



**ARTICLE**

## COVID-19 Detection via a 6-Layer Deep Convolutional Neural Network

Shouming Hou and Ji Han\*

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China

\*Corresponding Author: Ji Han. Email: HanJi@home.hpu.edu.cn

Received: 12 March 2021 Accepted: 05 August 2021

### ABSTRACT

Many people around the world have lost their lives due to COVID-19. The symptoms of most COVID-19 patients are fever, tiredness and dry cough, and the disease can easily spread to those around them. If the infected people can be detected early, this will help local authorities control the speed of the virus, and the infected can also be treated in time. We proposed a six-layer convolutional neural network combined with max pooling, batch normalization and Adam algorithm to improve the detection effect of COVID-19 patients. In the 10-fold cross-validation methods, our method is superior to several state-of-the-art methods. In addition, we use Grad-CAM technology to realize heat map visualization to observe the process of model training and detection.

### KEYWORDS

COVID-19; deep learning; convolutional neural network; max pooling; batch normalization; Adam; Grad-CAM

## 1 Introduction

COVID-19 is a disease that easily spreads among people. It originated from the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [1]. The spread of this disease includes human-to-human contact, or contact with polluted air, as well as respiratory droplets and feces [2]. Therefore, the authorities have adopted a series of measures, including wearing masks in public places, quarantining people entering the country from abroad, and reminding people of the country not to travel to high-risk areas [3].

Early detection of COVID-19 helps various departments to take preventive and control measures in advance to protect the safety of local residents. The most commonly used testing methods are real-time reverse transcription polymerase chain reaction (RT-PCR) and point-of-care (POC) methods [4]. POC can be used to detect genes encoding viral proteins in respiratory samples, and this method test takes less than an hour to get the result. POC can be used to detect genes encoding viral proteins in respiratory samples. This method only takes a few minutes to get the test results. However, the sensitivity of these two methods may not be sufficient to detect early infections caused by low virus concentrations.



Traditional artificial intelligence (AI) methods [5] may not work well on handling complicated image processing tasks [6]. Now, many researchers use deep learning (DL) methods to optimize the detection of certain diseases based on medical images. For example, Guo et al. [7] employed ResNet-18 to detect Thyroid Ultrasound Standard Plane images. Wu [8] chose to combine wavelet Renyi entropy with their proposed three-segment biogeography-based optimization. Ni et al. [9] proposed a deep learning approach (DPA) for COVID-19 detection. Wang [10] combined graph convolutional network (GCN) with convolutional neural network (CNN) using deep feature fusion method. Wang [11] proposed a novel CCSH network to detect COVID-19. There are many other successful applications of deep learning cases [12–14], which all prove the powerfulness of DL.

In addition, more and more researchers also use transfer learning. Transfer learning is suitable for situations where a large number of source data features in the training model are similar to a small number of target data features in the detection model, so it is not suitable for our experiments. In this paper, we collected CT images of COVID-19 and proposed a 6-layer convolutional neural network method to detect COVID-19. Max pooling proved to perform better than other traditional pooling methods. Batch normalization effectively improves the training speed of convolutional neural networks. Adam algorithm is better than other algorithms in terms of model training effect.

The remaining chapters of this paper are as follows. Section 2 introduces the collection of datasets and the characteristics of the datasets. Section 3 describes the various modules of the convolutional neural network. Section 4 introduces the model we built and analyzes the experimental results. In the last section, we made a summary of our experiments and results.

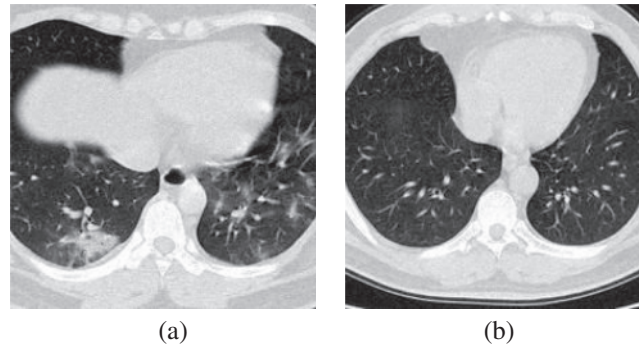
## 2 Dataset

The image dataset was from [15]. In the experiment, Philips Ingenuity 64-line spiral CT machines were used to collect lung pictures. During the CT scan, keep the patient supine and breathe deeply back, which helps scan from the lung tip to the rib diaphragm angle.

The image slices we collected came from 142 COVID-19 patients and 142 healthy people. From the CT images of each subject, 1–4 slices were selected as experimental data, and the resolution rate of all images was  $1,024 \times 1,024$ . Table 1 shows the characteristic data of the collected objects. In Fig. 1, we can find that the lung biopsy samples of COVID-19 patients have obvious white lesions.

