COVID-19 Detection Based on Chest X-Ray Image Classification using Tailored CNN Model

Mahmoud Z. Fetoh
Information Technology Department,
Faculty of Computers and Information,
Menoufia University, Egypt
mahmoud.zaky189@ci.menofia.edu.eg

Khalid M. Amin
Information Technology Department,
Faculty of Computers and Information,
Menoufia University, Egypt
k.amin@ci.menofia.edu.eg

Ahmed M. Hamad
Information Technology Department,
Faculty of Computers and Information,
Menoufia University, Egypt
ahmahit@ci.menofia.edu.eg

Abstract—**The outbreak of Covid-19 epidemic led to millions of injuries and deaths and pressure on the health system. The limited availability of diagnosis tools and expert radiologists raise the need of using computer aided tools to diagnosis Covid-19 cases. In this study, a tailored Convolutional Neural Network (CNN) architecture model is proposed to automatically detect Covid-19 cases using chest X-Ray (CXR) images. The proposed CNN model consist of three phases preprocessing, feature extraction and classification. The proposed CNN model depends on kernel separability to reduce the training parameters to a large extent. Furthermore, the proposed model used residual connection and batch normalization extensively to maintain the network stability during the training process and provide the model with the regularization effect in order to reduce the overfitting. Training process hyperparameter (such as batch size and learning rate) are determined dynamically. The proposed architecture is trained using QaTa-Cov19 benchmark dataset achieving 100% for accuracy, sensitivity, precision and F1-score with a very low parameter count (150K) compared with the other methods in the literature.**

Index Terms—**Covid-19, CNN, Deep Learning, Residual Connection.**

I. INTRODUCTION

Coronavirus disease 2019 (COVID-19) caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) is a contagious disease [1]. First reported case was from Wuhan, China in December 2019 [2] [3]. On 11 March 2020, World Health Organization (WHO) has declared the outbreak of coronavirus 2 (SARS-CoV-2) as a pandemic [4]. The typical diagnostic method is by detection of the virus' nucleic acid by real-time reverse transcription polymerase chain reaction (rRT-PCR), reverse transcription loop-mediated isothermal amplification (RT-LAMP) from a nasopharyngeal swab, or by transcription-mediated amplification (TMA). All these methods are manual, complex, and time-consuming. Therefore, the need to automate the screening of patients is urgent to rapidly diagnose the infected people which leads to help them to receive immediate treatment and be isolated to mitigate the spread of the virus.

In the war with this disease, the important task is the rapid and effective screening of patients. chest X-Ray is the most common tool in medicine used to diagnose conditions affecting the chest. Chest X-ray can be used to discriminate COVID-19 pneumonia from normal chest. The lung texture is used to discriminate the infected cases from normal cases.

The recent advances in a Deep Learning [5], particularly CNN [6], contributes the great improvement in the image classification accuracy. CNN is one of the deep learning models that showed a great success in processing the 1D and 2D signals. CNN basically composed of two parts convbase part and densebase part. Convbase part is responsible for feature extraction. It consists of a sequence of convolutional layers, each layer is followed by an activation function and pooling layer. Convolutional layers consist of N filters, each filter is adapted during the training process to extract a particular feature from the input signal. Activation function [7] is a nonlinear function to provide the nonlinearity to the network. Activation function makes the network to approximate any arbitrary function to solve non-trivial problems. Typical Activation function used is rectified linear unit (ReLU) and its variants. Pooling layer summarizes the strongest activations, reduces the spatial size of the feature maps, and increases the receptive field [8] of the network. Max-pooling is the most common pooling layer that achieves accurate results.

Residual connections between the convolutional layers may be used. Residual connection [9] allow the flow of information from the higher layer to the initial layers. The deep layers of deeper network sometime needs to approximate the identity function which is not a simple task especially the existence of a nonlinear functions. Residual connection [23], skip connection, come to solve this problem. In [23], it is claimed that the function $F(X) := H(X) - X$, where $H(X)$ is the required function to be approximated, is easier to optimize. This implies the optimization of $F(X) + X$. This form allows the identity function to be easily approximated as $F(X)$ can easily tend to zero as the consequence alleviates the problem of degradation [23]. Also, residual connection alleviate the vanishing gradient problem as it allows the flow of gradient to the earlier layer.

The other part of CNN is the densebase part which is responsible for classification the output of the convebase part. It consists of a fully connected dense layers.

