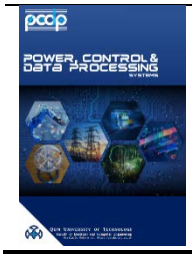
DOI: 10.30511/pcdp.2025.2063435.1030 - ISSN: 3060-8376 - Vol 2(3), 2025



POWER, CONTROL AND DATA PROCESSING SYSTEMS

Available Online at: https://pcdp.qut.ac.ir/

Transfer Learning-Based Detection of COVID-19 Cases from Chest CT Scans

ARTICLE INFO

Article Type
Original Research

Authors
Safiye Ghasemi[1*]
Somayeh Ghasemi[2]
Amir Masoud Rahmani[3]
Roohollah Barzamini[4]

[1] Department of computer, Sep.C., Islamic Azad University, Sepidan, Iran
[2] Department of computer, Christs School, London, UK
[3] Future Technology Research Center, National Yunlin University of Science and Technology, Douliou, Yunlin 64002, Taiwan
[4] Department of Electrical Engineering, CT.C. Islamic Azad University, Tehran, Iran

**** Correspondence***
Address: Department of computer, Sep.C., Islamic Azad University, Sepidan, Iran.
Phone: -
Fax: -
sf.ghasemi@iau.ac.ir

Article History
Received: June 11, 2025
Accepted: July 25, 2025
ePublished: September 01, 2025

ABSTRACT

When taking into account the prevailing COVID-19 pandemic scenario, by early detecting COVID-19, we can formulate an effective treatment plan and make decisions regarding disease containment. As a result of this issue, Artificial Intelligence (AI) specialists have been encouraged to develop models that employ deep learning techniques in COVID-19 detection. These models diagnose infection severity rapidly and economically. The present study proposes a Deep Convolutional Neural Network (CNN) model based on PSO, which helps identify COVID-19 infections by chest Computerized Tomography (CT) scans. Moreover, we demonstrate how pre-trained models can classify the disease through transfer learning. Initially, the random search is used to identify an optimal CNN model. The transfer learning strategy presents an analysis of several popular pre-trained models. The optimal CNN model inherits several layers from these previously trained models, and we then fine-tune the selected optimal CNN model accordingly. The proposed architecture is built using three pre-trained models with the highest quality. PSO algorithm is applied to estimate how each pre-trained model will affect the ultimate detection of the suggested model. To train the model, we analyzed two publicly available datasets COVID-CT and SARS-CoV-2 applying distinct pre-processing techniques to each. According to the experimental results, our PSO-based configuration optimization performed well on this dataset and can achieve better results with more training data. As a result of extensive parameter tuning, the proposed model can identify COVID-19 with an accuracy of up to 90.32%. This model will facilitate the detection and diagnosis of COVID-19 promptly.

Keywords: Computerized Tomography (CT) Scans, Convolutional Neural Network (CNN), Coronavirus, Deep learning, Pre-trained models, Transfer learning, Particle Swarm Optimization (PSO).

1 Introduction

Significant developments have occurred in infection control history since the incidences of COVID-19. After first appearing in Wuhan, China, a global pandemic has been caused by this virus. Screening infected patients for COVID-19 is vital to fighting the disease [1]. Computed Tomography (CT) and Chest X-rays (CXR) are vital for confirming a COVID-19 pneumonia diagnosis and monitoring its progression. CT scans can show ground-glass opacities and consolidations associated with COVID-19; CXR can also show similar findings but is less sensitive than a CT scan. CT has been demonstrated to comment RT-PCR testing and screen for COVID-19 infections. There were some cases in which the patients initially had negative PCR tests; however, following a CT scan, it was confirmed that the patient was, in fact, infected. Since manual checking of CT images might not be feasible in emergencies, automated detection tools are necessary to speed up diagnosis. This can provide more detailed information about the pathology and provide a better classification of lung involvement, which may affect prognosis.

Artificial Intelligence (AI) can widely benefit physicians and prove to be a great help in decision-making skills in the COVID-19 pandemic, where sensitivity in treatment is paramount. Machine learning technologies, including deep learning, are experiencing rapid development and have captured considerable public attention. The diagnosis of COVID-19 has commonly relied on machine learning methods including transfer learning and Deep Convolutional Neural Networks (CNNs). CNNs are extensively used in deep learning studies for COVID-19 detection [2]. Lung CT scans can be automatically processed through deep learning techniques, accelerating the analyzing process. These techniques enable training neural networks from scratch on massive databases as well as fine-tuning the pre-trained networks on small ones [3]. As one of the most widely studied deep learning models, they offer superior performance for image quantification, segmentation, classification, and detection due to advances in computing power and learning algorithms [4]. As an attempt to detect COVID-19 from CT scans, numerous implementations have already been proposed that utilize deep learning approaches. ResNet-101 and Xception emerge as the most effective neural networks in a new study based on a comparison of ten well-known CNN architectures for detecting COVID-19 [4, 5]. Using a CNN architecture, [6] presents an efficient and accurate method for screening of COVID-19 cases using lung CT scans. Consequently, different scanning methods have been utilized by researchers in [7] for automating the diagnosis of COVID-19.

Our prior knowledge and experience guide our approach to every upcoming task. Unlike humans, neural networks learn entirely from scratch based on random weight initialization. Having the network pre-trained on an entirely different task will help us bridge this initial knowledge gap. In the literature, several transfer learning strategies have been presented to detect COVID-19 [2, 8, 9, 10, 11]. Transfer learning improves detection and diagnosis performance. It avoids the weakness of small cases by using pre-training weights as a basis for learning the desired task. Transfer learning has been effective in many fields. When large datasets are available, models can improve performance and reduce the need for extensive labeling on tasks with limited labeled data. Besides, some researchers apply heuristic algorithms to optimize the performance of classification models. PSOCNN [12] is a recent deep network generation algorithm that takes advantage of the remarkable potentials of global search in a Particle Swarm Optimization (PSO) algorithm. A performance comparison is made with the test dataset after training the model on the training dataset. It is challenging to perform COVID-19 classification using CNNs; first, the public datasets for this task are relatively small (hundreds or thousands) compared to other image classification datasets (up to millions). In addition, COVID-19 classification task examples differ very little, whereas most existing CNN classifiers, such as dogs and cars, deal with clearly distinct classes.

With the ongoing challenges posed by the COVID-19 pandemic, significant advancements have been made in the realm of AI and deep learning for effective detection methodologies. Recent studies, such as those conducted by Shaik et al. [2], Nettur et al. [13], Ali et al. [14], Huang et al. [15], and Kathamuthu et al. [16], have showcased enhanced models that facilitate timely and accurate COVID-19 screenings. This paper proposes a CNN model that builds upon these innovations, offering superior performance by integrating transfer learning techniques and optimization algorithms tailored for CT scan analysis. In this model, an enhanced six-layer CNN architecture performs better than its counterparts and is applied to transfer learning. Our proposed model combines several pre-trained models; to achieve optimal outcomes, the applied configuration of the combined pre-trained models is optimized using PSO. The effectiveness of pre-trained models significantly impacts the outcomes of the PSO algorithm. Thus, assessing the robustness of the proposed PSO-based CNN model across various datasets is essential. Therefore, we have expanded our research to include additional datasets that encompass various imaging conditions, including different resolutions and data obtained from multiple imaging devices and scanner types. This enhancement aims to strengthen our findings and ensure that our model demonstrates consistent performance across diverse scenarios relevant to clinical practice. This strategy ensures a more effective learning schema than previous studies. The proposed model makes the following contributions:

- The most influential CNN architecture is determined through a random search process using various convolutions, weight sharing, layers, and filter families.
- Various pre-trained CNNs were utilized to identify the deep features in order to be applied to the most effective architecture to find the most efficient pre-trained models.

- A fast and reliable intelligence algorithm (PSO) is applied to the combination of pre-trained models to determine their effect on detecting COVID-19 infection.
- COVID-CT and SARS-CoV-2 datasets are used to fine-tune and analyse the proposed architecture.
- As a result of this model, physicians, radiologists, and field specialists can make more informed decisions.
- Using the proposed model, you can perform retrospective evaluations and reduce misdiagnosis rates.

The rest of the paper is structured as follows. Following is a discussion of related work. Section 3 provides an explanation of the materials and methods. This section overviews the transfer learning strategy we propose. Results of the experiments are analyzed in section 4. Section 5 summarizes the main conclusions.

2 Related works

This section discusses some of the latest COVID-19 research involving AI algorithms, starting with CNN-based methodologies for detecting and diagnosing COVID-19 in affected patients. As a starting point, we present CNN-based methodologies for detecting and diagnosing COVID-19 in affected patients. Then, we explore transfer learning methods utilized in developing COVID-19 classifiers based on AI and examine various chest image processing techniques. The studies mentioned in this section can be supplemented by further research. Furthermore, based on the studies, network architecture, and images are both highly suitable for diagnosing COVID-19 infection.

