Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio

Saman Fouladi, M.J. Ebadi, Ali A. Safaei, Mohd Yazid Bajuri, Ali Ahmadian

PII: S0140-3664(21)00235-8
DOI: https://doi.org/10.1016/j.comcom.2021.06.011
Reference: COMCOM 6805

To appear in: Computer Communications

Received date: 23 February 2021
Revised date: 28 May 2021
Accepted date: 10 June 2021

Please cite this article as: S. Fouladi, M.J. Ebadi, A.A. Safaei et al., Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio, Computer Communications (2021), doi: https://doi.org/10.1016/j.comcom.2021.06.011.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Elsevier B.V. All rights reserved.
Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via Software Defined Radio

Saman Fouladi¹, M. J. Ebadi², Ali A. Safaei¹, Mohd Yazid Bajuri³, Ali Ahmadian⁴, *

¹Department of Medical Informatics, Faculty of Medical Sciences, Tarbiat Modares University, Tehran, Iran
²DICEAM Department, Mediterranean University of Reggio Calabria Via Graziella Feo di Vito, 89060 Reggio Calabria, Italy
³Department of Orthopaedics and Traumatology, Faculty of Medicine, Universiti Kebangsaan Malaysia (UKM), Kuala Lumpur, Malaysia
⁴Institute of IR 4.0, The National University of Malaysia, 43600 Bangi, Malaysia

* Corresponding Author

Abstract

The novel 2019 coronavirus disease (COVID-19) has infected over 141 million people worldwide since April 20, 2021. More than 200 countries around the world have been affected by the coronavirus pandemic. Screening for COVID-19, we use fast and inexpensive images from computed tomography (CT) scans. In this paper, ResNet-50, VGG-16, convolutional neural network (CNN), convolutional auto-encoder neural network (CAENN), and machine learning (ML) methods are proposed for classifying Chest CT Images of COVID-19. The dataset consists of 1252 CT scans that are positive and 1230 CT scans that are negative for COVID-19 virus. The proposed models have priority over the other models that there is no need of pre-trained networks and data augmentation for them. The classification accuracies of ResNet-50, VGG-16, CNN, and CAENN) were obtained 92.24%, 94.07%, 93.84%, and 93.04% respectively. Among ML classifiers, the nearest neighbor (NN) had the highest performance with an accuracy of 94%.

Keywords: Computed Tomography, ResNet-50, VGG-16, Convolutional Neural Networks (CNN), Convolutional Auto-Encoder Neural Network (CAENN), COVID-19

1. Introduction

Coronavirus named Severe Acute Respiratory Syndrome-Corona Virus-2 (SARS-Cov-2) appeared in Wuhan at the end of 2019 [1]. Coronavirus may cause serious respiratory disease, including Acute Respiratory Distress Syndrome (ARDS), which can be fatal [2]. The most common symptoms that patients experience after being infected include fever, fatigue, cough, and respiratory distress. The degree of illness ranges from mild influenza to multi-organ failure, and even death depending on the patient’s immune system response to infection. As of April 2021, there was a serious outbreak of COVID-19 virus across the world, resulting in 141,057,106 confirmed cases and 3,015,043 confirmed deaths. This virus has been declared as a pandemic by the World Health Organization (WHO) after causing a failure in the health services of more than 200 countries due to the lack of medical staff and personnel [3].
Reverse transcription-polymerase chain reaction (RT-PCR), computed tomography (CT), Gene sequencing, and X-Ray based imaging methods are all among the widely used tests for screening the patients with COVID-19 [4], [5]. Radiological screening, such as chest CT scans or chest X-ray can help us closely monitor the disease and quickly isolate infected people [6]. CT-based imaging is inexpensive, reliable, practical, and available in all hospitals as an efficient tool for diagnosis, prediction, and follow-up of the patients with COVID-19 [7]. According to the wide range of studies conducted around the globe, the sensitivity of X-ray images is less than CT scans in the detection of patients with COVID-19 [8]. However, the number of patients and the spread rate of the disease are too high for experts to deal with. It has been observed in previous studies that manual examination in hospitals is a time-consuming process [9]. Thus, intelligent technologies will have a great contribution to disease diagnosis. Hence, artificial intelligence (AI) methods for making exact and early diagnoses have attracted the attention of many researchers. These methods make significant contributions to a wide range of fields from manufacturing to the healthcare industry [10]. Subsequent to the appearance of deep learning, AI methods have entered a new age. Deep learning (DL) structures learn the features and classify them automatically by providing an adequate amount of training examples. CNN architectures of high discriminative capacity are primarily used for image processing and reshaping medical imaging. In the automated processing of radiological images, DL techniques are commonly used [11].

Numerous CNN methods of high-efficiency and high-performance have been used for medical studies in the literature. By increasing the number of COVID-19 samples in hospitals, CT and X-Ray images have been shared. However, these methods were no longer so efficient for the existing COVID-19 datasets with unequal distribution of insufficient samples. Therefore, radiology experts have turned their attention to approaches that do not need such close supervision or data augmentation. In spite of the satisfactory results of new approaches, none of them is perfect in achieving the ultimate result.

Here, the recent studies using CT scans for classifying the patients with COVID-19 and healthy controls are summarized. We examined only articles that have used various types of CNNs including pre-trained networks. Several statistical tests such as accuracy, precision, sensitivity, specificity, F1 ranking, and so forth are used to evaluate the two-class classification results of CNNs. The studies were divided into three categories: articles that have used several pre-trained CNN, articles with one pre-trained CNN, and articles that have used CNN or other methods. Many researchers investigate feature selection such as [31]-[37]. In articles with more than one network, the maximum metrics of each network were considered. The findings of this survey are summarized in Table 1.

Rahimzadeh et al. [12] used the ResNet50V2 network, the Xception network, and a proposed CNN for the detection of COVID-19 from lung HRCT scans. The dataset was composed of 15589 images of normal people and 48260 images of infected with COVID-19. In this study, an image processing algorithm was proposed to filter the proper CT scans of the patients, showing inside their lungs. By using this algorithm, network accuracy and speed were increased. After training the three networks, the trained networks were used for running a full-automated system of COVID-19 identification. The system was studied on two different datasets: one with more than 7796 images and the other with 41892 images of different thicknesses from almost 245 patients. The model showed an overall accuracy of 98.49% for single image classification.

Also, Pham in [13] has presented a study of sixteen pre-trained CNNs for COVID-19 classification. In this study, higher classification rates were obtained by using transfer learning instead of data augmentation. The dataset was randomly divided into 80% and 20% respectively for training and testing data with the highest accuracy of 95% achieved by MobileNet-v2, the highest sensitivity of 98% achieved by ResNet-18, the highest specificity of
There were four separate networks of MobileNet-v2, ShuffleNet, ResNet-18, and DenseNet-201.

El-Kenawy et al. [14] have focused on experiments on three scenarios to evaluate accuracy and performance of the suggested framework for the classification of COVID-19. In scenario I, accuracies of many CNN models were compared on CT images from the COVID-19 dataset. The highest classification accuracy was reported 79% for AlexNet model. The highest precision of 84%, the highest sensitivity of 95%, the highest specificity of 92%, and the highest F1-score of 77% were respectively reported for GoogLeNet, VGG16Net, GoogLeNet, and AlexNet models.

