Applied Intelligence (2021) 51:2805–2817
https://doi.org/10.1007/s10489-020-02002-w

Stacked-autoencoder-based model for COVID-19 diagnosis on CT images

Daqiu Li [1,2] · Zhangjie Fu [1,2] · Jun Xu [3]

Received: 2 June 2020 / Revised: 28 September 2020 / Accepted: 1 October 2020 / Published online: 9 November 2020

Abstract
With the outbreak of COVID-19, medical imaging such as computed tomography (CT) based diagnosis is proved to be an effective way to fight against the rapid spread of the virus. Therefore, it is important to study computerized models for infectious detection based on CT imaging. New deep learning-based approaches are developed for CT assisted diagnosis of COVID-19. However, most of the current studies are based on a small size dataset of COVID-19 CT images as there are less publicly available datasets for patient privacy reasons. As a result, the performance of deep learning-based detection models needs to be improved based on a small size dataset. In this paper, a stacked autoencoder detector model is proposed to greatly improve the performance of the detection models such as precision rate and recall rate. Firstly, four autoencoders are constructed as the first four layers of the whole stacked autoencoder detector model being developed to extract better features of CT images. Secondly, the four autoencoders are cascaded together and connected to the dense layer and the softmax classifier to constitute the model. Finally, a new classification loss function is constructed by superimposing reconstruction loss to enhance the detection accuracy of the model. The experiment results show that our model is performed well on a small size COVID-2019 CT image dataset. Our model achieves the average accuracy, precision, recall, and F1-score rate of 94.7%, 96.54%, 94.1%, and 94.8%, respectively. The results reflect the ability of our model in discriminating COVID-19 images which might help radiologists in the diagnosis of suspected COVID-19 patients.

Keywords COVID-19 diagnosis · Computed tomography · Deep learning · Stacked autoencoder

1 Introduction

Coronavirus severe acute respiratory syndrome (SARS)-CoV-2 broke out in December 2019. All patients infected with COVID-19 virus developed symptoms of mild or severe respiratory disease COVID-19 [1, 2]. In the following months, COVID-19 spreads rapidly around the world. On March 11, 2020, the World Health Organization declared COVID-19 disease to be a global pandemic [3, 4]. The inefficiency of the global detection of the disease is one of the reasons for its rapid spread [5]. Since the isolation and genome sequencing of COVID-19 virus [6, 7], the current diagnostic methods for detection of COVID-19 virus include nucleic acid detection kit method (TR-PCR method) and COVID-19 nucleic acid sequencing method. However, TR-PCR method requires at least 4 hours to obtain the test results. And nucleic acid sequencing method takes much longer [8, 9]. Moreover, for some countries and regions with insufficient funds, the reagent and equipment needed for these diagnostic methods will be relatively tight, thus delaying the rapid diagnosis of infected people, which led to the rapid spread of COVID-19 in the world [10].

Chest CT images play a significant role in the auxiliary diagnosis of COVID-19. The COVID-19 chest CT assisted diagnosis method based on deep learning might take a few seconds to obtain accurate test results [11–13]. Currently, many researchers have proposed the chest CT diagnostic model of COVID-19 [14–17], such as patch-based deep neural network architecture [14], and dual-sampling attention network [17]. However, a little chest CT images are available obtained because of patient privacy. Therefore, most of these diagnostic models are trained on a small chest CT dataset of

✉ Zhangjie Fu
wwwfzj@126.com

[1] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

[2] Peng Cheng Laboratory, Shenzhen 518000, China

[3] School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044, China

COVID-19 patients. However, these detection models based on deep learning have large variances and are prone to gradient disappearance and overfitting. The performance of these detection models still has great room for improvement. To solve the problem of gradient disappearance and overfitting, a stacked autoencoder detector model is proposed to improve the performance of COVID-19 diagnostic in this paper. The stacked autoencoder detector model is trained on the currently available small chest CT dataset of COVID-19 patients. And comparing the performance of our model with the baseline model, we achieved an average 10% improvement in the accuracy of our model. In this paper, our main contributions include:

- A new stacked-autoencoder-based model was proposed for COVID-19 diagnosis that can overcome the gradient disappearance and overfitting caused by deep neural network training on a small dataset to some extent.
- A new reconstruction loss was constructed as a regular term, which can improve the detection accuracy.
- The average performance of our model outperforms the current binary baseline COVID-19 diagnostic model based on the same small chest CT dataset.

The remainder of this paper is organized as follows: the previous work is presented in Section 2, the methodologies including the proposed model, datasets, and training strategy are described in Section 3, and the experimental design and results are presented and analyzed in Section 4. Section 5 and Section 6 discusses and summarizes this paper.

2 Previous works

In recent years, deep learning technology has achieved promising results in the automatic analysis of multimodal medical images to complete radiological tasks [18–20]. Deep convolutional neural networks are a powerful deep learning architecture, which has been widely applied to image classification, pattern recognition, and other fields [21]. In previous studies, deep convolutional neural networks have been exploited to classify chest CT images and successfully diagnosed common chest diseases such as Tuberculosis screening [22] and mediastinal lymph nodes in CT images [23]. During the current COVID-19 outbreak, researchers are trying to make their efforts to alleviate the epidemic through their research [14–17]. Based on the previous studies, the application of deep convolutional neural networks in the COVID-19 auxiliary diagnosis of chest CT is a worthy research direction. Many researchers are working in this direction. Generally, there are two different kinds of deep learning diagnostic models for COVID-19.

One kind of COVID-19 diagnostic methods is binary classification diagnostic model, including DenseNet [24], DRE-Net [25], M-Inception [26], etc. In [24], a publicly available COVID-CT dataset was built, in which there were 275 chest CT scans of COVID-19 positive patients. A deep convolutional neural network was trained on this dataset and the model achieved an accuracy rate of 84.7%. In [25], a deep learning model was built for pneumonia (COVID-19) classification. By tuning hyperparameters according to the validation set, the model achieved an accuracy rate of 86%. In [26], an inception migration neural network was constructed, and which achieved 82.9% accuracy finally.

The other is multi-classification diagnostic model, including location-attention oriented model [27], CoroNet [28], COV-Net [29], etc. In [27], it used Res-Net to extract features from CT images together with a location-attention mechanism model, could accurately distinguish COVID-19 cases from Influenza-A viral pneumonia cases and health cases, with an overall accuracy rate of 86.7%. In [28], a deep convolutional neural network was trained on the dataset which included COVID-19 positive chest CT images, pneumonia bacterial, pneumonia viral, and normal CT images. And the model achieved an overall accuracy of 89.5%. In [29], a deep learning-based CT diagnostic system was developed to identify COVID-19 patients based on the collected CT datasets. The experimental results showed that the accuracy of the model was 87%.

These chest CT assisted diagnostic models using deep learning for COVID-19 are mostly based on a limited number of COVID-19 CT datasets. The performance indicators of these models, such as accuracy and recall rate have not reached the requirements for the actual detection of COVID-19. And the application of deep learning techniques to identify and detect novel COVID-19 in chest CT is still quite limited so far. Therefore, this paper aims to propose a new framework of deep learning classifiers to assist radiologists to automatically diagnose COVID-19 in chest CT images.

3 Methodology

3.1 Overall architecture of the proposed model

Generally, the better experimental results are obtained by deep network fitting training based on large datasets. Deep learning modeling is usually to establish a deeper network structure, and the deeper the neural network is in theory, the higher the fitting degree of the model. However, since the traditional multi-layer neural network uses the way of loss backpropagation, the smaller the loss of propagation as the deeper the network layer, which results in the problem of gradient disappearance [30]. For the training of deep neural networks on a small dataset, the problem of gradient disappearance is more and more serious. Solving the gradient disappearance problem in the training process of a small CT image dataset is of great importance. Kaiming He put forward by using deep residual network structure to improve the gradient disappeared. By adding a shortcut connection structure in the deep neural network, the gradient can be transferred from the first layer to the last layer of the network, which can alleviate the problem of gradient getting smaller and smaller in the deep neural network to some extent. Different from this approach, stack autoencoder improves gradient disappearance from the perspective of improved training way. In general, stack autoencoder uses a separate encoding and decoding network for all convolution layers in the network for separate training. In single layer encoding and decoding networks, gradient disappearance is generally not easy to occur, thus avoiding the problem of gradient disappearance that may occur during the training of the entire deep network. This is also the core idea of a stacked autoencoder neural network [31, 32]. A stacked autoencoder neural network is a modeling method to use an autoencoder neural network. The overall architecture of the stacked autoencoder detector model is shown in Fig. 1.