This paper is organized as follows: Section II presents the recent Related work and the corresponding results. Section III presents the proposed CNN architecture and explores it designing issues. Section IV shows the experimental results

and compares between the proposed model and a set of state-of-the-art methods using precision, specificity, F1-score, and accuracy metrics.

II. RELATED WORK

Afshar et al. [10] presented a deep learning model that is based on capsule networks. The proposed network architecture consists of 4 convolutional layers and 3 capsule layers. The last capsule layer contains the instantiation parameters of the two classes. They achieved 95.7%, 90%, 95.8%, and 0.97 for Accuracy, Sensitivity, Specificity and Area Under the Curve, respectively. The model used 0.295M trainable parameters.

In [11], the performance of eight pre-trained models such as MobileNet-V2, SqueezeNet, GoogleNet, AlexNet, VGG-16, ResNet-34, ResNet-50 and Inception-V3 is evaluated for classifying positive and negative COVID-19 cases. The methodology consists of two stages; preprocessing and classification using a pre-trained model. In the preprocessing stage, the input chest X-Ray image is initially normalized to range from 0 to 1. Normalization maintained the network numerical stability. The second part of the preprocessing stage is data augmentation [12] it extend the dataset training samples. ResNet-34 achieved the maximum performance with an accuracy of 98.33%.

COVIDx dataset which contain a 13,975 chest X-Ray images for a 13,870 patient cases is presented in [13]. In addition, a CNN architecture for classifying normal, non-Covid-19, and Covid-19 cases is proposed. The architecture used the residual connection heavily achieving an overall accuracy of 93% and total number of 11.75M trainable parameters.

In [14], a two-phase classification technique to discriminate between pneumonia, COVID-19 and normal chest X-Ray images is introduced. In phase I, chest X-Ray Image is classified as Normal or abnormal. The abnormal chest X-Ray image is further classified to Covid-19 or pneumonia in Phase II. Both Phase I and Phase II have the same anatomy. Features are extracted using eight first order statistical features, 88 grey level co-occurrence matrix [15] and 8100 histogram of oriented gradients. Binary Grey Wolf Optimization (BGWO) is used as a feature selector to select the most important features. A total of 8196 features is classified using ensemble classifier. This classifier is a majority voting that is based on ensemble classifier and composed of SVM, ANN, KNN, NB, and DT. The technique achieved an accuracy of 98.062 and 91.329 for Phase I and Phase II, respectively.

In [16], a number of light-weigh architecture for binary class architecture for Covid-19 and non-Covid-19 classification and multi-class architecture for Bectria, normal, and covid-19 classification is proposed. The binary class architecture consists of a convolutional layer with 32 filter and 2D filter of size 4 × 4, followed by an activation function (ReLU) and max-pooling layer. A densebase consists of dense layer of size 128 neurons, dropout and a two neurons dense layer with softmax activation function. The model contains 49,05 trainable parameters. The multi-class architecture contains 5 convolutional layers. Each layer is followed by an activation function (ReLU). After the second convolutional layer and
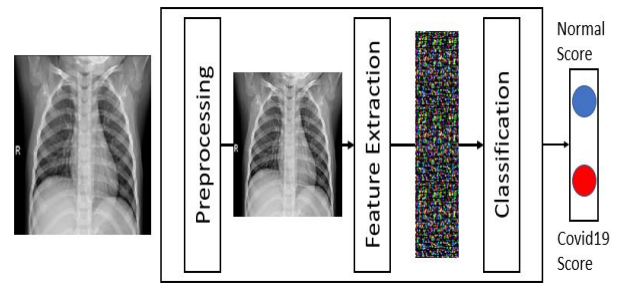

Fig. 1. The phases of the proposed method.

its ReLU, there is max-pooling layer and dropout layer [17]. The binary class model achieved 98.7%, 100% and 98.3% for accuracy, sensitivity, and specificity, respectively. The multi-class model achieved 98.3%, 99.3% and 98.1% for accuracy, sensitivity, and specificity, respectively.