For the COVID-19 diagnosis, a self-developed model was built, called CTnet-10. Shah et al. in [15], has proposed the CTnet-10 model that composed of four convolutional blocks. It passed through two convolutional blocks with 126x126x32 and 124x124x32 dimensions. Then it went into a 62x62x32 max-pooling followed by two convolutional layers with dimensions of 60x60x32 and 58x58x32. It then passed by a pooling layer of 29x29x32. After passing a fully connected layer of size 256, data were classified into negative or positive classes of COVID-19. The accuracy obtained from CTnet-10 model was 82%. VGG-19, InceptionV3, ResNet-50, VGG-16, and DenseNet-169 were some of the other models tested. The highest accuracy of 94% was obtained for the VGG-19 model. Due to the small dataset of COVID-19, it was possible to use pre-trained neural networks for COVID-19 (+) or COVID-19 (-) classification.

Attallah et al. in [16] have proposed a new computer-aided diagnosis (CAD) system called MULTI-DEEP, on the base of merger of multiple NCs to distinguish COVID-19 from other cases. There were four main scenarios in the framework. In scenario I and scenario II, the dataset of 397 normal chest CT images and 347 chest CT images of COVID-19 positive cases were classified. In scenario I, four pre-trained CNNs were applied to diagnose COVID-19 and non-COVID-19 cases. The highest performance was obtained for ResNet-18 with 78% accuracy, 76% sensitivity, 79% specificity, 81% precision, and 78% F1-score. In scenario II, the authors extracted deep features from every pre-trained CNN to be used for training SVM classifiers. Compared to other CNNs, ResNet-18 network resulted in the highest performance on deep features with 92% accuracy, 93% sensitivity, 91% specificity, 91% precision, and 92% F1-score.

In [17], Serte et al. have used the ResNet-50 deep learning model to predict COVID-19 on each CT image of 3D CT scans. This model fused image-level predictions to diagnose COVID-19 on a 3D CT volume. This dataset was composed of 3D CT scans of the patients, each comprised about 40 axial slices. This dataset included 1110 3D CT scans of patients taken in hospitals of Moscow, Russia. The Mosmed-1110 dataset consisted of five categories of 3D CT volumes. These groups were named CT0, CT1 to CT4. The CT0 was comprised of 254 3D CT volumes that were normal scans. CT2 consisted of 684 3D CT scans showing COVID-19 infection on the lungs. First, middle axial lung slices were selected and then each of them passed through ResNet-50 model. The classification accuracy was obtained 98% from this method.

Horry et al. [18] first have selected the VGG19 model and then extensively tuned to appropriate parameters in order to be performed at different levels of COVID-19 detection from pneumonia or normal cases for all three modes of lung images. This led to the precision of 84%, recall of 81%, and F1-score of 83%. In this study, the dataset consisted of 349 COVID-19 and 397 Non-COVID-19 images.

DenseNet201 on the base of deep transfer learning (DTL) model was utilized in [19] by Jaiswal et al. to identify whether patients were COVID-infected or not. The model achieved an accuracy of 96%, F1-score of 96%, recall of 96%, and precision of 96%.
Several datasets are needed to build a multi-task pipeline for coronavirus detection, prediction, and classification. For this goal, El-Bana et al. [20] used three datasets including two of the most popular datasets: 1) RSNA Pneumonia Detection Challenge Dataset, 2) COVID-19 Image Data Collection Repository, and 3) COVID-19 CT Segmentation Dataset. All scans were reshaped to $512 \times 512 \times 3$. Due to the small number of CT volumes, data augmentation techniques were used such as rotation, vertical and horizontal transformations, shearing, and zooming. The total images after data augmentation were 3,724. For fine-tuning, the Inception-v3 model was used for multi-label classifiers and multi-class classification. The multi-class classification resulted in 99% accuracy, 99% precision, 99% sensitivity, 98% specificity, and 99% F1-score.

Wang et al. [19] have built a deep learning system, using 3D volumes of CT for lesion localization and COVID-19 classification. A pre-trained UNet network was utilized for the segmentation of the lung area. The COVID-19 lesions were classified after the segmentation of 3D lung area fed into a 3D deep neural network (DNN) to predict the risk of COVID-19 infection. The classification specificity, sensitivity, and accuracy were obtained at 91%, 90%, and 90% respectively.

Chen et al. [20] have constructed a system based on UNet++ neural network for COVID-19 pneumonia detection at high CT resolution. To validate the model, 46,096 anonymous images were retrospectively collected from 106 admitted patients, of which 51 people were laboratory-confirmed cases of COVID-19 pneumonia and 55 people were control cases of other diseases in Renmin Hospital of Wuhan University. After filtering out images with good lung conditions, 35,355 images remained and were divided into retrospectively training and testing datasets. By applying findings of the radiologists as the archetype, an accuracy of 92.59%, sensitivity of 100%, and specificity of 81.82% were attained by the model per patient in the 27 prospective cases.

Wang et al. [21] have used 1065 CT images collected from 259 patients with the cohort including 180 patients of typical viral pneumonia and 79 patients of confirmed SARS-COV-2 by nucleic acid testing in three hospitals. Three main processes constituted the architecture: first, input image preprocessing; second ROI image feature-derivation and training; third classification by two fully connected layers and prediction by binary classifiers. Transfer learning was also performed involving a predefined model trained by the use of well-known GoogleNet Inception v3 CNN. The number of picture types in the training set was equal to the total number of 320 images. The remaining CT images were considered for internal validation. The internal validation resulted in a total accuracy of 89.5%, a sensitivity of 87%, and a specificity of 88%. Moreover, total accuracy of 79.3%, a sensitivity of 67%, and a specificity of 83% were reported for the external testing dataset.

Wang et al. [22] have used extracted characteristics of an auto-created CNN to learn individual representations at the image level. The CNN employed several new techniques such as average pooling by rank and data augmentation in several directions. Relational representations were learned using the Graphic Convolution Network (GCN). The Deep Merge of Functions (DFF) was developed to merge the individual features at the image level and the relational features of CCN and GNN. The best template was called FGCNet. The model achieved 97% accuracy, 96% precision, 97% sensitivity, and 96% specificity.

Wang et al. [23] have proposed a new collaborative learning framework to accurately identify COVID-19 by effective learning of heterogeneous datasets with distribution gaps. The network was made up of two branches, where the upper branch with a lightweight design had four different layers of convolution, and another branch was composed of blocks of denser connections for learning. In this analysis, COVID-CT and SARS-CoV-2 were used to test a joint learning system utilizing two common CT datasets of COVID-19. The dataset was composed of 2482 CT images from 120 patients, of whom 1230 were non-COVID with
different lung infection symptoms and 1252 were COVID-19. The clinical results of 397 CT images from 171 patients without COVID-19 and 349 CT images from 216 patients with COVID-19 were used in the COVIDCT dataset. Evaluation metrics were computed for both datasets. The achieved results of the SARS-CoV-2 dataset were better than those of the COVID-CT dataset. The overall accuracy was estimated at 90% with a recall of 85%, precision of 95%, and F1-score of 90%.

Singh et al. [24] have used a CNN for classification of positive COVID-19 and negative COVID-19 cases. To extract useful features, CNN utilized several convolutional and pooling layers. For experimental purposes, different ratios of training to testing datasets were considered, such as 20:80, 30:70, 40:60, 50:50, 60:40, 70:30, 80:20, and 90:10. The accuracy, F1-score, and sensitivity of CNN models were obtained by 92%, 98%, and 90%, respectively. The main objective of this study is to propose DL networks such as ResNet-50, VGG-16, CNN, and deep CAENN to differentiate between COVID-19 and Non-COVID using lung CT scans. The efficiency of the proposed models is compared to that of the current deep architectures on a typically accessible lung CT dataset. For neural networks of CNN and convolutional Auto-Encoder, we propose an optimal architecture and change the parameters and hyper parameters of optimization function, learning rate, and activation function. The optimal parameters of ResNet-50 and VGG-19 for training are changed but their architectures are not. The classification is done by the use of original images of dataset but no use of data augmentation techniques.