In Fig. 1a, a 3 by 3 convolution layer is used to extract the feature layer by layer, in which Local Response Normalization(LRN) and Max-Pooling are conducted before the convolution operation of feature **h2**、 **h3** and **h4** (feature **h2**、 **h3** and **h4** are the encoding outputs of Autoencoder 2, 3 and 4). LRN is proposed in Alex-Net by Hinton [33]. LRN layer simulates the lateral inhibition mechanism of the biological nervous system and creates a competitive mechanism for the activity of local neurons, making the relatively large value of response relatively larger and improving the generalization ability of the model. In Fig. 1b, Chest CT images are used directly for encoding and decoding. The layer 1 network includes the encoder part for feature extraction and decoder part for restoring the original feature map of the input. Different from Fig. 1b, the input feature map of Fig. 1 is firstly operated by LRN and Max-Pooling. Figure 1c shows the network of the Autoencoder 2 of the whole stacked autoencoder diagnosis model.

The network of layer 3 and layer 4 is the same as that of the layer 2. It is important to note that the loss functions of the first four layers of autoencoder networks are different. The loss of the layer 1 autoencoder network is the reconstruction loss, as shown in Eq. (1):

$$\mathrm{J}_1 = \frac{1}{N}\sum_{i=1}^{N} L\left(f_{decoding1}(\boldsymbol{w},\boldsymbol{x}),\boldsymbol{x}\right) \tag{1}$$

Where N denotes the size of batch size, $\boldsymbol{w}$ is the parameter matrix of layer 1,$\boldsymbol{x}$ is the input CT image, and L is loss function Mean-Squared-Error (MSE), as shown in Eq. (2):

$$L_{MSE}\left(f_{decoding}(\boldsymbol{w},\boldsymbol{x}),\boldsymbol{x}\right) = \left(f_{decoding}(\boldsymbol{w},\boldsymbol{x}) - \boldsymbol{x}\right)^2 \tag{2}$$

The loss function term selected for the first fourth layers of the model is the mean squared error, aiming to approximate the real data as much as possible [34]. The value of the loss function becomes smaller and smaller after iteration training until it is reduced to the minimum. The layer 2 loss function is based on the layer 2 itself reconstruction loss plus the layer 1 autoencoder network loss function J_1 as the regularization term, as shown in Eq. (3):

$$\mathrm{J}_2 = \mathrm{J}_1 + \frac{1}{N}\sum_{i=1}^{N} L\left(f_{decoding2}\left(\boldsymbol{w}',\boldsymbol{h1}\right),\boldsymbol{h1}\right) \tag{3}$$

Where J_1 denotes the layer 1 autoencoder network loss function as the regularization item, $\boldsymbol{w}'$ is the parameter matrix of layer 2, and $\boldsymbol{h1}$ is the output of layer 1 $f_{encoding}$, as shown in Eq. (4):

$$\mathbf{h1} = \mathbf{f}_{\mathbf{encoding}}(\boldsymbol{w},\boldsymbol{x}) \tag{4}$$

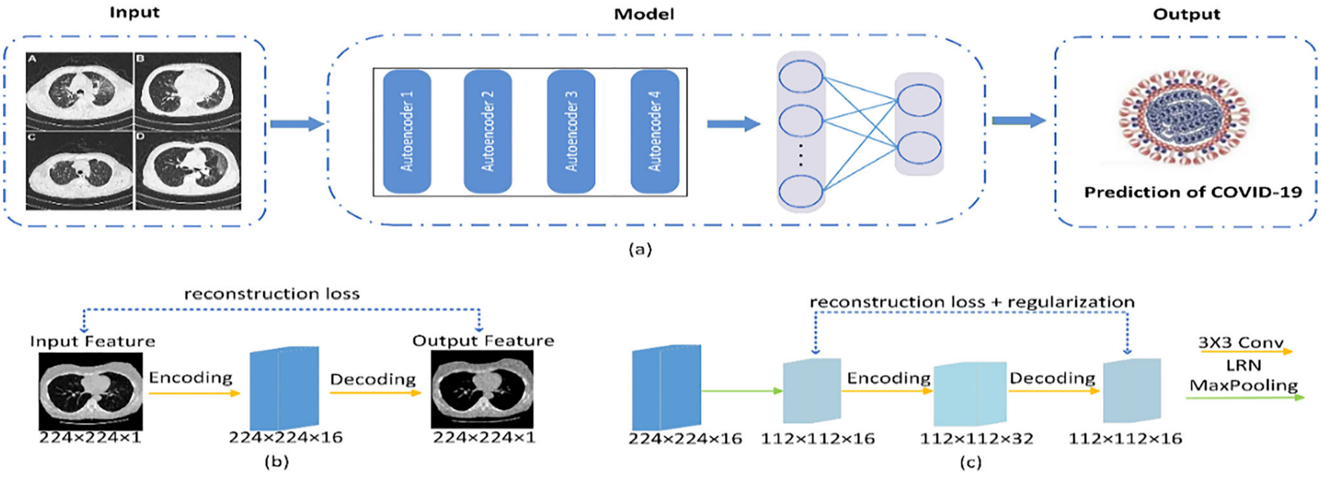


Fig. 1 Stacked autoencoder model structure, **a** the overall architecture of the stacked autoencoder detection model, **b** stacked autoencoder layer 1 structure, **c** stacked autoencoder layer 2 structure

The loss function of layer 3 and 4 are similar to those of the layer 2. The loss function of layer 3 is the reconstruction loss of layer 3 plus the loss function of the first two layers of the autoencoder network as shown in Eq. (5). The loss function of layer 4 is the reconstruction loss of layer 4 plus the loss function of the first three layers of the autoencoder network as shown in Eq. (6).

$$\mathrm{J}_3 = \mathrm{J}_2 + \frac{1}{N}\sum_{i=1}^{N} L\left(f_{deconding3}, \boldsymbol{h2}\right) \tag{5}$$

$$\mathrm{J}_4 = \mathrm{J}_3 + \frac{1}{N}\sum_{i=1}^{N} L\left(f_{deconding4}, \boldsymbol{h3}\right) \tag{6}$$

Where $\boldsymbol{h2}$ and $\boldsymbol{h3}$ are the output of layer 2 $f_{encoding}$ and the output of layer 3 $f_{encoding}$. And when calculating the classification loss, we add the loss function of each of the four layers autoencoder networks as the regular term, as shown in Eq. (5):

$$\mathrm{J}_{classification} = \sum_{i=1}^{4} J_i + \frac{1}{N} \times \sum_{i=1}^{N} L'\left(f_{classification}\left(\boldsymbol{w}'', \boldsymbol{h5}\right), \boldsymbol{y}\right) \tag{7}$$

Where $\boldsymbol{w}''$ denotes the parameter matrix of the last layer, $\boldsymbol{y}$ is the label of CT image, $\boldsymbol{h5}$ is the output of layer 5, and L' is loss function term $L_{cross-entropy}$, as shown in Eq. (6):

$$L_{cross-entropy}\left(f_{classification}\left(\boldsymbol{w}'', \boldsymbol{h5}\right), \boldsymbol{y}\right) = -\boldsymbol{y}\log\left(f_{classification}\left(\boldsymbol{w}'', \boldsymbol{h5}\right)\right) \tag{8}$$

The loss function term selected for the last layer of the model is the cross-entropy loss, aiming to obtain the probability values of COVID-19 and non-COVID-19.

