IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

SPECIAL SECTION ON DEEP LEARNING
ALGORITHMS FOR INTERNET OF MEDICAL THINGS

Received April 29, 2020, accepted May 13, 2020, date of publication May 22, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2996569

Multiple Lesions Detection of Fundus Images Based on Convolution Neural Network Algorithm With Improved SFLA

WEIPING DING, (Senior Member, IEEE), YING SUN, LONGJIE REN, HENGRONG JU, ZHIHAO FENG, AND MING LI
School of Information Science and Technology, Nantong University, Nantong 226019, China

Corresponding author: Weiping Ding (dwp9988@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976120, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191445, in part by the Six Talent Peaks Project of Jiangsu Province under Grant XYDXXJS-048, and in part by the Qing Lan Project of Jiangsu Province.

ABSTRACT In order to effectively solve the problem of interlaced overlap in the fundus image lesions, large and small blood vessels packed densely and severely affected by light, and to achieve multi-label classification of fundus images. In this paper, a single population leapfrog optimization convolutional neural network algorithm (SFCNN) is proposed to detect and classify various fundus lesions. The algorithm uses the efficient search ability of the shuffled frog leaping algorithm to optimize the weight initialization and back propagation of the convolutional neural network. In order to deal with the problem of fundus image classification in the big data environment, the novel grouping optimization strategy is presented to effectively combine Spark platform and SFCNN algorithm to achieve large-scale fundus image classification and detection of multiple lesions. The experiment of the detection of fundus image lesions shows that the accuracy rate of SFCNN is better improved in both single lesion detection and overall detection, compared with other algorithms.

INDEX TERMS Shuffled frog leaping algorithm, convolutional neural networks, fundus image, detection of multiple lesions, weight optimization.

I. INTRODUCTION

Color fundus images are the most basic way to diagnose eye diseases [1]. At the same time, Fundus image can make people discover various eye diseases as early as possible, such as glaucoma, optic neuritis, macular degeneration, etc. This can facilitate timely and effective treatment [2]. Early diagnosis and timely treatment can effectively reduce the prevalence. However, due to the large population of China and the relatively limited number of ophthalmologists, relying solely on doctors to diagnose ophthalmic diseases requires a lot of time. For this reason, other methods are urgently needed for large-scale screening. Medical image analysis technology can not only greatly reduce the workload of doctors, but also has the advantages of being objective, fast and accurate [3].

Many scholars in the world are engaged in the research of retinopathy. In the detection of fundus image lesions [4], [5]: Ganguly *et al.* [6] proposed a detection method based on adaptive threshold for diabetic retinopathy with red lesions, a widespread ocular disease. The method can process each fundus image using thresholds of different intensities, so that it can be applied to fundus images with different quality and resolution acquired by different cameras. Balasubramanian *et al.* [7] used directional gradient histogram (hog) feature extraction and support vector machine classification method of retinal fundus image based on gabor filter blood vessel extraction to evaluate glaucoma detection. Liu *et al.* [8] proposed an automatic measurement system of cup disc ratio based on fundus image, which can automatically diagnose glaucoma. In terms of fundus image segmentation [9]–[11]: Samad *et al.* [12] used fuzzy C-means to cluster and detect the boundary of diabetic retinopathy object in fundus image, which can segment diabetic retinopathy efficiently and quickly. Hossain and Reza [13] adopted the vascular segmentation method based on MRF model, which was more effective than the highly complex image

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Hugo Albuquerque.

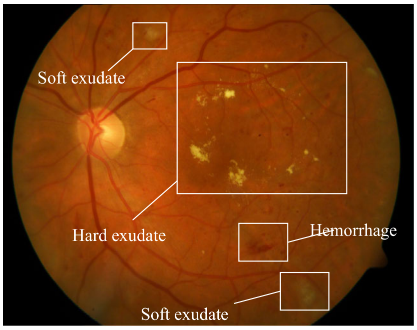


FIGURE 1. Fundus images with multiple lesions.

segmentation method. The deep learning method is also widely used in fundus image recognition [14]–[16]. Kele *et al.* [17] used the deep convolutional neural network method to automatically classify color fundus images for diabetic retinopathy. Prentasic and Loncaric [18] used convolutional neural network to detect the exudate in the color fundus image. The experiment proved that CNN can effectively detect the exudate in the fundus image.

