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.
Covid-19 detection via deep neural network and occlusion sensitivity maps

Muhammad Aminu\textsuperscript{a}, Noor Atinah Ahmad\textsuperscript{a,∗}, Mohd Halim Mohd Noor\textsuperscript{b}

\textsuperscript{a}School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia
\textsuperscript{b}School of Computer Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia

Abstract

Deep learning approaches have attracted a lot of attention in the automatic detection of Covid-19 and transfer learning is the most common approach. However, majority of the pre-trained models are trained on color images, which can cause inefficiencies when fine-tuning the models on Covid-19 images which are often grayscale. To address this issue, we propose a deep learning architecture called CovidNet which requires a relatively smaller number of parameters. CovidNet accepts grayscale images as inputs and is suitable for training with limited training dataset. Experimental results show that CovidNet outperforms other state-of-the-art deep learning models for Covid-19 detection.

Key words: Covid-19, Pneumonia, Deep neural networks, Occlusion sensitivity maps

1. Introduction

Since its first discovery in Wuhan, Hubei province of China in December 2019, the coronavirus disease 2019 (Covid-19) has now spread to around 219 countries and territories, resulting in 1,204,028 deaths as of November 4th, 2020\textsuperscript{1}. Early screening of high risk and critically ill patients are prominent in reducing the rate of mortality in Covid-19 patients [1].

Reverse Transcription Polymerase Chain Reaction (RT-PCR) has widely been used for Covid-19 diagnosis. However, a major problem encountered when using the RT-PCR testing is the possibility of obtaining false positive and false negative results [2]. RT-PCR test for Covid-19 may be falsely negative due to low viral load, presence of substance that hinder chemical action in the sample and laboratory errors [2, 3]. Thus, RT-PCR result should not be considered as the only principle in patient management decision. In a study conducted by Li et al. [4] on 610 hospitalized

\textsuperscript{∗}Corresponding author.

Email addresses: muhammadaminu47@gmail.com (Muhammad Aminu), nooratinah@usm.my (Noor Atinah Ahmad), halimnoor@usm.my (Mohd Halim Mohd Noor)

\textsuperscript{1}https://www.who.int/emergencies/diseases/novel-coronavirus-2019

Preprint submitted to Elsevier

March 21, 2021
patients in Wuhan, clinically diagnosed with Covid-19. They found a potentially high false negative rate of RT-PCR testing for Covid-19 in the hospitalized patients. Their findings also indicate the need for clinical indicators such as chest X-ray and computed tomography (CT) images for Covid-19 diagnosis. Most recently, Herpe et al. [5] published a study in which they compare the results of chest CT and RT-PCR. The authors found out that a lot of Covid-19 patients might not be diagnose due to the low sensitivity of RT-PCR, which can lead to further spread of the disease. However, using the chest CT scan, several features of Covid-19 are detected in almost all the patients diagnosed with Covid-19. Patients with clinical symptoms who have negative RT-PCR results were found to have Covid-19 features as captured by the chest CT. Due to the large number of individuals at high risk of Covid-19, and with the limited number of radiologist, regular screening of Covid-19 from chest CT and X-ray could cause significant workload challenges. Artificial intelligence techniques have the potential to aid radiologists in Covid-19 screening by eradicating various uncertainties and reducing reading time.

Artificial intelligence techniques have attracted a lot of attention in the fight against Covid-19 [6–9]. Various artificial intelligence techniques, especially deep learning approaches have been successfully applied in Covid-19 diagnosis [10–13]. Since Covid-19 samples are often very limited, transfer learning becomes the most widely used approach for Covid-19 diagnosis [14–17]. However, majority of the pre-trained models are trained on the ImageNet database [18], which contains color images. But Covid-19 chest X-ray and CT scan images are often grayscale which necessitates the pre-processing of these images to conform with the size and structure of the original color images used to initially train the model. It also remains unclear whether the filters learned by the model on the original color images are fully exploited in the transfer learning.

In this paper, we present a solution to make Covid-19 detection using deep learning feasible in situations where training images are limited and grayscale. We propose a new deep learning architecture called Covid-19 detection neural network (CovidNet), for Covid-19 diagnosis. CovidNet uses grayscale images and is suitable for training with limited training dataset. We use $L^2$ regularization and global average pooling layer to control the model complexity.

The following points highlight the main contributions of this paper:

1. A new deep learning architecture based on convolutional neural network is proposed to detect Covid-19. The proposed architecture has relatively small number of parameters and is suitable for training with limited training data. We used the $L^2$ regularization approach and global average pooling layer to tackle the problem of model complexity (overfitting) caused by limited training data.

2. Another important contribution of this paper is the used of occlusion sensitivity maps to visualize the image features CovidNet uses to make classification decisions. Gradient-weighted class activation map (Grad-CAM) has widely been used to depict decisions made by deep learning models for Covid-19 diagnosis using chest X-ray images [10, 19, 20]. However, a major issue with the Grad-CAM method is that it only focuses on broad main lesion, the method
cannot identify multifocal lesions in the image [13]. With the occlusion sensitivity maps, multifocal ground glass opacities (GGO) and consolidations were visualized effectively.

3. We investigate the effectiveness of CovidNet for deep feature extraction. Features extracted using CovidNet are fed to several classifiers including KNN, SVM and RF for classification. The features extracted at the different convolutional layers of the proposed CovidNet are also analyzed and visualized.

4. In order to obtain good performances of the different classifiers, we used Bayesian optimization approach to select the optimal parameters for the different classifiers. The Bayesian optimization approach proves effective in selecting the optimal parameters, which leads to better classification performances.

5. Extensive experiments are carried out on both chest X-ray and CT scan images to demonstrate the effectiveness of the proposed CovidNet. In this context, we compared CovidNet with other state-of-the-art architectures from the perspective of deep feature extraction and classification of Covid-19 chest X-ray and CT scan images.

The remainder of the paper is organized as follows: In section 2, we provide a brief review of some methods proposed for Covid-19 detection. Section 3 introduces our proposed CovidNet. The extensive experimental results are presented in section 4. Finally, we provide some discussions and concluding remarks in section 5.

2. Related work

Deep learning have recently received a lot of attention in fighting the Covid-19 pandemic. A lot of deep learning approaches have recently been proposed as Covid-19 diagnostic aids to help physicians for better medical decision making. In this section, we present some studies that are closely related to this paper.

2.1. Signal processing for Covid-19

Shortness of breath and coughing are two of the most widely seen symptoms in patients with Covid-19. Several studies have used cough and breathing pattern to detect Covid-19. This requires the collection of Covid-19 and non Covid-19 cough and breathing sounds so as to train an artificial intelligence model that can detect if a patient has Covid-19 using the trained model. Moreover, relevant data needs to be collected in order to train such a model. A lot of effort has been put forward by many researchers to collect cough and breathing audio samples using simple smartphone microphones such as done by the Cambridge University. The university provides a web based platform for participants to record their speech and cough sounds, which will be used to build an App that can detect if a person is suffering

2https://www.covid-19-sounds.org/en/
from Covid-19. A similar App called CoughVid\(^3\) has been proposed by researchers from Ecole Polytechnique Fédrale de Lausanne (EPFL) to detect Covid-19 cough from other coughs such as ones related to the normal and seasonal allergies. Imran et al. [21] develop a mobile app called AI4COVID-19 for the diagnosis of Covid-19 from cough sounds. Their app record cough sounds when prompted and then send it to an artificial intelligence engine running in the cloud. The artificial intelligence engine analyzes the cough sound and return a result indicating whether the cough sound is Covid-19 likely, Covid-19 not likely or Test inconclusive. Halgurd et al. [22] proposed using built-in smart phone sensors to detect Covid-19. They develop an artificial intelligence enabled framework that reads smartphone sensor signal measurements to detect Covid-19.