The followings are the key contributions of this article:

- Two efficient COVID-19 classification models have been developed on the base of ResNet-50 and VGG-16 neural networks.
- Two efficient COVID-19 classification models have been developed on the base of CNN and CAENN.
- The proposed models have been implemented on a big dataset including COVID-19 and Non-COVID-19 CT Scans.

The other sections of the paper are: The study in the area of deep learning neural networks and ML classifier for chest CT image classification is given in Section 2. Section 3 describes the proposed models. Section 4 is dedicated to the investigational findings and discussions. The conclusion is drawn in Section 5.

| Reference | Method | Dataset | Results |
|-----------|--------|---------|---------|
| [12]      | The pre-trained Xception network, the pre-trained ResNet50V2 network and a proposed pre-trained CNN | COVID-19: 48260 Non- COVID: 15589 | Accuracy (Acc.) proposed CNN 98.49% Acc. Xception 96.55% |
| [13]      | Sixteen pre-trained CNNs such as: SqueezeNet, GoogLeNet, Inception-v3, DenseNet-201, ResNet and etc. | COVID-19: 216 Non- COVID: 397 | Sensitivity (Sens.) 98.99 MobileNet-v2 95.97 ResNet-18 Acc. 96.67 DenseNet-201 Specificity (Spec.) 0.96 DenseNet-201 F1-score 0.96 MobileNet-v2 ShuffleNet ResNet-18 DenseNet-201 |
| Reference | Description | COVID-19: | Non- COVID: | Acc. | Prec. | Sens. | Spec. | F1-score |
|-----------|-------------|-----------|-------------|------|-------|-------|-------|----------|
| [14]      | AlexNet, VGG-COVID-19: ResNet-50, GoogLeNet, and 334 Net (VGG19Net and VGG16Net) are among the CNN versions. | 334 | 794 | 0.79 | 0.8475 | 0.95 | 0.92 | -|
| [15]      | Several pre-trained CNN such as: VGG-19, VGG-16, DenseNet-169, ResNet-50, CTNet-10, InceptionV3 | 349 | 216 | 0.945 | - | - | - | -|
| [16]      | Pre-trained CNNs such as AlexNet, GoogleNet, ResNet-18, ShuffleNet | 347 | 397 | 0.7829 | 0.81 | 0.769 | 0.799 | 0.789 |
| [17]      | The ResNet-50 neural network | 684 | 254 | - | - | - | - | -|
| [18]      | The VGG19 neural network | 349 | 397 | 0.84 | 0.81 | 0.769 | 0.799 | 0.789 |
| [25]      | Pre-trained DenseNet201 based on DTL | 1252 | 1230 | 0.96 | 0.96 | 0.96 | 0.96 | -|
| [26]      | Fine-tuned inception-v3 network with multiple-way data augmentation | Total data after Augmentation: 3,724 | - | 0.995 | 0.992 | 0.998 | 0.982 | 0.995 |
| [19]      | COVID-19 detection using a 3D CNN | 540 | 229 | 0.901 | 0.907 | 0.911 | - | -|
| [20]      | The UNet++ neural network | Total data: 35355 images | - | 92.59% | 100% | 81.82% | - | -|
| [21]      | The GoogleNet Inception v3 | Total data: 1065 images | - | 89.5% | 88% | 87% | - | -|
| [22]      | FGCNet is the best model of the eight suggested networks, showing the convergence of GCN and CNN networks with multiple-way data augmentation. | 113 | 209 | 0.9714 | 0.9661 | 0.9771 | 0.9656 | -|
| [23]      | The network was made up of two branches, where the upper branch a lightweight design | 349 | 397 | 0.9083 | 0.9575 | - | - | -|
having four different layers of convolution, and another branch is known lower which was made up of blocks having denser connections to represent the learning.

| [24] | CNN | COVID-19: 68 | Non-COVID: 64 | Recall (Rec.) | 0.8589 | F1-score | 0.9087 |
| | | | | Acc. | 0.9250 | Sens. | 0.9025 |

2. Material and Methods

2.1. Data acquisition

For this experiment, the 'Covid-19-Dataset' CT scan dataset was used retrieved from Kaggle\(^1\). The data set includes a total of 2482 CT scans, composed of 1230 negative CT scans and 1252 positive CT scans for SARS-CoV-2 (COVID-19) infection. Soares et al. [27] collected these images from hospitals in Sao Paulo, Brazil. This dataset is freely available in Kaggle. Fig. 1 shows a selection of CT scans from the dataset of non-COVID-19 and COVID-19 patients. The existing dataset of images must be resized before being fed to the convolutional network. All the CT images have been resized to 173*100 pixels for width and height, with a depth of 3.

![Sample of a) COVID-19 and b) Non-COVID CT scans](image)

2.2. Model Architecture and Model Training

Deep learning-driven models have recently been proven to be the winner in many clinical trials. These models outperform conventional mathematical models in terms of hand-crafted features in medical image processing and machine vision issues. CNNs are a form of DL technique in which several layers are strongly trained. These networks have very efficient and popular

\(^1\) Available in www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset
applications in classification, image processing, and neural computer vision [35]. A CNN network is made up of three major layers: convolution, pooling, and fully connected layers. Different layers function in different ways which leads to the ultimate learning. This network can be applied either alone or alongside other networks for data classification.

Features of chest CT scans are used to accurately classify into two classes of COVID-19 and non-COVID-19. The following steps are performed to classify COVID-19-infected person using the CNN-based proposed models:

I. Feature extraction

In this step, CNN uses several convolutions and a pooling layer to monitor and evaluate possible features. Fig. 2 shows how the stride may be used by the kernel/filter to extract possible features. After that, the max pooling layer is used to reduce the spatial size of the convolved features. It has capable of overcoming the overfitting problem. It assesses the maximum of the region from the feature map created by the convolution operator. Fig. 3 shows a max-pooling layer with a kernel size of 2 and a stride of 2. The rectified linear activation function (ReLU) is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

![Figure 2: CNN operator with kernel size 3 and stride of 1](image)

*Figure 2: CNN operator with kernel size 3 and stride of 1*

![Figure 3: Max pooling with a single pooled feature](image)

*Figure 3: Max pooling with a single pooled feature*
II. Classification

In this step, fully connected layers classify extracted features and assess the probability of the object in the input image. Typically, an activation function and a dropout layer are used to establish non-linearity and reduce overfitting. As shown in Fig. 4, the fully connected layer classifies features into two classes.

![Figure 4: Fully Connected layer](image)

3.2.1 ResNet Neural Network

The ResNet is a type of CNN often used in the field of computer vision since winning the 2015 ILSVRC competition [27]. With a depth of 152 layers, this architecture was named the deepest at that time. Deep ResNet operate similar to deep CNNs but they implement a residual connection between each layer and the output of the following layer. This requires that each layer receives features as input from its two previous layers. The residual connections perform better upon regular neural networks in two ways. First, they reduce the vanishing gradient problem by allowing the use of a different path for gradient flow. Second, they allow the model to learn referenced functions which ensures deeper layers will execute either better or as good as shallower layers [27]. In this paper, ResNet-50 has been used and Fig. 5 describes the architecture of this neural network. "ID BLOCK" in the diagram stands for "Identity block"," and "ID BLOCK x3" means stack 3 identity blocks together.
For this network, from the total number of 2482 data, 90% (2233 numbers) were used as training data and 10% (249 numbers) as test data. The activation for all layers except the last layer was Relu function. Adam with a learning rate of 0.0001 was selected for the optimization function. This network was trained over 100 training epochs and data were transmitted to the network in batches of 8 size (batch-size). The duration of each epoch was the 20s.