2.1 CNN-based methods

The advent of machine learning algorithms has facilitated the evaluation of massive data sets. It has been demonstrated that CNNs can significantly contribute to computer vision, speech recognition, and robotic control [17]. In various machine learning tasks, CNN-based deep learning models are state-of-the-art and prevalent in the field of medical. Various studies have incorporated these techniques for a variety of purposes, including diagnosis and estimation of cancer and Alzheimer's disease, biopsy, and dermoscopy analysis [8]. Different CNN models were analyzed using chest CT scans or X-rays images. According to Mahbub et al. [18], an accuracy rate of 99.50% for the applied CXR images was achieved using a deep feature-based CNN model. In [19], Goncharov et al. developed a deep feature-based approach on the basis of two major configurations: Prioritizing COVID-19-infected patients for rapid isolation; Identifying patients with severe COVID-19 and providing them with urgent medical treatment based on severity; a CNN model with high ROC AUC for classification and prediction of the percentage of infected lungs is provided. Researchers used various models and analyzed their performance on lung CTscans or X-ray images. A deep convolutional model was employed in a study by [11] to detect infections caused by COVID-19 on the basis of CT scans with tuning parameters. This study aimed to diagnose infections associated with COVID-19 using ten well-known CNNs, including AlexNet, Xception, ResNet-50, ResNet-101, MobileNet-V2, ResNet-18, VGG-19, VGG-16, SqueezeNet, and GoogleNet. Xception and ResNet-101 networks achieved the most promising results of all the networks.

Three classes of chest images were analyzed in the study [20]: pneumonia images, COVID-19 images, and typical lung scans. Prior to training deep learning models on images, each image was preprocessed. The fuzzy color technique was applied as a preprocessing phase to restructure the data and stack them with the original ones. Following the training of the resulted dataset with MobileNetV2 and SqueezeNet architectures, the features were analyzed by the Social Mimic method. Finally, they were classified based on Support Vector Machine (SVM).

Recent studies have further advanced this field. The study by Nettur et al. [13] introduces a hybrid deep learning model for early COVID-19 detection through CT scans, combining VGG16 and DenseNet121 while employing PCA and SVM for enhanced classification accuracy. In another research effort, Ali et al. [14] developed a two-phase AI engine for classifying COVID-19 pneumonia levels through modified CNN and K-nearest Neighbor (KNN) approaches. Kathamuthu et al. [21] evaluated deep learning-based CNNs integrated with transfer learning, highlighting VGG16's superior accuracy for COVID-19 detection using CT scans.

2.2 Transfer learning-based methods

Human beings bring much prior experience and knowledge to each challenging task. In contrast, neural networks are often required to learn tasks entirely from scratch due to random weight initializations. The transfer learning approach addresses this initial knowledge gap by pre-training the network on a completely different task; the weights learned during pre-training are applied as a basis for learning the desired task. Several lung infections, including COVID-19, were studied using deep transfer learning techniques on CXRs and CT scans. For diagnosing COVID-19 infection by CT scans, an essential transfer learning technique was developed that can be used on devices that are limited in resources [9]. According to Das et al. [3], using the Xception model, an automated approach was developed to screen this pandemic using chest CXR images. Regarding a study conducted by Ohata et al. [10], CNNs with transfer learning were combined to classify X-rays images of infected and uninfected individuals automatically. A total of 144 experiments were performed with 12 CNNs and multiple classifiers, including Random Forest, Bayes, Multilayer Perceptron (MLP), SVM, and KNN. To automatically classify COVID-19, popular deep-learning configurations based on feature identification were discussed [11]. Various machine learning classifiers were used to categorize subjects into COVID-19 cases or controls using extracted features from X-ray and CT images. For automatically identifying COVID-19, Kassani et al. [11]

compare popular frameworks for extracting features using deep learning techniques. Several deep CNNs were used to extract features, including NASNet, ResNet, DenseNet, Xception, MobileNet, VGGNet, and Inception to identify subjects with COVID-19 or controls. After obtaining the extracted features, several machine learning classification algorithms were used to classify subjects to infected and healthy. In evaluating the model, the feature extractor based on DenseNet121 and the classifier based on Bagging tree performed best on the publicly available datasets of lung CT and CXR.

The proposed method in [21] uses different types of CovXNets to identify the virus using CXRs of various resolutions. A basic structural unit builds models by incorporating the extracted features from a variety of receptive fields with depth-wise convolutions and variable dilation rates. A stacking algorithm further optimizes them. As a next step, the previously trained convolutional layers are directly transferred to be trained with the smaller set of chests CXR along with some fine-tuning layers. The final step incorporates gradient-based localization to distinguish abnormal X-ray images of different pneumonia forms.

In particular, the optimal model highlighted by Kathamuthu et al. [16] utilized diverse CNN architectures, successfully demonstrating the effectiveness of transfer learning for accurate detection of COVID-19 via CT scans.

2.3 Augmentation and Generation Techniques on Chest Images

COVID-19 is known to cause respiratory symptoms among its first symptoms, so CT scan and CXR are used to treat the lungs and chest. CXR and CTs are critical imaging modalities for diagnosing and monitoring pneumonia. They are the most common imaging techniques used to examine patients with the disease. A variety of authors used deep learning techniques to screen COVID-19-infected individuals using CXR and CT images as a starting point [11, 22]. Researchers predicted CT imaging would become increasingly important as COVID-19 spreads [2, 6, 7, 11, 19, 23]. Moreover, some Computer-Assisted Diagnosis (CAD) systems incorporate CXR processing [1, 9, 10, 11, 17, 18, 20]. A CT scan can demonstrate lung abnormalities associated with this pandemic, such as opacities and consolidation, which can support a diagnosis; a CXR can show similar findings but with less sensitivity [11].

Various datasets of CT images are available to evaluate the approach. In order to perform COVID-19 classification using CNNs, there are some significant challenges. One of these challenges is that the public datasets for the task are relatively small (hundreds or thousands) compared to other image classification datasets (up to millions). To address this challenge, augmentation and generalization are the primary research questions. Many well-established CNN architectures have successfully utilized various forms of data augmentation. Using the data augmentation method discussed in [6], the training dataset was scaled, zoomed, resized and horizontally flipped. With data augmentation, Aslan et al. increased the number of segmented images to provide data diversity [8]. As a result of the scarcity of training samples, Alom et al. [9] applied class-specific data augmentation to resolve the class imbalance problem as well. The majority of studies perform image size reduction, image augmentation, cropping, and other pre-processing before feeding raw images to deep neural networks [8].

3 Materials and Methods

3.1 Data

This section first describes the applied dataset, then the preprocessing phase of the dataset is presented. Last, we explain the data augmentation process utilized throughout this study.

3.1.1 Dataset Description

The first step towards developing machine learning COVID-19 applications is to collect inputs. Our experiments were conducted using two publicly available datasets: the COVID-CT dataset [24] and the SARS-CoV-2 CT-Scan dataset [25]. The use of both datasets allows for a broader representation of COVID-19 manifestations and enhances the reliability of the model through cross-dataset validation.

The COVID-CT dataset has been confirmed to be useful by Tongji Hospital, Wuhan, China following the treatment of many COVID-19 patients. The open-sourced COVID-CT dataset contains 349 CT images of the COVID-19 with different types of pathologies and clinical findings related to 216 patients and 463 non-COVID-19 samples of CT scans. Among the image sources are papers published in bioRxiv, JAMA, Lancet, medRxiv, NEJM, and other papers related to COVID-19. This dataset has been demonstrated to assist with designing COVID-19 diagnosis models based on AI in further experiments. Various learning methods are developed based on the dataset, including multi-task learning [1, 23] and self-supervised learning diagnosis [3, 22].

Complementing this, we incorporated the SARS-CoV-2 CT-Scan dataset compiled by Soares et al. [25], which contains 1,252 CT images from COVID-19-positive cases and 1,230 images from non-COVID cases, including both healthy lungs and other lung pathologies. This dataset provides a diverse and comprehensive sample of CT scans, covering a wide range of disease severity and patient demographics. The inclusion of this dataset significantly strengthens the experimental setup by offering additional data variability and helping mitigate potential overfitting to a single data source.

Both datasets consist of grayscale CT images in PNG format, preprocessed to focus on relevant pulmonary regions. Upon combining the datasets, a total of 1,601 CT scans were analyzed, consisting of a balanced representation of positive

(COVID-19) and negative cases. Detailed characteristics of the images, such as their dimensions, varied across both datasets, with the average height reaching 491 pixels and the width ranging from 124 to 1,485 pixels. This variation required a methodical preprocessing approach to standardize the images before inputting them into the model. For consistency, all CT images were resized while maintaining their original aspect ratio, and regions of interest were cropped to 224×224 pixels. Importantly, while both datasets provide valuable metadata, including patient demographics and clinical status, they lack precise information about the timing of CT acquisition relative to symptom onset. This absence introduces a potential limitation in evaluating disease progression stages, which is relevant for the generalizability of automated diagnostic models.

In the next step, we used zero-phase component analysis whitening to normalize contrast and brightness [2]. A view of the input images shows three significant regions. The bulk is visible in the black area of the lung, where the patient can breathe freely. The capillaries of the lungs absorb oxygen and other gases into the blood. Damage to this area dislodges the lungs from their proper function. A light gray region infected by COVID-19, in the lower part of the lung image, is visible among the dark areas. Using a brain-computer to identify this part of the lung is difficult.