2.2. Machine learning for Covid-19

Many machine learning techniques has been utilized recently for the prediction and detection of Covid-19. Rustam et al. [23] uses machine learning models to forecast the number of upcoming patients affected by Covid-19. In particular, the authors used four forecasting models including support vector machines (SVM), least absolute shrinkage and selection operator (LASSO), exponential smoothing (ES) and linear regression to forecast the threatening factors of Covid-19. In [24], the authors used support vector regression (SVR) to predict the number of Covid-19 cases for several affected countries. They also analyze the effect of different hyperparameter settings on the performance of the SVR model. Pinter et al. [25] used a hybrid machine learning model of multi-layered perceptron-imperialist competitive algorithm (MLP-ICA) and adaptive network-based fuzzy inference system (ANFIS) to predict covid-19 infection and mortality rate in Hungary. Burdick et al. [26] considers the problem of identifying patient decompensation. They use machine learning techniques to predict respiratory decompensation in Covid-19 patients. Specifically, a machine learning algorithm was used for the prediction of invasive mechanical ventilation for Covid-19 patients within 24 hours of an initial encounter. The algorithm was able to demonstrate accurate prediction of the need for mechanical ventilation within the next 24 hours. In addition, Elaziz et al. [27] extracted features from chest X-ray images using new fractional multichannel exponent moments (FrMEMs). A modified manta-ray foraging optimization (MRFO) based on differential evolution is used to select the most relevant features from the extracted ones. Finally, the nearest neighbor (KNN) classifier is used to classify a given chest X-ray image as Covid-19 or normal case. Sara et al. [28] compared several deep learning based feature extraction techniques for covid-19 classification. The authors used several deep learning based feature extraction frameworks to extract the most relevant features which are then fed into several machine learning classifiers such as decision tree (DT) and random forest (RF) to classify samples as either Covid-19 or a control case.

\(^3\)https://coughvid.epfl.ch/
2.3. Deep learning for Covid-19

Deep learning has also been widely adopted as a powerful tool for the detection of Covid-19. Deep learning has demonstrated great success in the detection of Covid-19 due to its high capability of significant feature extraction. Li et al. [10] develop a three-dimensional deep learning model called Covid-19 detection neural network (COVNet) that extract visual features from chest CT images to detect Covid-19. Their model was able to detect Covid-19 and differentiate it from community acquired pneumonia (CAP) and other lung diseases. Brunese et al. [20] transferred the VGG-16 deep learning model pre-trained from ImageNet for Covid-19 detection. Ozturk et al. [6] proposed a deep learning model called DarkCovidNet for automatic Covid-19 detection from chest X-ray images. The DarkCovidNet architecture was designed based on the DarkNet-19 [29] architecture which has proven successful in deep learning. DarkCovidNet was developed for binary classification (Covid vs No-Findings) and multi-class classification (Covid vs Pneumonia vs No-Findings). The model has also proven efficient with 98.08% accuracy for the binary cases and 87.02% accuracy for the multi-class cases. Khan, Shah and Bhat [7] proposed a deep convolutional neural network model called CoroNet to automatically detect Covid-19 from chest X-ray images. CoroNet was designed based on the Xception CNN architecture [30] pre-trained on the ImageNet dataset. CoroNet perform significantly well in detecting Covid-19 from chest X-ray images and in distinguishing Covid-19 from other types of Pneumonia such as bacterial pneumonia and viral pneumonia. Apostolopoulos and Mpesiana [31] conduct a study to evaluate the effectiveness of the state-of-the-art pre-trained convolutional neural networks for the automatic detection of Covid-19 from thoracic X-ray images. In their study, they compared five state-of-the-art convolutional neural networks, including VGG19 [32], MobileNet v2 [33], Inception [34], Xception [30], and Inception ResNet v2 [34]. Based on their experimental study, they found that VGG19 achieves better accuracy. However, the MobileNet v2 outperform VGG19 in terms of specificity. Thus, MobileNet v2 turns out to be the most effective model for their study. Mahmud, Rahman and Fattah [19] also proposed a deep convolutional neural network called CovXNet for automatic detection of Covid-19 and other traditional pneumonia from chest X-ray images. CovXNet uses depthwise convolution with varying dilation rates to efficiently extract significant features from chest X-ray images. The authors also used a gradient-based discriminative localization to identify the abnormal regions of the chest X-ray images. More studies on the application of deep learning approaches for Covid-19 diagnosis can be found in [8, 35, 36].

3. Proposed method

3.1. Problem formulation

Given chest X-ray images as training datasets, our aim is to build a deep learning model that returns a three dimensional vector for multi-class case (Covid-19 vs Pneumonia vs normal) or a two dimensional vector for the binary case (Covid-19 vs normal), each element of which represents the probability of the input chest X-ray image belonging to a particular class. We aim at developing an efficient and effective deep learning model that can provide accurate predictions.
Figure 1: CovidNet architecture
| Layer (type) | Output shape | # Filters | Filter size | Stride | Padding | Padding type | Pool size | # Parameters |
|-------------|--------------|-----------|-------------|--------|---------|--------------|-----------|--------------|
| Conv(7 × 7) | [448, 448, 8]| 8         | [7 7]       | [1 1]  | 0 0 0 0 | same         | -         | 400          |
| BN          | [448, 448, 8]| -         | -           | -      | -       | -            | -         | 16           |
| ReLU        | [448, 448, 8]| -         | -           | -      | -       | -            | -         | 0            |
| Max Pooling | [224, 224, 8]| -         | [2 2]       | [0 0 0 0] | manual   | [2 2]        | 0         |
| Conv(3 × 3) | [224, 224, 16]| 16       | [3 3]       | [1 1]  | 0 0 0 0 | same         | -         | 1168         |
| BN          | [224, 224, 16]| -         | -           | -      | -       | -            | -         | 16           |
| ReLU        | [224, 224, 16]| -         | -           | -      | -       | -            | -         | 0            |
| Max Pooling | [112, 112, 16]| -         | [2 2]       | [0 0 0 0] | manual   | [2 2]        | 0         |
| Conv(3 × 3) | [112, 112, 32]| 32       | [3 3]       | [1 1]  | 0 0 0 0 | same         | -         | 4640         |
| BN          | [112, 112, 32]| -         | -           | -      | -       | -            | -         | 64           |
| ReLU        | [112, 112, 32]| -         | -           | -      | -       | -            | -         | 0            |
| Max Pooling | [56, 56, 32]| -         | [2 2]       | [0 0 0 0] | manual   | [2 2]        | 0         |
| Conv(3 × 3) | [56, 56, 64]| 64       | [3 3]       | [1 1]  | 0 0 0 0 | same         | -         | 18496        |
| BN          | [56, 56, 64]| -         | -           | -      | -       | -            | -         | 128          |
| ReLU        | [56, 56, 64]| -         | -           | -      | -       | -            | -         | 0            |
| Max Pooling | [28, 28, 64]| -         | [2 2]       | [0 0 0 0] | manual   | [2 2]        | 0         |
| Conv(3 × 3) | [28, 28, 128]| 128      | [3 3]       | [1 1]  | 0 0 0 0 | same         | -         | 73856        |
| BN          | [28, 28, 128]| -         | -           | -      | -       | -            | -         | 256          |
| ReLU        | [28, 28, 128]| -         | -           | -      | -       | -            | -         | 0            |
| Max Pooling | [14, 14, 128]| -         | [2 2]       | [0 0 0 0] | manual   | [2 2]        | 0         |
| Conv(3 × 3) | [14, 14, 256]| 256      | [3 3]       | [1 1]  | 0 0 0 0 | same         | -         | 295168       |
| BN          | [14, 14, 256]| -         | -           | -      | -       | -            | -         | 512          |
| ReLU        | [14, 14, 256]| -         | -           | -      | -       | -            | -         | 0            |
| Max Pooling | [7, 7, 256]| -         | [2 2]       | [0 0 0 0] | manual   | [2 2]        | 0         |
| Layer (type) | Output shape | # Filters | Filter size | Stride | Padding | Padding type | Pool size | # Parameters |
|--------------|--------------|-----------|-------------|--------|----------|--------------|-----------|--------------|
| Conv(3 × 3)  | [7, 7, 512]  | 512       | [3 3]       | [1 1]  | [0 0 0 0] same | -          | 1180160    |
| BN           | [7, 7, 512]  | -         | -           | -      | -        | -            | -         | 1024         |
| ReLU         | [7, 7, 512]  | -         | -           | -      | -        | -            | -         | 0            |
| Max Pooling  | [3, 3, 512]  | -         | -           | [2 2]  | [0 0 0 0] manual | [2 2]     | 0           |
| Conv(3 × 3)  | [3, 3, 1024] | 1024      | [3 3]       | [1 1]  | [0 0 0 0] same | -          | 4719616    |
| BN           | [3, 3, 1024] | -         | -           | -      | -        | -            | -         | 2048         |
| ReLU         | [3, 3, 1024] | -         | -           | -      | -        | -            | -         | 0            |
| Gap          | [1, 1, 1024] | -         | -           | -      | -        | -            | -         | 0            |
| FC           | [1, 1, 512]  | -         | -           | -      | -        | -            | -         | 524800       |
| BN           | [1, 1, 512]  | -         | -           | -      | -        | -            | -         | 1024         |
| ReLU         | [1, 1, 512]  | -         | -           | -      | -        | -            | -         | 0            |
| FC           | [1, 1, 3]    | -         | -           | -      | -        | -            | -         | 1539         |
| SM           | [1, 1, 3]    | -         | -           | -      | -        | -            | -         | 0            |
3.2. CovidNet structure