**Table 1:** The characteristic data of the collected objects

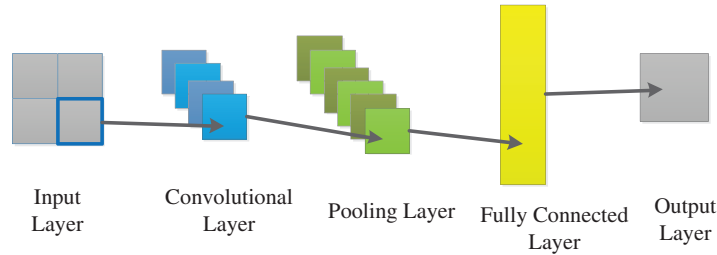
Experiment subject	Number of subjects	Sex(male/female)	Age	Slice images
Patients	142	95/47	22 to 91	320
Healthy	142	88/54	21 to 76	320



**Figure 1:** The sample of lung slices from patient with COVID-19 and healthy person. (a) COVID-19 (b) Normal

### 3 Methodology-CNN

Convolutional neural network is a classifier, this method can identify normal images and abnormal images from medical images [16]. A classic neural network consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer and an output layer [17,18]. The convolutional layer and the pooling layer are used to extract image features, and the fully connected layer is used for image classification. Fig. 2 shows the flowchart of CNN.



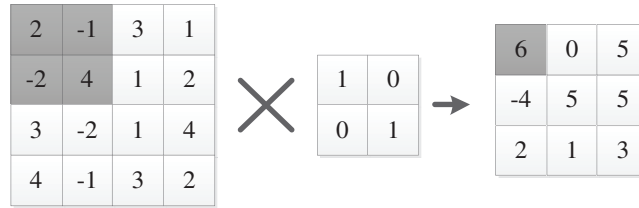
**Figure 2:** The flowchart of CNN

#### 3.1 Convolution Layer

In the convolutional layer, the input image data and the kernel are convolved to output the feature map. The operation of the convolutional layer contains three hyperparameters [19], which are kernel size, filter depth, and stride. The kernel size represents the pixel size of the convolution filter. The filter depth controls the number of output feature maps, representing the number of filters in the convolutional layer. The stride determines how many pixels the filter will skip in each convolution [20]. The case of convolution operation is shown in Fig. 3. And the convolution operation is as follows:

$$T_x^a = f \left( b_x^v + \sum_{v \in N_x} T_v^{a-1} h_{v,x}^a \right) \tag{1}$$

where  $T_x^a$  represents the output of the xth feature map of the ath convolutional layer,  $N_x$  is the input feature subset,  $h$  is the convolution core matrix,  $b$  refers to the offset value of the feature map, and  $f$  is the activation function.

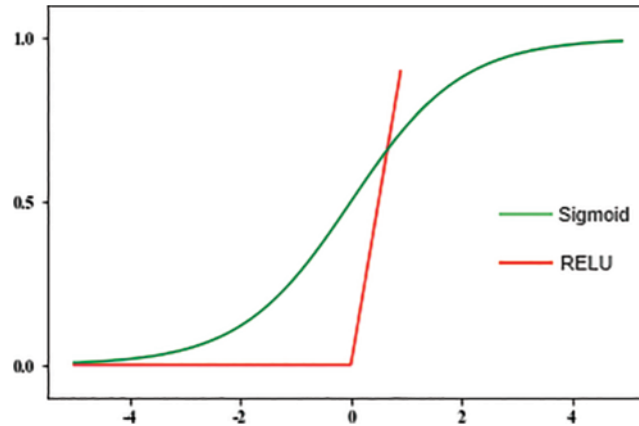


**Figure 3:** The case of convolution operation

### 3.2 ReLU Function

The convolutional layer will be followed by an activation function, such as Sigmoid and Relu, their activation curve is shown in Fig. 4. Sigmoid is a traditional non-linear activation function, and its output is bounded, and the output value ranges from 0 to 1 [21]. The activation formula is as follows:

$$\text{Sigmoid}(\gamma) = \frac{1}{1 + e^{-\gamma}} \quad (2)$$



**Figure 4:** The activation curve of Sigmoid and ReLU

Since the Sigmoid function encounters an input value that is too large or too small, its curve slope will tend to zero, which is likely to cause the gradient descent of the neural network. However, some of the output values of the ReLU function are 0, which avoids the problems existing in Sigmoid, which can reduce overfitting and solve the gradient descent problem. Therefore, compared with Sigmoid, the ReLU function can speed up model training. The calculation formula of Relu is as follows:

$$\text{ReLU}(\gamma) = \begin{cases} \gamma & \gamma > 0 \\ 0 & \gamma < 0 \end{cases} \quad (3)$$

### 3.3 Batch Normalization

In the experiment, we used batch normalization technology [22] to solve the problem of internal covariate shift. This technology ensures that the data set distribution after convolution is more uniform, thereby increasing the learning rate of the training model and speeding up the

training process [23]. The calculation steps for batch normalization are as follow. First, calculate the average of the minibatch  $\varphi$  based on the input value  $w_i$ .

$$\varphi = \frac{1}{L} \sum_{i=1}^m w_i \tag{4}$$

Second, calculate the variance of the minibatch

$$\rho^2 = \frac{1}{L} \sum_{i=1}^m (w_i - \varphi)^2 \tag{5}$$

Third, in order to prevent abnormal operations, we added a constant  $\tau$  to the denominator

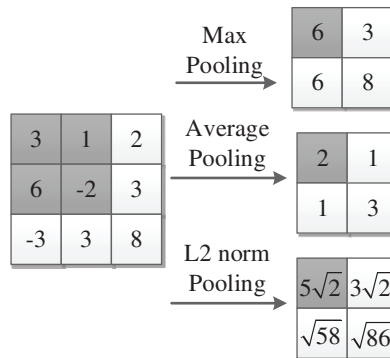
$$F_i = \frac{w_i - \varphi}{\sqrt{\rho^2 + \tau}} \tag{6}$$

Finally, multiply  $F_i$  by the scale  $\alpha$  and add the shift  $\theta$ .

$$F_i^{\sim} = \alpha F_i + \theta \tag{7}$$

### 3.4 Pooling

The pooling layer is used to reduce the dimensionality of the feature vector output after the convolution operation, which can prevent overfitting. The most common pooling methods are max pooling (Mp), average pooling (Ap) and  $l_2$  norm pooling ( $l_2p$ ) [24]. Mp calculates the max value of the pooling area, and Ap outputs the average value of the pooling area. The value obtained by  $l_2p$  is the arithmetic square root of the sum of the squares of the elements in the pooling area. The three pooling operations are shown in Fig. 5.