The optimization function acts on the loss function in the process of backpropagation, and the optimizer wants to find a minimum loss value, that is, the local optimal solution. We build new global loss functions by continually adding the loss function of the previous layer as regular terms to the loss of the current layer. On this new loss function, the optimizer can find a better local optimal solution. In other words, this new global loss function not only makes the feature extraction effect of each layer better but also improves the effect of the final classification.

3.2 The detection model

In this section, we introduce the modeling approach of the stacked autoencoder detection model. Our model is a neural network composed of autoencoders that train multi-layer networks layer by layer, which trains the convolution kernel of each layer by autoencoder in the order from front to back. The output of the last layer is taken as the input feature of the softmax classifier, and the classification results are output by softmax. Here we present the entire modeling process of our model in three steps.

Firstly, an autoencoder is trained to obtain the input first-order feature **h1** of the original CT scan image data, as shown in Fig. 2. We can use a formula to represent this process:

$$y_{out} = f_{decoding}\left(\boldsymbol{w}', f_{encoding}(\boldsymbol{w}, \boldsymbol{x})\right) \tag{9}$$

joined_input OR train_dataset hidden = 100

Where $\boldsymbol{x}$ represents the input CT image, $\boldsymbol{w}$ is the weight matrix of layer 1 encoder, $\boldsymbol{w}'$ is the weight matrix of the layer 1 decoder. In general, the result of training a convolutional neural network is equivalent to obtaining a complex encoding function $f_{encoding}$. The entire convolutional network is a high-level function with numerous parameters. In this process, there is generally no decoding process. Here we train each layer separately by adding a decoding process to get the corresponding encoding function $f_{encoding}$. The output y_{out} of the decoding function $f_{decoding}$ is required to be as similar as

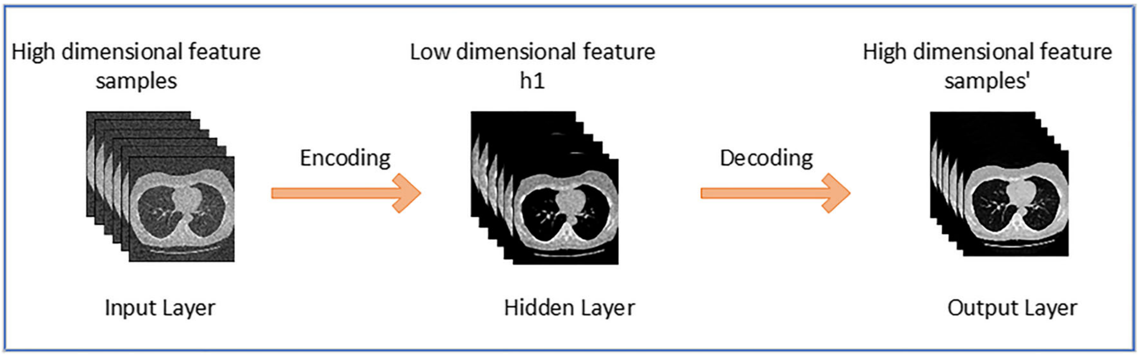

High dimensional feature samples
Low dimensional feature h1
High dimensional feature samples'
Encoding
Decoding
Input Layer
Hidden Layer
Output Layer

Fig. 2 Stacked autoencoder layer 1 structure

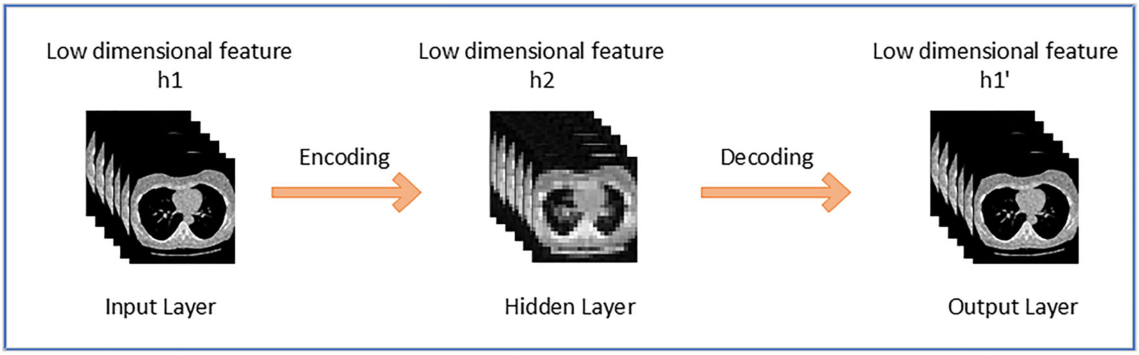


Fig. 3 Stacked autoencoder layer 2 structure

possible to the input. Thus, after each layer is trained separately, the weight **w** learned by the encoding function $f_{encoding}$ has a better ability to extract features than the weight **w** obtained by the traditional training method. And we can get the feature **h1**, which is the output of $f_{encoding}$, as shown in Eq. (4).

h1 ----
layer 1
h2 ----
layer 2
h3 ----
layer 3
h4 ---
layer 4
h5 ----
layer 5

h5_c ---
softmax

Secondly, the output feature **h1** in the previous step is taken as the input to acquire feature **h2** through layer 2 autoencoder, as shown in Fig. 3. In the same way, feature **h2** can be used as the input to obtain feature **h3** through autoencoder on the next layer. Again, we can get feature **h4**. After getting feature **h4**, feature **h4** is used as the input to obtain **h5** through the next dense layer.

Thirdly, we connect the feature **h5** of the previous step to the softmax classifier to get the results of classifying the digital labels of the image, as shown in Fig. 4. In this step, the softmax classifier obtains the probability of two types of labels by combining the feature data calculated from the previous layers. One is Y1, which is the CT scan image of non-COVID-19. The other category is Y2, which represents CT images of COVID-19 positive patients. Finally, these six layers are combined to form a stacked autoencoder detector with four convolution layers、one dense layer, and one softmax layer, which can classify CT scan images.

3.3 Training

3.3.1 Datasets

A baseline chest CT dataset of COVID-19 collected and published by UC San Diego is used in our research [24]. The artificial intelligence method for CT image detection of COVID-19 have the advantages of fast speed, low cost, and high accuracy. However, due to privacy concerns, the CT scans used in these works are not shared with the public. This greatly hinder the research and development of more advanced AI methods for more accurate testing of COVID-19 based on CT. To address this issue, the dataset collectors build a COVID-CT dataset that contain 275 COVID-19 positive chest scan images and 195 COVID-19 negative chest scan images. And the dataset is open-sourced to the public, to foster the R&D of CT-based testing of COVID-19. The published website for this dataset is in the footnote section of this page[1]. The COVID−CT dataset have been uploaded to GitHub by dataset collectors and is being continuously supplemented. The dataset collectors extract publicly available CT images from 760 preprints, which are from medRxiv and bioRxiv, and manually select images with clinical manifestations of COVID−19 by reading the image descriptions. The CT images vary in size, with the minimum, average and maximum heights of 153, 491, and 1853. The minimum, average, and maximum widths are 124, 383, and 1485. These scans are from 143 patient cases. Before training the model using the CT scan images, uniform resizing and standardization are adopted for these data. The final processing size is 224 by 224.

3.3.2 The training strategy

In this section, we describe the training methods of our model from three aspects: the division of training set, training process, and training results. Before training the model, the first step is to partition the dataset. By the hold-out method, the original dataset is divided into three mutually exclusive sets, which are divided into a training set, verification set, and test set. Table 1 describes the partitioning of the dataset. The training set includes 183 COVID-19 positive CT images and 146 COVID-19 negative CT images. The verification set includes 57 COVID-19 positive CT images and 15 COVID-19 negative CT images. The test set includes 35 COVID-19 positive CT images and 34 COVID-19 negative CT images. And the purpose of setting up the dataset is to maintain the consistency of the dataset with the baseline model on the current COVID-2019 binary classification. So that, we can provide more fair comparisons in the experiments

To obtain a reliable and stable model, the 5-fold cross-validation method is used here. Cross-validation is effective in overcoming the overfitting problem. It can make full use of all CT images in the limited dataset for training, and finally

[1] https://github.com/UCSD-AI4H/COVID-CT.