CovidNet is a convolutional neural network designed for the detection of Covid-19 from chest X-ray images. It finds and learn patterns in chest X-ray images to classify the images. To reflect the original images better as well as to capture as many patterns in the images as possible, we design the input layer in CovidNet to consider grayscale images of size $448 \times 448$. Specifically, we resize all the chest X-ray images to $448 \times 448$ and also transformed them to grayscale. The architecture of CovidNet is illustrated in Fig. 1. Architecture details of CovidNet model are also shown in Table 1. We can observe from Fig. 1 that the CovidNet consist of seven consecutive blocks.

**4-layer blocks.** As the name implies, a 4-layer block consist of 4 layers. Specifically, it is composed of a $3 \times 3$ convolutional layer (Conv($3 \times 3$)) followed by a BN, ReLU and a max pooling layer. Here, BN denotes a batch normalization layer. There are seven 4-layer blocks used in the CovidNet architecture. The seventh block uses a global average pooling layer to minimize overfitting by reducing the total number of parameters in the network. Thus, the seventh block consists of a Conv($3 \times 3$)-BN-ReLU sequence followed by a global average pooling layer.

**First layers.** As illustrated in Fig. 1, the early layers of the CovidNet consist of $7 \times 7$ convolutional layer (Conv($7 \times 7$)) followed by a BN, ReLU and a max pooling layer.

**Last layers.** For the last layers, constituting the final stage of the CovidNet, we used a sequence of FC-BN-ReLU-FC-SM to produce the probabilities, $p^T$ for the predicted classes. Here, FC and SM denotes a fully connected layer and a softmax layer, respectively.

3.3. CovidNet implementation and training

Two different settings of the CovidNet were implemented to detect and classify Covid-19 images. First, we trained CovidNet to classify chest X-ray images into three categories: Covid-19, Pneumonia and normal, and into two categories: Covid-19 and normal. Secondly, the CovidNet is used to extract deep features from CT images, which are then fed into several classifiers for classification.

The CovidNet model was implemented in MATLAB R2020a. All of our experiments are carried out on an Intel core i7 3.20GHz windows 10 machine with Nvidia GeForce GTX 1050 graphics card (and 8GB memory). The proposed CovidNet was trained on the datasets using Adam optimizer [37] with a learning rate of 0.0001, batch size of 128 and an epoch value of 5. The training data is shuffled before each training epoch. Note that, in our preliminary experiments, we also train CovidNet using a larger epoch value (20), but we did not see any significant performance improvement compared to the epoch value of 5. So we used an epoch value of 5 throughout our experiments. We evaluate the performance of CovidNet using the holdout validation approach for both the multi-class and binary classification tasks. The datasets are randomly split into training and validation set. We used 80% of the datasets for training and the remaining 20% are considered as validation sets. We repeat the experiments five times and report the results obtained by CovidNet for the five different splits.
4. Experimental results

We empirically demonstrate the effectiveness of CovidNet using both chest X-ray and CT image datasets and compare with state-of-the-art models. For the purpose of reproducibility, we provide the MATLAB codes used in our experiments at: https://github.com/muhammadaminu47/CovidNet.

![Sample chest X-ray images](image)

### 4.1. Datasets

One major challenge encountered in the detection of Covid-19 using deep learning approaches is the scarcity of Covid-19 patients chest X-ray and CT images. The publicly available Covid-19 chest X-ray and CT images are limited which hinders the performance of the deep learning models. In our experiments, we used both chest X-ray and CT scan images to study the performance of CovidNet. For the experimental study on chest X-ray images, we consider two datasets used in two different studies. The first one is the dataset used in [6], while the second one is the dataset considered by authors in [7]. All datasets are freely available for research purposes. Combining the chest X-ray images of patients with Covid-19 in the two datasets and using images belonging to the normal class in [7] and Pneumonia class in [6], we form a dataset with three different classes. Overall, a total of 1266 chest X-ray images were collected in this study. Among them, 321 chest X-ray images were from patients with Covid-19, 500 chest X-ray images were from patients with Pneumonia, and 445 chest X-ray
images were from healthy (normal) people. Figure 2 shows some sample images in the prepared dataset.

For the experimental study on CT images, we used the dataset from [38]. To the best of our knowledge, this is the largest publicly available Covid-19 CT image dataset. The dataset contains 2482 CT scan images from 120 patients, in which 1252 CT scan images belongs to 60 patients with Covid-19 and 1230 CT scan images belongs to 60 non Covid-19 patients. Some samples images from the datasets are shown in Fig. 4.

4.2. Performance metrics

The performance of the proposed CovidNet is evaluated using confusion matrix. We obtained the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) from the confusion matrix, and five performance metrics are computed as follows:

1. Accuracy = \[
\frac{(TP+TN)}{(TP+TN+FP+FN)}
\]

2. Precision = \[
\frac{(TP)}{(TP+FP)}
\]

3. Sensitivity/Recall = \[
\frac{(TP)}{(TP+FN)}
\]

4. Specificity = \[
\frac{(TN)}{(TN+FP)}
\]
5. F1-score = \( \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \)

4.3. Experiments on CovidNet using end-to-end learning

In this experiments, we used our prepared chest X-ray dataset to examine the effectiveness of our proposed CovidNet in an end-to-end fashion. The classification performance of CovidNet for the multi-class and binary classification problems on the chest X-ray images are provided in Table 2 and 3, respectively. The confusion matrices for the multi-class and binary classification problems are shown in Fig. 6 and 7, respectively. Fig. 4 and 5 shows the accuracy and loss plots obtained on split 1 for the multi-class and binary classification problems, respectively. We also present a performance comparison of our proposed CovidNet with other existing deep learning models in Table 4.