3.1.2 Dataset Preprocessing

Various data tools were used to collect the datasets, which led to different sampling rates and intensities. It is ubiquitous in computer vision applications to perform pre-processing. These methods highlight points in an image that may assist in recognition and eliminate undesirable noise. They even aid in deep learning model training. A series of standard pre-processing tasks were followed to optimize deep learning models' training process. The first step involves converting the input images into feature-wise normalized images of pixel intensity between 0 and 1. Analyzing each image pixel value by calculating its mean and variance, an image distribution with a unit standard deviation and zero-centered mean is obtained using this step. This pre-processing step, called normalization, is essential to ensure model convergence. Reshaping is also needed to ensure compatibility with architectures that identify features from normalized images. All input CTs were resized to 224×224 pixels for consistent dimensions. Next, greyscale images are generated for all CT scans of patients and computer-aided applications run on similar format images. Moreover, this step significantly reduces transaction and time costs without impairing process performance. Image sharpening is also used to enhance image clarity as part of our pre-processing phase.

3.1.3 Data augmentation

Artificially the number of samples for training should be increased. This can be achieved through data augmentation techniques. It is essential for deep learning applications that lack rich training sets. Consequently, the model can develop better generalization capabilities and avoid overfitting.

The most common data augmentation operations are cropping, rotation, padding, flipping, and noise injection. These basic operations may, nevertheless, not be appropriate Due to the complexity of medical images' structures, which include irregular tumor shapes and anatomical variation. Consequently, logical image structure can be disrupted by irrelevant alterations, potentially leading to changes in shape of the image and the incorporation of abnormal data that impacts performance of the model. For instance, flipping images, can result in inconsistencies in anatomical structure when used on medical images.

Deformable augmentation methods, such as spline interpolation and random displacement fields, have addressed the issue. Nevertheless, these techniques depend on data and can often be imitated. Generative models can now produce realistic images and simulate the underlying data distributions thanks to advances in deep learning. Compared to traditional data augmentation techniques, these synthesized images are more accurate. A more realistic and diverse data set can be generated when medical images are better coordinated and have more variability [26].

The use of deep models over traditional techniques of augmentation of data offers several advantages. These include generating realistic images that match the training dataset distribution. For example, diffusion models have been shown to create images with fine details and adequate sharpness, though they do necessitate considerable computational power to train and may introduce complexity that demands better interpretability. The GANs' up-sampling-only architecture provides finely detailed images of high quality while consuming minimal memory. Techniques like minibatch discrimination and WGAN support GAN training stabilization and increase the training set size.

Various data augmentation techniques are applied, such as height and width shifts, brightness changes, horizontal and vertical flips, and channel shifts. Additionally, we implement a minibatch discrimination method to reduce homogeneity between generated samples within a minibatch [27]. In order to gather more accurate predictions, small transformations are also performed, which are then averaged to produce a single prediction. It should result in better accuracy and robustness at test time. Section 4.2 presents a detailed analysis of the individual impact of each augmentation technique on the model's efficacy.

3.2 Transfer learning

Every neural network should be trained using randomly initialized weights. The transfer learning method, which focuses on transferring knowledge across domains, can be applied to solving problems with insufficiently labeled or unlabeled instances [28]. Transfer learning involves the following steps and methods: 1. Pre-training: Before exposing

a neural network to the target dataset, pre-training is used to initialize its weights. When working with a small dataset, networks are trained on datasets unrelated to the specific task, usually larger ones. This is to provide a better initial value for the convolutional layers. The pre-trained column parameters were developed using the CIFAR-10 dataset, which contains 60,000 32×32 images classified into ten categories. Notably, the features learned close to the neural network's input are linear edge-detecting filters, regardless of the task. 2. Feature Extraction: The next step is extracting the relevant features from the source model. The pre-trained model serves as a higher-level representation extractor, in which the generated features are fed to a new classifier for the target feature. 3. Architecture Selection: A target model architecture should be selected based on the amount of labeled data available, the complexity of the task, and the resources available. 4. Fine-tuning: During this process, the network learns general features and updates the model's parameters that can later be fine-tuned to suit the target task. A network is not fully optimized for the target task after the pre-training stage. Fine-tuning is training a pre-trained network on a specific dataset to adapt and learn features associated with specific tasks.

We fine-tuned the model using COVID-19 data after pre-training, but parameters were initialized with values learned from pre-training instead of random values. As the pre-trained model is fine-tuned, the frozen layers can be gradually unfrozen, allowing subsequent training cycles to update them. This gradual unfreezing process enables the target model to fine-tune the shared representations under task-specific data.

Finally, the network learns to make accurate predictions based on the inputs and desired outputs provided by the target dataset. During training, neural network weights and biases are optimized for a specific task or dataset. Training a neural network for classification generally involves optimizing its parameters to produce the intended output label for each input. The inputs in our training set are CT images of the chest, and the outputs are disease labels. As a result, we optimize the network to predict each label with the highest probability correctly.

3.3 The Architecture of the Proposed Model

Our first step was to collect and annotate the datasets for COVID-19 findings. Preprocessing techniques include scaling and normalizing images and data augmentation. We then used the proposed model to analyze our data.

Here are CNN architectures for categorizing normal and COVID-19 patients. The present study used a pre-trained deep CNN architecture, which performs well in image classification and detection but differs in several ways from the original [29]. There are primarily three convolutional layers in these pre-trained architectures. The last pooling layer uses the Global Average Pool (GAP). Adaptive pooling accommodates input feature maps to 150×150 pixels in greyscale. Our network now contains fewer feature maps to fit in GPU memory without sacrificing mini-batch size.

To enhance the clarity of our proposed model for COVID-19 detection, we introduce a structural illustration that depicts the architecture of our CNN and the integration of pre-trained models. This diagram outlines the key components and the flow of data through the network. As depicted in Fig. 1, as the input layer, the model begins with a grayscale CT image input resized to 224×224 pixels. The architecture comprises three different pre-trained CNN models: VGG19, DenseNet201, and ResNet50V2. Each model processes the input image independently. Features extracted from these models are combined to enhance prediction accuracy. The extracted columns of features comprise of three columns including:

Column 1: VGG19 - Consists of multiple convolutional and pooling layers designed to capture various image aspects.

Column 2: DenseNet201 - Utilizes dense connectivity patterns for efficient feature propagation.

Column 3: ResNet50V2 - Incorporates residual connections that facilitate training deeper networks.

In the combination layer, the outputs from each pre-trained model are weighted and summed, reflecting their respective influences, as determined by PSO algorithm. The final prediction is made through a sigmoid activation function that provides a probability score for each class (COVID-19 positive or negative). Finally, the model concludes with a fully connected layer that classifies the images based on the combined features in the output layer.

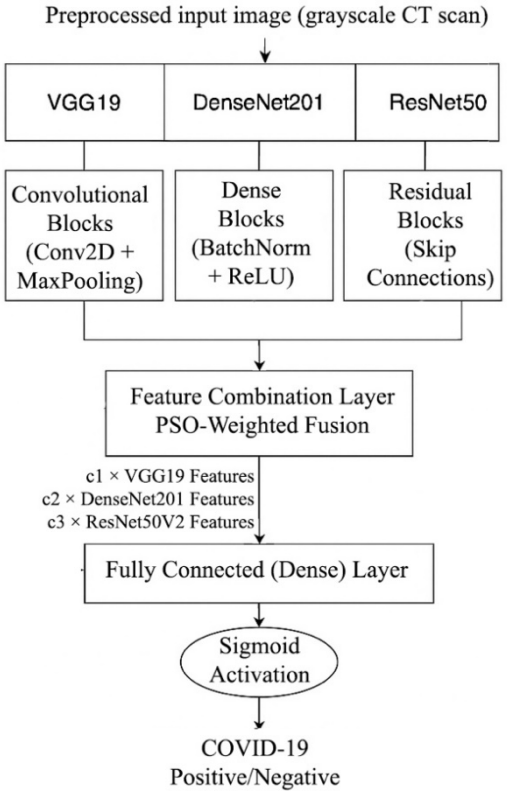


Fig. 1. Structural overview of the proposed CNN architecture for COVID-19 detection

In addition, the pre-trained models are modified not to have fully connected layers, resulting in fewer learnable parameters and higher translational invariance. The various pre-trained architectures differ in layers but use the same basic concept. A pre-trained model configuration can generally have *n* columns; each column represents a modified version of the pre-trained model. Multicolumn arrangements can be viewed as an ensemble of *n* models, each operating on a different view of lung lobe lesions.

$$y = \sigma(\sum_{i=1}^{n} c_i f_i(x_i, \theta_i)) \quad (1)$$

As shown in Eq. (1), the final prediction of the model, *y*, is determined by summing the column outputs. The sigmoid function (σ) produces a reasonable probability between 0 and 1. A pre-trained model instance is $f(x_i,\theta_i)$ in column *i*. Here, x_i refers to the *i*th planar view of the lesion, θ_i refers to its parameters, and c_i is the learned importance weighting of the instance *i*.

Several techniques were examined in addition to transfer learning to maximize classification accuracy. The following subsections explore these techniques in more detail.

