


RESEARCH



MFDNN: multi-channel feature deep neural network algorithm to identify COVID19 chest X-ray images

Liangrui Pan¹ , Boya Ji¹, Hetian Wang¹, Lian Wang¹, Mingting Liu¹, Mitchai Chongcheawchamnan² and Shaolaing Peng^{1*}

Abstract

The use of chest X-ray images (CXI) to detect Severe Acute Respiratory Syndrome Coronavirus 2 (SARS CoV-2) caused by Coronavirus Disease 2019 (COVID19) is life-saving important for both patients and doctors. This research proposes a multi-channel feature deep neural network (MFDNN) algorithm to screen people infected with COVID19. The algorithm integrates data over-sampling technology and MFDNN model to carry out the training. The oversampling technique reduces the deviation of the prior probability of the MFDNN algorithm on unbalanced data. Multi-channel feature fusion technology improves the efficiency of feature extraction and the accuracy of model diagnosis. In the experiment, Compared with traditional deep learning models (VGG19, GoogLeNet, Resnet50, Desnet201), the MFDNN model obtains an average test accuracy of 93.19% in all data. Furthermore, in each type of screening, the precision, recall, and F1 Score of the MFDNN model are also better than traditional deep learning networks. Furthermore, through ablation experiments, we proved that a multi-channel convolutional neural network (CNN) is superior to single-channel CNN, additional layer and PSN module, and indirectly proved the sufficiency and necessity of each step of the MFDNN classification method. Finally, our experimental code will be placed at <https://github.com/panliangrui/covid19>.

Keywords: COVID19, Chest X-ray, Multi-channel feature, MFDNN

Introduction

SARS-CoV-2 causes COVID19. Since the first report, it has become a global pandemic, with 180 million confirmed cases and 3.91 million deaths. It is extremely contagious characteristics and delayed vaccination has made developing countries vulnerable to virus attacks. Now the nucleic acid detection mechanism has played an essential role in screening the flow of people. Reverse Transcription Polymerase Chain Reaction (RT-PCR) is the most standard diagnostic technology available [1]. However, its sensitivity is relatively low, and the result is highly dependent on the sample area obtained and heavily reliant on the operator's technique [2]. The important

thing is that this method takes time. However, time is a key factor in isolating, preventing and treating people infected with COVID19, which will limit the efficiency of COVID19 screening. With the global spread of COVID19, medical research has found that CXI can identify people infected with COVID19. Therefore, as a supplement to RT-PCR technology, it plays an essential role in detecting and evaluating people infected with COVID19.

Computed tomography (CT), lung ultrasound (LUS), and Chest-X ray radiography are among the most commonly used imaging modalities to identify COVID19 infections [3–5]. It is widely used in large hospitals because of the safety, painlessness, non-invasiveness, clear image, high-density resolution, and apparent morbidity of CXI. In addition, experienced doctors can make real-time diagnoses through CXI. Therefore, CXI is one

*Correspondence: slpeng@hnu.edu.cn

¹ College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

Full list of author information is available at the end of the article

of the most commonly used and readily available methods to detect COVID19 infections [6]. However, there are many similarities in the CXI characteristics of patients with COVID19 and common pneumonia, which poses a massive challenge to radiologists in diagnosing patients with COVID19.

In recent years, artificial intelligence has prompted tremendous progress in the field of biomedicine, such as medical diagnosis, intelligent image recognition, intelligent health management, intelligent drug development, and medical robots [7–9]. Machine learning-based methods have developed many applications in the accurate analysis of CXI, such as diagnosing and evaluating people infected with COVID19 [10, 11]. Standard machine learning algorithms include linear regression, random forest (RF), K-nearest neighbor (KNN), decision tree (DT), etc. [12, 13]. Abolfazl et al. used dimensionality reduction methods to extract the best features of CXI to build an efficient machine learning classifier, and the classifier distinguishes COVID19 and non-COVID19 cases with high accuracy and sensitivity [6]. Dan et al. used three different machine learning models to predict the deterioration of the patient's condition, compare them with the currently recommended predictors and APACHEII risk prediction scores, and obtain high sensitivity, specificity, and accuracy [14]. Mohamed et al. used the new fractional multi-channel exponential moments (FrMEMs) to extract features from CXI [15], p. 19).

Then, the improved Manta-Ray Foraging Optimization (MRFO) method was used for feature selection, and the KNN method was used to classify the two types of CXI [15]. However, deep learning is the hottest research direction in the field of machine learning. The CXI deep learning method for COVID19 classification has been actively explored. Linda et al. proposed a deep convolutional neural network called COVID-Net to help clinicians improve screening [16]. Ali et al. proposed five models based on pre-trained convolutional neural networks (ResNet50, ResNet101, ResNet152, InceptionV3, and Inception-ResNetV2) to implement four different binary classifications: COVID19, normal (healthy), viral pneumonia and bacterial pneumonia. CXI has achieved a high accuracy rate [17]. Ioannis et al. automatically detected CXI based on the transfer learning method of convolutional neural network and achieved 96.78%, 98.66%, and 96.46% accuracy, sensitivity, and specificity [18]. Ezz et al. proposed the COVIDX-Net network based on seven deep convolutional network models with different architectures and obtained F1 scores of 0.89 and 0.91, respectively [19].