![Accuracy and Loss Plots](image)

Figure 4: Training and validation accuracy and loss plots for split 1. These results are for the multi-class case.
Figure 5: Training and validation accuracy and loss plots for split 1. These results are for the binary case.

Table 2: Performance of CovidNet for each split of the dataset. The results presented here are for the multi-class case.

| Splits   | Sensitivity | Specificity | Precision | F1-score | Accuracy |
|----------|-------------|-------------|-----------|----------|----------|
| Split 1  | 92.19       | 98.41       | 96.16     | 93.65    | 96.84    |
| Split 2  | 93.75       | 97.88       | 93.75     | 93.75    | 96.44    |
| Split 3  | 87.50       | 97.88       | 93.33     | 90.32    | 95.26    |
| Split 4  | 84.38       | 98.94       | 96.43     | 90.00    | 95.26    |
| Split 5  | 87.50       | 98.94       | 96.55     | 91.80    | 95.26    |
| Average  | 89.06       | 98.41       | 95.24     | 91.90    | 95.81    |

Table 3: Performance of CovidNet for each split of the dataset. The results presented here are for the binary case.

| Splits   | Sensitivity | Specificity | Precision | F1-score | Accuracy |
|----------|-------------|-------------|-----------|----------|----------|
| Split 1  | 100         | 100         | 100       | 100      | 100      |
| Split 2  | 100         | 98.88       | 98.46     | 99.22    | 99.35    |
| Split 3  | 100         | 100         | 100       | 100      | 100      |
| Split 4  | 100         | 100         | 100       | 100      | 100      |
| Split 5  | 100         | 100         | 100       | 100      | 100      |
| Average  | 100         | 99.78       | 99.69     | 99.84    | 99.87    |
Figure 6: The aggregate and 5 splits confusion matrices for the multi-class classification problem.
Figure 7: The aggregate and 5 splits confusion matrices for the binary classification problem.
Table 4: Performance comparison of CovidNet with other architectures.

| Study                  | Modality | Subjects                                      | Architecture       | Accuracy (%) | Parameters ($10^6$) |
|------------------------|----------|-----------------------------------------------|--------------------|--------------|---------------------|
| Hemdan et al. [39]     | X-ray    | 25 Covid-19 + 25 normal                       | COVIDX-Net         | 90           | 143                 |
| Sethy and Behera [40]  | X-ray    | 25 Covid-19 + 25 Non-Covid-19                 | ResNet-50          | 95.38        | 25.6                |
| Narin et al. [41]      | X-ray    | 50 Covid-19 + 50 Non-Covid-19                 | ResNet-50          | 98           | 25.6                |
| Ozturk et al. [6]      | X-ray    | 125 Covid-19 + 500 No-Findings               | DarkCovidNet       | 98.08        | 1.1                 |
| Ozturk et al. [6]      | X-ray    | 125 Covid-19 + 500 Pneumonia + 500 No-Findings| DarkCovidNet       | 87.02        | 1.1                 |
| Butt et al. [42]       | CT       | 219 Covid-19 + 500 Influenza-A viral Pneumonia + 175 Healthy | ResNet-18         | 86.7         | 11                  |
| Apostolopoulos and Mpesiana [31] | X-ray | 224 Covid-19 + 700 Pneumonia + 504 Healthy | VGG19              | 93.48        | 143                 |
| Oh et al. [13]         | X-ray    | 284 Covid-19 + 310 normal + 657 Pneumonia    | FC-DenseNet/ResNet-18 | 88.9         | 11.6                |
| Khan et al. [7]        | X-ray    | 284 Covid-19 + 310 normal                     | CoroNet            | 99           | 33                  |
| Khan et al. [7]        | X-ray    | 284 Covid-19 + 310 normal + 657 Pneumonia    | CoroNet            | 94.59        | 33                  |
| Ardakani et al. [36]   | CT       | 510 Covid-19 + 510 Non-Covid-19              | ResNet-101         | 99.51        | 44.6                |
| Ardakani et al. [36]   | CT       | 510 Covid-19 + 510 Non-Covid-19              | Xception           | 99.02        | 22.9                |
| Our proposed method    | X-ray    | 321 Covid-19 + 445 normal                     | CovidNet           | 99.87        | 6.8                 |
| Our proposed method    | X-ray    | 321 Covid-19 + 500 Pneumonia + 445 normal    | CovidNet           | 95.81        | 6.8                 |
The main observations from the experimental results includes:

1. Our proposed CovidNet is very effective in Covid-19 diagnosis. CovidNet performs significantly well for both the multi-class and binary classification problems. In particular, CovidNet achieved a classification accuracy of 96.84% and 100% for the multi-class and binary classification problems, respectively. Moreover, CovidNet achieved a sensitivity of 92.19% and 100% for the multi-class and binary classification problems, respectively. These results were obtained on the first split (split 1) of the dataset. Since the sensitivity of Covid-19 diagnosis from chest X-ray images for clinicians is 69% and that of RT-PCR is 91% [43] and 71% as reported in [44], we can deduce that CovidNet performed significantly well for Covid-19 diagnosis.

2. The proposed CovidNet consider the possibility of overfitting which is especially important, given that the training data is limited due to difficulty in the collection of Covid-19 chest X-ray images. The global average pooling layer is used to control the model complexity by reducing the total number of parameters in the model. We also add a regularization term to the loss function ($L_2$ regularization technique) to control the complexity of the model. It can be seen from Fig. 4 and 5 that, the proposed CovidNet does not overfit the data. This clearly indicates the efficacy of the proposed CovidNet.

3. Considering both accuracy and efficiency, CovidNet is the best choice among all the compared models. CovidNet obtained overall accuracies of 96.84% and 100% for the 3-class and 2-class classification problems, which are better than the classification accuracies obtained by all the compared models (see Table 4). Furthermore, CovidNet uses less number of parameters (6,824,947) compared to the other models except the DarkCovidNet which has a total of 1,164,434 parameters. This is a significant advantage which makes the CovidNet gain significant computational saving on time. It also help improve the performance and stability of CovidNet with the limited data.

4.4. Visualizations using occlusion sensitivity

In this experiment, we used occlusion sensitivity maps [45] to find out which parts of the X-ray image are most important for CovidNet classification decision. Occlusion sensitivity is a simple technique used to gain an understanding of what image features a deep neural network uses to make a classification decision. Specifically, occlusion sensitivity systematically occludes different portions of the input image with an occluding mask (usually a gray square), and measure the change in probability score for a given class as a function of the mask position. Figure 8 shows some Covid-19 input chest X-ray images and the occlusion sensitivity maps overlaid on the input images. The occlusion sensitivity maps show parts of the chest X-ray image that provides a higher contribution to the score for the Covid-19 class, and which parts have less or no contributions. By carefully examining the occlusion maps, it can be seen that CovidNet was able to localize regions with ground glass opacities (GGO),
consolidative opacities and peripheral and diffuse distribution of such opacities. Moreover, vascular thickening along with other traditional viral features were localized by CovidNet. These localized features are also commonly seen features in chest X-ray images of patients with Covid-19 [46–49]. The visualization results also support our claim that the occlusion sensitivity maps are more intuitive and interpretable compared to the Grad-CAM method. Using the occlusion sensitivity maps, multifocal GGO’s and consolidations were visualized effectively as shown in Fig. 8(d).

4.5. Experiments on deep feature extraction using CovidNet

In this experiment, we investigate the effectiveness of our proposed architecture in deep feature extraction. It is important for the proposed architecture to extract rich discriminative features that carries as much discriminative information in the images as possible. Using the CT scan images [38], we present the features extracted by the different convolutional layers of CovidNet in Fig. 9. The different convolutional layers extract features from different perspective. For example, the first convolutional layer extract basic image features such as spots and edges. The deeper layers extract much higher level features by combining the earlier features. It can be seen from Fig. 9 that the features becomes more abstract after the second convolutional layer. Features extracted using the deeper layers are better suited for classification as they contain information from all the preceding layers [50]. As a result, we used features extracted from the first fully connected layer of CovidNet for our experimental study.

