An automated diagnosis and classification of COVID-19 from chest CT images using a transfer learning-based convolutional neural network

Nadiah A. Baghdadi [a,1], Amer Malki [b], Sally F. Abdelaliem [a], Hossam Magdy Balaha [c], Mahmoud Badawy [c,*], Mostafa Elhosseini [b,c]

[a] *Princess Nourah bint Abdulrahman University, College of Nursing, Riyadh, 11671, Riyadh, P.O. BOX 84428, Saudi Arabia*
[b] *Taibah University, College of Computer Science and Engineering, Yanbu, 46421, Saudi Arabia*
[c] *Mansoura University, Faculty of Engineering, Computers and Control Systems Engineering Department, Mansoura, 46421, Egypt*

ARTICLE INFO

ABSTRACT

Researchers have developed more intelligent, highly responsive, and efficient detection methods owing to the COVID-19 demands for more widespread diagnosis. The work done deals with developing an AI-based framework that can help radiologists and other healthcare professionals diagnose COVID-19 cases at a high level of accuracy. However, in the absence of publicly available CT datasets, the development of such AI tools can prove challenging. Therefore, an algorithm for performing automatic and accurate COVID-19 classification using Convolutional Neural Network (CNN), pre-trained model, and Sparrow search algorithm (SSA) on CT lung images was proposed. The pre-trained CNN models used are SeresNext50, SeresNext101, SeNet154, MobileNet, MobileNetV2, MobileNetV3Small, and MobileNetV3Large. In addition, the SSA will be used to optimize the different CNN and transfer learning(TL) hyperparameters to find the best configuration for the pre-trained model used and enhance its performance. Two datasets are used in the experiments. There are two classes in the first dataset, while three in the second. The authors combined two publicly available COVID-19 datasets as the first dataset, namely the COVID-19 Lung CT Scans and COVID-19 CT Scan Dataset. In total, 14,486 images were included in this study. The authors analyzed the Large COVID-19 CT scan slice dataset in the second dataset, which utilized 17,104 images. Compared to other pre-trained models on both classes datasets, MobileNetV3Large pre-trained is the best model. As far as the three-classes dataset is concerned, a model trained on SeNet154 is the best available. Results show that, when compared to other CNN models like LeNet-5 CNN, COVID faster R–CNN, Light CNN, Fuzzy + CNN, Dynamic CNN, CNN and Optimized CNN, the proposed Framework achieves the best accuracy of 99.74% (two classes) and 98% (three classes).

1. Introduction

The COVID-19, the infectious disease caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) is the deadliest pandemic ever crept on humanity. The COVID-19 is still spreading rapidly throughout the real world, instilling fear in people's ability to communicate physically and a threat to global public health. The COVID-19 pandemic impacts everyone's life in a million ways with strict quarantine, travel restrictions, isolation measures, and causing many shutdowns in different sectors worldwide. The world is waiting for the end of the COVID-19 epidemic and its moral implications, which have changed human society over the last two years. However, the virus persists and exhibits different patterns.

The terrible and rapid spread of the third wave of pandemic and the emerging highly transmissible corona virus's new Delta, Omicron, and Ihu raised many concerns worldwide. These variants present a challenge in dealing with the COVID-19 pandemic [1]. Governments have begun to fund COVID-19 vaccine development and research [2]. However, the world does not know how long a vaccine would stay effective as the virus mutates. The Delta mutant causes significant lung damage and severe breathing difficulties, leading to death. While Omicron poses less threat than previous coronavirus strains, its spreading speed causes fear
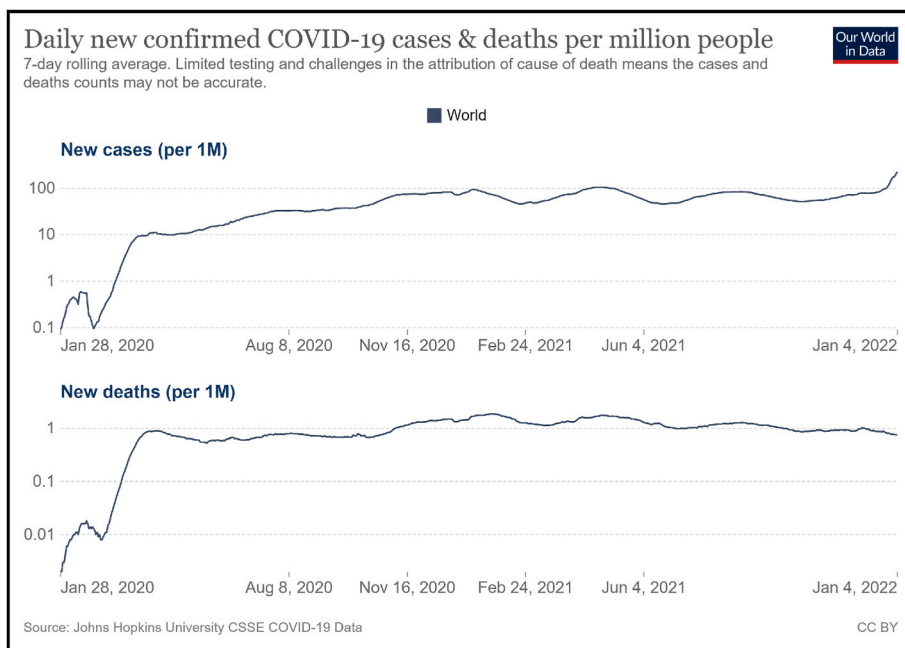

Fig. 1. Worldwide daily new confirmed COVID-19 cases and deaths per million people [4].
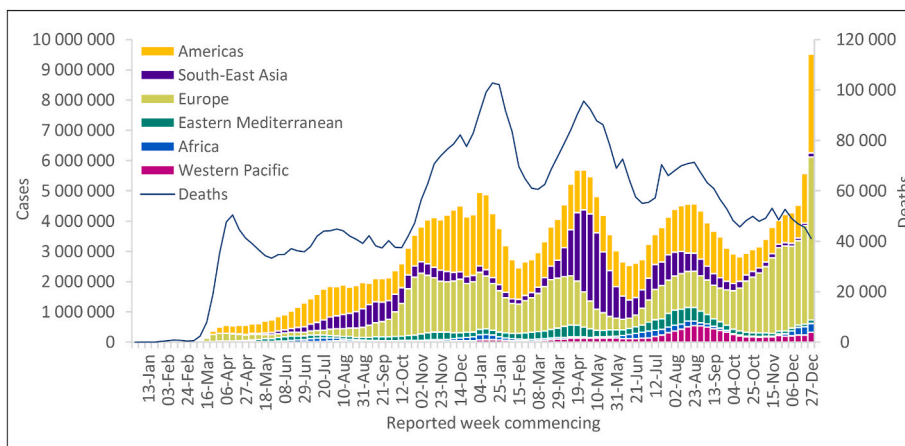

Fig. 2. COVID-19 cases reported weekly by WHO Region, and global deaths, as of 2 January 2022 [3].

and panic. The Omicron carries more than 50 mutations detected in roughly 100 countries. The number of COVID-19 cases involving Omicron is doubling every 1.5 to three days [3]. The sharp increase in omicron infections worldwide may increase the likelihood of a new, more dangerous mutation. Besides, The new "Ihu" mutant hit the French city of Marseille and spread rapidly. Ihu eludes the immune system with 46 mutations and has a high ability to spread and resist vaccines. The world is afraid of returning to the early days of the coronavirus when a terrible number of people died worldwide. Since the virus's first appearance in China in December 2019, over 289 million cases have been infected, including over 5.4 million deaths, as shown in Fig. 1 [4].

Despite extensive research being conducted to develop a vaccine, many immune countries are now experiencing record levels of a new infection. As shown in Fig. 2 [3], the global number of new cases increased by 71%, while the number of new deaths decreased by 10%. According to WHO weekly report (2nd of January 2022), there are approximately 9.5 million new cases and over 41000 new deaths. When we see such an increase in cases, more people with severe symptoms will likely end up in the hospital and possibly die. Even in capable and developed healthcare systems, real challenges are emerging at the

moment. Thus, accurate and rapid screening of infected patients is essential, and this is considered the primary stage for fighting this global pandemic and "flattening the curve" of the coronavirus pandemic.

With this pandemic explosion, it is necessary to combat the spread of COVID-19 in the early stages. Therefore, there is a crucial need for timely and accurate COVID-19 diagnosis tools to increase the effectiveness of patient care, proper treatment planning, and quarantine precautions. Fig. 3 shows four common methods for detecting COVID-19 [5]. The swab-based reverse transcription-polymerase chain reaction (RT-PCR) is the accepted standard screening strategy known as a microbiological test for diagnosing the presence of COVID-19. However, the method is laborious, has long turnaround times (takes 4–6 h) to furnish a result, is limited available, has low detection sensitivity in the initial stages, is tedious and time-consuming. Additionally, RT-PCR shows false-negative rates as large as 15%–20% [6]. Because COVID-19 is a contagious disease, many developing countries cannot provide RT-PCR test kits, especially on a large scale.

Due to the limitation of the swab test, the developed countries use routine blood tests as an inexpensive solution that may help identify COVID-19 patients [6]. In addition, blood sample-based tests can be
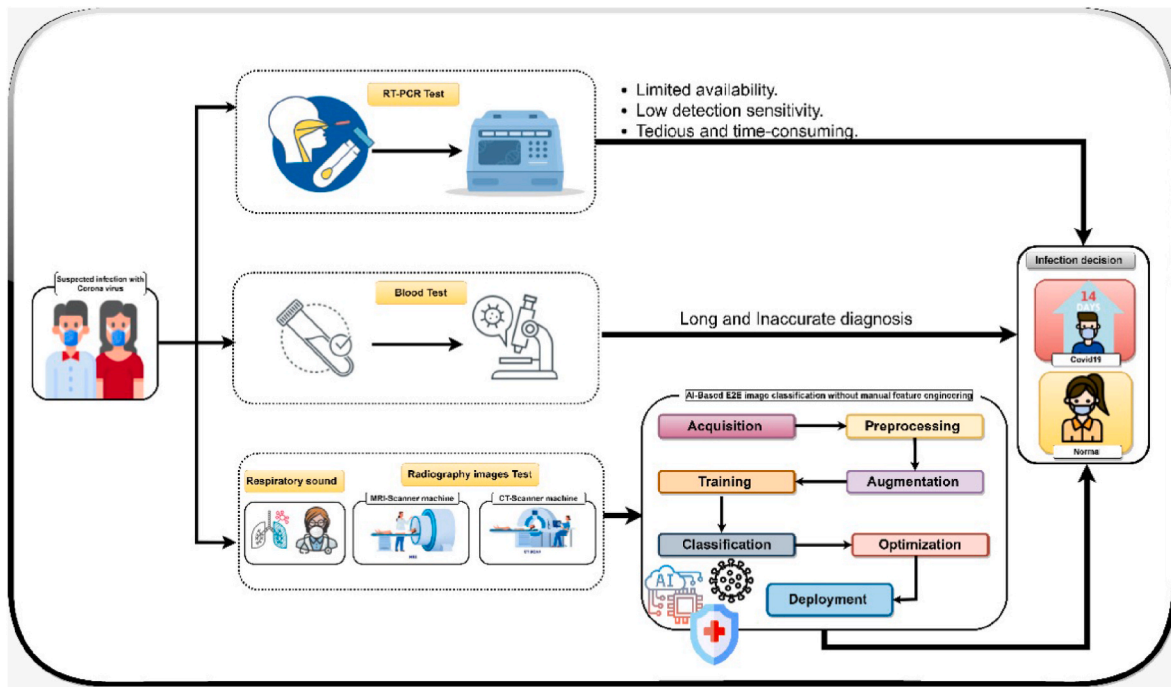
Fig. 3. COVID-19 diagnosis techniques.
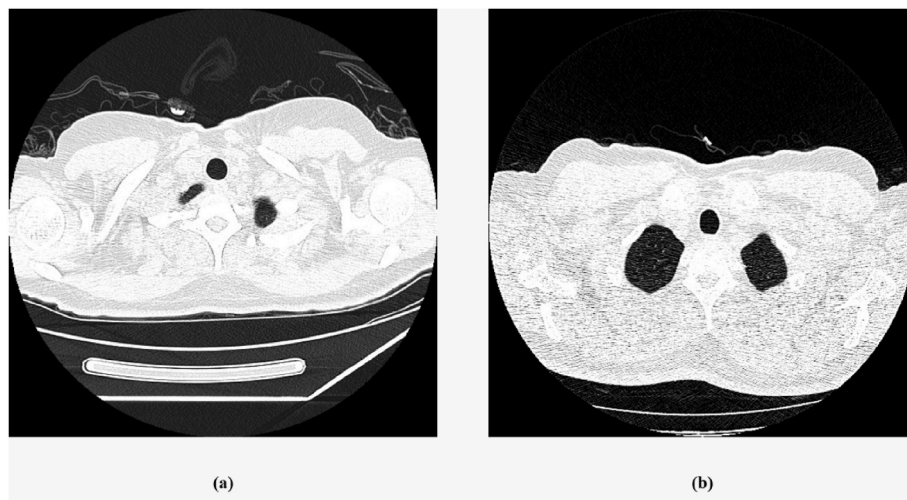
Fig. 4. CCT scan of lungs of a patient (a) affected by COVID-19 and (b) not affected by COVID-19.