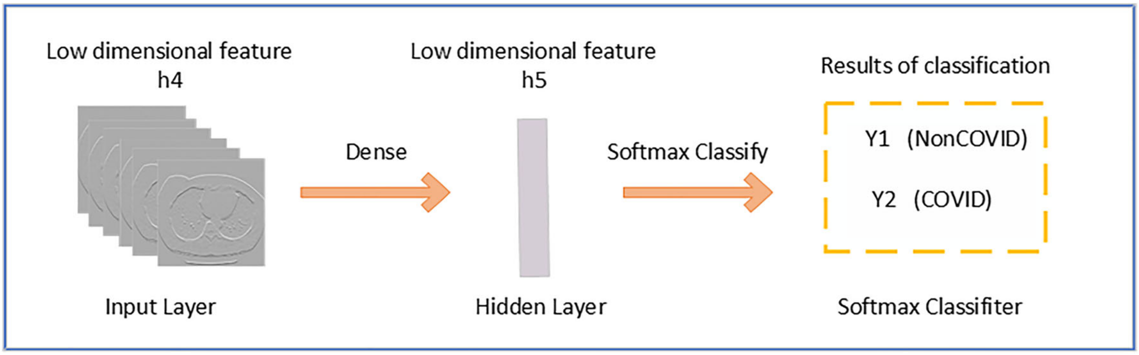


Fig. 4 Stacked autoencoder layer 5 structure

take the average of the results of cross-validation, which makes the evaluation results more convincing. We mix the training set and the test set into a new dataset. Then the new dataset is divided into 5-fold cross-validation, as shown in Fig. 5. The new dataset is bisected into five mutually exclusive subsets, selecting one of them as the test set each time and training the model five times. And the test results of the five test sets are summed and averaged to obtain the comprehensive performance evaluation parameters of the model.

After the dataset is divided, the next step was to train the model with the divided training set. According to the sequence of the network layer, the training starts from layer 1 autoencoder of the stacked autoencoder detector model. To improve the robustness of the model, gaussian noise is added to the CT scan images. Therefore, the layer 1 network can perform certain denoising ability on the input image after the layer 1 autoencoder training is completed. And we save the training parameters of layer 1 autoencoder to provide a good initial weight for the training of layer 2. When the layer 2 networks are trained, the original data are firstly inputted to the layer 1 network to acquire the input of layer 2. Similarly, the layer 3 and 4 are trained separately after the layer 2 network is trained.

When each layer network is trained separately, the optimization function selected is Adam optimizer [35]. The key to the neural network is to calculate each neuron, asfollows:

$$\begin{cases} \boldsymbol{z}^{[l]} = \boldsymbol{w}^{[l]}\boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]} \\ \boldsymbol{a}^{[l]} = g\left(\boldsymbol{z}^{[l]}\right) \end{cases} (l = 1, 2, .., \mathrm{n}) \tag{10}$$

Table 1 Original statistics of data split

Classes	Non- COVID	COVID − 2019	Total
Train	146	183	329
Validation	15	57	73
Test	34	35	69

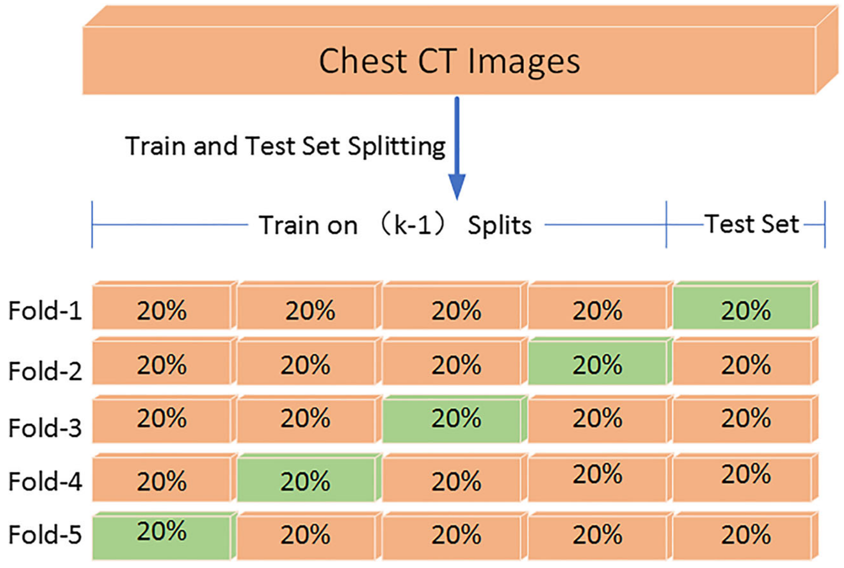
Where l is the number of layers, $\boldsymbol{w}^{[l]}$ is the parameter matrix, $\boldsymbol{b}^{[l]}$ is the deviation, $\boldsymbol{a}^{[l]}$ is the output matrix of each layer, and g is the activation function. The Relu function is used for better normalization with LRN [36]. By this formula, each layer of neurons in the neural network is calculated simultaneously. The predicted value is obtained by a series of calculations on the hidden layer. Then, the stochastic gradient descent method shown in Eq. (11) is applied to the backpropagation, so that the weight $\boldsymbol{w}^{[l]}$ and partial positive $\boldsymbol{b}^{[l]}$ of each layer can be fitted appropriately [37, 38].

$$\begin{cases} \boldsymbol{z}^{[l]} = \boldsymbol{w}^{[l]} - \alpha \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{w}^{[l]}} \\ \boldsymbol{b}^{[l]} = \boldsymbol{b}^{[l]} - \alpha \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{b}^{[l]}} \end{cases} (l = 1, 2, .., \mathrm{n}) \tag{11}$$

To further improve the performance of the model, a cascaded approach is used to further optimize network parameters. Cascading all layers together to create a new network. Here, the output of the first layer is reused, and the input data is also the original CT scan image data. Like the layer 1 of the cascade network structure, the other layers are redefined, but the parameters such as the weights trained in each layer are shared.

After the whole model is trained, we get the end-to-end stack autoencoder detection model. In the third step, we introduce our model hyperparameters and loss in the training process. Table 2 shows the model parameters that had been tested a lot. And we get the best-stacked autoencoder detection model through these parameters. The training loss of the first four layers of the autoencoder network is shown in Fig. 6a-d. As shown in Table 2, we set the epochs to be 500 rounds. During the training, we set the loss value to be saved every 1 round. In Fig. 6a, we can see that the initial loss value of layer 1 is above 1.0, which is a relatively high loss. Because the initial weight and other parameters of the autoencoder network in layer 1 are randomly initialized [39], and the CT image sent into the network for training is the image with added noise. The loss reduces to about 0.5 after 30 epochs, after which the loss is reduced to the minimum and starts to appear slight fluctuations. In Fig. 6b, since the input of the layer 2 autoencoder

Fig. 5 Schematic representation of training and testing schemes employed in the 5-fold cross-validation procedure



network is obtained through the layer 1 encoder network, the initial value of the layer 2 training loss is a small value like 0.55. Then it drops to about 0.45. The descending curves of the training loss in layer 3 and layer 4 are similar to that in layer 2, as shown in Fig. 6c-d.

The decline of classification loss in the last layer seems much more gradual than that in the first four layers, as shown in Fig. 6e. And the training loss after the drop is still larger in Fig. 6b-d. It shows that the overall loss has not fallen much. This is because the loss functions of the layer 2, 3, 4 and last layer each add the previous reconstruction loss. In the process of gradient descent and back-propagation, the encoding function of each layer can further improve the ability of feature extraction.