Inspired by machine learning and deep learning and the accumulation of previous work experience, we will further explore the impact of experimental data and deep convolutional neural networks on the detection

algorithm or detection system in this article. However, in most databases, unbalanced label classes often occur, which will cause the convolutional neural network to be biased to correctly identify image data with many class labels. Multi-channel CNN is usually superior to single-channel CNN in terms of computational efficiency and accuracy [20–22]. A similar method has been proposed due to previous work. However, the previous work mainly focused on optimizing the processing features on multiple channels, and did not discuss fusion, nor did it optimize the features after fusion. Therefore, the experiment selected a multi-channel input single-channel output parallel deep neural network as the front part of the algorithm, additional layer as the main method of feature fusion, and added the features extracted from the multi-channel. As the tail of MFDNN, PSN can perform secondary convolution operations on the basis of feature fusion to improve the efficiency and accuracy of feature extraction. Two similar neural networks map the input to a new space and represent the output in the new space. By calculating the value of loss, the similarity between image features is calculated [23].

Therefore, the main focus of this article is to solve the accuracy of the COVID19 detection algorithm. Around this problem, we will solve the following problems separately: (1) Deal with the imbalance of sample labels. (2) Optimize the feature extraction of the deep neural network algorithm. (3) Evaluate the classification effect of the network algorithm. To achieve this goal, we first analyze the degree of imbalance in the sample data. We found that chest X-ray data of people who were not infected with COVID19 was significantly more than other categories through the data set analysis. To be able to classify the chest radiograph data set more accurately, our main contributions are as follows.

- (1) To balance the impact of the unbalanced label data set on model training, when processing CXI, we embed the oversampling method into the model to balance all categories of data.
- (2) We propose an MFDNN algorithm based on multi-channel input, single-channel output, and centralized weight sharing. The algorithm model concentrates the feature maps of multi-channel chest radiographs and optimizes the feature extraction process. As the tail of MFDNN, PSN can extract features from the additional layer for the second time.
- (3) Finally, the MFDNN is compared with the classic deep neural networks (VGG19, GoogLeNet, Resnet50, Desnet201). The MFDNN model is better than other models in precision, recall, and F1 Score and confusion matrix.

Materials and methods

In this section, we first introduce the flow chart of the MFDNN algorithm. It mainly includes two parts: over-sampling and the MFDNN model. The first part is primarily data preprocessing, and the second part primarily uses the MFDNN model for feature extraction and patient diagnosis. Figure 1 shows our proposed algorithm classification process.

Materials

X-rays pass through the chest, and different body parts absorb the rays, and the film will not be exposed or partially exposed. After the film is processed, this part is white, forming an imaging manifestation. As shown in Fig. 2, COVID images have the symptom of "white lung"; a large lung area is white. The lung opacity class is very similar to the normal class. The main difference is the increase in lung texture. Small area texture image blur appears in viral pneumonia images, which may be due to lung inflammation caused by other viruses. Compared with COVID, the contour of both lungs is visible, the transparency is acceptable, and the texture is slightly obvious. The proposed MFDNN model is trained and tested on a public dataset (COVID19 chest X-ray dataset). The data set consists of 3616 COVID19 positive cases, 10,192 normal, 6012 lung opaque (non-COVID

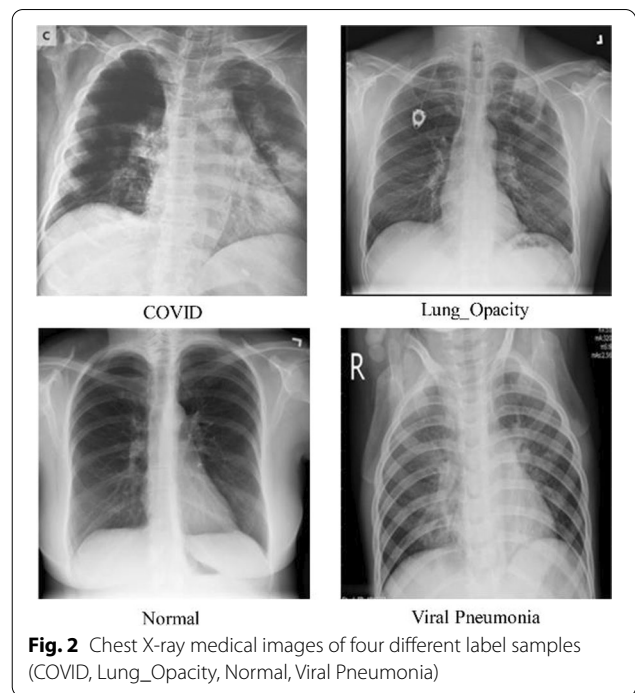


Fig. 2 Chest X-ray medical images of four different label samples (COVID, Lung_Opacity, Normal, Viral Pneumonia)

lung infection) and 1345 viral pneumonia images [24, 25]. The dataset can be downloaded from the website [25]. Before the experiment, we need to have a preliminary understanding of the CXI data set.