**Figure 5:** The three pooling operations

Suppose the pooling area is  $G$ , and the dataset to be activated in  $G$  is  $D$ . The definition of  $D$  is as follows:

$$D = [d_\delta | \delta \in G] \tag{8}$$

The  $M_p$  was defined as

$$P_M = \text{Max}\{D_G\} \quad (9)$$

The  $A_p$  was defined as

$$P_A = \frac{\sum^{D_G}}{|D_G|} \quad (10)$$

The  $l_2p$  was defined as

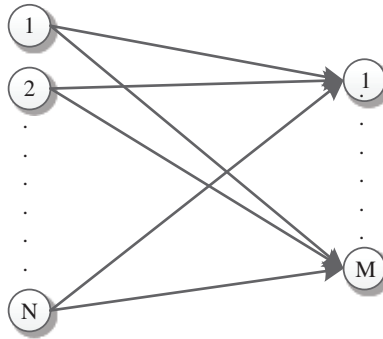
$$P_{L_2} = \sqrt{\sum D_G^2} \quad (11)$$

### 3.5 Fully Connected Layer and Softmax

Fully connected layer (FCL) [25] is used to classify feature images after pooling [26]. And the neurons in the fully connected layer are fully connected to the neurons in the adjacent layer. The flowchart of the FCL is shown in Fig. 6. The calculation formula of the FCL is as follows:

$$I^{\sim} = i * w^{\sim} + b^{\sim} \quad (12)$$

where  $i$  represents the value input to the FCL, and  $w^{\sim}$  and  $b^{\sim}$  are the weight matrix and bias respectively [27].  $I^{\sim}$  is the output of the FCL.



**Figure 6:** The flowchart of the FCL

When the FCL is used for linear feature extraction, an activation function will follow. The most commonly used is the softmax activation function [28]. Its calculation formula is as follows:

$$\beta_j(\partial) = \frac{S(y_j)S(\partial|y_j)}{\sum_{j=1}^N S(y_j)S(\partial|y_j)} \quad (13)$$

Let  $y_j(\partial) = \ln(S(y_j)S(\partial|y_j))$

$$\beta_j(\partial) = \frac{\exp(y_j(\partial))}{\sum_{j=1}^N \exp(y_j(\partial))} \quad (14)$$

where  $\vartheta$  is the input value and  $j$  represents the  $j$ th cluster.  $S(\vartheta|y_j)$  is the conditional probability of  $\vartheta$  belonging to the  $j$ th cluster, and  $S(y_j)$  is the prior probability of the cluster. The result of  $\beta_j(\vartheta)$  is between 0 and 1.

### 3.6 Training Algorithms

In the experiment, for the complexity of the deep learning training model, we chose a suitable optimization algorithm to optimize the model. Adam (Adaptive momentum) [29,30] is a gradient descent optimization technique that calculates the learning rate of each step by controlling the first and second moments of the gradient. And it can also correct the deviation and keep the parameters stable [31]. The formula for Adam is as follows:

$$k_z = \mu_1 k_{z-1} + (1 - \mu_1) * c \quad (15)$$

$$L_z = \mu_2 L_{z-1} + (1 - \mu_2) * c^2 \quad (16)$$

$$k_z^- = \frac{k_z}{1 - \mu_1^z} \quad (17)$$

$$L_z^- = \frac{L_z}{1 - \mu_2^z} \quad (18)$$

$$\vartheta_{z+1} = \vartheta_z - k_z^- * \frac{\sigma}{\sqrt{L_z^-} + \varepsilon} \quad (19)$$

where  $c$  represents the calculated gradient,  $k_z$  is the first moment of the gradient  $c$ , and  $L_z$  represents the second moment of the gradient  $c$ .  $\mu_1$  represents the first moment attenuation coefficient, and  $\mu_2$  represents the second moment attenuation coefficient.  $\vartheta$  represents the parameter to be updated, and  $K_z^-$  and  $L_z^-$  are the offset correction of  $K_z$  and  $L_z$ , respectively.