**VGG-16 Neural Network**

VGG-16 is a CNN model, also called the OxfordNet model, proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". It was one of the famous models submitted to ILSVRC-2014. Number 16 refers that it has a total of 16 layers that has some weights [28]. The architecture for VGG16 network is shown in Fig. 6. From the total number of 2482 data, 90% (2233 numbers) were used as training data and 10% (249 numbers) as test data. The input to the main VGG 16 model is images of 224x224x3 pixels changed to 173x100x3 in this study. Then, there are two convolution layers of 70x70x64 size, two convolution layers of 128 filter length, three convolution layers of 256 filter length, and three convolution layers of 512 filter length. Then again there are three convolution layers of 512 filter length. The Kernel size is 3x3 and the pool size is 2x2 with step 2x2 for all the layers. The output of the last pooling layer is 2D which must be converted into a 1D layer to be sent to fully connected layers, done by a flatten layer. After the convolution layers, two 4096 fully connected layers and two fully connected layers were used to classify data into two classes by softmax activation function. The activation for all layers except the last layer was Relu function. Adam with a learning rate of 0.0001 was selected for the optimization function. Also, the same padding was used for all convolution layers. This network was trained over 100 training epochs and data were transmitted to the network in batches of 32 size (batch-size). The duration of each epoch was 15s.
3.2.2 Convolutional Neural Network

Fig. 7 presents the proposed architecture for a two-dimensional CNN. From the total number of 2482 data, 80% (1985 numbers) and 20% (497 numbers) were used respectively as training and testing data. The network includes 2 convolution layers of 128 filter length, 2 convolution layers of 64 filter length, 2 convolution layers of 32 filter length, 2 convolution layers of 16 filter length, and 2 convolution layers of 8 filter length. The 2*2 kernel function was considered for all layers. In general, a convolution network is a hierarchical neural network in which the layers of convolution are interconnected with alternate pooling layers and then with several fully connected layers. However, there is no compulsion to utilize a pooling layer after each layer of convolution. It is observable in the figure that the network consists of 10 convolution layers but 5 pooling layers. In this architecture, the 2*2 max-pooling layer is considered with step 2. The last pooling layer output is 2D which must be converted into a 1D convolution layer to be sent to fully connected layers, done by a flatten layer. Also, one type of padding must be used to control the output size of each convolution layer. For all the networks in this study, the same padding has been used to fill the edges of input data with similar values in adjacent cells. After the convolution layers, a 128 fully connected layer and 2 fully connected layers were used to classify data into two classes by softmax activation function. In order to prevent overfitting, batch normalization layers were considered after all convolution layers and a dropout layer.
with a rate of 0.1 is considered after the first fully connected layer. The activation function used for this network was ReLU function excluding the last fully connected layer.

Extracted features from the convolution layers are the input for the first fully connected layer to $U_{fc} = 128$ hidden layers. The output of flatten layer equals $3 \times 5 \times 8 = 120$. Thus, the number of weights is $W_{conv} = \text{Outflatten} \times U_{fc} = 128 \times 120 = 15,360$ and the number of existing parameters to the second fully connected layer is $15,360 + 128$ (biases) = 15,488. Table 2 illustrates the learning parameters of this network. The Sum of all the used parameters can be calculated from the sum of the values of the Param column from Table 2. The resulting sum is 150,578 of which 149,586 numbers are associated with learning and 992 numbers with non-learning parameters.

**Figure 7**: The architecture of the proposed CNN

Lung CT Scans

173 * 100 * 3
Table 2. Parameters used in the convolution network for classification in 2 classes

| Layer (type)                      | Output Shape         | Param #  |
|-----------------------------------|----------------------|----------|
| conv2d_1 (Conv2D)                 | (None, 100, 173, 128) | 1664     |
| batch_normalization_1 (Batch)     | (None, 100, 173, 128) | 512      |
| conv2d_2 (Conv2D)                 | (None, 100, 173, 128) | 65664    |
| batch_normalization_2 (Batch)     | (None, 100, 173, 128) | 512      |
| max_pooling2d_1 (MaxPooling2D)    | (None, 50, 86, 128)  | 0        |
| conv2d_3 (Conv2D)                 | (None, 50, 86, 64)   | 32832    |
| batch_normalization_3 (Batch)     | (None, 50, 86, 64)   | 256      |
| conv2d_4 (Conv2D)                 | (None, 50, 86, 64)   | 16448    |
| batch_normalization_4 (Batch)     | (None, 50, 86, 64)   | 256      |
| max_pooling2d_2 (MaxPooling2D)    | (None, 25, 43, 64)   | 0        |
| conv2d_5 (Conv2D)                 | (None, 25, 43, 32)   | 8224     |
| batch_normalization_5 (Batch)     | (None, 25, 43, 32)   | 128      |
| conv2d_6 (Conv2D)                 | (None, 25, 43, 32)   | 4128     |
| batch_normalization_6 (Batch)     | (None, 25, 43, 32)   | 128      |
| max_pooling2d_3 (MaxPooling2D)    | (None, 12, 21, 32)   | 0        |
| conv2d_7 (Conv2D)                 | (None, 12, 21, 16)   | 2064     |
| batch_normalization_7 (Batch)     | (None, 12, 21, 16)   | 64       |
| conv2d_8 (Conv2D)                 | (None, 12, 21, 16)   | 1040     |
| batch_normalization_8 (Batch)     | (None, 12, 21, 16)   | 64       |
| max_pooling2d_4 (MaxPooling2D)    | (None, 6, 10, 16)    | 0        |
| conv2d_9 (Conv2D)                 | (None, 6, 10, 8)     | 520      |
| batch_normalization_9 (Batch)     | (None, 6, 10, 8)     | 32       |
| conv2d_10 (Conv2D)                | (None, 6, 10, 8)     | 264      |
| batch_normalization_10 (Batch)    | (None, 6, 10, 8)     | 32       |
| max_pooling2d_5 (MaxPooling2D)    | (None, 3, 5, 8)      | 0        |
| Flatten_1 (Flatten)               | (None, 120)          | 0        |
| Dense_1 (Dense)                   | (None, 128)          | 15488    |
| Dropout_1 (Dropout)               | (None, 128)          | 0        |
| dense_2 (Dense)                   | (None, 2)            | 258      |

Total params: 150,578
Trainable params: 149,586
Non-trainable params: 992
To select an optimization function, SGD, Adadelta, and Adam functions were investigated. The results of using Adam function were significantly better than those of the other two functions. The learning rate parameter was tested with values of 0.001 and 0.0001 while the best value with the least learning error was 0.0001. The network was trained over 100 training epochs and the data was transmitted to the network in batches of 16 size (batch-size). The duration of each epoch was 9s.