used as serology tests to help estimate how many people have already been infected (antibody test) [7]. The use of human respiratory sound can also be used as a diagnostic tool for detecting COVID-19 from human-generated sounds such as voice/speech, dry cough, and breath [8]. With many cases emerging every day, health systems in all countries are collapsing. Accordingly, Chest Computed Tomography (CCT) and chest X-ray (CXR) radiography images have been recently used as a useful diagnostic tool for COVID-19. The radiography images are characterized by less complexity, availability, and faster diagnosis. CXR-ray is less expensive; however, its performance in COVID-19 screening is weaker compared to CCT as less information is embedded in a CXR scan image. Fig. 4 (a) and (b) depict two CCT scans of COVID-19 and non-COVID-19, respectively.

Nevertheless, CCT has played a vital role in diagnosis during this pandemic, demonstrating typical radiographic features of most COVID-19 infected patients. In addition, CCT imaging is a valid alternative to

detect COVID-19 with a higher sensitivity up to 98% compared with 71% of RT-PCR [9]. One hundred forty laboratories-confirmed that COVID-19 patients had positive CCT results in the early stages, according to Ref. [10]. However, radiologists have to be experienced in medical imaging analysis and interpretation to decipher the radiography images.

Furthermore, due to the rapid spread of COVID-19 outbreaks, hospitals have long queues for CCT scan image examination. Therefore, there is a high infection risk of spreading to other patients. In addition, the medical staff has to evaluate many CCT images quickly, which causes an overburdened medical system. The healthcare systems can be disrupted or completely breakdown due to limitations of RT-PCR kits, the burden of specialized radiologists, and intensive care equipment availability for hospitals. Automatic, highly responsive, accurate, and scalable detection of COVID-19 patients is still a major problem and a crucial point for global health concerns. Therefore, intelligent

Fig. 5. The suggested framework.

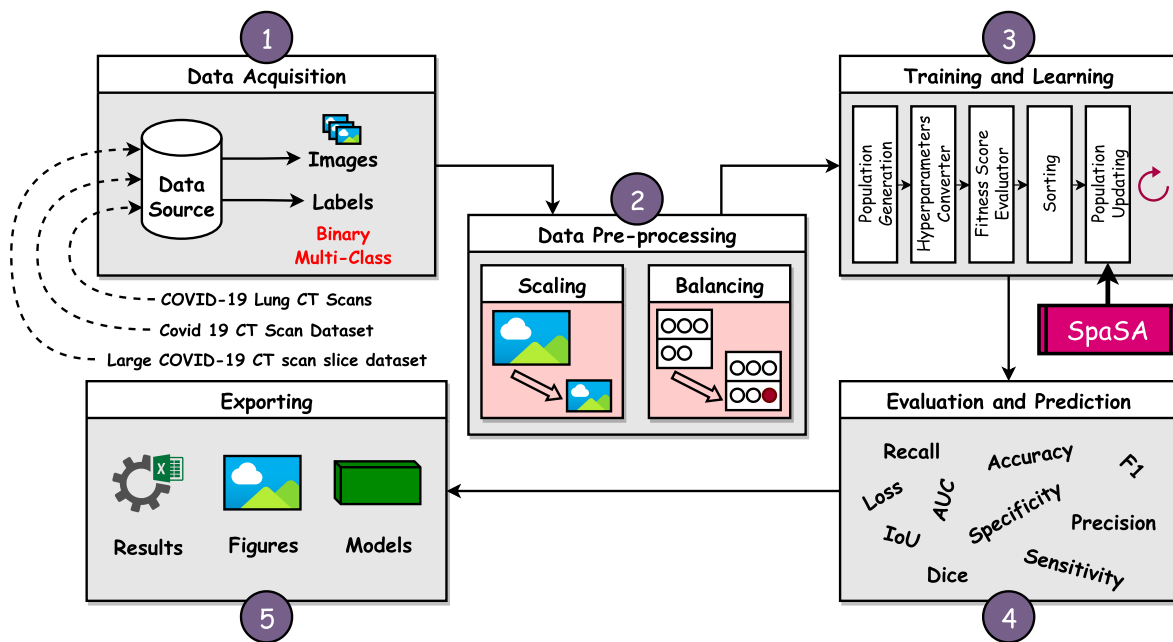approaches are needed to support the healthcare system and automatically classify CCT images. Therefore, there is a need for feasible alternative methods to support the medical aspects in automatic detecting COVID-19 in early stages that achieves optimal tradeoff between cost, testing accuracy, and consumed time. Therefore, the cooperation between the medical and computer-engineering researchers is crucial for developing Computer-Aided Detection approaches that efficiently diagnose COVID-19 faster and less laborious [11] to alleviate the pressure on healthcare systems.

Artificial Intelligence (AI), specifically machine learning, deep learning (DL), and transfer learning (TL) diagnostics techniques, has recently gained popularity in the medical field. This is because it enables end-to-end image classification without human intervention. However, despite the existence of a variety of COVID-19-AI-based diagnosis techniques, the desired diagnosis accuracy has yet to be achieved due to: (1) limitations of training data available for the research community, (2) data imbalances, (3) variation in image quality, (4) the classification performance, (5) time and space complexity, (6) insufficient generalisability, and 7) the optimization of the huge number of existing hyperparameters. Thus developing an intelligent, highly responsive, and automated COVID-19 diagnosis model is challenging.

The primary motivations for this study are: (1) The rapid spread of COVID-19 and the crucial need for the early detection to limit the occurrence of COVID-19 among individuals. (2) The RT-PCR tests have limited availability and require a significant amount of time. (3) Medical imaging modalities have an important role in automatically diagnosing COVID-19 patients, particularly infected children and pregnant women [12]. (4) The CAD systems based on deep learning strategies motivate the need for more accurate automated classification approaches for rapidly diagnosing COVID-19 patients. (5) Datasets availability limits DL networks training. (6) Deploying optimization strategies to choose the best model architectures and hyper-parameters.

The novel feature of the study is the use of transfer learning-based CNN's whose hyperparameters are optimized using sparrow search for automatic diagnosis and classification of COVID-19 from chest CT images. In addition, depending on the CNN and TL hyperparameters, the SpaSA algorithm is used to choose the best-pretrained model among all recommended models and the best optimal settings of model hyperparameters. It is beneficial to re-use a pretrained network and transfer an already-learned model to a new model using transfer learning. The

results show that pre-trained CNNs for MobileNetV3Large and SeNet154 deliver optimal or near-optimal results when used to train binary classification classifiers and multiclassification classifiers, respectively.

This study proposed a framework to perform automatic classification of COVID-19 based on CT lung images with the help of Convolutional Neural Network (CNN) and the Sparrow Search Algorithm (SpaSA) for hyperparameters optimization. Furthermore, this study proposes adapting the SpaSA [13] to improve and optimize the CNN network classification to obtain more accurate results. SpaSA is a swarm optimization approach inspired by sparrows' group wisdom, foraging, and anti-predation behaviors. The SpaSA outperforms other optimization algorithms regarding search speed, precision, convergence rate, stability, and local optimal value avoidance. The current study contributions can be summarized in the following points:

- Proposing a framework to perform automatic classification of COVID-19 based on the CT lung images with the help of CNN, TL, and SpaSA Algorithm.
- The SpaSA is used to optimize the different CNN and TL hyperparameters aiming to find the best configurations for each used pretrained model and to enhance the classification performance.
- The proposed technique is adaptable; there is no need to assign the CNN architecture's hyperparameters values manually.
- Two different datasets are used. The first dataset (15,186 images) is partitioned into two classes, while the second one (22,779 images) is partitioned into three classes. The dataset in the current study faces four different scaling techniques. The SpaSA is used to find the best scaler technique.
- A comparison between the suggested approach and the other state-of-the-art approaches is conducted. The achieved results of the standard performance metrics are very promising.

The rest of the paper is organized as follows: In Section 2, the related studies of COVID-19 diagnosis based on CT and X-ray are reviewed. In Section 3, the background of AI, deep learning, and its counterparts are introduced. In Section 4, the methodology and the proposed framework overview are discussed. In Section 5, the numerical results are analyzed and discussed. Finally, in Section 6, the paper is concluded. .

Table 1

The configurations used for the different augmentation techniques to balance the datasets.

Technique	Value
Rotation	30°
Width Shift Ratio	20%
Height Shift Ratio	20%
Shear Ratio	20%
Zoom Ratio	20%
Brightness change	[0.8, 1.2]
Vertical Flip	Yes
Horizontal Flip	Yes

Table 2

The corresponding hyperparameter for each element in the solution.

Element #	Value
1	Loss function
2	Batch size
3	Dropout ratio
4	TL learning ratio
5	Weights optimizer
6	Scaler technique
7	Apply augmentation or not
8	Rotation value (if augmentation is true)
9	Width shift value (if augmentation is true)
10	Height shift value (if augmentation is true)
11	Shear value (if augmentation is true)
12	Zoom value (if augmentation is true)
13	Horizontal flip flag (if augmentation is true)
14	Vertical flip flag (if augmentation is true)
15	Brightness change range (if augmentation is true)

Table 3

The used datasets specifications summarization.

Dataset	No. of Classes	Classes	No. of Images (Before)	No. of Images (After)
COVID-19 Lung CT Scans and **COVID 19 CT Scan Dataset**	2	"COVID" and "NonCOVID"	14, 486	15, 186
Large COVID-19 CT scan slice dataset	3	"CAP", "COVID", and "NonCOVID"	17, 104	22, 779

Table 4

Common experiments Configurations.

