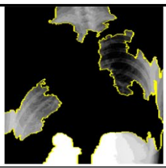
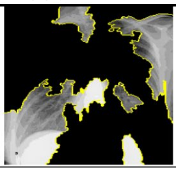
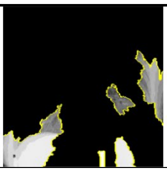
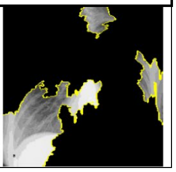
In addition, Fig. 11 compares the CNN-IKOA's performance using the accuracy and loss curves of the training and testing CXR images before and after the segmentation process. This figure shows that the performance of CNN-IKOA with the segmented images is approximately the same on the training and testing datasets; on the contrary, before performing the segmentation step, its performance on the training dataset is somewhat better than its performance on the testing dataset. Consequently, the segmentation of CXR images using IKOA has a substantial positive effect on the performance of the CNN model for accurately and rapidly detecting COVID-19 infection.



a) Before segmentation b) After segmentation at T-12

Fig. 11 The accuracy and loss curves of CNN-IKOA after and before the segmentation step

Table 7 illustration of the top 10 features as the number of perturbations or samples change

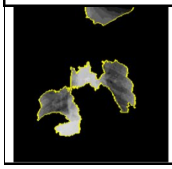
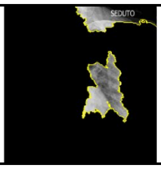
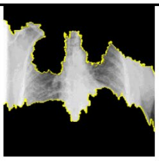
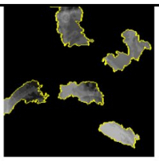
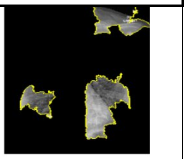
num_samples=50	num_samples=100	num_samples=200	num_samples=300	num_samples=400
Normal class				
				
COVID-19 class				
				

Local interpretable model-agnostic explanations as explainable technique

In this section, the overall prediction of the proposed CNN-IKOA is interpreted using the local interpretable model-agnostic explanations (LIME) technique. The LIME technique is widely used for its dependable explanations of image classification subtleties and is considered one of the few approaches that performs well with text, tabular data, and images [56]. LIME generates superpixels For image classification; these superpixels are considered the outcomes of image over-segmentation. Compared to rectangular image patches, superpixels are better aligned with the image's edges and hold more data than pixels for the main prediction [56].

After applying the lime technique under different numbers of samples, such as 50, 100, 200, 300, and 400, the top 10 features are selected for a COVID-19 image and a normal image and displayed in Table 7 to determine if the selected features vary as the number of samples changes. This table shows that there are a few differences in the selected features as the number of samples increases. Finally, the top five features that affect the predictions of CNN-IKOA trained by the images segmented using IKOA at T-12 are shown in Table 8. This table shows that the majority of these features for the considered COVID-19 images focus on the patients' lungs, and this shows that the proposed CNN-IKOA focuses more on the area that might contain COVID-19 infection.

Table 8 Top five features that enabled the detection of COVID-19 infection from the segmented CXR images

COVID-19 class				
				

Summarization of the conducted experiments

In the previous sections, two different experiments were conducted. The first experiment is performed to test and verify the performance of IKOA compared to classical KOA and seven rival optimizers when applied to segment ten CXR images at small, medium, and high threshold levels. All nine algorithms were run separately 20 times in this experiment. The obtained outcomes were then analyzed using AFV, F-rank, PSNR, SSIM, the convergence curve, the p-value from the WRS test, and the computational cost. The results for these metrics revealed that IKOA is significantly different from and more effective and efficient than all the rival optimizers.

