



Conditional cascaded network (CCN) approach for diagnosis of COVID-19 in chest X-ray and CT images using transfer learning

Amr E. Eldin Rashed^{a,*}, Waleed M. Bahgat^{b,c}

^a Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif, Saudi Arabia

^b Information Technology Department, Faculty of Computer and Information Sciences, Mansoura University, 35516, Egypt

^c Department of Computer Science and Informatics, Taibah University Al Medina Al Munawara, Medina, Saudi Arabia



ARTICLE INFO

Keywords:

Automated diagnosis
COVID-19
chest X-Ray
CT images
Transfer learning
Conditional cascaded networks
CNN architectures
Grad-CAM
Occlusion specificity

ABSTRACT

The COVID-19 pandemic has caused substantial global health and economic damage, with over five million confirmed cases worldwide. The importance of the rapid, accurate diagnosis of infected patients has been underscored. However, due to the shortage of testing kits and the time-consuming, troublesome nature of the manual RT-PCR test, an automated, efficient diagnosis system using chest medical images for the early screening of COVID-19 is crucial. In this paper, an automated approach for the rapid, accurate diagnosis of COVID-19 in chest X-ray and computed tomography (CT) images using transfer learning is presented. The proposed technique leverages the conditional cascaded network approach, which employs multiple levels of networks to analyze images with high confidence. Transfer learning is employed with seven commonly used existing convolutional neural network architectures and four datasets for X-ray and CT images. Various regularization, optimization, dropout, and data augmentation techniques are examined through a series of experiments with three optimizers. Our technique is compared with other state-of-the-art techniques, and the achieved results demonstrate highly promising performance metrics. Additionally, occlusion specificity and gradient-weighted class activation mapping techniques are employed to understand the network output better. The proposed technique is highly adaptable and scalable and does not require manual hyperparameter tuning.

1. Introduction

COVID-19 is a viral respiratory illness caused by the SARS-CoV-2 virus. The outbreak of COVID-19, which was first reported in Wuhan, China in December 2019, quickly became a global pandemic with far-reaching effects on public health and society as a whole. As of March 2023, over 460 million cases and over 6 million deaths worldwide have been confirmed [1]. The COVID-19 pandemic has had a considerable effect on public health, with healthcare systems across the globe overwhelmed by the surge in cases. The pandemic has also had profound social, economic, and psychological effects, with many countries implementing lockdowns, travel restrictions, and other measures to contain the spread of the virus [2].

Medical imaging has become a critical component in the diagnosis and management of COVID-19. Medical imaging, including chest X-ray (CXR) and computed tomography (CT), can help identify the presence and severity of lung involvement in patients with COVID-19, which is a critical factor in determining the appropriate treatment plan [3].

Additionally, medical imaging can assist in differentiating COVID-19 from other respiratory infections and diseases, such as influenza, pneumonia, and acute respiratory distress syndrome [4]. Compared with X-ray imaging, CT imaging provides higher resolution, greater accuracy, and better visualization of soft tissues, making it a preferred imaging modality in many clinical scenarios. However, CT imaging has some disadvantages, including higher radiation exposure and increased cost compared with X-ray imaging. Therefore, the choice between CT and X-ray imaging depends on the patient's clinical condition, the purpose of the imaging study, and the risks and benefits of each technique.

Current diagnostic methods for COVID-19 include RT-PCR testing, antigen testing, and serological testing. While these methods are useful, certain limitations can affect their accuracy and reliability. For example, RT-PCR testing has a high false-negative rate due to the variability in viral load and sampling errors [5]. Antigen testing has a higher false-negative rate than RT-PCR testing and is more likely to miss asymptomatic cases [6].

* Corresponding author.

E-mail address: a.rashed@tu.edu.sa (A.E. Eldin Rashed).

To address the limitations of the current diagnostic methods, automated medical diagnosis models for COVID-19 detection are needed. These models use machine learning algorithms and computer vision techniques to analyze medical imaging data, such as CXRs and CT scans, to detect the presence of COVID-19 with high accuracy [7]. Automated medical diagnosis models can provide a rapid, accurate diagnosis, which is essential for controlling the spread of the disease and ensuring timely treatment. Furthermore, automated diagnosis models can aid in overcoming the shortage of trained medical professionals needed to read and interpret medical images, particularly in areas with limited healthcare resources. Automated diagnosis models can provide an objective, standardized diagnosis that is not subject to interobserver variability, thus reducing the potential for diagnostic errors and improving the patient outcomes [8].

Deep learning (DL) is a subfield of machine learning that uses artificial neural networks to model and solve complex problems. DL has been widely used in medical image analysis, including the detection and diagnosis of COVID-19. One of the techniques used in DL is transfer learning, which involves using pretrained models on large datasets to fine-tune them for specific tasks.

Transfer learning is widely utilized in the development of automated medical diagnosis models for COVID-19 detection in CXRs and CT scans. By leveraging pretrained convolutional neural networks (CNNs) to extract features from medical images, transfer learning models can effectively learn from limited labeled data and substantially reduce the need for enormous amounts of training data. This approach has shown promising results in COVID-19 diagnosis with high sensitivity and specificity, making it a useful tool for medical professionals. Furthermore, transfer learning models can improve the performance of COVID-19 diagnosis models in low-resource settings, where obtaining sufficient training data can be challenging. Therefore, transfer learning is considered a powerful, effective technique for developing automated medical models for COVID-19 diagnosis [9].

The research problem addressed in this paper is the need for the accurate, efficient diagnosis of COVID-19 using medical imaging techniques. Transfer learning techniques are adopted to develop an automated medical diagnosis model for COVID-19 detection in CXRs and CT scans. The performance of the transfer learning models is compared with other state-of-the-art models. In addition, the influence of transfer learning models on improving the accuracy and efficiency of COVID-19 diagnosis using medical imaging is evaluated.

The contributions of this paper can be summarized as follows:

- **Introducing Conditional Cascaded Networks:** A novel three-stage methodology, conditional cascaded network (CCN), specifically designed for the automated diagnosis of COVID-19 using X-ray and CT images is presented. Our approach incorporates cascaded and conditional networks, setting it apart from traditional ensemble techniques. Each network in the cascade is trained on its own dataset with distinct output labels, a distinguishing feature that contributes to enhanced results compared with using a single network with numerous labels. By training the CCN on four datasets, our method achieves improved accuracy and robustness in COVID-19 detection, making it a highly promising, innovative approach for medical image analysis.
- **Leveraging Transfer Learning Technique:** To detect COVID-19 effectively in patients, transfer learning is employed with seven commonly used CNN architectures, namely, GoogleNet [10], SqueezeNet [11], ResNet-18, ResNet-50 [12], AlexNet [13], DarkNet [14], and ShuffleNet [15], along with four diverse datasets for X-ray and CT images. The advantage of our technique lies in its adaptability and scalability because it eliminates the need for the manual assignment of hyperparameter values for the CNN architectures.
- **Enhancing Reliability:** Our approach implements a robust five-fold cross-validation strategy for training and testing the automated COVID-19 diagnosis model. This approach ensures comprehensive

data utilization for testing, resulting in a more reliable performance estimation and reduced variance. Additionally, the five-fold cross-validation facilitates effective parameter tuning, enhancing the model's accuracy and stability compared with the conventional 80/20 data splitting technique.

- **Exploring Various Techniques:** Regularization, optimization, dropout, and data augmentation techniques are thoroughly explored through a series of experiments incorporating three optimizers: adaptive moment estimation (Adam), stochastic gradient descent with momentum (SGDM), and root mean square propagation (RMSprop).
- **Comparative Analysis:** The proposed technique is rigorously compared with other state-of-the-art approaches, and the achieved results demonstrate highly promising performance metrics, affirming the superiority of CCN in COVID-19 diagnosis.
- **Insightful Interpretation:** Occlusion specificity and gradient-weighted class activation mapping (Grad-CAM) techniques are employed to gain a better understanding of the network's output, providing valuable insights into the decision making of the model along with four diverse datasets for X-ray and CT images. The advantage of our technique lies in its adaptability and scalability because it eliminates the need for the manual assignment of hyperparameter values for the CNN architectures.

The rest of the paper is organized as follows: Section 2 covers the related work. Section 3 delves into the materials and methods used. Section 4 outlines the methodology employed. Section 5 presents the results achieved through the proposed technique. Section 6 discusses the heat map visualization and compares the proposed technique with state-of-the-art approaches in Section 7. Section 8 presents the overall conclusions. Section 9 discusses the limitations of this work and provides suggestions for future research.

2. Related work

Automated medical diagnosis models using DL in medical imaging have emerged as a promising tool for COVID-19 detection. These models have been applied to X-ray and CT images. Fig. 1 shows the X-ray and CT images of COVID-19, pneumonia, and normal lungs [16]. This discussion aims to summarize and critically evaluate the existing studies on automated medical models for COVID-19 diagnosis using transfer learning in X-ray and CT images.

2.1. COVID-19 diagnosis using X-ray

Several studies have investigated the use of automated medical diagnosis models for COVID-19 detection using X-ray images.

For instance, Liu et al. [17] suggested utilizing federated learning to train COVID-19 data and conducted experiments to evaluate its efficiency in diagnosing COVID-19 through X-ray images. They further compared the performances of four popular models, namely, Mobile Net, ResNet18, Mobile Net, and COVID Net, within and without the federated learning framework, to encourage further research in federated learning for COVID-19. The experimental findings revealed ResNet18 performed the best in training with and without federated learning. Additionally, ResNet exhibited the highest performance in images with COVID-19 labels, achieving a maximum accuracy of 91.26%.

Karim et al. [18] presented an explainable deep neural network (DNN)-based method named "Deep COVID Explainer" for the automated detection of COVID-19 symptoms from CXR images. They utilized a dataset of 16,995 CXR images from 13,808 patients, covering normal, pneumonia, and COVID-19 cases. They subjected the CXR images to comprehensive preprocessing, augmentation, and classification with a neural ensemble method. Then, they employed gradient-guided class activation maps (Grad-CAM++) and layer-wise relevance propagation

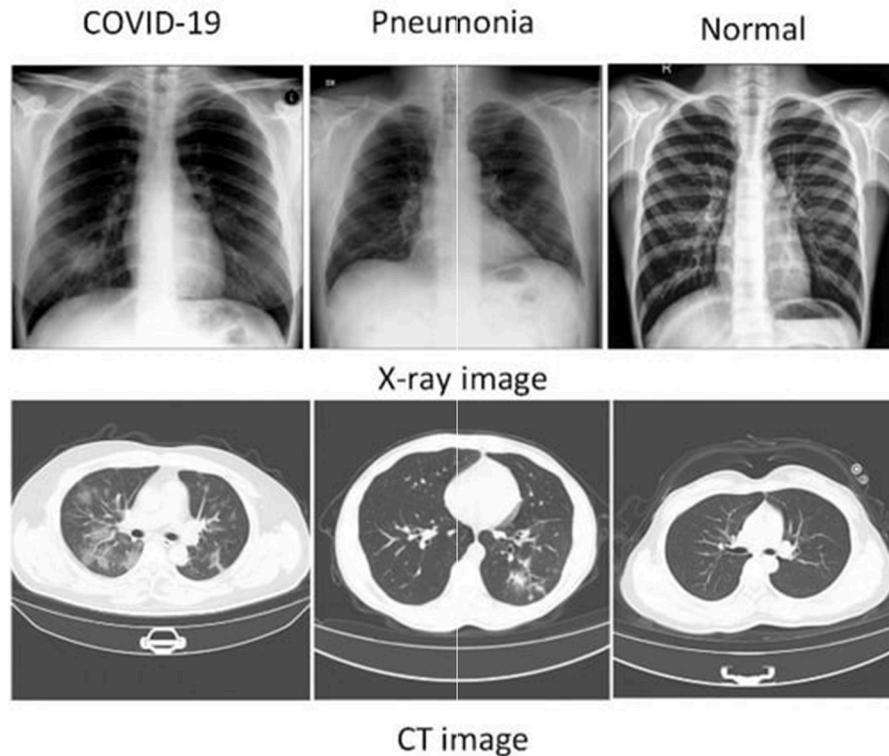


Fig. 1. X-ray and CT Images of COVID-19.

to highlight class-discriminating regions, providing human-interpretable explanations of the predictions. The evaluation results based on hold-out data demonstrated their approach confidently identifies COVID-19, with a positive predictive value (PPV) of 89.61% and a recall of 83%, outperforming recent comparable methods.

Farooq et al. [19] used advanced training techniques to train residual neural networks for COVID-19 detection. They employed a three-step technique called COVID ResNet that involved fine-tuning a pretrained ResNet-50 architecture, progressively resizing input images, and using automatic learning rate selection. They achieved an accuracy of 96.23% on the COVIDx dataset with only 41 epochs, which demonstrated state-of-the-art performance.

Hemdan et al. [20] developed a DL framework called COVIDX-Net to assist radiologists in diagnosing COVID-19 in X-ray images. They used a dataset of 50 CXR images, including 25 confirmed positive cases of COVID-19 because public COVID-19 datasets were not available. COVIDX-Net includes seven different deep CNN architectures that analyze the normalized intensities of the X-ray image to classify patients as either negative or positive for COVID-19. The VGG19 and Dense Net models showed good and similar performances with F1 scores of 0.89 and 0.91, respectively.

Bargshady and colleagues [21] utilized generative adversarial network (GAN) and semi-supervised cycle GAN techniques to expand the training dataset of X-ray images. Their proposed Inception-CycleGAN model achieved an accuracy of 94.2% and an area under the curve (AUC) of 92.2%.

Ozturk et al. [22] presented a DL-based approach for the automated detection of COVID-19 cases from CXR images. They proposed a two-stage DNN architecture, which first detects the presence of lung opacities and then classifies COVID-19. They reported experimental results on a publicly available dataset of CXR images, containing COVID-19 and non-COVID-19 cases. The proposed approach achieved an overall accuracy of 98.5% in COVID-19 detection.

Apostolopoulos and Mpesiana [23] proposed a DL-based approach for the automated detection of COVID-19 cases from CXR images using

transfer learning with CNNs. They fine-tuned pretrained CNN models (VGG16, ResNet50, and InceptionV3) on a dataset of COVID-19 and non-COVID-19 CXR images to improve classification performance. They evaluated the performance of these models using various evaluation metrics and found the InceptionV3 model achieved the highest accuracy of 99.3% in COVID-19 detection, whereas the VGG16 model achieved the highest F1-score of 0.989.

Shome et al. [24] introduced a DL pipeline based on Vision Transformer for detecting COVID-19 from CXR images. They combined three open-source datasets of CXR images to create a 30 K image dataset due to the limited availability of large datasets. The proposed transformer model achieved high accuracy and AUC scores in binary and multiclass classification tasks, outperforming other popular models such as EfficientNetB0, InceptionV3, ResNet50, MobileNetV3, Xception, and DenseNet-121. Additionally, the authors created a visualization based on Grad-CAM to aid interpretation by radiologists and monitor disease progression in affected lungs.

Bahgat et al. [25] introduced an optimized transfer learning-based approach for the automatic detection of COVID-19. This approach utilizes 12 different CNN architectures to diagnose COVID-19 cases from CXR images. They applied an optimization algorithm, specifically the Manta-Ray Foraging Optimization algorithm, to the hyperparameters of these CNN architectures to improve classification performance. The experimental results demonstrated the DenseNet121 optimized architecture achieved the best performance. The evaluation metrics, including accuracy, precision, and recall reached 98.47%, 98.50%, and 98.47%, respectively.