3.2.3 Convolutional auto-encoder Neural Network

The structure of the designed CAENN for this research can be seen in Fig. 8. This architecture comprises two parts; the first includes a CAENN for training data, and the second includes a simple convolutional network for classification making use of the last encoder layer output of the first part. Out of a total number of 16449 data, 67% (11020 numbers) were used as training data and 33% (5429 numbers) were used as test data for the network. Out of the total number of 2482 data, 80% (1985 numbers) and 20% (497 numbers) were respectively used as training and testing data. The network includes two convolution layers of 128 filter length, two convolution layers of 64 filter length, and one convolution layer of 32 filter length. The decoder part consists of one convolution layer of 32 filter length, 2 convolution layers of 64 filter length, and 2 convolution layers of 128 filter length. The 2*2 kernel function was considered for all layers. In the encoder, after a sequence of two layers of convolution, a 2*2 max-pooling layer is considered. In the decoder, after a sequence of two convolution layers, an up-sampling layer with size 2*2 was considered. To prevent the network from overfitting, the layer of batch normalization was used after each layer of convolution. In the last layer of the encoder part, important features of the input data are extracted and this layer output can be used for the classification part of the network as can be seen in Fig 5. The output of this layer is trained by two continuous convolution layers with 64 filter length, a kernel function of size 2*2, and a max-pooling layer of size 2*2. To avoid overfitting, layers of batch normalization and 0.1 dropout are used. After the Flatten layer, 2 fully connected layers are used to classify data into two classes by softmax activation function. The used activation function for other layers was Relu function in this network. To select an optimization function, SGD, Adadelta, and Adam functions were investigated. The results of using Adam function were significantly better than those of the other two functions. The learning rate parameter was tested with values of 0.001 and 0.0001 while the best value with the least learning error was 0.0001. The network was trained over 100 training epochs and the data was transmitted to the network in batches of 16 size (batch-size). The duration of each epoch was 9s.
For the CAENN, the parameters in the encoder and classifier are important for training and classification. Extracted features from the encoder last layer are trained by several layers of convolution, and the final extracted features become the input of the first fully connected layer to the hidden layer of $U_{fc} = 2$. The number of weights $W_{conv}$ depends on the output size of flatten layer and the number of hidden layers in the fully connected layer. The output of flatten layer equals $6 \times 11 \times 64 = 4,224$. Thus, the number of weights is $W_{conv} = \text{Outflatten} \times U_{fc} = 4,224 \times 2 = 8,448$ and the number of existing parameters to the second fully connected layer is $8,448 + 2$ (biases) = 8,450. Table 3 illustrates the learning parameters of this network. The sum of all the used parameters can be calculated from some of the values of the param column from Table 3. The resulting sum is 159,906, of which 158,946 numbers are associated with learning and 960 numbers with non-learning parameters.
Table 3. Parameters used in the CAENN for classification in 2 classes

| Layer (type)            | Output Shape       | Param# |
|-------------------------|--------------------|--------|
| input_1 (InputLayer)    | (None, 100, 173, 3)| 0      |
| conv2d (Conv2D)         | (None, 100, 173, 128)| 1664  |
| batch_normalization (Batch)| (None, 100, 173, 128)| 512   |
| conv2d_1 (Conv2D)       | (None, 100, 173, 128)| 65664 |
| batch_normalization_1(Batch)| (None, 100, 173, 128)| 512   |
| max_pooling2d (MaxPooling2D) | (None, 50, 87, 128)| 0      |
| conv2d_2 (Conv2D)       | (None, 50, 87, 64) | 32832  |
| batch_normalization_2(Batch)| (None, 50, 87, 64) | 256   |
| conv2d_3 (Conv2D)       | (None, 50, 87, 64) | 16448  |
| batch_normalization_3(Batch)| (None, 50, 87, 64) | 256   |
| max_pooling2d_1(MaxPooling2D)| (None, 25, 44, 64)| 0      |
| conv2d_4 (Conv2D)       | (None, 25, 44, 32)| 8224   |
| batch_normalization_4(Batch)| (None, 25, 44, 32)| 128   |
| max_pooling2d_2(MaxPooling2D)| (None, 13, 22, 32)| 0      |
| conv8 (Conv2D)          | (None, 13, 22, 64)| 8256   |
| conv2d_11 (Conv2D)      | (None, 13, 22, 64)| 16448  |
| batch_normalization_10(Batch)| (None, 13, 22, 64)| 256   |
| max7 (MaxPooling2D)     | (None, 6, 11, 64) | 0      |
| dropout (Dropout)       | (None, 6, 11, 64) | 0      |
| flatten (Flatten)       | (None, 3224)      | 0      |
| dense (Dense)           | (None, 2)         | 8450   |

Total params: 159,906
Trainable params: 158,946
Non-trainable params: 960

Our entire implementation was done in Keras with Tensorflow backend. The networks in this research were designed in Python environment, implemented using the cross library, and run in Google Colaboratory (Colab) environment. Colab provides a platform for running Python codes, especially ML, deep learning, and data analysis. Colab hardware specifications are listed in Table 4.
Table 4. Colab hardware specification

| Hardware | Description |
|----------|-------------|
| GPU      | 1xTesla K80, VRAM of 12GB GDDR5, 2496 CUDA cores, compute 3.7 |
| CPU      | One core, Two threads, Xeon Processors of 2.3Ghz |
| RAM      | ~12.6 GB |
| Disk     | ~33 GB |

3. Experiment Results

The current study aims to classify CT scans from patients into covid-19 and non-covid-19 categories. Metrics include F1-score, recall, precision, and accuracy. Accuracy is a metric obtained from dividing the number of correctly recognized to the total number of cases. Accuracy is the proximity of a calculated value to a normal or real value. In other words, the tool can measure the exact amount whose accuracy can be measured.

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]  

(1)

In ML, precision for a class is the number of truly classified items (correctly labeled) divided by the sum of either true or false items labeled as belonging to that class. Recall refers to the fraction of truly classified of the total number of classified items in a class.

\[
\text{Precision} = \frac{TP}{TP + FP}
\]  

(2)

\[
\text{Recall} = \frac{TP}{TP + FN}
\]  

(3)

Based on the precision and recall calculations, the weighting value for F1-score can be calculated. The F1-score is a useful metric for assessing classification efficiency and defining the weighted average of precision and recall quantities. The value of this measure for a classification algorithm is equal to 1 under the ideal condition and equals zero under the worst condition. This parameter is calculated according to the following equation [29]. Table 5 summarizes the results of the ResNet-50, VGG-16, CNN, and CAENN. The training accuracy of ResNet-50 is found to be 97.9408% and its validation accuracy is 92.2244%. The training and validation accuracies of VGG-16 are 98.3224% and 94.0763% respectively. The proposed CNN has training and validation accuracies of 97.3133% and 93.8430% respectively. Also, the training accuracy of suggested CAENN is obtained as 98.5949%, and its validation accuracy is 93.0482%.
Table 5. Results from the proposed network for classification into 2 classes.

| Network      | Acc.       | Prec.     | Rec.       | F1-score   |
|--------------|------------|-----------|------------|------------|
| ResNet-50    | 0.92244979 | 0.9650    | 0.9650     | 0.9650     |
| VGG-16       | 0.94076305 | 0.9550    | 0.9550     | 0.9550     |
| CNN          | 0.93843057 | 0.99252539| 0.83801088 | 0.84655203 |
| CAENN        | 0.93048090 | 0.96512168| 0.74901946 | 0.84344849 |

According to Table 5, the accuracy, precision, recall, and F1-score of each DL network are presented in Fig. 9.

![Figure 9: Comparison of the F1-score, recall, precision, and accuracy, for proposed networks](image-url)

The training and validation accuracies, and loss analysis of the suggested models with respect to the epochs number are seen in Fig. 10.
Figure 10: Training and validation analysis over 100 epochs for 1) ResNet-50 network: (a) Accuracy analysis of Training and Testing, (b) Loss analysis of Training and Testing, 2) VGG-16 network: (c) Accuracy analysis of Training and Testing, (d) Loss analysis of Training and Testing, 3) CNN: (e) Accuracy analysis of Training and Testing, (f) Loss analysis of Training and Testing, 4) CAENN: (g) Accuracy analysis of Training and Testing, (h) Loss analysis of Training and Testing.

