Applied Soft Computing Journal 98 (2021) 106742

Contents lists available at ScienceDirect

Applied Soft Computing Journal

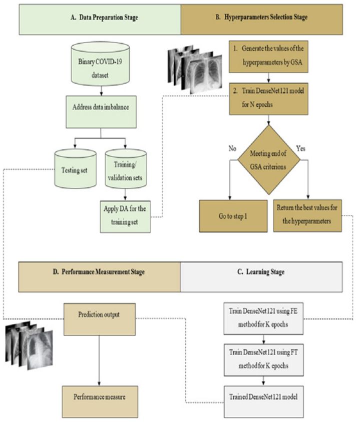
journal homepage: www.elsevier.com/locate/asoc

An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization

Dalia Ezzat [a,1], Aboul Ella Hassanien [a,*,1], Hassan Aboul Ella [b,1]

[a] *Faculty of Computers and Artificial Intelligence, Cairo University, Egypt*
[b] *Microbiology Department, Faculty of Veterinary Medicine, Cairo University, Egypt*

GRAPHICAL ABSTRACT


A. Data Preparation Stage
B. Hyperparameters Selection Stage
Binary COVID-19 dataset
Address data imbalance
Testing set
Training/ validation sets
Apply DA for the training set
1. Generate the values of the hyperparameters by GSA
2. Train DenseNet121 model for N epochs
Meeting end of GSA criterions
No
Yes
Go to step 1
Return the best values for the hyperparameters
D. Performance Measurement Stage
C. Learning Stage
Prediction output
Performance measure
Train DenseNet121 using FE method for K epochs
Train DenseNet121 using FT method for K epochs
Trained DenseNet121 model

ARTICLE INFO

Article history:
Received 28 April 2020
Received in revised form 5 September 2020
Accepted 17 September 2020
Available online 22 September 2020

Keywords:
SARS-CoV-2
Deep learning
Convolutional neural networks
Transfer learning
Gravitational search algorithm
Hyperparameters optimization

ABSTRACT

In this paper, a novel approach called GSA-DenseNet121-COVID-19 based on a hybrid convolutional neural network (CNN) architecture is proposed using an optimization algorithm. The CNN architecture that was used is called DenseNet121, and the optimization algorithm that was used is called the gravitational search algorithm (GSA). The GSA is used to determine the best values for the hyperparameters of the DenseNet121 architecture. To help this architecture to achieve a high level of accuracy in diagnosing COVID-19 through chest x-ray images. The obtained results showed that the proposed approach could classify 98.38% of the test set correctly. To test the efficacy of the GSA in setting the optimum values for the hyperparameters of DenseNet121. The GSA was compared to another approach called SSD-DenseNet121, which depends on the DenseNet121 and the optimization algorithm called social ski driver (SSD). The comparison results demonstrated the efficacy of the proposed GSA-DenseNet121-COVID-19. As it was able to diagnose COVID-19 better than SSD-DenseNet121 as the second was able to diagnose only 94% of the test set. The proposed approach was also compared to another method based on a CNN architecture called Inception-v3 and manual search to quantify hyperparameter values. The comparison results showed that the GSA-DenseNet121-COVID-19 was able to beat the comparison method, as the second was able to classify only 95% of the test set samples. The proposed GSA-DenseNet121-COVID-19 was also compared with some related work. The comparison results showed that GSA-DenseNet121-COVID-19 is very competitive.

[*] Corresponding author.
E-mail address: aboitcairo@cu.edu.eg (A.E. Hassanien).

[1] Scientific Research Group in Egypt (SRGE) www.egyptscience.net.

https://doi.org/10.1016/j.asoc.2020.106742
1568-4946/Published by Elsevier B.V.

1. Introduction

On 11/March/2020, the world health organization (WHO) announced that the novel coronavirus disease-2019 (COVID-19) had been a Pandemic outbreak. COVID-19 is a respiratory disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). SARS-CoV-2 is a virus that belongs to the Coronavirdeae family, which is the same family of both severe acute respiratory syndrome coronavirus and middle east respiratory syndrome coronavirus (MERS-CoV) [1]. Both SARS-CoV-1 and MERS-CoV are the 2002 severe acute respiratory syndrome causative agent and the 2012 middle east respiratory syndrome MERS epidemic [2,3]. This announcement was the beginning of the current medical health problem faced and shared by the whole world during the last few months. Up till now, there are no efficient protective vaccines, neutralizing antisera or curative medication have been developed or officially approved to be used in COVID-19's patients worldwide. The continuous increase in morbidities and mortalities due to SARS-CoV-2 leads to international medical health worsen situations day after day. Therefore, this emerging COVID-19 pandemic becomes an ongoing challenge for all medical health workers and researchers, by applying the natural timeline of infectious diseases [4] on COVID-19, as shown in Fig. 1. The importance of shortening the period between the onset of symptoms and the usual diagnosis will appear. Therefore, an efficient rapid diagnostic test or protocol will help to achieve proper early medical caring to COVID-19 patients that, by its role, will help to save a lot of lives worldwide. Finding a rapid, efficient diagnostic test or protocol becomes one of those top critical priorities.

Quantitative reverse transcriptase-polymerase chain reaction (qPCR) is the golden standard test for confirmed laboratory diagnosis of COVID-19. Other rapid, bedside, field, and point of care immunochromatographic lateral flow, nucleic acid lateral flow, nucleic immunochromatographic lateral flow, and CRISPR CAS-12 lateral flow are under development [5]. COVID-19 as a pneumonic disease characterized by general pneumonic lung affections with absolute uniqueness from other pneumonia-causing Coronaviruses. Although the radiological imaging features closely similar and overlapping those associating with SARS and MERS. The bilateral lung involvement on initial imaging is more likely to be seen with COVID-19, as those associating SARS and MERS are more predominantly unilateral. So using radiological imaging techniques such as X-rays and computed tomography (CT) is of great value as confirmed, need an expert but rapid diagnostic approach either separately or in combination with qPCR. To avoid the false negative/positive COVID-19 results, recorded and reported during separate qPCR in the disease's early stage [6].

Deep learning (DL) is the most common and accurate method for dealing with medical datasets that contain a relatively large number of training samples. For instance, the classification of brain abnormalities, the classification of different types of cancer, the classification of pathogenic bacteria, and biomedical image segmentation [7–11]. One of DL's most important characteristics is its ability to handle relatively large amounts of data efficiently [12]. As well as, it eliminates the need to extract essential features manually [13]. According to these reasons, many efforts have depended on the DL methods, especially CNN architectures, to diagnose COVID-19 through chest radiological imaging, especially X-rays. For instance, in this study [14], the authors used several state-of-the-art CNN architectures by applying the transfer learning method to diagnose COVID-19. The results of their experiments indicated that the MobileNet architecture was the best with an accuracy of 96.78%. In another study [15], the DarkCovidNet model was proposed to diagnose COVID-19 using a two-classes dataset (COVID-19, No-Findings) and three-classes dataset (COVID-1, No-Findings, Pneumonia). The experimental results showed that the DarkCovidNet model had diagnosed the COVID-19 with higher accuracy by using the two-classes dataset, where the accuracy reached 98.08%. In [16] to improve CNN architectures' performance in diagnosing COVID-19, a new model called CovidGAN is built, generating new samples from the dataset samples used. The experiment results demonstrated that the CovidGAN model helped the VGG16 network diagnose the COVID-19 with 95% accuracy. In [17], the authors produced a model called CoroNet, and this model was able to diagnose COVID-19 with 89.6% accuracy in a four-category classification (COVID-19 vs. bacterial pneumonia vs. viral pneumonia vs. normal). In a three-category classification (COVID vs. pneumonia vs. normal), the CoroNet model achieved 94% accuracy. While in a two-category classification (COVID vs. normal), the CoroNet achieved a higher accuracy.

Despite the promising results these CNN architectures have achieved in detecting the COVID-19, the large number of hyperparameters therein represents an obstacle to achieving better results. However, very few studies, such as [18], have sensed these hyperparameters' importance in obtaining high efficiency with CNN architectures and the necessity of treating them as an optimization problem. Where in [18], the authors presented a method for detecting COVID-19, which is the deep Bayes-SqueezeNet method. This method is based on Bayesian optimization to fine-tune hyperparameters of the CNN architecture called SqueezeNet. The deep Bayes-SqueezeNet reached 98.3% accuracy in the three-category classification (Normal vs. COVID-19 vs. Pneumonia).

Many studies, such as [19,20] conducted to determine the extent of hyperparameters' influence on various DL architectures. These studies have found the hyperparameters that offer significant performance improvements in simple networks do not have the same effect in more complex networks. The hyperparameters that fit one dataset do not fit another dataset with different properties The choice of values for these hyperparameters often depends on a combination of human experience, trial and error, or a grid search method [21]. Due to the nature of computationally expensive CNN architectures, which can take several days to train, the trial and error method is ineffective [22]. The grid search method is usually not suitable for CNN architectures because the number of combinations grows exponentially with the number of hyperparameters [23]. Therefore, the automatic optimization of the hyperparameters of CNNs architectures is so essential [24,25].