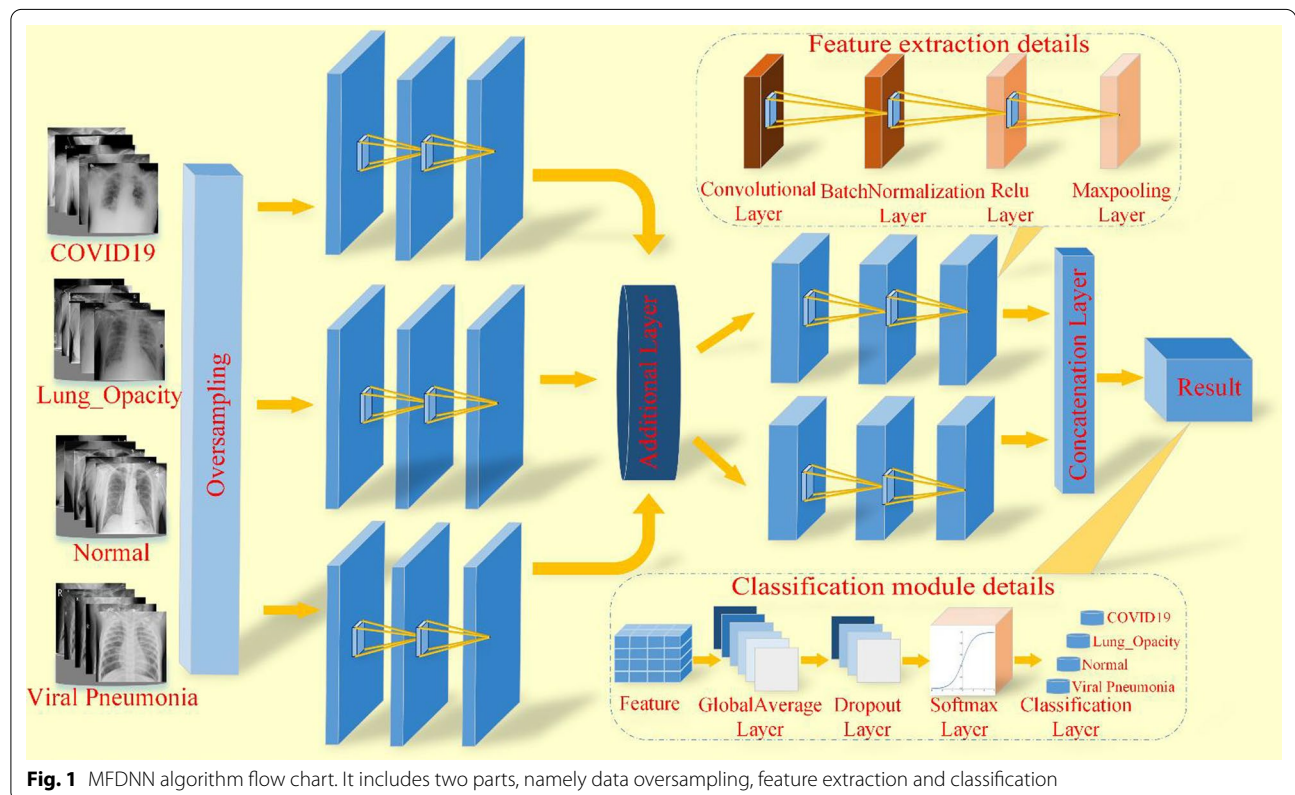


Fig. 1 MFDNN algorithm flow chart. It includes two parts, namely data oversampling, feature extraction and classification

Oversampling

The classification of imbalanced data sets is still a problematic point for deep neural network classification. Different methods have been used in the literature to deal with unbalanced data [26–28]. The commonly used method is the resampling technique. In addition, the resampling method includes two methods, namely under-sampling and over-sampling. In oversampling, the minority class samples are copied to balance the size of each class in the training data. In undersampling, some majority class samples are removed during the training process to balance the size of each class. Therefore, when the model is trained on balanced data, it should exhibit unbiased behavior. Different types of undersampling methods have been proposed [26–28]. However, it is reported that random oversampling is the simplest method and exhibits similar performance to other complex methods. Therefore, in this article, we use random oversampling to balance the training process and reduce the bias in building the model.

As shown in Fig. 3a, by comparing the characteristics of the database, the chest radiographs of people infected with COVID19 are significantly less than those of normal people. However, the steps of neural networks and human brain extraction are similar. Therefore, when the probability of memorizing normal chest radiographs of the MFDNN model is greater than remembering the chest radiographs of COVID19 infected persons, the model is more likely to recognize the chest radiographs of normal people, which may lead to COVID19 infection. Therefore, the probability of screening by the user is reduced, and the model cannot be applied to the actual detection process. Therefore, the experiment chooses to oversample to generate new samples for a few categories to ensure that the model has the same probability of remembering different CXI. Figure 3b shows the result of oversampling. The sample data size of the minority class is the same as the sample data size of the majority class.

Multi-channel feature deep neural network

The multi-channel feature deep neural network (MFDNN) algorithm is experimentally designed. The oversampling dataset passes through three identical feature extraction modules. First, the features will be merged in the middle of the frame. Then further feature extraction is performed on the collected

image features through the Siamese network, which improves image feature extraction work efficiency. Finally, we will introduce the function of each layer in detail from the feature extraction module.

The image can be seen as a high-dimensional matrix composed of feature vectors. In feature extraction, using a small convolution kernel can reduce the convolution operation's error rate, so the 3×3 convolution kernel is selected in the convolution layer, and the step size is 1 for matrix operation. The convolutional layer is defined as:

$$Y_{i,j,k}^l = W_k^{lT} X_{i,j}^l + b_k^l \quad (1)$$

The feature value at the position (i, j) of the k th feature map of the l th layer is $Y_{i,j,k}^l$. Where W_k^l is the weight of the l th layer, b_k^l is the bias of the l th layer, and $X_{i,j}^l$ is the (i, j) unknown input block of the l th layer. Thus, 64 convolution kernels simultaneously perform local perception and share parameters on the input image.