Configuration	Specifications
Apply Dataset Shuffling?	Yes (Random)
Input Image Size	$(100 \times 100 \times 3)$
Hyperparameters Metaheuristic Optimizer	Sparrow Search Algorithm (SpaSA)
Train Split Ratio	85%–15% (i.e., 85% for training and validation; and 15% for testing)
SpaSA Size of Population	10
SpaSA Number of Iterations	10
Number of Epochs	5
Output Activation Function	SoftMax
Pre-trained Models	SeresNext50, SeresNext101, SeNet154, MobileNet, MobileNetV2, MobileNetV3Small, and MobileNetV3Large
Pre-trained Parameters Initializers	ImageNet
Losses Range	Categorical Crossentropy, Categorical Hinge, KLDivergence, Poisson, Squared Hinge, and Hinge
Parameters Optimizers Range	Adam, NAdam, AdaGrad, AdaDelta, AdaMax, RMSProp, SGD, Ftrl, SGD Nesterov, RMSProp Centered, and Adam AMSGrad
Dropout Range	$[0 \rightarrow 0.6]$
Batch Size Range	$4 \rightarrow 48$ (step = 4)
Pre-trained Model Learn Ratio Range	$1 \rightarrow 100$ (step = 1)
Scaling Techniques	Normalize, Standard, Min Max, and Max Abs
Apply Data Augmentation (DA)	[*Yes, No*]
DA Rotation Range	$0° \rightarrow 45°$ (step = 1°)
DA Width Shift Range	$[0 \rightarrow 0.25]$
DA Height Shift Range	$[0 \rightarrow 0.25]$
DA Shear Range	$[0 \rightarrow 0.25]$
DA Zoom Range	$[0 \rightarrow 0.25]$
DA Horizontal Flip Range	[*Yes, No*]
DA Vertical Flip Range	[*Yes, No*]
DA Brightness Range	$[0.5 \rightarrow 2.0]$
Scripting Language	Python
Python Major Packages	Tensorflow, Keras, NumPy, OpenCV, and Matplotlib
Working Environment	Google Colab with GPU (i.e., Intel(R) Xeon(R) CPU @ 2.00 GHz, Tesla T4 16 GB GPU, CUDA v.11.2, and 12 GB RAM)

2. Related studies

The main barriers to containing the spread of COVID-19 are untrusted screening systems and a scarcity of clinical facilities. As a result, the artificial neural network plays a significant role in computer vision, particularly medical imaging, for achieving human-level accuracy in visual data processing, classification, and segmentation. Convolutional Neural Network (CNN) has made a significant contribution to the medical system by being extremely useful in digital image processing. Innovative Pre-trained CNN models trained on large datasets are used to capitalize on the knowledge of generic features from the images. Since the COVID-19 has become widespread, extensive research has been conducted to address the application of various deep learning methods that aid in developing a new end-to-end diagnosis of the COVID-19 that does not require manual feature engineering [14]. Deep learning algorithms are essential to developing new diagnosis methods that can achieve promising performance in detecting acute Pneumonia.

Polsinelli et al. [15] developed a CNN-based light classifier based on CCT images of the lungs for the COVID-19 efficient and rapid diagnosis. The classifier determines if the CCT image is Pneumonia or healthy. The proposed architecture is characterized by short classification time and low computation resources. The proposed classifier is based on the SqueezeNet model characterized by the fewer parameters deployment and achieved acceptable accuracy density and inference time. In addition, they used the Bayesian method for hyperparameters optimization. The optimized hyperparameters are Initial Learning Rate, Momentum, and L2-Regularization. However, preprocessing stage can be deployed to increase the classifier performance. Maghdid et al. [16] introduced a modified simple CNN Diagnosing model based on transfer learning AlexNet architecture. They used the CXR and CCT scan images dataset from multiple sources to develop, train, and evaluate their diagnosing model. Their datasets are divided equally for training CNN and model validation. The proposed model achieves accuracy up to 98%. However, Performance degradation is found in chest radiograph-based diagnosis. A VGG-16 Network-based faster region CNN approach is proposed for the detection of COVID-19 based on CXR scans [17]. The proposed deep learning-based approach used 13,800 X-Ray images with a classification accuracy of 97.36%. However, the model can be enhanced to detect CT images with higher accuracy. A deep learning lung CT scans prediction model is implemented by Islam et al. [18]. They used LeNet-5 CNN architecture with a dataset that involves 746 CCT images. They used the image augmentation technique for the sake of enlarging the dataset. As a result, 80% of lung CT frames are used for training and 20% for testing. The model can be further enhanced to be more convenient and efficient enough.

Kundu et al. [19] introduced an end-to-end transfer learning CNN binary classification framework based on CT-scan images. First, they used three models to generate the initial decision scores fused by the proposed ensemble model. Then, ensembling is used to incorporate the discriminating properties of all the contributing models and assign fuzzy
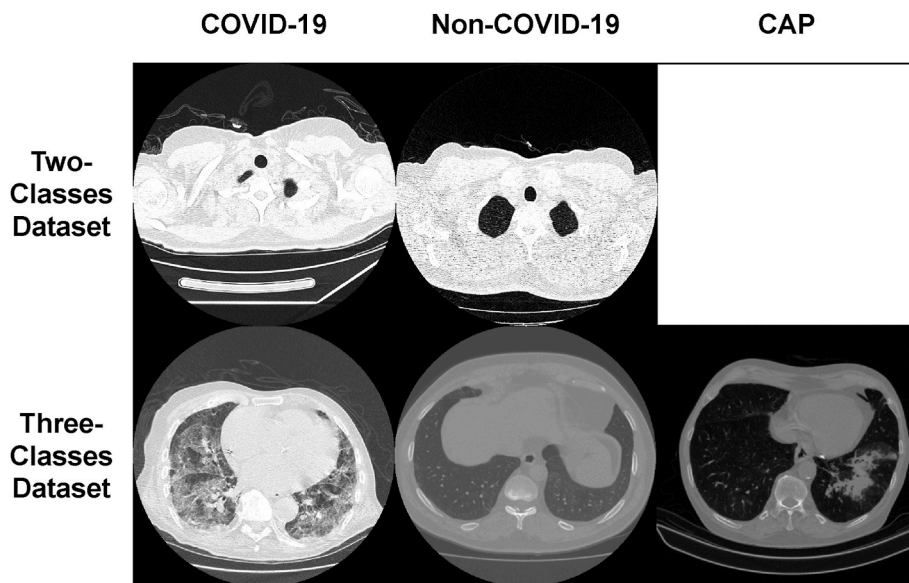
Fig. 6. Samples from the used datasets.

Table 5
Two-classes specific experiments Configurations.

Configuration	Specifications
Dataset Sources	COVID-19 Lung CT Scans [33] and Covid 19 CT Scan Dataset [34]
Number of Classes	2
Classes	('COVID' and 'NonCOVID')
Dataset Size before Data Balancing	"COVID": 7,593 and "NonCOVID": 6,893
Dataset Size after Data Balancing	"COVID": 7,593 and "NonCOVID": 7,593

Table 6
Confusion matrix results concerning the two-classes dataset.

Model Name	TP	TN	FP	FN
SeresNext50	15,022	15,022	158	158
SeresNext101	15,064	15,064	104	104
SeNet154	14,966	14,966	214	214
MobileNet	15,141	15,141	39	39
MobileNetV2	15,088	15,088	72	72
MobileNetV3Small	14,282	14,282	898	898
MobileNetV3Large	14,768	14,768	392	392

ranks to the classifiers. The proposed method achieves high classification accuracies of 98.80% based on experimental results. However, the proposed framework has drawbacks such as computation cost, overfitting issues, and recognition capability of the CNN models. Pathana et al. [20] introduced two classification architectures based on a transfer learning approach. The first architecture uses five standards, namely ResNet-50, AlexNet, VGG19, Densenet, and Inception V3. The second architecture deploys CNN hyperparameters optimization strategy using the WOA-BAT optimization algorithm. The optimized CNN extracts features and classifies CCT images into COVID-19 and normal. They used 746 CCT images combined from three datasets from different hospitals.

Tripti Goel et al. [12] developed a deep learning-based framework for the automatic diagnosis of COVID-19 patients. The proposed framework introduced an effective feature extraction and high performance in three stages. The first stage concerns the augmentation of the data through a generative adversarial network (GAN) architecture to generate more CCT images for DL networks training. The WOA

Optimization is used to optimize the GAN hyperparameters in the second stage. The main objective was to avoid issues with overfitting and instability. Finally, the classification stage used a pretrained InceptionV3 DL model to classify COVID-19 patients automatically. They used the SARS-CoV-2 CT-Scan dataset that contains 2,482 CCT scan images. The experimental study proved that the proposed model outperformed other state-of-the-art models with achieved accuracy of 99.22%.

Huang et al. [21] proposed a collaborative multi-center sparse learning (MCSL) and decision fusion approach that considered data inconsistency for COVID-19 classification based on CCT images. First, the CCT images are converted into HOG images to reduce structural differences. Then, feature extraction is performed via a proposed 3D-CNN model to extract deep features. The MCSL method selects discriminative features for training multi-center classifiers and then fuses the classifiers' decisions. To validate the effectiveness of the proposed method experimental study was performed based on five CCT datasets of 1,034 images. They achieved appealing accuracy(98.03%), sensitivity (95.89%), and specificity (99%). The authors intended to further enhance the MCSL approach by supporting multi-modal data, deploying a semisupervised method to adapt many cases, and adding the segmentation stage to improve diagnostic performance further. Abraham and Nair [22] proposed COVID-19 diagnosis method consisting of CNNs and Kernel SVM classifier. They aimed to classify patients into COVID-19 and non-COVID-19 using CT images. The proposed method combined features extracted from TL and five pre-trained CNNs. They used a dataset consisting of 746 CT images. The experimental analysis proved that the extracted features using the CNNs and KSVM classifier achieved an accuracy of 91.6%. R. Murugan and Tripti Goel [23] proposed an accurate modified pre-trained CNN-ResNet50 based on the Extreme Learning Machine (E-DiCoNet) model for diagnosing COVID-19 (COVID-19, bacterial Pneumonia, and normal). The E-DiCoNet model consists of an input layer, several hidden layers, a pooling layer, and a classifier. This model has utilized 2,700 chest CXR images from multiple data sources. The proposed model achieved accurate diagnosis with less training time and exceptionally exactness. In addition, the proposed framework achieved good performance metrics as follows: accuracy (94.07%), sensitivity (98.15%), specificity (91.48%), recall (85.21%), precision (98.15%), and F1-score (91.22%). Based on CXR and CCT images, Goura and Jain [24] introduced a novel deep learning-based stacked CNN (DLS-CNN) model for COVID-19 diagnosis. First, different sub-models are deployed from the VGG19 and the Xception models during the training and then ensembled together using

Table 7
The best solutions after the learning and optimization process concerning the two-classes dataset.

Model Name	Loss	Batch Size	Dropout	TL Learn Ratio	Optimizer	Scaler	Apply Augmentation	Rotation Range	Width Shift Range	Height Shift Range	Shear Range	Zoom Range	Horizontal Flip	Vertical Flip	Brightness Range
SeresNext50	Categorical Crossentropy	12	0.2	89	SGD	Standardize	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SeresNext101	KLDivergence	24	0.07	29	SGD Nesterov	MinMax	Yes	16	0.25	0.23	0.05	0.05	No	No	1.2–1.87
SeNet154	Poisson	44	0.22	63	SGD	MinMax	Yes	29	0.13	0.1	0.18	0	No	No	1.08–1.55
MobileNet	KLDivergence	44	0.37	60	SGD	MaxAbs	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
MobileNetV2	KLDivergence	40	0	62	SGD Nesterov	MinMax	Yes	36	0.04	0.09	0.11	0.09	Yes	Yes	1.32–1.79
MobileNetV3Small	Squared Hinge	12	0.23	100	SGD	Standardize	Yes	41	0.09	0.15	0.05	0.09	No	No	0.57–1.56
MobileNetV3Large	KLDivergence	40	0.1	45	SGD	Standardize	Yes	37	0.18	0.05	0.06	0.04	Yes	Yes	0.65–0.7

a softmax classifier. An available public data set of 3,040 CXR images were used for multiclass classification. They used 4,645 CCT images for binary classification. As a result, the DLS-CNN model achieved an accuracy of 97.27% and 98.30% for multiclass and binary classification, respectively.