This paper introduces an approach called GSA-DenseNet121-COVID-19. This approach relies on a pre-trained CNN architecture called DenseNet121 to diagnose COVID-19 by applying the transfer learning method. GSA [26] was used to select optimal values for the hyperparameters of DenseNet121. The aim of proposing this approach is to facilitate and expedite the analysis of chest X-rays taken during various COVID-19 diagnostic protocols. Especially now, after the regular increase in COVID-19 patients every day, it is complicated and exhaustive to all medical field personnel to achieve the same high-quality analysis of chest X-rays along the whole day, 24/7. Therefore, automating specific steps of diagnostic protocols is a must to keep the integrity of the diagnostic quality of medical field practitioners. Accordingly, the contributions of this paper can be summarized in the following points:

- This paper provides a diagnostic approach to COVID-19 that can be used alone or in combination with qPCR to reduce the false negative and false positive rates of qPCR.

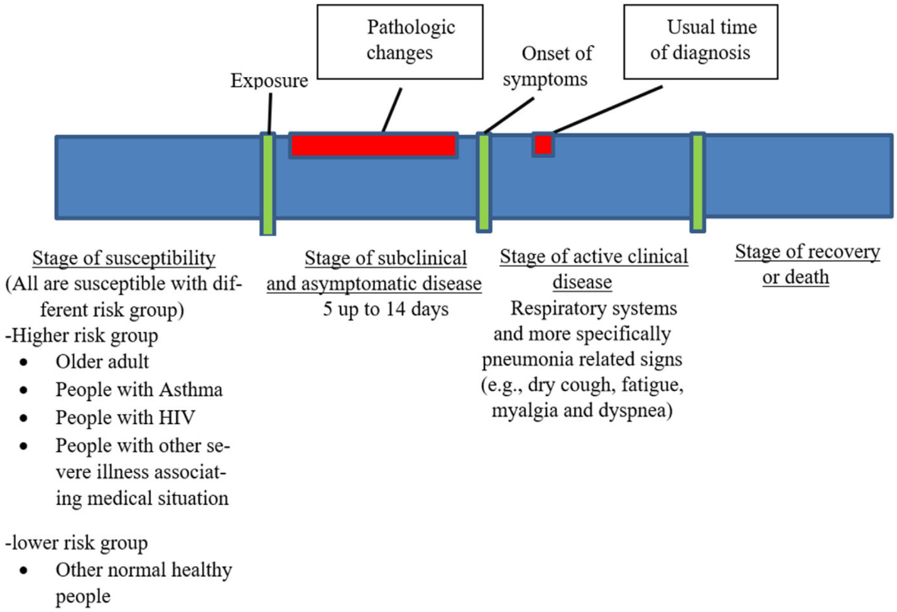

Pathologic changes
Usual time of diagnosis
Onset of symptoms
Exposure
Stage of susceptibility
(All are susceptible with different risk group)
-Higher risk group
Older adult
People with Asthma
People with HIV
People with other severe illness associating medical situation
-lower risk group
Other normal healthy people
Stage of subclinical and asymptomatic disease
5 up to 14 days
Stage of active clinical disease
Respiratory systems and more specifically pneumonia related signs (e.g., dry cough, fatigue, myalgia and dyspnea)
Stage of recovery or death

Fig. 1. Natural timeline of infectious diseases on COVID-19.
Source: Adapted from [4].

- The proposed approach is called GSA-DenseNet121-COVID-19, and it adopts a transfer learning method using a pre-trained CNN architecture called DenseNet121 that has been hybridized with an optimization algorithm called GSA.
- The GSA is used to improve the classification performance of DenseNet121 by optimizing its hyperparameters.
- The proposed approach is scalable, that is, it can expand the classification, as the number of training, testing, and validation samples can be increased without having to specify the values of the hyperparameters of DenseNet121 manually.
- The results of the proposed approach to diagnosing the COVID-19 were very felicitous, achieving a 98.38% accuracy level on the test set.
- The proposed approach was compared with other approaches proposed, the results of the comparison showed that the proposed approach is superior to the other approaches despite being trained on smaller and more varied samples.

The rest of the paper is structured as follows: Section 2 represents the CNNs and the GSA's theoretical background. The dataset used is discussed in Section 3. While Section 4 shows details of the proposed approach. The results achieved by the proposed approach are illustrated in Section 5.

2. Theory and method

2.1. Gravitational search algorithm

GSA is an optimization technique gaining attention in the last years and developed by Rashedi [26]. It is based on the law of gravity, as shown in Eq. (1) and the second law of motion, as shown in Eq. (3) [27]. It also depends on the general physical concept that there are three types of mass: inertial mass, active gravitational mass and passive gravitational mass [28]. The law of gravity states that every particle attracts every other particle with a gravitational force (F). The gravitational force (F) between two particles is directly proportional to the product of their masses (M_1 and M_2) and inversely proportional to the square of their distance (R^2). The second law of motion states that when a force (F) is utilized to a particle, its acceleration (a) is determined by the force and its mass (M).

$$\mathrm{F} = \mathrm{G}\frac{\mathrm{M_1 M_2}}{\mathrm{R^2}} \tag{1}$$

Where G is the gravitational constant, which decreases with increasing time, and it is calculated as equation (2) [29].

$$G(t) = G(t_0) \times \left(\frac{t_0}{t}\right)^{\beta} \quad , \beta < 1 \tag{2}$$

$$\mathrm{a} = \frac{\mathrm{F}}{\mathrm{M}} \tag{3}$$

The GSA is similar to the above basic laws with minor modifications in Eq. (1). Where Rashid [26] stated that based on the experimental results, inverse proportionality to distance (R) produces better results than R^2. The GSA can be expressed as an isolated N particle system, and their masses measure their performance. All particles attract each other by the force of gravity, and this force causes a universal movement of all particles towards the particles that have heavier mass. Consequently, masses collaborate using a direct form of communication through the force of gravity. The heavy masses represent good solutions as they move more slowly than lighter mass, while light masses represent worse solutions, moving towards the heavier masses faster. Each mass has four specifications: active gravitational mass, passive gravitational mass, inertial mass, and position. The mass' position corresponds to a solution to the problem, and the other specifications of the mass (active gravitational mass, passive gravitational, inertial mass) are determined utilizing the fitness function. The algorithm of GSA can be summarized in eight steps as follows:

- Step one: Initialization
  Assuming there is an isolated system with N particles (masses), the position of ith particle is denoted as :

  $$P_i = \left(p_i^1, p_i^2, \ldots, p_i^d, \ldots, p_i^N\right) \quad \text{for } i = 1, 2, 3, \ldots, \mathrm{N} \tag{4}$$

  Where p_i^d presents the position of ith particle in the dth dimension.
- Step two: Fitness evaluation of particles

In this step, the worst and best fitness is calculated as Eqs. (5) and (6) respectively for a minimization problem, and calculated as Eqs. (7) and (8) respectively for a maximization problem.

$$worst(t) = \max_{\mathrm{j}›\{1,\ldots,\ \mathrm{N}\}} \mathrm{fitness}_{\mathrm{j}}(t) \tag{5}$$

$$best(t) = \min_{\mathrm{j}›\{1,\ldots,\ \mathrm{N}\}} \mathrm{fitness}_{\mathrm{j}}(t) \tag{6}$$

$$worst(t) = \min_{\mathrm{j}›\{1,\ldots,\ \mathrm{N}\}} \mathrm{fitness}_{\mathrm{j}}(t) \tag{7}$$

$$best(t) = \max_{\mathrm{j}›\{1,\ldots,\ \mathrm{N}\}} \mathrm{fitness}_{\mathrm{j}}(t) \tag{8}$$

Where $\mathrm{fitness}_{\mathrm{j}}(t)$ is the fitness of the jth particle at time t.

- Step three: Calculate the gravitational constant $G(t)$
  In this step, the gravitational constant $G(t)$ at time t is calculated as follows [30]:

$$G(t) = G_0 \times (1 - \frac{t}{t_{\max}}) \tag{9}$$

  Where G_0 represents the initial value of the gravitational constant initializes randomly, t is the current time, $t_{\max}$ is the total time.
- Step four: Update the inertial and gravitational masses
  In this step, the inertia and gravitational masses are updated by the fitness function. Assuming the equality of the inertia and gravitational mass, the masses' values are calculated as follows:

$$\mathrm{M}_{ii} = \mathrm{M}_{pi} = \mathrm{M}_{ai} = \mathrm{M}_i \qquad \text{for } i = 1, 2, 3, \ldots,\ \mathrm{N} \tag{10}$$

$$\mathrm{m}_i(t) = \frac{\mathrm{fitness}_i(t) - worst(t)}{best(t) - worst(t)} \tag{11}$$

$$\mathrm{M}_i(t) = \frac{\mathrm{m}_i(t)}{\sum_{\mathrm{j}=1}^{N} \mathrm{m}_{\mathrm{j}}(t)} \tag{12}$$