Table 2 The parameters of the model

Parameters	Values
Hidden layer	5
Neurons	#1 3x3x16
	#2 3x3x32
	#3 3x3x64
	#4 3x3x128
	#5 6272 dense layer before softmax
Learning rate	0.001
Activation function	#1#2#3#4#5 Relu
	#6 Softmax
Loss function	#1#2#3#4 Mean squared loss
	#6 Sparse softmax cross entropy
Optimizer	Adam
Epochs	500
Batch size	128

In Fig. 6f, we can preliminarily see the effect of the new loss function constructed by the superposition reconstruction loss. Figure 6f is the training loss obtained from unified training by connecting all levels. This cascading network shares the parameters such as the weight of the trained decoding function, so it can be seen from the Fig. 6f that the initial training loss is small, and its value is below 0.4, and then down to about 0.3. And the training accuracy has been above 0.9.

4 Experimental design and results

4.1 Evaluation metrics

In the experiment, the following performance metrics are used to measure the performance of the stack autoencoder detection model. TP represents the true positive. TN represents the true negative. FP represents the false positive. FN represents the false negatives. The matrix consisting of the parameters of the four-test metrics is the confusion matrix. The values of each performance index can be calculated from the confusion matrix [40], where:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{12}$$

Accuracy reflects the judgment ability of the detection model to the whole test set, it can judge the positive as positive and the negative as negative.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{13}$$

Recall refers to the proportion of the predicted positive cases in the total positive cases.

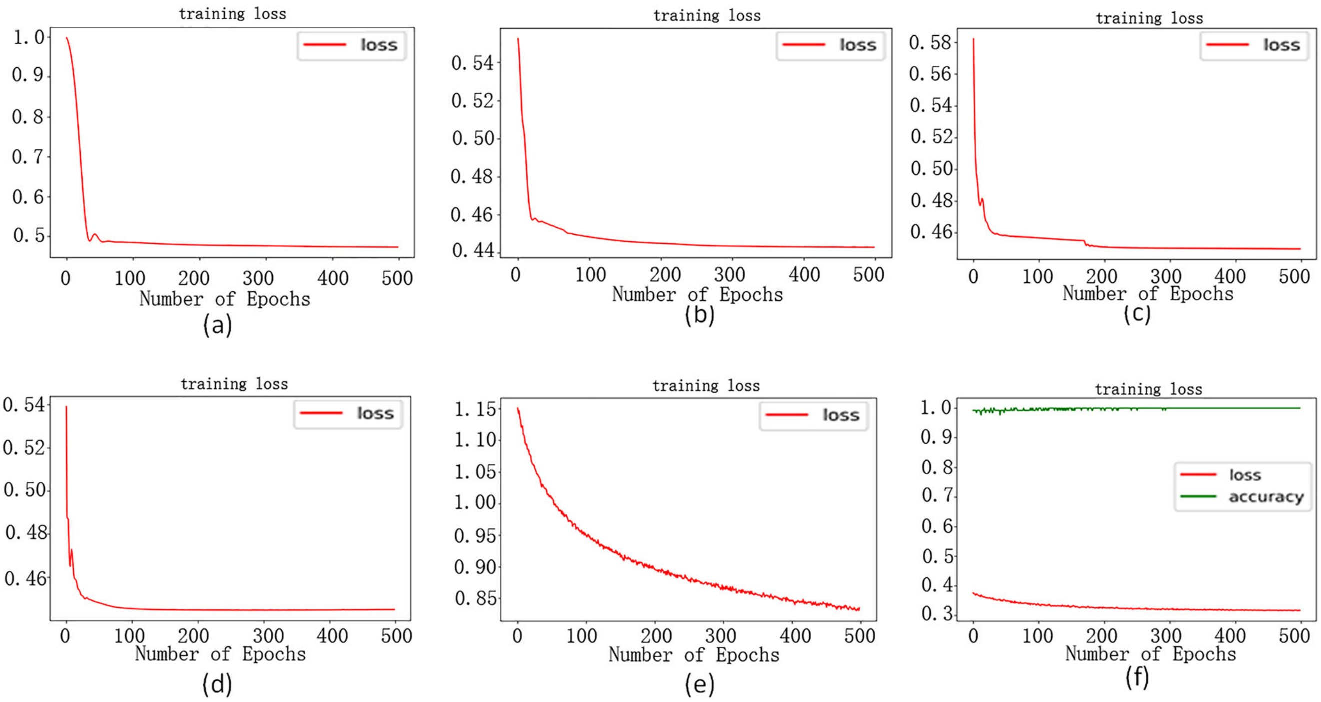


Fig. 6 Training loss of stacked autoencoder model: **a** is the training loss of autoencoder layer 1, **b** is the training loss of autoencoder layer 2, **c** is the training loss of autoencoder layer 3, **d** is the training loss of autoencoder layer 4, **e** is the training loss of the last classification layer, **f** is the training loss and accuracy of cascaded stacked autoencoder model

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{14}$$

Precision refers to the proportion of the real positive example in the positive example judged by the detection model.

$$F1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{15}$$

F1 refers to the harmonic mean of precision rate and recall rate and represents the discriminant ability of the model for each category.

4.2 Experimental results

In this section, we report the experimental results of our model on the test set and compare them with some existing models. The stacked autoencoder detection model has trained a total of six times. For the first time, we trained and tested our model with the dataset obtained by the original dataset partitioning method in Table 1. The test results of the confusion matrix are shown in Fig. 7a. Figure 7b-f is the test results of the confusion matrix obtained by using the 5-fold cross-validation method shown in Fig. 5. Also, accuracy, precision, F1-score, and recall results for the binary classification task are given in Table 3.

It can be noted from Table 3 that the proposed model has achieved an average accuracy of 94.7% in detecting COVID-19 and the obtained average precision, recall, and F1-score values of 96.54%, 94.1%, and 94.8%, respectively. This result is significantly better than the model performance trained by the dataset divided by the original method. From the results of 5-fold cross-validation, we speculate that this is because the distribution of the dataset divided by the original method deviates from the distribution of the standard COVID-19 CT image data. As a result, the model trained by this dataset can only detect COVID-19 CT images with certain features, so its test results in the test set are worse than the results of 5-fold cross-validation. The training set obtained by the method of Fold-1 and the original division method contain CT images deviating from the standard distribution, so the experimental result of Fold-1 is similar to the experimental result of the original division method. And the other four cross-validation models showed high generalization on the test set.

Table 3 Performance metrics the proposed model including each fold, an average of 5 folds, and the original dataset

Folds	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Fold-1	86.2	100	75.0	85.7
Fold-2	93.7	89.7	100	94.6
Fold-3	97.4	97.6	97.6	97.6
Fold-4	97.5	97.7	97.7	97.7
Fold-5	98.7	97.7	100	98.8
Average	94.7	96.54	94.1	94.8
Original	88.4	100	77.1	87.1