In this paper, a proposed tailored CNN architecture for automatically detecting Covid-19 cases classifying chest X-Ray images into normal or infected. The proposed architecture model depends on kernel separability that decomposes 2D convolution kernel to linear combinations of two 1D vectors. The separability of kernels leads to a large reduction in the output parameters. To decrease the covariate shift resulted due to the moving between model layers and to reduces the problem of vanishing gradient problem, the proposed model used batch normalization. In addition, Rectified Linear Unit (ReLU) is used as a nonlinear activation function to provide the model with the nonlinearity nature in order to allow it to be approximate to complex data. Experimental results illustrates the efficiency of the proposed model in detecting Covid-19 infection through a comparison between the proposed CNN model and a set of state-of-the-art methods using precision, sensitivity, F1-score, and accuracy metrics. The result show that the proposed CNN model achieved the highest accuracy and a lowest number of parameters.

III. THE PROPOSED METHOD

In this section, a proposed method to detect COVID-19 disease from chest X-Ray images is presented. The proposed method exploits CNN model to classify the input chest X-Ray image to one of two categories; normal case or Covid-19 case. The proposed method consists of three phases: preprocessing, feature extraction, and classification. The proposed method phases is shown in Fig.1.

A. The Preprocessing Phase

The preprocessing phase is responsible for resizing and normalizing the input chest X-Ray images. The pre-processing phase is employed to maintain the numerical stability of the model and reduce the co-variance shift [6]. In addition, this phase leads the learning model of CNN model to reduce the required overhead to adapt to the different scales of different features of the input data. Reshaping size is determined

empirically. The input chest X-Ray image is re-sized and then adapted and normalized to a normal distribution as follows:

$$Y := \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{1}$$

where μ and σ is the mean and standard deviation of chest X-Ray image (X), respectively.

After re-sizing the input chest X-Ray image, the input image is normalized to have a zero mean and unit standard deviation. Then, the image can be scaled and shifted with a normalization parameter which is determined and adapted by the training dataset during the training process according to the following equation:

$$Z := w_1 Y + w_2 \tag{2}$$

where w_1 and w_2 are a trainable parameter.

Unlike the normalization method presented in [18], the batch normalization process presented in this paper has z-score normalization parameter that is used in both training and validation phases.

B. Feature Extraction and Classification

CNN models achieved an outstanding success in image recognition [5]. This phase is responsible for extracting spatial features from the normalized chest X-Ray image using a tailored CNN model. This phase is based on learning the CNN model by the input preprocessed chest X-Ray images. The design of the tailored CNN model is described as follows:

1) Separable CNN kernels

Kernel separability [19] [20] is based on decomposing a 2D convolution kernel to linear combinations of two 1D vectors which leads to a large reduction in the total number of resulting parameters. For example, a 2D kernel of size 9×9 has a total number of $9^2 = 81$ trained parameters. Whereas in the case of separating this 2D kernel to linear combinations of two 1D vectors of sizes 9×1 and 1×9, this results in a total number of $9 + 9 = 18$ trained parameters. As a consequence, kernel separability reduces the number of CNN model operations (such as the multiplication and the addition). A 2D kernel of $k \times k$ applied for 2D signal with spatial dimensions of $M \times N$ has a total number of $(N-4)(M-4) \times k^2$ operations but in case of applying kernel separability yields $2(N-4)(M-4)k$ operations. The flow of separated convolution operations is summarized in Fig.2. Fig.3 represents the structure, denoted by Separated Convolutonal Layer, used in the proposed method with kernel size of $(M \times N)$ and satisfying the convolutional kernel separability. Separated Convolutonal Layer is composed of three consecutive layers. The first convolutional layer has a kernel size of $(M \times 1)$ and the number of convolutional neuron and filters are equal to the number of channels as the input feature map and the convolution operations are performed in a channel wise. The second layer operates in the same way as the first layer but it has a kernel of size $(1 \times M)$. The third layer is the convolutional layer with kernel of size (1×1) and number of convolutional neuron is N. The collaboration of the three layers are connected to preform similarly to the convolutional layer with kernel size of $(M \times M)$ and number of neuron and filter are the same as N but with large difference in the performance.

