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.
A lightweight network for COVID-19 detection in X-ray images

Yong Shi\textsuperscript{b}, Anda Tang\textsuperscript{a}, Yang Xiao\textsuperscript{d}, Lingfeng Niu\textsuperscript{b,c,*}

\textsuperscript{a} School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100190, China
\textsuperscript{b} Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
\textsuperscript{c} School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China
\textsuperscript{d} Faculty of Information Technology, Beijing University of Technology, Beijing, China

ARTICLE INFO

Keywords: COVID-19 detection Neural network Network pruning

ABSTRACT

The Novel Coronavirus 2019 (COVID-19) is a global pandemic which has a devastating impact. Due to its quick transmission, a prominent challenge in confronting this pandemic is the rapid diagnosis. Currently, the commonly-used diagnosis is the specific molecular tests aided with the medical imaging modalities such as chest X-ray (CXR). However, with the large demand, the diagnoses of CXR are time-consuming and laborious. Deep learning is promising for automatically diagnosing COVID-19 to ease the burden on medical systems. At present, the most applied neural networks are large, which hardly satisfy the rapid yet inexpensive requirements of COVID-19 detection. To reduce huge computation and memory demands, in this paper, we focus on implementing lightweight networks for COVID-19 detection in CXR. Concretely, we first augment data based on clinical visual features of CXR from expertise. Then, according to the fact that all the input data are CXR, we design a targeted four-layer network with either 11×11 or 3×3 kernels to recognize regional features and detail features. A pruning criterion based on the weights importance is also proposed to further prune the network. Experiments on a public COVID-19 dataset validate the effectiveness and efficiency of the proposed method.

1. Introduction

The Novel Coronavirus 2019 (COVID-19) is a public health-threatening virus that may infect the upper respiratory tract and lungs, even progress to multi-organ dysfunction. It spreads rapidly to become a global pandemic that the numbers of cases and deaths have increased on a daily basis [1]. At present, there is no effective cure for the COVID-19 disease, so an essential step in the combat against this virus is to effectively screen infected patients. Therefore the infected patients can receive in-time care and treatment, as well as be isolated to mitigate the spread [2].

Currently, the commonly-used diagnosis for COVID-19 is the specific molecular tests on respiratory samples, such as throat swab [3], aided with the medical imaging modalities, such as chest X-ray (CXR) images [4–6]. However, compared with the specific molecular tests, because there are relatively few medical professionals and experienced radiologists, the diagnoses for CXR images are more time-consuming and laborious. As the spreading of COVID-19 is fast, there is a need for a quick yet accurate automated mechanism to assist medical personnel to diagnose COVID-19 in CXR images. For this situation, COVID-19 detection with the help of deep learning model attracts a lot of attention [7–10].

Currently, available COVID-19 CXR images are usually limited. As a result, the samples may not be sufficient for training deep models. To solve these problems, Minae et al. [9] use the pretrained models as a feature extractor and train a classifier on top of them. Basu et al. [2] train AlexNet, VGG-16 and ResNet-50 based on domain extension transfer learning. Ismael et al. [10] utilize transfer learning on pre-trained deep networks to classify COVID-19 and normal CXR images. These works employ transfer learning by leveraging sophisticated deep networks and achieve good performances. However, the models are too complex to to be deployed in large-scale emergency detection of the medical field. Are there some easy-to-implement networks that have comparable performances in COVID-19 detection? To actualize the above target, in this work, we intend to design relative simpler networks for COVID-19 detection.

Another challenge is that deep models demand huge computation and memory resources. Until now, various networks have been proposed to make automated COVID-19 detection such as VGG [11], ResNet [12] or other modified networks with many layers [7,8]. These networks are too large to be applied for real-time COVID-19 detection. Considering the efficiency of the networks, some studies propose utilizing light networks.
network design such as EfficientNet and deep SqueezeNet [13–15]. These works achieve good accuracy and efficiency but limited on using the developed network or modified version. These models generally used to tackle with complicated datasets from multi-scenarios. Considering the fact that, the data sources are only the CXR images in this task, it is not necessary to use these sophisticated networks. Therefore, at this paper, instead of adopting some off-the-shelf networks directly, we intend to design more targeted but simpler networks based on the principle of ‘model capacity-model complexity’ balance.