3.3.1 The PSO-based Configuration Optimization

The performance for this study was significantly optimized by using three planar views while maintaining essential features for accurate classification. These three planar views represent three columns, each of which indicates a pre-trained model that performs more efficiently than those examined. As a result, the proposed model contains three columns: pre-trained model 1, pre-trained model 2, and pre-trained model 3. Eq. (1) for this three-column configuration is expressed as Eq. (2).

$$y = \sigma(c_1 f_1(x_1, \theta_1) + c_2 f_2(x_2, \theta_2) + c_3 f_3(x_3, \theta_3)) \quad (2)$$

To optimize the configuration of our proposed model, we employed the PSO algorithm. Applying PSO, we determined the optimal values for c_1, c_2, and c_3, which reflect the effects of the pre-trained models on the final output. This optimization algorithm creates a swarm of particles that explore a search space according to their best positions, global best (gbest), and personal best (pbest). As part of its search space exploration, PSO uses the best solutions, pbest and gbest, respectively. At the end of each iteration, each particle's position vector is updated by adding the particle's velocity, determined by three major components: inertia weight, cognitive component, and social component: inertia weight (current velocity), cognitive component (pbest), and social component (gbest). The initialization parameters employed in our implementation are as follows:

- Population Size: 30 (the number of particles in the swarm).
- Inertia Weight (w): 0.7 (provides a balance between exploration and exploitation).
- Cognitive Parameter: 1.5 (encourages particles to move towards their own best-known position).
- Social Parameter: 1.5 (motivates particles to move towards the best-known position of the swarm).

PSO algorithm for three-column configuration adjustment, explained in Algorithm 1. In the first step, the training and test datasets of each image database are obtained. Initially, a group of particles is generated, with each particle presenting a potential value for the parameters given. In particular, an updated weighted position updating procedure is used to explore the search space. In the event that the current solution is more effective than the previous one, we use the average loss in entropy of the training set to update both *gbest* and *pbest* appropriately. As a result of the search process, the global best position is determined as the optimal value to apply to the model. After training with a relatively large training epoch, the model is tested on an unknown dataset.

Algorithm 1 Configuration for optimal deep convolutional model based on PSO

1: **procedure** Creating three-column configuration using PSO
2: Data sets for training and testing are initialized
3: The population of the swarm is initialized
4: **for** (each *i* iteration) **do**
5: **for** (each particle *P* in the population of the swarm) **do**
6: A new configuration according to particles' current position
7: *loss* ← *Configurstion*()
8: **if** *loss* < *pbest* **then**
9: *pbest* ← *loss*
10: Local best position is updated
11: **end if**
12: **if** *loss* < *gbest* **then**
13: *gbest* ← *loss*
14: Global best position is updated
15: **end if**
16: Particle position is updated using a weighted position updating method
17: **end for**
18: **end for**
19: Make the initialization of the best model (*gbest*)
20: Use the training dataset and a more extended training period to train the finalized network
21: Use the unseen test set to test the final model
22: **end procedure**

3.3.2 Batch Normalization

Our learning rates are much higher when we use batch normalization, and our initialization is less precise, and even eliminate dropouts occasionally. In deep neural networks, batch normalization is a technique that normalizes the activations at intermediate layers [30]. The technique stabilizes layer input distributions in order to improve neural network training; this is accomplished by adding additional layers that control the first two moments, i.e., mean and variance, of the distribution. A constant standard deviation and zero mean of input data to neural networks benefit neural network training. A d-dimensional layer input $x = (x^{(1)} \ldots x^{(d)})$ treated as a vector, and an input set over a training set is known

as *X*. By using the transformation in Eq. (3), normalization can be expressed as

$$\hat{x} = Norm(x, X) \quad (3)$$

Not only does $\hat{x}$ depend on *x* as the training sample, but it also depends on all examples *X*.

For each layer that has an input with *d* dimension, $x = (x^{(1)} \ldots x^{(d)})$, the process of normalization is as follows, Eq. (4).

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \quad (4)$$

Based on the training data set, we compute Expectancy (*E*) and Variance (*Var*).

To speed up processing, mini-batches are normalized instead of the entire training set. We would rely on the entire training set to normalize activations steps are conducted in a batch setting. In the case of stochastic optimization, this is not possible. During stochastic gradient back-propagation mini-batches estimate the mean and variance of each activation allowing us to fully incorporate the statistics used for normalization during gradient back-propagation. Activations that are normalized based on the mini-batch can be helpful in training but not for inference; the output should depend only on the input. Algorithm 2 depicts the batch normalization applied in a mini-batch to activation x. This algorithm requires the learning of parameters β and γ.

As a result of mini-batch normalization, the proposed model learned more effectively and generalized well to new examples.

Algorithm 2 Applying the batch normalization to *x* using mini-batch.

Input: A mini-batch with *x* values: $B = \{x1..m\}$

Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$

Step 1: Computing mini-batch mean: $\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i$

Step 2: Computing mini-batch variance: $\sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2$

Step 3: normalizing input data: $\widehat{x_i} = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}$

4 Experimental Analysis and Discussion

As shown in Eq. (1), the final prediction of the model, *y*, is determined by summing the column outputs. The sigmoid function (s) produces a reasonable probability between 0 and 1. A pre-trained model instance is *f(xi, qi)* in column *i*. Here, *xi* refers to the *i*th planar view of the lesion, *qi* refers to its parameters, and *ci* is the learned importance weighting of the instance *i*.

Several techniques were examined in addition to transfer learning to maximize classification accuracy. The following subsections explore these techniques in more detail.

4.1 A comparison of different CNN variants on COVID-CT datasets

Several variants of the CNN architectural model with the highest performance are explored in this section. These variants are explored using the dataset of COVID-CT as a training, validation, and testing dataset. As described in [11], architectures are modified by tuning the parameters in Table 1, including pooling layers, filter size, number of filters, strides, dropout rate, and the like. Table 2 represents how many CT scans were applied as train, validation, and test data for the convolutional models. Table 3 compares the experimental models used in this study. We show all CNN architectures, detailed descriptions of the network architecture of each experiment, and the configurations they use in this table. All experiments used 5914 images, with 3089 COVID-19 cases and 2825 non-COVID samples. A total of 4435 images were uploaded for train as well as 887 for validation. Once all the training data has been fed into the networks, 25 epochs are needed for the model training process. In these experiments, the rate of learning is 0.005, with a binary cross-entropy as the function for loss computation and an Adam as the optimizer. We conducted additional statistical analyses to evaluate the significance of performance differences among the models. An ANOVA test was applied to assess the variation in model performance metrics across multiple runs, determining the statistical significance of improvements achieved by the proposed models.

TABLE 1. Optimizing the parameters of network based on the available search ranges.

Parameter	***Description***	***Range***
Depth	Number of layers	3 to 7
Convolution	Kernel k	(3,3), (4,4), (5,5)
filters	detect spatial patterns	(16, 32, 64, 128, 256)
Pooling	reduce the dimensions of the hidden layers	(2,2), (3,3), (4,4), (5,5)
Strides	the amount of movement over the image	1, 2, 3
Dense units	Number of units in fully connected layers	64, 128, 256
Dropout rate	Number of dropped neurons	0.2, 0.3, 0.4, 0.5

TABLE 2. Details of the datasets of training, validation, and testing.

Dataset	**Types**	**Number of existed images**			
		Samples	**Training set**	**Validation set**	**Testing set**
Main	Normal	397	4,435	887	592
	Infected	349			
Augmented	Normal	2,428			
	Infected	2,740			

Note that Architecture 5 is approximately a hybrid configuration combining Architecture 1 and 4. Furthermore, it differs from architecture to architecture how many features it extracts from a single image; it is not discussed in this study to avoid details. Each configuration requires an input image of 150´150 pixels. This step involves training the model using

90% dataset (75% train and 15% validation). The configuration of the classifier on the training set is determined by the hyper-parameters outlined in Table 1. Configured classifiers perform a grid search for 25 iterations. All classifiers use K-fold (with *K*=10) to determine their hyper-parameters. The best-obtained results of different CNN models are depicted in Table 3. These architectures are evaluated in the next subsections of this study.

TABLE 3. Description of different architectural variants of CNN.