2) Batch Normalization and Activation function

The proposed method follows the linear separable convolution kernels by a batch normalization and an activation function. Rectified Linear Unit (ReLU) is a nonlinear activation function that adds the nonlinearity to the model in order to allow it to be approximate to complex data. The proposed method employs the batch normalization which is described in [18]. Batch normalization algorithm is described as follows:

Require: : Minibatch activation values $x : \mathcal{B} = \{x_{1,\ldots,m}\}$; parameters to be learned γ ,β.
Ensure: : $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

1: $\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$

2: $\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$

3: $\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$

4: $y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$

where the batch normalization is performed for each layer independently such that μ_B is the mean value for all ith channel of each sample and σ_B^2 is the standard deviation value for all ith channel of each sample. γ and β are trainable parameters. In case of a single sample is processed by the model, which is not valid for calculating μ_B, σ_B^2, a moving average is calculated for μ_B, σ_B^2 of each batch. So, the batch normalization algorithm for inference updated to:

Require: :activation values x ; Calculated $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}^2$ of all training minibatchs.
Ensure: : $\{y_i = BN_{\gamma^{(k)},\beta^{(k)}}^{\mathrm{inf}}(x^{(k)})\}$

1: $E_x \leftarrow \frac{1}{m} \sum_{i=1}^{m} \mu_{\mathcal{B}}$

2: $V_x \leftarrow (\frac{m}{m-1}) \frac{1}{m} \sum_{i=1}^{m} \sigma_{\mathcal{B}}^2$

3: $y_i \leftarrow \frac{\gamma^{(k)}}{\sqrt{V_x + \epsilon}} x^{(k)} + \left(\beta^{(k)} - \frac{\gamma^{(k)} E_x}{\sqrt{V_x + \epsilon}} \right)$

4: $y_i \equiv BN_{\gamma^{(k)},\beta^{(k)}}^{\mathrm{inf}}(x^{(k)})$

Batch Normalization [18] reduces the covariate shift produced as a result of moving between layers during the feedforward procedure [18]. Bach Normalization makes the landscape loss smoother and reduces the number of saddle points [21] which allows to use higher learning rates. Using a higher learning rate makes the network training faster [18]. Batch normalization reduces the vanishing gradient problem and exploding gradient problem as it makes the resulted activation scale independent from the trainable parameter scale [18]. Batch normalization has the effect of regularization because of the inherited randomness when selecting the batch sample [18] which help the generalization to unseen chest X-Ray image.
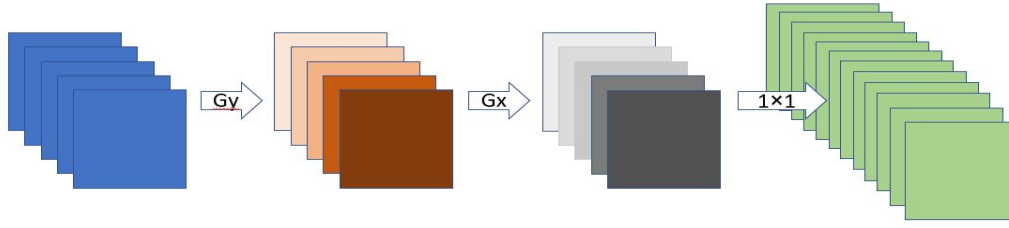
Fig. 2. Separable convolution Gy and Gx have kernel size of $M \times 1$ and $1 \times M$. The combination of these kernels is approximately a $M \times M$ kernel and depth wise convolution are applied by a 1×1 convolution. The output depth is padded with zeros to have the same spatial size of Gy, Gx. Gy, Gx are performed channel wise.
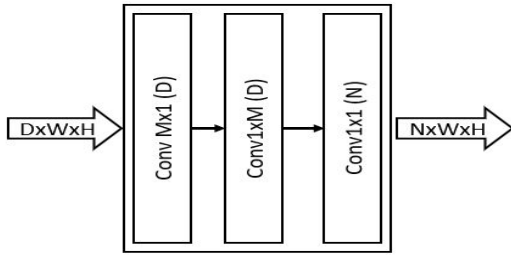

Fig. 3. Separated Convolutonal Layer: composed of three consecutive layers. The first Convolutional layer has a kernel size of $(M \times 1)$ and D convolutional neuron. The second layer operates in the same way as the first layer but it has a kernel of size $(1 \times M)$ and D convolutional neuron. The third layer is the convolutional layer with kernel of size (1×1) and number of convolutional neuron is N.

As the proposed method uses the variants of ReLU [22] extensively, where it is defined as $f(x) = max(0, x)$. ReLU does not have saturation problem, like Sigmoid, which limits and prevents the trainable parameter updates.

3) Deep and larger receptive field Network design

Deeper convolutional neural network design is a very important task for any image recognition task [23]. Training a deeper network is very expensive and has many challenges such as vanishing gradient problem, exploding gradient problem, and degradation problem [23]. Exploding gradient problem occurs when the gradient update becomes very large (approaching infinity) resulting in the network diversion. Vanishing gradient problem occurs when the gradient update becomes very small (approaching zero) resulting in preventing the parameter update for early layers [23] and preventing the network to learn new patterns. Batch normalization [18] and the use of ReLU activation function [22] alleviate these two problems.