In this paper, we propose novel lightweight networks for automatically detecting the COVID-19 in CXR images. Specifically, to solve the problems caused by insufficient data, we first conduct data augmentation based on clinical visual features extraction of CXR from expertise. In order to enhance the tendency that the model extract clinical visual features, such as interstitial abnormalities, ground-glass opacity or lesions in a particular area, we augment the images by flipped, cropped, noise added and whiten versions. And then, considering the lightweight need, we design a four-layer network architecture based on the principle of ‘model capacity-model complexity’ balance. To better extract features, we design the network architecture pointed at the CXR such as kernel size design. We utilize $11 \times 11$ kernels to extract regional features such as lesions in a relative large area, and use $3 \times 3$ kernels to recognize detail features such as interstitial abnormalities.

To further reduce the network complexity, after each iteration of network training, we prune the network by a specific pruning criterion which is based on four principles, including threshold effect, value decay modification, importance separation and sign-preserving property. Finally, the lightweight network is obtained by conducting the training and the pruning alternately until the network convergence. To validate the performance of the proposed method, we conduct experiments on

Fig. 1. The framework of the proposed lightweight network for the COVID-19 classification in CXR images.
the public COVID-19 dataset. The experimental results show that our proposed lightweight networks can achieve more than 95% accuracy and more than 98% recall on COVID-19 detection. In addition, the lightweight networks reduce about 95% of the parameters compared with the designed network prototype. In brief, the main contributions of this paper are summarized as follows.

1) We propose a lightweight network for automatically detecting the COVID-19 in CXR images.
2) To solve the problem caused by insufficient data, we conduct data augmentation based on clinical visual features of CXR from expertise.
3) To extract features in the CXR more effectively and efficiently, we design a more targeted but simpler network architecture.
4) To further compress the network, we give a pruning criterion by which the weights can be pruned and modified according to the weights importance.

The remainder of the paper is structured as follows. We review the literature concerning neural networks and lightweight networks employed for COVID-19 detection in the CXR images in Section 2. In Section 3, we introduce our methodology in detail. We carry out extensive experiments to evaluate the performance of our method in Section 4. Finally, Section 5 summarizes this paper.

2. Literature review

2.1. Neural networks for COVID-19 detection in CXR

CXR images have been proven an attractive option for the COVID-19 detection since its wide availability in diverse health care settings [16,17,11]. Motivated by the need for faster interpretation of the radiography images, a number of deep learning based methods have been proposed. Wang et al. [7] propose the COVID-Net tailored for the COVID-19 detection from CXR images. Kaur et al. [18] propose a metaheuristic deep COVID-19 screening model based on modified AlexNet for CXR. Ozturk et al. [8] implement the DarkNet with seventeen convolutional layer and different filters. Bukhari et al. [12] apply ResNet-50 network architectures to diagnose the cases of COVID-19 infections. Hemdan et al. [11] utilize seven different deep architectures to identify the COVID-19 infection. To solve the problem of inadequate COVID-19 data, some works employ transfer learning or pretrained model. Mineae et al. [9] train ResNet-18, ResNet-50, SqueezeNet, and DenseNet-121 via transfer learning to identify COVID-19 disease in the CXR. Basu et al. [2] employ domain extension transfer learning to train AlexNet, VGG-16 and ResNet-50 to predict COVID-19 from CXR. Apostolopoulos et al. [19] evaluate the performance of various networks with transfer learning and achieve remarkable results over the detection of various abnormalities in small medical image datasets. Ismael et al. [10] utilize transfer learning on pretrained deep networks to classify COVID-19 and normal CXR. More and more results have shown that deep learning based methods have promising performances in COVID-19 detection.

2.2. Lightweight networks for COVID-19 detection in CXR

Currently, most methods use deep neural networks for COVID-19 detection in CXR images. These networks have enormous amount of parameters so that they demand huge computation and storage. Considering the efficiency of the networks, lightweight networks are investigated to detect COVID-19 viral infection. Ucar et al. [13] propose a deep Bayes-SqueezeNet based COVID-19 diagnosis where the SqueezeNet comes forward for its light network design. They conduct hyper-parameters fine-tuning and data augmentation to make the proposed network obtain an out-performed COVID-19 diagnosis accuracy. Luz et al. [14] propose a new family of models based on the EfficientNet architecture [20] for COVID-19 screening in CXR images. Chaudhary et al. [15] propose to utilize EfficientNet to surpass COVID-19. They focus on models EfficientNet-B0 to EfficientNet-B3 and find that EfficientNet-B1 generates best accuracy and efficiency. Rajaraman et al. [21] utilize a pruned deep learning model for COVID-19 detection via iteratively increasing pruning percentage and achieve nearly half the compression. To our knowledge, although some efficient networks achieve good performance for COVID-19 detection in the CXR images, the networks are developed networks, or modified on them. Considering the fact that in this task of COVID-19 detection, the data source is only the CXR images rather than complicated datasets from multi-scenarios, it is not necessary to use these sophisticated networks. Therefore, at this paper, instead of adopting some off-the-shelf networks directly, we intend to design more targeted and lightweight networks.