First, we randomly split the dataset into training set consisting of 80% of the data and a test set consisting of the remaining 20% of the data. Feature vectors are then extracted from the first fully connected layer of CovidNet. The extracted training features are then used to train several classifiers including KNN, SVM and RF. There are parameters in the different classifiers that needs to be tuned prior to training. Although these parameters can be tuned manually, this approach is time consuming and can reduce the performance of the different classifiers. To overcome the challenges associated with hyperparameter tuning in the different classifiers, we used the Bayesian optimization approach [51] to determine the optimal parameters. Bayesian optimization is an efficient and automated approach for parameter selection. The method uses a guided sampling strategy to determine the best parameter from a list of possible candidates (parameter space) and testing the chosen parameter on different subsets of the dataset. Figures 14 to 18 shows the Bayesian optimization results for the different classifiers. The objective function is the five fold cross-validated loss of the different classifiers.

After training the different classifiers using the optimal parameters as determined by the Bayesian optimization. We then evaluate the performance of the classifiers on the test set. Several performance metrics including sensitivity, specificity, precision, F1-score, area under curve (AUC) and classification accuracy are utilized to evaluate the performance of the classifiers. We also compare the effectiveness of features extracted using CovidNet to those extracted using other state-of-the-art architectures. Specifically, we extracted features from the fully connected layers of AlexNet, ResNet50, GoogleNet, MobileNetv2 and Inceptionv3, which are then fed to the different classifiers for classification. Performance comparison of the different classifiers
with features extracted using the different architectures are presented in tables 5 to 7. As can be seen from these tables, we repeat the random splitting of the dataset five

Figure 8: Examples of occlusion sensitivity maps on the Covid-19 class
times and report the classification results obtained on each split together with the average classification results over the five random splits. The misclassification rates of the different methods on split 1 are also reported in form of confusion matrices in Figs. 11 to 13. Figure 10 shows the ROC curves for the different methods on split 1.

The main observations from the experimental results includes:

1. Our proposed CovidNet provides an effective approach for discriminant feature extraction. Features extracted using CovidNet are very effective for classification. The different classifiers produces the best classification results when fed in features extracted using CovidNet (see tables 5 to 7). On split 1, the KNN classifier was able to classify all the test samples correctly i.e., with 100% accuracy. Only one test sample is misclassified using the SVM and RF classifiers (see Fig. 12 and 13). On the other hand, the different classifiers misclassifies a lot of samples when fed in features extracted using the other architectures. For example, utilizing features extracted using AlexNet, GoogleNet, MobileNetv2, Inceptionv3 and ResNet50, the RF classifier misclassified 30, 38, 20, 45 and 27 samples, respectively. The excellent performance of the different classifiers when fed in features extracted using CovidNet can also be reflected from the ROC curves in Fig. 10. Features extracted using CovidNet results in significantly higher AUC values as compared to the other architectures. For all the different splits, the KNN, SVM and RF classifiers achieves significantly better performances with features extracted using CovidNet than with features extracted using the other architectures. This suggest the effectiveness of the features extracted using our proposed CovidNet.

2. Utilizing features extracted using CovidNet, the KNN classifier produces better results compared to the SVM and RF classifiers. For example, with features extracted using CovidNet on split 1, the KNN classifier was able to obtain a sensitivity, specificity, precision, F1-score, AUC and accuracy rates of 100%, respectively. The average sensitivity, specificity, precision, F1-score, AUC and accuracy rates for the KNN classifier when fed in features extracted using CovidNet are 100%, 99.92%, 99.92%, 99.96%, 99.96% and 99.96%, respectively. The results obtained using the KNN classifier are slightly better than the results produce by the SVM and RF classifiers.

3. The Bayesian optimization significantly improves the performances of the different classifiers. For example, using the optimal parameters determined by the Bayesian optimization algorithm on split 1, the classification accuracy rates of the optimized KNN, SVM and the RF classifiers when fed in features extracted using CovidNet are 100%, 99.80% and 99.80%, respectively. However, on the same split of the dataset with unoptimized parameters (MATLAB default parameters), the KNN, SVM and the RF classifiers produces classification accuracy rates of 99.40%, 86.29% and 99.19%, respectively. The performance of the Bayesian optimized classification models are significantly better than the unoptimized ones.
4. Apart from optimizing classifier performance, the Bayesian optimization approach can be used to compare the influence of the different parameters on the the classifier performance. It can be observed from Fig. 15 that distance metric had greater impact on the performance of KNN than did the number of neighbors. As for the SVM classifier, it is observed from Fig. 17 that varying the kernel scale had more influence on the classifier performance than did box-constraint. This was seen in all methods, which indicate great emphasis need to be placed on this important parameters. In our experiments with features extracted using CovidNet, the best KNN performance on split 1 is achieved when the distance metric is spearman and the number of neighbors is 3. As for the SVM classifier, the best performance on split 1 is achieved when the kernel scale is 12.96 and the box-constraint is 73.64. The minimum leaf size that gives the best performance of the the RF classifier on split 1 is 58. The excellent performances of the different classifiers using these optimal parameters can also be reflected from Figs. 14, 16 and 18.

5. Discussion and conclusion

Proper design of deep learning architecture is a crucial issue that needs to be taking into consideration while using deep learning approaches for Covid-19 diagnosis. Several complex deep learning architectures with large number of parameters have been proposed to detect Covid-19 using chest X-ray or CT scan images [7, 36, 39]. However, it is well known that large number of parameters affect classification accuracy and also increases computational complexity [52]. In this paper, we have developed a deep learning architecture based on convolutional neural network for Covid-19 detection. In the design of our proposed architecture, we used only eight convolutional layers with $3 \times 3$ convolution kernel except for the first convolutional layer that has a $7 \times 7$ convolution kernel. After each convolutional layer, we add a pooling layer with $2 \times 2$ pooling kernel to reduce the dimensionality of the convolutional layer output and to preserve relevant features while getting rid of redundant ones. To tackle the issue of overfitting which is usually common in situations where there is limited data, we adopt $L$-2 regularization and the global average pooling layer. This strategy proves effective for accurate Covid-19 diagnosis. The proper design of our proposed architecture leads to an excellent performance even with the limited training data.

To further exploit the effectiveness of the proposed CovidNet, we used features extracted using CovidNet as inputs to several classifiers. Specifically, we extract deep features using CovidNet and fed them to several classifiers including KNN, SVM and RF for classification. We used the Bayesian optimization approach to select optimal parameters for the different classifiers. The effectiveness of features extract using our proposed CovidNet is compared to that of several state-of-the-art architectures including ResNet50, Inceptionv3, MobileNetv2, GoogleNet and AlexNet. CovidNet outperforms the other state-of-the-art architectures considering both effectiveness and efficiency.
Figure 9: The Visual representations of features extracted at the different convolutional layers of CovidNet.
Figure 10: ROC curves of the different methods using the KNN, SVM and RF classifiers.
Figure 11: Misclassification rates of the KNN classifier when fed in features extracted using the different architectures. These results are obtained on split 1.
Figure 12: Misclassification rates of the SVM classifier when fed in features extracted using the different architectures. These results are obtained on split 1.
Figure 13: Misclassification rates of the RF classifier when fed in features extracted using the different architectures. These results are obtained on split 1.
Figure 14: Plots of cross-validated loss against function evaluations on split 1. (a) KNN fed with features extracted using AlexNet. (b) KNN fed with features extracted using CovidNet. (c) KNN fed with features extracted using GoogleNet. (d) KNN fed with features extracted using MobileNetv2. (e) KNN fed with features extracted using Inceptionv2. (f) KNN fed with features extracted using ResNet50.
Figure 15: Objective function model showing the space of number of neighbors and distance metric in terms of KNN performance on split 1.
Figure 16: Plots of cross-validated loss against function evaluations on split 1. (a) SVM fed with features extracted using AlexNet. (b) SVM fed with features extracted using CovidNet. (c) SVM fed with features extracted using GoogleNet. (d) SVM fed with features extracted using MobileNetv2. (e) SVM fed with features extracted using Inceptionv2. (f) SVM fed with features extracted using ResNet50.
Figure 17: Objective function model showing the space of kernel scale and boxconstraint in terms of SVM performance on split 1.
Figure 18: Plots of cross-validated loss against function evaluations on split 1. (a) RF fed with features extracted using AlexNet. (b) RF fed with features extracted using CovidNet. (c) RF fed with features extracted using GoogleNet. (d) RF fed with features extracted using MobileNetv2. (e) RF fed with features extracted using Inceptionv2. (f) RF fed with features extracted using ResNet50.
Table 5: Performance comparison on CT images. Here we used KNN to classify features extracted using the different architectures.

