

A Hybrid Convolutional Neural Network Model Based on Different Evolution for Medical Image Classification

Yinyin Hu, Xiaoxia Zhang, Jiao Yang, Shuai Fu

ABSTRACT—In clinical practice, the X-Ray image classification for the presence or absence of lesions by the medical auxiliary diagnosis system is the basic work, and the classification of lesion images and normal images can be completed by reading the medical X-Ray graphics. The actual process is the convolution and pooling layer of CNN to extract the features of medical images, and the full connection layer of CNN is used to classify the local information. The structure of CNN needs to be adjusted repeatedly. This paper focuses on utilizing Differential Evolution (DE) to automatically search for the optimal architecture of CNN. The research idea of this paper is to use the global optimization ability of Differential Evolution algorithm to regulate the structure of CNN. When the termination condition is satisfied, DE-CNN can automatically adjust the parameters and optimize the structure of the CNN. Among them, for mutation operation, we study a new mutation strategy in this paper, which inherits the vector that may solve the next generation, and appropriately accepts the basic elements of the inferior solution to increase the disturbance of the new species. The optimal individual is found through the optimization process of the DE algorithm, and the CNN model composed of all basic elements in the individual and trained on the medical X-Ray image data set is saved. The new mutation strategy generates mutation solution vector, through experimental verification which is beneficial to the optimization of CNN structure, and the accurate value of DE-CNN algorithm processing medical image classification is better than GoogLeNet, ResNet34, VGGNet16, DenseNet, Lenet, AlexNet etc network structure.

Index Terms—Differential Evolution Algorithm, Convolution Neural Network, GoogLeNet, ResNet34, VGGNet16, DenseNet

I. INTRODUCTION

The new corona virus triggered pneumonia pandemic. Because of the long survival time and strong contagious

nature of the virus, the number of cases worldwide skyrocketed. Insufficient medical resources and overloaded work of doctors were common in areas where the epidemic was high. In addition, according to 2018 global cancer statistics, the mortality rate of breast cancer was 11.2%, ranked second among 10 common cancers [1]. Breast cancers seriously threaten women's health and lives. The types of breast tumors can be benign or malignant, in the diagnosis of breast cancer, it is the most basic work for doctors to distinguish them. The basic work also includes classifying the pathological features of X-Ray images and obtaining the corresponding diagnostic results. Reading X-Ray images artificially is a repetitive work for doctors, it can easily lead to visual and psychological fatigue and even cause diagnostic errors [2]. Imagine if the medical assistant diagnosis system could determine whether the patient's X-Ray image has lesion features and plays a very important role in clinical applications. It can intelligently simulate the process of "doctors looking at pictures" and undertake part of the work of doctors. In order to implement the algorithm of this system, we need to train the convolution neural network (CNN) structure driven by medical X-Ray images.

CNN is the most widely used classification method for deep learning. The original CNN for image classification is LeNet, it has been successfully applied to hand writing recognition and other fields. Researchers have been exploring the relationship between the depth of network structure and network performance, and have made significant progress in artificial intelligence issues such as face recognition and speech recognition. Deep learning has always been a key topic in artificial intelligence. In 1970, the MYCIN expert system was invented in the United States, this is the earliest case of artificial intelligence applied to medical auxiliary diagnosis system. With the maturity of image processing technology, medical auxiliary diagnosis system produces some new functions. For example, predicting diseases through the extraction and classification of medical image features such as CT and X-Ray. The combination of CNN and image processing technology has a higher accuracy in the classification of medical X-Ray image, and the processing of medical image data sets based on the deep learning framework has achieved more results. Used the chest X-Ray image data set, Dong et al. trained two kinds of state-of-the-art CNN [3]. The accuracy of them reached 0.82 in identifying pneumonia and normal. Bharati et al. proposed a classification method of X-Ray image about lung diseases, named VDSNet [4]. The verification accuracy of VDSNet was 0.73, it is better than VGGNet in medical image classification. In order to alleviate the shortage of medical resources, it is necessary to further

Manuscript received April 17, 2021; revised by December 3, 2021. This work was supported by Project of Liaoning Xincheng Co., Ltd (Grant No. L20170989).

X. X. Zhang is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 11041, China (corresponding author, phone: 86-0412-5929812; e-mail: aszhangxx@163.com).

Y. Y. Hu is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, AnShan 114051, China (e-mail: 2799386740@qq.com).

J. Yang is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, AnShan 114051, China (e-mail: 2036588541@qq.com).

S. Fu is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, AnShan 114051, China (e-mail: 2779664742@qq.com).

optimize the performance of algorithm for the medical assistance system. There are several aspects to optimize the performance of CNN. First is hyper parameters optimization, it needs to train and adjust the number of convolution kernels, convolution kernel size, activation function, dropout rate, and learning rate of the CNN [5]. Second is using intelligent optimization algorithm to optimize CNN structure, researchers also apply it to image classification [6]. For example, there is a new hybrid differential evolution (DE) algorithm, which evolves CNN structures of arbitrary length by adding crossover operators, called DECNN [7]. Admittedly, the above research has made some progress in CNN hyper parameters optimization, but the process of CNN hyper parameters optimization is very complex. Moreover, no researchers have proved the applicability of DE to CNN structure optimization in the process of evolution.

Since manual change of hyper parameters is a monotonous and repetitive task, we designed and implemented an algorithm which combine CNN and DE to classify X-Ray image. Before using DE to optimize the CNN, CNN is set as a conventional hyper parameter and remains unchanged, but the number of convolution layers and pooling layers of DE-CNN is adjusted. Then the optimal structure is selected according to the images' classification results. In order to initialize the DE-CNN algorithm, we randomly encode each type of CNN hyper parameters into the individuals of the initial population. Then CNN is built according to the sub individual, and it is trained to obtain the *acc* value. Generating the next population through the operations of initialization, mutation, crossover and selection with DE algorithm. Finally, select the best CNN from the current generation to complete the optimization of network hyper parameter. In the research process, we found a new mutation strategy to generate mutation solutions, so that DE-CNN can obtain better experimental results than the previous generation through iterative training. When the termination condition is satisfied, the optimal individual is generated by DE-CNN. The experiment shows the results of the DE algorithm in the iterative process of CNN optimization and proves that the DE-CNN algorithm's compilation strategy which generate a mutation solution vector is an effective solution for optimizing the network structure. The experiment also compares the classification accuracy values of DE-CNN, GoogLeNet, VGGNet16, ResNet34, DenseNet etc algorithm that are trained by medical image data set, and the accuracy value of DE-CNN is obviously higher.