Architectures	Architectures of different investigated CNN models			
	layer	Conv2D, MaxPooling	Dropout	Dense
Model 1	3	[(32, (3,3), valid) + ((4,4), 2, valid] [(32, (5,5), valid) + ((4,4), 3, valid] [(128, (5,5), valid) + ((4,4), 2, valid]	0.3	128
Model 2	4	[(32, (3,3), same) + ((3,3), 3, same] [(64, (3,3), same) + ((4,4), 2, valid] [(128, (5,5), same) + ((4,4), 2, same] [(128, (5,5), same) + ((4,4), 2, valid]	0.5	64
Model 3	5	[(32, (3,3), same) + ((3,3), 2, same] [(64, (3,3), same) + ((4,4), 3, valid] [(128, (5,5), same) + ((3,3), 2, valid] [(128, (3,3), same) + ((4,4), 2, same] [(128, (4,4), same) + ((4,4), 2, valid]	0.4	128
Model 4	6	[(32, (3,3), same) + ((3,3), 2, same] [(64, (5,5), same) + ((3,3), 2, same] [(128, (3,3), same) + ((4,4), 3, same] [(128, (5,5), same) + ((3,3), 3, same] [(128, (5,5), same) + ((3,3), 3, same] [(128, (3,3), same) + ((4,4), 3, same]	0.3	128
Model 5	7	[(32, (3,3), same) + ((3,3), 3, same] [(64, (4,4), same) + ((4,4), 2, same] [(128, (4,4), same) + ((4,4), 2, same] [(128, (3,3), same) + ((3,3), 2, same] [(128, (5,5), same) + ((4,4), 2, same] [(256, (5,5), same) + ((4,4), 2, same] [(256, (5,5), same) + ((4,4), 2, same]	0.3	128

4.2 Evaluation of COVID-CT dataset performance with the best model structure

This section examines comparing and evaluating different CNN models introduced in Table 3. In order to evaluate models, different metrics are used. We evaluated different performance evaluation parameters. In each model, we count the true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP). Thus, we compute the metrics listed below.

- *Accuracy (ACC)*: The fraction of correctly classified images determines accuracy.
- *Precision (PREC)*: A positive predictive value, or precision, is the percentage of correct positive detections.
- *Sensitivity (SEN) or Recall*: Out of all positive cases, the proportion that was correctly detected.
- *Specificity (SPE)*: The fraction of all negative cases that are correctly classified determines specificity.
- *F1_score*: This score indicates the approach performance through its harmonic mean of Sensitivity and Precision.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (3)$$

$$Specificity = \frac{TN}{TN+FP} \quad (4)$$

$$F1\_score = \frac{2*TP}{2*TP+FP+FN} \quad (5)$$

As part of our evaluation of our binary models' performance, we conducted a cross-validation (10-fold). Randomly, in total, five equal parts of the training set were used. A CNN model was trained using four of the five sets, and the fifth was used for validation. The validation and training sets were shifted ten times during this procedure. We calculated the final model performance based on the average of each fold. Fig. 2 shows loss and accuracy plots over the train and validation iteration for the competitive models proposed in Table 3.

Compared to the other models, model 4 performs better. The loss values of this model are lower than those of other models. It is observed that some models oscillate a great deal while others are more stable. There appears to be less oscillation in model 4 after the 13th epoch. Table 4 provides comprehensive and detailed performance metrics of each model. Model 4 has significantly higher accuracy comparing to the other models. As a whole, models 4 and 5 are the highest. The fourth and fifth models perform better in accuracy, precision, and other characteristics in comparison with the other models. The values of performance from testing results are presented in Table 4.

TABLE 4 The CNN architectures analysis for the classification of COVID-19

Architectures	Dataset	Test time(s)	Accuracy	Precision	Sensitivity	Specificity	F1_score
Model 1	Main	0.3652	0.7977528	0.8247422	0.6837606	0.8866667	0.7476635
	Augmented	0.4104	0.8372098	0.7884615	0.8039215	0.78846154	0.7961165
Model 2	Main	0.3814	0.8	0.7647058	0.8053097	0.79562043	0.7844827
	Augmented	0.3873	0.8609022	0.8244274	0.8852459	0.84027777	0.8537549
Model 3	Main	0.3571	0.7971014	0.7757009	0.7217391	0.7757009	0.7477477
	Augmented	0.3876	0.8372093	0.7884615	0.8039215	0.78846154	0.7961165
Model 4	Main	0.3814	0.8	0.7647058	0.8053097	0.79562043	0.7844827
	Augmented	0.3293	**0.9004092**	**0.8909657**	0.8827160	**0.9144254**	**0.8868217**
Model 5	Main	0.4176	0.8045977	0.7723577	0.8050847	0.8041958	0.7883817
	Augmented	0.4574	0.862606	0.8195266	**0.8849841**	0.8447837	0.8509984

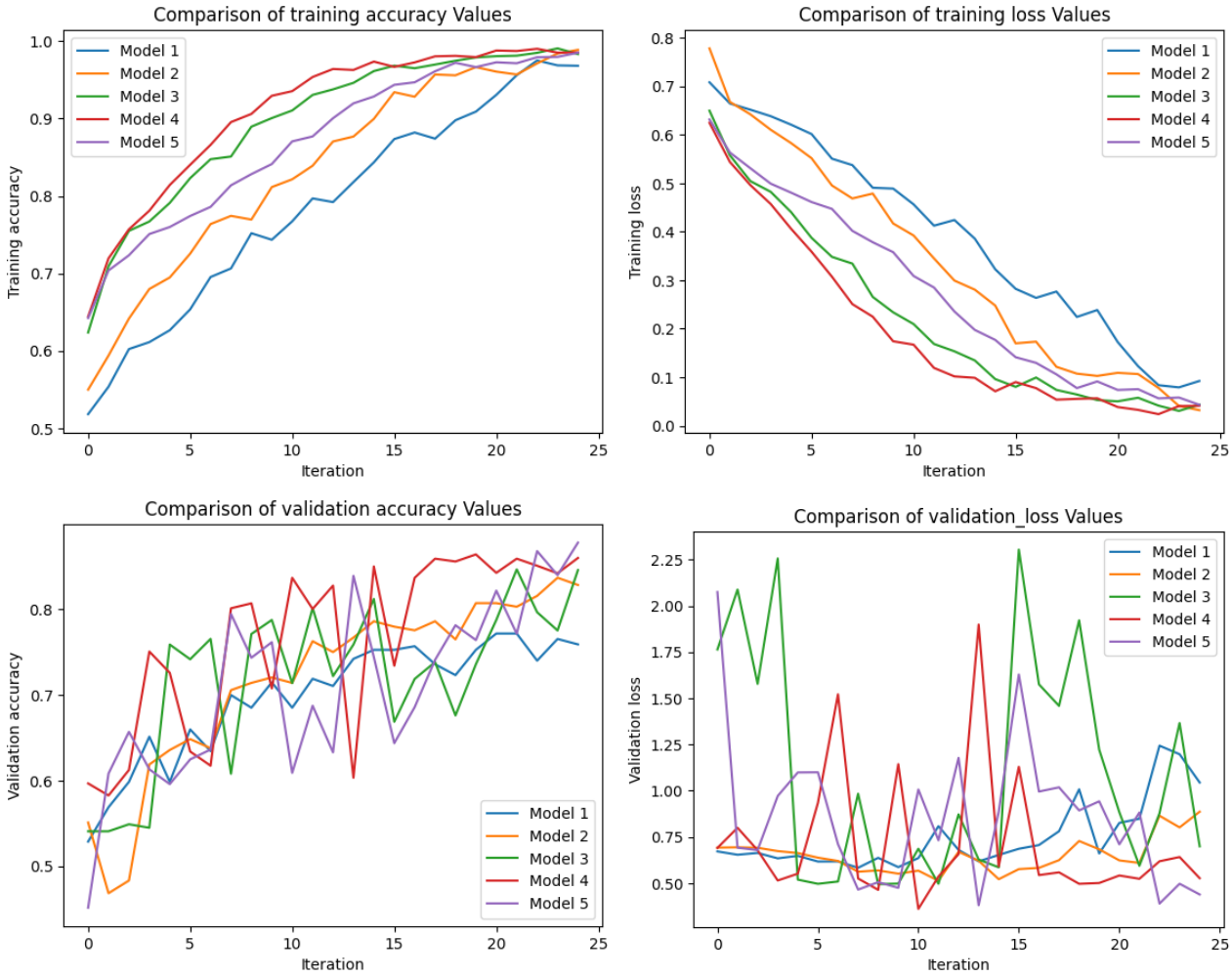


Fig. 2 Analyses of the loss and accuracy of the experimented models

Table 4 compares the depicted models in Table 3 based on accuracy, precision, sensitivity, specificity, and F1-score statistics. This table outlines the evaluation metrics calculated from test data applied to the trained models. This table includes not only evaluations of models with augmented datasets but also evaluations of models using only the original dataset. The fifth model is the most accurate using the original dataset without augmentation, as shown in Table 4. There is no doubt that the obtained results from all the models are notably poorer than those obtained from augmented datasets, leading to higher accuracy. Accordingly, this illustrates the effect of augmentations introduced in subsection 3.1.3.

Performance of the fourth model with six layers differs significantly with the other competitive models. With data augmentation, the fourth model's F1-score, precision, specificity, and accuracy are significantly better, suggesting consistency in the training results. Model performance, in this case, was 90.04% accuracy, 0.8909 precision, 0.8827 sensitivity, 0.9144 specificity, and 0.8868 F1-score. Furthermore, as shown in Table 4, this is advantageous for real-time implementations since the time taken to classify an image into COVID-19 class or Healthy class, was approximately 0.323 seconds. In addition, despite the absence of GPUs and advanced system infrastructure configurations, the model proposed in this study achieved satisfactory train and extraction time. Additionally, medical diagnosis systems do not require hospitals and clinics to invest in expensive equipment. The experiments evaluated do not reveal statistically significant differences between architectures 4 and 5. As shown in Table 4, the top three architectures obtained a minimum accuracy of 80% and a minimum precision of 77.57%. However, architecture 4, with six convolutional layers, has a maximum accuracy, worth highlighting.