The deep layers of CNN networks sometimes need to approximate the identity function which is not a simple task especially with the existence of a nonlinear functions. Residual connection [23] overcomes this problem by using skip connection as shown in Fig.4.

Fig.4 represents the building block layer of the feature extraction phase, denoted by stack of Residual Separated Block (RSB). RSB consists of four layers of separated convolutional layers, each layer is followed by a batch normalization and an activation function. It has an output of depth N where each sublayer produces an output of depth $N/4$ which is concatenated at the end of the layer to produce a depth N. RSB produces a feature map that includes both low level features and high level features.

Unlike the traditional neural network, which is fully connected to the previous layer, convolutional neural network is connected locally to a local region of the previous feature map. This introduces the concept of the network receptive field [8]. Receptive field should be large enough to capture large patterns in the input chest X-Ray image. Therefore, any consecutive convolutional layers in the proposed method without a pooling layer in between a larger kernel size is used in one of them. Residual Separated block, RSB, in Fig.4 may have kernel sizes of 3, 5, 7, and 9, respectively. Fig.5 Represent a complete CNN architecture.

IV. EXPERIMENTAL RESULTS

To test and validate the performance of the proposed method, a benchmark QaTa-COV19 dataset is used. The dataset includes 12,544 normal chest X-Ray images and 4,603 Covid-19 chest X-Ray images. The proposed method used a balanced number of samples in the training phase such that the first 4,603 normal samples and 4,603 Covid-19 samples are used. For testing phase, 461 chest X-Ray images are sampled randomly and are used for each class. Totally, 8284 images are used for training and 922 images are used for testing. Experiments are conducted on a Lenovo Z50-70 with Intel CORE i7-4510U CPU 2.00 GHz, 8GB RAM, NVIDIA GeForce 840M GPU; and with python and PyTorch library.

The Proposed architecture consists of Convbase and Densebase. Convbase is composed of a 6 feature extraction modules *(FX)* preceded by a batch normalization layer as shown in Fig.5. Each FX module can be considered as a sub-sequential model that consists of RSB layer followed by a batch normalization, Max-pooling, and LeakyReLU activation function. The Densebase is a two fully connected layers that classifies the Convbase output.

A. Hyperparameter Specification

All input chest X-Ray images are resized to be 200×200. After resizing the input images, these images are fed the Convbase model part which consists of 6 layers of residual
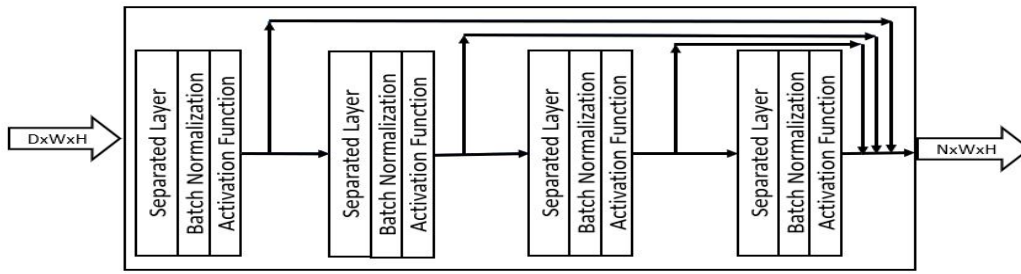
Fig. 4. The stack of residual separated block (RSB) consists of four layer of separated convolutional layer each of which is followed by batch normalization and activation function.
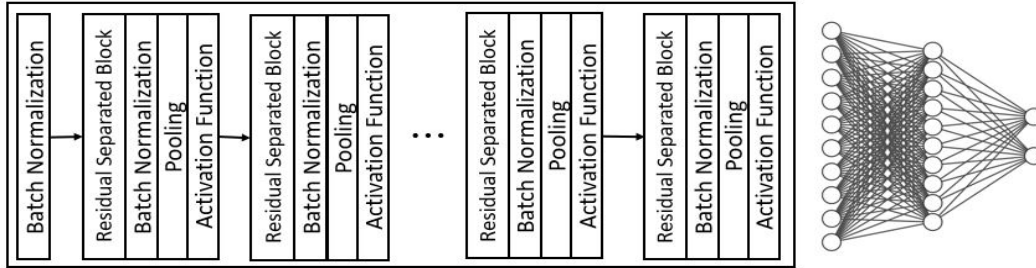