Elpeltegy and Sallam [26] developed an automated tool to detect and diagnose COVID-19 in CXR and CT images by utilizing pretrained DL CNN architectures. They proposed a modification to the ResNet50 architecture by incorporating three new layers, namely, Conv, batch normalization, and activation rectified linear unit (ReLU) layers, to improve discrimination and extract robust features for accurate classification of patients infected with COVID. They evaluated proposed model on CXR and CT scan images of patients with COVID-19 and

achieved diagnosis accuracy of 97.7% for the CT dataset and 97.1% for the X-ray dataset.

Kumar et al. [27] proposed a model by improving the generalization ability of the classifier for binary and multiclass COVID-19 datasets. They employed two commonly used datasets to evaluate the performance of the proposed ensemble model. The results of a comparative analysis demonstrated that the proposed model surpassed state-of-the-art models based on various performance metrics. They developed an ensemble model for the early detection of COVID-19 infection in CXR images, determined the capability of the proposed model to classify suspected patients as either infected with pneumonia or tuberculosis or as healthy, and used pretrained models in the ensemble model to extract potential features and classify them.

Reddy et al. [28] proposed a technique called multimodal fusion of deep transfer learning (MMF-DTL) to classify COVID-19 using radiological images from the Chest-X-Ray dataset. The MMF-DTL model comprises preprocessing, feature extraction, and classification. Three DL models, namely, VGG16, Inception v3, and ResNet 50, were used for feature extraction, and a decision-based multimodal fusion approach was utilized to integrate these models and increase the detection rate. The proposed fusion model achieved high sensitivity, specificity, precision, accuracy, and F-score of 92.96%, 98.54%, 93.60%, 98.80%, and 93.26%, respectively.

Deb et al. [29] proposed CoVSeverity-Net, a DL-based severity estimator to predict the severity of COVID-19 from CXR images. The model extracts low-level features from input CXR images of patients positive for COVID-19 and can quickly determine the severity level of their COVID-19 infection. They trained the model on a publicly available COVID-19 dataset and achieved an accuracy rate of 85.71%, with cross-validation demonstrating five- and ten-fold accuracies of $87.82\% \pm 6.25\%$ and $91.26 \pm 3.42\%$, respectively. CoVSeverity-Net outperformed other state-of-the-art architectures.

Das et al. [30] presented an automated Covid-19 screening model that utilizes CXR images to identify patients with the disease. The model classifies the images into three categories: Covid-19 positive, other pneumonia infections, and no infection. It employs three learning schemes, namely, CNN, VGG-16, and ResNet-50, for model training. The CXR images were obtained from a standard COVID-19 radiography dataset available on Kaggle. The performance evaluation of the model demonstrated VGG-16 performed better compared with CNN and ResNet-50, achieving an impressive accuracy of 97.67%, precision of 96.65%, recall of 96.54%, and F1 score of 96.59%. Moreover, the proposed model outperformed two existing models for Covid-19 screening, highlighting its effectiveness in early and accurate diagnosis.

2.2. COVID-19 diagnosis using CT scan

In addition to automated medical diagnosis models for COVID-19 detection using X-ray images, studies have been conducted using CT images. CT scans provide a more detailed view of the lungs and are more sensitive in detecting COVID-19 than CXRs. Automated models using transfer learning have shown promise in analyzing CT scans for COVID-19 diagnosis.

Voulodimos et al. [31] explored the efficacy of few-shot learning in U-Net architectures for COVID-19 segmentation and initially utilized a trained U-Net model to perform the segmentation of CT scans into regions infected or not infected by COVID-19. During the testing phase, experts evaluated the network results, identifying correctly classified or misclassified outputs. Subsequently, an online few-shot learning was employed to select a few samples for further network training. This dynamic fine-tuning of the U-Net's weights with new few samples contributed to enhanced segmentation accuracy for COVID-19-infected regions. The experimental findings revealed improved segmentation accuracy in identifying COVID-19-infected regions, with a $5.388\% \pm 3.046\%$ enhancement for the IoU metric and a similar increment of $5.394\% \pm 3.015\%$ for the F1 score based on four-fold cross-validation

results of various classifiers.

Li et al. [32] developed a transfer learning-based model for COVID-19 detection using CT images. They used a pretrained a ResNet-50 model and achieved high accuracy and sensitivity in COVID-19 detection. They used a dataset of 618 CT scans (219 COVID-19 positive and 399 COVID-19 negative). The results showed the model achieved an accuracy of 96.78% in detecting COVID-19 cases.

Gaur et al. [33] presented a novel approach for preprocessing and classifying CT scan images into COVID-19 positive and negative cases. The proposed method utilizes empirical wavelet transformation for preprocessing and selects the optimal components of the red, green, and blue channels of the image for training on a proposed network. This approach achieved an accuracy of 85.5%, an F1 score of 85.28%, and an AUC of 96.6% for the classification of COVID-19 cases.

Bejoy and colleagues [34] utilized five pretrained CNNs, namely, MovenetV2, ShuffleNet, Xception, Darknet53, and EfficientnetB0, to extract features. These extracted features were then combined, and an ensemble classifier kernel support vector machine (SVM) was utilized to diagnose COVID-19 cases. The proposed model achieved an accuracy of 0.916, a kappa score of 0.8305, an F-score of 0.91, a sensitivity of 0.917, and a PPV of 0.904.

Pathak et al. [35] employed a DTL technique with a top-2 smooth loss function to classify patients infected with COVID-19. The model outperformed other supervised learning models in terms of efficiency and was evaluated using 10-fold cross-validation on a dataset split into 60% for training, 40% for testing, and 10% for validation. The proposed model achieved high training and testing accuracies of up to 96.2264% and 93.0189%, respectively, demonstrating remarkable results. The top-2 smooth loss function with cost-sensitive attributes was utilized to handle noisy, imbalanced COVID-19 datasets.

Yao et al. [36] developed a computer-aided diagnosis system based on DL for the automatic classification of chest CT-based COVID-19, tuberculosis, and healthy control subjects. The classification model, AdaD-FNN, transfers trained knowledge of an FNN estimator sequentially to the next FNN estimator while updating the weights of the samples in the training set with a decaying learning rate. A novel image preprocessing model, F-U2MNet-C, was also designed to enhance image features and eliminate interference factors. The system achieved high classification accuracies of 99.52%, 92.96%, 97.86%, and 91.97% on four publicly available datasets (TLDCA, UCSD-AI4H, SARS-CoV-2, and TCIA, respectively) and demonstrated compelling performance in assisting COVID-19 detection compared with 22 state-of-the-art methods.

Kundu and colleagues [37] developed a computer-aided diagnosis system, called ET-NET, for fast COVID-19 screening using chest CT-scan images and DL techniques. The approach employs DTL and an average probability-based ensemble method for binary classification. The proposed framework uses a Bagging ensemble of three transfer learning models, namely, Inception v3, ResNet34, and DenseNet201, to improve performance. The ET-NET system achieved high accuracy, precision, sensitivity, and specificity of 97.81%, 97.77%, 97.81%, and 97.77% on the five-fold cross-validation of a publicly available dataset, respectively, outperforming the state-of-the-art methods on the same dataset.

Kathamuthu and colleagues [38] created a CNN framework enhanced by conceptual transfer learning to identify COVID-19 using CT scan images, and their techniques were effective even with limited datasets. This current research builds on their work by examining several DTL-based CNN approaches for detecting COVID-19 in chest CT images. They used six foundational models (VGG16, VGG19, DenseNet121, InceptionV3, Xception, and Resnet50) and evaluated their performance with various performance metrics. VGG16 outperformed the other models with an accuracy of 98.00%. These results indicated the potential of this model for detecting and monitoring patients with COVID-19.

To enhance the transfer learning technique and improve its generalizability across multiple data sources, Alhares and colleagues [39]

proposed a multisource adversarial transfer learning model called AMTLDC. The AMTLDC model learns representations that are similar across different sources, enabling generalization beyond the specific dataset domain. The researchers applied this model to predict COVID-19 using medical images and a CNN. Their results demonstrated the AMTLDC framework can improve accuracy and outperform current successful transfer learning methods, particularly when dealing with different dataset domains or limited data. Their approach achieved at least 2%, 18%, 18%, and 9% improvement in accuracy, precision, recall, and F1 score, respectively.

2.3. Multimodal COVID-19 diagnosis

In addition, some research has been conducted utilizing X-ray and CT images together. Using CT scans and X-ray images can provide complementary information that may aid in the accurate diagnosis of COVID-19. CT scans are more sensitive than CXRs in detecting COVID-19 because they can reveal more details about the lungs. However, CXRs are less expensive, faster to obtain, and expose the patient to less radiation than CT scans. By using imaging modalities together, doctors can improve their accuracy in diagnosing COVID-19 while balancing cost and radiation exposure.

Srivastava and colleagues [40] proposed CoviXNet, a deep-learning neural network architecture for accurately screening COVID-19 using CXRs. The 15-layer CNN architecture has fewer parameters than pre-existing networks and detects visual markers in the chest radiography imaging of patients with COVID-19. The dataset used in this study includes carefully preprocessed images divided into COVID-19, normal, and viral pneumonia CXRs. The CoviXNet model achieved high accuracy of 96.61% in a 10-fold cross-validation test on the three-class dataset, providing accurate, practical COVID-19 detection for radiologists, medical staff, and patients.

Yadav and colleagues [41] developed a framework to classify lung diseases using chest CT and X-ray images. The framework uses multiple-layer generative adversarial networks called Lung-GANs, which learn interpretable representations of lung disease images using unlabeled data. The learned lung features were used to train SVM and a stacking classifier. This approach does not rely on a large quantity of labeled data and can be extended to other similar lung diseases. The proposed framework achieved a high classification accuracy of 97.7% for the CT dataset and 97.1% for the X-ray dataset, demonstrating its effectiveness in simplifying the detection of lung diseases, reducing diagnosis time, and increasing diagnostic convenience.

Manav et al. [42] evaluated the effectiveness of CNN in the analysis of medical images. Specifically, they employed DL models to classify X-ray images into three categories: COVID, viral pneumonia, and normal. In addition to the CNN model (9 LC) developed for this paper, two transfer learning models (VGG16 and VGG19) were utilized and compared. The paper utilized two datasets, one from the Radiodiagnosis department of an institution and another from the Kaggle database. The outcomes showed VGG16 had the highest accuracy rate for both datasets and achieved accuracy rates of 94.96% and 85.71% for the Kaggle dataset and the Radiodiagnosis department dataset, respectively.

Huang et al. [43] optimized seven CNNs, namely, InceptionV3, ResNet50V2, Xception, DenseNet121, MobileNetV2, EfficientNet-B0, and EfficientNetV2, for COVID-19 detection. They also introduced a lightweight CNN called LightEfficientNetV2 to analyze limited numbers of CXR and CT images. They used five-fold cross-validation to validate the performance of each model. The highest accuracy rate achieved on the CXR dataset was 97.73% by EfficientNetV2 after fine-tuning. The proposed LightEfficientNetV2 model achieved an accuracy rate of 98.33% for CXR images. For CT images, Xception had the highest accuracy rate of 96.78% after fine-tuning, whereas the proposed LightEfficientNetV2 model achieved an accuracy rate of 97.48%.

Mukhi et al. [44] proposed a CNN-based method for detecting COVID-19 from CXR images and CT scans. They used the Kaggle dataset

to evaluate the performance of DL-based CNN models such as VGG-19, ResNet-50, Inception v3, and Xception after preprocessing the data. They found that CXRs outperformed CT scans in terms of detection accuracy, with the fine-tuned VGG-19 model achieving the highest accuracy of 94.17% for CXRs and 93% for CT scans. Thus, they recommended using CXRs rather than CT scans for detecting COVID-19, and VGG-19 was the most suitable model for the task.

Hayat et al. [45] identified a new trend in the use of machine learning and DL techniques to overcome limitations in automatic analysis. Specifically, medical imaging techniques can achieve high diagnosis results, as demonstrated by their proposed machine and DL model using CXR and CT scan images to identify patients with COVID-19. They tested the model using four different classifiers: logistic regression, SVM, CNN (SCovNet), and ResNet18. Overall, their study examined 17,599 images and found the proposed SCovNet architecture achieved the highest accuracies of 98.67% and 97.62% for chest CT scan images and CXR images, respectively.

Roy et al. [46] presented a DL model for early COVID-19 detection using CXR and CT images. They merged three benchmark datasets to create a diverse training dataset. Various transfer learning techniques were applied for efficient feature extraction, and a capsule network was used for patient classification. The model achieved impressive accuracies of 98.2% and 97.8% for X-ray and CT images, respectively, and an overall average accuracy of 98%. This integrated approach is effective for the accurate, early COVID-19 detection, leveraging both imaging modalities.

The existing studies suggest that automated medical diagnosis models using X-ray and CT images can be a promising tool for COVID-19 detection. The models achieved high accuracy, sensitivity, and specificity in COVID-19 detection, which is crucial for effective disease management and control. Transfer learning enables the models to learn from pretrained models, which can improve their performance and reduce the need for large datasets. Some existing CNN architectures, such as ResNet, AlexNet, SqueezeNet, Inception, DenseNet, VGG, and Efficient Net, have been proven vital in medical image feature extraction. However, the performance of these models is highly dependent on the quality and size of the dataset used for training. The models may also suffer from biases if the dataset is not representative of the population. More research is needed to validate the performance of these models on larger and more diverse datasets. Table 1 shows a comprehensive review of the related work in COVID-19 diagnosis.

2.4. Research gap and motivation

Despite remarkable advancements in medical imaging and machine learning, the automated diagnosis of COVID-19 remains challenging. Many existing methods lack the required robustness and accuracy, often leading to misdiagnoses or inadequate performance in real-world scenarios. Furthermore, conventional approaches often rely on single-network models, limiting their ability to harness the full potential of diverse datasets and specialized architectures. This limitation underscores the need for a more innovative and adaptive approach that can effectively combine multiple networks while maintaining interpretability and performance.

Our motivation to undertake this research stems from the urgency to develop a reliable, efficient automated COVID-19 diagnosis system. The global effect of the COVID-19 pandemic has placed immense pressure on healthcare systems worldwide, demanding fast, accurate diagnosis to triage and treat patients effectively. By addressing the limitations of the existing methods and leveraging the power of multiple cascaded and conditional networks, we envision a model that can provide accurate, timely diagnoses, aiding healthcare professionals in making informed decisions and improving patient outcomes.

To achieve this vision, CCN for automated COVID-19 diagnosis from X-ray and CT images is proposed. The CCN architecture is designed to cascade multiple neural networks, each with a unique output label

Table 1

State of the Art-A Comprehensive Review of Related Work in COVID-19 Diagnosis.