Section 3.1.3 outlines the application of several augmentation techniques, including spatial transformations and intensity adjustments, to improve dataset diversity and support model robustness. To further understand the impact of these techniques on model performance, we focused on these methods. The experimental results demonstrated that the application of these augmentation strategies led to notable performance improvements in the CNN models. Specifically, we observed an accuracy improvement of approximately 1-3% due to these techniques, emphasizing the model's enhanced adaptability to patient positioning.

4.3 The effectiveness of the pre-trained models on COVID-CT using transfer learning techniques

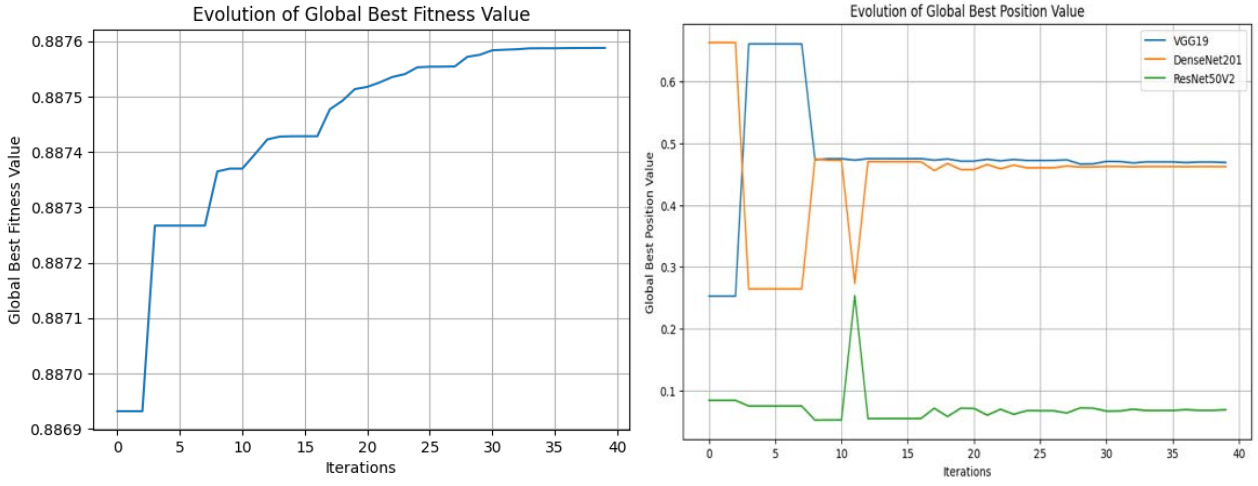
Using the COVID-CT dataset, we evaluate the effectiveness of our proposed architecture with several pre-trained networks, including the Xception, MobileNetV2, Resnet50V2, DenseNet201, and the like networks with fine-tuning. Table 5 shows these potential candidate architectures that used the COVID-CT dataset for training. We evaluate these pre-trained models quantitatively based on their precision, sensitivity, accuracy, F1 score, and specificity.

As seen in Table 5, VGG19 had 86.1 percent accuracy, which was higher than the other models. DenseNet201, Resnet50V2, and VGG16 on this dataset scored 84.6%, 82.5%, and 81.8% accuracy, respectively. All other pre-trained architectures, including Xception, InceptionV3, and MobileNet, failed to reach 80% accuracy. Compared to InceptionV3, ResNet50, and DenseNet121 models, Xception, Resnet101, Resnet152, and DenseNet169 provide better accuracy. The accuracy of InceptionV3, ResNet50, and DenseNet121 models was 63.5%, 67.2%, and 65.4%, respectively. Xception and DenseNet169 showed moderate accuracy of 78.9% and 78.7%, respectively. Regarding accuracy, the ResNet50V2 model outperforms all the other ResNet models. Our study found that the ResNet50 model performed the least effectively in detecting COVID-19. Given these results, the VGG19 pre-trained model performed better in the COVID-19 detection task for the considered dataset, with lower misclassification rates.

According to the transfer learning-based approaches, MobileNet provides a sensitivity of 0.831; The sensitivity of MobileNetV2 and VGG16 is lower. With a detection sensitivity of 0.903, ResNet50V2 provides reasonable sensitivity. The greater the sensitivity, the lower the chance of misdiagnosing a COVID-19 class image. Based on Table 5, ResNet50V2 and VGG19 effectively detect COVID-19 classes. Sensitivity is crucial in medical image diagnosis, so a model based on InceptionV3, Xception, DenseNet121, and ResNet50 cannot be used here.

TABLE 5 An analysis of improved CNN models for the classification of COVID-19

Architectures	Versions	Test time	Accuracy	Precision	Sensitivity	Specificity	F1_score
Xception		0.3675	0.789473	0.836734	0.672131	0.888889	0.745454
DenseNet	DenseNet121	0.2932	0.654391	0.985915	0.223642	0.997455	0.364583
	DenseNet169	0.2819	0.787535	0.865470	0.616613	0.923664	0.720149
	DenseNet201	0.2843	**0.845865**	0.824	0.844262	0.847222	0.834008
MobileNet	MobileNet	0.4327	0.744	0.676259	0.831858	0.671532	0.746031
	MobileNetV2	0.4184	0.762646	0.759615	0.686956	0.8239436	0.721461
ResNet	ResNet50	0.3386	0.672473	0.7111	0.484848	0.832258	0.576576
	ResNet50V2	0.3218	**0.825174**	0.746667	0.903226	0.765432	0.817518
	ResNet101	0.3261	0.714285	0.771739	0.537878	0.864516	0.633928
	ResNet152	0.3572	0.791086	0.779874	0.756097	0.820513	0.767801
VGG	VGG16	0.3102	0.818182	0.8214285	0.741935	0.876543	0.779661
	VGG19	0.3187	**0.860902**	0.819548	0.893443	0.833333	0.854901
InceptionV3		0.371	0.635338	0.820513	0.262295	0.951389	0.397515



(a) (b)

Fig. 3 PSO algorithm evaluation. (a) The evolutions of c_1, c_2, and c_3, (b) The evolutions of fitness in PSO

Table 6 compares the highest-performing model's performance depicted in previously introduced models (from Tables 4 and 5), the PSO-based configuration optimization model, and existing models on the COVID-CT dataset. PSO-based configuration optimization model combines the VGG19, DenseNet201, and ResNet50V2 architectures. The best values

for *c1*, *c2*, and *c3* are respectively 0.47, 0.46, and 0.07 (Fig. 3); this implies that 0.47, 0.46, and 0.07 predictions are made of the VGG19, DenseNet201, and ResNet50V2 models, respectively. The proposed model maximizes accuracy without compromising other criteria by minimizing false negatives. Compared to existing models, the proposed model has significantly higher sensitivity as a result of the reduction in false negatives. Moreover, in comparison with existing models, the introduced model performed better F1 score a metric that harmonizes precision and sensitivity. Accordingly, the proposed model has a high degree of accuracy compared to existing models.

Our proposed model has an accuracy of 90.32%, which is lower than the 93.33% reported by Shaik and Cherukuri (2021) [2]. However, this accuracy is superior to those found in studies by Nettur et al. (2025) [13], Ali et al. (2024) [14], LightEfficientNetV2 [15], and Kathamuthu et al. [16]. The effectiveness of our model is compelling, as it exhibits a significantly higher sensitivity and precision compared to existing models, indicating its potential in minimizing false negatives.

As reported in related studies, COVID-CT classification accuracy ranged from 88% to 93.33%, with the highest accuracy, 93.33 achieved by Shaik, and Cherukuri [2] in 2022. Notably, our PSO-based configuration optimization model ranks highly among recent research efforts, as evident from the comparative metrics in Table 6. The model delivers outstanding performance with the highest precision and F1 score of 91.34%, indicating its reliability and effectiveness in COVID-19 diagnosis.

The PSO-based configuration optimization further exhibits superior sensitivity and specificity compared to traditional CNN approaches, offering a prominent advancement in AI-based diagnostic tools for COVID-19. Overall, the proposed model stands out for its robust performance in accuracy, precision, and F1 scoring, positioning it as an effective solution for early COVID-19 detection using chest CT images.

The incorporation of the SARS-CoV-2 CT-Scan dataset allowed us to further validate the robustness of our PSO-based CNN model in a broader context. Our model maintained high performance metrics, reinforcing its applicability in various clinical settings. The diversity within the SARS-CoV-2 CT-Scan dataset highlighted the model's generalization capabilities, which is essential for reliable automated COVID-19 diagnostics.