| Architecture | Splits | Performance metrics (%) |       |       |       |       |
|--------------|--------|--------------------------|-------|-------|-------|-------|
|              |        | Sensitivity | Specificity | Precision | F1-score | AUC | Accuracy |
| AlexNet      | Split 1 | 97.20       | 94.31       | 94.55      | 95.86    | 95.75 | 95.77    |
|              | Split 2 | 97.60       | 95.53       | 95.69      | 96.63    | 96.56 | 96.57    |
|              | Split 3 | 96.80       | 95.53       | 95.65      | 96.22    | 96.16 | 96.17    |
|              | Split 4 | 96.40       | 93.90       | 94.14      | 95.26    | 95.15 | 95.16    |
|              | Split 5 | 99.20       | 93.09       | 93.58      | 96.31    | 96.14 | 96.17    |
|              | Average | 97.44       | 94.47       | 94.72      | 96.06    | 95.95 | 95.96    |
| CovidNet     | Split 1 | 100         | 100         | 100        | 100      | 100   | 100      |
|              | Split 2 | 100         | 100         | 100        | 100      | 100   | 100      |
|              | Split 3 | 100         | 99.59       | 99.60      | 99.80    | 99.80 | 99.80    |
|              | Split 4 | 100         | 100         | 100        | 100      | 100   | 100      |
|              | Split 5 | 100         | 100         | 100        | 100      | 100   | 100      |
|              | Average | 100         | 99.92       | 99.92      | 99.96    | 99.96 | 99.96    |
| GoogleNet    | Split 1 | 94.00       | 91.06       | 91.44      | 92.70    | 92.53 | 92.54    |
|              | Split 2 | 95.60       | 91.46       | 91.92      | 93.73    | 93.53 | 93.55    |
|              | Split 3 | 96.00       | 92.28       | 92.66      | 94.30    | 94.14 | 94.15    |
|              | Split 4 | 93.20       | 90.65       | 91.02      | 92.09    | 91.93 | 91.94    |
|              | Split 5 | 94.80       | 90.65       | 91.15      | 92.94    | 92.73 | 92.74    |
|              | Average | 94.72       | 91.22       | 91.64      | 93.15    | 92.97 | 92.98    |
| Inceptionv3  | Split 1 | 97.20       | 94.72       | 94.92      | 96.05    | 95.96 | 95.97    |
|              | Split 2 | 92.80       | 94.31       | 94.31      | 93.55    | 93.55 | 93.55    |
|              | Split 3 | 94.40       | 93.90       | 94.02      | 94.21    | 94.15 | 94.15    |
|              | Split 4 | 94.40       | 95.12       | 95.16      | 94.78    | 94.76 | 94.76    |
|              | Split 5 | 94.40       | 90.65       | 91.12      | 92.73    | 92.53 | 92.54    |
|              | Average | 94.64       | 93.74       | 93.91      | 94.26    | 94.19 | 94.19    |
| MobileNetv2  | Split 1 | 98.00       | 96.75       | 96.84      | 97.42    | 97.37 | 97.38    |
|              | Split 2 | 98.00       | 95.93       | 96.08      | 97.03    | 96.97 | 96.98    |
|              | Split 3 | 99.20       | 96.34       | 96.50      | 97.83    | 97.77 | 97.78    |
|              | Split 4 | 98.40       | 97.15       | 97.23      | 97.81    | 97.78 | 97.78    |
|              | Split 5 | 99.20       | 93.50       | 93.94      | 96.50    | 96.35 | 96.37    |
|              | Average | 98.56       | 95.93       | 96.12      | 97.32    | 97.25 | 97.26    |
| ResNet50     | Split 1 | 96.80       | 95.93       | 96.03      | 96.41    | 96.37 | 96.37    |
|              | Split 2 | 98.40       | 97.15       | 97.23      | 97.81    | 97.78 | 97.78    |
|              | Split 3 | 97.60       | 95.12       | 95.31      | 96.44    | 96.36 | 96.37    |
|              | Split 4 | 97.20       | 97.97       | 97.98      | 97.59    | 97.58 | 97.58    |
|              | Split 5 | 98.40       | 96.34       | 96.47      | 97.43    | 97.37 | 97.38    |
|              | Average | 97.68       | 96.50       | 96.60      | 97.14    | 97.09 | 97.10    |
Table 6: Performance comparison on CT images. Here we used SVM to classify features extracted using the different architectures.