  Where $\mathrm{fitness}_i(t)$ is the fitness of the ith particle at time t, $\mathrm{M}_i(t)$ is the mass of the ith particle at time t.
- Step five: Compute the total force
  In this step, the total force $F_i^d(t)$ that exerting on particle i in a dimension d at time t is calculated as follows:

$$F_i^d(t) = \sum_{\mathrm{j}=1, \mathrm{j}\neq i}^{kbest} rand_{\mathrm{j}} F_{\mathrm{ij}}^d(t) \tag{13}$$

  Where $rand_{\mathrm{j}}$ is a random number ∈ [0, 1], $kbest$ is the set of first $kbest$ particles with the best fitness value and the biggest masses. $F_{\mathrm{ij}}^d(t)$ is the force exerting from mass 'j' on mass 'i' at time 't' and is calculated as the following equation:

$$F_{\mathrm{ij}}^d(t) = G(t)\frac{\mathrm{M}_{pi}(t) \times \mathrm{M}_{a\mathrm{j}}(t)}{\mathrm{R}_{i\mathrm{j}}(t) + \tau}\left(p_{\mathrm{j}}^d(t) - p_i^d(t)\right) \tag{14}$$

  Where M_{pi} represents the passive gravitational mass associated with particle i, $\mathrm{M}_{a\mathrm{j}}$ represents the active gravitational mass associated with particle j. τ is a small positive constant to prevent division by zero, $\mathrm{R}_{i\mathrm{j}}(t)$ represents the Euclidian distance between particles j and i:

$$\mathrm{R}_{i\mathrm{j}}(t) = \left\| p_i(t), p_{\mathrm{j}}(t) \right\|_2 \tag{15}$$

- Step six: compute the velocity and acceleration
  In this step based on $F_i^d(t)$, the acceleration of the particle i, $a_i^d(t)$, at time t in the direction dth, and the next velocity of the particle i in the direction dth, $u_i^d\ (t+1)$, are calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{\mathrm{M}_{ii}(t)} \tag{16}$$

$$u_i^d\ (t+1) = rand_i \times u_i^d(t) + a_i^d(t) \tag{17}$$

  Where $\mathrm{M}_{ii}(t)$ represents the inertial mass of $i$thparticle, $rand_i$ is a random number ∈ [0, 1].
- Step seven: update particles' position
  In this step the next position of the particle i in the direction dth, $p_i^d\ (t+1)$, is calculated as follows:

$$p_i^d\ (t+1) = p_i^d(t) + u_i^d\ (t+1) \tag{18}$$

- Step eight: Repeat steps two to seven until the stop criteria are reached; these eight steps are illustrated by a flowchart shown in Fig. 2.

2.2. Convolutional neural networks

In this section, the main structure of any CNN architecture will be illustrated. Additionally, the transfer learning method and how to apply this method using a pre-trained CNN architecture will be explained.

Network Structure: CNN architectures consist of two bases, namely convolutional base and classifier base.

The convolutional base includes three significant types of layers are: convolutional layers [31], activation layers [32], and pooling layers [33]. These types of layers are used to discover the basic features of input images, which are called feature maps. A feature map is getting by performing convolution processes to the input image or prior features using a linear filter, merging a bias term. Then passing this feature map through a non-linear activation function such as Sigmoid [34] and Rectified Linear Unit (RELU) [35]. In contrast, the classifier base includes the dense layers combined with the activation layers to convert the feature maps to one dimension vectors to expedite the classification task using many neurons. Usually, one or more dropout layers [36] are added to the classifier base to minimize the overfitting that may encounter CNN architectures and improve their generalization. Adding any dropout layer to the classifier base introduces a new hyperparameter called dropout rate. This hyperparameter determines the probability at which outputs of the layer are removed, or reciprocally, the probability at which outputs of the layer are kept. Typically, the dropout rate is set in the range from 0.1 to 0.9 [37].

Transfer Learning: One of the famous and very influential techniques for dealing with small datasets is using a pre-trained network. A pre-trained network is a network that was trained on a vast dataset, usually in the task of categorizing images, and then its architecture and weights were preserved. If this initial dataset is big enough and general enough, the set of features that the pre-trained network has learned can be useful as a general visual model. Therefore, these features can help several different computer vision tasks, even if the new tasks may contain fully different categories from the initial task [38,39]. For example, networks that have been trained on the ImageNet database, such as DenseNet121 [40], can reset to something as remote as exploring medical image features. Transfer learning from a pre-trained network can be applied in two ways, namely feature extraction and fine-tuning. The Feature extraction involves taking the convolutional base of a pre-trained network to extract the new dataset features and then training a new classifier on top of these outputs. The fine-tuning is complementary to the feature extraction method, where it involves unfreezing the last layers of the frozen convolutional base utilized for the feature extraction. The unfrozen layers are then retrained in combination with the new classifier previously learned in the feature extraction method. The fine-tuning method aims to adjust the pre-trained model's most abstract features to make them more relevant to the new task. The steps for using these ways can be explained as follows [41]:

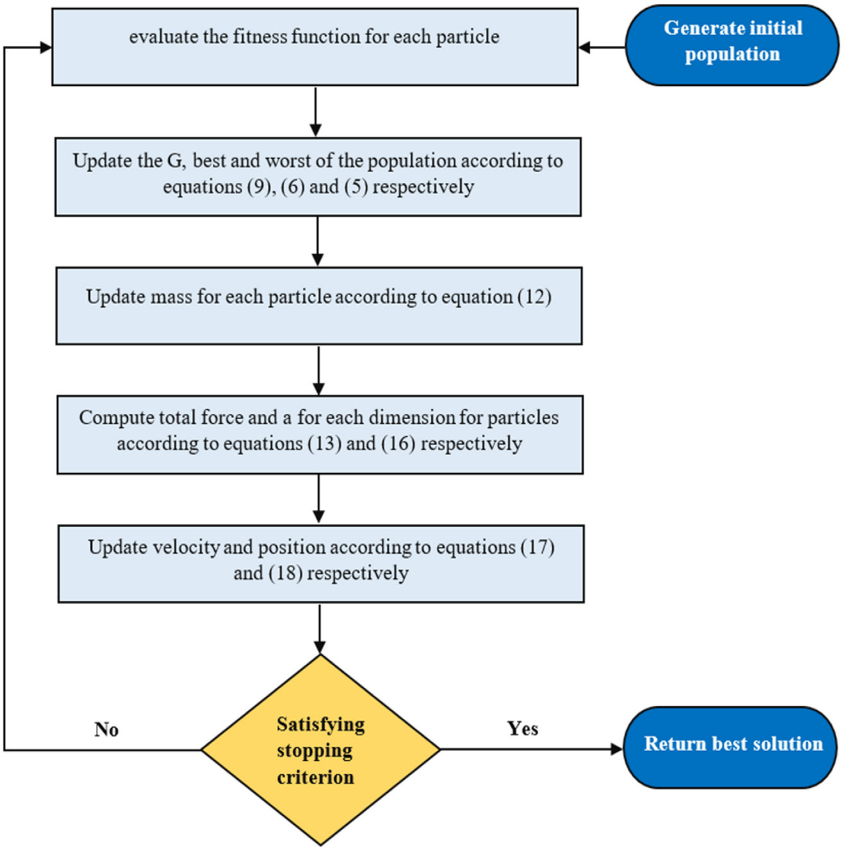

Generate initial population
evaluate the fitness function for each particle
Update the G, best and worst of the population according to equations (9), (6) and (5) respectively
Update mass for each particle according to equation (12)
Compute total force and a for each dimension for particles according to equations (13) and (16) respectively
Update velocity and position according to equations (17) and (18) respectively
Satisfying stopping criterion
No
Yes
Return best solution

Fig. 2. GSA flowchart.

Table 1
Number of cases of COVID-19, SARS, ARDS, Pneumocystis and Streptococcus, number of cases diagnosed with X-rays, and CT scans in each cause of pneumonia, and the total number of cases in the COVID19 Chest X-ray dataset.

The cause of pneumonia	Total number of samples	Number of cases diagnosed with X-ray images	Number of cases diagnosed with CT scans
COVID-19	121	99	22
SARS	11	11	–
ARDS	4	4	–
Pneumocystis	6	6	–
Streptococcus	2	2	–
Undefined	6	6	–
Total	**150**	**128**	**22**

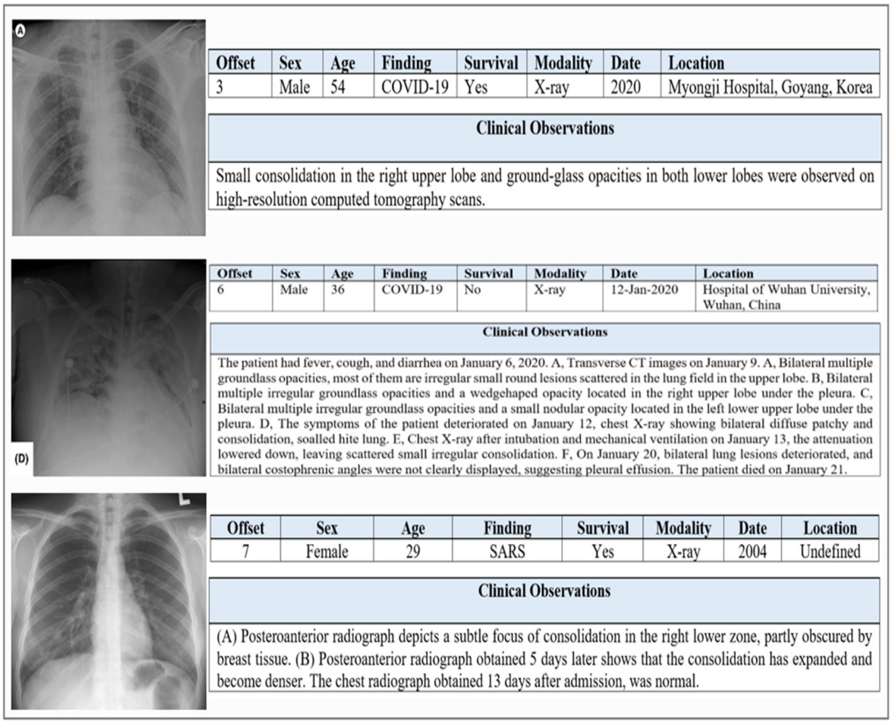
- A pre-trained network is taken, and its classifier base is removed.
- The convolutional base of the pre-trained model is frozen.
- A new classifier is added and trained on top of the convolutional base of the pre-trained network.
- Unfreeze some layers of the convolutional base of the pre-trained network.
- Finally, both these unfrozen layers and the new classifier are jointly trained.