TABLE 6 On COVID-CT dataset, the proposed model was compared to previous models

Architectures	Accuracy	Precision	Sensitivity	Specificity	F1_score
CNN of Model 4	90.04	89.1	88.27	91.44	88.68
Pre-trained model VGG19	86.09	81.95	89.34	83.33	85.49
PSO-based configuration optimization	90.32	92.13	90.56	92.64	91.34
Shaik and Cherukuri (2021) [2]	93.33	**93.17**	**93.54**	-	**93.29**
Nettur et al. (2025) [13]	91.04	91.28	91	89.28	91.23
Ali et al. (2024) [14]	88.65	**88.14**	**88.96**	88.3	**88.41**
LightEfficientNetV2 [15]	88.67	87.28	87.43	-	87.55
Kathamuthu et al. [16]	90.16	91.11	90.38	91.26	91.13

TABLE 7 Evaluation of PSO-based configuration optimization model on COVID-CT and SARS-CoV-2 CT Datasets

Datasets	Accuracy	Precision	Sensitivity	Specificity	F1_score
COVID-CT	90.32	92.13	90.56	92.64	91.34
SARS-CoV-2 CT	92.47	**92.3**	**93.52**	92.67	**92.11**
Trained on COVID-CT tested on both datasets	84.5	**86.21**	**85.65**	86.45	**84.23**
Trained on SARS-CoV-2 CT tested on both datasets	88.24	**89.76**	**87.04**	88.46	**87.63**
Both datasets	92. 88	**93.12**	**93.37**	93.7	**91.86**

The incorporation of the SARS-CoV-2 CT-Scan dataset allowed us to further validate the robustness of our PSO-based CNN model in a broader context. Our model maintained high performance metrics, reinforcing its applicability in various clinical settings. The diversity within the SARS-CoV-2 CT-Scan dataset highlighted the model's generalization capabilities, which is essential for reliable automated COVID-19 diagnostics. Table 7 presents the performance metrics of the prpposed architecture when applied to two different datasets: the COVID-CT dataset and the SARS-CoV-2 CT-Scan dataset. The included metrics collectively offer a comprehensive view of the model's performance. Regarding the diversity within the SARS-CoV-2 CT-Scan dataset, analyzing the performance across these datasets allows for evaluation of the model's robustness and generalization ability, particularly against potential variations in image quality and data sources.

The findings presented in Table 7 highlight the commendable performance of the proposed model when evaluated on the COVID-CT and SARS-CoV-2 CT datasets. In the comparative analysis, the model attained an accuracy of 90.32% on the COVID-CT dataset, which improved to 92.47% on the SARS-CoV-2 CT dataset. These findings underscore the model's efficacy in the accurate classification of CT images and its enhanced capacity for generalization across varied datasets. The superior accuracy observed with the SARS-CoV-2 CT dataset may be attributed to several factors, including disparities in image quality, variations in pathological representations, the distinctive characteristics of the training

data employed, and the considerably larger number of images in this dataset compared to the COVID-CT dataset.

Precision metrics reflect strong performance with values of 92.13% for the COVID-CT dataset and 92.3% for the SARS-CoV-2 CT dataset. These elevated precision rates are indicative of the model's reliability in generating positive predictions, thereby minimizing the incidence of false positives in clinical contexts. Additionally, sensitivity is a critical metric in medical diagnostics, as it examines the model's ability to accurately identify true positive cases; the model achieved a sensitivity of 90.56% on the COVID-CT dataset and a noteworthy 93.52% on the SARS-CoV-2 CT dataset. These substantial sensitivity values reflect the model's robust capacity to assist clinicians in detecting COVID-19 infections, facilitating timely medical interventions.

Conversely, specificity values, which signify the model's accuracy in identifying negative cases, were found to be 92.64% for the COVID-CT dataset and 92.67% for the SARS-CoV-2 CT dataset. These outcomes underscore the model's robustness in minimizing false negatives. The overall F1 scores were recorded at 91.34% for the COVID-CT dataset and 92.11% for the SARS-CoV-2 CT dataset, highlighting the model's clinical efficacy. When analyzing both datasets concurrently, the overall F1 score decreased to 87.63%, alongside an accuracy of 88.24%, suggesting that integrating the datasets may introduce challenges that compromise consistency, likely due to variations in image quality and other dataset characteristics. Addressing these issues is imperative for enhancing the model's robustness in real-world clinical applications. The efficacy of the proposed architecture suggests its potential as a diagnostic adjunct within clinical settings, augmenting radiologists' capabilities in effectively identifying and diagnosing COVID-19, thereby improving patient outcomes during an ongoing pandemic necessitating rapid and reliable diagnostic methods.

4.4 Statistical Analysis

To assess the comparative efficacy of the applied various models, a one-way Analysis of Variance (ANOVA) was conducted. It helps to understand if the variance within models is less than the variance between them [31]. This method is widely used in various fields, including computing, psychology, agriculture, and healthcare, when comparing multiple groups.

The ANOVA process initiates with the computation of means for each model, in addition to an overall mean for all observations. This analysis entails the estimation of between-group variance, representing the variability of group means in relation to the overall mean, and within-group variance, which delineates the distribution of individual data points relative to their respective model means. The results provided an F-statistic that indicates the ratio of variance between groups to that within groups, while the corresponding p-value assessed the statistical significance of these differences. Specifically, a p-value less than 0.05 would lead us to reject the null hypothesis, thus suggesting that at least one model's performance is significantly different. Our findings, with a reported F-statistic of 11.84 and a p-value of 0.006, confirm the presence of significant performance disparities among the models.

Given these significant differences, we proceeded with post hoc tests, specifically Tukey’s HSD test [32], to further identify which models differ from one another. The Tukey test allows us to make multiple pairwise comparisons while controlling for the overall Type I error rate, set at 0.05. This was crucial for validating the performance evaluations and ensuring that any interpretations made about model superiority or inferiority were statistically robust.

The results from Tukey's HSD indicated logical and statistically significant differences in performance metrics across models. The results indicated that the PSO-based configuration optimization model achieved a significantly higher accuracy (90.32%) compared to the pre-trained model VGG19 (86.09%) and the CNN of Model 4 (90.04%). However, there was no significant difference in accuracy between our model and the model proposed by Shaik and Cherukuri (93.33%). These findings suggest that the proposed model is a promising approach for detecting COVID-19 from chest CT scans, as it outperforms several other models in terms of accuracy. This rigorous statistical evaluation of model performance enhances the credibility of our findings and provides deeper insights into optimizing methodologies for COVID-19 detection, thereby guiding future research and clinical applications.

The rigorous statistical evaluation of the models' performance using ANOVA and Tukey's HSD test enhances the robustness of our findings and provides deeper insights into optimizing COVID-19 detection methodologies. By applying these statistical methods, we ensure that the assessments of model efficacy are not only based on raw performance metrics but are also statistically validated, which is essential for drawing reliable conclusions in the context of disease detection and healthcare improvement.

4.5 Practical Challenges of Model Implementation

While our proposed PSO-based CNN model demonstrates notable performance in detecting COVID-19 from chest CT scans, several practical challenges must be addressed for effective implementation in real-world medical settings:

Computational Cost: Implementing deep learning models, particularly for large datasets and complex architectures, often requires significant computational resources. This includes the need for high-performance GPUs and adequate memory for processing and storing images. Adoption in clinical settings may necessitate a cost-benefit analysis to justify infrastructure investments. Therefore, optimization techniques, such as model pruning and quantization, may be explored to reduce computational demands and improve efficiency without sacrificing accuracy.

Model Interpretability: In medical diagnostics, understanding the rationale behind model predictions is crucial for clinician trust and decision-making. Techniques such as Grad-CAM can be employed to visualize which parts of the CT images contribute most to a model's predictions. This transparency can help healthcare professionals validate AI decisions and integrate them into their diagnostic processes.

Handling Non-Standard Image Formats: Variability in imaging techniques, CT machine specifications, and patient conditions can affect the quality and format of images. Our model should be robust against these variations, which may require extensive preprocessing steps, including standardization of image dimensions and normalization techniques. Data augmentation strategies can also enhance the model's generalization capabilities, enabling it to perform reliably across diverse datasets.

5 Conclusion

The COVID-19 outbreak was caused by an explosion of pneumonia triggered by the COVID-19 virus in China (December 2019). Deep learning-based medical diagnosis systems are increasingly helpful for such clinical diagnosis. Taking into account the limited public data available, we focused on two-class classification in this study. Diagnosing COVID-19 using CT images, this study comprehensively analyzes different deep learning-based approaches utilizing particular parameter specifications and training strategies. A comprehensive experiment was conducted on the dataset named COVID-CT using the most advanced CNN architectures and their variants. Moreover, we developed a PSO-based optimization model for configuring CNNs with the highest performance by evaluating different pre-trained CNNs. This study aims to evaluate the effectiveness of our proposed techniques in diagnosing COVID-19 applied to two of the largest publicly available lung CT scan datasets, namely the COVID-CT and SARS-CoV-2 datasets. In addition to contributing to the development of more accurate point-of-care diagnostic tools for COVID-19, an efficient and fast diagnostic tool can be developed using the findings of this study. The proposed model achieved cutting-edge performance on the aforementioned dataset with an average accuracy of 90.32%, precision of 92.13%, and sensitivity of 90.56%. We further contextualized our findings by integrating recent studies, which showcase the evolving methodologies in the field, reinforcing the significance and relevance of our work in enhancing AI-driven diagnostic solutions for COVID-19.