| Architecture | Splits    | Performance metrics (%) |
|--------------|-----------|-------------------------|
|              |           | Sensitivity | Specificity | Precision | F1-score | AUC  | Accuracy |
| AlexNet      | Split 1   | 95.60       | 98.78       | 97.76     | 97.15     | 97.19 | 97.18    |
|              | Split 2   | 97.20       | 95.53       | 95.67     | 96.43     | 96.36 | 96.37    |
|              | Split 3   | 93.60       | 95.93       | 95.90     | 94.74     | 94.77 | 94.76    |
|              | Split 4   | 98.00       | 95.93       | 96.08     | 97.03     | 96.97 | 96.98    |
|              | Split 5   | 96.80       | 96.34       | 96.41     | 96.61     | 96.57 | 96.57    |
|              | Average   | 96.24       | 96.50       | 96.56     | 96.39     | 96.37 | 96.37    |
| CovidNet     | Split 1   | 99.60       | 100         | 100       | 99.80     | 99.80 | 99.80    |
|              | Split 2   | 99.20       | 100         | 100       | 99.60     | 99.60 | 99.60    |
|              | Split 3   | 99.20       | 100         | 100       | 99.60     | 99.60 | 99.60    |
|              | Split 4   | 99.60       | 100         | 100       | 99.80     | 99.80 | 99.80    |
|              | Split 5   | 99.60       | 99.59       | 99.60     | 99.60     | 99.60 | 99.60    |
|              | Average   | 99.44       | 99.92       | 99.92     | 99.68     | 99.68 | 99.68    |
| GoogleNet    | Split 1   | 94.00       | 96.34       | 96.31     | 95.14     | 95.17 | 95.16    |
|              | Split 2   | 96.00       | 93.50       | 93.75     | 94.86     | 94.75 | 94.76    |
|              | Split 3   | 95.20       | 94.31       | 94.44     | 94.82     | 94.75 | 94.76    |
|              | Split 4   | 96.00       | 94.31       | 94.49     | 95.24     | 95.15 | 95.16    |
|              | Split 5   | 96.40       | 94.31       | 94.51     | 95.45     | 95.35 | 95.36    |
|              | Average   | 95.52       | 94.55       | 94.70     | 95.10     | 95.03 | 95.04    |
| Inceptionv3  | Split 1   | 92.80       | 94.31       | 94.31     | 93.55     | 93.55 | 93.55    |
|              | Split 2   | 96.80       | 94.72       | 94.90     | 95.84     | 95.76 | 95.77    |
|              | Split 3   | 96.80       | 93.90       | 94.16     | 95.46     | 95.35 | 95.36    |
|              | Split 4   | 93.20       | 94.31       | 94.33     | 93.76     | 93.75 | 93.75    |
|              | Split 5   | 96.00       | 95.12       | 95.24     | 95.62     | 95.56 | 95.56    |
|              | Average   | 95.12       | 94.47       | 94.59     | 94.85     | 94.79 | 94.80    |
| MobileNetv2  | Split 1   | 97.20       | 95.53       | 95.67     | 96.43     | 96.36 | 96.37    |
|              | Split 2   | 96.40       | 97.56       | 97.57     | 96.98     | 96.98 | 96.98    |
|              | Split 3   | 97.60       | 97.15       | 97.21     | 97.41     | 97.38 | 97.38    |
|              | Split 4   | 98.80       | 96.75       | 96.86     | 97.82     | 97.77 | 97.78    |
|              | Split 5   | 97.60       | 98.78       | 98.79     | 98.19     | 98.19 | 98.19    |
|              | Average   | 97.52       | 97.15       | 97.22     | 97.37     | 97.34 | 97.34    |
| ResNet50     | Split 1   | 97.20       | 95.53       | 95.67     | 96.43     | 96.36 | 96.37    |
|              | Split 2   | 98.40       | 97.15       | 97.23     | 97.81     | 97.78 | 97.78    |
|              | Split 3   | 95.20       | 97.56       | 97.54     | 96.36     | 96.38 | 96.37    |
|              | Split 4   | 98.40       | 97.56       | 97.62     | 98.01     | 97.98 | 97.98    |
|              | Split 5   | 92.00       | 95.93       | 95.83     | 93.88     | 93.97 | 93.95    |
|              | Average   | 96.24       | 96.75       | 96.78     | 96.50     | 96.49 | 96.49    |
Table 7: Performance comparison on CT images. Here we used RF to classify features extracted using the different architectures.

| Architecture | Splits | Performance metrics (%) |  |
|--------------|--------|--------------------------|---|
|              |        | Sensitivity | Specificity | Precision | F1-score | AUC | Accuracy |
| AlexNet      | Split 1 | 92.40       | 95.53       | 95.45     | 93.90     | 93.96| 93.95    |
|              | Split 2 | 93.60       | 93.50       | 93.60     | 93.60     | 93.55| 93.55    |
|              | Split 3 | 94.00       | 93.50       | 93.63     | 93.81     | 93.75| 93.75    |
|              | Split 4 | 91.20       | 95.93       | 95.80     | 93.44     | 93.57| 93.55    |
|              | Split 5 | 94.00       | 93.50       | 93.63     | 93.81     | 93.75| 93.75    |
|              | Average | 93.04       | 94.39       | 94.42     | 93.71     | 93.72| 93.71    |
| CovidNet     | Split 1 | 100         | 99.59       | 99.60     | 99.80     | 99.80| 99.80    |
|              | Split 2 | 99.60       | 99.59       | 99.60     | 99.60     | 99.60| 99.60    |
|              | Split 3 | 100         | 99.19       | 99.21     | 99.60     | 99.59| 99.60    |
|              | Split 4 | 99.20       | 100         | 100       | 99.60     | 99.60| 99.60    |
|              | Split 5 | 99.60       | 99.59       | 99.60     | 99.60     | 99.60| 99.60    |
|              | Average | 99.68       | 99.59       | 99.60     | 99.64     | 99.64| 99.64    |
| GoogleNet    | Split 1 | 90.80       | 93.90       | 93.80     | 92.28     | 92.35| 92.34    |
|              | Split 2 | 92.00       | 93.09       | 93.12     | 92.56     | 92.54| 92.54    |
|              | Split 3 | 91.20       | 91.87       | 91.94     | 91.57     | 91.53| 91.53    |
|              | Split 4 | 90.80       | 91.46       | 91.53     | 91.16     | 91.13| 91.13    |
|              | Split 5 | 92.80       | 93.50       | 93.55     | 93.17     | 93.15| 93.15    |
|              | Average | 91.52       | 92.76       | 92.79     | 92.15     | 92.14| 92.14    |
| Inceptionv3  | Split 1 | 92.00       | 89.84       | 90.20     | 91.09     | 90.92| 90.93    |
|              | Split 2 | 89.20       | 91.06       | 91.02     | 90.10     | 90.13| 90.12    |
|              | Split 3 | 88.40       | 86.59       | 87.01     | 87.70     | 87.49| 87.50    |
|              | Split 4 | 87.60       | 89.84       | 89.75     | 88.66     | 88.72| 88.71    |
|              | Split 5 | 92.80       | 92.28       | 92.43     | 92.61     | 92.54| 92.54    |
|              | Average | 90.00       | 89.92       | 90.08     | 90.03     | 89.96| 89.96    |
| MobileNetv2  | Split 1 | 95.60       | 96.34       | 96.37     | 95.98     | 95.97| 95.97    |
|              | Split 2 | 92.00       | 90.24       | 90.55     | 91.27     | 91.12| 91.13    |
|              | Split 3 | 92.40       | 94.31       | 94.29     | 93.33     | 93.35| 93.35    |
|              | Split 4 | 95.20       | 95.93       | 95.97     | 95.58     | 95.57| 95.56    |
|              | Split 5 | 92.80       | 95.93       | 95.87     | 94.31     | 94.37| 94.35    |
|              | Average | 93.60       | 94.55       | 94.61     | 94.09     | 94.08| 94.07    |
| ResNet50     | Split 1 | 94.80       | 94.31       | 94.42     | 94.61     | 94.55| 94.56    |
|              | Split 2 | 94.40       | 96.34       | 96.33     | 95.35     | 95.37| 95.36    |
|              | Split 3 | 96.00       | 96.75       | 96.77     | 96.39     | 96.37| 96.37    |
|              | Split 4 | 95.60       | 94.31       | 94.47     | 95.03     | 94.95| 94.96    |
|              | Split 5 | 92.80       | 96.75       | 96.67     | 94.69     | 94.77| 94.76    |
|              | Average | 94.72       | 95.69       | 95.73     | 95.21     | 95.20| 95.20    |
References

[1] Q. Sun, H. Qiu, M. Huang, Y. Yang, Lower mortality of covid-19 by early recognition and intervention: experience from jiangsu province, Annals of intensive care 10 (2020) 1–4.

[2] A. Tahamtan, A. Ardebili, Real-time rt-pcr in covid-19 detection: issues affecting the results, 2020.

[3] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, J. Liu, Chest ct for typical 2019-ncov pneumonia: relationship to negative rt-pcr testing, Radiology (2020) 200343.

[4] Y. Li, L. Yao, J. Li, L. Chen, Y. Song, Z. Cai, C. Yang, Stability issues of rt-pcr testing of sars-cov-2 for hospitalized patients clinically diagnosed with covid-19, Journal of medical virology (2020).

[5] G. Herpe, M. Lederlin, M. Naudin, M. Ohana, K. Chaumoitre, J. Gregory, V. Vilgrain, C. A. Freitag, C. De Margerie-Mellon, V. Flory, et al., Efficacy of chest ct for covid-19 pneumonia in france, Radiology (2020) 202568.

[6] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, U. R. Acharya, Automated detection of covid-19 cases using deep neural networks with x-ray images, Computers in Biology and Medicine (2020) 103792.