3. Methodology

3.1. Overall description of the proposed method

In this work, based on each given CXR image, we intend to make one of the three predictions such as (1) the patient is not infected; (2) the patient is infected with COVID-19; (3) the patient is infected with other types of viral pneumonia. Therefore, the model can not only help identifying whether the patient is infected, but also determine whether the patient is infected with COVID-19 or other types of viral pneumonia so that the clinicians can choose specific treatment strategies depending on the different cause of infection. For brevity, we denote these three categories with the label ‘Normal’, ‘COVID-19’ and ‘viral pneumonia’ respectively. We train lightweight networks to automatically identifying the three classes. The Fig. 1 depicts the overall workflow of our proposed method.

In the first step, we conduct data preprocessing for the images of the three classes. Specifically, we use data augmentation to enlarge the training dataset. And then, to make the data suitable as inputs for the neural network, we convert the images into proper size and shuffle them. And we convert labels into vectors to indicate the case of ‘Normal’, ‘COVID-19’ and ‘viral pneumonia’ for each image. In the second step, we design two network architecture prototypes based on the principle of ‘model capacity-model complexity’ balance. Concretely, we choose the proper kernel sizes to better extract patterns and features from images while we constrain the layer number of the network to keep efficiency. In the third step, we conduct network training and compression. In particular, we update the weights by optimizing the loss function of the network. And then we prune weights based on a proposed non-structural pruning criterion. We alternately conduct training and pruning until convergence. We choose the final model based on over-fitting minimization and the satisfying compression ratio. Specifically, if the validation accuracy results under different compression ratios do not vary much, we choose the network with the highest compression ratio. We also set a value to control the maximum epochs. If the epoch number reach this value, we will stop the training process. Finally, we get the lightweight network based on over-fitting minimization and the satisfying sparsity. After achieving the final lightweight network, we can detect COVID-19 by implementing classification on the images. According to the ‘softmax’ values of the three categories outputted by the lightweight network, we predict the image as the category with the largest ‘softmax’ value.

3.2. Data augmentation

The sample complexity refers to the number of training data that a model needs in order to be successfully trained. The number of training data is related to the amount of information provided by the entire training sample set. The sample quality can strongly affect the generalization ability of neural networks. Currently, available COVID-19 CXR images are usually limited so that the sample complexity may not be sufficient for neural networks. To solve this problem, we utilize data
We do not choose too deep networks in the prototype. To sufficiently learn the patterns of images, we use two fully connected layers with adequate connections as the third and fourth layer.

The diagnosis for the radiography examination is generally based on clinical visual indicators. The clinical visual indicators include bilateral infiltrates, interstitial abnormalities, ground-glass opacity and lesions in a particular area [1,13,7,23]. We think that the lesions in lung show regional features in the CXR image. ‘COVID-19’ and ‘viral pneumonia’ may show lesions in different area of the lung. And the visual indicators such as ground-glass opacity show detail and local features in the CXR image. These detail features of CXR images in different classes may be distinct. Under this assumption, we propose to either use a larger convolution kernel size, such as \( 11 \times 11 \), to distinguish COVID-19 by lesions in specific areas, or use a smaller convolution kernel size, such as \( 3 \times 3 \), to make classification by detailed features. Besides, since the decisive features may appears in a specific area and this specific area cannot be determined without prior knowledge of professionals, in the feature extraction phase, we set the stride of convolution kernel step as one so that it will extract features of each area in the CXR images. According to these concept, we have designed two network architecture with different convolution kernel sizes, called ‘Conv4net-S’ and ‘Conv4net-L’ respectively. The details of the network architectures of ‘Conv4net-S’ and ‘Conv4net-L’ are given in Table 1.

### 3.4. Network training and compression

In this stage, the final network architecture will be generated by a proposed strategy where data, the initial network prototype along with some specific criteria act as a guide to prune and modify the weights of the network. The final lightweight networks based on ‘Conv4net-S’ and ‘Conv4net-L’ are denoted as ‘Light4net-S’ and ‘Light4net-L’ respectively.