Fig. 5. The complete proposed tailored CNN architecture.

separated block. Each residual separated block is followed with batch normalization and LeakyReLU [7] as activation function as shown in Fig.5. The output depth of each residual separated block is 4×16, 4×32, 4×64, 4×64, 4×64 and 4×16, respectively. The output of Convbase model part is 1D feature vector of 576 length. Densebase model part consists of two hidden layers. Each layer has the size of 64 and the output layer of size 2. Each layer of Densebase layers is fully connected to its previous layer. The activation function used in the densebase model part is LeakyReLU. Table I summarizes the architecture hyperparameters.

TABLE I
THE PROPOSED ARCHITECTURE HYPERPARAMETERS

Layer Number	Layer Size	Activation Function
RSBLayer1	4 × 16	LeakyReLU
RSBLayer2	4 × 23	LeakyReLU
RSBLayer3	4 × 64	LeakyReLU
RSBLayer4	4 × 64	LeakyReLU
RSBLayer5	4 × 64	LeakyReLU
RSBLayer6	4 × 16	LeakyReLU
Flatten The Feature maps to 1D 576 feature vector		
LinearLayer1	64	LeakyReLU
LinearLayer2	64	LeakyReLU
LinearLayer3	2	Softmax

B. Network Training

The proposed CNN model is trained for 22 epoch. Adaptive Moment Estimation (Adam) optimizer [24] is a popular optimization technique for training deep networks. Adam optimizer is used during the training phase of the proposed

CNN model. Both batch size and Adam optimizer learning rate is changed during the training phase if the training loss stopped decreasing. Table II summarizes the parameters values used in the training phase of the proposed CNN model. Fig.6(a) show the progress for training and validation loss across each epoch. The difference between the training loss and validation loss through epochs show that our did not memorize the dataset.

TABLE II
THE CHANGE OF BATCH SIZE AND LEARNING RATE THROUGH THE TRAINING PROCESS

Epoch Number	Batch Size	Learning Rate
From 0 to 6	128	1e-3
From 7 to 12	256	1e-3
From 13 to 21	256	1e-4

C. Model Evaluation

To assess the efficiency of the proposed method, the proposed method is compared to recent state-of-the-art methods for detecting Covid-19 cases. Experiments are conducted using QaTa-COV19 dataset to evaluate the performance of all models with the corresponding hyperparameter of each model. All the methods depend on CNN. The comparison is performed using precision, sensitivity, F1-score, and accuracy [25]. In addition, the number of the parameters used in the training phase is very important comparison factor. Table III depicts the comparison between state-of-the-art methods and the proposed method. As shown in the comparison, the proposed method outperforms other methods achieving the maximum accuracy and the lowest parameter count.

TABLE III
A performance comparison between the proposed method and state-of-the-art models.

Method	PC	P(%)	S(%)	F1(%)	A(%)
Proposed Method	0.15M	100.00	100.00	100.00	100.00
ResNet-34 [11]	21.8M	96.77	100.00	98.36	98.33
ACoS Phase I [14]	-	98.266	96.512	98.551	98.062
ResNet-50 [11]	25.6M	95.24	100.00	97.56	97.50
GoogleNet [11]	5M	96.67	96.67	96.67	96.67
VGG-16 [11]	138M	95.08	96.67	95.87	95.83
AlexNet [11]	60M	96.72	98.33	97.52	97.50
MobileNet-V2 [11]	3.4M	98.24	93.33	95.73	95.83
Inception-V3 [11]	24M	96.36	88.33	92.17	92.50
SqueezeNet [11]	1.25M	98.27	95.00	96.61	96.67