II. MATERIALS AND METHOD

A. Convolution Neural Network

In 2010, Convolution neural network was come up by Yan Lecun [8]. They established CNN architecture of Lenet-5, which was applied to postal code identification of post offices. In 2012, the CNN achieved its first breakthrough, AlexNet appeared. It was developed by Alex Krizhevsky and Jeffry Hinton and is widely used with relatively simple network structure. In the ImageNet challenge of 2014, the visual geometry team of Oxford University built VGGNet and won the second place in the

competition. After further study, the model has been developed into two versions VGG-16 and VGG-19. The same year, Google launched its own algorithm GoogLeNet, which performance better than VGGNet. The general rule of CNN is that the more convolution layers, the easier it is to over fitting in the training stage. However, in repeated exploration, there searchers found that the deeper with network structure, the smaller loss value about it. So GoogLeNet replaced the fully connected layer with inception module and GAP to prevent over fitting. In the 2015 Image Net ILSVRC Challenge, the winning network was ResNet from Microsoft, and the error was just 3.7%. They can build the ResNet structure at different depths with 34, 50, 101 and 152 layers. These are deeper than the previously proposed network structure, the main idea is to skip multiple layers by adding links to the neural network.

CNN structure mainly comprises the convolution layer, pooling layer, full connection layer, and output layer. The convolution layer obtains local important information through feature extraction, and the feature information of convolution layers with different depths is different. The fully connected layer is classified according to the local information with category differentiation extracted from the convolution layer. In order to build a CNN with an optimal structure, researchers often use the optimizer to optimize the network structure's weight and offset to minimize the error rate and maximize the accuracy. Classification function after layer is used to solve nonlinear classification problem. We adjust the hyper parameters on the CNN to achieve unsupervised learning of the model, so that the classification result is closer to the real label.

Convolution layer is used for feature extraction of input image. Each pixel of the output feature map of the convolution layer is mapped by a certain rule from the part of the input image, all pixels of the feature map are formed according to this rule. In convolution layers, weight links are shared between nodes to facilitate the quest for the same local feature for input channels. Therefore, the input image is extracted into n feature maps, through n convolution kernels of the convolution layer. The following function shows how it works [9]:

$$O_{i,x,y} = \sum_{j=1}^N \sum_{s=1}^W \sum_{t=1}^H I_{j,x+s,y+t} k_{i,j,s,t} \quad (1)$$

Assuming $I = \{I_1, I_2, \dots, I_N\}$ is the input feature images, $K = \{K_1, K_2, \dots, K_M\}$ is the convolution kernels, $O = \{O_1, O_2, \dots, O_M\}$ is the output feature images, and M means the number of kernels. The size of K_i is $N*W*H$, among this, N denotes the number of channels for kernels, W denotes the width of kernels, H represents the height of kernels. When kernel scans the feature image, x and y represent step values for width and height, respectively.

The pooling layer is another important part of the CNN, which mainly compresses feature images. The pooling layer compresses the pixels of the feature map in a certain rule. Its working principle is that CNN's filter scans part of the

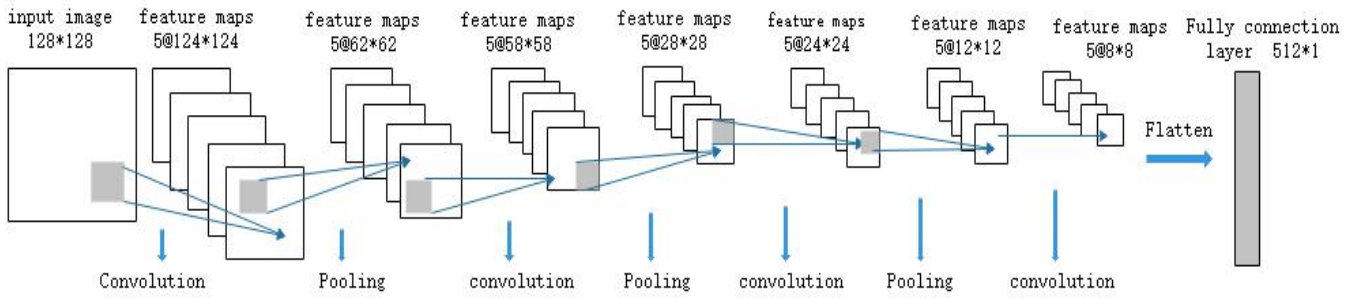


Fig. 1. the basic structure of CNN.

feature map, pooling function selects the maximum value, minimum value or average value according to the pooling rules and maps the value to a pixel of the output feature map. When the filter scans the whole feature map according to a certain step size, the convolution neural network pooling layer derives the dimensionality reduction feature map.

The fully connection layers follow several layers of convolution layers and pooling layers. The characteristic of the fully connection layer is to connect all the neurons of the next layer one by one [10]. First, the feature map needs to go through the convolution layer and the pooling layer to get many nodes with local feature information after the flattening operation, these nodes become the input of the fully connected layer. Finally, the probability values of the characteristic nodes are got through the classification operation of the full connection layer.

In back-propagation optimization, dropout layers are a simple operation to prevent over fitting of the CNN. Some neuron nodes in CNN are temporarily removed with probability P , other neuron nodes are retained with probability $1-P$. Therefore, the probability that each neuron is temporarily turned off is the same. The dropout method can compress the neural network model by reducing the number of parameters that need to be optimized during the back-propagation operation. This method can effectively improve the training speed of the network and prevent the model from over fitting.

Finally, the classification operation as the output layer is implemented based on a non-linear classification function [11]. In the classification layer, using the nonlinear classification function SVM or softmax to solve the discrete classification problem. The fundamental working principle of CNN optimizer is to compute the gradient value according to loss value and back propagation. In order to improve the performance of CNN model, this paper mainly uses SGD optimizer to update the connection weight according to the loss value and gradient value in back propagation. The above is the basic composition and principle of CNN. Refer to figure 1 above for the basic structure of CNN.

Activation function is usually added after the convolution layer and the fully connection layer on CNN. Introducing the activation function can enhance the expression ability of deep neural network. Thus, when designing CNN algorithm, the choice of activation function is very important [12]. *Relu* is a continuous nonlinear activation function [13]. We introduce the *Relu* activation function expression below:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

When the *Relu* input factor is less than 0 and the output is 0, when the *Relu* input factor is greater than or equal to 0, the input and output are equal.

Sigmoid is a logistic regression function. The function expression is introduced below:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

The continuous nonlinear activation function *Sigmoid* compresses any input x ($x \in \mathbb{R}$) into a $x \in (0, 1)$ interval.

Tanh activation function is a hyperbolic tangent function. Its input and output can maintain a non-linear rise and fall relationship, it has good fault tolerance. The function expression is as follows:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

The input x ($x \in \mathbb{R}$) of the *tanh* function is set between $x \in (-1, 1)$. Compared with *sigmoid* function, *tanh* can better ease gradient disappearance when applied to the algorithm.