Batch Normalization (BN) is widely used training technique in deep networks. A BN layer whitens activations within a mini-batch of N examples for each channel dimension and transforms the whitened activations using affine parameters γ and β , denoting by $\chi \in R^{H \times W \times N}$ activations in each channel, BN is expressed as [29]:

$$BN(X[i, j, n]; \gamma, \beta) = \gamma \cdot \hat{X}[i, j, n] + \beta \quad (2)$$

where

$$\hat{X}[i, j, n] = \frac{X[i, j, n] - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (3)$$

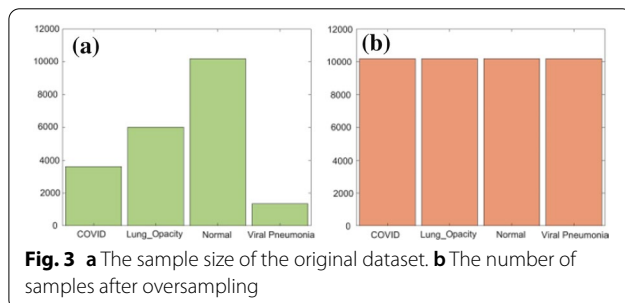
The mean and variance of activations within a mini-batch,

$$\mu = \frac{\sum_n \sum_{i,j} X[i, j, n]}{N \cdot H \cdot W} \quad (4)$$

$$\sigma^2 = \frac{\sum_n \sum_{i,j} (X[i, j, n] - \mu)^2}{N \cdot H \cdot W} \quad (5)$$

Select the Rectified Linear Unit (ReLU) function as the convolutional layer's activation function to reduce the probability of model overfitting [30]. The ReLU function will make part of the neuron output 0 to enhance the sparsity of the network. Besides, it reduces the interdependence between parameters and alleviates the problem of overfitting. For the MFDNN model, the ReLU function enables each neuron to exert the greatest screening effect, saving a lot of calculations in the whole process, which is defined as:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \quad (6)$$



The maximum pooling method is selected to obtain the maximum value of the feature tiles from the convolutional layer as the output in the pooling layer. The down-sampling convolution kernel size is set to 2×2 , the stride size is set to 2, and the feature matrix after the pooling operation is filled in a "same" manner to alleviate the excessive sensitivity of the convolution layer to the position. The maximum pooling layer reduces the parameters by reducing the dimension, removing redundant features, simplifying the network's complexity, and other methods to achieve nonlinear feature extraction. The input X_i^l of the l th layer is mapped to the output Y_i^l through the neuron, which is defined as:

$$Y_i^l = \max \{X_i^l\}. \tag{7}$$

The additional layer is the main part of multi-channel feature fusion. The additional layer serves as an intermediate hub for merging the output from the pooling layer, combining feature weights. Assuming the output is, its effect can be expressed as:

$$Y = \sum_{i=1}^n x_i^l \tag{8}$$

As the input Y of the global average pooling (GAP) layer, the learned "distribution feature representation" is selectively mapped to the labelled sample space. The activation function of each neuron in the GAP layer generally uses the ReLU function [31]. It can replace the fully connected layer in the traditional structure, thereby reducing the amount of storage required for the large weight matrix of the fully connected layer. Moreover, it also has the features and capabilities of easy fine-tuning of a pre-trained model with a conventional structure. Since the working principle of the fully connected layer involves calculating the inner product of the input vector and the weight of each row, the row size of the weight matrix needs to be the same as the number of input elements [32]. Therefore, as the input changes, we also need to adjust the weight matrix f, W , to a corresponding size by

$$W'_{i,j} = \sum_{l=(j-1) \times \text{size}_{fm}^2 + 1}^{j \times \text{size}_{fm}^2} W_{i,l} \tag{9}$$

where size_{fm} is the size of the input feature map, i, j is the index of the output neurons and input feature maps, and W' is the modified weight matrix [32]. Given the computational complexity in the GAP layer, the dropout layer chooses a 40% random probability to discard some feature weights to reduce the model complexity and prevent overfitting. Finally, it is classified by Softmax, and the output of multiple neurons is mapped to the interval of $(0, 1)$, which is defined as:

$$S_i = \frac{e^i}{\sum_j e^j} \tag{10}$$

We use $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ to represent m training samples and $y^{(i)}$ to represent the label of i samples, and train the neural network by using gradient descent. In this article, the cross-entropy function is used to calculate the Loss of the MFDNN model. For a single example, the cross-entropy loss function can be expressed as:

$$L(w, b, x^{(i)}, y^{(i)}) = - \sum_{l=1}^k 1\{y^{(i)} = l\} \log \frac{e^{h_l(x,w,b)}}{\sum_l^k e^{h_l(x,w,b)}} \tag{11}$$

where $h_l(x, w, b)$ represents the l th neuron in the output layer corresponding to the l th type, $1\{\cdot\}$ is the indicator function. The weight parameter is continuously updated through the backpropagation loss function. In order to better integrate upsampling and MFDNN, we propose an MFDNN classification algorithm for detecting COVID19 patients.

Algorithm 1: MFDNN Classification Model

Input: Image Dataset shape = $224 \times 224 \times 3$

Output: $y = [P(y_{COVID}), P(y_{Lung_Opacity}), P(y_{Normal}), P(y_{Viral\ Pneumonia})]$

Steps:

1. Read Images in batches of 32 batch of size $224 \times 224 \times 3$
2. Oversampled CXI database
3. Initialization: $\theta_c = \text{Xavier}$ initialization, epoch=0
4. MFDNN for feature extraction of data (feature vector = $[(\theta_1^1 \xi_1 + \theta_2^1 \xi_2 + \dots + \theta_n^1 \xi_n)]$)
5. BN(feature vector), Relu(feature vector)
6. Additional layer merge feature map, (feature vector = $[(\theta_1^1 \xi_1 + \theta_2^1 \xi_2 + \dots + \theta_n^1 \xi_n) + (\theta_1^2 \xi_1 + \theta_2^2 \xi_2 + \dots + \theta_n^2 \xi_n) + (\theta_1^3 \xi_1 + \theta_2^3 \xi_2 + \dots + \theta_n^3 \xi_n)]$)
7. The PSN extracts features from the feature map of the additional layer
8. GAP layer (all feature vector) \rightarrow Dropout(40%)
9. Softmax(features_to_predict_on) \rightarrow predictions = $f(\text{feature_to_predict_on}[i]) = S_i = \frac{e^i}{\sum_j e^j}$
10. Find loss on the prediction according to categorical_crossentropy loss \rightarrow

$$L(w, b, x^{(i)}, y^{(i)}) = - \sum_{l=1}^k 1\{y^{(i)} = l\} \log \frac{e^{h_l(x,w,b)}}{\sum_l^k e^{h_l(x,w,b)}}$$