PC is Parameter count, P is precision, S is sensitivity
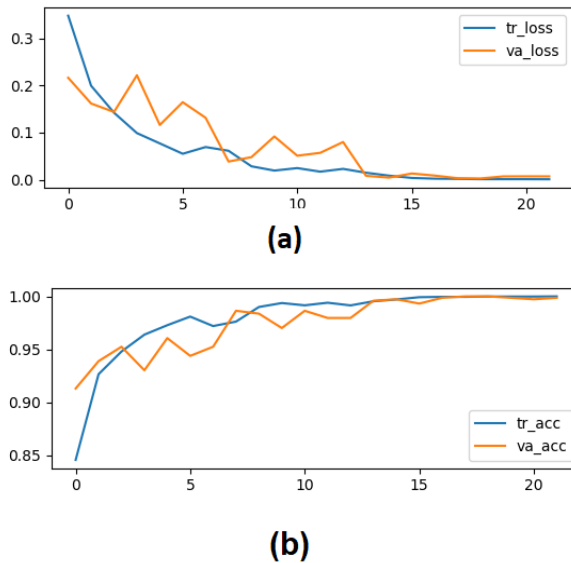*F1 is F1-score, and A is accuracy*

Fig. 6. (a) The training loss and the validation loss of each epoch and (b) The training accuracy and the validation accuracy of each epoch.

V. CONCLUSION

This paper proposed a new CNN architecture model for classifying CXR to Covid-19 cases or normal cases. The proposed architecture consists of separated kernels convolutional layers that is connected by a residual connection. The proposed architecture uses batch normalization to maintain the network stability during the training process. During the training process, the hyperparameters (such as batch size and learning rate) are determined dynamically. Experimental results showed that the proposed architecture model outperforms previous works for binary classification of chest X-Ray images to normal or Covid-19 cases. The proposed architecture has a very low parameter count (150K trainable parameter) compared to previous work. The proposed architecture achieved a performance of 100% for accuracy, sensitivity, precision and F1-score. In the future, adapting the model to perform Covid-19 segmentation from medical images will be added.

REFERENCES

[1] Acter, Thamina, et al. "Evolution of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) as coronavirus disease 2019 (COVID-19) pandemic: A global health emergency." (2020): 138996.
[2] Tobías, Aurelio, et al. "Changes in air quality during the lockdown in Barcelona (Spain) one month into the SARS-CoV-2 epidemic." (2020): 138540.
[3] Saglietto, Andrea, et al. "COVID-19 in Europe: the Italian lesson." (2020): 1110-1111.
[4] Platto, Sara, Tongtong Xue, and Ernesto Carafoli. "COVID19: an announced pandemic." (2020): 1-13.
[5] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.
[6] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
[7] Sharma, Sagar, and Simone Sharma. "Activation functions in neural networks." Towards Data Science 6.12 (2017): 310-316.
[8] Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016.
[9] Liu, Tianyi, et al. "Towards understanding the importance of shortcut connections in residual networks." arXiv preprint arXiv:1909.04653 (2019).
[10] Afshar, Parnian, et al. "Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images." Pattern Recognition Letters 138 (2020): 638-643.
[11] Nayak, Soumya Ranjan, et al. "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.
[12] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of Big Data 6.1 (2019): 1-48.
[13] Wang, Linda, Zhong Qiu Lin, and Alexander Wong. "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images." Scientific Reports 10.1 (2020): 1-12.
[14] Chandra, Tej Bahadur, et al. "Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble." Expert systems with applications 165 (2021): 113909.
[15] Gómez, Walter, Wagner Coelho Albuquerque Pereira, and Antonio Fernando C. Infantosi. "Analysis of co-occurrence texture statistics as a function of gray-level quantization for classifying breast ultrasound." IEEE transactions on medical imaging 31.10 (2012): 1889-1899.
[16] Karakanis, Stefanos, and Georgios Leontidis. "Lightweight deep learning models for detecting COVID-19 from chest X-ray images." Computers in biology and medicine 130 (2021): 104181.
[17] Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton. "Improving deep neural networks for LVCSR using rectified linear units and dropout." 2013 IEEE international conference on acoustics, speech and signal processing., 2013.
[18] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International conference on machine learning. PMLR, 2015.
[19] Rigamonti, Roberto, et al. "Learning separable filters." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.
[20] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." Thirty-first AAAI conference on artificial intelligence. 2017.
[21] Santurkar, Shibani, et al. "How does batch normalization help optimization?." Proceedings of the 32nd international conference on neural information processing systems. 2018.
[22] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012): 1097-1105.
[23] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[24] Bock, Sebastian, and Martin Weiß. "A proof of local convergence for the Adam optimizer." 2019 International Joint Conference on Neural Networks (IJCNN)., 2019.
[25] Hossin, M., & Sulaiman, M. N. (2015). "A review on evaluation metrics for data classification evaluations". International journal of data mining & knowledge management process, 5(2), 1.