S/N	Ref	Year	Image type	Performance/algorithm	Data splitting	DB name	Accuracy
1	[17]	2020	X-ray	Federated learning for COVID-19 data training	90% training and 10% testing	COVID-19 Image Data Collection	91.26%
2	[18]	2020	X-ray	Deep COVID Explainer model that employs Transfer learning	Five-fold cross-validation	COVIDx v1.0, v2.0,v3.0 and dataset, Kaggle CXR Pneumonia dataset,	Not specified
3	[19]	2020	X-ray	Deep learning (Covid-ResNet framework)	80% training, 10% validation, and 10% testing	COVID-19 Chest X-ray Dataset Initiative	96.23%
4	[21]	2022	X-ray	CycleGAN and transfer learning techniques	80% training, 10% validation, and 10% testing	COVID-19 Image Data Collection	94.2%
5	[22]	2020	X-ray	Deep neural networks	Not specified	collected from different sources	98.5%
6	[23]	2020	X-ray	Transfer learning with CNNs	70% training, 15% validation, and 15% testing	COVID-19 Image Data Collection	99.3%
7	[24]	2021	X-ray	Transfer Learning	75% training, 15% validation, and 10% testing data	Mendeley and Kaggle datasets	92% (multiclass classification)
8	[25]	2021	X-ray	Optimized transfer learning-based approach	80% training, 10% validation, and 10% testing	COVID-19 Radiography Database (Kaggle dataset)	98.47%
9	[40]	2022	X-ray	CoviXNet, a deep-learning neural network architecture	70% training, 10% validation, and 20% testing data	Mendeley dataset	96.23%
10	[27]	2023	X-ray	Transfer learning models with Efficient Net, GoogleNet, and Xception Net networks	15% as testing data. Of the remaining 85%, 17% is used for validation and 68% is used for training.	Dataset 1 (Kaggle dataset). Dataset 2 (Mporas and Naronglerdrit dataset)	99.21% for multiclass and 98.95% for binary
11	[28]	2023	X-ray	MMF-DTL model using three DL models, namely, VGG16, Inception v3, and ResNet 50, for feature extraction	Not specified	COVID-chest Xray-dataset (GitHub dataset)	98.80%
12	[29]	2023	X-ray	CoVSeverity-Net, a deep-learning-based severity estimator	Five-fold cross-validation and ten-fold cross-validation	Kaggle dataset	87.82% (5 folds) 91.26% (10 folds)
13	[30]	2021	X-ray	Automated Covid-19 screening model	Not specified	Kaggle dataset	97.67%
14	[31]	2021	CT	Few-shot learning in U-Net architectures for COVID-19 segmentation	Four-fold cross-validation	Collected dataset	Not specified
15	[33]	2022	CT scan	Transfer learning of five pretrained CNNs, namely, MobilenetV2, ShuffleNet, Xception, Darknet53, and EfficientnetB0	Split into training, validation, and testing data randomly with dataset sizes of 1000, 100, and 152 images respectively	Kaggle dataset	85.5%
15	[34]	2022	CT scan	Ensemble of CNNs and kernel SVM (KSVM) classifier	70% training set and a 30% test set	COVID-CT dataset	91.6%
16	[35]	2022	CT scan	Deep transfer learning (DTL) is used to build a COVID-19- patient classification model. Epochs of classification models are set to be 1.	60% for training and 40% for testing. Out of 60% of training data, 10% of data is used for validation purposes.	Collected dataset	93.02%
17	[36]	2023	CT scan	Novel image preprocessing model F-U2Mnet-C by enhancing the image features using fuzzy stacking and eliminating the interference factors using U2Mnet segmentation	Five-fold cross-validation	Four publicly available datasets, namely, TLDCA, UCSD-AI4H, SARS-CoV-2, and TCIA	99.52%, 92.96%, 97.86%, and 91.97% w.r.t. datasets
18	[37]	2022	CT scan	A bootstrap aggregating or Bagging ensemble of three transfer learning models, namely, Inception v3, ResNet34, and DenseNet201, is used to boost the performance of the individual models.	70% training data and 30% as testing data.	Kaggle dataset	97.81%,
19	[38]	2023	CT scan	DTL techniques with CNNs with the VGG16, VGG19, Densenet121, InceptionV3, Xception, and ResNet50 models	80% for training and 20% for test data. 10% of test data is preserved as validation data.	Kaggle dataset	98.00%
20	[39]	2023	CT scan	Multisource adversarial transfer learning model, namely, AMTLDCC using the ResNet 50 network	Not specified	Three datasets (SARS-CoV-2 CT, COVID19-CT, and COVID19-CT_v2)	99.8%, 95.1%, 90.41% w.r.t. datasets
21	[26]	2021	X-ray CT scan	Pretrained deep learning CNN architectures	80% in the training set and 20% in the testing set	Mendeley dataset	97.1% 97.7%
22	[41]	2021	X-ray CT scan	Support vector machine (SVM) and a stacking classifier	70% of the dataset for training and the rest 30% as a testing set	Mendeley dataset	94.0% 97.3%
23	[42]	2021	X-ray CT scan	CNN model compared with two TL models (VGG16 and VGG19)	70% for training, 15% for validation, and 15% for the test.	Kaggle dataset Radiodiagnosis dataset	94.96% 85.71%
24	[43]	2022	X-ray CT	Seven CNN (InceptionV3, ResNet50V2, Xception, DenseNet121, MobileNetV2, EfficientNet-B0, and EfficientNetV2)	Five-fold cross-validation	Kaggle dataset Mendeley dataset	98.33% 97.48%
25	[44]	2023	X-ray CT	Deep-learning-based CNN models such as VGG-19, ResNet-50, Inception v3, and Xception	80% of the images were used and the remaining 20% of the images were used for testing.	Kaggle dataset	94.17% 93%
26	[45]	2023	X-ray CT scan	Machine learning and deep learning techniques	70% for the training, 15% for the testing, and 15% for the validation purpose	Mendeley dataset	97.62% 98.67%
27	[46]	2023	X-ray CT scan	Deep learning model for early COVID-19 detection using chest X-ray and CT images	Not specified	Collected dataset	98.2% 97.8%

corresponding to specific diagnostic tasks. This cascade approach facilitates the effective combination and refinement of diagnostic information from different network stages, ultimately improving accuracy and robustness compared with traditional ensemble techniques and single-network models.

In this paper, various publicly available datasets of COVID-19, namely, the Kaggle X-ray dataset, the Mendeley X-ray dataset, and the Mendeley CT dataset, are utilized. From these datasets, several subsets with different classes are carefully constructed for our analysis, ensuring the comprehensive coverage of relevant diagnostic scenarios. The availability of diverse datasets enables our model to learn from a wide range of COVID-19 manifestations and different imaging protocols, enhancing its ability to generalize to unseen data.

Transfer learning plays a crucial role in our approach, offering remarkable advantages in COVID-19 diagnosis. By leveraging the pre-trained models, our CCN can extract relevant features from limited datasets effectively. This technique not only accelerates model training but also enhances generalization to new cases, addressing the challenges posed by data scarcity in the context of the ongoing pandemic.

Moreover, a five-fold cross-validation strategy is incorporated during the model training and evaluation. This approach partitions the dataset into five subsets and performs five training and testing cycles, ensuring each subset serves as training and testing data once. The cross-validation strategy enhances the model's performance estimation, reduces variance, and facilitates effective parameter tuning, resulting in a more reliable and stable diagnostic system.

The combination of occlusion specificity and Grad-CAM techniques enhances our model's output understanding. This adaptable, scalable solution eliminates manual hyperparameter tuning and holds the potential for critical early COVID-19 screening and diagnosis. By leveraging these cutting-edge techniques, our proposed method not only improves the model's reliability and performance but also opens avenues for further research and refinement. The adaptability and scalability of our system enable its deployment across various medical settings, ranging from local clinics to large-scale healthcare facilities. This widespread applicability enhances its potential to reach a broader population and contribute to global efforts in combatting COVID-19.

In conclusion, the primary contribution of this work lies in the proposal and implementation of CCN for automated COVID-19 diagnosis from X-ray and CT images. Our research addresses the limitations of existing methods and bridges the gap between conventional single-network approaches and more advanced ensemble techniques. By leveraging transfer learning, diverse datasets, and a robust evaluation strategy, our CCN model offers a promising approach for automated COVID-19 diagnosis, with potential applications in clinical settings and public health emergencies. The development of accurate, and efficient diagnostic tools is vital in the ongoing battle against COVID-19, and our paper aims to make a valuable contribution to this important endeavor.

3. Material and methods

3.1. Data collection and preprocessing

Various publicly available datasets of COVID-19 were utilized to perform our analysis: Kaggle X-ray dataset [4748], Mendeley X-ray dataset, and Mendeley CT dataset. From the Kaggle X-ray dataset, three datasets were constructed: DB1, DB1-v2, and DB2. DB1 contains 5856 X-ray images categorized into three classes: normal images (1583), viral images (1493), and bacterial images (2780). DB1-v2 contains the same number of X-ray images as DB1, but it has only two classes, namely, normal images (1583) and pneumonia images (4273), which include viral and bacterial X-ray images. DB2 has 4273 image and two classes: viral images (1493) and bacterial images (2780). Additionally, Enhanced-DB2 was constructed by merging Kaggle X-ray dataset and Mendeley X-ray dataset, which contains 5560 images classified into two: viral images (2780) and bacterial images (2780).

Furthermore, DB3 was constructed from the Mendeley X-ray dataset, which has 9544 images categorized into two: COVID-19 Images (4044) and non-COVID-19 Images (5500). Lastly, the Mendeley CT dataset was used to construct DB4, which contains 8055 images classified into two: COVID-19 Images (2628) and non-COVID-19 Images (5427). Table 2 describes the adopted datasets for our analysis, and Fig. 2 illustrates examples of images from the different classes of the adopted datasets.

3.2. Transfer learning models

DNNs have revolutionized the field of artificial intelligence in recent years, particularly in the area of image classification. DNNs are typically trained using large datasets to learn patterns and features that enable them to classify images accurately. However, training a DNN from scratch on a large dataset can be computationally expensive and time consuming. Pretrained DNNs solve this problem by leveraging the power of transfer learning, where a DNN trained on a large dataset is used as a starting point for a new task.

Several pretrained DNNs are available for image classification tasks, including GoogleNet, SqueezeNet, ResNet 18, ResNet 50, AlexNet, and Dark Net 19. These models have been trained on large datasets, such as ImageNet, and have achieved state-of-the-art performance on various benchmarks. By using a pretrained DNN as a starting point, researchers and practitioners can save time and computational resources and still achieve high accuracy in their classification tasks. In this article, an overview of these popular pretrained DNNs is provided, and their strengths and weaknesses are discussed.

GoogleNet is a DNN architecture that won the ImageNet Large Scale Visual Recognition Challenge in 2014. It introduced the concept of Inception modules, which use multiple convolutional filters of different sizes in parallel to capture features at different scales. GoogleNet has 22 layers and is known for its efficiency and accuracy in image classification tasks.

SqueezeNet is a lightweight DNN architecture designed for mobile and embedded devices with limited computational resources. It achieves high accuracy with a substantially fewer parameters than other DNNs. SqueezeNet uses a novel Fire module that combines 1×1 and 3×3 convolutional filters to reduce the number of parameters.

ResNet 18 and ResNet 50 are part of the residual network (ResNet) architecture family. ResNet introduced the concept of residual connections, which allow the network to learn residual features that are easier to optimize than the original features. ResNet 18 has 18 layers, whereas

Table 2
Description of the Adopted Datasets.

Data Set No	Ref	Database Source Name	Total number of images	Number of Classes and distribution
DB1	[47]	Kaggle Xray dataset [48]	5856 images	3 (1583 normal images, 1493 viral images, and 2780 bacterial images)
DB1-v2	[47]	Kaggle Xray dataset [48]	5856 images	2 (1583 normal images and 4273 pneumonia images) pneumonia = viral + bacterial
DB2	[47]	Kaggle Xray dataset [48]	4273 images	2 (1493 viral images and 2780 bacterial images)
Enhanced-DB2	[47,49]	Kaggle Xray dataset [48] + Mendeley Xray dataset	5560 images	2 (2780 viral images and 2780 bacterial images)
DB3	[42]	Mendeley Xray dataset	9544 images	2 (4044 COVID-19 images and 5500 non-COVID-19 images)
DB4	[42]	Mendeley CT dataset	8055 images	2 (2628 COVID-19 images and 5427 non-COVID-19 images)

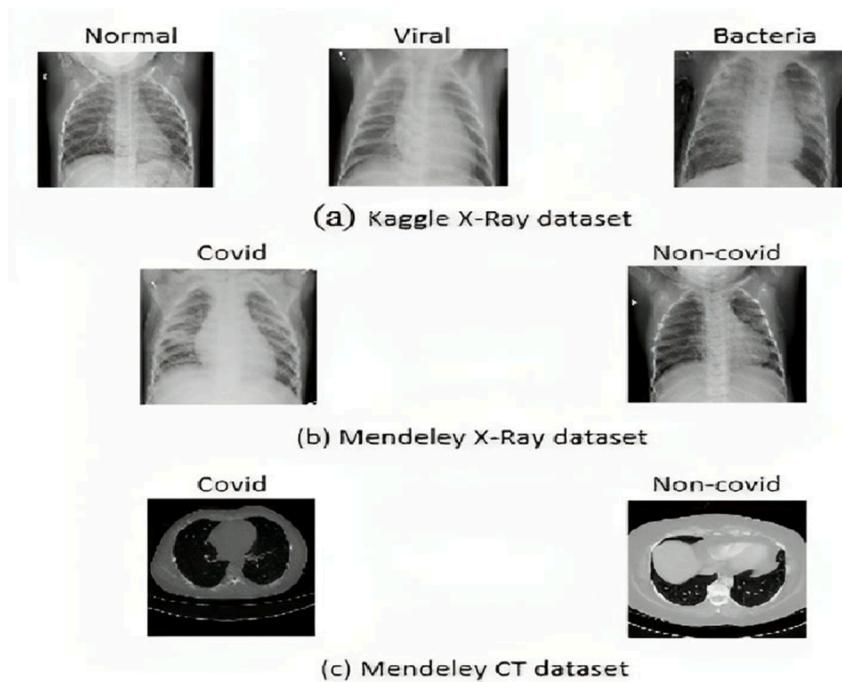


Fig. 2. Examples for Classes from the Adopted Dataset.

ResNet 50 has 50 layers. Both ResNet models have achieved state-of-the-art performance on various image classification benchmarks.

AlexNet is one of the earliest DNN architectures that achieved breakthrough performance on the ImageNet challenge in 2012. It has 8 layers and uses a combination of convolutional, max pooling, and fully connected layers. AlexNet was the first DNN to use ReLU as activation functions, which improved the training time and accuracy of the model.

Dark Net 19 is a DNN architecture designed for the You Only Look Once object detection system. It has 19 layers and is known for its speed and accuracy in real-time object detection tasks. Dark Net 19 uses a combination of convolutional and max pooling layers with a few fully connected layers at the end.

ShuffleNet is a neural network architecture that was introduced in 2018 for performing image classification tasks. The key innovation of ShuffleNet is its use of channel shuffling, which allows feature maps to be routed between channels in different parts of the network. This feature enables ShuffleNet to achieve high accuracy on image classification tasks with substantially fewer parameters than other state-of-the-art models.

ShuffleNet comes in different variants with varying numbers of layers, ranging from ShuffleNet V1 with $2.4 \times$ fewer parameters than MobileNetV1 to ShuffleNet V2 with $1.5 \times$ fewer parameters than MobileNetV2. The number of layers in ShuffleNet varies depending on the specific variant being used but generally ranges from tens to hundreds of layers.

Table 3 lists the adopted pretrained networks trained on ImageNet and some of their properties. The network depth is defined as the largest number of sequential convolutional or fully connected layers on a path from the input layer to the output layer. All networks take RGB images or 3D images as input. Grayscale images or 2D images are replicated three times to ensure compatibility with the network's input requirements.

The table below presents the names of various pretrained models, such as GoogleNet, SqueezeNet, ResNet18, ResNet50, Alex Net, DarkNet19, and ShuffleNet. It also includes information on their initial weights, which could be none, ImageNet, or Space365, as well as the image input size for each network.

Transfer learning is a commonly used technique in DL, in which a

Table 3
Pretrained Models and Their Input Image Size.

Model	No. of Convolutional layers	Parameters	Initial weights	Input size
AlexNet	8	61 M	None, ImageNet ImageNet)	$227 \times 227 \times 3$
GoogleNet	22	7 M	Place365, ImageNet	$224 \times 224 \times 3$
DarkNet19	19	20.8 M	ImageNet	$256 \times 256 \times 3$
ResNet 18	18	11.7 M	ImageNet	$224 \times 224 \times 3$
ResNet 50	50	25.6 M	ImageNet	$224 \times 224 \times 3$
SqueezeNet	18	1.24 M	ImageNet	$227 \times 227 \times 3$
ShuffleNet	50	1.4 M	ImageNet	$224 \times 224 \times 3$

model trained on one task is reused for a second related task, enabling leveraging the knowledge gained while solving the first problem. It is particularly valuable in DL when working with limited data or computational resources because it can reduce the need for training a new model from scratch, which can be expensive and require enormous amounts of data. The technique allows researchers to take advantage of pretrained models on large-scale datasets, such as ImageNet, which have been used to pretrain many popular DL models. The pretrained models are utilized based on the dataset and its size. When the dataset is small, using all weights of the pretrained models as feature extractors and retraining the classifier are recommended. For medium to large datasets, fine-tuning the pretrained models is a better approach where previous weights are used for initialization, and training is performed on the complete network or some of the higher layers.