We first trains the network based on data fidelity. After one iteration of weights update, we intend to zero out some weights by a non-structural pruning criterion. We design the criterion based on the following principles. First, we think that the smaller the weight value is, the less important this weight is. Therefore, we design a criterion to zero-out weights based on the weight magnitude where a weight will be set to zero if its value is below a given threshold. Second, we think that the preserved weights should be easier to be zeroed in next iteration. Thus we decrease the value of weights that are not zeroed so that they tend to be less than threshold in next iteration. Third, among the preserved weights, the important ones and the secondary ones should be further separated. We use modification when decreasing the weights so that the value reduction of the less important weights should be larger than that

### Table 1

| network      | architecture | size          | stride | padding | FLOP      | Para      |
|--------------|--------------|---------------|--------|---------|-----------|-----------|
| Conv4net-L   | conv-1       | [11, 11, 3, 64]| [1, 1, 1, 1]| SAME | 1,197,121,728 | 23,232 |
|              | ac + bn      |               |        |         |           |           |
|              | mp           | [1, 2, 2, 1]  | [1, 2, 2, 1]| SAME | –         | –         |
|              | conv-2       | [7, 7, 64, 64]| [1, 1, 1, 1]| SAME | 2,608,349,184 | 200,704 |
|              | ac + bn      |               |        |         |           |           |
|              | mp           | [1, 2, 1]     | [1, 2, 2, 1]| SAME | –         | –         |
|              | fc-3         | [57 \times 57 \times 64, 1024]| –| – | 212,926,464 | 212,926,464 |
|              | dropout + ac |               |        |         |           |           |
|              | fc-4         | [1024,3]      | –      | – | 3,072 | 3,072 |
| Conv4net-S   | conv-1       | [3, 3, 3, 64]| [1, 1, 1, 1]| SAME | 89,042,112 | 1,728 |
|              | ac + bn      |               |        |         |           |           |
|              | mp           | [1, 2, 1]     | [1, 2, 2, 1]| SAME | –         | –         |
|              | conv-2       | [3, 3, 64, 64]| [1, 1, 1, 1]| SAME | 479,084,544 | 36,864 |
|              | ac + bn      |               |        |         |           |           |
|              | mp           | [1, 2, 1]     | [1, 2, 2, 1]| SAME | –         | –         |
|              | fc-3         | [57 \times 57 \times 64, 1024]| –| – | 212,926,464 | 212,926,464 |
|              | dropout + ac |               |        |         |           |           |
|              | fc-4         | [1024,3]      | –      | – | 3,072 | 3,072 |
\[ w_{l+1}^{(t)} = \begin{cases} \text{sign}(w_l^{(t)}) \left( |w_l^{(t)}| - \frac{\left| \mu_l \right|^{3/2}}{k \sqrt{|w_l^{(t)}|}} \right), & \text{if } |w_l^{(t)}| - \mu_l > 0 \\ 0, & \text{otherwise} \end{cases} \] (1)

where the subscript \( t \) denotes the \( t \)-th iteration, the superscript \( (l) \) denotes the \( l \)-th layer, \( w_l^{(t)} \) denotes each element \( W_l^{(t)} \) of the weight matrix \( W_l^{(t)} \), \( k > 1 \) is the hyper-parameter to control the value reduction of the weight \( w_l^{(t)}, \mu_l > 0 \) is the threshold hyper-parameter of the \( l \)-th weight-layer. In Eq. (1), if a weight \( |w_l| < \mu_l \), this weight will be estimated to zero, which satisfies the first principle. If a weight \( |w_l^{(t)}| \geq \mu_l \) there is an operation as \( |w_l^{(t)}| - \frac{\left| \mu_l \right|^{3/2}}{k \sqrt{|w_l^{(t)}|}} \) which is used to shrink the weight \( w_l^{(t)} \). It satisfies the second principle. Besides, the term \( \frac{\left| \mu_l \right|^{3/2}}{k \sqrt{|w_l^{(t)}|}} \) can be seen as \( \frac{1}{k} \sqrt{\frac{\left| \mu_l \right|}{k \sqrt{|w_l^{(t)}|}}} \). It means that when \( |w_l^{(t)}| = \mu_l \), the value reduction of \( w_l^{(t)} \) is equal to \( \frac{1}{k} \). When \( |w_l^{(t)}| > \mu_l \), the value reductions are smaller than \( \frac{1}{k} \). Depending on the weights importance, the value reductions will vary. Since \( \frac{1}{k} \sqrt{\frac{\left| \mu_l \right|}{k \sqrt{|w_l^{(t)}|}}} \) will be decreasing with the increasing of the weight, the value reduction \( \frac{\left| \mu_l \right|^{3/2}}{k \sqrt{|w_l^{(t)}|}} \) will be smaller if the weight is more important. And if the weight is less important, \( \frac{1}{k} \sqrt{\frac{\left| \mu_l \right|}{k \sqrt{|w_l^{(t)}|}}} \) will be larger, so that the value reduction will be comparably larger. It controls different drops according to the weight magnitude which satisfies the third principle. In addition, since \( |w_l^{(t)}| - \mu_l | \geq 0, \left( |w_l^{(t)}| - \frac{\left| \mu_l \right|^{3/2}}{k \sqrt{|w_l^{(t)}|}} \right) \geq 0 \). Thus \( \text{sign}(w_l^{(t)}) \) keeps the positive and negative of the weights.