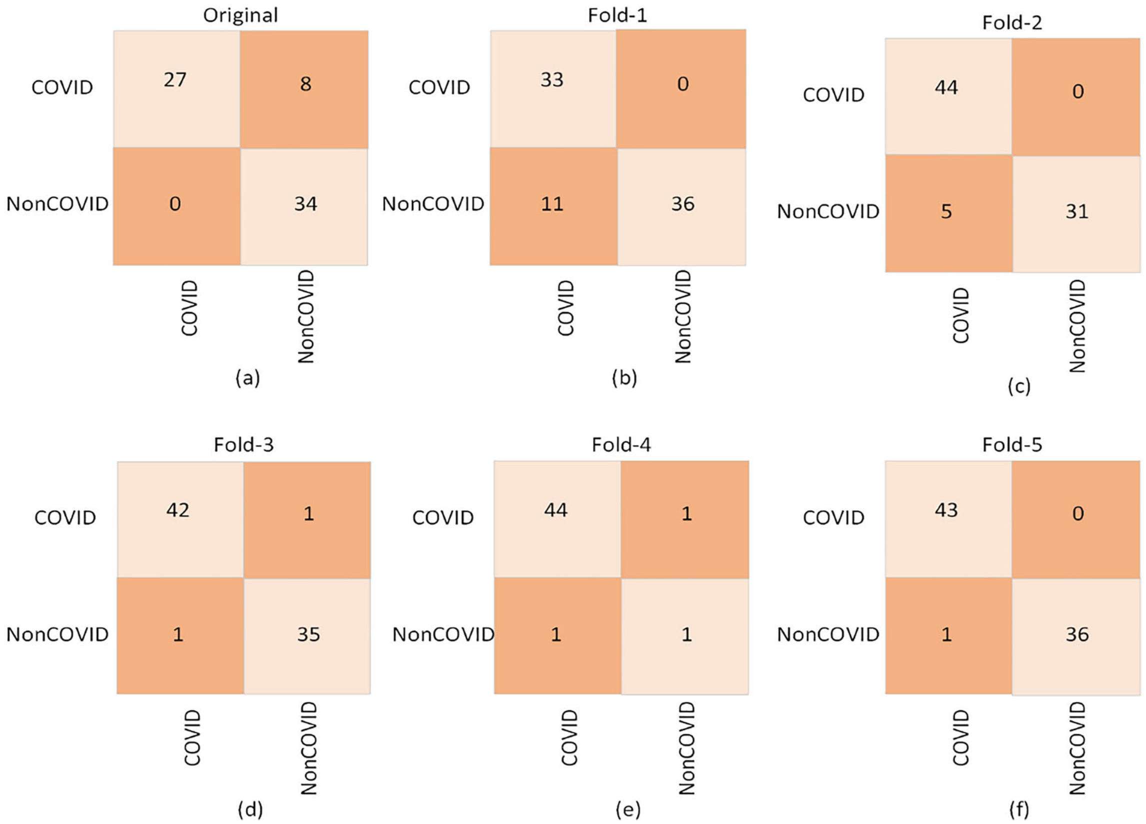


Fig. 7 The original and 5-fold confusion matrix results for the binary classification task: **a** Original confusion matrix, **b** Fold-1 CM, **c** Fold-2 CM, **d** Fold-3 CM, **e** Fold-4 CM, and **f** Fold-5 CM

Meanwhile, in addition to training our model, we also trained the baseline model in [26] on the original partitioned dataset and obtained test results, as shown in the first row of Table 4. Moreover, we build the same convolutional network detection model as our model framework. Similarly, we train the convolutional network detection model on the original partitioned data sets and obtain the test results, as shown in the second row of Table 4. The first three models in Table 4 are all trained and tested with the data sets obtained by the original data partitioning method. The fourth and fifth rows show the performance of the model presented in recent related studies. Both the two models focus on the COVID-19 CT image binary classification task. The performance of all models is shown in Table 4. From the comparison of the performance, we can see that our model archives the best performance. And it is the first time to use a stacked autoencoder to train the COVID-19 detection model on a small CT dataset. Our technical contribution lies in we build a stacked convolutional autoencoder and design a new loss function which adds the reconstruction loss to classification loss for our detection model. The comparison results show that our stacked autoencoder model is indeed effective. This is mainly due to the stacked autoencoder neural network has a strong feature expression ability and the advantages of deep convolutional neural network. It can usually obtain the hierarchical grouping structure feature or the partial-whole structure feature of the input. The stacked autoencoder

Table 4 Performance comparison of different deep learning models

The Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Baseline Model [26]	84.7	97	76.2	85.3
Convolution Model	84.2	90	77.1	83.1
Our Model (Original dataset)	**88.4**	**100**	**77.1**	**87.1**
DRE-Net [27]	86.0	79.0	96.0	87.0
M-Inception [28]	82.9	73.0	88.0	77.0
Our Model (Average of 5 folds)	**94.7**	**96.54**	**94.1**	**94.8**

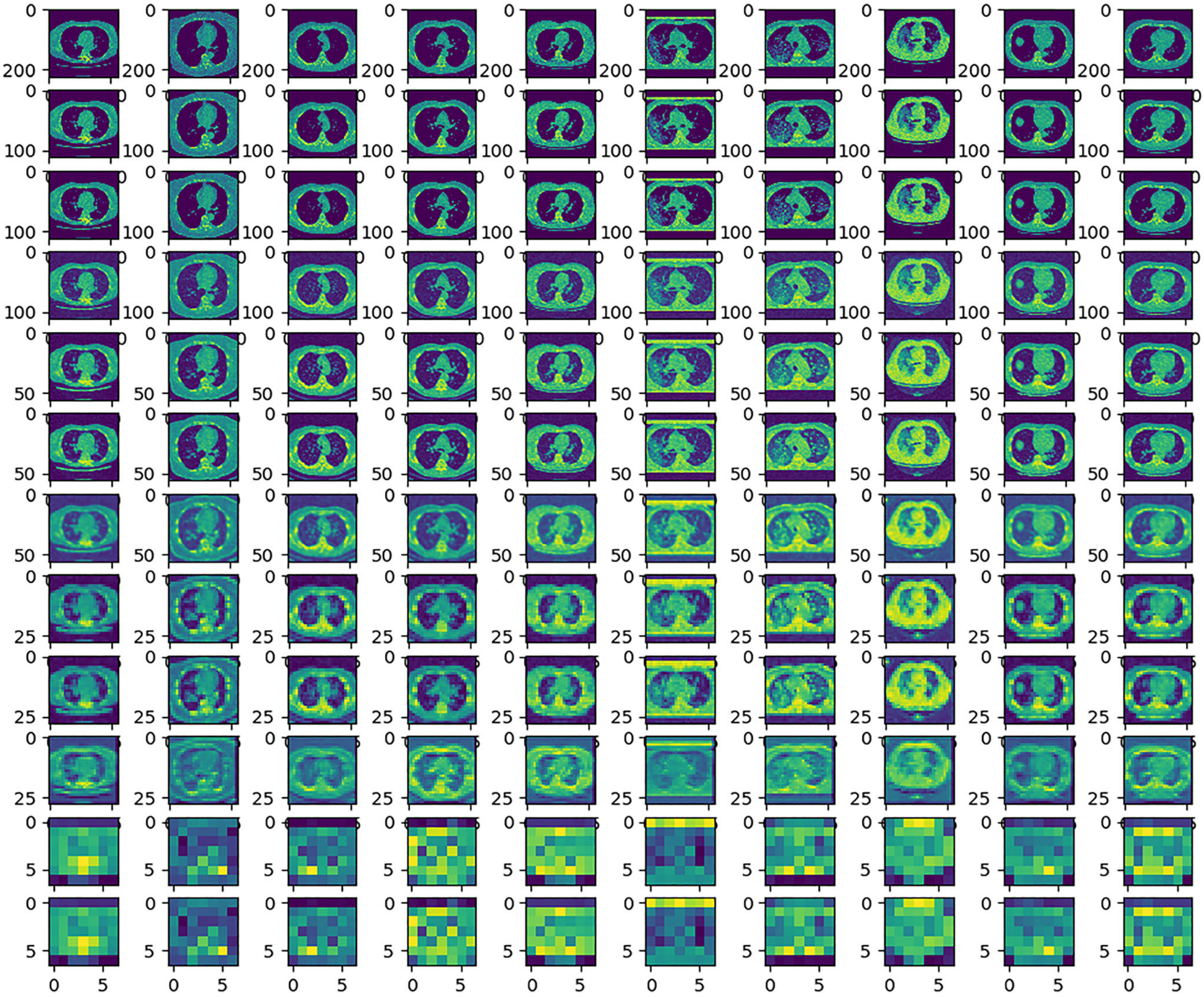


Fig. 8 The process of extracting features in the stacked autoencoder detection model. From top to bottom: twelve different feature extraction operations. From left to right: ten different CT scan images feature maps