To cater to the different sizes of datasets used in our paper, feature extraction and fine-tuning approaches were adopted by using pretrained models with initial weights of none, ImageNet, and Space365. Feature extraction is a fast, straightforward way to utilize the power of DL without requiring GPU or investing time and effort into training a full

network. It involves a single pass over the training images and is particularly useful when the new dataset is exceedingly small.

Fine-tuning deeper layers of the network can improve accuracy when the new dataset is similar to the original data. However, if the new dataset is considerably different from the original data, then training the final classifier on more general features extracted from an earlier network layer may be more useful. If the new dataset is large, then training a network from scratch can also be attempted.

In this paper, transfer learning was used to classify COVID-19 from X-ray and CT images using pretrained models. The pretrained models' convolutional layers were used as feature extractors, and the classifier layers were replaced with new layers. The pretrained models were already proficient in the top layers because they were designed for the classification of 1000 images. In the fine-tuning approach, the last three layers of the pretrained models were replaced with a fully connected layer, an activation layer with SoftMax, and a classification layer, given the binary classification problem.

3.3. Cross-validation strategy

The K-fold cross-validation strategy is used to train and test a model by dividing the dataset into equally sized folds, with each fold used as validation data to test the model. By using cross-validation, all the data points are used at least once for testing, providing a better estimation of model performance. Cross-validation also helps in reducing the variance in estimated performance, making the evaluation of the model more stable. Additionally, it provides an opportunity to tune model parameters more effectively by identifying the best parameters for the model and avoiding overfitting. Overall, cross-validation with five folds is a more reliable and accurate method compared with splitting data into 80/20.

3.4. Data augmentation

Data augmentation is a technique used in machine learning and DL to increase the size and diversity of a dataset by creating new training examples from the existing ones. It involves applying a set of transformations to the original data to create new samples that are similar but not identical to the original ones. Data augmentation aims to improve the generalization of the model by exposing it to a more diverse range of examples during training. By creating new training examples, data augmentation can help the model learn more robust, invariant features that are useful in recognizing patterns in the data. Data augmentation techniques can vary depending on the type of data being used. For example, in image classification, data augmentation can involve random cropping, flipping, rotating, scaling, or changing the brightness and contrast of images. Data augmentation can be especially useful in scenarios with a limited amount of training data available or when the model needs to be trained on data from different domains. It can also help in dealing with class imbalances by creating new samples for the minority classes. It is considered a powerful technique that can remarkably improve the performance of machine learning and DL models by creating more diverse and representative datasets. The following tables the augmentation techniques applied to the training datasets for X-ray and CT images.

Tables 4 and 5 present different augmentation types, such as random rotation, random reflection, and random shear, along with their intended purpose and values for X-ray and CT images. The only notable difference between the two tables is the random y reflection value, which is 1 for CT images and 0 for X-ray images.

The development environment used to train and test the models is Windows operating system, Intel(R) Core (TM) i7-9750H, 2.60-GHz processor CPU, HDD 550 GB, RAM GB 16 GB, an Nvidia GeForce GTX 1660 Ti 6 GB GPU, MATLAB 2022b, Keras 2.1.1 with backend TensorFlow 1.4.0, and for GPU acceleration CUDA 8.0/cuDNN 5.1 dependencies.

Table 4
Image Augmentation Parameters Used for All X-ray Images.

Augmentation Type	Purpose	Value
Rand Rotation	Rotate an image. Randomly choose a rotation angle in the specified range from a continuous uniform distribution.	[-55]
Rand X Reflection	This parameter is used for the reflection of an image in the left-right direction. If the value is 1, horizontally reflect an image with 50% probability.	1
Rand Y Reflection	This parameter is used for reflection of an image in the top-bottom direction. When the value is 1, vertically reflect an image with a 50% probability.	0
Rand X Shear	Shear an image horizontally. Randomly choose a shear angle within the specified interval from a continuous uniform distribution.	[-0.05 0.05]
Rand Y Shear	Shear an image horizontally. Randomly choose a shear angle within the specified interval from a continuous uniform distribution.	[-0.05 0.05]

Table 5
Image Augmentation Parameters Used for all CT Images.

Augmentation type	Purpose	Value
Rand Rotation	Rotate an image. Randomly choose a rotation angle in the specified range from a continuous uniform distribution.	[-55]
Rand X Reflection	This parameter is used for the reflection of an image in the left-right direction. If the value is 1, horizontally reflect an image with 50% probability.	1
Rand Y Reflection	This parameter is used for the reflection of an image in the top-bottom direction. When the value is 1, vertically reflect an image with 50% probability.	1
Rand X Shear	Shear an image horizontally. Randomly choose a shear angle within the specified interval from a continuous uniform distribution.	[-0.05 0.05]
Rand Y Shear	Shear an image horizontally. Randomly choose a shear angle within the specified interval from a continuous uniform distribution.	[-0.05 0.05]

3.5. Optimizer techniques

In DL, optimization algorithms are used to train the neural network by adjusting the weights and biases of the model during the training. Several optimization algorithms are available, and RMSprop, Adam, and SGDM, which are popular and widely used, were adopted.

RMSprop is an optimization algorithm that adjusts the learning rate of each weight and bias parameter based on the root mean square of the past gradients. This approach helps prevent the learning rate from oscillating too much or being too large. RMSprop is an extension of gradient descent and improves the convergence of the training.

Adam is a popular optimization algorithm that combines the ideas of momentum and RMSprop. Adam maintains exponential moving averages of the gradients and the squared gradients and uses them to adjust the learning rate of each parameter adaptively. Adam converges faster and achieves better performance than RMSprop and other optimization algorithms in many DL tasks.

SGDM is an extension of the standard stochastic gradient descent algorithm that adds momentum to the optimization [50]. Momentum helps the optimization algorithm move faster along the gradient direction and smoothens the fluctuations in the gradient direction. SGDM is effective in accelerating the training and achieving better convergence.

Each of these optimization algorithms has its strengths and weaknesses, and the choice of which one to use depends on the specific task and the model being trained. RMSprop is effective in preventing the

learning rate from oscillating too much, Adam is adaptive and can converge faster, and SGDM is effective in accelerating the training. Experimenting with different optimization algorithms is recommended to find the one that works best for a specific task.

3.6. Performance metrics

The performance metrics adopted in this paper to evaluate the performance of DL classifiers include accuracy, precision, recall, F1-score, specificity, and AUC. Accuracy is the most commonly used metric and measures the fraction of correct predictions made by the model out of all the predictions made. It is useful when the data are well-balanced and not skewed toward a specific class. Precision is another important evaluation metric that measures the proportion of true positive predictions out of all positive predictions made by the model. It is useful when the goal is to minimize false positives, such as in medical diagnoses. Recall or sensitivity (true positive rate) measures the proportion of true positives out of all actual positives. It is useful when the goal is to minimize false negatives, such as in identifying fraud cases. F1-score is a harmonic mean of precision and recall and is an overall measure of the model's accuracy. It provides a single value that combines precision and recall, making it useful when both metrics are important. Specificity, also known as false positive rate, is the proportion of negative data correctly identified as negative by the model compared with all negative data. A high specificity indicates the model is better at correctly identifying negative cases. AUC or area under the ROC curve is a single-value metric that measures the overall performance of the model in distinguishing between classes [50]. Table 6 depicts the formulas of the adopted performance metrics.

4. Methodology

Our paper aims to enhance the accuracy of COVID-19 diagnosis in X-ray and CT images. Our proposed method, as depicted in Fig. 3, involves utilizing cascaded conditional deep learning networks (CCDLN), which are trained on four datasets. To assess the model's effectiveness, a five-fold cross-validation technique, where the outcome of each validation iteration is averaged, is utilized. This approach helps mitigate the likelihood of overfitting and provides a more dependable assessment of the model's performance compared with a single train-test split. Furthermore, it optimizes data usage by utilizing each data point for training and validation purposes.

Our proposed method involves a comprehensive three-stage methodology, where each stage is composed of three essential processes. The first process, which is data preparation, involves resizing the input image based on the input layer of the transfer learning model and implementing various augmentation techniques, as detailed in Tables 4 and 5. Then, the input image is normalized. During the train/test transfer learning, one or more pretrained models trained on four distinct datasets, consisting of three X-ray datasets and one CT dataset explained in the dataset description section, for each stage, are used. This

categorization guarantees our classification outcomes are highly accurate and reliable. Finally, the performance of our approach is evaluated using various metrics, such as accuracy, precision, sensitivity, specificity, recall, and AUC.

During the first stage, three different phases that aim to achieve optimal performance and accuracy are assessed. The phase that performs the best is incorporated into our final approach or platform. In the first phase of our paper, which involves a single network, the Kaggle DB1 dataset is used. This dataset has three categories, namely, normal, viral, and bacterial, comprising 1583, 1493, and 2780 images, respectively. All the adopted transfer learning models in our paper, namely, GoogleNet, SqueezeNet, ResNet-18, ResNet-50, and AlexNet, are trained on this dataset. Additionally, three optimizers (ADAM, RMSPROP, and SGDM) are applied to train the models. Finally, for each network, various performance metrics such as accuracy, precision, and recall are evaluated.

The second phase involves training two CCNs on two datasets: DB1-ver2 and DB2. The first network is trained on DB1-ver2, which features two output labels: normal and pneumonia. The second network is trained on DB2, which also has two output labels: viral and bacterial. The Kaggle dataset is reused by dividing the images into two classes called normal and pneumonia categories (DB1-ver2). Then, a subset of the Kaggle DB1 dataset containing only viral and bacterial images (DB2) is used to train the second network. The first network is used to classify images into normal and pneumonia, and the second network classifies images into viral and bacterial. All transfer learning models are trained on these datasets to obtain the best performance model for the first and second network models. Unlike the first phase that only contains a single network obtained after training/testing some pretrained model and identifying the optimal one, the second and third phases train two cascaded or multilayer networks.

The authors hope this recommendation will bolster the overall performance. To illustrate our scenario for the second and third phases, if an X-ray image is classified as pneumonia by the first DL network, it is then passed onto the second DL model to determine whether the pneumonia is caused by a viral or bacterial infection. The overall accuracy for this phase is the average accuracy of the two best models trained on the two datasets.

The third phase is similar to the second phase, but the number of viral images in the database (DB2) is increased. It involves training two cascaded conditional networks on two datasets: DB1-ver2 and enhanced-DB2. The first network in the third phase is the same as the first network in the second phase with matching datasets and labels. It utilizes the Kaggle dataset with images divided into normal and pneumonia categories (DB1-ver2), and the second network in the third phase uses enhanced-DB2, which is a subset of the Kaggle DB1 dataset, including another Mendeley COVID-19 X-ray dataset to increase the number of viral images.

The first phase of the study focuses on training a single network on a single dataset, whereas the second phase involves training two networks on two datasets. However, the second and third phases use the same first network and dataset (DB1-ver2). As a result, the third phase only requires training the second network on an enhanced DB2 dataset, unlike the second phase.

In the second stage, a single network is trained on the Mendeley X-ray DB3 dataset, which consists of two classes (COVID and non-COVID). All adopted transfer learning models, namely, GoogleNet, SqueezeNet, ResNet-18, ResNet-50, and AlexNet, are trained on the Mendeley X-ray dataset (referred to as DB3 in this paper).

After utilizing the first and second stages for X-ray images, our model can accurately determine whether a patient is diagnosed with COVID-19 because the image is evaluated by two or three networks (based on the phase used by the first stage). Evidence suggests using multiple networks with few labels improves results compared with using a single network with many labels.

The use of multiple cascaded and conditional networks, as suggested

Table 6
Performance Metrics Formulas.

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall or Sensitivity	$\frac{TP}{TP + FN}$
F1-score	$\frac{2 \times Precision \times Recall}{Precision + Recall}$
Specificity	$\frac{FP}{TN + FP}$
AUC	$TPR = \frac{TP}{TP + FN} \text{ and } FPR = \frac{FP}{FP + FN}$

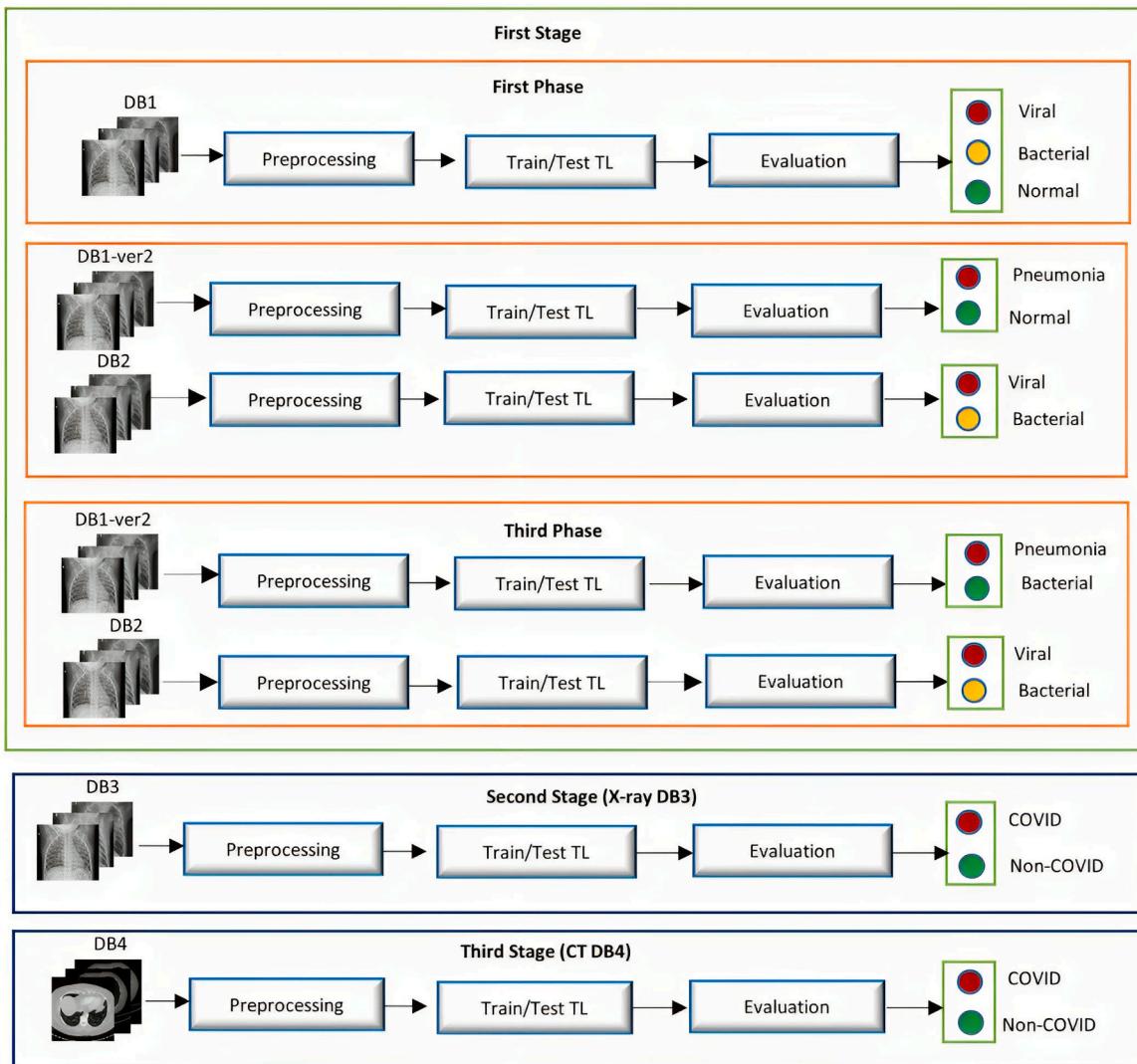


Fig. 3. Architecture of our CCN approach for COVID-19 Diagnosis.

here, is different from ensembling techniques that combine the outputs of multiple independent networks to boost system accuracy and robustness compared with using a single model because each network in the cascade is trained on a distinct dataset with unique output labels.