To indicate this function clearly, we visualize the curve of the function (1) in Fig. 2. We plot the curves of the function with different \( \mu-k \) pairs. And we plot the curve of \( y = x \) as a reference to display the value reduction effect of our proposed function. As seen in this figure, \( \mu \) is the threshold which controls the compression level. With the value of \( \mu \) increasing, more weights tend to be pruned. It means that the larger the \( \mu_l \) value is, the more significant the compression effect in \( l \)-th layer is. \( k \) controls the value reduction of the weight. The larger the value of \( k \) is, the fewer value reduction of the weight produces.

### 4. Experiments

#### 4.1. Datasets

We utilize a COVID-19 dataset of CXR images from publicly available data repository called COVID-19 radiography database \[25\]. This database comes from a group of researchers from Qatar University in Doha, Qatar and Dhaka University in Bangladesh, as well as collaborators from Pakistan and Malaysia, who cooperated with doctors to establish a database of CXR images for covid-19 positive cases, as well as normal and viral pneumonia images. All the images are in portable network graphics (PNG) file format and the resolution is 1024 x 1024 pixels, which can be easily converted to 224 x 224 or 227 x 227 pixels typically required by the popular convolutional neural networks. In this dataset, there are 219 COVID-19 positive images along with 1341 normal images and 1345 viral pneumonia images. Obviously, the phenomenon of data imbalance exists that the images of class ‘COVID-19’ is significantly fewer than that of the other two classes. In the research \[26\], the authors indicate that the data imbalance of the class distribution has a detrimental effect on the classification performance of the models. Learning with an unbalanced dataset could bias the prediction model towards the classes with more samples, leading to inferior classification models. To overcome this issue, we find extra CXR images of COVID-19 patients \(^1\) to augment the raw dataset. After that, we randomly selected around 10% of the images from each category, with a total of 400 samples as the validation set. In the validation set, there are 141 ‘COVID-19’ samples, 134 ‘Normal’ samples and 125 ‘Viral Pneumonia’ samples.

#### 4.2. Evaluation criterion

In this work, the disease detection performance of the model is quantitatively measured via several metrics. Accuracy(Acc) is the basic measurement to evaluate the classification performance. Besides the metric accuracy, we use some specific metrics like precision (Prec), recall (Rec), F1 score (F1) and Matthew Correlation Coefficient (MCC) which are the convincing evaluations in the medical field. Precision refers to the correct proportion of the positive prediction. Recall refers to the correct proportion of the predictions among the samples with positive ground truth. F1-score is the harmonic mean of the precision and the recall. MCC is a statistic from the confusion matrix to measure the quality of the model.

\[
\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN} 
\]
\[
\text{Prec} = \frac{TP}{TP + FP} 
\]
\[
\text{Rec} = \frac{TP}{TP + FN} 
\]
\[
F1 = \frac{2 \times \text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}} 
\]
\[
\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} 
\]

where TP refers to the number of True Positive samples of a category. FN refers to the number of False Negative samples of a category. TN refers to the number of True Negative samples of a category. FP refers to the number of False Positive samples of a category.