11. Update weight of hidden layers as well as input W_3 according to Adam optimizer \rightarrow

$$\theta = \theta - \frac{(a - V_{d\theta}^{corrected})}{\sqrt{S_{d\theta}^{corrected} + \psi}}$$

Output $y = [P(y_{class1}), P(y_{class2}), P(y_{class3}), P(y_{class4})]$ (where $P(a) \rightarrow$ Probability of class)

Results and discussion

Experimental settings

Before oversampling, the data set is divided into training and test data at a ratio of 0.8:0.2. Then, the over-sampled training data is divided into training set and validation set according to the ratio of 0.8:0.2. Before training the MFDNN model, we choose to flip the data and augment the data with translation to expand the training data to avoid over-fitting the model. The experiment is set to 30 epochs, the batch size is set to 32, and the Adam algorithm is used as the optimizer of the model. The initial learning rate is 0.003, and after each epoch, the learning rate will drop by half. Before each epoch training, the training data and verification data will be randomly shuffled. Each model is trained on a single RTX3060.

Results

In this section, we will explain the evaluation indicators used to quantify model classification. To this end, we use an indicator based on a confusion matrix. These indicators include test accuracy, precision, recall, and F1 Score. To evaluate the model, we need to perform a detailed analysis of each category. Therefore, we need to count true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [33].

1. Test accuracy: the proportion of samples correctly predicted to the total samples

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

2. Precision: the ratio of true positive predictions to total positive predictions

$$\text{Precision} = \frac{TP}{TP + FP} \quad (13)$$

3. Recall: Ratio of true positive to the total observation made by the proposed model

$$\text{Recall} = \frac{TP}{TP + FN} \quad (14)$$

4. F1 Score: It is the harmonic mean of precision and recall

$$F1 \text{ score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (15)$$

5. Confusion matrix: It is the measurement of the performance of the model. It compares the actual and predicted values in the form of True Positive, False Negative, True Negative and False Positive

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (16)$$

- True Positive (TP): True positive are the forecasts which were at first positive and, additionally, anticipated by the AI model as positive.
- False Positive (FP): False positives are the forecasts which were initially negative and anticipated by the AI model as positive.
- True Negative (TN): True negatives are the forecasts which were initially negative and anticipated by the AI model as unfavourable.
- False Negative (FN): False-negative are the forecasts which were initially positives and anticipated by the model as negative

The experiment first trained five models under the algorithm of MFDNN, namely Densenet201, ResNet50, VGG19, GoogLeNet, and MFDNN. Among them, the accuracy of the MFDNN model is 93.19%. Among them, the COVID category received a Recall of 0.9447 and an F1 score of 0.9358; the Lung_Opacity category received a precision of 0.9144 and an F1 score of 0.9106; the Normal class received a recall of 0.9431 and an F1 score of 0.9389; the Viral Pneumonia category received an F1 score of 0.9504. Table 1 details the test reports of each type of chest radiograph under different models. From the classic deep learning model analysis, for the COVID19 data set, the deeper the network layer, the worse the effect of the model. For example, the test results of the Densenet201 model only get good prediction results in a few categories. GoogLeNet obtains the best results in the classic deep learning network, but compared to the MFDNN model, the traditional deep learning model does not achieve the best test results.

Secondly, Fig. 4 describes the confusion matrix of each model prediction test set. This can give us a rough idea of how all images are classified and where most misclassifications occur. It can be seen from the figure that the probability of the prediction error of the Normal class is greater than the probability of the prediction error of the other classes. This shows that the up-sampling method embedded in the algorithm has a positive effect. It makes the model not biased to ignore infected patients during the detection process. When the MFDNN model discriminates four types of samples, the misjudgment rate is relatively low.

Ablation experiment

To prove the necessity of steps 6 and 7 in the MFDNN classification model, we designed an ablation experiment to explore the accuracy of multi-channel feature

fusion combined with PSN. The first step is to remove the PSN and additional layers, generate a simple CNN, train the model, and record the evaluation index value of the model. The second step is to remove the additional layer, keep the remaining part of the MFDNN model and name it DNN. Again, train the model and record the evaluation index value of the model. The third step is to use only the feature fusion method to generate a multi-channel feature convolutional neural network (MFCNN), train the model and record the evaluation index value of the model. Finally, the fourth step compares all the model detection results with the MFDNN model classification results.

The impact of additional layers

In order to verify the influence of the additional layer, we compared the performance of the three models of CNN, DNN, and MFDNN. As shown in Table 2, we analyzed the performance of CNN and DNN models from accuracy, recall, F1 score and test accuracy. In most indicators of all categories, the evaluation indicators of the MFDNN model are higher than other models. In terms of test accuracy, the MFDNN model is 3.33% higher than the DNN model.

Secondly, we have a detailed understanding of the classification effect of the model in different categories through the confusion matrix. By comparing the confusion matrix analysis of the CNN and DNN models in Fig. 5 with the MFDNN model in Fig. 4, we find that the MDNN model has smaller errors than the CNN and DNN models in detecting all categories. Therefore, the

Role of PSN

According to the data analysis in Table 3, we found that MFCNN and MFDNN models have a large gap in accuracy, recall and F1 score. The test accuracy of the MFDNN model is 0.0333 higher than that of the MFCNN model. Compared with the confusion matrix of the MFCNN model and the MFDNN model in Fig. 6, we find that the MFCNN model has greater errors than MFDNN in detecting all categories. Because the MFCNN model does not include the PSN module, the MFDNN model includes the PSN module. Therefore, we conclude that the function of PSN is to perform secondary feature extraction based on the extracted features, so that the features can be used more effectively and the accuracy of the model can be improved.