### 3.7 Cross Validation

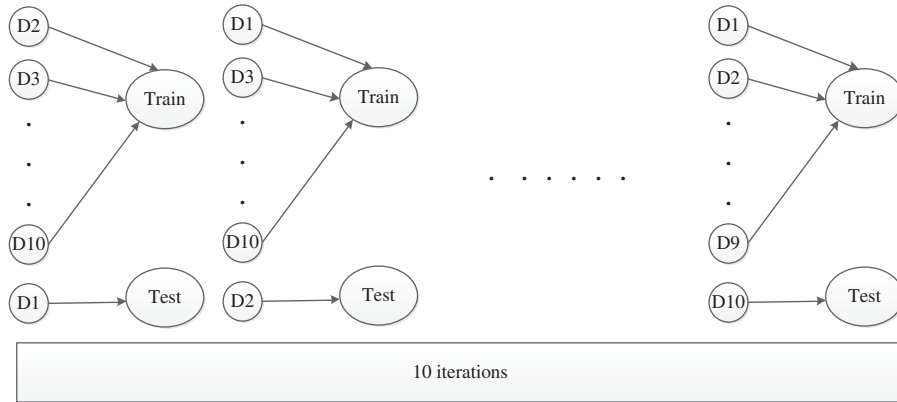
In the experiment, we need to train and test the data set to verify the detection effect of the model. In order to analyze the performance of the constructed model, we adopted the cross-validation technology, which is a widely used method for optimizing and evaluating model performance [32,33].

We chose the  $K$ -fold cross-validation method to divide the collected data set into  $K$  equal subsets.  $K-1$  equal subsets are trained in the experiment, and the one that is not trained is used for testing. This process is iterated  $k$  times, and each subset will be used for testing. In this paper, we used 10-fold cross validation, which has very little error in evaluating model performance. The operation of 10-fold cross validation is shown in Fig. 7.

### 3.8 Measures and Heatmap

In order to evaluate the performance of the built CNN model in training and testing the data set in the experiment, we selected some ideal indicators, including Sensitivity ( $E_1$ ), Specificity ( $E_2$ ), Precision ( $E_3$ ), Accuracy ( $E_4$ ), F1-Score ( $E_5$ ), Matthews correlation coefficient ( $E_6$ ), Fowlkes Mallows index ( $E_7$ ) [34]. The calculation formulas for these indicators are as follows:

$$E_1 = \frac{t_1}{t_1 + f_2} \quad (20)$$



**Figure 7:** The operation of 10-fold cross validation

$$E_2 = \frac{t_2}{f_1 + t_2} \quad (21)$$

$$E_3 = \frac{t_1}{f_1 + t_1} \quad (22)$$

$$E_4 = \frac{t_1 + t_2}{t_1 + t_2 + f_1 + f_2} \quad (23)$$

$$E_5 = \frac{2t_1}{f_1 + f_2 + 2t_1} \quad (24)$$

$$E_6 = \frac{t_1 t_2 - f_1 f_2}{\sqrt{(t_1 + f_1)(t_1 + f_2)(t_2 + f_1)(t_2 + f_2)}} \quad (25)$$

$$E_7 = \sqrt{\frac{t_1}{(t_1 + f_1)} \frac{t_1}{(t_1 + f_2)}} \quad (26)$$

where  $t_1$ ,  $t_2$ ,  $f_1$ , and  $f_2$  represent true positive (TP), true negative (TN), false positive (FP), and false negative (FN), respectively.  $t_1$  and  $t_2$  indicate the correct classification of COVID-19 patients and healthy people.  $f_1$  and  $f_2$  indicate the misclassification of COVID-19 patients and healthy people, respectively.

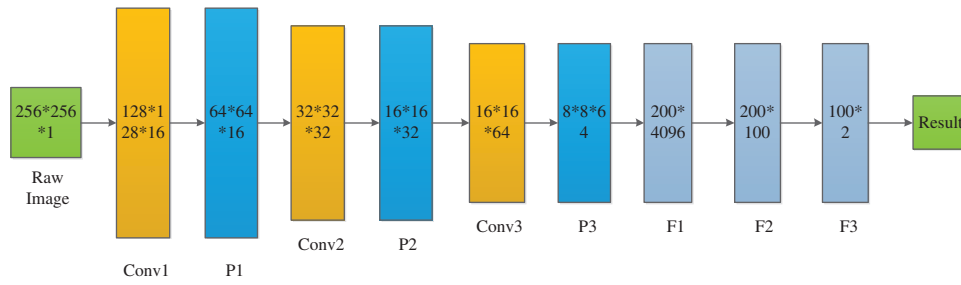
In the deep learning model, the entire training process cannot be visualized intuitively, so it is easy for radiologists to be confused whether the model can accurately detect abnormal areas in the CT image. We applied Grad-CAM technology to our model so that image features can be colored to easily distinguish between normal and abnormal regions in CT images [35]. Grad-CAM technology helps the model to accurately focus on key areas.

## 4 Experiment Results and Discussions

### 4.1 Structure of Proposed CNN

In the paper, we built a six-layer CNN. The architecture of the CNN is shown in Fig. 8. This CNN includes three convolutional layers, three max pooling layers and three FCLs. The parameters in the activation map are marked on each layer.





**Figure 8:** The architecture diagram of the 6-layer CNN

**4.2 Statistical Results**

Table 2 shows the 7 evaluation index results of the 6-layer CNN we built under 10-fold cross-validation. The results of Sensitivity, Specificity, Precision, Accuracy, F1-Score, Matthews correlation coefficient, and Fowlkes Mallows index are 90.97%, 89.58%, 89.51%, 89.52%, 89.58%, 79.07%, and 89.59, respectively.