In addition, we utilize the percentage of floating-point operations compared with the non-compressed network (FLOP) and the percentage of parameters used in the networks (Para) to evaluate the compression effect. FLOPS indicates the computation cost of networks to some extent. Parameter used reflects the memory-consuming saving, which is an

\(^1\) https://aistudio.baidu.com/aistudio/datasetdetail/84451
important indicator of compression effect. Generally speaking, the smaller the value of the FLOP and the parameter used are, the more efficient the model is.

4.3. Experimental results

To verify the effectiveness of the data augmentation, we perform our lightweight network 'Light4net-S' and 'Light4net-L' on the raw dataset and augmented dataset respectively. Besides, to demonstrate the advantages of our kernel size assumption, we compare our networks to a network with 5 × 5 kernels. We denote this network as 'Light4net-5 × 5'. Except for the convolution size, other settings of 'Light4net-5 × 5' are consistent with our networks. We list the experimental results in Table 2.

The best results in terms of Acc, FLOP and Para are highlighted in the bold form in the Table 2. From the table, it can be observed that, (1) compared with the disease detection performance on the raw dataset, the performances on the augmented dataset are promoted in terms of the accuracy, precision, recall, F1 score and MCC. It verifies the effectiveness of the data augmentation. In addition, the compression ratio of each network is approximately doubled under the data augmentation. The reason may be that the data augmentation reduces the pressure of feature extraction and transformation. (2) On considering in the kernel size design, the disease detection performances of the networks with three kernel designs are comparable on the raw dataset. However, in the case of data augmentation, compared with the network with 5 × 5 kernels, our proposed networks both outperform with the margin of about 4% in terms of disease detection performance. The possible reason is that when data features are difficult to extract, the convolution size...
has little impact on the model performance. After the data is augmented, the model features are easier to extract. Thus the network needs a proper kernel size to extract features. It verifies the effectiveness of our kernel size design.

Then we focus on the precision (Prec), recall (Rec), F1 score (F1) and Matthew Correlation Coefficient (MCC) of methods in Table 2. Since our task is a multi-classification, we interpret the precision, recall, F1 score and MCC results of each class. In order to better display these indicators, we choose macro-average to calculate the four and list the average results in the table. To better analyze our model, we further plot the confusion matrix of our method in Fig. 3. Overall, our method has the best results in the table. To better analyze our model, we further plot the confusion matrix of our method in Fig. 3. Overall, our method has the best performance in terms of the four measures.

To further analyze the network training process of our method, we report the variation of validation accuracy, training loss, FLOP and parameter usage during training. From Fig. 4, we can see that, (1) As training proceeds, the training loss is decreasing with fluctuations. Its first cliff drop occurs about the 10 thousand iterations. With the sharp decline of the loss value, the validation accuracy has improved significantly at this moment. And then the training loss achieves a large reduction at around 13–18 thousand iterations. Consequently, the training loss converges when the validation accuracy has a slight increase, which represents the model saturation. (2) FLOP and Para keep decreasing. This phenomenon validates the effect of the compression ability of our method. In general, the downtrend of FLOP decreases with the downtrend of Para. The reason for these phenomena is that the reduction of computation is based on the sparsity of weights. The computational cost becomes less and less because of the continuing reduction of weights.

In addition, we report the FLOP and Para in each layer to test the effect of our method. In general, the convolution layer has a small number of parameters but a large amount of computation due to the mechanism of parameter sharing. The fully-connected layer does not have a very large amount of computation, but has a very large number of parameters and a large parameter redundancy. By tuning the \( \mu_1 \) of each layer, we can achieve various FLOP and Para level according to the requirements for the computation or storage. It also needs to consider the network performance when tuning the \( \mu_1 \). Since the high compression ratio in the convolution layers would influence feature extraction ability and expressive power, it needs to enforce a certain compression ratio to between different layers. Former studies state that features in primary layers may be more important than the features transformed in the fully-connected layers [27]. Therefore, we mainly zero-out the weights in