3. Binary COVID-19 dataset description

The binary COVID-19 dataset used in this paper is a combination of two datasets. The first dataset is the COVID19 Chest X-ray dataset made available by Dr. Joseph Paul Cohen of the University of Montreal [42]. The COVID19 Chest X-ray dataset consists of 150 chest X-ray and CT images as of the time of writing this paper, 121 images of this dataset represent cases infected with the COVID-19. While 11 images represent cases infected with SARS, and four images represent cases infected with acute respiratory distress syndrome (ARDS). This dataset also contains five pneumocystis cases and six cases of streptococcus, as shown in Table 1.

The COVID19 Chest X-ray dataset contains many metadata for each image. The most important of which are: offset, sex, age, finding, survival, modality, date, location, and clinical observations about the radiograph in particular, not just the patient. The offset is the number of days since the onset of symptoms or hospitalization; the offset values ranged from 0 to 32. The ages of the patients enrolled in this dataset ranged from 12 to 87. While the finding field explains the cause of pneumonia, and the surviving field clarifies whether the patient is still alive or not. The modality defines how the diagnosis was made, either X-ray or CT scan. Fig. 3 shows some samples of the COVID19 Chest X-ray dataset and metadata for each sample.

The second dataset is the Kaggle Chest X-ray dataset made available for a Data Science competition [43]. This dataset consists of 5811 X-ray images, 1538 images represent normal cases, and 4273 images represent pneumonia cases. The binary COVID-19 dataset was built to distinguish COVID-19 cases from those suffering from other diseases and healthy cases using only X-ray images. The cases that were diagnosed using CT scans were



Offset	Sex	Age	Finding	Survival	Modality	Date	Location
3	Male	54	COVID-19	Yes	X-ray	2020	Myongji Hospital, Goyang, Korea

Clinical Observations

Small consolidation in the right upper lobe and ground-glass opacities in both lower lobes were observed on high-resolution computed tomography scans.

Offset	Sex	Age	Finding	Survival	Modality	Date	Location
6	Male	36	COVID-19	No	X-ray	12-Jan-2020	Hospital of Wuhan University, Wuhan, China

Clinical Observations

The patient had fever, cough, and diarrhea on January 6, 2020. A, Transverse CT images on January 9. A, Bilateral multiple groundlass opacities, most of them are irregular small round lesions scattered in the lung field in the upper lobe. B, Bilateral multiple irregular groundlass opacities and a wedgehaped opacity located in the right upper lobe under the pleura. C, Bilateral multiple irregular groundlass opacities and a small nodular opacity located in the left lower upper lobe under the pleura. D, The symptoms of the patient deteriorated on January 12, chest X-ray showing bilateral diffuse patchy and consolidation, soalled hite lung. E, Chest X-ray after intubation and mechanical ventilation on January 13, the attenuation lowered down, leaving scattered small irregular consolidation. F, On January 20, bilateral lung lesions deteriorated, and bilateral costophrenic angles were not clearly displayed, suggesting pleural effusion. The patient died on January 21.

Offset	Sex	Age	Finding	Survival	Modality	Date	Location
7	Female	29	SARS	Yes	X-ray	2004	Undefined

Clinical Observations

(A) Posteroanterior radiograph depicts a subtle focus of consolidation in the right lower zone, partly obscured by breast tissue. (B) Posteroanterior radiograph obtained 5 days later shows that the consolidation has expanded and become denser. The chest radiograph obtained 13 days after admission, was normal.

Fig. 3. Some images of the COVID19 Chest X-ray dataset, next to each image is its metadata.

excluded. The used dataset consists of two categories: positive and negative, as shown in Table 2. The positive category contains 99 X-ray images representing cases infected with the COVID-19, taken from the COVID19 Chest X-ray dataset. The negative category contains 207 X-ray images, 104 images in this category represent healthy cases, and 80 images represent pneumonia cases, taken from the Kaggle Chest X-ray dataset. The other 23 images in the negative category represent cases affected by SARS, ARDS, pneumocystis, or streptococcus, taken from the COVID19 Chest X-ray dataset. Some images of each category of the binary COVID-19 dataset are shown in Fig. 4.

4. The proposed GSA-DenseNet121-COVID-19 approach

The proposed approach GSA-DenseNet121-COVID-19, relies on the transfer learning from a pre-trained CNN architecture. The pre-trained architecture utilized in the proposed approach is DenseNet121. For this architecture's best performance, its hyperparameters have been optimized using the GSA. After determining the optimal values for these hyperparameters, DenseNet121 was trained using transfer learning techniques. Once this training is completed, it is evaluated using a separate test set. In other words, the training and validation sets were used to determine the optimal values for the hyperparameters of the DenseNet121 and trained it. Whereas the fully trained DenseNet121 is then evaluated using the test set. The proposed approach consists of four main stages, as shown in Fig. 5. The first stage is the data preparation, the second stage is the hyperparameters selection, the third stage is the learning, and the performance measurement is the fourth stage. Each stage will be explained in detail in the following sections.

4.1. Data preparation stage

As explained in the data description section, the positive category of the binary COVID-19 dataset contains 99 samples, while the negative category includes 207 samples, which means that this dataset is not balanced. In most cases, not all ML algorithms can handle this type of dataset well. Because most of the information available in this type of dataset belongs to the dominant category, making any ML algorithm learn to categorize the dominant class and not categorize the other minor category. Therefore, samples in the positive category have been increased by randomly copying some images after cutting each image. So that random copying does not cause the used CNN architecture to overfit the dataset. After that, the dataset became balanced, with each category containing 207 images.

The balanced binary COVID-19 dataset was divided into three sets: training set, validation set, and testing. The training set contains 70% of the dataset; that is, it has about 146 images in each category. While each of the validation and the test set contains 15% of the dataset samples; that is, each set includes 31 images in each category. Various data augmentation techniques [44] have been applied to increase the number of training samples to reduce overfitting and improve generalization. The data augmentation techniques used are brightness, rotation, width shift, height shift, shearing, zooming, vertical flip, and horizontal flip. Also, featurewise centering, featurewise standard deviation normalization and fill mode. Before the images were supplied to other stages, they were resized to a 180 x 180.

4.2. Hyperparameters selection stage

As previously reviewed in Section 2.2, the transfer learning method takes the same structure of the pre-trained network after making minor changes. The most crucial change is to replace the classifier with a new one, which requires changing the values of some hyperparameters or adding new ones. Examples of the hyperparameters that require modification are the batch size, the learning rat's value, and the number of neurons in the dense layer. The hyperparameter that may add is the rate of the dropout layer. In the proposed GSA-DenseNet121-COVID-19, three hyperparameters have been optimized, namely the batch size, the rate of the newly added dropout layer, and the number of the neurons of the

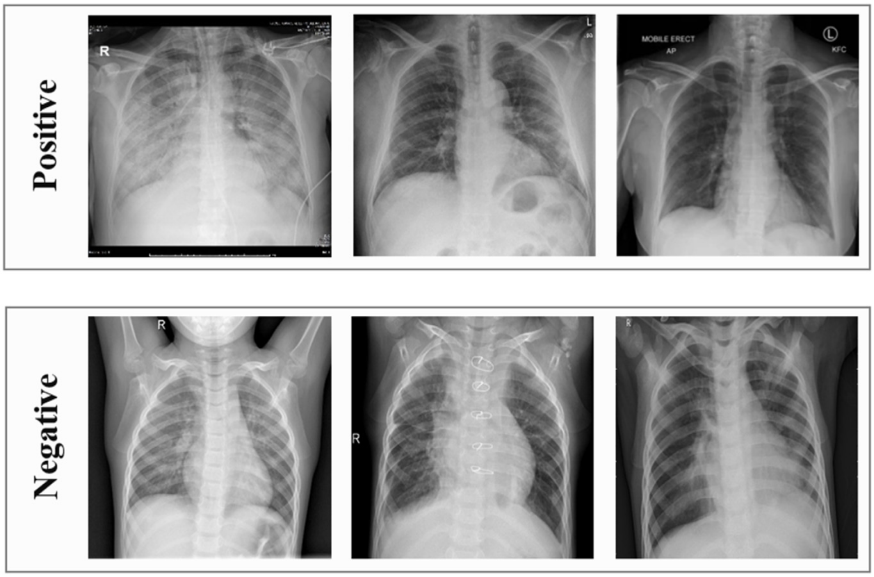


Fig. 4. Representative images of the positive and negative categories of the binary COVID-19 dataset.