It is important in medical research, particularly for critical diseases like COVID-19, to minimize false negative and false positive outcomes in the modeling process. Classification accuracy of all networks is mentioned and impacts of false negative and false positive rate are presented in Fig. 11. It is clear that networks well performed in terms of reducing the number of false negatives and false positives.
Receiver Operating Characteristics (ROC) plots are helpful for grouping, analyzing, and visualizing the output of classifiers. ROC plots are widely used in medical decision-making and have recently gained popularity in data mining and ML. The ROC curve is provided by drawing true positive rate (TPR) versus false positive rate (FPR) in a diagram of various threshold settings. Maximizing the TPR, thus lowering the FPR, is optimal. This implies that the optimal point (FPR=0 and TPR=1) is in the upper left corner of the plot. Fig. 12 shows the ROC curve of the networks along with class 0 and class 1 based COVID-19 classification networks. The ideal point is observable for both class 0 and class 1. The maximum value (area=1.00) was used to calculate the area under the ROC curve (also known as the area under the curve (AUC)).

![Figure 11: The Confusion Matrix measurements of TP, TN, FP, and FN ratios of the proposed model derived from the testing dataset of a) ResNet-50 neural network b) VGG-16 neural network c) CNN and d) CAENN](image)
The outcome of classical ML classifiers such as NN, SVM, RF, SGD LR, and MLP was compared for classification into 2 classes. The obtained accuracy was 94% for nearest neighbor, 88% for random forest, 73% for logistic regression, 69% for stochastic gradient descent, 59% multilayer for perceptron, and 48% for support vector machine. Fig. 13 shows the comparison of these results. The F1-score, recall, and precision for each set of Healthy Control and Covid-19 data were calculated by these methods and summarized in Table 6. For COVID-19 data, the highest precision was obtained for random forest 93%, the highest recall for SVM 100%, and the highest F1-score for KNN 94%. For Non-COVID data, the highest precision, F1-score, and recall were respectively calculated 97%, 94%, and 95% for RF.
Table 6 ML precision, F1-score, and recall COVID-19 and Non-COVID-19 classification of classifiers

| Method | COVID-19 | | | Non-COVID | | |
|--------|----------|---|---|----------|---|---|
|        | Prec.    | Rec. | F1-score | Prec.    | Rec. | F1-score |
| KNN    | 0.92     | 0.97 | 0.94     | 0.97     | 0.92 | 0.94     |
| RF     | 0.93     | 0.82 | 0.88     | 0.85     | 0.95 | 0.90     |
| LR     | 0.75     | 0.68 | 0.72     | 0.73     | 0.79 | 0.76     |
| SGD    | 0.89     | 0.42 | 0.57     | 0.64     | 0.95 | 0.76     |
| MLP    | 0.58     | 0.57 | 0.57     | 0.61     | 0.62 | 0.61     |
| SVM    | 0.48     | 1.00 | 0.65     | 0.00     | 0.00 | 0.00     |

According to Table 6, the average precision, average recall, and average F1-score are presented for each ML classification method in Fig. 14. The highest average results are observed at the KNN, RF, LR, SGD, MLP, and SVM, respectively.

Figure 14: Comparison of the average precision, recall, and F1-score for ML classifiers

Similar data to ours in this study have been investigated in other studies using other neural network architectures. Also, some studies have used pre-trained networks such as VGG-16 and ResNet-50. The results of our proposed networks have been compared with those of other
studies as shown in Table 7. Among our proposed models, KNN and VGG-16 outperformed other models including CNN, ResNet-50, and CAENN by at least 1% higher accuracy. The accuracy obtained from the proposed CNN is equal to that of the proposed CAENN. Similar to our study, [25] and [23] have also used the same dataset. In [13], [14] and [16], ResNet-50 and CNN neural networks have been used and in [24], only CNN has been used. In [25], DenseNet201 based DTL has been suggested with a classification accuracy 2% more than that of KNN and VGG-16 classifiers in this research.

In [23], a special architecture of CNN was developed to classify COVID-19 dataset. This study obtained accuracy 4% less than that of VGG-16 and KNN classifiers, 3% less than that of CAENN and CNN, and 2% less than that of ResNet-50 network proposed in our study.

In [13], ResNet-50 was used and accuracy was obtained equal to that of ResNet-50 and less than that of other networks in our study. The accuracy of ResNet-50 used in [14] was 15% less than the accuracy of that used in our research.

Another type of ResNet is the ResNET-18 CNN. In [16], ResNET-18 and other pre-trained CNNs such as GoogleNet, AlexNet, and ShuffleNet were proposed for being used in CAD system to distinguish COVID-19 from other cases and classify them. The accuracy of proposed ResNET-18 in this study was 92%, equal to the accuracy of proposed ResNET-50 in our study, whereas it was less than that of our proposed VGG-16, CNN, KNN, and CAENN models.

In [24], a CNN with several pooling layers and convolutional layers was used to extract CT image features and classify them into two classes. This network achieved an accuracy of 92%, which is equal to the accuracy of our ResNet-50, but 1% less than that of our CNN model.

Table 7. Comparison of the classification resulted from deep learning networks of this study and other studies with the same data.

| Reference | Method | Acc. | Prec. | Sens. (Rec.) | Spec. | F1-score |
|-----------|--------|------|-------|--------------|-------|----------|
| [25]      | Pre-trained DenseNet201 based DTL | 0.9625 | 0.9629 | 0.9629 | - | 0.9629 |
| [23]      | There are two branches of the network: 1) four separate convolutional layers 2) heavier dens connections for Representation learning | 0.9083 | 0.9575 | 0.8589 | - | 0.9087 |
| [13]      | ResNet-50 | 0.9262 | - | 0.9114 | 0.9429 | 0.93 |
| [14]      | ResNet-50 | 0.77 | - | 0.6250 | 0.8862 | 0.7059 |
| [16]      | ResNet-18 | 0.925 | 0.916 | 0.933 | 0.918 | 0.925 |
| [24]      | CNN | 0.9250 | - | 0.9025 | - | 0.985 |
| Proposed ResNet-50 | ResNet-50 | 0.9224 | 0.9650 | 0.9650 | - | 0.9650 |
| Proposed VGG-16 | VGG-16 | 0.9407 | 0.9550 | 0.9550 | - | 0.9550 |
| Proposed CNN | CNN | 0.9384 | 0.9925 | 0.8380 | - | 0.8465 |
| Proposed CAENN | 2D-CAENN | 0.9304 | 0.9651 | 0.7490 | - | 0.8434 |
| Proposed KNN | - | 0.94 | 0.9450 | 0.9450 | - | 0.94 |
According to Table 7, the accuracy of each model is presented in Fig. 15.

![Figure 15: Comparison of the accuracy of our proposed models and other studies](image)

The pre-trained CNN models are widely used in various studies with promising results. A comparison was made between the results of these models in Table 8. Among them, ResNet and VGG networks are the most used with varied results. The highest accuracy and sensitivity were respectively obtained 99.5% for Inception-v3 and 100% for ResNet-50 and U-Net++.

As mentioned, we have implemented models in Google Colaboratory (Colab) environment and the duration of each epoch was 20s, 15s for ResNet-50 and VGG-19 respectively, and 9s for both CNN and CAENN. As a result, 100 epoch duration for ResNet-50 was 33.33m, for VGG-19 was 25m and for CNN and CAENN was 15m. In this study, we have proposed a CNN with ten convolution layers and batch normalization layer, and dropout layer to avoid overfitting. The train parameters and hyper-parameters of networks such as learning rate, batch size, and activation function were adjusted to obtain high results. Although in our study pre-processing techniques were not used, compared with [24], our proposed CNN has improved accuracy and obtained precision of 99%. Also, we have proposed a CAENN with five convolution layers in the encoder section, five convolution layers in the decoder section, and two convolution layers for classification using the output of the last layer of the encoder. In this network, batch normalization layers and dropout layers were used to avoid overfitting as well. Among the studies reviewed, none of them used CAENN but the results of this network are promising. The accuracy obtained from CAENN is almost equal to CNN (93%) and has achieved a precision of 96%.