| method(performance) | class          | Prec (%) | Rec (%) | F1 (%) | MCC (%) |
|---------------------|----------------|----------|---------|--------|---------|
| Conv4net-L          | COVID-19       | 95.17    | 97.87   | 96.50  | 94.58   |
|                     | Normal         | 96.88    | 92.54   | 94.66  | 92.11   |
|                     | Viral Pneumonia| 94.49    | 96.0    | 95.24  | 93.05   |
|                     | average        | 95.52    | 95.47   | 95.47  | 93.25   |
| \( \ell_2 \)        | COVID-19       | 96.48    | 97.16   | 96.82  | 95.08   |
|                     | Normal         | 97.66    | 93.28   | 95.42  | 93.25   |
|                     | Viral Pneumonia| 93.85    | 97.6    | 95.69  | 93.71   |
|                     | average        | 95.99    | 96.02   | 95.98  | 94.01   |
| \( \ell_1 \)        | COVID-19       | 95.77    | 96.45   | 96.11  | 93.99   |
|                     | Normal         | 96.85    | 91.79   | 94.25  | 91.55   |
|                     | Viral Pneumonia| 90.84    | 95.2    | 92.97  | 89.72   |
|                     | average        | 94.49    | 94.48   | 94.44  | 91.75   |
| Light4net-L         | COVID-19       | 97.22    | 99.30   | 98.25  | 97.29   |
|                     | Normal         | 97.64    | 92.54   | 95.02  | 92.68   |
|                     | Viral Pneumonia| 93.80    | 96.8    | 95.28  | 93.10   |
|                     | average        | 96.22    | 96.21   | 96.20  | 94.36   |

**Fig. 4.** Training loss, validation accuracy, FLOP and Para used during training.
third layer to achieve high compression ratio and keep the important features in this work.

We report the amount and percentage of FLOP and Para of each layer in ‘Light4net-L’ and its prototype respectively in Table 4, where conv-1 and conv-2 represent the first and the second layer, fc-3 and fc-4 represent the third and the fourth layer. From the table, it can be observed that (1) For the prototype network, the convolution layers possess a small number of parameters accounting for 1.0% of the whole, while it produces the most of computation accounting for 95% of the whole. (2) The reduction of computation is mainly derived from the second layer accounting for 65% of the whole reduction. The reduction of parameter amount mainly comes from the third layer accounting for 99% of the whole reduction. The results coincide with the above statement.

5. Conclusion

A rapid diagnosis method of COVID-19 is a prominent factor to control the global pandemic situations. With the increasing demand for COVID-19 screenings, deep learning has emerged as a promising supplement to automatically diagnose COVID-19. Some limitations of the resource-demand deep networks reveal a need for lightweight networks to make a quick and accurate diagnosis of COVID-19. In this paper, we propose lightweight networks for automatically detecting the COVID-19 in CXR images. To solve the problem caused by insufficient data, we first conduct data augmentation according to the CXR clinical visual features from expertise. Then, in order to extract features in the CXR more effectively and efficiently, we design a more targeted but simpler network architecture. Finally, to further compress the network, we give a pruning criterion with pruning effect, value decay effect, importance separation effect, and sign-preserving effect on weights.

To validate the performance of the proposed method, we conduct experiments on the public COVID-19 dataset. The experimental results show that our proposed lightweight networks can achieve more than 95% validation accuracy and more than 98% recall on COVID-19 detection. In addition, the lightweight networks reduce about 95% of the parameters compared with the designed network prototype. Thus the lightweight networks have competitive classification performance with a few parameter usage. And the F1 and MCC results prove that our method obtains a stable diagnosis model. Besides, the high recall rate means that our models miss few cases of the ‘COVID-19’. This diagnosis results is critical to COVID-19 detection so that the medical systems can take rapid measures of isolation of infected patients to slow down the virus propagation.

For future work, we take ethical considerations of the neural network based method for COVID-19 detection in X-ray images. As a learning based method, the COVID-19 detection effect of the neural network depends on the quantity and quality of X-ray data. However, since these training X-ray images are from patients, the gap between technical feasibility and what is allowed from privacy legislation is growing. To integrate the medical data from medical institutions worldwide without violating privacy rights and interests, in the future, federated learning can be considered to build a decentralized model, so as to share the model and learn parameters without sharing data.

CRediT authorship contribution statement

Yong Shi: Supervision, Writing - review & editing. Anda Tang: Conceptualization, Methodology, Software, Writing - original draft, Validation. Yang Xiao: Conceptualization, Methodology, Software, Writing - review & editing. Lingfeng Niu: Conceptualization, Data curation, Supervision, Writing - review & editing.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgements

This paper has been supported by the National Science Foundation of China (No.12271503, No.11991021,No.72231010, No.71932008).