Murugan et al. [25] proposed an optimized DL network (WOANet) for feature extraction and binary classification of COVID-19. They used the ResNet-50 CNN network to diagnose the COVID-19 through CCT images. They used backpropagation and WOA algorithms for hyperparameter optimization to ensure maximum performance. The proposed method doesn't need preprocessing and ROI extraction. They used the COVID-CT dataset contains 2,700 CCT images. The Proposed WOANet achieved Accuracy, Sensitivity, Specificity, Precision, and F1 score of 98.78%, 98.37%, 99.19%, 99.18%, and 98.37% respectively.

Gayathri et al. [26] proposed a CNN-based CAD system for the COVID-19 binary classification of using CXR images. The proposed model used (i) feature extraction from several combinations of pre-trained networks, (ii) dimensionality reduction for extracted features with Sparse autoencoder, and (iii) classification using a Feed-Forward Neural Network (FFNN). Two CXR image datasets consisting of 1046 scans were used. The InceptionResnetV2 and Xception models achieved an accuracy of 0.9578 and an AUC of 0.9821. Tripti Goel et al. [27] presented a new model made up of pre-trained networks InceptionV3 and ResNet50. In addition, they proposed an optimized, fully automated dual-stage DL (Multi-COVID-Net) architecture based on CXR to classify COVID-19 patients into normal, COVID-19, and Pneumonia. The first stage concerns automatic feature extraction, and the second stage is multiclass classification. They introduced a Multi-Objective Grasshopper Optimization Algorithm (MOGOA) for hyperparameters optimization. A dataset consisting of 2,700 CXR images was used. An extensive experimental analysis proved the efficiency of the proposed model (accuracy of 98.27%). However, decreasing computational complexity due to using two DL networks is required.

Guoqing et al. [28] introduced a multitask learning (MTL) framework for COVID-19 automated diagnosis. Unsupervised lung segmentation, Shift3D, and a novel random-weighted loss function are used. The MTL framework achieved vulnerable COVID-19 tasks prioritization, convergence acceleration, and joint learning performance improvement. The MTL framework detected COVID-19 Pneumonia using 3D CNN and auxiliary FNN against CCT scans and RT-PCR. A dataset of 1,329 CCT images was used as an input. The MTL achieved accuracies of 90.23% and 79.20% for detecting COVID-19 based on CT and RT-PCR. Shaik and Cherukuri [29] introduced a novel ensemble DNN strategy that used various TL-based pre-trained models for COVID-19 diagnosis based on CCT images. The strategy steps are as follows: preprocessing the CT images, feature extraction using the deep pre-trained models, fine-tuning the obtained features on a three-layered DNN, and classification via ensemble classifier. Two benchmark datasets containing 3,228 CT images were used. The proposed strategy achieved an accuracy of 93.33% and minimized misclassifications.

The innovation of DL techniques enables accurate image classification without manual feature engineering. Deep learning models outperform on larger datasets. Therefore, a larger dataset is essential to strengthen the classification model and expand the investigation. Different data sources, including X-ray and CT images from various countries, should be used to generate a sophisticated tool to assist radiologists in diagnosing COVID-19. CNN's hyper-parameter optimization has a significant impact on performance. On the other hand, the selection of hyperparameters is application-dependent and may result in low-performance metrics. As a result, application-specific values derived from an optimization methodology should be used rather than selecting hyperparameter values at random.

To summarize, the COVID-19 classification has been the subject of considerable literature. However, most of these studies suffer from limited data sets, low accuracy, and high computational complexity. With all these challenges at hand, there is still debate about how best to

classify COVID-19. Many questions have been raised regarding the transfer of knowledge from one application to another, how to reduce the learning time for the model, and how to avoid affecting the model results due to hyperparameter settings.

3. Background

The artificial intelligence (AI) industry has focused on intense media coverage over the past few years. In a nutshell, AI is the field that involves automating intellectual tasks normally performed by humans. Thus, AI is a general field that includes machine learning, deep learning, and a variety of other approaches that do not require learning. Machine Learning is the science and art of programming computers to learn from data. A popular area of research in artificial intelligence is deep learning. Models can be run end-to-end using input data without extracting features manually [17]. Machine learning has recently become a popular diagnostic tool for doctors, providing them with a complementary tool.

As Deep Learning learns the best features and contributes to the overall result, it performs feature engineering. Deep Learning partially replaces feature engineering or creating better predictive features by hand. Deep Learning is also known as representation learning. Deep networks can be thought of as multistage distillation operations where information goes through successive filters and becomes increasingly purified. By developing deep learning techniques, advanced image classification is possible without manual feature engineering [14].

The Convolution neural networks (CNN) model ranks among the most prominent and important deep learning models, demonstrating advantages in computer vision, speech recognition, and medical diagnosis. CNN Deep Learning Models extract relevant features through a sequence of convolutional layers followed by fully connected neural networks. Convolution neural networks (CNN) and recurrent neural networks (RNN) are among the various types of deep learning algorithms [16]. As with image processing applications, CNNs could be applied when retrieved data from these solutions reside in a spatial domain. The RNN, on the other hand, works on the concept of reusing the output of each layer as the input for the next layer. Moreover, RNNs are compatible with applications that get sequential data, such as those that get text or readings from signals.

Deep CNN architecture named AlexNet demonstrated excellent performance on high-challenge datasets in the ImageNet LSVRC-2012 competition [14]. In this study, Alex et al. developed a wide range of network settings and training skills, including dropouts, pooling, and local response normalization, which enabled deeper CNN training more effectively and improved performance. In recent years, several networks have been created based on AlexNet, such as VGG, GoogleNet, ResNet, DenseNet, MobileNet, SqueezeNet, etc.

Learning from abstract representations enables CNNs to analyze the images with a high level of semantics. For example, a CNN uses filter banks to exploit the texture in the images, rather than handcrafted filter banks. It is widely acknowledged that the availability of huge amounts of data is one of the bottlenecks in the literature [20].

In medical imaging, deep learning methods are extensively used. Especially, convolutional neural networks (CNNs) have been used to solve classification and segmentation problems in CT images, among other problems. There are only a few COVID-19 datasets available, and of those that are available, they contain a limited number of CT images. Thus, during the training phase, there is a need to avoid/reduce overfitting that is, if the CNN does not learn the discriminant features of the COVID-19 CT scans but rather memorizes them) [15]. CNN inference is also a computationally-intensive process.

Despite the success of reported applications, current studies on COVID-19 classification also disclose some limitations [14]. Since there are limited available training data, there is much literature about data imbalances between classes. The unbalanced data makes deep-learning models unlikely to be trained well, and the high accuracy in such circumstances cannot guarantee COVID-19 detection effectiveness.

Deep learning has received great praise in artificial intelligence, but it requires considerable time and data. However, another method has been developed that can overcome the limitations of deep learning: transfer learning [16]. Training a large DNN from scratch is generally not a good idea: instead, you should look for an existing neural network that accomplishes a similar task to the one you are attempting. Transfer learning is the process of reusing a pretrained network and transferring the learned model into a new model. Additional training data and modified neural layers can also be incorporated into the new model.

In order to achieve good results with limited available training data when using a CNN, it is crucial to optimize the training phase. The training phase of a CNN is strongly influenced by the hyperparameter settings [20]. The hyperparameters differ from the model weights. The former is calculated before the training phase, while the latter is optimized during training. There are several ways to set hyperparameters, and different strategies can be adopted. Using the manual selection method would be the first option, though avoiding it is preferable due to many different configurations.

Similarly, grid search (GS) is a conventional and popular approch for the hyperparameters optimization of DL networks. The combination which gives the best results from the grid will be selected hyperparameters. However, the main drawback is the increase in the number of iterations exponentially with the insertion of each hyperparameter [27]. Another drawback is that GS do not use past evaluations and hence much time has to be spent evaluating bad hyperparameter configurations. Applying the above-reported methods in clinical scenarios lacks reliability because deep learning models perform better with larger datasets. Further, these models are developed using standard parameters. It is the hyperparameters chosen and the dataset that influence the classification performance of a CNN. Hyperparameter selection is an application-dependent process that may result in low-performance metrics. In place of choosing hyperparameter values randomly, application-specific values are selected through a method of optimization.

An exact optimization algorithm cannot provide an optimal solution to a high-dimensional search space problem. Because of the exponential growth of the search space with the size of the problem, a comprehensive search is not possible. Near-optimal solutions can be found for difficult optimization problems Using population-based optimization algorithms. Therefore, the population is shifted towards better solutions in the search space [30].

The number of layers, the size, shape, type, number of neurons, intermediate processing elements, and other structural characteristics can cover a large solution space, requiring search heuristics for efficient exploration. Neural Architecture Search (NAS) has been coined to describe all the techniques that aim to automate the design of neural networks. One of the most studied branches of Artificial Intelligence is bio-inspired computation. Nature-inspired metaheuristic algorithms have gained huge popularity in recent years because they have demonstrated promising results in solving tough optimization problems [31].

Bio-inspired algorithms do not impose any requirement on the objective function to be optimized, nor do they require it to be differentiable. The advantage of using metaheuristics over calculus-based methods, or simple heuristics, is their capability to search over large sets of feasible solutions with less computational effort than calculus-based methods or simple heuristics. SI has recently become the most rapidly growing of the bio-inspired computing fields. Swarm Intelligence is a branch of bio-inspired computation based on the development of collective intelligence from large populations of agents with simple communication and interaction patterns. Swarm-based algorithms take their cues from social organisms like ants, termites, birds, and fishes. Swarm-based systems can self-organize and have decentralized control, as in nature, which allows them to produce emergent behavior. However, no system components can act alone to achieve emergent behavior, which arises through local interactions between components [30].

Table 8
The two-classes dataset experiments with the maxmimized metrics.

Model Name	Accuracy	F1	Precision	Recall	Specificity	Sensitivity	AUC	IoU	Dice	Cosine Similarity
SeresNext50	98.96%	98.96%	98.96%	98.96%	98.96%	98.96%	99.89%	98.40%	98.72%	99.09%
SeresNext101	97.41%	97.41%	97.41%	97.41%	97.41%	97.41%	99.68%	96.61%	97.25%	97.84%
SeNet154	99.31%	99.31%	99.31%	99.31%	99.31%	99.31%	99.87%	99.18%	99.32%	99.40%
MobileNet	98.59%	98.59%	98.59%	98.59%	98.59%	98.59%	99.83%	97.66%	98.13%	98.68%
MobileNetV2	94.08%	94.08%	94.08%	94.08%	94.08%	94.08%	97.81%	95.01%	95.52%	94.78%
MobileNetV3Small	99.53%	99.53%	99.53%	99.53%	99.53%	99.53%	99.96%	98.89%	99.15%	99.54%
MobileNetV3Large	99.74%	99.74%	99.74%	99.74%	99.74%	99.74%	99.97%	99.69%	99.74%	99.78%

Table 9
The two-classes dataset experiments with the minimized metrics.

Model Name	Categorical Crossentropy	KLDivergence	Categorical Hinge	Hinge	Squared Hinge	Poisson	Logcosh Error	Mean Absolute Error	Mean Squared Error	Mean Squared Logarithmic Error	Root Mean Squared Error
SeresNext50	0.033	0.033	0.038	0.519	0.528	0.517	0.004	0.019	0.009	0.004	0.092
SeresNext101	0.069	0.069	0.083	0.541	0.561	0.534	0.009	0.041	0.020	0.010	0.140
SeNet154	0.024	0.024	0.021	0.510	0.516	0.512	0.003	0.010	0.006	0.003	0.075
MobileNet	0.047	0.047	0.056	0.528	0.540	0.523	0.006	0.028	0.012	0.006	0.111
MobileNetV2	0.229	0.229	0.134	0.567	0.616	0.614	0.022	0.067	0.049	0.024	0.221
MobileNetV3Small	0.019	0.019	0.026	0.513	0.517	0.510	0.002	0.013	0.004	0.002	0.067
MobileNetV3Large	0.008	0.008	0.008	0.504	0.506	0.504	0.001	0.004	0.002	0.001	0.045

Table 10
Three-classes specific experiments Configurations.