B. Differential Evolution

Differential evolution algorithm is a global search algorithm that uses global and local information to provide the population and then select the best individual [14]. The algorithm needs to define mutation operator F , crossover operator CR , evolutionary generation number G and population size np , its process consists of four steps. The following vector parameter can present all individuals of each population:

$$X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\} (i=1, 2, \dots, np) \quad (5)$$

Where i represents individual number, D is the dimension of each individual in the population. The first part is the initialization of the population and the following formula is used to initialize the individual:

$$x_{i,j} = \text{rand}[0,1] \cdot (x_j^{(U)} - x_j^{(L)}) + x_j^{(L)} \quad (6)$$

among $(i=1, 2, \dots, np; j=1, 2, \dots, D)$

Where $x_j^{(U)}$ denotes upper bound of element $x_{i,j}$ and $x_j^{(L)}$ represents lower bound of element $x_{i,j}$. The population can

be expressed as: $P = \{X_1, X_2, \dots, X_{np}\}$. The second part of DE is mutation, the following formula is used for target population mutation:

$$V_i = X_{r_1} + F \cdot (X_{r_2} - X_{r_3}) \quad (7)$$

Where r_1, r_2, r_3 are random indexes in the range of $[1, NP]$ and F is the scaling factor in the range of $[0,1]$ which controls $(X_{r_2}-X_{r_3})$. The third part of DE is crossover which role is increasing the disturbance of population and enhance the diversity of parameters. Following formula creates a new vector, this vector is the trial vector $u_{i,j}$, the element of $u_{i,j}$ is generated as follows [15]:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \{(rand(j) \leq CR) \text{ or } (j = randn(j))\} \\ x_{i,j}, & \text{if } \{(rand(j) > CR) \text{ or } (j \neq randn(j))\} \end{cases} \quad (8)$$

Where j is the index of the element for any vector, $rand(j)$ is a random number in the range of $[1, D]$ and CR is the crossover factor in the range $[0, 1]$.

The fourth part of DE is using adaptive value function to select individuals, the principle of this operation is comparing adaptive function values of each individual. The fitness of the individual is then calculated by $fit(X_i)$, program select individuals with high fitness values and pass them to next generation, ensure that the fitness value of individuals in the next generation are better than that of the current generation.

Algorithm 1 The DE algorithm

```

1: Initialize population randomly.
2: Evaluate the fitness of each individual to decide the best individual  $X_{best}$ .
3: While terminal condition not met do
4:   for each individual  $i(i=1,2,\dots,np)$  do
5:     generate the mutation individual  $V_i$  with equation (7);
6:     generate the trail vector  $U_i$  with equation (8);
7:     selection criterion  $X_{best}$ ;
8:   end for
9: end while
    
```

Fig. 2. DE algorithm pseudo code.

C. Related Work

Our paper designs and implements the DE-CNN algorithm to classify medical images. In this research, the second part introduces the basic principles and structure of the CNN and differential evolution algorithm. The third part explains in detail how to use the differential evolution algorithm to automatically optimize network structure of the convolution neural network. In the fourth part, an experimental test was performed to determine whether there is pneumonia in the lungs. The pneumonia images of the chest X-Ray data set and the collected pneumonia images caused by the new corona virus were used as the experimental data set. The main difference between pneumonia image and normal image is explained. Models are trained on the Lung X-Ray image data set and the performance of the model is tested. The experiment compares the classification results of DE-CNN with other

deep learning algorithms. The fifth part summarizes the key process of DE-CNN algorithm optimizing the structure of CNN and the superiority of DE-CNN shown by the experimental results.

III. THE PROPOSED ALGORITHM

This part mainly discusses the flow of DE-CNN algorithm and the principle of DE optimizing the structure of CNN. The purpose of this method is to find suitable network structure for lung image classification. The DE-CNN framework we proposed comprises four stages: initial population, mutation, crossover and selection. DE-CNN first generates a random population P with np individuals, the dimension of each individual is D . D represents the 7 hyper parameters that make up CNN, as shown in table I. DE-CNN designs the structure of CNN according to the basic elements of individuals in the population. In DE-CNN, a new population P_{G+1} is generated using selection, mutation and crossover operations. These operations are employed to generate mutation individual solution and crossover individual solution. Because the structural optimization of CNN is a discrete problem, the mutation and crossover operation of DE is often used to solve continuous problem. So, a new mutation strategy is put forward to generate the more likely best individual. Then, the selection scheme is applied to determine the surviving individual for the next generation. The termination of DE-CNN is that the number of iterations of the population is equal to the evolutionary generation number G .

A. The initialization of population

The hyper parameters of building CNN are as follows: the number of kernels, the size of kernel, activation and so on. Each epoch of training data set of CNN model will output loss value and accuracy value. Before the CNN starts running, these hyper parameters that build a CNN are arranged and combined to form an individual X_i , until the number of individuals meet the population's size np . These basic parameters make up all the elements of an individual, expressed as $x_{i,j}$. The values of D components are distributed as follows:

TABLE I
TABLE OF INDIVIDUAL BASE ELEMENTS

the kind of base elements	Several values
The number of kernels	[16, 32, 64, 128, 256]
Kernel size	[1, 3, 5, 7, 9]
activation	[relu, sigmoid, tanh]
Drop out	[0.1, 0.2, 0.3, 0.4, 0.5]
Learning rate	[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 0.03]
Loss_initial	0.0
Accuracy_initial	0.0

Basic elements can be added to the above table according to the actual situation. The initialization of population is to initialize all individuals, and the initialization of each individual is to initialize all components. The D components

of an individual are determined by D different basic elements in the above table. Each component has a corresponding basic element. Population initialization is shown as follows:

$$X_i = \{X_1, X_2, \dots, X_{np}\} \quad (9)$$

Where np is the size of the population and X_i denotes solution. Then the individual of the K th solution can be expressed as: $X_k = \{x_{k,1}, x_{k,2}, \dots, x_{k,D}\}$, D represents the number of components in an individual.

B. Mutation

Mutation operation is to pass better individuals to the next generation and accept poor solutions. The mutation stage is according to the original DE to propose a new mutation strategy. The difference from DE is that a new individual is generated by mutation between the two best solution individuals of current generation and the randomly generated solution individual. We set up mutation operator $mr=0.3$, which is a real constant factor and ranges between $[0, 2]$.

Before the mutation operation starting, it is necessary to run the CNN network structure of each individual in the population to obtain the acc value. When the operate of mutation is running.