Ablation experiment

This paper proposes an MFDNN algorithm to screen people infected with COVID19. The algorithm integrates data oversampling technology and a MFDNN model to carry out the training. In the experiment, we used the publicly available CXI database to train the model. First, compared with traditional deep learning models (VGG19, GoogLeNet, ResNet50, Densenet201), the MFDNN model obtains an average test accuracy of 93.19% in all data. Furthermore, in each type of screening, the precision, recall, and F1 Score of the MFDNN model are also better than traditional deep learning networks. Through ablation experiments, we proved that multi-channel CNN is superior to single-channel

Table 1 Densenet201, ResNet50, VGG19, GoogLeNet, MFDNN classification technical report

		Densenet201	ResNet50	VGG19	GoogLeNet	MFDNN
COVID	Precision	0.9272	0.9473	0.9532	0.9572	0.9369
	Recall	0.8105	0.87	0.7331	0.8976	0.9447
	F1 score	0.8649	0.907	0.8288	0.9264	0.9358
Lung_Opacity	Precision	0.7198	0.801	0.7968	0.854	0.9144
	Recall	0.9359	0.9143	0.8611	0.9002	0.9068
	F1 score	0.8137	0.8539	0.8277	0.8765	0.9106
Normal	Precision	0.941	0.9302	0.8627	0.9266	0.9348
	Recall	0.8057	0.89	0.9001	0.923	0.9431
	F1 score	0.8681	0.9097	0.881	0.9248	0.9389
Viral pneumonia	Precision	0.9007	0.9675	0.956	0.9611	0.9257
	Recall	0.9777	0.8847	0.8885	0.9182	0.9765
	F1 score	0.9376	0.9242	0.921	0.9392	0.9504
Test accuracy		0.8544	0.8932	0.8599	0.9113	0.9319

Bold values indicate that the metric is optimal for that row

additional layer can capture a wider range of information from the image, thereby significantly improving the performance of the model.

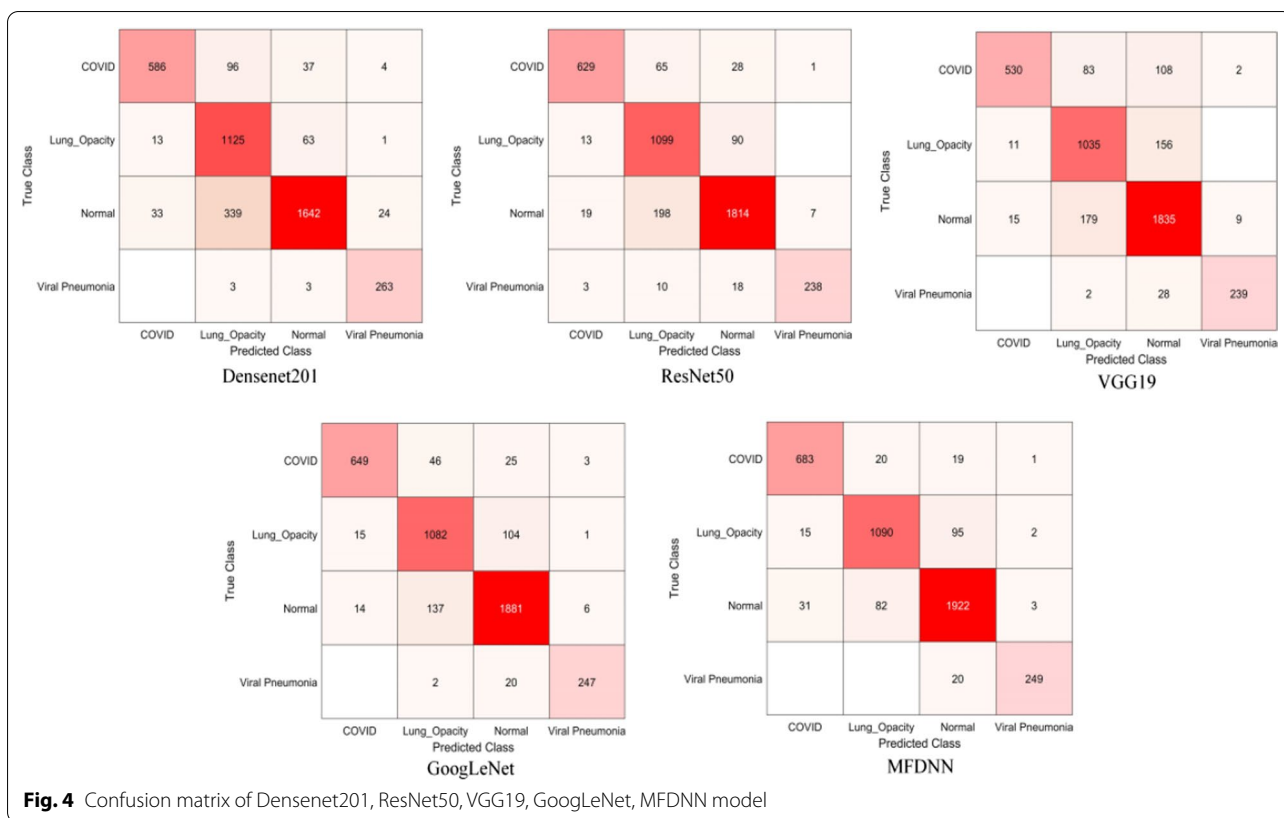
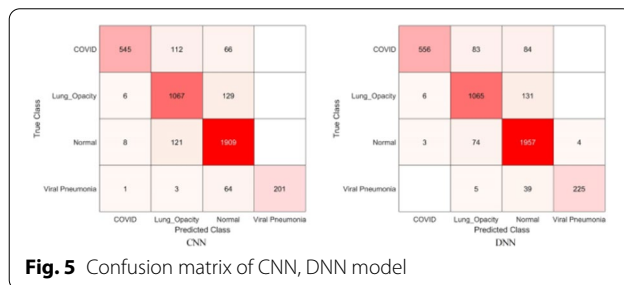