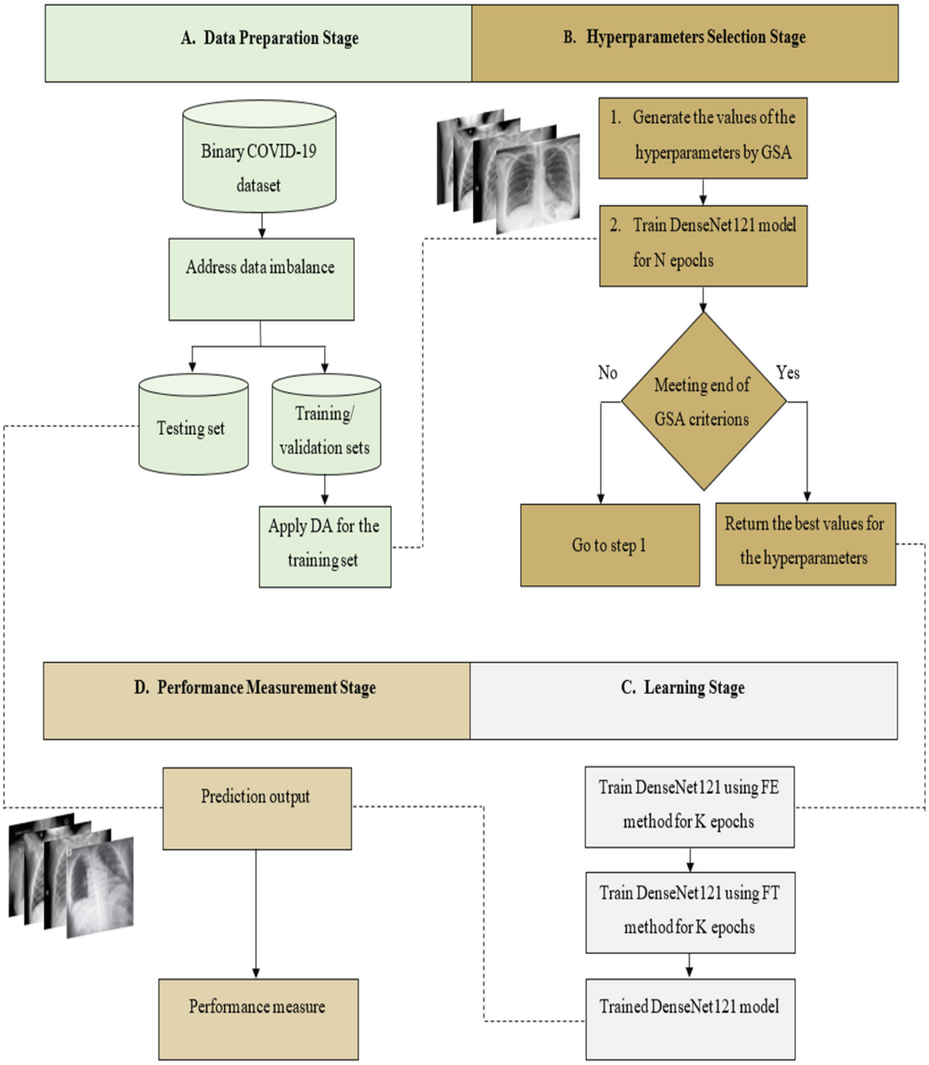

A. Data Preparation Stage
B. Hyperparameters Selection Stage
Binary COVID-19 dataset
Address data imbalance
Testing set
Training/ validation sets
Apply DA for the training set
1. Generate the values of the hyperparameters by GSA
2. Train DenseNet121 model for N epochs
Meeting end of GSA criterions
No
Yes
Go to step 1
Return the best values for the hyperparameters
D. Performance Measurement Stage
C. Learning Stage
Prediction output
Performance measure
Train DenseNet121 using FE method for K epochs
Train DenseNet121 using FT method for K epochs
Trained DenseNet121 model

Fig. 5. The structural form of the proposed approach GSA-DenseNet121-COVID-19 for the binary COVID-19 dataset, DA=Data Augmentation, FE=Feature Extraction, FT=Fine Tuning.

Table 2
Details of samples included in each category of the binary COVID-19 dataset in terms of sample type, number of samples, and sample source. For each category, the total number of samples is reported.

Category	Sample type	Number of samples	Sample source	Total number of samples in the category
Positive	COVID-19	99	COVID19 Chest X-ray dataset	99
Negative	Healthy	104	Kaggle Chest X-ray dataset	207
	Pneumonia	80	Kaggle Chest X-ray dataset	
	SARS	11	COVID19 Chest X-ray dataset	
	ARDS	4	COVID19 Chest X-ray dataset	
	Pneumocystis	6	COVID19 Chest X-ray dataset	
	Streptococcus	2	COVID19 Chest X-ray dataset	

first dense layer. Therefore, the search space is three-dimensional, and each point in the space represents a mixture of these three hyperparameters.

4.3. Learning stage

The feature extraction and fine-tuning techniques are utilized to prepare the DenseNet121 architecture to learn from the binary COVID-19 dataset. In the feature extraction, the convolutional base is kept unchanged, Whereas, the original classifier base is replaced by a new one that fits the binary COVID-19 dataset. The new classifier consists of four stacked layers: a flatten layer, and two dense layers separated by a new dropout layer. GSA determines the number of neurons in the first dense layer that uses RELU as an activation function and the dropout layer rate. The second dense layer has one neuron with a sigmoid function. After training the new classifier for some epochs, the fine-tuning is configured by retraining the last two blocks of the convolutional base of the DenseNet121 with the newly added classifier simultaneously.

4.4. Performance measurement stage

At this phase, the proposed approach is evaluated. Six measures are utilized to evaluate the proposed approach, namely accuracy, precision, recall, F1 score, and confusion matrix. Accuracy is among the foremost remarkably used measures for measuring the performance of classification models. It is outlined as a proportion between correctly classified samples to the overall number of samples as shown in Eq. (19). The error rate is the complement of the accuracy; it represents the samples that are misclassified by the model and calculated as Eq. (20) [45].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$Error\ rate = 1 - Accuracy = \frac{FP + FN}{TP + TN + FP + FN} \tag{20}$$

Where P= the number of the positive samples, and N= the numbers of the negative samples.

Precision as shown in Eq. (21), it is the number of true positives divided by the number of true positives and false positives. In other words, it is the number of positive predictions divided by the total number of positive category values predicted. Precision can be considered a measure of the rigor of a classifier. A low precision can also indicate a large number of false positives [46].

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

Recall, which also termed as sensitivity is the number of true positives divided by the number of true positives and the number of false negatives as shown in Eq. (22). In other words, it is the number of positive predictions divided by the number of positive class values in the test set. Recall can be considered a measure of how complete a classifier is. A low Recall indicates many false negatives [46].

Table 3
The augmentation techniques and the range of each technique.

Augmentation technique	Range
Shearing	0.2
Zooming	0.3
Width shift	0.4
Height shift	0.4
Rotation	15
Brightness	[0.5, 1.5]
Featurewise center	True
Featurewise standard deviation normalization	True
Fill mode	Reflect
Vertical flip	True
Horizontal flip	True

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

F1 score, which is also termed as F score, is a function of precision and recall and calculated as Eq. (23). It is used to seek a balance between precision and recall [46].

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{23}$$

Confusion Matrix is a synopsis of the prediction results regarding the classification problem. The confusion matrix gives insight into not only the mistakes committed by the classifier but, more importantly, the types of errors that are made [47].

5. Experiment results

This section presents and analyzes the results obtained through the proposed approach described in detail in Section 4. All the proposed approach procedures have been executed on Google Colaboratory [48] and implemented using Python with Keras [49]. Keras is a high-level neural network API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed for rapid use and the ability to conduct several experiments and get results as quickly as possible and the lowest delay, which helps to conduct adequate research. The results are divided into five sections to display clearly.

5.1. Conducting the data augmentation techniques

The Keras ImageDataGenerator was utilized to implement augmentation techniques to increase the number of images of the binary COVID-19 dataset's training set. The data augmentation techniques used and the range used for each technique are listed in Table 3. Fig. 6 illustrates some of the images obtained by applying augmentation techniques to one image from each category.