**Table 2:** Evaluation results of 6-layer CNN

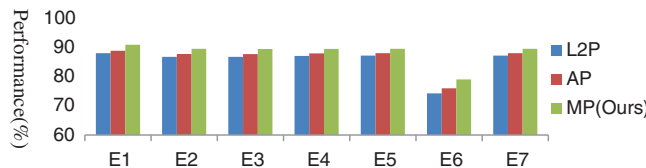
R	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>
1	87.19	89.06	88.85	88.13	88.01	76.26	88.02
2	89.38	85.94	86.40	87.66	87.86	75.36	87.88
3	90.00	89.69	89.72	89.84	89.86	79.69	89.86
4	90.00	85.00	85.71	87.50	87.80	75.09	87.83
5	88.13	90.00	89.81	89.06	88.96	78.14	88.96
6	90.63	85.94	86.57	88.28	88.55	76.65	88.57
7	92.50	88.13	88.62	90.31	90.52	80.70	90.54
8	89.06	85.94	86.36	87.50	87.69	75.04	87.70
9	88.13	90.00	89.81	89.06	88.96	78.14	88.96
10	89.69	85.00	85.67	87.34	87.63	74.77	87.66
AV	89.47±1.50	87.47±2.11	87.75±1.76	88.47±1.05	88.59±0.99	76.98±2.09	88.60±0.99

**4.3 Pooling Comparison**

In the experiment, we applied the Mp layer to the six-layer CNN model, and compared with  $l_2p$  layer and Ap layer. Table 3 shows the comparison of the results of the 3 pooling methods based on 7 indicators under the 10-fold cross validation. In Fig. 9, we can see the performance comparison of different pooling more intuitively, and the results show that using the Mp method can obtain better results than the other two pooling methods.

**Table 3:** Comparison of three pooling techniques

	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>
L2P	85.56±2.53	84.41±2.40	84.60±2.21	84.98±2.15	85.07±2.16	69.99±4.29	85.07±2.16
AP	86.47±2.45	84.91±2.88	85.17±2.58	85.69±2.32	85.80±2.27	71.41±4.63	85.81±2.27
MP	89.47±1.50	87.47±2.11	87.75±1.76	88.47±1.05	88.59±0.99	76.98±2.09	88.60±0.99
(Ours)							



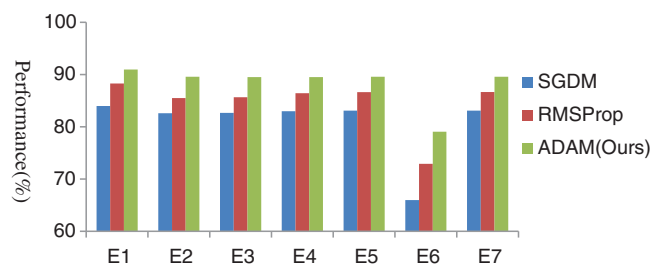
**Figure 9:** Comparison of three pooling techniques

#### 4.4 Training Algorithm Comparison

In the experiment, we used Adam algorithm to optimize the 6-layer convolutional neural network and compared it with the SGDM and RMSProp optimization algorithms. SGDM is based on first-order momentum to reduce the oscillation in the best direction along the steepest path during the gradient descent process. RMSProp is an adaptive learning rate method that normalizes the gradient by using the exponential moving average of the gradient magnitude of each parameter. The experimental results are shown in Table 4 and Fig. 10. From the perspective of sensitivity ( $E_1$ ), the results of SGDM and RMSProp algorithms are 83.99% and 88.29%, respectively, but ADAM algorithm reaches 90.97%, so ADAM algorithm performs better than SGDM and RMSProp algorithms. At the same time, the result of ADAM algorithm based on index Matthews correlation coefficient ( $E_6$ ) is 79.17%, which is obviously better than the results 65.98% and 72.92% obtained by SGDM and RMSProp algorithms. Therefore, the performance of the ADAM algorithm in the model we built is significantly better than the other two algorithms.

**Table 4:** Comparison of three training algorithm

	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$
SGDM	82.56±1.43	80.53±2.08	80.94±1.74	81.55±1.44	81.74±1.36	63.12±2.86	81.74±1.36
RMSProp	85.72±2.57	83.38±2.12	83.77±1.90	84.55±1.89	84.72±1.93	69.14±3.78	84.73±1.94
ADAM (Ours)	89.47±1.50	87.47±2.11	87.75±1.76	88.47±1.05	88.59±0.99	76.98±2.09	88.60±0.99



**Figure 10:** Comparison of three training algorithm

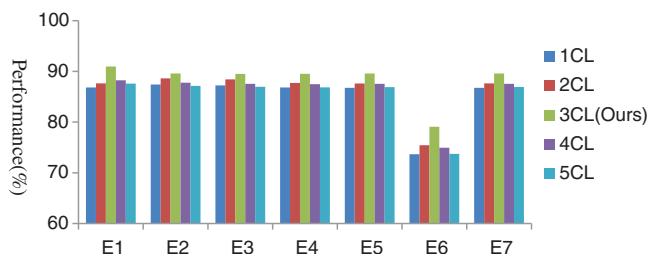
#### 4.5 Comparison of Different Number of Conv Layers

When using CNN to detect CT images, increasing the number of convolutional layers is beneficial to improve the detection effect. But this does not mean that the more layers of convolutional layers, the better the result of the CNN. In order to select an appropriate number of convolutional layers, we compared the performance of convolutional neural networks with different convolutional layers in our experiments. The experimental results are shown in Table 5. We found that when the number of convolutional layers increased from 1 to 3, the performance

became better and better, but when the number of layers continued to increase, the effect began to decrease. Therefore, our model works best when the number of convolutional layers is 3. At the same time, we made a clearer comparison in Fig. 11.