Disclosure of Potential Conflicts of Interest

1- Ethical and Informed Consent for Data Used: Not applicable
2- Data Availability and Access: The results will be accessible upon request.
3- Competing Interests: There are no competing interests.
4- Authors contribution statement: All authors contribute in all parts.

Reference

[1] Z. Ullah, M. Usman and J. Gwak, "MTSS-AAE: Multi-task semi-supervised adversarial autoencoding for COVID-19 detection based on chest X-ray images," *Expert Systems with Applications*, p. 119475, 2023.

[2] N. S. Shaik and T. K. Cherukuri, "Transfer learning based novel ensemble classifier for COVID-19 detection from chest CT-scans," *Computers in Biology and Medicine 141*, p. 105127, 2022.

[3] N. N. Das, N. Kumar, M. Kaur, V. Kumar and D. Singh, "Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays," *Irbm 43*, vol. no. 2, pp. 114-119, 2022.

[4] C. Ouchicha, O. Ammor and M. Meknassi, "CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images," *Chaos, Solitons & Fractals*, no. 140, p. 110245, 2020.

[5] F. Ali, F. Ullah, J. I. Khan, J. Khan, A. W. Sardar and S. Lee, "COVID-19 spread control policies based early dynamics forecasting using deep learning algorithm," *Chaos, Solitons & Fractals*, no. 167, p. 112984, 2023.

[6] M. Rahimzadeh, A. Attar and S. M. Sakhaei, "A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset," *Biomedical Signal Processing and Control 68*, p. 102588, 2021.

[7] K. Gupta and V. Bajaj, "Deep learning models-based CT-scan image classification for automated screening of COVID-19," *Biomedical Signal Processing and Control*, vol. 80, p. 104268, 2023.

[8] M. F. Aslan, M. F. Unlersen, K. Sabanci and A. Durdu, "CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection," *Applied Soft Computing98*, p. 106912, 2021.

[9] A. Altan and S. Karasu, "Recognition of COVID-19 disease from X-ray images by hybrid model consisting of 2D curvelet transform, chaotic salp swarm algorithm and deep learning technique," *Chaos, solitons & fractals*, no. 140, p. 110071, 2020.

[10] E. F. Ohata, G. M. Bezerra, J. V. S. d. Chagas, A. V. L. Neto, A. B. Albuquerque, V. H. C. D. Albuquerque and P. P. R. Filho, "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning," *IEEE/CAA Journal of Automatica Sinica 8*, vol. no. 1, pp. 239-248, 2020.

[11] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez and V. Singh, "Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet.," *Chaos, Solitons & Fractals*, no. 138, p. 109944, 2020.

[12] F. E. F. Junior and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Swarm and Evolutionary Computation*, vol. 49, pp. 62-74, 2019.

[13] S. B. Nettur, S. Karpurapu, U. Nettur, L. S. Gajja, S. Myneni, A. Dusi and L. Posham, "A Hybrid Deep Learning CNN Model for Enhanced COVID-19 Detection from Computed Tomography (CT) Scan Images."," *arXiv preprint arXiv:2501.17160*, 2025.

[14] A. M. Ali, K. Ghafoor, A. Mulahuwaish and H. Maghdid, "COVID-19 pneumonia level detection using deep learning algorithm and transfer learning," *Evolutionary Intelligence*, vol. 2, no. 17, pp. 1035-1046, 2024.

[15] M.-L. Huang and Y.-C. Liao, "A lightweight CNN-based network on COVID-19 detection using X-ray and CT images," *Computers in Biology and Medicine*, vol. 146, p. 105604, 2022.

[16] N. D. Kathamuthu, S. Subramaniam, Q. H. Le, S. Muthusamy, H. Panchal, S. C. M. Sundararajan, A. J. Alrubaie and M. M. A. Zahra, "A deep transfer learning-based convolution neural network model for COVID-19 detection using computed tomography scan images for medical applications," *Advances in Engineering Software*, no. 175, p. 103317, 2023.

[17] S. Hassantabar, M. Ahmadi and A. Sharifi, "Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches," *Chaos, Solitons & Fractals*, vol. 140, p. 110170, 2020.

[18] M. K. Mahbub, M. Biswas, L. Gaur, F. Alenezi and K. C. Santosh, "Deep features to detect pulmonary abnormalities in chest X-rays due to infectious diseaseX: Covid-19, pneumonia, and tuberculosis," *Information Sciences*, vol. 592, pp. 389-4, 2022.

[19] M. Goncharov, M. Pisov, A. Shevtsov, B. Shirokikh, A. Kurmukov, I. Blokhin and V. C. e. al, "CT-Based COVID-19 triage: Deep multitask learning improves joint identification and severity quantification," *Medical image analysis*, vol. 71, p. 102054, 2021.

[20] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna and M. Z. Parvez, "CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images," *Chaos, Solitons & Fractals*, no. 142, p. 110495, 2021.

[21] A. K. Das, S. Kalam, C. Kumar and D. Sinha, "TLCoV-An automated Covid-19 screening model using Transfer Learning from chest X-ray images.," *Chaos, Solitons & Fractals*, no. 144, p. 110713, 2021.

[22] A. Saygılı, "A new approach for computer-aided detection of coronavirus (COVID-19) from CT and X-ray images using machine learning methods," *Applied Soft Computing*, vol. 105, p. 107323, 2021.

[23] A. Zeroual, F. Harrou, A. Dairi and Y. Sun, "Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study," *Chaos, solitons & fractals*, no. 140, p. 110121, 2020.

[24] J. Zhao, Y. Zhang, X. He and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865 490*, 2020.

[25] E. Soares, P. Angelov, S. Biaso, M. H. Froes and D. K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification," *MedRxiv*, 2020.

[26] A. Kebaili, J. Lapuyade-Lahorgue and S. Ruan, "Deep Learning Approaches for Data Augmentation in Medical Imaging: A Review," *ournal of Imaging*, vol. no. 4, no. 9, p. 81, 2023.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung and a. X. C. Alec Radford, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[28] M. Canayaz, "C+ EffxNet: A novel hybrid approach for COVID-19 diagnosis on CT images based on CBAM and EfficientNet," *Chaos, Solitons & Fractals*, no. 151, p. 111310, 2021.

[29] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: a survey," *Applied Sciences*, vol. no. 18, no. 12, p. 8972, 2022.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pmlr, 2015.

[31] A. Chatzi and O. Doody, "The one-way ANOVA test explained," *Nurse researcher* , vol. 32, no. 4, 2024.

[32] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1-30, 2006.

[33] W. M. Shaban, A. H. Rabie, A. I. Saleh and M. A. Abo-Elsoud, "A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier," *Knowledge-Based Systems*, vol. 205, p. 106270, 2020.

[34] A. K. Mishra, S. K. Das, P. Roy and S. Bandyopadhyay, "Identifying COVID19 from chest CT images: a deep convolutional neural networks based approach," *Journal of Healthcare Engineering*, 2020.

[35] X. X. H. J. Z. Y. Z. Yang, S. Zhang and P. Xie., "COVID-CT-Dataset: a CT scan dataset about COVID-19," *arXiv preprint arXiv:2003.13865*, 2020.

[36] P. Afshar, S. Heidarian, N. Enshaei, F. Naderkhani, M. J. Rafiee, A. Oikonomou, F. B. Fard, K. Samimi, K. N. Plataniotis and A. Mohammadi, "COVID-CT-MD, COVID-19 computed tomography scan dataset applicable in machine learning and deep learning," *Scientific Data 8, no. 1*, p. 121, 2021.

[37] A. Sadeghi, M. Sadeghi, A. Sharifpour, M. Fakhar, Z. Zakariaei, M. Sadeghi, M. Rokni, A. Zakariaei, E. S. Banimostafavi and F. Hajati, "Potential diagnostic application of a novel deep learning-based approach for COVID-19," *Scientific Reports 14, no. 1*, p. 280, 2024.

[38] R. Rajpoot, M. Gour, S. Jain and V. B. Semwal, "Integrated ensemble CNN and explainable AI for COVID-19 diagnosis from CT scan and X-ray images," *Scientific Reports 14, no. 1*, p. 24985, 2024.

[39] I. M. Pires, F. Hussain, N. M. Garcia, P. Lameski and E. Zdravevski, "Homogeneous data normalization and deep learning: A case study in human activity classification," *Future Internet 12, no. 11*, p. 194, 2020.

[40] H. Alhichri, "CNN Ensemble Approach to Detect COVID-19 from Computed Tomography Chest Images," *Computers, Materials & Continua 67, no. 3*, pp. 115-123, 2021.

[41] K. R. Bhatele, A. Jha, D. Tiwari, M. Bhatele, S. Sharma, M. R. Mithora and S. Singhal, "Covid-19 detection: A systematic review of machine and deep learning-based approaches utilizing chest x-rays and ct scans," *Cognitive Computation 16, no. 4*, pp. 1889-1926, 2024.

[42] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, Berlin, Heidelberg, 2000.

[43] V. K. Verma, K. Saxena and U. Banodha, "Analysis Effect of K Values Used in K Fold Cross Validation for Enhancing Performance of Machine Learning Model with Decision Tree," in *In International Advanced Computing Conference*, Switzerland, 2023.