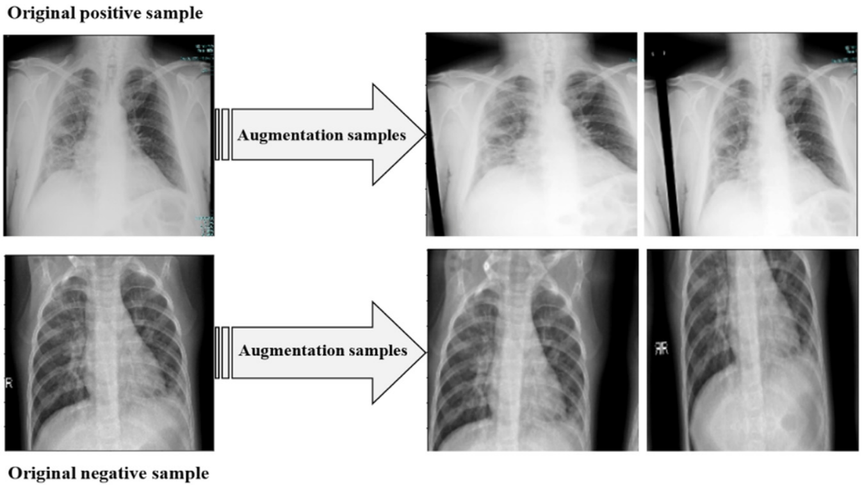


Fig. 6. Some samples generated from applying various augmentation techniques.

Table 4
Preparation of the required parameter values for GSA in conjunction with DensNet121.

Parameter	Value
Maximum number of iterations	15
Population size	30
Dimension	3
Batch size bounds	[1,64]
Dropout rate bounds	[0.1,0.9]
Number of neurons bounds	[50,500]
Maximum number of DenseNet121 training epochs	10

Table 5
Optimal values for the hyperparameters that were determined by GSA.

Hyperparameters	Optimal values
Batch size	8
Dropout rate	0.1
Number of neurons of the first dense layer	110

5.2. Setting up the GSA for the hyperparameters selection stage

The search space for hyperparameters whose values are to be set by the GSA was bounded as follows. The searching range of batch size was bounded by [1,64], and the searching space of the dropout rate was bounded by [0.1,0.9]. While the searching space of the number of neurons is bounded by [5,500], as listed in Table 4.

The GSA parameters' values were randomly specified, where the maximum number of iteration and population size set to 15 and 30, respectively, as listed in Table 4. The number of DenseNet121 training epochs was chosen by experimenting with more than one value. The experiment concluded that when using a number of epochs over ten, the training process for each of the GSA takes exponential time. While when using less than ten epochs, the results of the DenseNet121 were not sufficiently accurate. Therefore, the number of epochs used to train the DenseNet121 was set to ten epochs. The goal of using the GSA is to reduce the loss rate of the validation set as much as possible. The proposed GSA solutions' suitability is evaluated based on the achieved loss rate of the validation set using these proposed solutions after ten network training periods.

After completing the approximately 11-hour GSA training, the optimum values for the batch size, dropout rate, and number of neurons of the first dense layer were determined. Table 5 shows the optimal values for the hyperparameters selected by GSA, where the batch size, dropout and the number of neurons are 8, 0.1, 110 respectively.

5.3. Learning the DenseNet121 using the optimized hyperparameters

At this stage, the DenseNet121 was trained using the optimal values for the hyperparameters chosen by GSA. DenseNet121 architecture was trained on the training set and evaluated on the validation set for K number of epochs. To determine the value of K, several experiments were conducted, and it was found that the DenseNet121 achieved the best results on the validation set around the 30th epoch within the feature extraction method. While about the 40th epoch within the fine-tuning method and that no improvement was observed after that. Thus, the value of K was marked to 30 within the feature extraction and 40 within the fine-tuning. To minimize the overfitting, the process of the training was forced to finish before repetition K if no improvement was perceived for seven iterations, this control was made using early stopping [50]. As the COVID-19 dataset used is a binary-class classification problem, the DenseNet121 is compiled with the binary cross-entropy [51]. The Adam optimizer algorithm [52] was used with a constant learning rate =2e-5 within the feature extraction method. Within the fine-tuning method, a step decay schedule [53] was utilized, where the initial learning rate $LR_0 = 1e - 5$, and the value of the learning rate drops by 0.5 every 10 training epochs. The use of a low learning rate in the fine-tuning method is due to the fact that the number of changes that will occur in this method should be very small. So that the features learned from the feature extraction method are not lost.

5.4. Measuring the performance of GSA-DenseNet121-COVID-19

This section presents the results of the performance evaluation of the DenseNet121 architecture using hyperparameters values specified by the GSA. The performance of the proposed approach GSA-DenseNet121-COVID-19 was evaluated using accuracy, loss rate, precision, recall, and F1 score. The proposed approach achieved 98.38% accuracy in the test set, the average precision, recall, and f1 score were 98.5%, 98.5%, and 98%, respectively. The macro average and weighted average for the precision, recall, and F score was equal as the values for both were 98%, as listed in Table 6.

To find out the number of samples incorrectly classified by the proposed approach

GSA-DenseNet121-COVID-19. As well as the number of samples that it was able to classify correctly, the confusion matrix

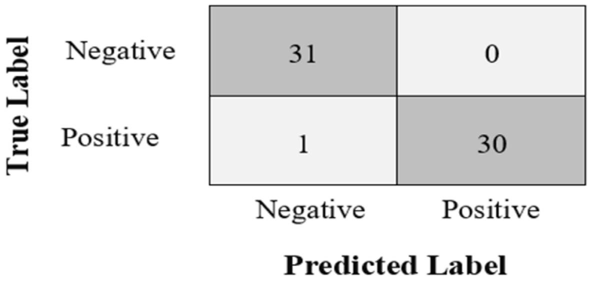


Fig. 7. The confusion matrix obtained from the proposed approach GSA-DenseNet121-COVID-19.

Table 6
The performance of the proposed approach GSA-DenseNet121-COVID-19 and the overall performance is calculated using macro and weighted average.

Categories	Precision	Recall	F1 score
Negative	97%	100%	98%
Positive	100%	97%	98%
Average	**98.5%**	**98.5%**	**98%**
Macro Average	**98%**	**98%**	**98%**
Weighted Average	**98%**	**98%**	**98%**

Table 7
Comparison of the proposed GSA-DenseNet121-COVID-19 performance with the performance of the SSD-GSA approach and Inception-v3. MA-Precision=macro average of precision, MA-Recall = macro average of recall, and MA-F score = macro average of F-score.

Comparison items	Proposed approach	SSD-DenseNet121	Inception-v3
Batch size	**8**	6	10
Dropout rate	**0.1**	0.71	0.5
Number of neurons	**110**	220	150
Accuracy	**98.38%**	94%	95%
MA-Precision	**98%**	94%	96%
MA-Recall	**98%**	94%	95%
MA-F Score	**98%**	94%	95%

was used as shown in Fig. 7. The dark-colored shaded cells of the confusion matrix represent samples that were correctly categorized in each category. Whereas, the light-colored shaded cells of the confusion matrix represent incorrectly categorized samples in each category.

As shown in the confusion matrix in Fig. 7, the proposed approach GSA-DenseNet121-COVID-19 erroneously classified only one sample from the test group, while it succeeded in classifying all other samples. Fig. 8 shows the results of the proposed approach for four images. The images surrounded by a green rectangle represent the images that the proposed approach correctly classified. While the image surrounded by the red rectangle represents the only sample that the proposed approach has incorrectly classified. The image in the red rectangle is an image that belongs to the positive category. However, the proposed approach classified this image as belonging to the negative category with a certainty of 83.8% as a first decision and to a positive category with a certainty of 16.2% as a second decision.

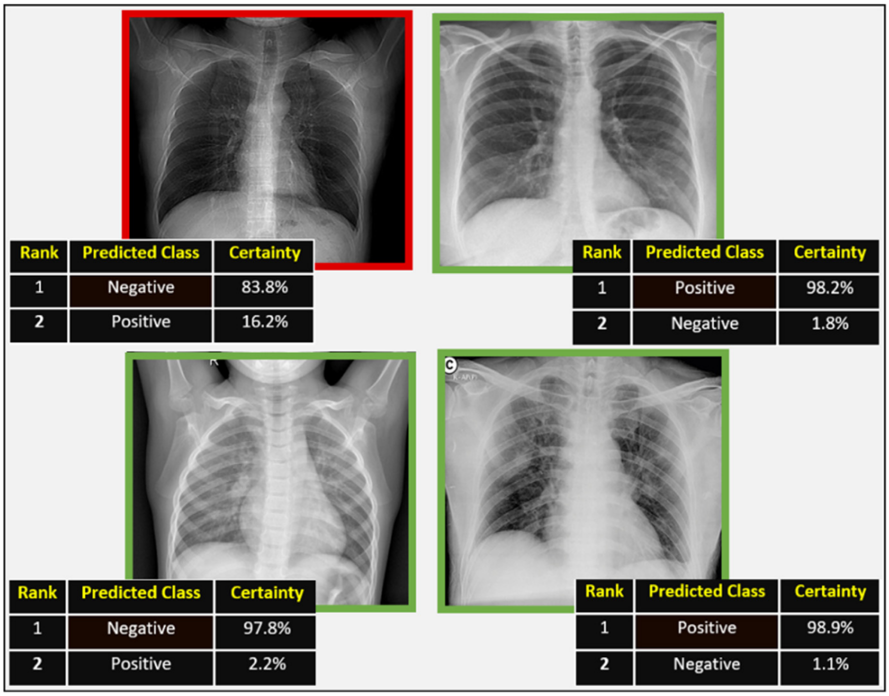
To know whether the proposed GSA-DenseNet121-COVID-19 is aware of X-rays as radiologists do, or whether it is learning unhelpful features to make predictions. The gradient weighted class activation mapping (Grad-CAM) [54] was used. As shown in Fig. 9, the Grad-CAM visualization of a positive sample and a negative sample, prove the effectiveness of the proposed approach in determining the important features relevant to each category.