[7] A. I. Khan, J. L. Shah, M. M. Bhat, Coronet: A deep neural network for detection and diagnosis of covid-19 from chest x-ray images, Computer Methods and Programs in Biomedicine (2020) 105581.

[8] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, D. Shen, Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19, IEEE reviews in biomedical engineering (2020).

[9] J. S. Obeid, M. Davis, M. Turner, S. M. Meystre, P. M. Heider, L. A. Lenert, An ai approach to covid-19 infection risk assessment in virtual visits: a case report, Journal of the American Medical Informatics Association (2020).

[10] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, et al., Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct, Radiology (2020).

[11] S. Hu, Y. Gao, Z. Niu, Y. Jiang, L. Li, X. Xiao, M. Wang, E. F. Fang, W. Menpes-Smith, J. Xia, et al., Weakly supervised deep learning for covid-19 infection detection and classification from ct images, IEEE Access 8 (2020) 118869–118883.

[12] L. Huang, R. Han, T. Ai, P. Yu, H. Kang, Q. Tao, L. Xia, Serial quantitative chest ct assessment of covid-19: Deep-learning approach, Radiology: Cardiothoracic Imaging 2 (2020) e200075.
[13] Y. Oh, S. Park, J. C. Ye, Deep learning covid-19 features on cxr using limited training data sets, IEEE Transactions on Medical Imaging (2020).

[14] A. M. Ismael, A. Şengür, Deep learning approaches for covid-19 detection based on chest x-ray images, Expert Systems with Applications 164 (2021) 114054.

[15] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, R. B. Pachori, Application of deep learning techniques for detection of covid-19 cases using chest x-ray images: A comprehensive study, Biomedical Signal Processing and Control 64 (2021) 102365.

[16] T. Zebin, S. Rezvy, Covid-19 detection and disease progression visualization: Deep learning on chest x-rays for classification and coarse localization, Applied Intelligence 51 (2021) 1010–1021.

[17] S. Sheykhivand, Z. Mousavi, S. Mojtabahed, T. Y. Rezaai, A. Farzamnia, S. Meshgini, I. Saad, Developing an efficient deep neural network for automatic detection of covid-19 using chest x-ray images, Alexandria Engineering Journal 60 (2021) 2885–2903.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (2015) 211–252.

[19] T. Mahmud, M. A. Rahman, S. A. Fattah, Covxnet: A multi-dilation convolutional neural network for automatic covid-19 and other pneumonia detection from chest x-ray images with transferable multi-receptive feature optimization, Computers in biology and medicine (2020) 103869.

[20] L. Brunese, F. Mercaldo, A. Reginelli, A. Santone, Explainable deep learning for pulmonary disease and coronavirus covid-19 detection from x-rays, Computer Methods and Programs in Biomedicine (2020) 105608.

[21] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, S. Riaz, K. Ali, C. N. John, I. Hussain, M. Nabeel, Aidcovid-19: Ai enabled preliminary diagnosis for covid-19 from cough samples via an app, Informatics in Medicine Unlocked (2020) 100378.

[22] H. S. Maghdid, K. Z. Ghafoor, A. S. Sadiq, K. Curran, K. Rabie, A novel ai-enabled framework to diagnose coronavirus covid 19 using smartphone embedded sensors: Design study, arXiv preprint arXiv:2003.07434 (2020).

[23] F. Rustam, A. A. Reshi, A. Mehmood, S. Ullah, B. On, W. Aslam, G. S. Choi, Covid-19 future forecasting using supervised machine learning models, IEEE Access (2020).

[24] Y. Peng, M. H. Nagata, An empirical overview of nonlinearity and overfitting in machine learning using covid-19 data, Chaos, Solitons & Fractals (2020) 110055.
[25] G. Pinter, I. Felde, A. Mosavi, P. Ghamisi, R. Gloaguen, Covid-19 pandemic prediction for Hungary; a hybrid machine learning approach, Mathematics 8 (2020) 890.

[26] H. Burdick, C. Lam, S. Mataraso, A. Lynn-Palevsky, G. Braden, R. P. Dellinger, A. McCoy, J.-L. Vincent, A. Green-Saxena, G. Barnes, et al., Prediction of respiratory decompensation in Covid-19 patients using machine learning: The ready trial, Computers in Biology and Medicine (2020) 103949.

[27] M. A. Elaziz, K. M. Hosny, A. Salah, M. M. Darwish, S. Lu, A. T. Sahlol, New machine learning method for image-based diagnosis of Covid-19, PloS one 15 (2020) e0235187.

[28] S. H. Kassani, P. H. Kassasni, M. J. Wesolowski, K. A. Schneider, R. Deters, Automatic detection of coronavirus disease (Covid-19) in x-ray and CT images: A machine learning-based approach, arXiv preprint arXiv:2004.10641 (2020).

[29] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[30] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[31] I. D. Apostolopoulos, T. A. Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, Physical and Engineering Sciences in Medicine (2020) 1.

[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[34] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, arXiv preprint arXiv:1602.07261 (2016).

[35] S. rekha Hanumanthu, Role of intelligent computing in covid-19 prognosis: A state-of-the-art review, Chaos, Solitons & Fractals (2020) 109947.

[36] A. A. Ardakani, A. R. Kanafi, U. R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage covid-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks, Computers in Biology and Medicine (2020) 103795.
[37] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[38] E. Soares, P. Angelov, S. Biaso, M. H. Froes, D. K. Abe, Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification, medRxiv (2020).

[39] E. E.-D. Hemdan, M. A. Shouman, M. E. Karar, Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images, arXiv preprint arXiv:2003.11055 (2020).

[40] P. K. Sethy, S. K. Behera, Detection of coronavirus disease (covid-19) based on deep features, Preprints 2020030300 (2020) 2020.

[41] A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, arXiv preprint arXiv:2003.10849 (2020).

[42] C. Butt, J. Gill, D. Chun, B. A. Babu, Deep learning system to screen coronavirus disease 2019 pneumonia, Applied Intelligence (2020) 1.

[43] H. Y. F. Wong, H. Y. S. Lam, A. H.-T. Fong, S. T. Leung, T. W.-Y. Chin, C. S. Y. Lo, M. M.-S. Lui, J. C. Y. Lee, K. W.-H. Chiu, T. Chung, et al., Frequency and distribution of chest radiographic findings in covid-19 positive patients, Radiology (2020) 201160.

[44] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, W. Ji, Sensitivity of chest ct for covid-19: comparison to rt-pcr, Radiology (2020) 200432.

[45] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.

[46] A. Jacobi, M. Chung, A. Bernheim, C. Eber, Portable chest x-ray in coronavirus disease-19 (covid-19): A pictorial review, Clinical Imaging (2020).

[47] M.-Y. Ng, E. Y. Lee, J. Yang, F. Yang, X. Li, H. Wang, M. M.-s. Lui, C. S.-Y. Lo, B. Leung, P.-L. Khong, et al., Imaging profile of the covid-19 infection: radiologic findings and literature review, Radiology: Cardiothoracic Imaging 2 (2020) e200034.

[48] H. Shi, X. Han, N. Jiang, Y. Cao, O. Alwalid, J. Gu, Y. Fan, C. Zheng, Radiological findings from 81 patients with covid-19 pneumonia in wuhan, china: a descriptive study, The Lancet Infectious Diseases (2020).

[49] J. Cleverley, J. Piper, M. M. Jones, The role of chest radiography in confirming covid-19 pneumonia, bmj 370 (2020).

[50] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: International conference on machine learning, 2014, pp. 647–655.
[51] J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, in: Advances in neural information processing systems, 2012, pp. 2951–2959.

[52] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE Transactions on Geoscience and Remote Sensing 54 (2016) 6232–6251.