Configuration	Specifications
Dataset Source	Large COVID-19 CT scan slice dataset [35]
Number of Classes	3
Classes	('COVID', 'NonCOVID', and 'CAP')
Dataset Size before Data Balancing	"COVID": 7,593, "NonCOVID": 6,893, and "CAP": 2,618
Dataset Size after Data Balancing	"COVID": 7,593, "NonCOVID": 7,593, and "CAP": 7,593

Table 11
Confusion matrix results concerning the three-classes dataset.

Model Name	TP	TN	FP	FN
SeresNext50	22,200	44,956	540	548
SeresNext101	21,585	44,554	966	1,175
SeNet154	21,312	44,136	1,384	1,448
MobileNet	22,299	45,074	446	461
MobileNetV2	21,574	44,364	1,172	1,194
MobileNetV3Small	16,961	40,707	4,845	5,815
MobileNetV3Large	21,318	44,088	1,416	1,434

Based on the characteristics of inspiration, several optimization algorithms have been proposed. Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Cuckoo Search (CSA), Elephant Herding Optimization (EHO), Whale Optimization Algorithm (WOA) are examples of SI algorithms. Based on the Barnacles Mating Optimizer (BMO), an optimization algorithm mimics barnacles' mating behavior in nature. In Search and Rescue (SAR), an optimization algorithm mimics the exploration behavior of humans. The Lévy flight distribution algorithm (LFD) uses a distribution similar to that used in Lévy flight random walks to explore large search spaces. Slime mould algorithm (SMA), Student psychology-based optimization (SPBO), Wingsuit Flying Search (WFS), Political Optimizer (PO), Aquila Optimizer (AO), The Equilibrium Optimizer (EO), Learner performance-based behavior algorithm (LPB), The Sine Cosine Algorithm (SCA), The Honey Badger Algorithm (HBA) To name a few [30]. There is an excellent classification of natural optimization algorithms, and 132 are listed [31]. A review of evolutionary algorithms and their applications to engineering problems is provided [32].

In contrast to other swarm meta-heuristic algorithms, the Sparrow optimization algorithm (SpaSA) [20] is used in this case. Furthermore, the Sparrow reaches global optimum solutions without any structural reformation and avoids local optimum strategies. The SpaSA is used to select the best-pretrained model among all recommended models, optimizing the CNN and TL hyperparameters to figure out the best configuration for each used pre-trained model. The sparrows are generally gregarious birds with various species [13]. Almost everywhere in the world, they live around where humans live. They primarily feed on seeds of grains and weeds. The Sparrow is a well-known resident bird. The captive house sparrow has two types, both producers and scroungers. Producers actively seek out food sources, while scroungers obtain food from the producers. It has also been shown that the birds generally switch between producing and scrounging. Sparrows usually use the strategies of both the producers and the scroungers to find food.

In studies, it has been found that individuals in groups monitor the behavior of their colleagues. Moreover, the predators in the flock, which want to increase their predation rate, use high intakes of food to compete for food resources with the companions. Additionally, sparrows may modulate their foraging strategies based on their energy reserves, with sparrows with low energy reserves scrounging more. The birds at the perimeter of the population are also to be noted, as they are more likely to be attacked by predators and continuously attempt to get better positions. In order to minimize their danger domain, the animals located in the center may move closer to their neighbors. The sparrows have also been shown to be very curious and always vigilant. One or more birds chirp when the group spots a predator approaches, for instance, and the entire flock flies away [13]. In light of the previous description of sparrows, here are the rules that describe the behavior of sparrows as idealized for simplicity:

- Scroungers are offered directions or areas of foraging by producers who usually have high energy reserves. Producers are responsible for identifying areas that contain rich food sources. An assessment of the fitness values of each individual determines their energy reserves.
- The sparrows begin chirping once they detect the predator. If the alarm value exceeds the safety threshold, the producers should lead all scroungers to the safe area.
- Each Sparrow can become a producer as long as it seeks out the best food sources, but the proportion of producers and scroungers remains constant within the population.
- Producers would be those sparrows that have the highest energy. Starving scroungers will fly to other places, searching for food to gain more energy.
- Scroungers hunt for food by following producers who can provide the best food. In the meantime, some scroungers track the producers constantly, keeping tabs on the food supply and competing with them.
- As soon as they see danger approaching, the sparrows on the edge of the group move toward a safe area to get a better position. In contrast, the center's sparrows wander randomly to be near other sparrows.

4. Methodology

As mentioned, the current study proposes an empirical quantitative framework to perform automatic and classification of COVID-19 based on the CT lung images with the help of CNN, TL, and the SpaSA Algorithm for parameters and hyperparameters optimization. The suggested framework is shown in Fig. 5.

4.1. Dataset acquisition phase

The dataset are acquired from three public datasets from Kaggle. The details are discussed in Section 5.1.

4.2. Dataset pre-processing phase

In the pre-processing phase, each image is entered to a pipeline that consists of two operations: resizing and scaling. After that, each individual dataset is up-balanced to equalize the number of images per category.

4.2.1. Dataset resizing

The dataset are resized to the size of (100, 100, 3) in the RGB color mode. The reason behind this is the limited capacity of the memory and GPU and to avoid overflow crashes.

4.2.2. Dataset scaling

The dataset in the current study faces four different sclaing techniques: normalization, standardization, min-max, and max-abs. One of the target of using the SpaSA is to find the best scaler technique. The equations behind them are shown in Equation (1), Equation (2), Equation (3), and Equation (4).

$$X_{scaled} = \frac{X}{\max(X)} \tag{1}$$

$$X_{scaled} = \frac{X - \mu}{\sigma} \tag{2}$$

$$X_{scaled} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{3}$$

$$X_{scaled} = \frac{X}{|\max(X)|} \tag{4}$$

where X is the input image, μ is the mean, σ is the standard deviation.

Table 12
The best solutions after the learning and optimization process concerning the three-classes dataset.

Model Name	Loss	Batch Size	Dropout	TL Learn Ratio	Optimizer	Scaler	Apply Augmentation	Rotation Range	Width Shift Range	Height Shift Range	Shear Range	Zoom Range	Horizontal Flip	Vertical Flip	Brightness Range
SeresNext50	Poisson	44	0.2	26	AdaMax	MinMax	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SeresNext101	Poisson	20	0.45	52	SGD Nesterov	MinMax	Yes	23	0.15	0.02	0	0.01	Yes	Yes	0.57–1.25
SeNet154	Squared Hinge	40	0	27	AdaGrad	MinMax	Yes	11	0.03	0.22	0.07	0.25	Yes	No	1.4–1.52
MobileNet	Categorical Crossentropy	20	0.08	75	AdaMax	MaxAbs	Yes	11	0.06	0.05	0.13	0.14	Yes	No	1.45–1.59
MobileNetV2	Squared Hinge	16	0.53	63	SGD Nesterov	MinMax	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
MobileNetV3Small	Squared Hinge	12	0.2	91	AdaGrad	Normalize	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
MobileNetV3Large	Categorical Crossentropy	36	0.3	31	AdaMax	Standardize	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

4.2.3. Dataset balancing

Each used dataset in the current study is imbalanced. To overcome this issue, the data augmentation approach is used. The current study uses the rotation, shifting in the width and height, shearing, zooming, flipping in the horizontal and vertical axes, and brightness changing augmentation techniques. Table 1 shows the configurations used for the different augmentation techniques to balance the datasets.

4.3. Training and learning phase using TL and SpaSA

The current study uses SpaSA to optimize the different CNN and TL hyperparameters aiming to find the best configurations for each used pre-trained model. The working process inherits the working mechanism of the metaheuristic population-based optimizers. It is combined of the population generation, fitness score (i.e., function) evaluation, population sorting, and population updating. The last three steps are iteratively repeated for a number of iterations *M*.

4.3.1. Initial population generation

Initially, the population is numerically generated randomly where the number of solutions is *n*. Each solution is in the range of [0, 1] and the size of each is *D*. Each element is the solution reflects a specific hyperparameter. Table 2 shows the corresponding hyperparameter for each element in the solution.

From Table 2, we can deduce that $D = 15$ if data augmentation during training is applied and $D = 7$ if not.

4.3.2. Fitness score evaluator

In the current step, the fitness score of each solution is calculated iteratively. It consists of inner steps:

Hyperparameters Converter: This step converts the numerically generated random values to the corresponding value of the specified hyperparameter. How does this happen? For the first element, as an example, it should reflect the loss function as mentioned in Table 2. So, we need to map from the range [0, 1] to the corresponding loss function. The used loss functions in the current study are Categorical Crossentropy, Categorical Hinge, KLDivergence, Poisson, Squared Hinge, and Hinge (Table 4). If the value of the element is 0, it should refer to the Categorical Crossentropy loss function, if it is 1, it should refer to the Hinge loss function, and so on.

TL Pre-trained Model Creator and Injector: After converting each element in the solution to the corresponding hyperparameter, the target pre-trained model will be initialized and the hyperparameters will be injected in it. The used pre-trained CNN models in the current study are SeresNext50, SeresNext101, SeNet154, MobileNet, MobileNetV2, MobileNetV3Small, and MobileNetV3Large with the ImageNet pre-trained weights.

TL Pre-trained Model Training: The pre-trained TL after that will start the training and learning process using the specified hyperparameters. In this process, the dataset is split into training, testing, and validation subsets.

TL Pre-trained Model Evaluation: After the training and learning process, the model is evaluated on the whole entered dataset. Different performance metrics are evaluated such as accuracy, precision, and recall.

The different used performance metrics in the current study are accuracy (Equation (5)), precision (Equation (6)), specificity (Equation (7)), recall (i.e., sensitivity) (Equation (8)), F1-score (Equation (10)), AUC, IoU, Dice coef. (Equation (9)), and cosine similarity.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

$$Recall = Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{9}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

4.3.3. Population sorting

In this step, the population is sorted in descending order concerning the fitness score so that the best solution is placed at the top while the worst solution is placed at the bottom.

4.3.4. Population updating using SpaSA

The population is updated using SpaSA equations in this step. Equation (11) represents the discoverer location update formula. The followers' positional update is presented in Equation (12). The anti-predation behavior is described in Equation (13).

$$X^{t+1} = \begin{cases} X^t \times exp\left(\frac{-h}{\alpha \times M_h}\right), & \text{if } (R_2 < ST) \\ X^t + Q \times L, & \text{Otherwise} \end{cases} \tag{11}$$

$$X^{t+1} = \begin{cases} Q \times exp\left(\frac{X^t_{worst} - X^t}{i^2}\right), & \text{if } \left(i > \frac{n}{2}\right) \\ X_P^{t+1} + |X^t - X_P^{t+1}| \times A^+ \times L, & \text{Otherwise} \end{cases} \tag{12}$$

$$X^{t+1} = \begin{cases} X^t_{best} + \beta \times |X^t - X^t_{best}|, & \text{if } (f_i \neq f_g) \\ X^t + K \times \left(\frac{|X^t - X^t_{worst}|}{(f_i - f_w) + \epsilon}\right), & \text{Otherwise} \end{cases} \tag{13}$$

From Equation (11), X^t is the current solution at iteration *t*, *h* is the current iterations number, *M* is the maximal iterations number, α is a random number $\in$ [0, 1], *Q* is a random number from the normal distribution. *L* represents a $1 \times D$ matrix containing all 1 element, R_2 and *ST* are the warning and safety values respectively, $R2 \in$ [0, 1], and $ST \in$ [0.5, 1].