X_{random} is a solution randomly select from current population which excluded the best two solutions. X_{random} provides random solution for mutation operations, gives high-quality elements of that may be inferior solution the opportunity to form suitable solution, increases the perturbation of the next population. Based on the two optimal solutions X_{best1} , X_{best2} of current generation and a random solution X_{random} , individual V_i of the mutation solution is generated by formula (10), (11) and (12). The following formula expresses the mutation:

$$V_i = X_{random} + F(\alpha)(X_{best1} \odot X_{best2}) \alpha \in (0, 1) \quad (10)$$

$$\phi(j) = X_{best1} \odot X_{best2} = \begin{cases} \text{None, if } x_{best1,j} \neq x_{best2,j} \\ x_{best1,j}, \text{ otherwise} \end{cases} \quad (11)$$

$$F(\alpha) = \begin{cases} 1, \alpha < mr \\ 0, \alpha \geq mr \end{cases} \quad (12)$$

Where $\alpha \in (0, 1)$ represents the random real numbers generated between $(0, 1)$, mr denotes the mutation operator with 0.3 , X_{best1} and X_{best2} are two best solutions for G th generation. In the formula, $j=1, 2, \dots, D$, $\Phi(j) = \{\Phi(1), \Phi(2), \dots, \Phi(D)\}$ represent the intermediate solution. Specific mutation steps are as follows:

Step 1. In the current population, DE-CNN selects two solutions X_{best1} , X_{best2} with the best accuracy value, randomly selects a solution individual X_{random} .

Step 2. Mutation operation judges the relationship between mr and α . If the mutation operator $mr \geq \alpha$, then the mutation individual $V_i = X_{random}$, if the mutation operator $mr < \alpha$, then perform the following operations:

- a. If $x_{best1,j} \neq x_{best2,j}$, $\Phi(j) = \text{None}$, $v_{i,j} = x_{random,j}$.

- b. If $x_{best1,j} = x_{best2,j}$, $\Phi(j) = x_{best1,j}$, $v_{i,j} = x_{best1,j}$.

- c. $V_i = x_{random,j} \cup \Phi(j)$, $j \in (0, D-2)$.

- d. $j = j+1$.

Step 3. The termination condition of mutation operation is determined by the relationship between j and $D-2$. If dimension index value $j \leq D-2$, the program returns to Step 2. If $rand \geq \alpha$ and $j > D-2$, the program output V_i .

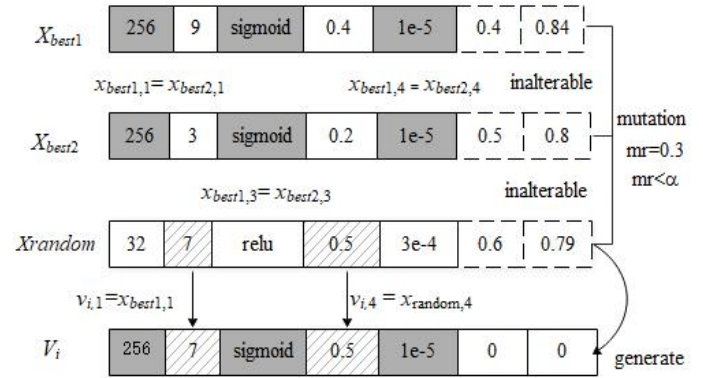


Fig. 3. Generation of the mutation solution V_i .

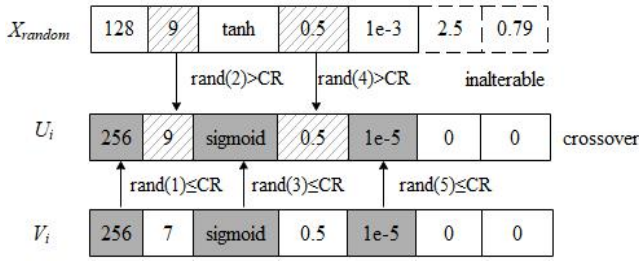
The figure above shows the mutation operation that meets the mutation conditions. Among them, x_{best1} , x_{best2} represent the optimal solution individuals of the current generation and X_{random} represents randomly selected solution individual. In the gray area of x_{best1} and x_{best2} , the basic elements are the same and assigned to V_i . The basic elements of colorless area are different. Select the element corresponding to the position of x_{random} and assign the value to V_i .

C. Crossover

DE-CNN: Crossover operator can produce a new individual in the next generation. X_{random} is the individual solution selected from the population of previous generation. The random selection range should exclude the random solution of mutation operation and the best two solutions. Crossover operation is to generate an offspring individual through the crossover of random solution X_{random} and mutation solution V_i , then according to the formula (13) forms a new individual $U_i = \{u_{i,1}, u_{i,2}, \dots, u_{i,D}\}$ of the cross solution through specific operation. Its principle is as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, \text{ if } rand(j) \leq CR; j = 1, 2, \dots, D-2 \\ x_{random,j}, \text{ otherwise} \end{cases} \quad (13)$$

The crossover operation needs to traverse the first $np-2$ elements of X_{random} and V_i . In this paper, np is a positive integer 6, j records the traversal position. $rand(j)$ represents a real number randomly selected from the interval $(0, 1)$, the number of random operations for $rand(j)$ is equal to $np-2$. CR is a real number randomly selected from $[0, 1]$, if the random number $rand(j) \leq CR$ (crossover factor), the j th element of U_i $u_{i,j}$ is assigned value $v_{i,j}$; If the random number $rand(j) > CR$, $u_{i,j}$ is assigned value $x_{random,j}$. After the algorithm has completed the operation of the above steps, the cross individual solution is generated.


 Fig. 4. The experimental solution U_i is generated.

The above figure shows the generation process of cross individual. The gray part of individual V_i indicates that the random value under the current component is less than or equal to the crossover factor, $u_{i,j} = v_{i,j}$;

The gray slash part indicates that the random number under the current component is greater than the crossover factor, $u_{i,j} = x_{random,j}$.

D. Selection

Take the algorithm of CNN as the aim function of DE-CNN. Sort the individuals in the population in reverse order according to the high accuracy value. Take the first half of the individual population solution, and inherit the mutation solution and crossover solution to the next generation population. The main steps of the algorithm DE-CNN are as follows:

Step 1. Initialize the population P , randomly code each individual $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}$ from the table I which contains the 7 hyper parameters of the CNN network. Then form the initial population $P = \{X_1, X_2, \dots, X_{np}\}$ which size is np . The individual's dimension defined as D , its value is consistent with the number of the types of hyper parameters. Initializing hyper parameters of DE-CNN, including evolutionary generation number $noGen$ of the population, the dimension D of the individuals, the mutation operator mr and the crossover factor CR of the DE algorithm; Define the termination conditions;

Step 2. CNN built by individuals $\{X_1, X_2, \dots, X_{np}\}$ of the image data set training population P , all individuals of the population obtain the training data acc and $loss$ value, assign them to the basic elements $x_{i,D-1}$ and $x_{i,D}$ of each individual;