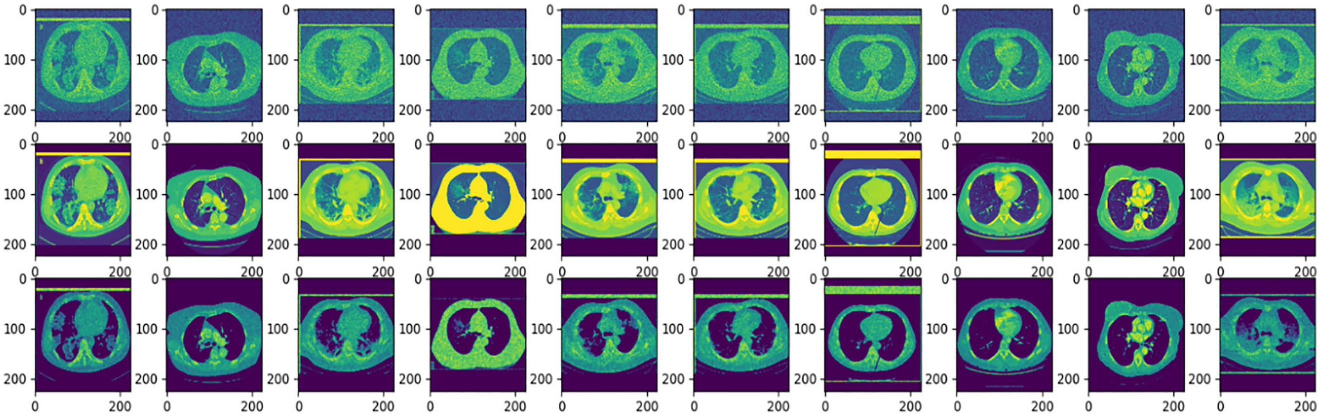


Fig. 9 Test results of layer 1

tends to learn the characteristic vector corresponding to the sample, which can better represent the data characteristics of the high-level sample.

Besides, after 5-fold cross-validation, the average performance metrics of our model are much better than that of the baseline model except for precision. The average precision of our model is not higher than that of the baseline model. Through further analysis and verification, we find that is because many samples in the training set divided by Fold-2 have special characteristics, which interferes with the discriminant results of the model. We can also see that from the Fold-2 experiment in Table 3. In the experimental results of Fold-2, the precision of the model is only 89.7%, which is quite different from the precision of the other four crossover experiments. This also proves that our analysis results are reasonable. The inspiration is to try to screen out those special samples that have a great influence on the model when training the model.

5 Discussions

In this section, we discuss the reasons why our model can achieve better detection performance. We find that this is mainly because the stacked autoencoder detector model has the following advantages:

Firstly, each layer of the stacked autoencoder detector model can be trained separately, which ensures the controllability of the dimensionality reduction of the CT scan image features. The training results of each convolution layer can be obtained. Figure 8 shows the process of extracting features from each layer convolution in the stacked autoencoder detection model. In Fig. 8, there are a total of ten columns, and each column displays different CT images feature maps extracted by each layer in the stacked autoencoder model. Besides, the figure has 12 rows, and each row shows feature maps obtained by a feature extraction operation. The 12 operations were divided into four blocks, and the operations of each block are 3 by 3 convolution, Max-Pooling, and LRN. The first, fourth, seventh, and tenth rows are the feature maps obtained after the convolution operation. These four convolution layers have been trained by the autoencoder. After convolution, the maximum pooling operation and LRN normalization are performed to reduce the dimensionality of the feature maps and improve the response value of useful features.

From the last raw feature maps of Fig. 8, we can see that our model can extract sample features useful for binary classification from the original CT input image after four-layer autoencoder training alone. But it is worth noting that if the original input image has some special features that are not related to classification, it will have an impact on the final classification result. For example, in the extraction process of the sixth column of the feature maps in Fig. 8, we found that there are some useless features outside the chest CT range. However, these useless features are characterized in the last row of the feature map, which will interfere with the subsequent classification operations. This was also mentioned when discussing the experimental results earlier.

Secondly, the regularization item added in each layer also plays an important role in improving the detection accuracy of the detection model. The regularization term is helpful for the model to find a better local optimal point when gradient descent is carried out so that the model finally achieves a good convergence effect. Figures 9 and 10 respectively show the testing effect of autoencoder layer 1 and layer 2. In Fig. 9, the first row is the original image after adding noise, and the second row is the original CT image. The third row is the recovered output image by the layer 1 decoding. As can be seen from Fig. 9, the Gaussian noise added in the original CT scan image can be removed, and some high-dimensional features of the original data can be extracted. In Fig. 10, the first

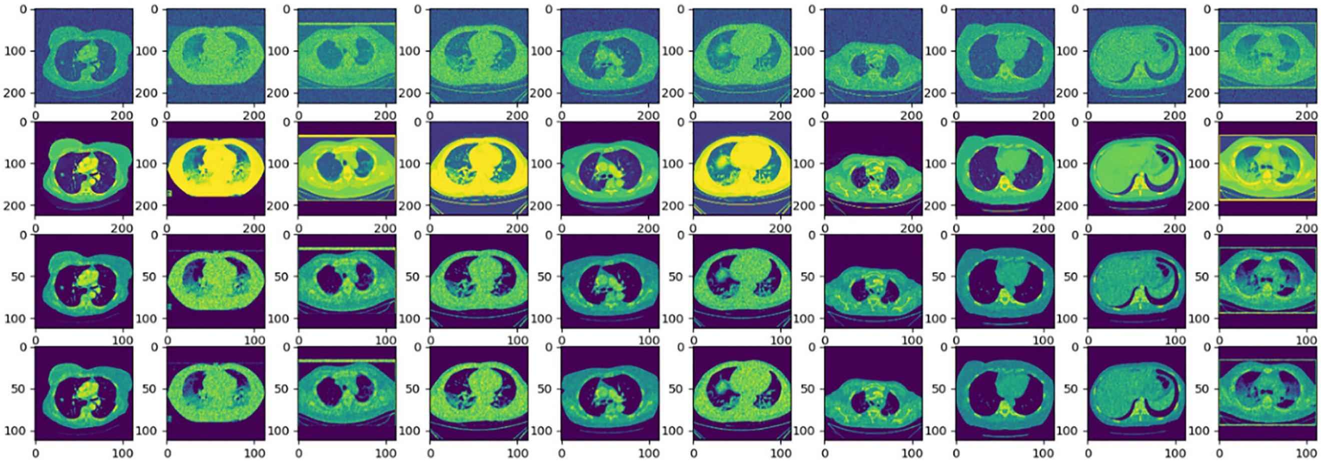


Fig. 10 Test results of layer 2

two rows of images are the original CT images and the CT images after adding noise. The third row is the output of the original CT image after the dimension reduction of the first layer. The fourth row is the decoding output of the layer 2. We can see from Fig. 10 that the output of features after deeper extraction is in a state of low-dimensional features. And the detection model can quickly obtain the low-dimensional features of this layer and optimize the parameters of this layer through gradient descent. We can also see from Figs. 9 and 10 that the input and output feature maps of autoencoder layer 1 and layer 2 are very close. This is also the obvious manifestation of the regularization effect after adding the reconstruction loss as the regularization item in each layer.

Thirdly, a stacked autoencoder network is a method of using an autoencoder network, which is a neural network composed of multi-layer trained autoencoder. Since each layer in the network is trained separately, it is equivalent to initialize a reasonable value for the parameters of each layer in the network before the cascade training. So, this network is easier to train and has faster convergence and higher accuracy. It is usually not easy to build a complete set of available models for high-dimensional classification problems. Only blindly increasing depth will only make the results more and more uncontrollable. And the network will be an uncontrollable black box in the end [41]. The dimension reduction layer by layer can simplify the complex problem. The stacked autoencoder detector can be used to train any deep network. For the stacked autoencoder, the features after dimensionality reduction are directly used for the secondary training. The depth of arbitrary layers can be deepened without worrying about the gradient disappearance in the training.

6 Conclusions

In this paper, we have proposed a fast and accurate stacked autoencoder detection model to detect COVID-19 cases from chest CT images. And our model is fully automated with an end-to-end structure without the need for manual feature extraction. In the current severe epidemic, our model can detect COVID-19 positive cases quickly and efficiently. The stacked autoencoder detector model can help front-line clinicians to diagnose suspected cases. And the auxiliary diagnostic model developed by using artificial intelligence methods such as deep learning is of great significance to the prevention and control of epidemic diseases in countries and regions with a shortage of medical materials and equipment in the world. Besides, with the release of more and more COVID-19 chest CT scan image datasets, the detection accuracy of such deep learning models as the stacked autoencoder detector will be greatly improved. It will play a great role in the prevention and control of the COVID-19 epidemic and cutting off the transmission chain.