The second experiment was done to reveal the influence of the CXR images segmented by IKOA at different threshold levels on the performance of CNN-IKOA. Therefore, the proposed IKOA was applied to segment a set of CXR images, which are classified into three classes: normal, COVID-19, and viral pneumonia, at each threshold level from the following levels: T-3, T-4, T-5, T-7, T-8, T-10, T-12, and T-15. Then, the segmented CXR images were used to train and test CNN-IKOA at five independent times, and its performance was analyzed using eight performance metrics, including accuracy, precision (M), precision (W), recall (M), recall (W), F1-score (M), F1-score (W), and specificity. The results of these metrics indicate that CNN-IKOA has the potential to perform more effectively when applied to images segmented at T-12, where it achieved an overall accuracy of 94.88%, a specificity of 96.57%, a precision of 95.40%, and a recall of 95.40%. The conclusion that can be drawn from this experiment is that metaheuristic algorithms have the potential to assist in improving the classification accuracy of deep learning models, not only for the COVID-19 infection but also for any image classification problem.

Conclusion and future work

Over the last four years, deep learning and machine learning techniques have been widely used to automatically detect COVID-19 infection from CXR and CT images in an attempt to speed the diagnosis process and alleviate human mistakes. The performance of those techniques can be improved if the CXR and CT images are accurately segmented to separate the most relevant features that might aid in accurately classifying the COVID-19 infection. Several approaches have been proposed for accurately performing the image segmentation step. Among those approaches, the multilevel thresholding approach has been widely used due to its simplicity and accuracy. However, multilevel threshold-based image segmentation techniques, such as Kapur's entropy and the Otsu method, cannot accurately segment the images as the number of threshold levels increases. Therefore, metaheuristic algorithms have recently collaborated with those traditional techniques to better tackle this problem, especially when increasing the number of threshold levels. Unfortunately, those algorithms have some drawbacks, including slow convergence speed, stagnation into local minima, and expensive computational costs, that make them unable to achieve outstanding outcomes when applied to tackle this problem. As a result, this study presents an improved version of the Kepler optimization algorithm (IKOA) for alleviating those drawbacks, thereby– achieving a better image segmentation process for the CXR images. IKOA is first assessed using ten CXR images at ten various threshold levels to test and verify its performance for small, medium, and high threshold levels. The outcomes of IKOA are compared to those of several rival optimization techniques in terms of several performance metrics to expose its effectiveness. This comparison shows the superiority of IKOA in terms of all performance metrics used. In addition, the IKOA-based segmented CXR images at eight distinct threshold levels are used to train a new CNN model dubbed CNN-IKOA in order to determine the influence of the segmentation step on the performance of the deep learning models. Five performance indicators—overall accuracy, F1-score, recall, precision, and specificity—were used to show the effectiveness of the segmented CXR images at each threshold level for training CNN-IKOA. The test results show that CNN-IKOA works well when trained on CXR images that have been segmented at T-12. It achieved an overall accuracy of 94.88%, a specificity of 96.57%, a precision of 95.40%, and a recall of 95.40%. Our future work will investigate the effectiveness of the segmented CXR images by IKOA for some of the existing deep-learning models. In addition, some of the other recently proposed metaheuristic algorithms, like the nutcracker optimizer, spider wasp optimizer, and gorilla troops optimizer, will be applied for better segmenting the CXR images to further improve the classification accuracy.

Author contributions

MA; RM and IA Investigation, Methodology, Resources, Supervision, Visualization, Writingoriginal draft, Writing-—review and editing, Software, Conceptualization. KMS and IAH Investigation, Methodology, Validation, Writing—original draft, Writing-review and editing.

Funding

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education Saudi Arabia for funding this research work through the project number 223202.

Declarations

Ethics approval and consent to participate

This article does not contain any studies with human participants or animals performed by any of the authors.

Competing interests

The authors declare that there is no competing interests in the research.

Received: 28 September 2023 Accepted: 14 December 2023
Published online: 10 January 2024

References

1. World Health, O., Coronavirus disease (COVID-19), 12 Oct 2020. 2020.
2. Kong W-H, et al. SARS-CoV-2 detection in patients with influenza-like illness. Nat Microbiol. 2020;5(5):675–8.
3. Bassi, P.R.A.S. and R. Attux, A deep convolutional neural network for COVID-19 detection using chest X-rays. Res Biomed Eng. 2021 1–10.
4. Sahinbas K, Catak FO. Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images. In: Data science for COVID-19. Amsterdam: Elsevier; 2021. p. 451–66.