Step 3. According to the training data of each individual and the maximum acc value, program selects half of the excellent individuals in the population to form a set $\{X_{best1}, X_{best2}, \dots, X_{best(np/2)}\}$, and then select two from the excellent individuals as the best individuals X_{best1} and X_{best2} ;

Step 4. Algorithm performs mutation operation according to formula (10-12). The index range of X_{best1} and X_{best2} are $[0, D-2]$, which is called the index number. If the mutation condition is satisfied, it is $mr < \alpha$. The basic elements of the two optimal solutions are traversed according to the index range, and the basic elements of the two optimal individuals in the same index are compared. If $x_{best1,j} = x_{best2,j}$, then $v_{i,j} = x_{best1,j}$; If $x_{best1,j} \neq x_{best2,j}$, $v_{i,j} = x_{random,j}$. If the mutation condition is not satisfied, i.e. it is $mr \geq \alpha$, then $V_i = X_{random}$. The mutation solution is $V_i = \{v_1, v_2, \dots, 0\}$;

Step 5. DE-CNN performs crossover operation according

to formula (13) to generate individual U_i . Initialize random individual U_i and randomly selected solution X_{random} . Traverse each basic element $u_{i,j}$ of U_i , if $rand(j) \leq CR$, $u_{i,j} = v_{i,j}$; otherwise, $u_{i,j} = x_{random,j}$;

Step 6. The cross-solving individual U_i , the mutation individual V_i and the half of the excellent individuals in the current population form the next generation population P_{G+1} ;

Step 7. Judge whether the program meets the termination condition. If the traversal generation number $G < noGen$, the procedure return to the second step; If $G \geq noGen$, the program outputs the optimal individual and saves the corresponding CNN network model.

Step 8. Load the model of individuals with the best training performance of the last generation, and test the model with the test data set.

The following pseudo code shows how DE is combined with CNN.

DE-CNN Algorithm

```

1: Initial parameters: Population:  $P$ ; Dimension:  $D$ ; Population
size:  $NP$ ; Mutation:  $F$ ; Crossover:  $CR$ ; Generation:  $noGen$ ;
Individual:  $p$  or  $X_i, G$ ; Basic element:  $x_{i,j}$ .
2: Let CNN algorithm be the given objective function;
3: Initialize the population: initialize with formula (9) and the
principle of subsection a in part III.
4: for  $i=1$  to  $NP$  do
5:   for  $j=1$  to  $D$  do
6:      $x_{i,j} = random(\text{num kernels}[], \text{activation}[], \dots, \text{Acc}[])$ ;
7:   end;
8: end;
9:  $G=1$ ;
10: while  $G < noGen$  do
11:   (training CNN function)
12:   for  $p$  in Population  $NP$  do
13:     CNN model{
14:       Conv2D(filters= $p_0/2$ , kernel_size=( $p_1, p_1$ ));
15:       Activation( $p_2$ );
16:       Maxpooling2D(pooling_size=(2,2));
17:       Dropout( $p_3$ );
18:       Dense(500);
19:       Dense(2);
20:       Activation('softmax');
21:       compile( $p_4$ );
22:        $p_5, p_6 = model.fit()$ ;
23:        $p.append(p_5, p_6)$ ;
24:     end;
25:   (selection function)
26:   for  $i=1$  to  $NP$  do
27:     Sort in ascending order by acc CNN( $X_i, G$ );
28:      $X_{best1} = CNN(X_1, G)$ ;
29:      $X_{best2} = CNN(X_2, G)$ ;
30:   end;
31:   (Mutation and Crossover functions)
32:   for  $j=1$  to  $D$  do
33:      $v_{i,j}(G) = Mutation(X_{best1}, X_{best2}, X_{random1})$ ;
34:      $u_{i,j}(G) = Crossover(X_{best1}, v_{i,j}(G), X_{random1})$ ;
35:   end;
36:   Refactoring populations based on The best individuals,
37:    $u_{i,j}(G)$  and  $v_{i,j}(G)$ ;
38:    $G = G+1$ ;
39: end while;
    
```

Fig. 5. DE-CNN pseudo code.