In the third stage, a single network is trained on the Mendeley CT DB4 dataset, which consists of two classes (COVID and non-COVID). All transfer learning models adopted in the paper, namely, GoogleNet, SqueezeNet, ResNet-18, ResNet-50, and AlexNet, and two custom models are trained on the Mendeley CT dataset (referred to as DB4 in this paper).

Table 7 displays the general hyperparameters chosen for training and evaluating the models. To train the models, experiments are conducted with various optimizers, such as SGDM, Adam, and RMSprop. The initial learning rate value for the models is set to 0.0001, and a constant learning rate schedule is used with a maximum of 10 epochs. To reduce overfitting, five-fold cross-validation is used, and the data are shuffled every epoch. A GPU is employed as the execution environment, and the categorical cross-entropy loss function is used. Additionally, the weights are transferred from the pretrained models with ImageNet, None, and Place365 weights.

5. Results

This section presents the outcomes of our comprehensive study,

Table 7
General Hyperparameters of the Pretrained Models for Training and Testing.

Parameter	Value/Type
Image size	224 × 224 × 3, 227 × 227 × 3, and 256 × 256 × 3
Initial learn rate	1.0000e-04 or 10e-5
Learn rate schedule	Constant
Mini-batch size	10 and 20
Shuffle	Every epoch
optimizer	Adam, RMSprop, and SGDM
Cross-validation	5 fold
Max epochs	10
Execution Environment	GPU
Loss function	Categorical cross-entropy
Activation	SoftMax (final dense layer)
CNN weights	ImageNet, None, and Place365

highlighting the effectiveness of our proposed approach for the diagnosis of COVID-19 in CXR and CT images through transfer learning. Our approach is carefully designed to work in three stages, with the first stage comprising three distinct phases, each contributing to the overall diagnostic process. The second and third stages involve training transfer learning models on different datasets, leveraging the power of pretrained models to enhance the diagnostic accuracy.

Throughout our evaluation, various performance metrics, including

accuracy, sensitivity, specificity, precision, recall, and AUC are employed. These metrics enable assessing the performance of our models thoroughly and gaining valuable insights into the effectiveness of each phase and the individual transfer learning models utilized.

One of the substantial strengths of our approach lies in its adaptability and scalability. By working in multiple stages, distinct aspects of the diagnosis can be systematically addressed, allowing a more comprehensive evaluation and improvement of the model's performance. This multistage framework sets our approach apart from other works that often adopt a single-stage or one-size-fits-all approach, limiting their ability to fine-tune and optimize diagnostic accuracy effectively.

Moreover, the incorporation of transfer learning in our approach offers remarkable advantages. By leveraging pretrained models, relevant features can be effectively extracted from limited COVID-19 datasets, leading to better generalization and improved diagnostic accuracy. This aspect distinguishes our work from traditional methods that heavily rely on manual feature engineering and may suffer from overfitting due to limited data availability.

Overall, the results in this section demonstrate the potential of our approach for accurate, reliable COVID-19 diagnosis from CXR and CT images. By employing a systematic multistage framework, along with transfer learning, the challenges of COVID-19 diagnosis can be effectively tackled, making a substantial contribution to the field of medical image analysis and paving the way for improved patient care and management during the ongoing pandemic.

5.1. First stage-First phase

Table 8 displays the distribution of Kaggle DB1 images across five-fold cross-validation. The number of images for each fold dataset is presented. **Table 9** depicts the performance metrics for the models in the first stage-first phase. The first column shows the name of the model, and the second column indicates the optimizer used, which can be ADAM, SGDM, or RMSPROP. The remaining columns exhibit various evaluation metrics, such as AUC, accuracy, sensitivity, specificity, precision, and recall. Among the models evaluated, the **ResNet 50 model trained with the ADAM optimizer** achieved the highest accuracy of 82.821% in this phase. Moreover, this model had a sensitivity of 0.972205, a specificity of 0.774865, a precision of 0.615354, a recall of 0.972205, and an AUC of 0.923228.

5.2. First stage-Second phase

First network.

Table 10 presents the distribution of Kaggle DB1-ver2 images across five-fold cross-validation. The number of images for each fold dataset is presented.

Table 11 displays the performance metrics for the models in the first stage, second phase-first network. The first column shows the name of the model, and the second column indicates the optimizer used, which can be ADAM, SGDM, or RMSPROP. The remaining columns exhibit various evaluation metrics, such as AUC, accuracy, sensitivity, specificity, precision, and recall. Among the models evaluated, the Resnet 50

Table 8
Data Distribution in Five Folds after Applying Five-Fold Cross-Validation for the First Phase.

	Train			Test		
	Bacteria	Normal	Viral	Bacteria	Normal	Viral
Fold 1	2224	1266	1194	556	317	299
Fold 2	2224	1266	1195	556	317	298
Fold 3	2224	1266	1195	556	317	298
Fold 4	2224	1267	1194	556	316	299
Fold 5	2224	1267	1194	556	316	299

model trained with the RMSPROP optimizer achieved the highest accuracy of 97.67% in this phase. Moreover, this model had a sensitivity of 0.969046, a specificity of 0.97964, a precision of 0.946329, a recall of 0.969046, and an AUC of 0.995621.

Our evaluation showed that utilizing pretrained weights from ImageNet improved the performance of the AlexNet and GoogleNet models compared with using no pretrained weights. Similarly, for the GoogleNet model, utilizing ImageNet weights resulted in superior performance compared with using pretrained weights from the space365 dataset.

Second network.

Table 12 presents the distribution of Kaggle DB2 images across five-fold cross-validation. The number of images for each fold dataset in **Table 13** displays the performance metrics for the models in the first stage, second phase-second network. The first column shows the name of the model, and the second column indicates the optimizer used, which can be ADAM, SGDM, or RMSPROP. The remaining columns exhibit various evaluation metrics, such as AUC, accuracy, sensitivity, specificity, precision, and recall. Among the models evaluated, the AlexNet model trained with the ADAM optimizer achieved the highest accuracy of 79.546% in this phase. Moreover, this model had a sensitivity of 0.681179, a specificity of 0.856835, a precision of 0.718728, a recall of 0.681179, and an AUC of 0.84523.

Due to the excessive time taken for evaluation, the performance of the Resnet50 model could not be assessed with the SGDM optimizer.

The results indicate the first network in the second stage achieved an accuracy of approximately 97.6776% using the RESNET model with the RMSPROP optimizer, whereas the second network in the second stage achieved an accuracy of about 79.546% using the AlexNet model with the ADAM optimizer. Therefore, the average accuracy of the two networks is approximately $(97.6\%+79.5\%)/2 = 88.5\%$. These findings indicate the models in the second stage outperformed the models in the first stage (82.821%).

First stage-Third phase:

As previously mentioned, the first step of the second and third stages involves training pretrained models on the Kaggle DB1-ver2 dataset. **Table 14** presents the distribution of enhanced-DB2 images across five-fold cross-validation. The number of images for each fold dataset is presented. **Table 15** displays the performance metrics for the models in the third phase-second network. The first column shows the name of the model, and the second column indicates the optimizer used, which can be ADAM, SGDM, or RMSPROP. The remaining columns exhibit various evaluation metrics, such as AUC, accuracy, sensitivity, specificity, precision, and recall. Among the models evaluated, the GoogleNet model trained with the ADAM optimizer achieved the highest accuracy of 84.0468% in this stage. Moreover, this model had a sensitivity of 0.882014, a specificity of 0.798921, a precision of 0.814347, a recall of 0.882014, and an AUC of 0.910868.

Due to time constraints, the performance of the ResNet50 model with the SGDM and ADAM optimizer was not evaluated. The results indicate the first network in the third stage achieved an accuracy of approximately 97.6776% using the ResNet model with the RMSPROP optimizer, whereas the second network in the second stage achieved an accuracy of about 84.0468% using the GoogleNet model with the ADAM optimizer. Therefore, the average accuracy of the two networks is approximately $(97.546\%+84.04\%)/2 = 90.793\%$. These findings indicate the models in the third stage outperformed those in the second stage (88.5%).

SECOND STAGE

Table 16 presents the distribution of DB3 images across five-fold cross-validation. The number of images for each fold dataset is presented. **Table 17** displays the performance metrics for the models in the second stage. The first column shows the name of the model, and the second column indicates the optimizer used, which can be ADAM, SGDM, or RMSPROP. The remaining columns exhibit various evaluation metrics, such as AUC, accuracy, sensitivity, specificity, precision, and recall. Among the models evaluated, the ResNet18 model trained with

Table 9

Performance Metrics for First Phase Models.

Model Name	Training Function	AUC	Accuracy	Sensitivity	Specificity	Precision	Recall
GoogleNet	SGDM	0.887959	0.789447	0.938092	0.734379	0.566794	0.938092
GoogleNet	ADAM	0.916032	0.818306	0.95957	0.765972	0.603017	0.95957
GoogleNet	RMSPROP	0.912291	0.814549	0.951358	0.763866	0.598807	0.951358
SqueezeNet	SGDM	0.868658	0.760417	0.887555	0.713316	0.534221	0.887555
SqueezeNet	ADAM	0.888864	0.781592	0.937461	0.723847	0.557057	0.937461
SqueezeNet	RMSPROP	0.851271	0.73668	0.903348	0.674936	0.507272	0.903348
ResNet 18	SGDM	0.887213	0.788081	0.941251	0.731336	0.564822	0.941251
ResNet 18	ADAM	0.917861	0.819672	0.961466	0.767143	0.604688	0.961466
ResNet 18	RMSPROP	0.917332	0.826673	0.957044	0.778376	0.615353	0.957044
ResNet 50	SGDM	0.897447	0.8014	0.941251	0.74959	0.582031	0.941251
ResNet 50	ADAM	0.923228	0.82821	0.972205	0.774865	0.615354	0.972205
ResNet 50	RMSPROP	0.917292	0.820867	0.962097	0.768547	0.60629	0.962097
AlexNet (ImageNet)	SGDM	0.900825	0.790642	0.934302	0.737421	0.568627	0.934302
AlexNet (ImageNet)	ADAM	0.918703	0.820014	0.940619	0.775333	0.608003	0.940619
AlexNet (ImageNet)	RMSPROP	0.914446	0.817623	0.948831	0.769015	0.603455	0.948831
AlexNet (weights none)	SGDM	0.894413	0.790642	0.902716	0.749122	0.571371	0.902716
AlexNet (weights none)	ADAM	0.89092	0.783299	0.92988	0.728996	0.559696	0.92988
AlexNet (weights none)	RMSPROP	0.900311	0.791496	0.925458	0.741868	0.570483	0.925458

Table 10

Data Distribution in Five Folds after Applying Five-Fold Cross-Validation (Second Phase-First Network).

	Train Normal	Test Normal	Test Pneumonia
	Pneumonia		
Fold 1	1266	3418	317
Fold 2	1266	3419	317
Fold 3	1266	3419	317
Fold 4	1267	3418	316
Fold 5	1267	3418	316

the ADAM optimizer achieved the highest accuracy of 96.34% in this phase. Moreover, this model had a sensitivity of 0.969585, a specificity of 0.958857, a precision of 0.945503, a recall of 0.969585, and an AUC of 0.991043.

Due to time constraints, the performance of the Resnet50 model was not evaluated with the SGDM and ADAM optimizer.

5.3. Third stage

Table 18 presents the distribution of DB4 images across five-fold cross-validation. The number of images for each fold dataset is presented. **Table 19** displays the performance metrics for the models in the

third stage. The first column shows the name of the model, and the second column indicates the optimizer used, which can be ADAM, SGDM, or RMSPROP. The remaining columns exhibit various evaluation metrics, such as AUC, accuracy, sensitivity, specificity, precision, and recall. The Darknet19 model trained with the RMSPROP optimizer achieved the highest accuracy of 97.88% in this stage. Moreover, this model had a sensitivity of 0.988944, a specificity of 0.958127, a precision of 0.979916, a recall of 0.988944, and an AUC of 0.997139.

Two custom models are proposed for the third stage of our study. **Figs. 4 and 5** illustrate the architecture of the models, which are designed to improve accuracy and reduce overfitting. The first model

Table 12

Data Distribution in Five Folds after Applying Five-Fold Cross-Validation (Second Phase-Second Network).

	Train		Test	
	Bacteria	Viral	Bacteria	Viral
Fold 1	2224	1194	556	299
Fold 2	2224	1194	556	299
Fold 3	2224	1194	556	299
Fold 4	2224	1195	556	298
Fold 5	2224	1195	556	298

Table 11

Performance Metrics for Second Approach Models (Second Phase-First Network).

Model Name	Training Function	AUC	Accuracy	Sensitivity	Specificity	Precision	Recall
GoogleNet	SGDM	0.983257	0.947404	0.906507	0.962556	0.899687	0.906507
GoogleNet	ADAM	0.993138	0.967725	0.953885	0.972853	0.928659	0.953885
GoogleNet	RMSPROP	0.993163	0.966701	0.946304	0.974257	0.931592	0.946304
SqueezeNet	SGDM	0.978454	0.93972	0.83765	0.977533	0.932489	0.83765
SqueezeNet	ADAM	0.990987	0.961407	0.943146	0.968172	0.916513	0.943146
SqueezeNet	RMSPROP	0.986091	0.95304	0.935565	0.959513	0.895405	0.935565
ResNet 18	SGDM	0.98384	0.946209	0.924826	0.954131	0.881928	0.924826
ResNet 18	ADAM	0.995093	0.97541	0.952622	0.983852	0.956246	0.952622
ResNet 18	RMSPROP	0.995127	0.974898	0.9482	0.984788	0.958493	0.9482
ResNet 50	SGDM	0.986805	0.950307	0.922931	0.960449	0.896319	0.922931
ResNet 50	ADAM	0.995557	0.976434	0.95957	0.982682	0.953547	0.95957
ResNet 50	RMSPROP	0.995621	0.976776	0.969046	0.97964	0.946329	0.969046
AlexNet (ImageNet)	SGDM	0.994398	0.967896	0.952622	0.973555	0.93029	0.952622
AlexNet (ImageNet)	ADAM	0.993491	0.967555	0.931775	0.98081	0.947335	0.931775
AlexNet (ImageNet)	RMSPROP	0.994893	0.972165	0.936829	0.985256	0.95925	0.936829
AlexNet (weights none)	SGDM	0.986681	0.952698	0.922931	0.963726	0.904084	0.922931
AlexNet (weights none)	ADAM	0.988239	0.957992	0.888819	0.983618	0.952607	0.888819
AlexNet (weights none)	RMSPROP	0.988978	0.960041	0.924826	0.973087	0.927169	0.924826
GoogleNet (Place365)	SGDM	0.982756	0.936646	0.92988	0.939153	0.849885	0.92988
GoogleNet (Place365)	ADAM	0.992627	0.964652	0.921036	0.98081	0.946753	0.921036
GoogleNet (Place365)	RMSPROP	0.992303	0.96653	0.936829	0.977533	0.939202	0.936829

Table 13

Performance Metrics for Second Approach Models (Second Phase-Second Network).