The architecture of pre-trained neural networks such as ResNet-50 and VGG-16 are defined, so different results can be obtained by changing the parameter and hyper-parameters of these networks. We have used a VGG-16 and our obtained results are higher than studies that used
VGG-16 or VGG-19 network. According to Table 8, our proposed VGG even increased the accuracy by 10%. Also, the precision of 95%, Recall of 95%, and F1-score of 95% have been achieved by this network.

In this study, another proposed pre-trained neural network was ResNet-50. For this model, we adjusted the parameters of network training to achieve optimal results of classification. Compared with [14], our ResNet-50 increased the accuracy by 15%. Although in [13], transfer learning technique was used to train the ResNet-50 network the obtained accuracy was equal to the accuracy obtained by our proposed ResNet-50. Also, our proposed ResNet-50 improved precision, Recall, and F1-score. In [16], features of CT images extracted by the ResNet-18 network and then classified by SVM. Although in our study machine learning classifiers were not used, obtained results were improved.

| Classification Model | Acc.  | Prec. | Spec. | Sens. |
|----------------------|-------|-------|-------|-------|
| Xception             | 96.55%| 81.74%| 97.45%| 97.24%|
| Alex-Net             | 79%   | -     | 77%   | 81%   |
| MobileNet-v2         | 95.97%| -     | 95.14%| 96.71%|
| VGG-19               | 94.5% | -     | -     | -     |
| VGG-19               | 84%   | -     | -     | 81%   |
| ResNet-18            | 78.29%| 81%   | 79.9% | 76.9% |
| ResNet-18            | 90.16%| -     | 90.95%| 89.45%|
| ResNet-50            | 98%   | -     | 80%   | 100%  |
| DenseNet201          | 96%   | 96%   | -     | 96%   |
| Inception-v3         | 99.5% | 99.2% | 98.2% | 99.8% |
| U-Net++              | 92%   | -     | 81.82%| 100%  |
| Google-Net           | 89.5% | -     | 87%   | 88%   |

4. Conclusion

In this study, the data set included 2492 CT-scan images with 1262 images of positive (COVID-19) and 1230 images of negative (non-COVID) cases. All scans were resized to 173 × 100 × 3. Using these images, four deep learning networks were implemented for differentiating between COVID and Non-COVID cases including ResNet-50, VGG-16, CNN, and CAENNs. Also, ML classifiers of SVM, NN, RF, SGD, LR, and MLP were compared for two-class classification. The ResNet-50, VGG-16, and CAENNs classified the chest CT-scans with validation accuracies of 92.24%, 94.07%, 93.84%, and 93.04%, respectively. The best performance among ML classifiers belonged to CNN with an accuracy of 94%. Also, other evaluation metrics such as recall, precision, and F1-score were computed for the proposed networks. The highest precision was obtained 99% by CNN and the highest recall and F1-score were obtained 96% by ResNet-50. In this research, the proposed networks were tested to classify COVID-19, SARS, MERS, and EBOLA images showing acceptable results. In future research, it is suggested to test other desirable networks to classify these images.
Compared to the existing studies with a similar dataset, this research achieved better results although techniques of image preprocessing and data augmentation were not used. Also, the ResNet and CNN networks used in our study showed a higher performance compared to similar networks in other studies. The use of pre-trained network or data augmentation yielded an accuracy of 99%, reported in previous research.

The use of other datasets including more images than the initially published ones is also suggested for future work. To improve the results, it is recommended to use data augmentation and data analysis techniques for further investigation. Additionally, DL-based optimal models can be used for the diagnosis of COVID-19, SARS, MERS, and EBOLA diseases from X-ray or CT images. DL models used to quantify and segment the severity of infected region of chest from CT images. So it is recommended to use these models for detecting the range of infected tissues. To aid physicians in the real-time diagnosing of COVID-19, it is suggested to develop a computer aided diagnosis (CAD) system using proposed models. The future work intends to develop deep learning models to classify recorded voice from people into covid-19 and non-covid-19 categories.

References

[1] C.-C. Lai, T.-P. Shih, W.-C. Ko, H.-J. Tang, P.-R. Hsueh, Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges, Int. J. Antimicrob. Agents. 55 (2020) 105924. https://doi.org/10.1016/j.ijantimicag.2020.105924.

[2] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei, J. Xia, T. Yu, X. Zhang, L. Zhang, Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study, Lancet. 395 (2020). https://doi.org/10.1016/S0140-6736(20)30211-7.

[3] Coronavirus disease (covid-19) outbreak situation, WHO. (n.d.). https://www.who.int/emergencies/diseases/novel-coronavirus-2019.

[4] Y. Yang, M. Yang, C. Shen, F. Wang, J. Yuan, J. Li, M. Zhang, Z. Wang, L. Xing, J. Wei, L. Peng, G. Wong, H. Zheng, W. Wu, M. Liao, K. Feng, J. Li, Q. Yang, J. Zhao, Z. Zhang, L. Liu, Y. Liu, Evaluating the accuracy of different respiratory specimens in the laboratory diagnosis and monitoring the viral shedding of 2019-nCoV infections, MedRxiv. (2020) 2020.02.11.20021493. https://doi.org/10.1101/2020.02.11.20021493.

[5] Y. Yang, M. Yang, C. Shen, F. Wang, J. Yuan, J. Li, M. Zhang, Z. Wang, L. Xing, J. Wei, L. Peng, H. Zheng, M. Liao, M. Liao, K. Feng, J. Li, Q. Yang, J. Zhao, Y. Liu, Laboratory diagnosis and monitoring the viral shedding of 2019-nCoV infections, 2020. https://doi.org/10.1101/2020.02.11.20021493.

[6] A. Gupta, D. Singh, M. Kaur, An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps, J. Ambient Intell. Humaniz. Comput. 11 (2020) 1309–1324. https://doi.org/10.1007/s12652-019-01493-x.

[7] S. Salehi, A. Abedi, S. Balakrishnan, A. Gholamrezae-zhad, Coronavirus Disease 2019 (COVID-19): A Systematic Review of Imaging Findings in 919 Patients., AJR. Am. J. Roentgenol. 215 (2020) 87–93. https://doi.org/10.2214/AJR.20.23034.

[8] H.Y.F. Wong, H. Lam, A.H.-T. Fong, B.S.T. Leung, T. Chin, C. Lo, M. Lui, J. Lee, W.H. Chiu, T. Chung, E. Lee, E. Wan, F. Hung, T. Lam, M. Kuo, M.-Y. Ng, Frequency and Distribution of Chest Radiographic Findings in COVID-19 Positive Patients, Radiology. (2020) 201160. https://doi.org/10.1148/radiol.2020201160.
[9] J.H. Tanne, E. Hayasaki, M. Zastrow, P. Pulla, P. Smith, A.G. Rada, Covid-19: how doctors and healthcare systems are tackling coronavirus worldwide., BMJ. 368 (2020) m1090. https://doi.org/10.1136/bmj.m1090.

[10] H. Jaakkola, J. Henno, J. Mäkelä, B. Thalheim, Artificial Intelligence Yesterday, Today and Tomorrow, in: 2019 42nd Int. Conf. Inf. Commun. Technol. Electron. Microelectron., 2019: pp. 860–867. https://doi.org/10.23919/MIPRO.2019.8756913.