5.5. Comparison with other methods and related work

To ensure the effectiveness of the GSA in determining optimum values for the hyperparameters of the DenseNet121 architecture that can achieve the highest level of accuracy. It has been compared with the SSD algorithm [55] that has been proven effective in determining optimal values for the hyperparameters of the CNN architecture used to detect the nanoscience scanning electron microscope images [25]. For a fair comparison between GSA and SSD algorithm. The values of the SSD algorithm parameters have been set with the same values that have been set for the GSA parameters, as shown in Table 4. After the SSD algorithm has completed the training process, the batch size value was set to 6, while the number of neurons and the dropout rate were set to 220 and 0.71, respectively. The comparative results showed that the GSA is more suitable for pairing with the DenseNet121 to classify the binary COVID-19 dataset. Where the GSA was able to choose better values for the hyperparameters of DenseNet121 architecture, which in turn made this architecture achieve a higher accuracy ratio. Where the approach SSD-DenseNet121 achieved an accuracy rate of 94% on the test set. As well as the macro average for the precision, recall, F1 score of the SSD-DenseNet121 were equal as the values for both were 94%, as listed in Table 7.

To ensure the performance of the proposed approach GSA-DenseNet121-COVID-19 as a whole. It was compared with the Inception-v3 architecture based on the manual search. The manual search indicates that the values of the Inception-v3 hyperparameters were randomly chosen. Where the values of batch size, dropout rate, and the number of neurons were set manually to 16, 0.5, 250, respectively. The results of this comparison showed that the proposed GSA-DenseNet121-COVID-19 was superior to the Inception-v3 architecture based on the manual search. Where, the accuracy of the Inception-v3 architecture is 95%, while the macro average precision, recall, and F1 score of this architecture are 95%, 96%, 95%, 95%, respectively, as shown in Table 7.

The proposed GSA-DenseNet121-COVID-19 performance was also compared with other published approaches that were introduced for the same purpose of diagnosing COVID-19 using X-ray images. The other approaches included in [14–17,19] were selected for comparison, as they relied on CNN architectures and were trained on a variety of data samples. The proposed approach has been compared to other approaches in terms of the number and variety of samples used, accuracy, precision, recall, and f score, as shown in Table 8. In [14], many of the CNN architectures were evaluated to classify two datasets. However, the best performance was for the MobileNet architecture on the second dataset containing 224 samples of COVID-19, 504 healthy samples, and 714 samples of pneumonia. The MobileNet obtained an accuracy, precision, and recall rate of 96.78%, 96.46%, and 98.66%, respectively, in the second dataset classification when it was treated as a binary classification problem, as shown in the second row of Table 8. In [15], the proposed DarkCovidNet was able to be more accurate by classifying the two categories more than classifying the three categories, as shown in rows 3 and 4 of Table 8. Whereas in [16], after the CNN-SA approach was trained in healthy and COVID-19 cases only, it was able to achieve a rate of 96% for both accuracy, precision, recall, and f score. The CoroNet proposed in [17] has performed better in the binary classification that includes only healthy and COVID-19 cases, than the multi-category classification that includes various types of samples, as shown in rows 5, 6, and 7 of Table 8. While the Deep Bayes-SqueezeNet approach [19], having been trained on a relatively large number of samples, was able to achieve 97% for both accuracy, precision, recall, and f score, as shown in the eighth row of Table 8.



Rank	Predicted Class	Certainty
1	Negative	83.8%
2	Positive	16.2%

Rank	Predicted Class	Certainty
1	Positive	98.2%
2	Negative	1.8%

Rank	Predicted Class	Certainty
1	Negative	97.8%
2	Positive	2.2%

Rank	Predicted Class	Certainty
1	Positive	98.9%
2	Negative	1.1%

Fig. 8. Illustrative examples of some correctly labeled samples (enclosed in a green rectangle) and the incorrectly labeled sample (enclosed in a red rectangle) from the test set.

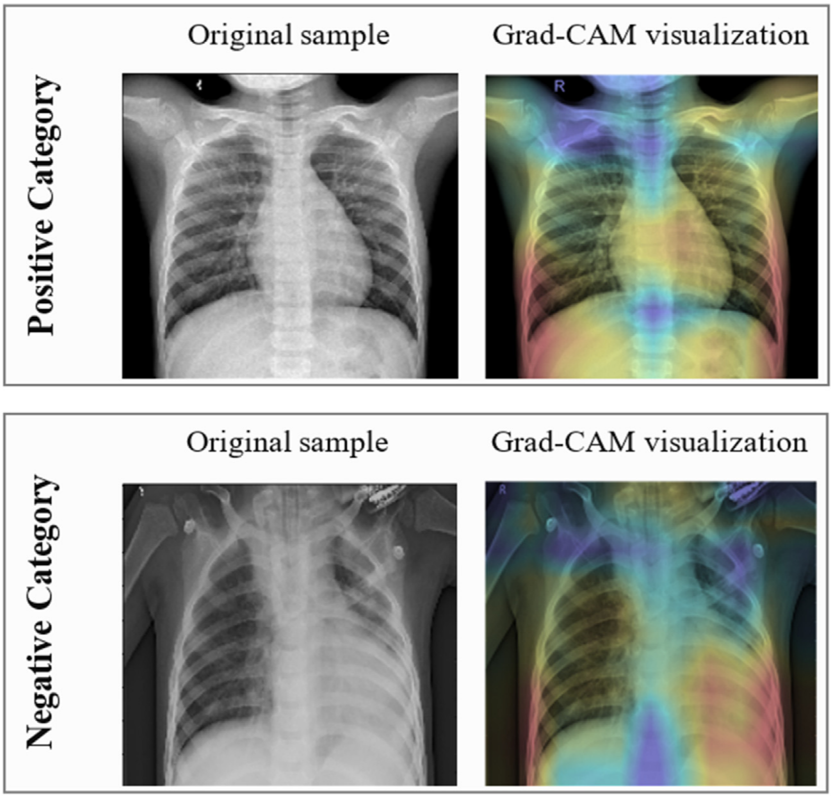


Fig. 9. The Grad-CAM visualization of a positive sample and a negative sample.

Table 8 shows that although the proposed GSA-DenseNet121-COVID-19 has been trained in a smaller and more diverse number of samples than its counterparts. The proposed GSA-DenseNet121-COVID-19 managed to outperform both DarkCovid-Net and CNN-SA. Likewise, the results of GSA-DenseNet121-COVID-19 outperformed the results of MobileNet in terms of accuracy and precision but were slightly smaller in the recall. The proposed GSA-DenseNet121-COVID-19 is very competitive with the Deep Bayes-SqueezeNet. Since the results of GSA-DenseNet121-COVID-19 are better than the results of Deep Bayes-SqueezeNet in terms of precision and recall, they are approximately equal in accuracy and slightly less in result f score.

Table 8
A comparison between the results of the proposed approach and the best results achieved by the other proposed approaches listed in [14–17,19].

Approach	Number of original samples	Number of classes	Accuracy	Average precision	Average Recall	Average F score
MobileNet [14]	224 COVID-19 504 Healthy 714 Pneumonia (400 bacterial + 314 viral)	2 classes	96.78%	96.46%	98.66%	–
DarkCovidNet [15]	125 COVID-19 500 No- Findings	2 classes	98.08%.	98.03%	95.13%	96.51%
DarkCovidNet [15]	125 COVID-19 500 No- Findings 500 Pneumonia	3 classes	87.02%	89.96%	85.35%	87.37%
CNN-SA [16]	403 COVID-19 721 Normal	2 classes	95%	95%	95%	95%
CoroNet [17]	284 COVID-19 310 Normal 330 Pneumonia Bacterial 327 Pneumonia Viral	4 classes	89.6%	90%	89.92%	89.8%
CoroNet [17]	284 COVID-19 310 Normal 657 Pneumonia (330 bacterial + 327 viral)	3 classes	95%	95%	96.9%	95.6%
CoroNet [17]	284 COVID-19 310 Normal	2 classes	99%	98.3%	99.3%	98.5%
Deep Bayes-SqueezeNet [19]	76 COVID-19 4290 Pneumonia (bacterial + viral) 1583 Normal	3 classes	98.3%	98.3%	98.3%	98.3%
Proposed GSA-DenseNet121-COVID-19	**99 COVID-19** **11 SARS** **4 ARDS** **6 Pneumocystis** **2 Streptococcus** **104 Healthy** **80 pneumonia**	**2 classes**	**98.38%**	**98.5%**	**98.5%**	**98%**

The DenseNet121-COVID-19 was superior to the CoroNet when the latter was trained on a variety of samples, just as the DenseNet121-COVID-19 was trained.