**Table 5:** Comparison of different conv layers

# of Conv layer	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>
1	85.81±1.02	85.44±1.98	85.52±1.72	85.63±1.20	85.66±1.11	71.26±2.40	85.66±1.11
2	86.84±0.79	86.41±2.22	86.51±1.91	86.63±1.09	86.66±0.96	73.27±2.17	86.67±0.96
3	89.47±1.50	87.47±2.11	87.75±1.76	88.47±1.05	88.59±0.99	76.98±2.09	88.60±0.99
4	86.09±2.15	86.72±1.06	86.64±0.91	86.41±1.06	86.35±1.20	72.84±2.10	86.36±1.19
5	85.78±1.80	85.66±1.48	85.69±1.28	85.72±1.14	85.72±1.18	71.46±2.27	85.73±1.18



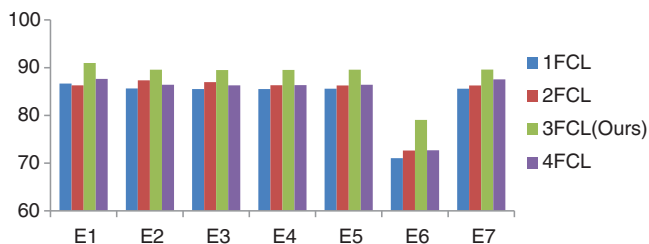
**Figure 11:** Comparison of different conv layers (conv layers = CL)

#### 4.6 Comparison of Different Number of FCLs

Most convolutional neural network models contain 2 fully connected layers, which can already achieve good results. But in our experiment, comparing the performance of models containing different numbers of FCLs, the experimental results are shown in Table 6, and Fig. 12 clearly shows their performance differences under various indicators. We found that when the number of FCLs in the model is 3, the performance is best.

**Table 6:** Comparison of different FCL layers

# of FCL	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>
1	85.13±1.55	83.97±1.68	84.17±1.35	84.55±0.96	84.63±0.96	69.12±1.92	84.64±0.96
2	84.88±1.41	85.88±1.47	85.75±1.23	85.37±0.95	85.30±0.96	70.77±1.88	85.31±0.96
3	89.47±1.50	87.47±2.11	87.75±1.76	88.47±1.05	88.59±0.99	76.98±2.09	88.60±0.99
4	86.06±1.60	84.59±1.83	84.84±1.46	85.33±1.01	85.43±1.00	70.69±2.00	85.44±1.00



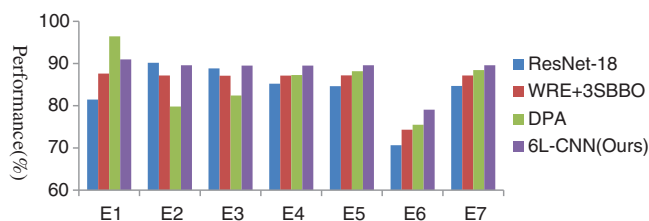
**Figure 12:** Comparison of different FCL layers

#### 4.7 Comparison to State-of-the-Art Approaches

In the experiment, we compared our proposed method with several advanced methods, including ResNet-18 [7], WRE+ 3SBBO [8], and DPA [9]. Guo et al. [7]. proposed an 18-layer CNN model ResNet for image classification. Wu [8]. proposed a method based on a feedforward neural network and combining wavelet Renyi entropy and a proposed three-segment biogeography-based optimization (3SBBO) algorithm to detect COVID-19. Ni et al. [9]. used a deep learning method to accurately identify and quantitatively evaluate chest CT image features of patients with COVID-19. The comparison results based on 7 indicators are shown in Table 7, and the difference in the comparison results is clearly shown in Fig. 13. Although ResNet-18 [7] performs a little better on indicators E<sub>2</sub> and E<sub>3</sub> than our method, the results of our proposed method are better than it under the other five indicators. The result of DPA [9] under indicator E<sub>1</sub> is superior than our proposed method, but the performance under the other six indicators is not as good as our proposed method. Therefore, our method performs best in a combination of 7 indicators.

**Table 7:** Comparison of state-of-the-art approaches

Method	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>
ResNet-18 [7]	78.88±2.57	89.28±0.90	88.05±0.79	84.08±1.14	83.19±1.44	68.55±2.10	83.32±1.36
WRE+ 3SBBO [8]	85.94±1.68	84.75±2.42	84.96±2.16	85.34±1.81	85.44±1.74	70.71±3.61	85.44±1.73
DPA [9]	95.31±1.13	77.19±2.62	80.73±1.70	86.25±1.02	87.40±0.79	73.76±1.73	87.71±0.73
6L-CNN(ours)	89.47±1.50	87.47±2.11	87.75±1.76	88.47±1.05	88.59±0.99	76.98±2.09	88.60±0.99