From Equation (12), X_P^{t+1} is the currently optimal discoverer position at iteration *t*, X^t_{worst} indicates the current worst position at iteration *t*, *A* is a $1 \times D$ matrix, and $A^+ = A^T \times (A \times A^T)^{-1}$.

From Equation (13), X_{best} is the global optimum solution. β is the control step-size parameter and is a random number obeying a normal distribution, *K* is a random number $\in$ [− 1, 1] and it represents the direction of movement and the sparrow and also controls the moving step size, f_i denotes the current sparrow individual fitness value, f_g and f_w are the optimal and worst fitness values respectively, and ϵ is the smallest real number that is used to avoid the division by zeros.

Algorithm 1 summarizes the population updating sub-phase using SpaSA. *n* is the number of sparrows (i.e., population size).

Algorithm 1. The population updating sub-phase pesudocode

```
 1  Function UpdatePopulation(population, scoresList)
 2  best, worst, optimal, bestScore, worstScore = ExtractFromPopulation(population, scores)     // Extract the
       best, worst, and optimial solutions; and best and worst scores.
    // Start the updating process using SpaSA equations.
 3  i = 1                                                                   // Initialize a counter.
 4  while (i ≤ PD) do
 5      if (R2 < ST) then
 6          population[i] = population[i] × exp^(−h/(α×M))
 7      else
 8          population[i] = population[i] + Q × L
 9      i = i + 1                                                           // Increment the counter.
10  i = 1                                                                   // Initialize a counter.
11  while (i ≤ (n − PD)) do
12      if (i > 0.5 × n) then
13          population[i] = Q × exp^((worst−population[i])/i²)
14      else
15          population[i] = optimal + |population[i] − optimal| × A⁺ × L
16      i = i + 1                                                           // Increment the counter.
17  i = 1                                                                   // Initialize a counter.
18  while (i ≤ SD) do
19      if (scores[i] ≠ bestScore) then
20          population[i] = best + β × |population[i] − best|
21      else
22          population[i] = population[i] + K × (|population[i]−worst| / (scores[i]−worstScore/+ε))
23      i = i + 1                                                           // Increment the counter.
24  return population                                                      // Return the updated population
25  End Function
```

4.4. Evaluation and prediction phase

After the learning and optimization iterations are completed, the best combination can be used in the production systems.

4.5. Exporting phase

The models are exported to be used in further phases, the results are exported in suitable files such as Excel and CSV files, and the graphs are

Table 13
The three-classes dataset experiments with the maxmimized metrics.

Model Name	Accuracy	F1	Precision	Recall	Specificity	Sensitivity	AUC	IoU	Dice	Cosine Similarity
SeresNext50	95.25%	95.21%	95.65%	94.84%	97.88%	94.84%	99.54%	94.55%	95.44%	96.12%
SeresNext101	97.61%	97.61%	97.63%	97.59%	98.81%	97.59%	99.83%	97.31%	97.75%	98.02%
SeNet154	98.00%	98.00%	98.04%	97.97%	99.02%	97.97%	99.92%	96.94%	97.57%	98.36%
MobileNet	94.80%	94.80%	94.85%	94.76%	97.43%	94.76%	98.07%	95.43%	95.88%	95.19%
MobileNetV2	76.15%	75.92%	77.90%	74.47%	89.36%	74.47%	88.59%	76.67%	79.68%	79.66%
MobileNetV3Small	93.70%	93.76%	93.89%	93.64%	96.96%	93.64%	97.43%	94.31%	94.92%	94.29%
MobileNetV3Large	93.73%	93.73%	93.77%	93.70%	96.89%	93.70%	98.20%	94.71%	95.25%	94.47%

Table 14
The three-classes dataset experiments with the minimized metrics.

Model Name	Categorical Crossentropy	KLDivergence	Categorical Hinge	Hinge	Squared Hinge	Poisson	Logcosh Error	Mean Absolute Error	Mean Squared Error	Mean Squared Logarithmic Error	Root Mean Squared Error
SeresNext50	0.123	0.123	0.129	0.712	0.735	0.374	0.011	0.046	0.023	0.011	0.151
SeresNext101	0.065	0.065	0.067	0.689	0.701	0.355	0.006	0.022	0.012	0.006	0.110
SeNet154	0.054	0.054	0.072	0.691	0.701	0.351	0.005	0.024	0.010	0.005	0.100
MobileNet	0.279	0.278	0.122	0.708	0.738	0.426	0.014	0.041	0.031	0.015	0.175
MobileNetV2	0.793	0.793	0.567	0.870	0.989	0.598	0.054	0.203	0.119	0.059	0.345
MobileNetV3Small	0.398	0.386	0.148	0.717	0.753	0.461	0.016	0.051	0.036	0.017	0.189
MobileNetV3Large	0.294	0.285	0.141	0.714	0.749	0.428	0.015	0.048	0.035	0.017	0.186

displayed and stored.

4.6. The suggested framework pseudocode

The steps are iteratively computed for a number of iterations. Algorithm 2 summarizes the proposed learning and optimization approach.

Algorithm 2. The suggested framework pesudocode

Input: *model, dataset*   `// Model name, Dataset`
Output: *best, bestScore*   `// The best overall score and solution`
1   $trainX, validationX, testX, trainY, validationY, testY = \textbf{SplitDataset}(dataset)$ `// Partition the dataset into`
    `training, testing, and validation portions concerning the SR.`
2   $model = \textbf{CreateTLModel}(/$ `// Create the initial pre-trained CNN model.`
3   $population = \textbf{GenerateInitialPopulation}(/$ `// Generate (i.e., create) the initial population.`
    `// Execute the learning SpaSA hyperparameters optimization process for` M `iterations.`
4   $m = 1$ `// Initialize the iterations' counter where` $m \leq M$.
5   **while** $(m \leq M)$ **do**
       `// Calculate the scores for the population.`
6      $i = 1$ `// Initialize the sparrow' counter where` $i \leq n$.
7      $scoresList = []$ `// Initialize the scores list.`
8   **while** $(i \leq n)$ **do**
9         $score = \textbf{CalculateFitnessScore}(model, population[i], trainX, trainY, validationX, validationY)$
           `// Calculate the fitness score (i.e., accuracy) for the current solution.`
10   Append$(score, scoresList)$ `// Calculate the score into the scores list.`
11       $i = i + 1$ `// Increment the sparrow' counter.`
       `// Sort the population scores.`
12     $\textbf{SortPopulation}(population, scoresList)$ `// Sort the scores list in descending order.`
13     $population = \textbf{UpdatePopulation}(population, scoresList)$ `// Update the population using SpaSA`
      `(Algorithm 1).`
14 return *best, bestScore*   `// Return the best score and solution`

---

5. Experiments and discussions

5.1. Datasets

The experiments are performed on two different datasets. The first dataset is partitioned into two classes while the second one is partitioned into three classes. For the first dataset, the authors combined two public COVID-19 datasets namely **COVID-19 Lung CT Scans** and **COVID 19 CT Scan Dataset**. The number of overall images is 14, 486. For the second dataset, the authors used **Large COVID-19 CT scan slice dataset** which contained 17, 104 images.

For both datasets, data augmentation is used before the learning process to equalize (i.e., balance) the number of images per class. After equalization, the first dataset contained 15, 186 images where each class contained 7, 593 images. Also, the second dataset contained 22, 779 images after equalization where each class contained 7, 593 images. Table 3 summarizes the specifications of the used datasets.

Samples from the used datasets are displayed in Fig. 6.

5.2. Experiments configurations

Table 4 summarizes the common configurations of all experiments.

5.3. Two-classes dataset experiments

Table 5 summarizes the configurations related to the two-classes

dataset.

Table 6 shows the TP, TN, FP, and FN of the best solutions after the learning and optimization processes on each pre-trained model concerning the two-classes dataset. It shows that MobileNet pre-trained model has the lowest FP and FN values. On the other hand, MobileNetV3Small has the highest FP and FN values.

The best solutions combinations concerning each model are reported in Table 7. It shows that the KLDivergence loss is recommended by four models. The SGD parameters optimizer and applying data augmentation are recommended by seven models. The standardization and min-max scaler are recommended by three models each. Data augmentation is recommended by five models where horizontal and vertical flipping are recommended to be 60% turned off.

From the values reported in Table 6 and the learning history, we can report different performance metrics. The reported metrics are partitioned into two types. The first reflects the metrics that are required to be maximized (i.e., Accuracy, F1, Precision, Recall, Specificity, Sensitivity, AUC, IoU, Dice, and Cosine Similarity). The second reflects the metrics that are required to be minimized (i.e., Categorical Crossentropy, KLDivergence, Categorical Hinge, Hinge, SquaredHinge, Poisson, Logcosh Error, Mean Absolute Error, Mean IoU, Mean Squared Error, Mean Squared Logarithmic Error, and Root Mean Squared Error). The first category metrics are reported in Table 8 while the second is in Table 9.

From them, we can report that the MobileNetV3Large pre-trained model is the best model compared to others concerning the two-classes dataset. It is worth noting that the Sensitivity and Recall reflect the same results and formulas.

5.4. Three-classes dataset experiments

Table 10 summarizes the configurations related to the three-classes dataset.