Table 2 CNN, DNN, MFDNN classification technical report

		CNN	DNN	MFDNN
COVID	Precision	0.7538	0.769	0.9369
	Recall	0.9732	0.984	0.9447
	F1 score	0.8496	0.8633	0.9358
Lung_Opacity	Precision	0.8877	0.886	0.9144
	Recall	0.8189	0.868	0.9068
	F1 score	0.8519	0.8769	0.9106
Normal	Precision	0.9367	0.9603	0.9348
	Recall	0.8805	0.8851	0.9431
	F1 score	0.9077	0.9211	0.9389
Viral pneumonia	Precision	0.7472	0.8364	0.9257
	Recall	1	0.9825	0.9765
	F1 score	0.8553	0.9036	0.9504
Test accuracy		0.875	0.8986	0.9319

Bold values indicate that the metric is optimal for that row

CNN, additional layer and PSN module, and indirectly proved the sufficiency and necessity of each step of the MFDNN classification method. Secondly, comparing the latest CoroDet model, the MFDNN algorithm is 1.91% higher than the CoroDet model in the four-classification experiment of COVID19 infected persons.



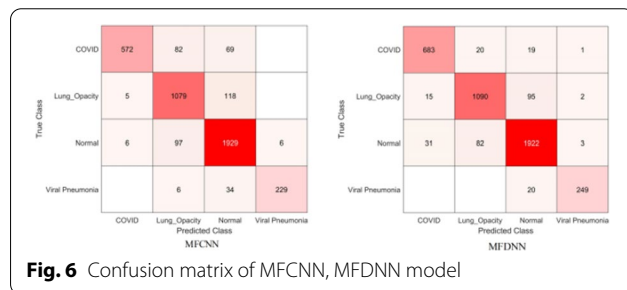
However, the limitation of this experiment is mainly in the disadvantages of X-rays. For opaque images of the lungs, RT-PCR is needed to assist in the screening of COVID19 infections.

Acknowledgements

This work was supported by National Key R&D Program of China 2017YFB0202602, 2018YFC0910405, 2017YFC1311003, 2016YFC1302500, 2016YFB0200400, 2017YFB0202104; NSFC Grants U19A2067, 61772543, U1435222, 61625202, 61272056; Science Foundation for Distinguished Young Scholars of Hunan Province (2020JJ2009); Science Foundation of Changsha kq2004010; JZ20195242029, JH20199142034, Z202069420652; The Funds of Peng Cheng Lab, State Key Laboratory of Chemo/Biosensing and Chemometrics; the Fundamental Research Funds for the Central Universities, and Guangdong Provincial Department of Science and Technology under Grant No. 2016B090918122.

Table 3 MFCNN, MFDNN classification technical report

		MFCNN	MFDNN
COVID	Precision	0.7911	0.9369
	Recall	0.9811	0.9447
	F1 score	0.8759	0.9358
Lung_Opacity	Precision	0.8977	0.9144
	Recall	0.8536	0.9068
	F1 score	0.8751	0.9106
Normal	Precision	0.9465	0.9348
	Recall	0.8972	0.9431
	F1 score	0.9212	0.9389
Viral pneumonia	Precision	0.8513	0.9257
	Recall	0.9745	0.9765
	F1 score	0.9087	0.9504
Test accuracy		0.9	0.9319

**Fig. 6** Confusion matrix of MFCNN, MFDNN model**Author details**

¹College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. ²Department of Electrical Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Thailand.