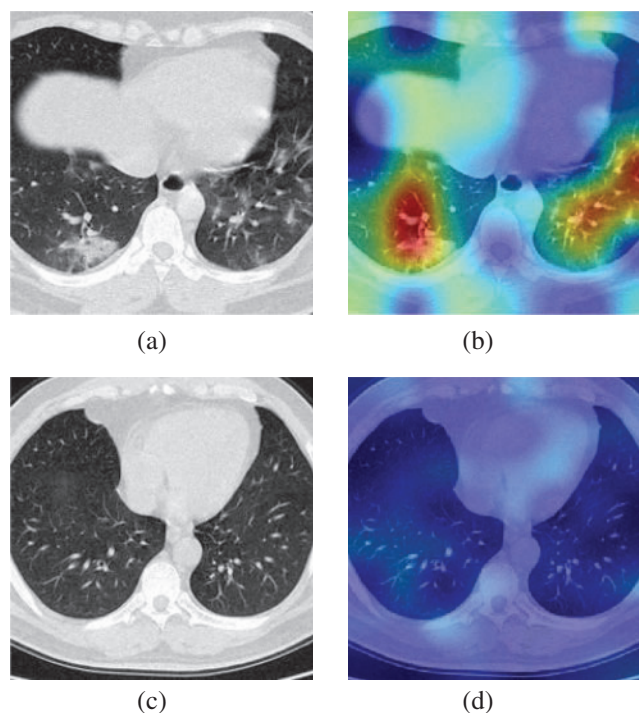


**Figure 13:** Comparison of state-of-the-art approaches

#### 4.8 Heatmap

Fig. 14 shows the heatmap effect produced by using Grad-CAM technology to manipulate the image. Images b and d are heatmaps of the lung CT images of COVID-19 patients and healthy people, respectively. In Fig. 14b, the lung lesion area with COVID-19 is marked in red, while the healthy lung in Fig. 14d is not marked. It is found that the heatmap can clearly and accurately

visualize the model detection area, which is beneficial to the guarantee of the model training process.



**Figure 14:** Heatmap of lungs of patients with COVID-19 and healthy people (a) COVID, (b) Heatmap of (a), (c) Healthy, (d) Heatmap of (c)

## 5 Conclusions

In this paper, we proposed a 6-layer CNN for the detection of COVID-19 and combined the Mp, batch normalization and Adam optimization algorithms. The effect of our proposed method is better than other state-of-the-art methods. The accuracy ( $E_4$ ) of our method reached 89.52%. Grad-CAM technology makes our models to be displayed more intuitively.

However, there is also a flaw in our research that the dataset is not very large, which will have little impact on the effect of model training. So, in future research, we will collect more data to ensure the adequacy of our proposed method in the training process. At the same time, we will build a more superior model based on DL methods to improve the result of COVID-19 detection. We will also share our methods so that other researchers can conduct research on our basis and accelerate the research speed of COVID-19 detection.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Hotez, P. J., Fenwick, A., Molyneux, D. (2021). The new COVID-19 poor and the neglected tropical diseases resurgence. *Infectious Diseases of Poverty*, 10(1), 3. DOI 10.1186/s40249-020-00784-2.
2. Patricio Silva, A. L., Prata, J. C., Walker, T. R., Duarte, A. C., Ouyang, W. et al. (2021). Increased plastic pollution due to COVID-19 pandemic: Challenges and recommendations. *Chemical Engineering Journal*, 405, 126683. DOI 10.1016/j.cej.2020.126683.
3. Yildirim, M., Guler, A. (2021). Positivity explains how COVID-19 perceived risk increases death distress and reduces happiness. *Personality and Individual Differences*, 168, 110347. DOI 10.1016/j.paid.2020.110347.
4. Giri, B., Pandey, S., Shrestha, R., Pokharel, K., Ligler, F. S. et al. (2020). Review of analytical performance of COVID-19 detection methods. *Analytical and Bioanalytical Chemistry*, 413(1), 35–48. DOI 10.1007/s00216-020-02889-x.
5. Salehi, M., Farhadi, S., Moieni, A., Safaie, N., Hesami, M. (2021). A hybrid model based on general regression neural network and fruit fly optimization algorithm for forecasting and optimizing paclitaxel biosynthesis in *Corylus avellana* cell culture. *Plant Methods*, 17(1), 13. DOI 10.1186/s13007-021-00714-9.
6. Reddy, B. S. N., Pramada, S. K., Roshni, T. (2021). Monthly surface runoff prediction using artificial intelligence: A study from a tropical climate river basin. *Journal of Earth System Science*, 130(1), 15. DOI 10.1007/s12040-020-01508-8.
7. Guo, M., Du, Y. (2019). Classification of thyroid ultrasound standard plane images using ResNet-18 networks. *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification*, pp. 324–328. IEEE.
8. Wu, X. (2020). Diagnosis of COVID-19 by wavelet renyi entropy and three-segment biogeography-based optimization. *International Journal of Computational Intelligence Systems*, 13(1), 1332–1344. DOI 10.2991/ijcis.d.200828.001.
9. Ni, Q., Sun, Z. Y., Qi, L., Chen, W., Yang, Y. et al. (2020). A deep learning approach to characterize 2019 coronavirus disease (COVID-19) pneumonia in chest CT images. *European Radiology*, 30(12), 6517–6527. DOI 10.1007/s00330-020-07044-9.
10. Wang, S.-H. (2021). COVID-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. *Information Fusion*, 67, 208–229. DOI 10.1016/j.inffus.2020.10.004.
11. Wang, S. H. (2021). COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis. *Information Fusion*, 68, 131–148. DOI 10.1016/j.inffus.2020.11.005.
12. Alcaraz, J. C., Moghaddamnia, S., Peissig, J. (2021). Efficiency of deep neural networks for joint angle modeling in digital gait assessment. *Eurasip Journal on Advances in Signal Processing*, 2021(1), 20. DOI 10.1186/s13634-020-00715-1.
13. Cabitza, F., Campagner, A., Sconfienza, L. M. (2021). Studying human-AI collaboration protocols: The case of the Kasparov's law in radiological double reading. *Health Information Science and Systems*, 9(1), 20. DOI 10.1007/s13755-021-00138-8.
14. Santana, M. V. S., Silva, F. P. (2021). De novo design and bioactivity prediction of SARS-CoV-2 main protease inhibitors using recurrent neural network-based transfer learning. *BMC Chemistry*, 15(1), 20. DOI 10.1186/s13065-021-00737-2.
15. Zhang, Y. D., Satapathy, S. C., Zhu, L. Y., Górriz, J. M., Wang, S. H. (2020). A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling. *IEEE Sensors Journal*. DOI 10.1109/JSEN.2020.3025855.
16. Panckow, R. P., McHardy, C., Rudolph, A., Muthig, M., Kostova, J. et al. (2021). Characterization of fast-growing foams in bottling processes by endoscopic imaging and convolutional neural networks. *Journal of Food Engineering*, 289, 12. DOI 10.1016/j.jfoodeng.2020.110151.
17. Zhang, Y. D., Dong, Z. C. (2020). Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation. *Information Fusion*, 64, 149–187. DOI 10.1016/j.inffus.2020.07.006.