Table 11 shows the TP, TN, FP, and FN of the best solutions after the learning and optimization processes on each pre-trained model concerning the thee-classes dataset. It shows that MobileNet pre-trained model has the lowest FP and FN values. On the other hand, MobileNetV3Small has the highest FP and FN values.
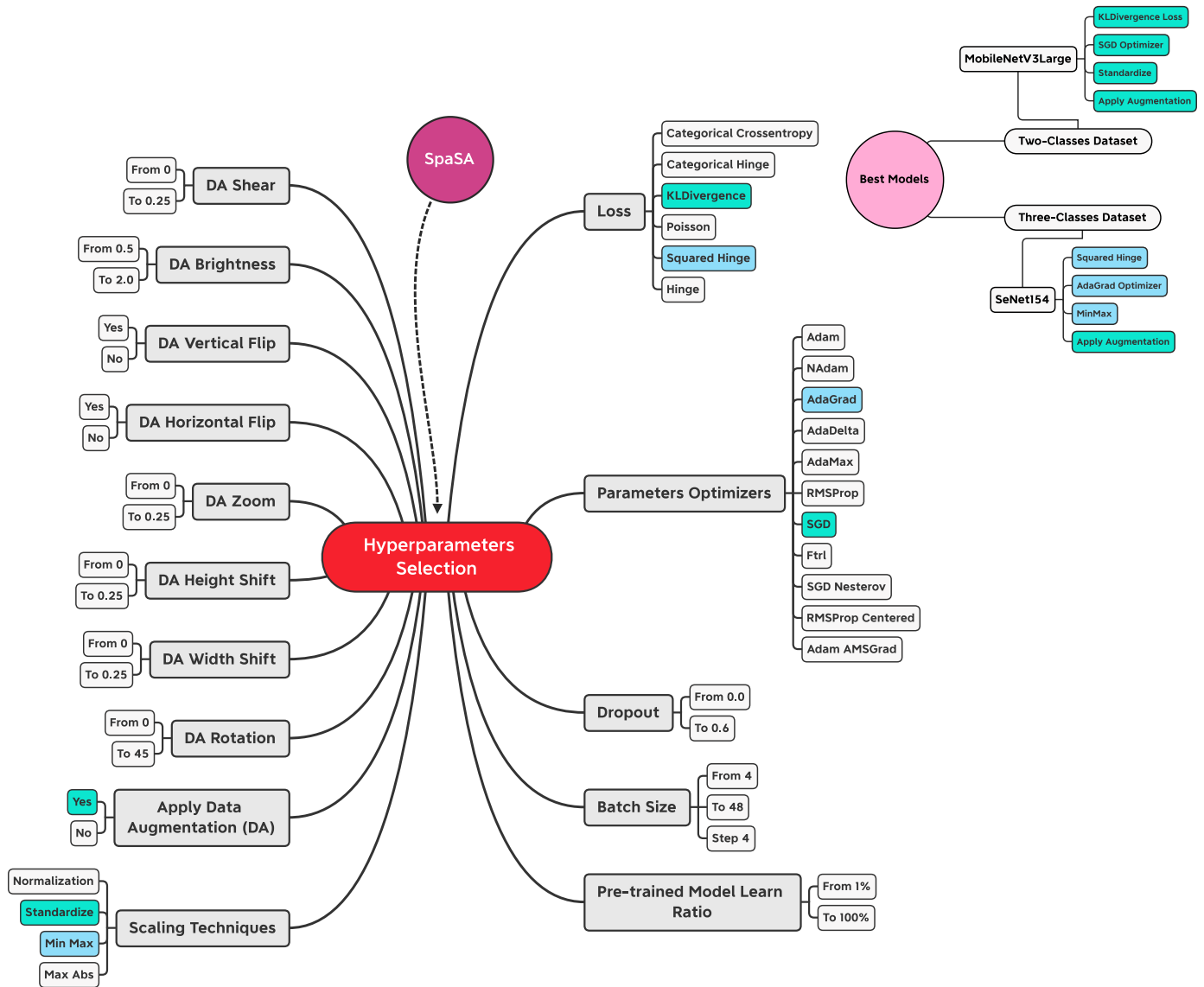
Fig. 7. Hyperparameters selection and best combinations graphical summarization.
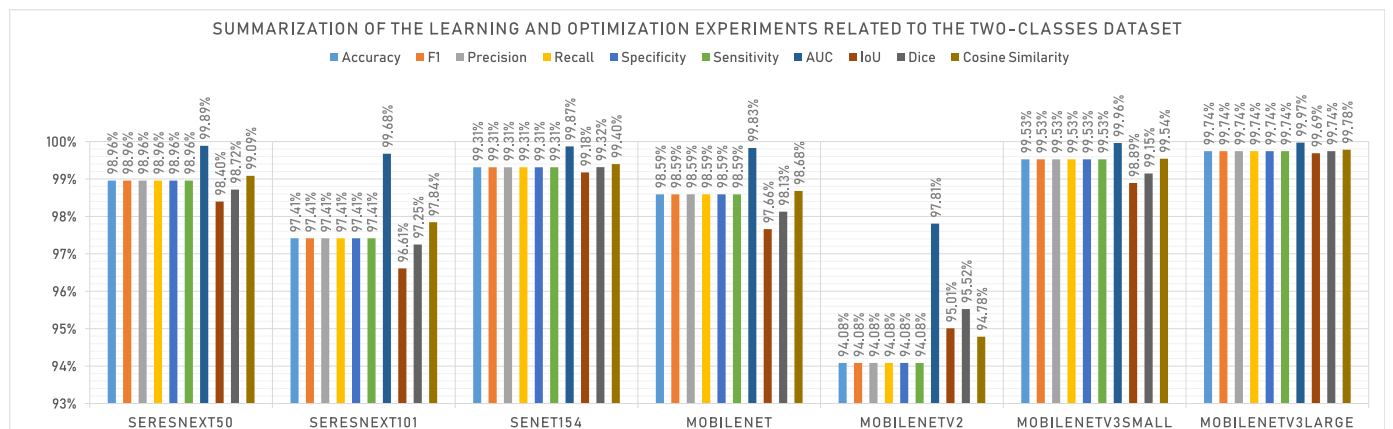

Fig. 8. Summarization of the learning and optimization experiments related to the two-classes dataset.

The best solutions combinations concerning each model are reported in Table 12. It shows that the Squared Hinge loss and AdaMax parameters optimizer are recommended by three models. The MinMax scaler and neglecting data augmentation are recommended by four models. The three models that recommended applying data augmentation, recommended also to apply horizontal flipping by 100% and ignoring
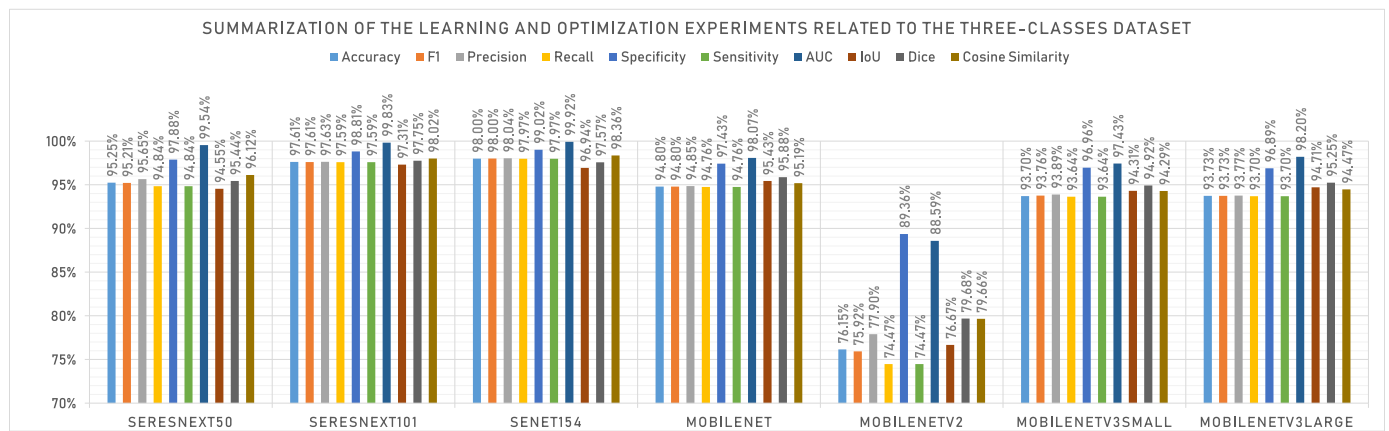
Fig. 9. Summarization of the learning and optimization experiments related to the three-classes dataset.

Table 15
A comparison between the data augmentation and train-to-test splitting approach and cross-validation approach.

Approach	Accuracy	AUC	Cosine Similarity	TP	TN	FP	FN
Data augmentation and train-to-test splitting approach	99.74%	99.97%	99.78%	14,768	14,768	392	392
Cross-validation approach	84.99%	92.60%	87.72%	2,581	2,581	455	455

vertical flipping by 66.67%.

From the values reported in Table 11 and the learning history, we can report different performance metrics. The reported metrics are partitioned into two types. The first reflects the metrics that are required to be maximized (i.e., Accuracy, F1, Precision, Recall, Specificity, Sensitivity, AUC, IoU, Dice, and Cosine Similarity). The second reflects the metrics that are required to be minimized (i.e., Categorical Cross-entropy, KLDivergence, Categorical Hinge, Hinge, SquaredHinge, Poisson, Logcosh Error, Mean Absolute Error, Mean IoU, Mean Squared Error, Mean Squared Logarithmic Error, and Root Mean Squared Error). The first category metrics are reported in Table 13 while the second is in Table 14.

From them, we can report that the SeNet154 pre-trained model is the best model compared to others concerning the three-classes dataset. It is worth noting that the Sensitivity and Recall reflect the same results and formulas.

5.5. Graphical summarizations

From the experiments applied on the suggested approach, we can summarize the best combination of different alternatives in Fig. 7.

Fig. 8 and Fig. 9 present graphical summarizations of the reported learning and optimization results using the two-classes and three-classes datasets respectively.

5.6. Cross-validation comparison

An experiment is applied using cross-validation (i.e., without data augmentation) on the "MobileNetV3Large" CNN model using the following configurations: K-folds of 5, Epochs of 7, batch size of 32, 10% dropout, Adam parameters optimizer, Categorical Crossentropy loss function, SoftMax output activation function, and 10% TL learning ratio. The "MobileNetV3Large" model is selected as it reported the best metrics using the data augmentation and train-to-test splitting approach. The reported average metrics after 5-fold cross-validation are: 0.421 loss, 2,581 TP, 2,581 TN, 455 FP, 455 FN, 84.99% accuracy, 84.99% precision, 84.99% recall, 87.72% cosine similarity, and 0.926 AUC. The training took 4,484 s.

The reported metrics are lower than the reported metrics using the data augmentation and train-to-test splitting approach. Also, it took longer as it performs the training and evaluation 5 times. The latter approach concerning the used datasets is recommended. Table 15 shows a tabular comparison between the two approaches.

5.7. Optimized vs. non-optimized approaches comparison

Suppose the authors decided to formulate the problem as non-optimized CNN. In that case, we have to face challenges like a limited dataset and the low performance of a deep learning model with a limited dataset. Moreover, manual trial and error of hyperparameter settings must be addressed. In addition, we are not sure about the reported

Table 16
A comparison between the optimized and non-optimized approaches.

Approach	Accuracy	AUC	Cosine Similarity	TP	TN	FP	FN
Optimized Approach	99.74%	99.97%	99.78%	14,768	14,768	392	392
Non-optimized Approach	83.33%	91.70%	86.60%	3,164	3,164	633	633

Table 17
A comparison between the existence and non-existence of transfer learning.

Approach	Accuracy	AUC	Cosine Similarity	TP	TN	FP	FN
With Transfer Learning	99.74%	99.97%	99.78%	14,768	14,768	392	392
Without Transfer Learning	49.67%	70.70%	49.70%	1,886	1,886	1,911	1,911

Table 18
Comparison between the suggested approach and related studies.

Study	Year	Dataset	Approach	Best Accuracy
Islam et al. [18]	2020	CCT	LeNet-5 CNN	86.06%
Shibly et al. [17]	2020	CXR	COVID faster R–CNN	97.36%
Polsinelli et al. [15]	2020	CCT	Light CNN	85.03%
Tripti Goel et al. [12]	2020	CCT	CNN + GAN	99.22%
Huang et al. [21]	2020	CCT	MCSL	98.03%
Abraham and Nair [22]	2020	CCT	CNN + KSVM	91.60%
Kundu et al. [19]	2021	CCT	Fuzzy + CNN	98.93% and 98.80%
Jia et al. [14]	2021	CXR and CCT	Dynamic CNN	99.6% (CXR) and 99.3% (CCT)
Maghdid et al. [16]	2021	CXR and CCT	CNN	98%
Pathan et al. [20]	2021	CCT	Optimized CNN	98%
R. Murugan and Tripti Goel [23]	2021	CXR	E-DiCoNet	94.07%
Goura and Jain [24]	2022	CCT + CXR	DLS-CNN	98.78%
Gayathri et al. [26]	2022	CXR	FFNN	95.78%
Tripti Goel et al. [27]	2022	CXR	MOGOA	98.27%
Guoqing et al. [28]	2022	CCT + CXR	COVID-MTL	98.78%
Shaik and Cherukuri [29]	2022	CCT	DNN	93.33%
Current Study	2022	CT	Hybrid (SpaSA and CNN)	99.74% (two-classes) and 98% (three-classes)

accuracy of this model against the variability of these datasets.

The authors conduct an experiment with the best recommended hyperparameters settings to compare between the optimized and non-optimized networks to rest assured about the feasibility of the proposed framework. The experiment is applied without the meta-heuristic optimizer (i.e., SpaSA) on the "MobileNetV3Large" CNN model using the following configurations: Epochs of 7, batch size of 32, 10% dropout, Adam parameters optimizer, Categorical Crossentropy loss function, SoftMax output activation function, and 10% TL learning ratio. Data augmentation is applied with the configurations presented in Table 1. The reported metrics are: 0.4096 loss, 3,164 TP, 3,164 TN, 633 FP, 633 FN, 83.33% accuracy, 83.33% precision, 83.33% recall, 86.60% cosine similarity, and 0.917 AUC. The training took 1,238 s.