References

[1] T. Singhal, A review of coronavirus disease-2019 (covid-19), Ind. J. Pediatr. 87 (4) (2020) 281–286.
[2] S. Basi, S. Mitra, N. Saha, Deep learning for screening covid-19 using chest x-ray images, IEEE Symposium Series on Computational Intelligence, IEEE 2020 (2020) 2521–2527.
[3] Y.-H. Jin, L. Cai, Z.-S. Cheng, H. Cheng, T. Deng, Y.-P. Fan, C. Fang, D. Huang, L.-Q. Huang, Q. Huang, et al., A rapid advice guideline for the diagnosis and treatment of 2019 novel coronavirus (2019-ncov) infected pneumonia (standard version), Military Med. Res. 7 (1) (2020) 1–23.
[4] 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 (1).
[5] G.D. Rubin, C.J. Ryerson, L.B. Haramati, N. Sverzellati, J.P. Kanne, S. Raasof, N.W. Schluger, A. Volpi, J.-J. Yim, L.B. Martin, et al., The role of chest imaging in patient management during the covid-19 pandemic: a multiracial consensus statement from the Fleischner society, Chest 158 (1) (2020) 106–116.
[6] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei, et al., Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, china: a descriptive study, Lancet 395 (10223) (2020) 507–513.
[7] L. Wang, Z.Q. Lin, A. Wong, Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, Sci. Rep. 10 (1) (2020) 1–12.
[8] 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, Comput. Biol. Med. 121 (2020), 103792.
[9] S. Minare, R. Kafieh, M. Sonka, S. Yazdani, G.J. Soufi, Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning, Med. Image Anal. 65 (2020), 101794.
[10] A.M. Ismaiel, A. Şengür, Deep learning approaches for covid-19 detection based on chest x-ray images, Expert Syst. Appl. 164 (2021), 114054.
[11] 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.
[12] S.U.K. Bukhari, S.S.K. Bukhari, A. Syed, S.S.H. Shah, The diagnostic evaluation of convolutional neural network (cnn) for the assessment of chest x-ray of patients infected with covid-19, MedRxiv.
[13] F. Ucar, D. Korkmaz, Covidiagnosis-net: Deep bayes-squeezenet based diagnosis of the coronavirus disease 2019 (covid-19) from x-ray images, Med. Hypotheses 140 (2020), 109761.
[14] E. Luz, F. Silva, R. Silva, L. Silva, J. Guimarães, G. Miozzo, G. Moreira, D. Menotti, Towards an effective and efficient deep learning model for covid-19 patterns detection in x-ray images, Res. Biomed. Eng. 38 (1) (2022) 149–162.
[15] Y. Chaudhary, M. Mehta, R. Sharma, D. Gupta, A. Khanna, J.J. Rodrigues, Efficient-covidnet: deep learning based covid-19 detection from chest x-ray images, IEEE international conference on e-health networking, application & services, IEEE 2021 (2020) 1–6.
[16] X. Li, C. Li, D. Zhu, Covid-mobileexpert: On-device covid-19 screening using snapshots of chest x-ray, arXiv preprint arXiv:2004.03042.
[17] X. Li, C. Li, D. Zhu, Covid-mobileexpert: On-device covid-19 patient triage and follow-up using chest x-rays, IEEE international conference on bioinformatics and biomedicine, IEEE 2020 (2020) 1063–1067.
[18] M. Kaur, V. Kumar, V. Yadav, D. Singh, N. Kumar, N.N. Das, Metaheuristic-based deep covid-19 screening model from chest x-ray images, J. Healthcare Eng. (2021).
[19] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. 43 (2) (2020) 635–640.
[20] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.
[21] S. Rajaraman, J. Siegelman, P.O. Alderson, L.S. Folio, L.R. Folio, S.K. Antani, Iteratively pruned deep learning ensembles for covid-19 detection in chest x-rays, IEEE Access 8 (2020) 115041–115050.
[22] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press (2016).
[23] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, et al., Clinical features of patients infected with 2019 novel coronavirus in wuhan, china, Lancet 395 (10223) (2020) 497–506.
[24] K.K. Pal, K. Sudeep, Preprocessing for image classification by convolutional neural networks, in: 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, IEEE, 2016, pp. 1778–1781.
[25] R.S. of North America, Covid-19 radiography database, https://www.kaggle.com/tawsifurrahman/covid19-radiography-database, https://drive.google.com/file/d/1xt7g5LkZnX09e1a8K9a8X6rGFnO6jdl/view?usp=sharing (2020).
[26] M. Buda, A. Maki, M.A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Networks 106 (2018) 249–259.
[27] J. Yoon, S.J. Hwang, Combined group and exclusive sparsity for deep neural networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 3958–3966.