18. Wang, S. H., Zhang, Y. D. (2020). Densenet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(2s), 1–19. DOI 10.1145/3341095.
19. Zunair, H., Ben Hamza, A. (2021). Synthesis of COVID-19 chest X-rays using unpaired image-to-image translation. *Social Network Analysis and Mining*, 11(1), 12. DOI 10.1007/s13278-021-00731-5.
20. Park, E., Moon, Y. J., Shin, S., Yi, K., Lim, D. et al. (2018). Application of the deep convolutional neural network to the forecast of solar flare occurrence using full-disk solar magnetograms. *The Astrophysical Journal*, 869(2), 91. DOI 10.3847/1538-4357/aaed40.
21. Sangaiyah, A. K. (2020). Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. *Neural Computing and Applications*, 32, 665–680. DOI 10.1007/s00521-018-3924-0.
22. Garbin, C., Zhu, X. Q., Marques, O. (2020). Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79(19–20), 12777–12815. DOI 10.1007/s11042-019-08453-9.
23. Tiwari, S. (2021). Dermatoscopy using multi-layer perceptron, convolution neural network, and capsule network to differentiate malignant melanoma from benign nevus. *International Journal of Cooperative Information Systems*, 16(3), 58–73. DOI 10.4018/IJHISI.20210701.oa4.
24. Zhang, Y. D., Satapathy, S. C., Liu, S., Li, G. R. (2021). A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. *Machine Vision and Applications*, 32(1), 1–13.
25. Assiri, A. S. (2021). Efficient training of multi-layer neural networks to achieve faster validation. *Computer Systems Science and Engineering*, 36(3), 435–450. DOI 10.32604/csse.2021.014894.
26. Wang, S. H., Xie, S., Chen, X., Guttery, D. S., Tang, C. et al. (2019). Alcoholism identification based on an AlexNet transfer learning model. *Frontiers in Psychiatry*, 10, 205. DOI 10.3389/fpsy.2019.00205.
27. Urbaniak, I., Wolter, M. (2021). Quality assessment of compressed and resized medical images based on pattern recognition using a convolutional neural network. *Communications in Nonlinear Science and Numerical Simulation*, 95, 13. DOI 10.1016/j.cnsns.2020.105582.
28. Huang, N., He, J., Zhu, N., Xuan, X., Liu, G. et al. (2018). Identification of the source camera of images based on convolutional neural network. *Digital Investigation*, 26, 72–80. DOI 10.1016/j.diin.2018.08.001.
29. Loey, M., Manogaran, G., Taha, M. H. N., Khalifa, N. E. M. (2021). Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustainable Cities and Society*, 65, 8. DOI 10.1016/j.scs.2020.102600.
30. Benbahria, Z., Sebari, I., Hajji, H., Smiej, M. F. (2021). Intelligent mapping of irrigated areas from landsat 8 images using transfer learning. *International Journal of Engineering Geosciences*, 6(1), 41–51. DOI 10.26833/ijeg.681312.
31. Mbah, T. J., Ye, H., Zhang, J., Long, M. (2021). Using LSTM and ARIMA to simulate and predict limestone price variations. *Mining, Metallurgy & Exploration*, 38(2), 913–926. DOI 10.1007/s42461-020-00362-y.
32. Mutlu, A. Y., Yucel, O. (2018). An artificial intelligence based approach to predicting syngas composition for downdraft biomass gasification. *Energy*, 165, 895–901. DOI 10.1016/j.energy.2018.09.131.
33. Nayak, D. R. (2017). Detection of unilateral hearing loss by stationary wavelet entropy. *CNS & Neurological Disorders–Drug Targets*, 16(2), 15–24. DOI 10.2174/1871527315666161026115046.
34. Jena, R., Pradhan, B., Alamri, A. M. (2020). Susceptibility to seismic amplification and earthquake probability estimation using recurrent neural network (RNN) model in Odisha, India. *Applied Sciences*, 10(15), 5355. DOI 10.3390/app10155355.
35. Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P. et al. (2020). A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-scan images. *Chaos, Solitons & Fractals*, 140, 110190. DOI 10.1016/j.chaos.2020.110190.