5. Ismael AM, Şengür A. Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl. 2021;164: 114054.
6. Shah PM, et al. Deep GRU-CNN model for COVID-19 detection from chest X-rays data. Ieee Access. 2021;10:35094–105.
7. Dhivya, P., et al., Square static–deep hyper optimization and genetic meta-learning approach for disease classification. IETE J Res. 2023. 1–10.
8. Sharma S, Parmar M. Heart diseases prediction using deep learning neural network model. Int J Innov Technol Exploring Eng. 2020;9(3):2244–8.
9. Liu Y, et al. A deep learning system for differential diagnosis of skin diseases. Nat Med. 2020;26(6):900–8.
10. Sharma, R., et al., Plant disease diagnosis and image classification using deep learning. computers materials Continua, 2022. 71(2).
11. Sahoo SK, et al. Self-adaptive moth flame optimizer combined with crossover operator and *Fibonacci* search strategy for COVID-19 CT image segmentation. Expert Syst Appl. 2023;227: 120367.
12. Han Y, et al. A solution to the stagnation of multi-verse optimization: an efficient method for breast cancer pathologic images segmentation. Biomed Signal Process Control. 2023;86: 105208.
13. Upadhyay P, Chhabra JK. Multilevel thresholding based image segmentation using new multistage hybrid optimization algorithm. J Ambient Intell Humaniz Comput. 2021;12:1081–98.
14. Kanna PR, Santhi P. Hybrid intrusion detection using mapreduce based black widow optimized convolutional long short-term memory neural networks. Expert Syst Appl. 2022;194: 116545.
15. Abdel-Basset M, et al. Solar photovoltaic parameter estimation using an improved equilibrium optimizer. Sol Energy. 2020;209:694–708.
16. Saber S, Salem S. High-performance technique for estimating the unknown parameters of photovoltaic cells and modules based on improved spider wasp optimizer. SMIJ. 2023. https://doi.org/10.61185/SMIJ.2023.55102.
17. Kanna PR, Santhi P. Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features. Knowl-Based Syst. 2021;226: 107132.
18. Salem S. An improved binary quadratic interpolation optimization for 0–1 knapsack problems. SMIJ. 2023. https://doi.org/10.1185/SMIJ.2023.44101.
19. Sun J, et al. MFBCNNC: momentum factor biogeography convolutional neural network for COVID-19 detection via chest X-ray images. Knowl-Based Syst. 2021;232: 107494.
20. Wang L, Lin ZQ, Wong A. Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Sci Rep. 2020;10(1):19549.
21. Islam MZ, Islam MM, Asraf A. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Inform Med Unlocked. 2020;20: 100412.
22. Malik H, et al. CDC_Net: Multi-classification convolutional neural network model for detection of COVID-19, pneumothorax, pneumonia, lung cancer, and tuberculosis using chest X-rays. Multimedia Tools Appl. 2023;82(9):13855–80.
23. Hussein HI, et al. Lightweight deep CNN-based models for early detection of COVID-19 patients from chest X-ray images. Expert Syst Appl. 2023;223: 119900.
24. Gupta H, et al. A hybrid convolutional neural network model to detect COVID-19 and pneumonia using chest X-ray images. Int J Imaging Syst Technol. 2023;33(1):39–52.
25. Nafisah SI, et al. A Comparative evaluation between convolutional neural networks and vision transformers for COVID-19 detection. Mathematics. 2023;11(6):1489.
26. George GS, et al. COVID-19 detection on chest X-ray images using Homomorphic Transformation and VGG inspired deep convolutional neural network. Biocybernet Biomed Eng. 2023;43(1):1–16.
27. Joshi AM, et al. LiMS-Net: a lightweight multi-scale CNN for COVID-19 detection from chest CT scans. ACM Trans Manag Inf Syst. 2023;14(1):1–17.
28. Kaya Y, Gürsoy E. A MobileNet-based CNN model with a novel fine-tuning mechanism for COVID-19 infection detection. Soft Comput. 2023;27(9):5521–35.
29. Aslani S, Jacob J. Utilisation of deep learning for COVID-19 diagnosis. Clin Radiol. 2023;78(2):150–7.
30. Chakraborty S, et al. COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. Comput Biol Med. 2021;139: 104984.
31. Han Y, et al. Multi-verse optimizer with rosenbrock and diffusion mechanisms for multilevel threshold image segmentation from COVID-19 chest X-ray images. J Bionic Eng. 2023;20(3):1198–262.
32. Abualigah L, et al. Improved reptile search algorithm by salp swarm algorithm for medical image segmentation. J Bionic Eng. 2023;7:1–25.
33. Su H, et al. Horizontal and vertical search artificial bee colony for image segmentation of COVID-19 X-ray images. Comput Biol Med. 2022;142: 105181.
34. Nama S. A novel improved SMA with quasi reflection operator: performance analysis, application to the image segmentation problem of Covid-19 chest X-ray images. Appl Soft Comput. 2022;118: 108483.
35. Xing J, et al. Boosting whale optimizer with quasi-oppositional learning and gaussian barebone for feature selection and COVID-19 image segmentation. J Bionic Eng. 2023;20(2):797–818.
36. Yang C, et al. Performance optimization of photovoltaic and solar cells via a hybrid and efficient chimp algorithm. Sol Energy. 2023;253:343–59.
37. Qi A, et al. Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation. Comput Biol Med. 2022;148: 105810.
38. Zhao S, et al. Boosted crow search algorithm for handling multi-threshold image problems with application to X-ray images of COVID-19. Expert Syst Appl. 2023;213: 119095.
39. Liu L, et al. Ant colony optimization with Cauchy and greedy Levy mutations for multilevel COVID 19 X-ray image segmentation. Comput Biol Med. 2021;136: 104609.
40. Ryalat MH, et al. Harris hawks optimization for COVID-19 diagnosis based on multi-threshold image segmentation. Neural Comput Appl. 2023;35(9):6855–73.