Acknowledgements This work is supported by the National Natural Science Foundation of China under grant U1836110, 61602253, 1809205, 61771249, 91959207, 81871352; Natural Science Foundation of Jiangsu Province of China (No. BK20181411); Special Foundation by Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) and Jiangsu Key Laboratory of Big Data Analysis Technology (B-DAT) (No. 2020xtzx005).

References

1. Gorbalenya AE, Baker SC, Baric RS et al (2020) The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2. Nat Microbiol 5:536–544. https://doi.org/10.1038/s41564-020-0695-z
2. Chow KYC, Raymond KHH et al (2003) Molecular advances in severe acute respiratory syndrome-associated coronavirus (SARS-CoV). Genomics Proteomics Bioinforma 1(4):247–262
3. Thompson R (2020) Pandemic potential of 2019-nCoV. Lancet Infect Dis 20(3):261. https://doi.org/10.1016/S1473-3099(20)30068-2
4. Kumar D, Oriol M, Yoichiro N et al (2020) COVID-19: A global transplant perspective on successfully navigating a pandemic. Am J Transplant. https://doi.org/10.1111/ajt.15876
5. Subbaraman N (2020) Coronavirus tests: researchers chase new diagnostics to Fight the pandemic. Nature. https://doi.org/10.1038/d41586-020-00827-6
6. Shen ZJ, Yan X, Lu K et al (2020) Genomic diversity of SARS-CoV-2 in coronavirus disease 2019 patients. Clin Infect Dis. https://doi.org/10.1093/cid/ciaa203
7. Wang CT, Liu ZP, Chen ZX et al (2020) The establishment of reference sequence for SARS-CoV-2 and variation analysis. J Med Virol. https://doi.org/10.1002/jmv.25762
8. Yang P, Wang X (2020) COVID-19: a new challenge for human beings. Cell Mol Immunol 17:555–557. https://doi.org/10.1038/s41423-020-0407-x
9. Udugama B, Pranav K, Hannah NK et al (2020) Diagnosing COVID-19: The disease and tools for detection. ACS Nano. https://doi.org/10.1021/acsnano.0c02624
10. Maxmen A (2020) How poorer countries are scrambling to prevent a coronavirus disaster. Nature. https://doi.org/10.1038/d41586-020-00983-9
11. Li Y, Xia LM (2020) Coronavirus disease 2019 (COVID-19): role of chest CT in diagnosis and management. Am J Roentgenol :1–7
12. Ting DSW, Carin L, Dzau V et al (2020) Digital technology and COVID-19. Nat Med 26:459–461. https://doi.org/10.1038/s41591-020-0824-5
13. Bai HX, Wang R, Xiong Z et al (n.d.) AI Augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of origin at chest CT. Published Online:Apr 27, 2020. https://doi.org/10.1148/radiol.2020201491
14. Oh Y, Park S, Ye JC (2020) Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets. IEEE Trans Med Imaging 39(8):2688–2700. https://doi.org/10.1109/TMI.2020.2993291
15. Hernandez-Matamoros A, Fujita H, Hayashi T, Perez-Meana H (2020) Forecasting of COVID19 per regions using ARIMA models and polynomial functions. Appl Soft Comput 106610. https://doi.org/10.1016/j.asoc.2020.106610
16. Xinggang W, Xianbo D, Qing F et al (2020) A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT. IEEE Trans Med Imaging 39(8):2615–2625. https://doi.org/10.1109/TMI.2020.2995965
17. Ouyang X, Jiayu H, Liming X et al (2020) Dual-sampling attention network for diagnosis of COVID-19 from community acquired

pneumonia. IEEE Trans Med Imaging 39(8):2595–2605. https://doi.org/10.1109/TMI.2020.2995508

18. Chen L, Bentley P, Mori K et al (n.d.) Self-supervised learning for medical image analysis using image context restoration. Med Image Anal 58:101539. https://doi.org/10.1016/j.media.2019.101539
19. Gao F, Yoon H, Wu T et al (2020) A feature transfer enabled multi-task deep learning model on medical imaging. Expert Syst Appl 143:112957. https://doi.org/10.1016/j.eswa.2019.112957
20. Kim M, Yan C, Yang D et al (2020) Chapter Eight-Deep learning in biomedical image analysis. In Biomedical Information Technology (Second Edition), Feng DD (ed) Academic Press, Cambridge, pp 239–263. https://doi.org/10.1016/B978-0-12-816034-3.00008-0
21. Zhou D-X (2020) Theory of deep convolutional neural networks: Downsampling. Neural Netw 124:319–327. https://doi.org/10.1016/j.neunet.2020.01.018
22. Pasa F, Golkov V, Pfeiffer F et al (2019) Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization. Sci Rep 9(1):6268. https://doi.org/10.1038/s41598-019-42557-4
23. Miki Y, Muramatsu C, Hayashi T et al (n.d.) Classification of teeth in cone-beam CT using deep convolutional neural network. Comput Biol Med 80:24–29. https://doi.org/10.1016/j.compbiomed.2016.11.003
24. Zhao JY, Zhang YC, He XH et al (2020) COVID-CT-Dataset: a CT scan dataset about COVID-19. ArXiv: abs/2003.13865
25. Song Y, Zheng S, Li L et al. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. medRxiv. https://doi.org/10.1101/2020.02.23.20026930
26. Wang S, Kang B, Ma J et al. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). medRxiv. 2020. https://doi.org/10.1101/2020.02.14.20023028
27. Butt C, Gill J, Chun D et al (2020) Deep learning system to screen coronavirus disease 2019 pneumonia. Appl Intell. https://doi.org/10.1007/s10489-020-01714-3
28. Khan AL, Junaid LS (2020) CoroNet: MB A Deep Neural Network for Detection and Diagnosis of Covid-19 from Chest X-ray Images. Comput Methods Programs Biomed 196(11):105581. https://doi.org/10.1016/j.cmpb.2020.105581
29. Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B et al (2020) Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology :200905. https://doi.org/10.1148/radiol.2020200905
30. Hanin B (2018) Which neural net architectures give rise to exploding and vanishing gradients? Neural information processing systems, pp 582–591. ArVix: 1801.03744
31. Vincent P, Larochelle H, Lajoie I et al (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res :3371–3408. https://doi.org/10.5555/1756006.1953039
32. Majumdar A, Aditay T (2017) Asymmetric stacked autoencoder. 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, Anchorage, pp 911–918. https://doi.org/10.1109/IJCNN.2017.7965949
33. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
34. Köksoy O (2006) Multiresponse robust design: Mean square loss (MSE) criterion. Appl Math Comput 175(2):1716–1729. https://doi.org/10.1016/j.amc.2005.09.016
35. Kingma D, Ba J (2015) Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015). arXiv:1412.6980
36. Xu B, Ruitong H, Mu L (2016) Revise saturated activation functions. ICLR. arXiv:1602.05980
37. Bottou L (2012) Stochastic gradient descent tricks. Neural Networks: Tricks of the Trade, pp 421–436. https://doi.org/10.1007/978-3-642-35289-8_25
38. Lillicrap TP, Santoro A, Marris L et al (2020) Backpropagation and the brain. Nat Rev Neurosci 21:335–346. https://doi.org/10.1038/s41583-020-0277-3
39. Saxe AM, Koh PW, Chen ZH et al (2011) On random weights and unsupervised feature learning. ICML 2(3). https://doi.org/10.5555/3104482.3104619
40. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4): 427–437. https://doi.org/10.1016/j.ipm.2009.03.002
41. Aggarwal A, Lohia P, Nagar S, Dey K, Saha D (2019) Black box fairness testing of machine learning models. Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering - ESEC/FSE 2019. https://doi.org/10.1145/3338906.3338937

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.