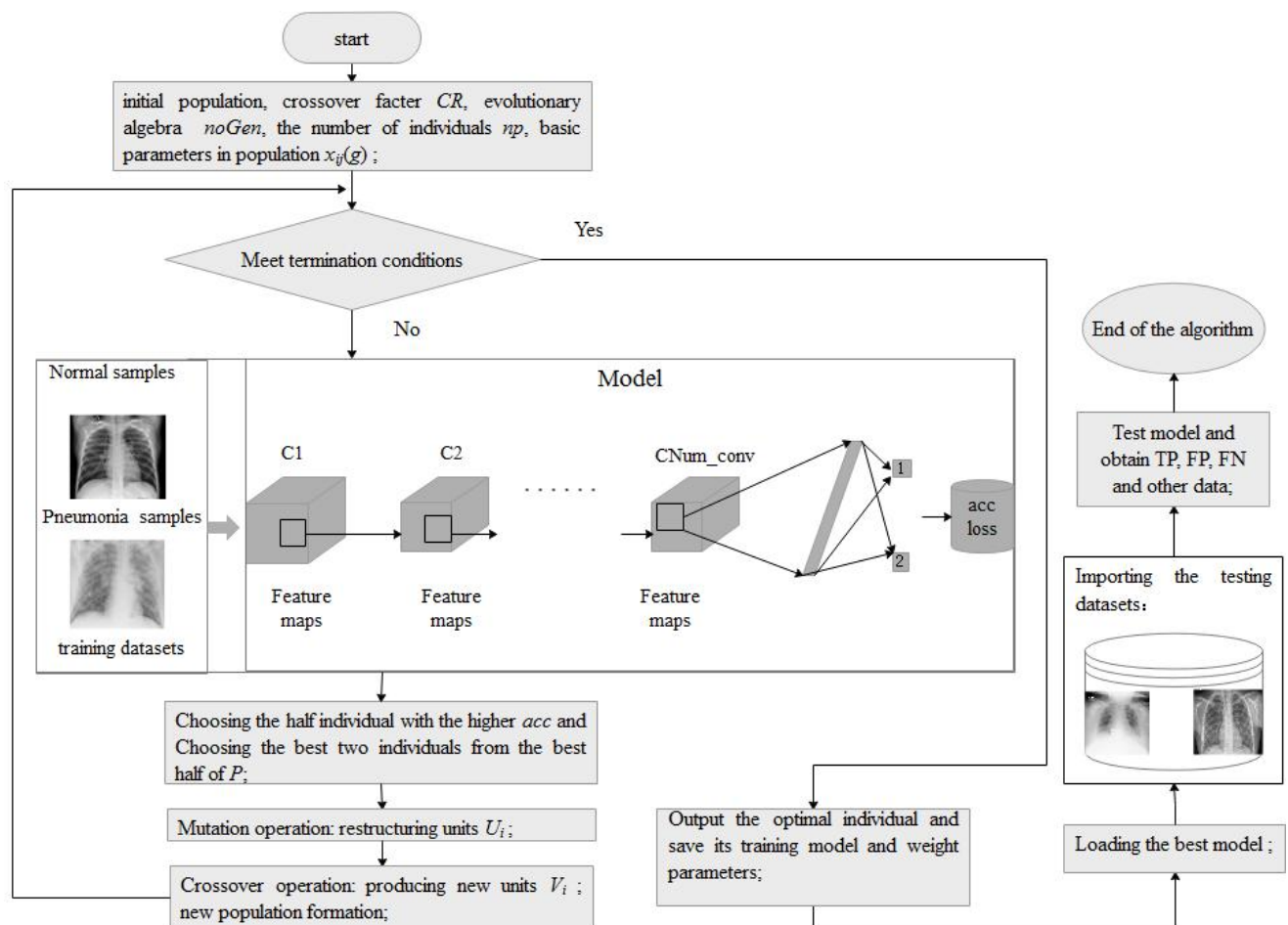


Fig. 6. Model DE-CNN working principle diagram.

IV. EXPERIMENT

A. Experiment environment

Lung X-Ray image data set, mainly includes lung images of patients with pneumonia and normal people. It consists of covid-19 data and chest X-Ray data set. Its training dataset contains 879 images totally, 461 pneumonia images and 418 normal images. Test dataset has 500 images, 416 pneumonia images and 84 normal images. In fig.7, the left picture is an image of pneumonia, and the right image is a normal lung. The left image has either multiple lentil-sized air inlets measuring a few millimeters in diameter or a tree-shaped structure [16]. In contrast, in the right image, there are no lentil-sized air inlets in the lung cavity and the outline of the ribs is clear. CNN can extract pixel values, contour features and deep features to distinguish the two images.

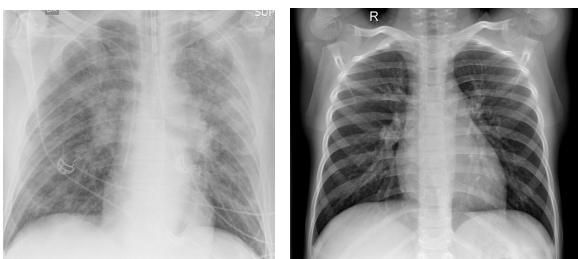


Fig. 7. X-Ray image of lungs.

Initialize the parameters at the beginning of the experiment, set the population size np to 6 and the individual dimension D to 7. The first five elements of an individual are composed of CNN hyper parameters. These five elements form a CNN network, the network's training results are the last two elements. The experiment sets the conventional hyper parameters for the CNN algorithm, changes the number of feature extraction layers of DE-CNN, and runs on the same data set.

The number of feature extraction layers of DE-CNN framework is different, the features information extracted from the X-Ray image are also different, and the accuracy value and loss value of the classifier are biased. The structure of DE-CNN is different in CNN depth settings, when they train the same data set, the experimental results are different, as shown in Table II.

TABLE II
COMPARISON OF DIFFERENT CONVOLUTION LAYERS OF THE DE-CNN ALGORITHM

Data set	Neural Network	Performance			
		Min-acc	Mean-acc	Min-loss	Mean-loss
Lung X-Ray image	DE-CNN -1conv	0.5272	0.9247	1.515	83.3652
	DE-CNN -2conv	0.6	0.9456	0.164	5.8054
	DE-CNN -3conv	0.4854	0.8253	0.2766	1.1742

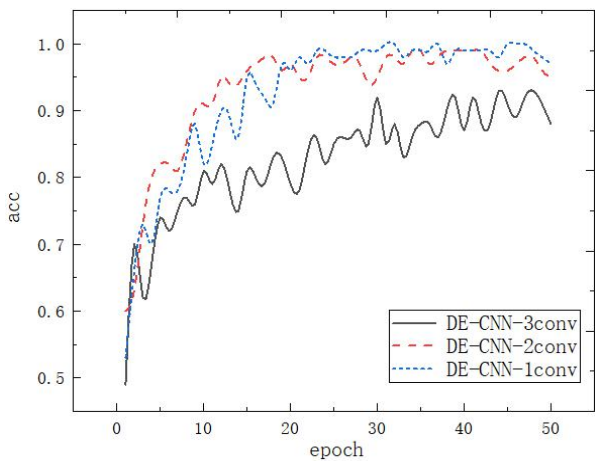


Fig. 8. Compare the influence of the number of convolution layers.

This dataset is applicable to DE-CNN-2conv, as can be seen from the fig.8, DE-CNN-2conv has the fastest convergence speed, is more suitable for small data set.

The number of feature extraction layers is determined according to the above experiment, the DE-CNN program keep running until the termination conditions are met to complete the hyper parameters optimization of CNN. According to the DE-CNN training results, the optimal CNN structure of each generation was selected and saved for subsequent testing.

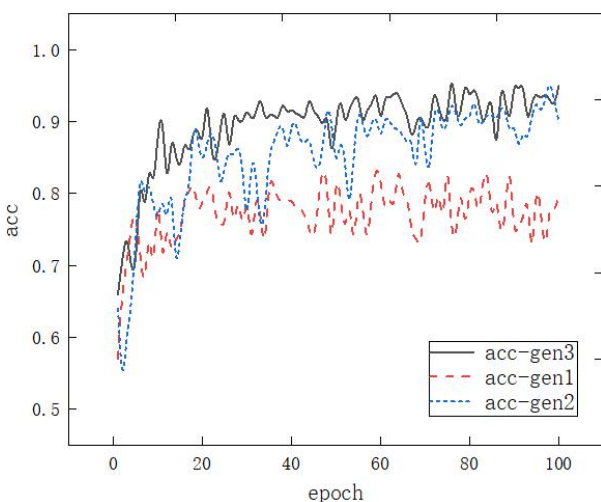


Fig. 9. Comparison of train for gen1, gen2 and gen3.

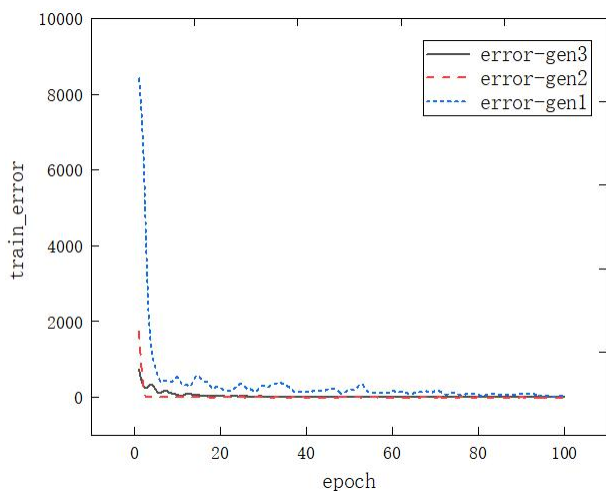


Fig. 10. Comparison of train for gen1, gen2, gen3.