Table 16 shows a tabular comparison between the two approaches. The reported metrics are lower than the reported metrics using the SpaSA optimization approach.

5.8. Transfer learning existence comparison

The usage of transfer learning in the current study is to map the knowledge of detection of the objects and learning the patterns from the ImageNet that consists of more than 16 M images to the current dataset that consists of 15K images approximately. Without transfer learning, the number of epochs would be more than 5 epochs to reach approximately similar performance metrics.

To be more concise, an experiment is applied without using the ImageNet pretrained weights (i.e., without transfer learning) on the "MobileNetV3Large" CNN model using the following configurations: Epochs of 7, batch size of 32, 10% dropout, Adam parameters optimizer, Categorical Crossentropy loss function, SoftMax output activation function, and 10% TL learning ratio. Data augmentation is applied with the configurations presented in Table 1. The reported metrics are: 0.6933 loss, 1,886 TP, 1,886 TN, 1,911 FP, 1,911 FN, 49.67% accuracy, 49.67% precision, 49.67% recall, 70.70% cosine similarity, and 0.497 AUC.

Table 17 shows a tabular comparison between the two approaches. The experiment without transfer learning reported poor performance metrics.

5.9. Related studies comparisons

Table 18 shows a comparison between the suggested approach and related studies. It shows that the current study outperforms most of the related studies.

6. Conclusions and future work

As a complementary and enhanced method for early detection of COVID-19, CNN Deep Learning and pre-trained models have been heavily used to analyze CT image datasets. However, pre-trained CNN models are crucial for obtaining good results with a limited dataset. In addition, the hyperparameter settings strongly influence CNNs during the training phase. Therefore, CNN performs best when their hyperparameters are chosen in conjunction with their dataset. With SpaSA, we optimize the various CNN and TL hyperparameters to find the best configuration for each used pre-trained model in the current study. A pre-trained model will be initialized, and the hyperparameters will be injected. The models used in this study were SeresNext50, SeresNext101, SeNet154, MobileNet, MobileNetV2, MobileNetV3Small, and MobileNetV3Large with the weights pre-trained from ImageNet. The experiments were performed using two datasets. In the first dataset, there are two classes, while in the second, there are three. COVID-19 lung CT scans and COVID-19 CT scans are the two publicly available datasets used as the first dataset by the authors. Overall, 14,486 images were included in this study. In the second dataset, which included 17,104 images, the authors analyzed the Large COVID-19 CT scan slice dataset. According to the results, the pre-trained CNN models for MobileNetV3Large and SeNet154 deliver optimal or near-optimal results to a binary classification classifier and multiclassification classifier, respectively. Various metaheuristics will be used in future work to tweak the classifier and optimizer hyperparameters in order to validate and confirm the superiority of the Sparrow algorithm. Our ongoing work includes the combination of classifiers, as well as optimizations and adaptations to allow deployment on a smartphone or similar mobile platform.

Declaration of competing interest

The authors whose names are listed immediately below certify that they have NO conflict of interest with another person or organization that might influence the work of this study.

Acknowledgment

References

[1] Gabriel Augusto Pires de Souza, Marion Le Bideau, Celine Boschi, Lorène Ferreira, Nathalie Wurtz, Christian Devaux, Philippe Colson, Bernard La Scola, Emerging sars-cov-2 genotypes show different replication patterns in human pulmonary and intestinal epithelial cells, Viruses 14 (1) (2022) 23.

[2] Mohammad Momeny, Ali Asghar Neshat, Mohammad Arafat Hussain, Solmaz Kia, Mahmoud Marhamati, Jahanbakhshi Ahmad, Ghassan Hamarneh, Learning-to-augment strategy using noisy and denoised data: improving generalizability of deep cnn for the detection of covid-19 in x-ray images, Comput. Biol. Med. 136 (2021) 104704.

[3] Coronavirus Disease 2019 (Covid-19) Situation Reports, world health organization, 2022. https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19—6-january-2022. (Accessed 10 January 2022).

[4] Our world in data,johns hopkins university csse covid-19 data. https://ourworldindata.org/, 2022. (Accessed 10 January 2022).

[5] Waleed M. Bahgat, Hossam Magdy Balaha, Yousry AbdulAzeem, Mahmoud M. Badawy, An optimized transfer learning-based approach for automatic diagnosis of covid-19 from chest x-ray images, PeerJ Computer Science 7 (2021) e555.

[6] Davide Ferrari, Andrea Motta, Marta Strollo, Giuseppe Banfi, Massimo Locatelli, Routine blood tests as a potential diagnostic tool for covid-19, Clin. Chem. Lab. Med. 58 (7) (2020) 1095–1099.

[7] Serology Testing for Covid-19 at Cdc, 2022. https://www.cdc.gov/coronavirus/2019-ncov/lab/serology-testing.html. (Accessed 10 January 2022).

[8] Kranthi Kumar Lella, Pja Alphonse, Automatic diagnosis of covid-19 disease using deep convolutional neural network with multi-feature channel from respiratory sound data: cough, voice, and breath, Alex. Eng. J. 61 (2) (2022) 1319–1334.

[9] Qiongjie Hu, Hanxiong Guan, Ziyan Sun, Lu Huang, Chong Chen, Ai Tao, Yueying Pan, Liming Xia, Early ct features and temporal lung changes in covid-19 pneumonia in wuhan, China, Eur. J. Radiol. 128 (2020) 109017.

[10] Yicheng Fang, Huangqi Zhang, Jicheng Xie, Minjie Lin, Lingjun Ying, Peipei Pang, Wenbin Ji, Sensitivity of chest ct for covid-19: comparison to rt-pcr, Radiology 296 (2) (2020) E115–E117.

[11] Rohit Kundu, Pawan Kumar Singh, Seyedali Mirjalili, Ram Sarkar, Covid-19 detection from lung ct-scans using a fuzzy integral-based cnn ensemble, Comput. Biol. Med. 138 (2021) 104895.

[12] R Murugan Tripti Goel, Seyedali Mirjalili, Deba Kumar Chakrabartty, Automatic screening of covid-19 using an optimized generative adversarial network, Cognitive computation (2021) 1–16.

[13] Jiankai Xue, Bo Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Syst. Sci.Control Eng. 8 (1) (2020) 22–34.

[14] Guangyu Jia, Hak-Keung Lam, Yujia Xu, Classification of covid-19 chest x-ray and ct images using a type of dynamic cnn modification method, Comput. Biol. Med. 134 (2021) 104425.

[15] Matteo Polsinelli, Luigi Cinque, Giuseppe Placidi, A light cnn for detecting covid-19 from ct scans of the chest, Pattern Recogn. Lett. 140 (2020) 95–100.

[16] Halgurd S. Maghdid, Aras T. Asaad, Kayhan Zrar Ghafoor, Ali Safaa Sadiq, Seyedali Mirjalili, Muhammad Khurram Khan, Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning algorithms,

[17] Multimodal Image Exploitation and Learning 2021 11734 (2021) 117340E. International Society for Optics and Photonics.

[17] Kabid Hassan Shibly, Md Tahzib-Ul Islam Samrat Kumar Dey, Md Mahbubur Rahman, Covid faster r–cnn: a novel framework to diagnose novel coronavirus disease (covid-19) in x-ray images, Inf.Med.Unlocked 20 (2020) 100405.

[18] Md Rakibul Islam, Abdul Matin, Detection of covid 19 from ct image by the novel lenet-5 cnn architecture, in: 2020 23rd International Conference on Computer and Information Technology (ICCIT), IEEE, 2020, pp. 1–5.

[19] Rohit Kundu, Hritam Basak, Pawan Kumar Singh, Ahmadian Ali, Massimiliano Ferrara, Ram Sarkar, Fuzzy rank-based fusion of cnn models using gompertz function for screening covid-19 ct-scans, Sci. Rep. 11 (1) (2021) 1–12.

[20] Sameena Pathan, P.C. Siddalingaswamy, Preetham Kumar, M.M. Manohara Pai, Tanweer Ali, U Rajendra Acharya, Novel ensemble of optimized cnn and dynamic selection techniques for accurate covid-19 screening using chest ct images, Comput. Biol. Med. 137 (2021) 104835.

[21] Zhongwei Huang, Haijun Lei, Guoliang Chen, Haimei Li, Chuandong Li, Wenwen Gao, Yue Chen, Yaofa Wang, Haibo Xu, Guolin Ma, et al., Multi-center sparse learning and decision fusion for automatic covid-19 diagnosis, Appl. Soft Comput. 115 (2022) 108088.

[22] Bejoy Abraham, Madhu S. Nair, Computer-aided detection of covid-19 from ct scans using an ensemble of cnns and ksvm classifier, Signal, Image and Video Processing (2021) 1–8.

[23] R. Murugan, Tripti Goel E-diconet, Extreme learning machine based classifier for diagnosis of covid-19 using deep convolutional network, J. Ambient Intell. Hum. Comput. 12 (9) (2021) 8887–8898.

[24] Mahesh Gour, Sweta Jain, Automated covid-19 detection from x-ray and ct images with stacked ensemble convolutional neural network, Biocybern.Biomed. Eng. 42 (1) (2022) 27–41.

[25] R. Murugan, Tripti Goel, Seyedali Mirjalili, Deba Kumar Chakrabartty, Woanet: Whale optimized deep neural network for the classification of covid-19 from radiography images, Biocybern.Biomed. Eng. 41 (4) (2021) 1702–1718.

[26] J.L. Gayathri, Bejoy Abraham, M.S. Sujarani, Madhu S. Nair, A computer-aided diagnosis system for the classification of covid-19 and non-covid-19 pneumonia on chest x-ray images by integrating cnn with sparse autoencoder and feed forward neural network, Comput. Biol. Med. 141 (2022) 105134.

[27] R Murugan Tripti Goel, Seyedali Mirjalili, Deba Kumar Chakrabartty, Multi-covid-net: multi-objective optimized network for covid-19 diagnosis from chest x-ray images, Appl. Soft Comput. 115 (2022) 108250.

[28] Guoqing Bao, Huai Chen, Tongliang Liu, Guanzhong Gong, Yong Yin, Lisheng Wang, Xiuying Wang, Covid-mtl: multitask learning with shift3d and random-weighted loss for covid-19 diagnosis and severity assessment, Pattern Recogn. 124 (2022) 108499.

[29] Nagur Shareef Shaik, Teja Krishna Cherukuri, Transfer learning based novel ensemble classifier for covid-19 detection from chest ct-scans, Comput. Biol. Med. 141 (2022) 105127.

[30] Veysari, Elham Fazeli, A new optimization algorithm inspired by the quest for the evolution of human society: human felicity algorithm, Expert Syst. Appl. (2022) 116468.

[31] Krishna Gopal Dhal, Swarnajit Ray, Arunita Das, Sanjoy Das, A survey on nature-inspired optimization algorithms and their application in image enhancement domain, Arch. Comput. Methods Eng. 26 (5) (2019) 1607–1638.

[32] Slowik Adam, Halina Kwasnicka, Evolutionary algorithms and their applications to engineering problems, Neural Comput. Appl. (2020) 1–17.

[33] Mehrad Aria, Covid-19 lung ct scans, Available from, https://www.kaggle.com/mehradaria/COVID19-lung-ct-scans, 2021.

[34] Dr Surabhi Thorat, Covid 19 ct scan dataset, Available from, https://www.kaggle.com/drsurabhithorat/COVID-19-ct-scan-dataset, 2020.

[35] Maede Maftouni, Large covid-19 ct scan slice dataset, Available from, https://www.kaggle.com/maedemaftouni/large-COVID19-ct-slice-dataset, 2021.