41. Abdel-Basset M, et al. Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. Knowl-Based Syst. 2023;268: 110454.
42. Gu J, et al. Recent advances in convolutional neural networks. Pattern Recogn. 2018;77:354–77.
43. Yamashita R, et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging. 2018;9:611–29.
44. Ghosh, A., et al., Fundamental concepts of convolutional neural network. Recent trends and advances in artificial intelligence and Internet of Things, 2020: p. 519–567.
45. Otsu N. A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern. 1979;9(1):62–6.
46. Siddhartha, M. and A. Santra, COVIDLite: A depth-wise separable deep neural network with white balance and CLAHE for detection of COVID-19. arXiv preprint arXiv:2006.13873, 2020.
47. Abdel-Basset M, et al. HWOA: A hybrid whale optimization algorithm with a novel local minima avoidance method for multi-level thresholding color image segmentation. Expert Syst Appl. 2022;190: 116145.
48. Wang S, Jia H, Peng X. Modified salp swarm algorithm based multilevel thresholding for color image segmentation. Math Biosci Eng. 2020;17(1):700–24.
49. Rather SA, Bala PS. Constriction coefficient based particle swarm optimization and gravitational search algorithm for multilevel image thresholding. Expert Syst. 2021;38(7): e12717.
50. Abdel-Basset M, et al. Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowl-Based Syst. 2023;262: 110248.
51. Abdel-Basset M, Chang V, Mohamed R. A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. Neural Comput Appl. 2021;33:10685–718.
52. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des. 2011;43(3):303–15.
53. Cuevas E, Zaldivar D, Pérez-Cisneros M. A novel multi-threshold segmentation approach based on differential evolution optimization. Expert Syst Appl. 2010;37(7):5265–71.
54. Ahmadianfar I, Bozorg-Haddad O, Chu X. Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf Sci. 2020;540:131–59.
55. Lam FC, Longnecker MT. A modified Wilcoxon rank sum test for paired data. Biometrika. 1983;70(2):510–3.
56. Ahsan, M.M., et al. Detection of COVID-19 patients from CT scan and chest X-ray data using modified MobileNetV2 and LIME. MDPI.

Publisher's Note

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