Fig.9 shows how training *acc* value of the best individual of each generation of DE-CNN varies with epoch. Fig.10 shows the change process of loss value during DE-CNN evolution.

It can be seen from the Fig.9 and Fig.10 that when the CNN optimized by DE algorithm in DE-CNN structure classified images, with the increase of evolutionary generation number, the accuracy of training is improved and the error is reduce. The *acc* value and the *loss* value showed by the best individual CNN in the third generation are best.

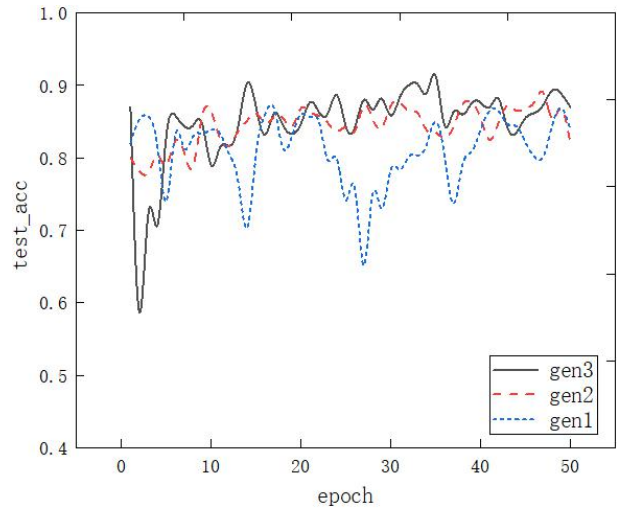


Fig. 11. Comparison of test for gen1, gen2 and gen3.

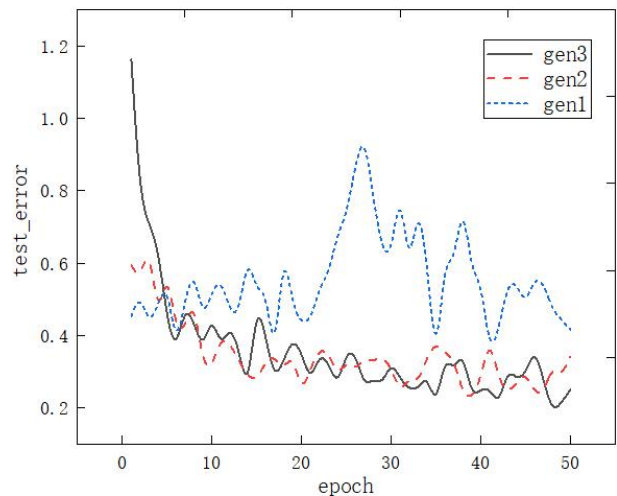


Fig. 12. Comparison of test for gen1, gen2 and gen3.

Fig.11 shows the situation that the *acc* value of the best individuals in each generation of DE-CNN changes with the number of epochs. Fig.12 shows a line diagram of the error value varying with the number of cycles.

On the basis of the training of the model, DE-CNN selects the model with the best training result from each generation. We selected 1/5 images from the original data set to test the accuracy and loss value of the best model. From the above experiments, it can be seen that the changes of test results are similar to the changes of training, and the result gradually converges after 50 epochs. And selecting the best CNN in the third generation can better solve the problem of lung image classification. The hyper parameter combination of this CNN is: the number of convolution

kernels is 32, the size of convolution kernel is 7, the activation function is *tanh*, the dropout rate is 0.2, and the learning rate is 0.0001.

TABLE III
COMPARISON OF COMMON NETWORK FRAMEWORKS AND THIS ALGORITHM

Data set	Neural Network	Performance		
		acc	#parameters	#epochs
Lung X-Ray image	GoogLenet	0.832	101M	40
	ResNet34	0.836	60.5M	60
	VGGNet16	0.832	26M	50
	DenseNet	0.689	27.2M	50
	DE-CNN	0.9165	64.7M	100
	MobileNet	0.800	5.1M	50
	AlexNet	0.816	55.6M	50
	LeNet	0.712	12.3M	50

With the same experimental environment, GoogLenet, Resnet34, DenseNet, MobileNet, AlexNet, LeNet and VGGNet16 were trained under the drive of Lung X-Ray image data set. The experiment results of the optimal individual of third generation of DE-CNN are the best.

TABLE IV
CROSS VALIDATION DE-CNN ALGORITHM ON THE LUNG X-RAY IMAGE DATA SET

Data count	train_acc	train_loss	Val_acc	Val_loss
First fold data	0.95	0.3284	0.85	0.3896
Second fold data	0.9125	0.5153	0.825	0.4829
Third fold data	0.8375	0.7262	0.9545	0.4793
Forth fold data	0.95	0.2332	0.925	0.4201
Fifth fold data	0.975	0.3173	0.825	0.4896

The Lung X-Ray image data set is divided into five parts on average, and the 5 fold cross verification is done. According to the data in Table IV, the DE-CNN algorithm has stable training results and verification results. The experiment proves that the classification effect of DE-CNN has stability and reliability.

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

$$\text{precision} = \frac{TP}{TP + FP} \quad (15)$$

$$\text{F1-score} = \frac{2TP}{2TP + FP + FN} \quad (16)$$

TP means that the image with pneumonia features is correctly classified as having pneumonia. FN means that an image with pneumonia features, but the algorithm does not found it. FP means that an image is mistakenly detected to have pneumonia. We used the following methods to test the performance of the model.

TABLE V
THE PRESENTATION OF RECALL AND PRECISION OF COMMON CNN AND DE-CNN

Method	Pneumonia			Normal		
	recall	precision	F1	recall	precision	F1
Googlenet	1	0.37	0.54	0	0	0
Dense	0.25	0.89	0.39	0.91	0.29	0.44
VGG16	0.64	0.5	0.56	0.036	0.0625	0.46
DE-CNN	0.8	0.64	0.71	0.61	0.73	0.66

In the above experiments, the threshold value was set to 0.5, and different CNN structures were used to test the pneumonia image and normal lung image, the predictive score and threshold value of the image test were compared. Then, the image is judged as TP, FP or FN according to the threshold and the real label. To calculate the information in the table according to formula (14,15,16). From the above table, it can be concluded that the precision and recall of DE-CNN are relatively stable, and F1's index is the highest.