[11] D. Singh, M. Kaur, Fusion of medical images using deep belief networks, Cluster Comput. 23 (2020). https://doi.org/10.1007/s10586-019-02999-x.

[12] M. Rahimzadeh, A. Attar, S.M. Sakhai, A Fully Automated Deep Learning-based Network For Detecting COVID-19 from a New And Large Lung CT Scan Dataset, Biomed. Signal Process. Control. (2021) 102588. https://doi.org/10.1016/j.bspc.2021.102588.

[13] T.D. Pham, A comprehensive study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks, Sci. Rep. 10 (2020) 16942. https://doi.org/10.1038/s41598-020-74164-z.

[14] E.-S.M. El-Kenawy, A. Ibrahim, S. Mirjalili, M.M. Eid, S.E. Hussein, Novel Feature Selection and Voting Classifier Algorithms for COVID-19 Classification in CT Images, IEEE Access. 8 (2020) 179317–179335. https://doi.org/10.1109/ACCESS.2020.3028012.

[15] V. Shah, R. Keniya, A. Shridharani, M. Punjab, J. Shah, M. Mehendale, Diagnosis of COVID-19 using CT scan images and deep learning techniques, MedRxiv. (2020) 2020.07.11.20151332. https://doi.org/10.1101/2020.07.11.20151332.

[16] O. Attallah, D.A. Ragab, M. Sharks, MULTI-DEEP: A novel CAD system for coronavirus (COVID-19) diagnosis from CT images using multiple convolution neural networks., PeerJ. 8 (2020) e10086. https://doi.org/10.7717/peerj.10086.

[17] S. Serte, H. Demirel, Deep learning for diagnosis of COVID-19 using 3D CT scans, Comput. Biol. Med. 132 (2021) 104306. https://doi.org/10.1016/j.compbiomed.2021.104306.

[18] M.J. Horry, S. Chakraborty, M. Paul, A. Ulhaq, B. Pradhan, M. Saha, N. Shukla, COVID-19 Detection Through Transfer Learning Using Multimodal Imaging Data, IEEE Access. 8 (2020) 149808–149824. https://doi.org/10.1109/ACCESS.2020.3016780.

[19] X. Wang, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, C. Zheng, A Weakly-Supervised Framework for COVID-19 Classification and Lesion Localization from Chest CT, IEEE Trans. Med. Imaging. 39 (2020) 2615–2625. https://doi.org/10.1109/TMI.2020.2995965.

[20] J. Chen, L. Wu, J. Zhang, L. Zhang, D. Gong, Y. Zhao, Q. Chen, S. Huang, M. Yang, X. Yang, S. Hu, Y. Wang, X. Hu, B. Zheng, K. Zhang, H. Wu, Z. Dong, Y. Xu, Y. Zhu, X. Chen, M. Zhang, L. Yu, F. Cheng, H. Yu, Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography, Sci. Rep. 10 (2020) 19196. https://doi.org/10.1038/s41598-020-76282-0.

[21] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng, B. Xu, A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19), Eur. Radiol. (2021). https://doi.org/10.1007/s00330-021-07715-1.

[22] S.-H. Wang, V.V. Govindaraj, J.M. Górriz, X. Zhang, Y.-D. Zhang, Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network, Inf. Fusion. 67 (2021) 208–229. https://doi.org/https://doi.org/10.1016/j.inffus.2020.10.004.
[23] Z. Wang, Q. Liu, Q. Dou, Contrastive Cross-Site Learning With Redesigned Net for COVID-19 CT Classification, IEEE J. Biomed. Heal. Informatics. 24 (2020) 2806–2813. https://doi.org/10.1109/JBHI.2020.3023246.

[24] D. Singh, V. Kumar, Vaishali, M. Kaur, Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks, Eur. J. Clin. Microbiol. Infect. Dis. 39 (2020) 1379–1389. https://doi.org/10.1007/s10096-020-03901-z.

[25] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, M. Kaur, Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning, J. Biomol. Struct. Dyn. (2020) 1–8. https://doi.org/10.1080/07391102.2020.1788642.

[26] S. El-Bana, A. Al-Kabbany, M. Sharkas, A multi-task pipeline with specialized streams for classification and segmentation of infection manifestations in COVID-19 scans, PeerJ Comput. Sci. 6 (2020) e303. https://doi.org/10.7717/peerj-cs.303.

[27] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conf. Comput. Vis. Pattern Recognit., 2016: pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[28] X. Zhang, J. Zou, K. He, J. Sun, Accelerating Very Deep Convolutional Networks for Classification and Detection, (2015). http://arxiv.org/abs/1505.06798 (accessed October 9, 2020).

[29] R. Riazi, M. Asrardel, M. Shafaei, S. Vakilipour, H. Zare, H. Veisi, A Data Mining Study on Combustion Dynamics and NOx Emission of a Swirl Stabilised Combustor With Secondary Fuel Injection, Int. J. Heavy Veh. Syst. 24 (2016). https://doi.org/10.1504/IJHVS.2017.10005324.

[30] M. Rostami, K. Berahmand, S. Forouzandeh, A novel community detection based genetic algorithm for feature selection, Journal of Big Data. 8(2) (2021). https://doi.org/10.1186/s40537-020-00398-3.

[31] M.J. Ebadi, Alireza Hosseini, M.M. Hosseini, A projection type steepest descent neural network for solving a class of nonsmooth optimization problems, Neurocomputing 235 (2017) 164-181. https://doi:10.1016/j.neucom.2017.01.010.

[32] M. Rostami, K. Berahmand, S. Forouzandeh, A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty. J Big Data 7(83) (2020). https://doi.org/10.1186/s40537-020-00352-3.

[33] M. Rostami, S. Forouzandeh, K. Berahmand, M. Soltani, Integration of multi-objective PSO based feature selection and node centrality for medical datasets, Genomics. 112(6) (2020) 4370–4384. https://doi.org/10.1016/j.ygeno.2020.07.027.

[34] M.J. Ebadi, A. Ebrahimi. Video Data Compression by Progressive Iterative Approximation, International Journal of Interactive Multimedia and Artificial Intelligence, 6(6) (2021): 189-195. https://doi.org/10.9781/ijimai.2020.12.002.

[35] P. Saha, D. Mukherjee, P.K. Singh, et al., GraphCovidNet: A graph neural network based model for detecting COVID-19 from CT scans and X-rays of chest. Scientific Reports. 11, 8304 (2021). https://doi.org/10.1038/s41598-021-87523-1.

[36] V. Gupta, N. Jain, P. Katariya, A. Kumar, S. Mohan, A. Ahmadian, M. Ferrara, An emotion care model using multimodal textual analysis on COVID-19, Chaos, Solitons & Fractals. 144, 110708 (2021). https://doi.org/10.1016/j.chaos.2021.110708.
Highlights:

- To propose two efficient COVID-19 classification models based on convolutional neural network and convolutional auto-encoder neural network for classification.
- To implement the proposed models on a big dataset including COVID-19 and Non-COVID-19 CT Scans.
- The most important benefit of the proposed models in comparison the other models is lack of need to pre-trained networks and data augmentation.
Authors’ Contribution:

Saman Fouladi: Conceptualization, Methodology, Software

Ali A. Safaei and M. J. Ebadi: Data curation, Writing- Original draft preparation.

Mohd Yazid Bajuri, Ali Ahmadian: Supervision and Validation of Data- Reviewing Original draft
Conflict of Interest and Authorship Conformation Form

Please check the following as appropriate:

✓ All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

✓ This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

✓ The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.