6. Conclusions and future work

This paper proposes an approach called GSA-DenseNet121-COVID-19 that can be used to diagnose COVID-19 cases through chest X-ray images. The proposed GSA-DenseNet121-COVID-19 consists of four main stages are (1) data preparation stage, (2) the hyperparameters selection stage, (3) the learning stage, (4) the performance measurement stage. In the first stage, the binary COVID-19 dataset was handled from the imbalance and then divided into three sets, namely training set, validation set, and test set. After increasing the number of samples of the training set in the first stage using different data augmentation techniques, they were used in the second stage with the validation set. In the second stage, GSA is used to optimize some of the hyperparameters in the CNN architecture used which is called DenseNet121. In the third stage, DenseNet121 was completely trained using the values of the hyperparameters that were identified in the previous stage which in turn helped this architecture to diagnose 98.38% of the test set in the fourth stage. The proposed approach was compared to more than one approach, and the results of the comparison showed the effectiveness of the proposed approach in diagnosing the COVID-19. In future work, the number of samples used to train the proposed approach can be increased to improve its performance in the diagnosis of COVID-19. In addition, the number of other diseases causing pneumonia may be increased and the proposed approach can be used to distinguish them from the COVID-19.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Vivaldo, et al., The emergence of SARS, MERS and novel SARS-2 coronaviruses in the 21st century, Arch. Virol. 165 (7) (2020) 1517–1526.
[2] R.A. Fouchier, et al., Aetiology: Koch's postulates fulfilled for SARS virus, Nature 423 (6937) (2003) 240–240.
[3] A.M. Zaki, et al., Isolation of a novel coronavirus from a man with pneumonia in Saudi Arabia, New Engl. J. Med. 367 (19) (2012) 1814–1820.
[4] Centers for Disease Control and Preventio, Principles of Epidemiology, second ed., U.S. Department of Health and Human Services, Atlanta, 1992, [Online]. Available: https://www.cdc.gov/csels/dsepd/ss1978/lesson1/section9.html.
[5] J.P. Broughton, et al., CRISPR–Cas12-based detection of SARS-CoV-2, Nature Biotechnol. 38 (2020) 870–874.
[6] T. Alireza, et al., Real-time RT-PCR in COVID-19 detection: issues affecting the results, Expert Rev. Mol. Diagnost. 20 (5) (2020) 1–2.
[7] M. Talo, U.B. Baloglu, Ö. Yıldırım, U.R. Acharya, Application of deep transfer learning for automated brain abnormality classification using MR images, Cogn. Syst. Res. 54 (14) (2019) 176–188.
[8] S.A.A. Ismael, A. Mohammed, H. Hefny, An enhanced deep learning approach for brain cancer MRI images classification using residual networks, Artif. Intell. Med. 102 (2020) 101779.
[9] N.E.M. Khalifa, M.H.N. Taha, D.E. Ali, A. Slowik, A.E. Hassanien, Artificial intelligence technique for gene expression by tumor RNA-seq data: A novel optimized deep learning approach, IEEE Access 8 (2020) 22874–22883.
[10] C. Ho, N. Jean, C.A. Hogan, L. Blackmon, S.S. Jeffrey, M. Holodniy, N. Banaei, A.A.E. Saleh, S. Ermon, J. Dionne, Rapid identification of pathogenic bacteria using Raman spectroscopy and deep learning, Nature Commun. 10 (1) (2019) 1–8.
[11] I.R.I. Haque, J. Neubert, Deep learning approaches to biomedical image segmentation, Inf. Med. Unlocked 18 (2020) 100297.
[12] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[13] B.J. Erickson, P. Korfiatis, T.L. Kline, Z. Akkus, K. Philbrick, A.D. Weston, Deep learning in radiology: Does one size fit all?, J. Am. Coll. Radiol. 15 (3) (2018) 521–526.

[14] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. 43 (2) (2020) 635–640.

[15] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (2020) 103792.

[16] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, P.R. Pinheiro, CovidGAN: Data augmentation using auxiliary classifier GAN for improved Covid-19 detection, IEEE Access 8 (2020) 91916–91923.

[17] A.I. Khan, J.L. Shah, M.M. Bhatc, CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images, Comput. Methods Programs Biomed. 196 (2020) 105581.

[18] F. Ucara, D. Korkmazb, COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, Med. Hypotheses 140 (2020) 109761.

[19] A. Koutsoukas, K.J. Monaghan, X. Li, Jun Huan, Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data, J. Cheminformat. 9 (42) (2017).

[20] S. Albelwi, A. Mahmood, A framework for designing the architectures of deep convolutional neural networks, Entropy 19 (6) (2017) 242.

[21] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (1) (2012) 281–305.

[22] S.R. Young, D.C. Rose, T.P. Karnowski, S. Lim, R.M. Patton, Optimizing deep learning hyper-parameters through an evolutionary algorithm, in: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments (MLHPC 2015), ACM, Austin, Texas, 2015, pp. 1–5.

[23] Y. Yoo, Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches, Knowl.-Based Syst. 178 (2019) 74–83.

[24] A. Darwish, D. Ezzat, A.E. Hassanien, An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, Swarm Evol. Comput. 52 (2020) 100616.

[25] D. Ezzat, M.H.N. Taha, A.E. Hassanien, An optimized deep convolutional neural network to identify nanoscience scanning electron microscope images using social ski driver algorithm, in: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019 (AISI 2019), Springer, Cairo, Egypt, 2020, pp. 492–501.

[26] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inform. Sci. 179 (13) (2009) 232–2248.

[27] D. Holliday, R. Resnick, J. Walker, Fundamentals of Physics, John Wiley and Sons, 2013.

[28] B.F. Schutz, Gravity from the Ground Up: An Introductory Guide to Gravity and General Relativity, Cambridge University Press, 2003.

[29] R. Mansouri, F. Nasseri, M. Khorrami, Effective time variation of G in a model universe with variable space dimension, Phys. Lett. 259 (3–4) (1999) 194–200.

[30] X. Yuana, B. Ji, Shuangquan. Zhang, H. Tiana, Y. Houa, A new approach for unit commitment problem via binary gravitational search algorithm, Appl. Soft Comput. 22 (2014) 249–260.

[31] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural Comput. 29 (9) (2017) 2352–2449.

[32] J. Jang, H. Cho, J. Kim, J. Lee, S. Yang, Deep neural networks with a set of node-wise varying activation functions, Neural Netw. 126 (2020) 118–131.

[33] V. Suárez-Paniagua, I. Segura-Bedmar, Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction, BMC Bioinformatics 19 (8) (2018) 39–47.

[34] Y. Wang, Y. Li, Y. Song, X. Rong, The influence of the activation function in a convolution neural network model of facial expression recognition, Appl. Sci. 10 (5) (2020) 1897.

[35] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[36] N. Srivastava, G. Hinton, A. Krizhevs.ky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[37] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, vol. 1, MIT Press, Cambridge, 2016.

[38] A. Lumini, L. Nanni, Deep learning and transfer learning features for plankton classification, Ecol. Inform. 51 (2019) 33–43.

[39] J.G.A. Barbedo, Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification, Comput. Electron. Agric. 153 (2018) 46–53.

[40] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2261–2269.

[41] R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, Insights Imaging 9 (4) (2018) 611–629.

[42] COVID-19 image data collection, 2020, [Online]. Available: https://github.com/ieee8023/covid-chestxray-dataset. (Accessed 25 March 25 2020).

[43] Chest X-ray images (Pneumonia), 2020, [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. (Accessed 25 March 25 2020).

[44] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60.

[45] A. Tharwat, Classification assessment methods, Appl. Comput. Inf. (2018).

[46] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation, in: Proceedings of European Conference on Information Retrieval, Springer, Berlin, Heidelberg, 2005, pp. 345–359.

[47] K.M. Ting, Confusion Matrix, in: Encyclopedia of Machine Learning, Springer, Boston, MA, USA, 2011, p. 209.

[48] T. Carneiro, R.V. Medeiros Da NóBrega, T. Nepomuceno, G. Bian, V.H.C. De Albuquerque, P.P.R. Filho, Performance analysis of google colaboratory as a tool for accelerating deep learning applications, IEEE Access 6 (2018) 61677–61685.

[49] F. Chollet, Keras: Deep learning for humans, 2015, [Online]. Available: https://github.com/fchollet/keras.

[50] L. Prechelt, Early stopping - but when?, in: Neural Networks: Tricks of the Trade, second ed., 2012, pp. 53–67.

[51] A.S. Bosman, A. Engelbrecht, M. Helbig, Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions, Neurocomputing 400 (2020) 113–136.

[52] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, [Online]. Available: arXiv:1412.6980.

[53] A. Senior, G. Heigold, M. Ranzato, K. Yang, An empirical study of learning rates in deep neural networks for speech recognition, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6724–6728.

[54] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[55] A. Tharwat, T. Gabel, Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm, Neural Comput. Appl. (2019) 1–14.