Model Name	Training Function	AUC	Accuracy	Sensitivity	Specificity	Precision	Recall
GoogleNet	SGDM	0.79819	0.768547	0.904676	0.51507	0.776474	0.904676
GoogleNet	ADAM	0.837577	0.79429	0.891007	0.6142	0.811333	0.891007
GoogleNet	RMSPROP	0.83543	0.79312	0.889209	0.6142	0.811024	0.889209
SqueezeNet	SGDM	0.796363	0.76059	0.903957	0.493637	0.768737	0.903957
SqueezeNet	ADAM	0.779984	0.74491	0.838849	0.569993	0.784129	0.838849
SqueezeNet	RMSPROP	0.71968	0.710508	0.823022	0.501005	0.754369	0.823022
ResNet 18	SGDM	0.800695	0.76059	0.855036	0.584729	0.793126	0.855036
ResNet 18	ADAM	0.849982	0.788439	0.849281	0.675151	0.829585	0.849281
ResNet 18	RMSPROP	0.842079	0.785865	0.847482	0.671132	0.827538	0.847482
ResNet 50	SGDM	—	—	—	—	—	—
ResNet 50	ADAM	0.845133	0.784461	0.727395	0.815108	0.67875	0.727395
ResNet 50	RMSPROP	0.83876	0.789609	0.707971	0.833453	0.695395	0.707971
AlexNet (ImageNet)	SGDM	0.837799	0.778142	0.635633	0.854676	0.701404	0.635633
AlexNet (ImageNet)	ADAM	0.84523	0.79546	0.681179	0.856835	0.718728	0.681179
AlexNet (ImageNet)	RMSPROP	0.845232	0.794056	0.656397	0.867986	0.727543	0.656397
AlexNet (weights none)	SGDM	0.772295	0.732272	0.823741	0.561956	0.777853	0.823741
AlexNet (weights none)	ADAM	0.815637	0.767143	0.895683	0.527796	0.779343	0.895683
AlexNet (weights none)	RMSPROP	0.819686	0.76527	0.806475	0.688547	0.828223	0.806475

Table 14

Data Distribution in Five Folds after Applying Five-Fold Cross-Validation (Third Phase-Second Network).

	Train		Test	
	Bacteria	Viral	Bacteria	Viral
Fold 1	2224	2224	556	556
Fold 2	2224	2224	556	556
Fold 3	2224	2224	556	556
Fold 4	2224	2224	556	556
Fold 5	2224	2224	556	556

comprises five layers, starting with a convolution layer to extract features from the input images, followed by batch normalization to standardize the feature maps, ReLU activation to introduce nonlinearity, fully connected layers to learn the classification task, and a SoftMax output layer to produce the class probabilities. The second model builds on the first model and includes a dropout layer, which randomly drops out some neurons during training to prevent overfitting.

Our custom DL models were trained and tested without using any image augmentation technique. This approach enabled us to evaluate the models' performance using only the original dataset, without any artificially generated data. Additionally, Figs. 4 and 5 do not show the

image input layer and classification layer, which are essential components of the models. The results of our experiments show the first model achieved an accuracy rate of 98.32% with the SGDM optimizer, whereas the second model attained a higher accuracy of 99.45% with the same optimizer. Table 19 shows the DL models exhibit improved performance when trained and tested without any input image augmentation.

The proposed scenario in this paper is presented in Fig. 6. The X-ray image is processed through three different networks for classification. First, the first TL model or classifier classifies the image as either normal or pneumonia. If the output is normal, the system prints "normal" and exits. If the image is classified as pneumonia, it is passed on to the second

Table 16

Data Distribution in Five Folds after Applying Five-Fold Cross-Validation (Second Stage).

	Train		Test	
	Covid	Non-Covid	Covid	Non-Covid
Fold 1	3235	4394	809	1099
Fold 2	3235	4394	809	1099
Fold 3	3235	4395	809	1098
Fold 4	3235	4395	809	1098
Fold 5	3236	4394	808	1099

Table 15

Performance Metrics for Third Phase Models (Third Phase-Second Network).

Model Name	Training Function	AUC	Accuracy	Sensitivity	Specificity	Precision	Recall
GoogleNet	SGDM	0.880597	0.811511	0.869424	0.753597	0.779175	0.869424
GoogleNet	ADAM	0.910868	0.840468	0.882014	0.798921	0.814347	0.882014
GoogleNet	RMSPROP	0.910536	0.836691	0.891367	0.782014	0.803502	0.891367
SqueezeNet	SGDM	0.867754	0.799101	0.85	0.748201	0.771466	0.85
SqueezeNet	ADAM	0.888179	0.808453	0.872662	0.744245	0.77335	0.872662
SqueezeNet	RMSPROP	0.856027	0.769245	0.77554	0.76295	0.765897	0.77554
Resnet 18	SGDM	0.883808	0.817626	0.856475	0.778777	0.794726	0.856475
practical	ADAM	0.912524	0.83795	0.840647	0.835252	0.836136	0.840647
Resnet 18	RMSPROP	0.91269	0.835971	0.853957	0.817986	0.824306	0.853957
Resnet 50	SGDM	—	—	—	—	—	—
Resnet 50	ADAM	—	—	—	—	—	—
Resnet 50	RMSPROP	0.907689	0.817086	0.870863	0.763309	0.786294	0.870863
ShuffleNet	ADAM	0.9190	0.8317	0.8335	0.8299	0.8305	0.8335
ShuffleNet	RMSPROP	0.9148	0.8330	0.8565	0.8094	0.8180	0.8565
ShuffleNet	SGDM	0.8993	0.8245	0.8600	0.7889	0.8029	0.8600
AlexNet (ImageNet)	SGDM	0.90987	0.829317	0.826259	0.832374	0.831343	0.826259
AlexNet (ImageNet)	ADAM	0.919824	0.840108	0.810791	0.869424	0.861292	0.810791
AlexNet (ImageNet)	RMSPROP	0.916312	0.835971	0.77446	0.897482	0.883101	0.77446
AlexNet (weights none)	SGDM	0.891779	0.807374	0.9	0.714748	0.759332	0.9
AlexNet (weights none)	ADAM	0.895513	0.816367	0.892446	0.740288	0.774586	0.892446
AlexNet (weights none)	RMSPROP	0.905389	0.827338	0.844964	0.809712	0.816192	0.844964

Table 17

Performance Metrics for Second-Stage Models.

Model Name	Training Function	AUC	Accuracy	Sensitivity	Specificity	Precision	Recall
GoogleNet	SGDM	0.948554	0.87711	0.900841	0.85964	0.825329	0.900841
GoogleNet	ADAM	0.983723	0.93866	0.971068	0.914801	0.893515	0.971068
GoogleNet	RMSPROP	0.983171	0.940128	0.969337	0.918624	0.897641	0.969337
SqueezeNet	SGDM	0.947802	0.872182	0.911474	0.843255	0.810644	0.911474
SqueezeNet	ADAM	0.964202	0.903848	0.886746	0.916439	0.886527	0.886746
SqueezeNet	RMSPROP	0.959569	0.889693	0.887488	0.891316	0.857382	0.887488
Resnet18	SGDM	0.964074	0.906784	0.918645	0.898052	0.869006	0.918645
Resnet18	ADAM	0.991043	0.963406	0.969585	0.958857	0.945503	0.969585
Resnet18	RMSPROP	0.989594	0.961938	0.970821	0.955398	0.941261	0.970821
Resnet 50	SGDM	–	–	–	–	–	–
Resnet 50	ADAM	–	–	–	–	–	–
Resnet 50	RMSPROP	0.990219	0.962147	0.967854	0.957946	0.94427	0.967854
ShuffleNet	ADAM	0.9824	0.9279	0.9701	0.8970	0.8740	0.9700
ShuffleNet	RMSPROP	0.9819	0.9287	0.9667	0.9008	0.8777	0.9667
ShuffleNet	SGDM	0.9650	0.9012	0.9518	0.8639	0.8373	0.9518
AlexNet (ImageNet)	SGDM	0.977698	0.919472	0.969585	0.882578	0.858739	0.969585
AlexNet (ImageNet)	ADAM	0.983577	0.938975	0.964144	0.920444	0.899216	0.964144
AlexNet (ImageNet)	RMSPROP	0.985873	0.93866	0.967112	0.917713	0.896402	0.967112
AlexNet (weights none)	SGDM	0.915833	0.841145	0.849901	0.834699	0.791024	0.849901
AlexNet (weights none)	ADAM	0.952055	0.873021	0.821958	0.910614	0.871298	0.821958
AlexNet (weights none)	RMSPROP	0.956786	0.885289	0.9273	0.85436	0.824176	0.9273

Table 18

Data Distribution in Five Folds after Applying Five-Fold Cross-Validation (Third Stage).

	Train		Test	
	Covid	Non-Covid	Covid	Non-Covid
Fold 1	4341	2102	1086	525
Fold 2	4341	2102	1086	525
Fold 3	4342	2101	1085	526
Fold 4	4342	2101	1085	526
Fold 5	4342	2101	1085	526

classifier for further classification into viral or bacterial. If the image is classified as bacterial, the system prints “bacteria” and exits. If the image is classified as viral, it is forwarded to the third classifier, which is used to classify the image as either COVID or non-COVID. If the image is classified as non-COVID, the system prints “non-COVID” and exits. If the image is classified as COVID, the system asks the user for a CT scan of the patient to confirm if the patient has COVID, using the fourth classifier. Table 20 summarizes the results for all three stages and three phases. Stages 1 phase 2 and 3 have two networks each, and the first network is identical in both phases.

The results obtained in our paper are promising and have remarkable implications for clinical applications. Our proposed approach achieved an average accuracy of 90.7% with the first stage-third phase (contains

Table 19

Performance Metrics for Third-Stage Models (Third Stage).

Model Name	Training Function	AUC	Accuracy	Sensitivity	Specificity	Precision	Recall
GoogleNet	SGDM	0.948403	0.874224	0.932928	0.75295	0.88638	0.932928
GoogleNet	ADAM	0.987395	0.944251	0.963147	0.905215	0.954529	0.963147
GoogleNet	RMSPROP	0.988533	0.949839	0.962226	0.924248	0.963291	0.962226
SqueezeNet	SGDM	0.93863	0.848647	0.952091	0.634945	0.843454	0.952091
SqueezeNet	ADAM	0.969948	0.900919	0.899208	0.904454	0.951082	0.899208
SqueezeNet	RMSPROP	0.966307	0.901788	0.93864	0.825657	0.917507	0.93864
ResNet 18	SGDM	0.959779	0.89409	0.930901	0.818043	0.913562	0.930901
ResNet 18	ADAM	0.996037	0.971443	0.980652	0.952417	0.977052	0.980652
ResNet 18	RMSPROP	0.996019	0.972436	0.9801	0.956604	0.979017	0.9801
ResNet 50	SGDM	0.968745	0.911969	0.936429	0.861439	0.933162	0.936429
ResNet 50	ADAM	0.997442	0.980631	0.986549	0.968405	0.984734	0.986549
ResNet 50	RMSPROP	0.9955	0.976285	0.988944	0.950133	0.976173	0.988944
AlexNet (ImageNet)	SGDM	0.985793	0.942637	0.97236	0.881233	0.944176	0.97236
AlexNet (ImageNet)	ADAM	0.993029	0.962379	0.98489	0.915874	0.960295	0.98489
AlexNet (ImageNet)	RMSPROP	0.992189	0.963372	0.983232	0.922345	0.963177	0.983232
GoogleNet (Place365)	SGDM	0.959871	0.892972	0.934771	0.806624	0.908977	0.934771
GoogleNet (Place365)	ADAM	0.990058	0.948597	0.957988	0.929197	0.96546	0.957988
GoogleNet (Place365)	RMSPROP	0.990001	0.950956	0.963516	0.92501	0.963693	0.963516
AlexNet (weight none)	SGDM	0.966938	0.910107	0.945458	0.837077	0.923008	0.945458
AlexNet (weight none)	ADAM	0.979185	0.929104	0.957435	0.870575	0.938584	0.957435
AlexNet (weight none)	RMSPROP	0.977937	0.926248	0.962963	0.8504	0.930059	0.962963
DarkNet 19	SGDM	0.990705	0.954805	0.972176	0.918919	0.961195	0.972176
DarkNet 19	ADAM	0.996897	0.975788	0.984338	0.958127	0.979824	0.984338
DarkNet 19	RMSPROP	0.997139	0.978892	0.988944	0.958127	0.979916	0.988944
ShuffleNet	ADAM	0.9954	0.969	0.9877	0.9304	0.967	0.9877
ShuffleNet	RMSPROP	0.9961	0.9709	0.9829	0.946	0.9741	0.9829
ShuffleNet	SGDM	0.9802	0.9288	0.9481	0.8889	0.9463	0.9481
Custom model 1(no augmentation)	SGDM	0.998097	0.983238	0.993551	0.961934	0.981792	0.993551
Custom model 1(no augmentation)	ADAM	0.997171	0.977775	0.995578	0.940997	0.972112	0.995578
Custom model 1(no augmentation)	RMSPROP	0.986612	0.961013	0.969596	0.943281	0.972464	0.969596
Custom model 2(no augmentation)	SGDM	0.999846	0.994537	0.994841	0.993909	0.997045	0.994841

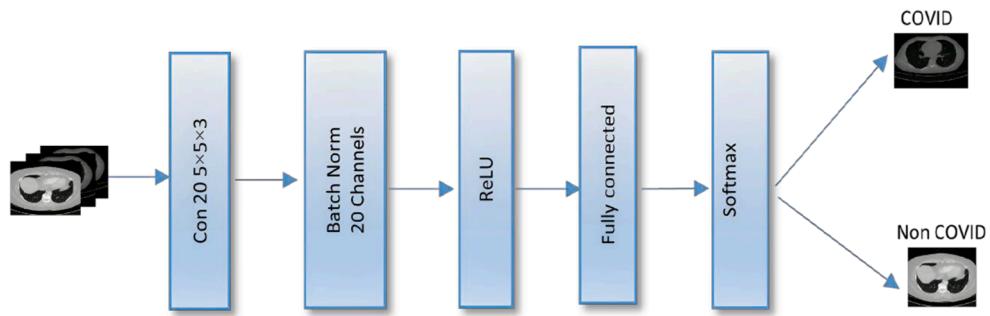


Fig. 4. The Architecture of Custom Model 1.

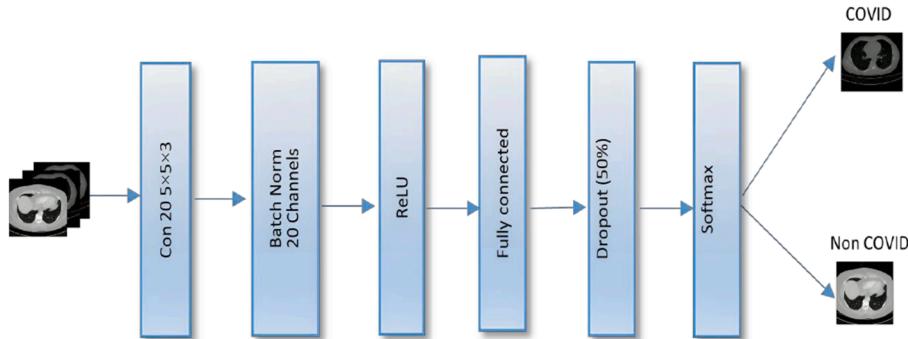


Fig. 5. The Architecture of Custom Model 2.