Received: 13 March 2022 Accepted: 4 April 2022

Published online: 12 April 2022

References

- Xu J, et al. Computed tomographic imaging of 3 patients with coronavirus disease 2019 pneumonia with negative virus real-time reverse-transcription polymerase chain reaction test. *Clin Infect Dis*. 2020;71(15):850–2. <https://doi.org/10.1093/cid/ciaa207>.
- Xu X, et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*. 2020;6(10):1122–9. <https://doi.org/10.1016/j.eng.2020.04.010>.
- Oh Y, Park S, Ye JC. Deep learning COVID-19 features on CXR using limited training data sets. *IEEE Trans Med Imaging*. 2020;39(8):2688–700. <https://doi.org/10.1109/TMI.2020.2993291>.
- Roy S, et al. Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound. *IEEE Trans Med Imaging*. 2020;39(8):2676–87. <https://doi.org/10.1109/TMI.2020.2994459>.
- Wang S, et al. A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. *Eur Respir J*. 2020;56(2):2000775. <https://doi.org/10.1183/13993003.00775-2020>.
- Zargari Khuzani A, Heidari M, Shariati SA. COVID-Classifier: an automated machine learning model to assist in the diagnosis of COVID-19 infection in chest X-ray images. *Sci Rep*. 2021;11(1):9887. <https://doi.org/10.1038/s41598-021-88807-2>.
- Wang X, et al. DeepR2cov: deep representation learning on heterogeneous drug networks to discover anti-inflammatory agents for COVID-19. *Brief Bioinform*. 2021. <https://doi.org/10.1093/bib/bbab226>.
- Bohr A, Memarzadeh K. The rise of artificial intelligence in healthcare applications. In: *Artificial intelligence in healthcare*. New York: Elsevier; 2020. p. 25–60. <https://doi.org/10.1016/B978-0-12-818438-7.00002-2>.
- P. Daniel et al. Artificially intelligent medical assistant robot: automating data collection and diagnostics for medical practitioners. 2021. <https://doi.org/10.13016/A9OZ-0OE7>.
- Du Y, et al. Classification of tumor epithelium and stroma by exploiting image features learned by deep convolutional neural networks. *Ann Biomed Eng*. 2018;46(12):1988–99. <https://doi.org/10.1007/s10439-018-2095-6>.
- Heidari M, et al. Prediction of breast cancer risk using a machine learning approach embedded with a locality preserving projection algorithm. *Phys Med Biol*. 2018;63(3): 035020. <https://doi.org/10.1088/1361-6560/aaa1ca>.
- ThanhNoi P, Kappas M. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors*. 2017;18(2):18. <https://doi.org/10.3390/s18010018>.
- Yoo SH, et al. Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. *Front Med*. 2020;7:427. <https://doi.org/10.3389/fmed.2020.00427>.
- Assaf D, et al. Utilization of machine-learning models to accurately predict the risk for critical COVID-19. *Intern Emerg Med*. 2020;15(8):1435–43. <https://doi.org/10.1007/s11739-020-02475-0>.
- Elaziz MA, Hosny KM, Salah A, Darwish MM, Lu S, Sahlol AT. New machine learning method for image-based diagnosis of COVID-19. *PLoS ONE*. 2020;15(6): e0235187. <https://doi.org/10.1371/journal.pone.0235187>.
- Wang L, Lin ZQ, Wong A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci Rep*. 2020;10(1):19549. <https://doi.org/10.1038/s41598-020-76550-z>.
- Narin A, Kaya C, Pamuk Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Anal Appl*. 2021. <https://doi.org/10.1007/s10044-021-00984-y>.
- Apostolopoulos ID, Mpesiana TA. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys Eng Sci Med*. 2020;43(2):635–40. <https://doi.org/10.1007/s13246-020-00865-4>.
- Hemdan EE-D, Shouman MA, Karar ME. COVIDX-Net: a framework of deep learning classifiers to diagnose COVID-19 in X-ray images. *arXiv:2003.11055* [cs, eess], 2020, Accessed: 03, 2021. [Online]. Available: <http://arxiv.org/abs/2003.11055>
- Vasudevan A, Anderson A, Gregg D. Parallel multi channel convolution using general matrix multiplication. In: *2017 IEEE 28th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, Seattle, WA, USA, 2017, pp. 19–24. <https://doi.org/10.1109/ASAP.2017.7995254>.
- Yang S, et al. Multi-channel multi-task optical performance monitoring based multi-input multi-output deep learning and transfer learning for SDM. *Opt Commun*. 2021;495: 127110. <https://doi.org/10.1016/j.optcom.2021.127110>.
- Yang B, Xiao Z. A multi-channel and multi-spatial attention convolutional neural network for prostate cancer ISUP grading. *Appl Sci*. 2021;11(10):4321. <https://doi.org/10.3390/app11104321>.
- Liu X, Zhou Y, Zhao J, Yao R, Liu B, Zheng Y. Siamese convolutional neural networks for remote sensing scene classification. *IEEE Geosci Remote Sens Lett*. 2019;16(8):1200–4. <https://doi.org/10.1109/LGRS.2019.2894399>.
- Arifin F, Artanto Nurhasanah H, Gunawan TS. Fast COVID-19 detection of chest X-ray images using single shot detection MobileNet convolutional neural networks. *J Southwest Jiaotong Univ*. 2021;56(2):235–48. <https://doi.org/10.35741/issn.0258-2724.56.2.19>.
- Chowdhury MEH, et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*. 2020;8:132665–76. <https://doi.org/10.1109/ACCESS.2020.3010287>.
- Shen F, Zhao X, Kou G, Alsaadi FE. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Appl Soft Comput*. 2021;98: 106852. <https://doi.org/10.1016/j.asoc.2020.106852>.

27. Lin W-C, Tsai C-F, Hu Y-H, Jhang J-S. Clustering-based undersampling in class-imbalanced data. *Inf Sci.* 2017;409–410:17–26. <https://doi.org/10.1016/j.ins.2017.05.008>.
28. Özdemir A, Polat K, Alhudhaif A. Classification of imbalanced hyperspectral images using SMOTE-based deep learning methods. *Expert Syst Appl.* 2021;178:114986. <https://doi.org/10.1016/j.eswa.2021.114986>.
29. Chang W-G, You T, Seo S, Kwak S, Han B. Domain-specific batch normalization for unsupervised domain adaptation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 7346–7354. <https://doi.org/10.1109/CVPR.2019.00753>.
30. Hara K, Saito D, Shouno H. Analysis of function of rectified linear unit used in deep learning. In: 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1–8. <https://doi.org/10.1109/IJCNN.2015.7280578>.
31. Gong C, et al. A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. *Sensors.* 2019;19(7):1693. <https://doi.org/10.3390/s19071693>.
32. Hsiao T-Y, Chang Y-C, Chou H-H, Chiu C-T. Filter-based deep-compression with global average pooling for convolutional networks. *J Syst Architect.* 2019;95:9–18. <https://doi.org/10.1016/j.sysarc.2019.02.008>.
33. Pan L, Pipitsunthonsan P, Daengngam C, Channumsin S, Sreesawet S, Chongcheawchamnan M. Identification of complex mixtures for raman spectroscopy using a novel scheme based on a new multi-label deep neural network. *IEEE Sensors J.* 2021;21(9):10834–43. <https://doi.org/10.1109/JSEN.2021.3059849>.