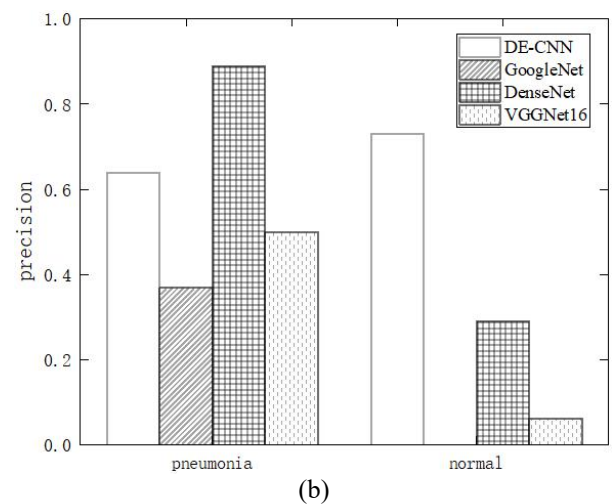
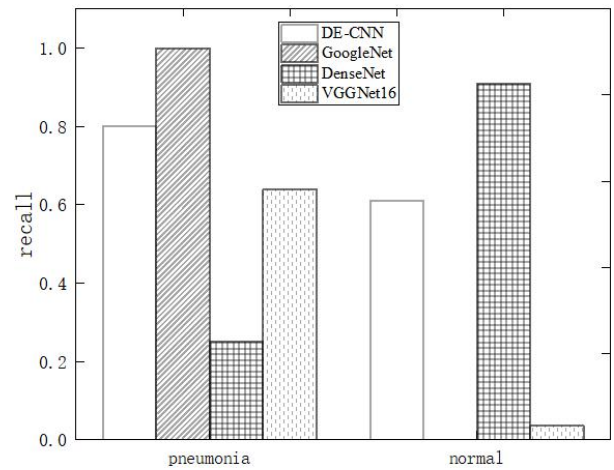


Fig. 13. The classification results compare our DE-CNN method with GoogleNet, DenseNet and VGGNet16: (a) recall and (b) precision.

DE-CNN can automatically design the optimal CNN structure and extract discriminant features without manual feature design, which achieved good benchmark performance. The algorithm has a relatively strong

robustness in the classification of lung pneumonia lesions.

V. CONCLUSIONS

In this research, a model DE-CNN that automatically optimizes the structure and hyper parameters of the CNN is proposed, which reduces the optimization time of the classification network CNN when solving image classification problems and improves the accuracy of the algorithm classification. Each individual solution of the population constructs a corresponding CNN net, and the optimal individual is selected according to the accurate value of the algorithm training data set. With the population iteration, DE-CNN combined with the new mutation strategy can perform progressively better experimental results in the experimental operation and testing stages. By analyzing the training results and the trend of test results of DE-CNN algorithm, the global optimal network structure is found. The classification accuracy of DE-CNN algorithm, GoogLenet, VGGnet16, Resnet34, Densenet, Mobilenet, Alexnet and Lenet algorithm based on medical image dataset training, indicating the accuracy of DE-CNN is significantly higher. The deficiency of this paper is that it did not take into account the impact of different optimizers on CNN network performance. Researchers can add the types of optimizers to the basic element table for experiments.

REFERENCES

- [1] F. Bray, J. Ferlay, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol.70, pp. 313-313, 2018.
- [2] T. K. Das, C. C. Lal, X. Z. Gao, "Chest X-Ray Investigation: A Convolutional Neural Network Approach," *Journal of Biomimetics, Biomaterials and Biomedical Engineering*, vol. 45, pp. 57-70, 2020.
- [3] Y. X. Dong, Y. C. Pan, J. Zhang, W. Xu, "Learning to read chest X-ray images from 16000+ examples using CNN," *Connected Health*, pp.51-57, 2017.
- [4] S. Bharati, P. Podder, et al. "Hybrid deep learning for detecting lung diseases from X-ray images," *Informatics in Medicine Unlocked*, vol. 20, 2020.
- [5] W. X. Lu, H. D. Rui, C. Y. Liang, "A Method Based on GA-CNN-LSTM for Daily Tourist Flow Prediction at Scenic Spots," *journal*, vol. 22, pp. 261, 2020.
- [6] B. Wang, Y. N. Sun, B. Xue, M. J. Zhang, "A Hybrid Differential Evolution Approach to Designing Deep Convolutional Neural Networks for Image Classification," *Springer Verlag*, pp. 237-250, 2018.
- [7] A. Dahou, M. A. Elaziz, "Arabic Sentiment Classification Using Convolutional Neural Network and Differential Evolution Algorithm." *Computational Intelligence and Neuroscience*, vol. 2019, pp. 25-31, 2019.
- [8] G. Silva, T. Valente, A. C. Silva, A. D. Paiva, "Convolutional neural network-based pso for lung nodule false positive reduction on ct images," *Computer Methods & Programs in Biomedicine*, vol. 162, pp. 109-118, 2018.
- [9] H. Chung, K. S. Shin, "Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction." *Neural Computing and Applications*, vol. 32, pp. 1-18, 2020.
- [10] W. Z. Sun, J. S. Wang, B. W. Zheng, and Z. F. Li, "A Novel Convolutional Neural Network Voiceprint Recognition Method Based on Improved Pooling Method and Dropout Idea," *IAENG International Journal of Computer Science*, vol. 48, no.1, pp. 202-212, 2021.
- [11] B. X. Guan, Y. H. Zhao, Y. Yin, "A differential evolution-based feature combination selection algorithm for high-dimensional data," *Information Science*, vol. 547, pp. 870-886, 2021.
- [12] D. Y. Li, D. Yang, J. Zhang, and X. D. Zhang, "AR-ANN: Incorporating Association Rule Mining in Artificial Neural Network for Thyroid Disease Knowledge Discovery and Diagnosis," *IAENG International Journal of Computer Science*, vol. 47, no.1, pp. 25-36, 2020.
- [13] M. Alkhaleefah, C. C. Wu, "A Hybrid CNN and RBF-Based SVM Approach for Breast Cancer Classification in Mammograms," *Institute of Electrical and Electronics Engineers Inc., United States*, pp. 894-899, 2018.
- [14] H. Yong, J. Huang, X. Hua, "Gradient Centralization: A New Optimization Technique for Deep Neural Networks," *Springer*, vol. 17, pp. 635-652, 2020.
- [15] T. Hamdi, J. B. Ali, V. D. Costanzo, "Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm," *Biocybernetics and Biomedical Engineering*, vol. 38, pp. 362-372, 2018.
- [16] A. Reissig, C. Kroegel, "Diagnosis of Pulmonary Embolism and Pneumonia Using Transthoracic Sonography," *Progress in Respiratory Research*, vol. 37, pp. 43-50, 2009.