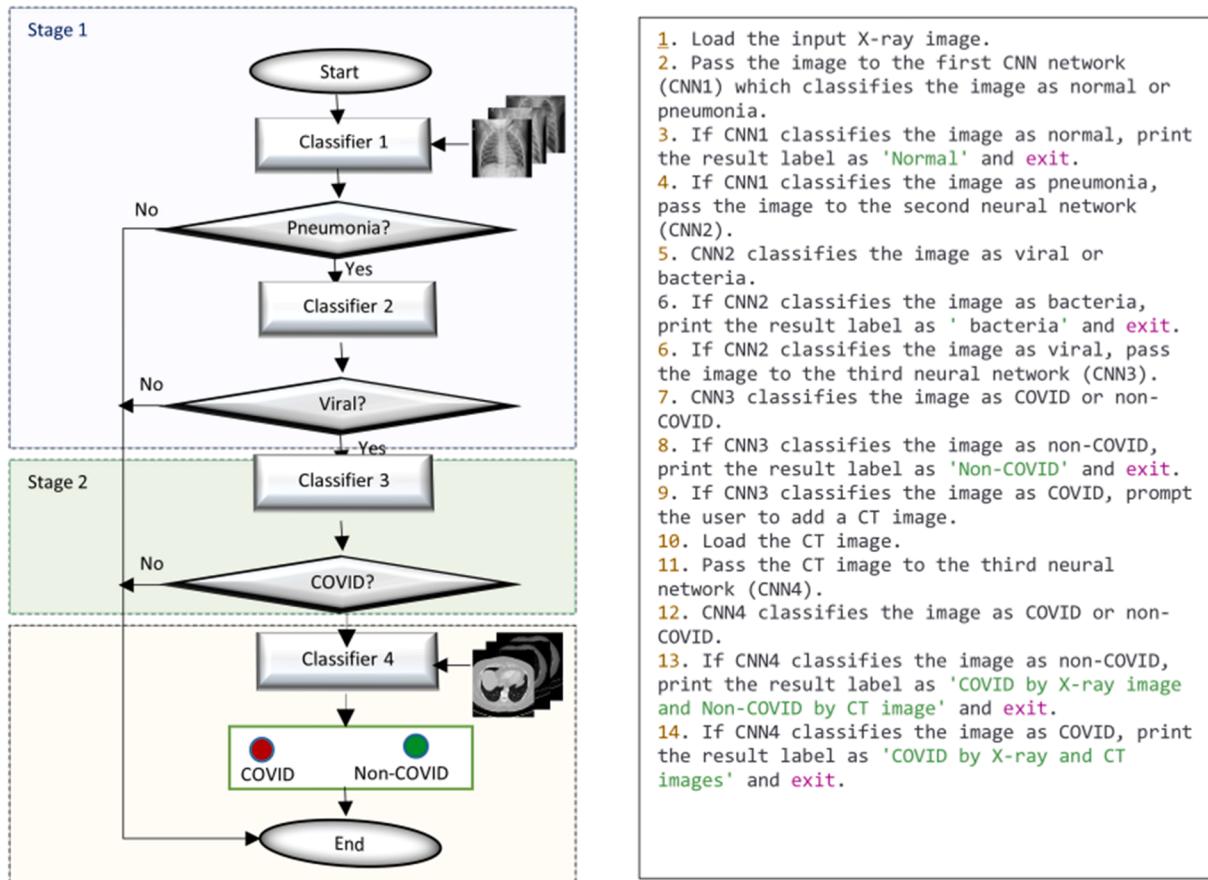


Fig. 6. Proposed Scenario Flowchart and Pseudo Code.

Table 20
Result Summary.

Stage name	Best network model	Optimizer used	Accuracy
First Stage-First phase	ResNet50	ADAM	82.82%
First Stage-Second phase-network 1	ResNet50	RMSPROP	97.68%
First Stage-Second phase-network 2	AlexNet	ADAM	79.55%
First Stage-Third phase-network 2	GoogleNet	ADAM	84.05%
Second Stage	ResNet18	ADAM	96.34%
Third Stage	DarkNet19	RMSPROP	97.89%

two cascaded networks), which is a substantial improvement over the first phase (contains only a single network) accuracy of 82.82%. The average accuracy of the second phase (containing two cascaded networks) was 88.5%.

The second stage achieved an impressive accuracy of 96.34% with ResNet18 and the Adam optimizer. The third stage achieved the highest accuracy of 97.88% with DarkNet19 and the RMSprop optimizer. These results demonstrate the effectiveness of our proposed approach in accurately diagnosing COVID-19 using CXR and CT images.

The potential clinical applications of our approach are considerable. The rapid, accurate diagnosis of COVID-19 is critical for the effective treatment and management of the disease. Our approach could be integrated into the existing diagnostic workflow to provide a reliable, efficient method for COVID-19 diagnosis. Furthermore, our approach could be adapted to other infectious diseases and medical conditions, providing a valuable tool for healthcare professionals in the diagnosis and management of various diseases.

The results of our paper have remarkable implications for clinical

practice because they provide a rapid, accurate diagnosis of COVID-19 in patients. This outcome is critical for effective treatment and management of the disease and can substantially reduce the workload of healthcare professionals. By automating the diagnosis and providing accurate results promptly, our approach can help in the early detection of the disease, which is crucial for preventing its spread. These findings can improve patient outcomes and contribute to the overall effort to control the COVID-19 pandemic.

6. Heat map visualization

The performance of the top pretrained model was evaluated in the second and third stages using various visualization tools such as Grad-CAM, Occlusion Specificity, and Local Interpretable Model-Agnostic Explanation (Lime) to understand how well the model can identify the regions of interest and distinguish between Covid and Non-Covid classes for X-ray and CT chest images. This approach enabled us to understand better the models' decision making and validate their performance during the training and validation phases.

Grad-CAM is a popular visualization method that identifies key areas in an image that are important for predicting the target classes. It works by examining the model's last convolutional layer and analyzing the gradient data fed into that layer. Grad-CAM generates a heatmap visualization that highlights the relevant portion of the image the model is focusing on for a particular class label [51]. The results in Figs. 7 and 8 demonstrate the occlusion specificity technique is more accurate and deterministic compared with the results obtained from Grad-CAM and Lime, which are considered the worst for analyzing X-ray and CT images.

Another related technique is occlusion specificity, which involves masking a portion of the input image and observing the corresponding

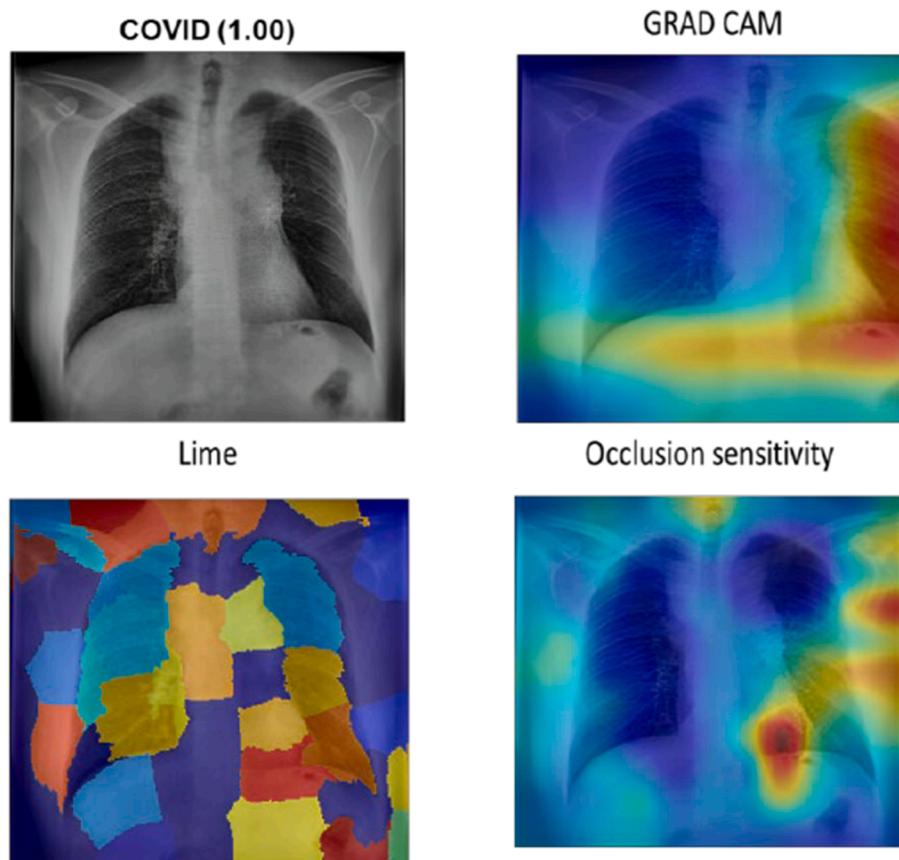


Fig. 7. Heat Map Visualization of DB3 X-ray Images Using the Best Pretrained Model (ResNet18 with ADAM Optimizer).

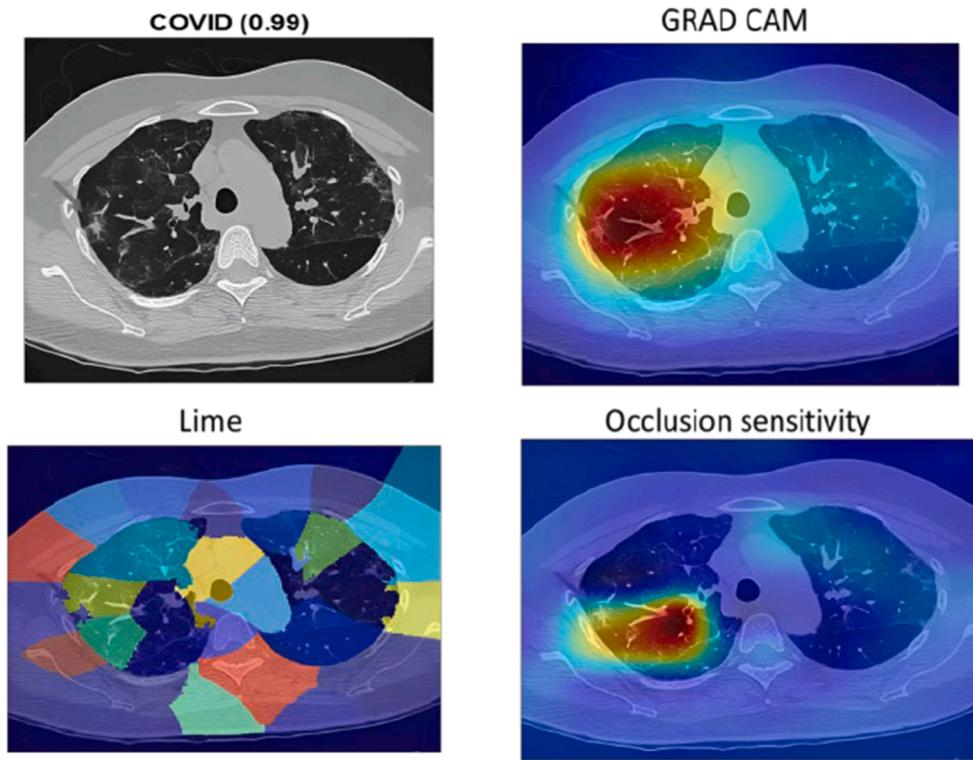


Fig. 8. Heat Map Visualization of DB4 CT Images Using the Best Pretrained Model (Darknet19 with RMSprop Optimizer).

change in the model's output. This method is useful for identifying which parts of an image the model is using to make its prediction and can be used to diagnose any weaknesses or biases in the model's performance [52].

Lime heat map visualization is a widely used technique in the field of machine learning explainability. It helps understand the decision making of complex black-box models by highlighting the most important features that influence the model's output for a particular instance. The Lime algorithm generates local surrogate models around the instance of interest and evaluates the importance of each feature based on its contribution to the final prediction. By creating a heat map, Lime provides a visual representation of the model's decision making, making interpreting and understanding easier for humans. This interpretability is critical in applications where transparency and accountability are essential, such as healthcare or finance [53].

7. Comparison with state-of-the-art techniques

[Table 21](#) comprehensively compares the proposed approach with state-of-the-art techniques for the diagnosis of COVID-19 in CXR and CT images. The table offers a detailed overview of the accuracy achieved by each research study, the algorithm or model used, the data splitting technique employed, and the database used. This information is valuable for researchers and practitioners to evaluate the effectiveness of the proposed technique in comparison to other methods.

8. Conclusion

This paper presents an automated, efficient approach for the rapid, accurate diagnosis of COVID-19 using CXR and CT images. The proposed technique leverages transfer learning with CCNs and explore various regularization, optimization, dropout, and data augmentation techniques. Through comprehensive experiments with three optimizers and

Table 21

Comparison with State-of-the-art Techniques.

Ref	Year	Scan type	Algorithm	Data splitting	DB name	Accuracy
Kaggle [54]	2020	X-ray	Custom CNN model	Train /validation/ test	Kaggle dataset	92.62%
[our] [24]	2021	X-ray	ResNet50 with RMSprop optimizer Transfer Learning	Cross-validation with 5-folds 75% training, 15% validation, and 10% testing data	Mendeley and Kaggle Datasets	97.68% 92% (multiclass classification)
[40]	2022		CoviXNet, a deep-learning neural network architecture	70% training, 10% validation, and 20% testing data	Mendeley dataset	96.23%
[41]	2021		SVM and a Stacking Classifier	70% of the dataset for training and the rest 30% as a testing set		94.0%
[our] [26]	2021	CT scan	ResNet18 with ADAM optimizer pre-trained deep learning CNN architectures	Cross-validation with 5-folds 80% in training set and 20% in the testing set	Mendeley dataset	96.34% 97.7%
[41]	2021		SVM and a Stacking Classifier	70% of the dataset for training and the rest 30% as a testing set		97.3%
[our]			DarkNet19 with RMSprop optimizer	Cross-validation with 5-folds		97.889%

comparisons with state-of-the-art techniques, our approach demonstrates highly promising performance metrics. The utilization of the proposed CCN approach, along with transfer learning, exhibits promising results in swiftly and accurately diagnosing COVID-19 using CXR and CT images. The paper is divided into three stages, each with different datasets and models, and evaluated using various metrics. Our findings reveal the use of transfer learning models with pretrained weights on large datasets such as ImageNet improves the accuracy of our models.

The second and third phases of the first stage, which utilize two-stage cascaded networks, achieved higher accuracy than the first phase. Specifically, the second phase achieved an average accuracy of 88.5%, whereas the third phase achieved an average accuracy of 90.7%. Additionally, the second stage achieved an accuracy of 96.34% with DB3, and the third stage achieved an accuracy of 97.88% with DB4 (CT images dataset). This proposed method can aid in the early detection and treatment of COVID-19, providing a substantial contribution to clinical practice by automating the diagnosis and providing accurate results promptly, which can considerably reduce the workload of healthcare professionals. The paper highlights the potential of transfer learning and cascaded network approaches for improving the accuracy of medical image analysis. Furthermore, the incorporation of occlusion specificity and Grad-CAM techniques enhances our understanding of the network output. The proposed technique offers a highly adaptable, scalable solution, eliminating the need for manual hyperparameter tuning. This system can play a critical role in early screening and diagnosing COVID-19, particularly in regions with limited testing resources and time constraints, ultimately contributing to better management of the pandemic's health and economic effect.

Future work can explore the application of our proposed method on larger and more diverse datasets and evaluate its performance against other state-of-the-art methods.

Limitations of the study and suggestions for future research.

While our proposed approach demonstrated exceptional accuracy in diagnosing COVID-19 within Chest X-ray (CXR) and Computed Tomography (CT) images, it is essential to acknowledge several noteworthy limitations within our research. Firstly, the datasets we employed in our study, comprising a range of 5,000 to 10,000 images, fall within the category of medium-sized datasets. This dimensionality, although substantial, does introduce potential constraints on the extrapolation of our findings to larger and more diverse datasets. Future investigations should prioritize the utilization of expansive datasets to further validate the robustness and reliability of our diagnostic approach.

Secondly, our study regrettably did not account for the potential impact of imaging artifacts on the accuracy of our diagnostic method. These artifacts, encompassing issues like motion blur or image distortion, are not uncommon in clinical practice. Future research endeavors must delve into the intricate influence of such artifacts on the effectiveness of COVID-19 diagnosis via imaging techniques, thereby enhancing the real-world applicability of our approach.

Thirdly, our research exclusively relied on transfer learning models as the cornerstone of our approach. Although these models yielded commendable results, we did not undertake a comparative evaluation against alternative Deep Learning (DL) techniques, such as unsupervised learning or reinforcement learning. Therefore, there remains unexplored potential in assessing the performance and efficacy of various DL methodologies in the context of COVID-19 diagnosis. This avenue could uncover novel strategies to enhance diagnostic accuracy.

Fourthly, it's important to underscore that our study concentrated primarily on the diagnosis of COVID-19 in CXR and CT images. While this served as an excellent starting point, future studies should venture into broader horizons. Exploring the versatility of our approach in different medical imaging applications and its synergy with complementary diagnostic tools, such as laboratory tests, could potentially unlock a more comprehensive and integrated approach to disease

diagnosis, thereby advancing the frontiers of medical science.

Lastly, our future endeavors for improving Deep Learning (DL) models, one can explore alternative optimization techniques such as the BFGS algorithm, Hill-climbing algorithm, and Simulated annealing. Additionally, when it comes to fine-tuning the model's parameters for optimal performance, avenues like genetic algorithms, grid search, and random search can be investigated. Furthermore, AutoML often incorporates ensemble methods to combine multiple DL models, enhancing predictive accuracy and robustness. Some popular AutoML platforms for DL include Google Cloud AutoML, H2O AutoML, and Azure Machine Learning's automated ML capabilities.

While our proposed approach shows promise in accurately diagnosing COVID-19 in CXR and CT images, several limitations need to be addressed in future research to improve the generalizability and accuracy.

CRediT authorship contribution statement

Amr E. Eldin Rashed: Conceptualization, Methodology, Software, Visualization, Investigation, Validation. **Waleed M. Bahgat:** Data curation, Writing – original draft, Supervision, Writing – review & editing.

Declaration of Competing Interest

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

Data availability

<https://github.com/amrrashed/-Diagnosis-of-COVID-19-in-Chest-X-Ray-and-CT-Images-using-Transfer-Learning/tree/main/database>

Acknowledgments

The researchers acknowledge the Deanship of Scientific Research, Taif University for funding this work.

References

- [1] "Weekly epidemiological update on COVID-19 - 13 April 2023." <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19—13-april-2023> (accessed Apr. 21, 2023).
- [2] "Symptoms of COVID-19 | CDC." <https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html> (accessed Apr. 21, 2023).
- [3] T. Ai, et al., Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases, Radiology 296 (2) (Aug. 2020) E32–E40, <https://doi.org/10.1148/RADIOL.2020200642>.
- [4] H.Y.F. Wong, et al., Frequency and Distribution of Chest Radiographic Findings in Patients Positive for COVID-19, Radiology 296 (2) (Aug. 2020) E72–E78, <https://doi.org/10.1148/RADIOL.2020201160>.
- [5] Y. Fang, et al., Sensitivity of chest CT for COVID-19: Comparison to RT-PCR, Radiology 296 (2) (Aug. 2020) E115–E117, <https://doi.org/10.1148/RADIOL.2020200432>.
- [6] A. Nalumansi, T. Lutalo, J. Kayiwa, ... C. W.-I. J. of, and undefined 2021, "Field evaluation of the performance of a SARS-CoV-2 antigen rapid diagnostic test in Uganda using nasopharyngeal samples," Elsevier, Accessed: Apr. 21, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S120197122032275X>.
- [7] S. Wang, et al., A deep learning algorithm using CT images to screen for Coronavirus disease (COVID-19), Eur. Radiol. 31 (8) (Aug. 2021) 6096–6104, <https://doi.org/10.1007/S00330-021-07715-1>.
- [8] C. Jin, et al., "Development and evaluation of an AI system for COVID-19," 2020, Accessed: Apr. 21, 2023. [Online]. Available: <https://pesquisa.bvsalud.org/portal/resource/pt/ppmedrxiv-20039834>.
- [9] A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, Pattern Anal. Appl. 24 (3) (Aug. 2021) 1207–1220, <https://doi.org/10.1007/S10044-021-00984-Y>.
- [10] C. Szegedy, et al., "Going deeper with convolutions."
- [11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB

- model size," Feb. 2016, Accessed: Apr. 21, 2023. [Online]. Available: <https://arxiv.org/abs/1602.07360v4>.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," pp. 770–778, 2016, Accessed: Apr. 21, 2023. [Online]. Available: <http://image-net.org/challenges/LSVRC/2015/>.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Accessed: Apr. 21, 2023. [Online]. Available: <http://code.google.com/p/cuda-convnet/>.
- [14] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-Decem (2016) 779–788, <https://doi.org/10.1109/CVPR.2016.91>.
- [15] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 6848–6856, Dec. 2018, doi: 10.1109/CVPR.2018.00716.
- [16] T. Zhou, F. Liu, H. Lu, C. Peng, X. Y.- Electronics, and undefined 2023, "A Review of Deep Learning Imaging Diagnostic Methods for COVID-19," *mdpi.com*, Accessed: Apr. 21, 2023. [Online]. Available: <https://www.mdpi.com/2079-9292/12/5/1167>.
- [17] B. Yan, et al., Experiments of Federated Learning for COVID-19 Chest X-ray Images, Commun. Comput. Inform. Sci. 1423 (2021) 41–53, https://doi.org/10.1007/978-3-030-78618-2_4.
- [18] M. R. Karim, D. Rebholz-Schuhmann, S. Decker, M. Cochez, and O. Beyan, "DeepCOVIDExplainer: Explainable COVID-19 diagnosis based on chest X-ray images," *arxiv.org*, Accessed: Aug. 04, 2023. [Online]. Available: <https://arxiv.org/abs/2004.04582>.
- [19] M. Farooq and A. Hafeez, "COVID-ResNet: A Deep Learning Framework for Screening of COVID-19 from Radiographs," Mar. 2020, Accessed: Apr. 21, 2023. [Online]. Available: <http://arxiv.org/abs/2003.14395>.
- [20] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images," *arxiv.org*, 2020, Accessed: Apr. 21, 2023. [Online]. Available: <https://arxiv.org/abs/2003.11055>.
- [21] G. Barghady, X. Zhou, P. Barua, ... R. G.-P. R., and undefined 2022, "Application of CycleGAN and transfer learning techniques for automated detection of COVID-19 using X-ray images," *Elsevier*, Accessed: Apr. 21, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167865521004128>.
- [22] T. Ozturk, M. Talo, E. Yildirim, ... U. B.-C. in biology, and undefined 2020, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Elsevier*, Accessed: Apr. 21, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482520301621>.
- [23] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. 43 (2) (Jun. 2020) 635–640, <https://doi.org/10.1007/S13246-020-00865-4>.
- [24] D. Shome, et al., "Covid-transformer: Interpretable covid-19 detection using vision transformer for healthcare," *mdpi.com*, 2021, doi 10.3390/jerph182111086.
- [25] W. Bahgat, H. Balaha, ... Y. A.-P. C., and undefined 2021, "An optimized transfer learning-based approach for automatic diagnosis of COVID-19 from chest x-ray images," *peerj.com*, Accessed: Apr. 21, 2023. [Online]. Available: <https://peerj.com/articles/cs-555/?ref=https://github.com/>.
- [26] M. Elpelty, H. Sallam, Automatic prediction of COVID-19 from chest images using modified ResNet50, Multimed. Tools Appl. 80 (17) (Jul. 2021) 26451–26463, <https://doi.org/10.1007/S11042-021-10783-6>.
- [27] N. Kumar, M. Gupta, D. Gupta, S. Tiwari, Novel deep transfer learning model for COVID-19 patient detection using X-ray chest images, J. Ambient Intell. Humaniz. Comput. 14 (1) (Jan. 2023) 469–478, <https://doi.org/10.1007/S12652-021-03006-6>.
- [28] A.S.K. Reddy, et al., Multi-modal fusion of deep transfer learning based COVID-19 diagnosis and classification using chest x-ray images, Multimed. Tools Appl. 82 (8) (Mar. 2023) 12653–12677, <https://doi.org/10.1007/S11042-022-13739-6>.
- [29] S.D. Deb, R.K. Jha, R. Kumar, P.S. Tripathi, Y. Talera, M. Kumar, CoVSeverity-Net: an efficient deep learning model for COVID-19 severity estimation from Chest X-Ray images, Res. Biomed. Eng. 39 (1) (Mar. 2023) 85–98, <https://doi.org/10.1007/S42600-022-00254-8>.
- [30] A. Das, S. Kalam, C. Kumar, D. S.- Chaos, S. & Fractals, and undefined 2021, "TLCov-An automated Covid-19 screening model using Transfer Learning from chest X-ray images," *Elsevier*, Accessed: Aug. 05, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077921000667>.
- [31] A. Voulodimos, E. Protopapadakis, I. Katsamanis, A. Doulamis, N. Doulamis, A few-shot U-Net deep learning model for COVID-19 infected area segmentation in CT images, Sensors 21 (6) (Mar. 2021) 1–22, <https://doi.org/10.3390/S21062215>.
- [32] L. Li, et al., Artificial Intelligence Distinguishes COVID-19 from Community-Acquired Pneumonia on Chest CT, Radiology 296 (2) (Aug. 2020) E65–E71, <https://doi.org/10.1148/RADIOL.2020200905>.
- [33] P. Gaur, V. Malaviya, A. Gupta, G. Bhatia, R.B. Pachori, D. Sharma, COVID-19 disease identification from chest CT images using empirical wavelet transformation and transfer learning, Biomed. Signal Process. Control 71 (Jan. 2022), <https://doi.org/10.1016/J.BSPC.2021.103076>.
- [34] B. Abraham, M.S. Nair, Computer-aided detection of COVID-19 from CT scans using an ensemble of CNNs and KSVM classifier, Signal, Image Video Process. 16 (3) (Apr. 2022) 587–594, <https://doi.org/10.1007/S11760-021-01991-6/TABLES/6>.
- [35] Y. Pathak, P.K. Shukla, A. Tiwari, S. Stalin, S. Singh, Deep Transfer Learning Based Classification Model for COVID-19 Disease, Ing. Rech. Biomed. 43 (2) (Apr. 2022) 87, <https://doi.org/10.1016/J.IRBMB.2020.05.003>.
- [36] X. Yao, Z. Zhu, C. Kang, S.H. Wang, J.M. Gorri, Y.D. Zhang, AdaD-FNN for Chest CT-Based COVID-19 Diagnosis, IEEE Trans. Emerg. Top. Comput. Intell. 7 (1) (Feb. 2023) 5–14, <https://doi.org/10.1109/TETCI.2022.3174868>.
- [37] R. Kundu, P.K. Singh, M. Ferrara, A. Ahmadian, R. Sarkar, ET-NET: an ensemble of transfer learning models for prediction of COVID-19 infection through chest CT-scan images, Multimed. Tools Appl. 81 (1) (Jan. 2022) 31–50, <https://doi.org/10.1007/S11042-021-11319-8/TABLES/9>.
- [38] N.D. Kathamuthu, et al., A deep transfer learning-based convolution neural network model for COVID-19 detection using computed tomography scan images for medical applications, Adv. Eng. Softw. 175 (Jan. 2023), 103317, <https://doi.org/10.1016/J.ADVENGSOFT.2022.103317>.
- [39] H. Alhares, J. Tanha, M.A. Balafer, AMTLDC: a new adversarial multi-source transfer learning framework to the diagnosis of COVID-19, Evol. Syst. (2023), <https://doi.org/10.1007/S12530-023-09484-2>.
- [40] G. Srivastava, A. Chauhan, ... M. J.... S. P. and, and undefined 2022, "CoviXNet: A novel and efficient deep learning model for detection of COVID-19 using chest X-Ray images," *Elsevier*, Accessed: Apr. 21, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809422003597>.
- [41] P. Yadav, N. Menon, V. Ravi, S. Vishvanathan, Lung-GANs: Unsupervised Representation Learning for Lung Disease Classification Using Chest CT and X-Ray Images, IEEE Trans. Eng. Manag. (2021), <https://doi.org/10.1109/TEM.2021.310334>.
- [42] M. Manav, M. Goyal, A. Kumar, A. Arya, H. Singh, A. Yadav, Deep Learning Approach for Analyzing the COVID-19 Chest X-Rays, J. Med. Phys. 46 (3) (Jul. 2021) 189, https://doi.org/10.4103/JMP.JMP_22_21.
- [43] M.L. Huang, Y.C. Liao, A lightweight CNN-based network on COVID-19 detection using X-ray and CT images, Comput. Biol. Med. 146 (Jul. 2022), <https://doi.org/10.1016/J.COMBIOMED.2022.105604>.
- [44] S.E. Mukhi, R.T. Varshini, S.E.F. Sherley, Diagnosis of COVID-19 from Multimodal Imaging Data Using Optimized Deep Learning Techniques, SN Comput. Sci. 4 (3) (May 2023) 1–9, <https://doi.org/10.1007/S42979-022-01653-5/TABLES/7>.
- [45] A. Hayat, P. Baglat, F. Mendonça, S.S. Mostafa, F. Morgado-Dias, Novel Comparative Study for the Detection of COVID-19 Using CT Scan and Chest X-ray Images, Int. J. Environ. Res. Public Health 20 (2) (Jan. 2023), <https://doi.org/10.3390/IJERPH20021268>.
- [46] S. Roy, A. D.-C. Intelligence, and undefined 2023, "Deep-CoV: An integrated deep learning model to detect COVID-19 using chest X-ray and CT images," Wiley Online Libr. Roy, AK DasComputational Intell. 2023•Wiley Online Libr., vol. 39, no. 2, pp. 369–400, Apr. 2023, doi: 10.1111/coin.12568.
- [47] D. Kermany, K. Zhang, and M. Goldbaum, "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification," vol. 2, 2018, doi: 10.17632/RSCBjBR9SJ.2.
- [48] "Chest X-Ray Images (Pneumonia) | Kaggle." <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia> (accessed Apr. 22, 2023).
- [49] W. El-Shafai and F. E. Abd El-Samie, "Extensive COVID-19 X-Ray and CT Chest Images Dataset," vol. 3, 2020, doi: 10.17632/8fH65YWD2JR.3.
- [50] M. Masud, A.E. Eldin Rashed, M.S. Hossain, Convolutional neural network-based models for diagnosis of breast cancer, Neural Comput. Appl. (2020), <https://doi.org/10.1007/s00521-020-05394-5>.
- [51] M. Masud, et al., Pre-Trained Convolutional Neural Networks for Breast Cancer Detection Using Ultrasound Images, ACM Trans. Internet Technol. 21 (4) (Nov. 2021), <https://doi.org/10.1145/3418355>.
- [52] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013," Comput. Vision-ECCV 2014, vol. 8689, no. PART 1, pp. 818–833, 2014, doi 10.1007/978-3-319-10590-1_53.
- [53] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-August-2016, pp. 1135–1144, Aug. 2016, doi: 10.1145/2939672.2939778.
- [54] "Pneumonia Detection using CNN(92.6% Accuracy) | Kaggle." <https://www.kaggle.com/code/madz2000/pneumonia-detection-using-cnn-92-6-accuracy> (accessed Apr. 22, 2023).