Part of the above research is about the classification of a certain disease, such as the judgment and classification of eye diseases caused by diabetes or the detection of diseases such as glaucoma. In the other part, some algorithms are used to segment the exudates and blood vessels in the fundus image. These research results are excellent and meaningful. However, in some cases, the results of the above algorithms or methods are too single to provide specific retinopathy. Therefore, this study proposes an optimized CNN algorithm, which can identify a variety of lesions in the fundus image and provide reliable help for medical diagnosis. Many lesions in the fundus image may be intertwined at the same time, as shown in Figure 1. Moreover, few researches have proposed fundus image processing methods in the big data environment. Firstly, our study proposes an improved convolutional neural network that can identify and judge multiple lesions that are present in the eyeball, such as hard exudates and soft exudates. Fundus image content is complex, and images belonging to different categories may have high similarity. Detection and recognition of fundus images are affected by blood vessel deformation, image brightness, contrast, and other diseased areas [19]. If the traditional convolutional neural network is used for classification detection, it requires a large number of training samples, and it is easy to fall into a local optimal, and the training efficiency is low. Therefore, this study proposes a single-popping leapfrog algorithm to optimize the weight initialization and backpropagation of convolutional neural networks. The frog population is generated by gaussian distribution function, the position of the worst frog is updated constantly, the global optimal solution is found and taken as the initial weight of convolutional neural network. Then, the loss value of each training is monitored, and the abnormal gradient is corrected by the leapfrog algorithm. Finally, the final weight of the network is further optimized, and finally the high accuracy of lesion recognition is obtained. Next, since the training of convolutional neural network often needs a lot of iterative calculation, the time cost is even greater when facing the large-scale data set training. In order to solve this problem, this paper proposes Spark-SFCNN algorithm based on improved deep learning for the Spark platform. The algorithm uses the idea of group optimization based on Spark to train the data set in a distributed way, then summarizes the weights of all the training and optimizes them by evolutionary leapfrog, and takes the optimal weights as the initial values of the next group training. By this way, the uncertainty of traditional average method can be avoided effectively, and the distributed training of convolutional neural network is more effective and reliable. The experimental results show that the performance of Spark-SFCNN algorithm is much higher than that of the general algorithm in large-scale data training on Spark platform.

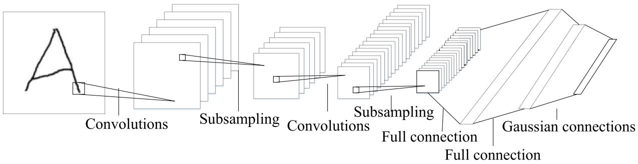


FIGURE 2. Classic LetNet structure.

The rest of the paper is structured as follows. In the second section, we introduce the convolutional neural network algorithm and SFLA. In the third section, CNN model based on single population leapfrog optimization is introduced. In the fourth section, we propose a packet optimization strategy and introduce Spark-SFCNN algorithm. In the fifth section, we analyze the experimental results of fundus image classification. At last, we summarize the thesis in the ninth section, and discuss the current challenges and future research directions.

II. RELATED ALGORITHM

A. CNN MODEL

Convolutional neural network is a multi-layer feedforward network [20], which was proposed by Lecun *et al.* [21] in 1989. In recent years, with the rapid increase in the amount of annotated data and the huge improvement in the performance of graphics processor units, the research on convolutional neural networks has risen rapidly and has achieved the latest results in various tasks [22]. Moreover, CNN is a deep neural network model containing convolutional layers, which has become a current research hotspot in the field of speech analysis and image recognition [23].

The time delay network and LetNet-5 are the earliest networks [24]. The classic LetNet-5 is shown in Figure 2. Structurally, the convolutional neural network includes an input layer, a hidden layer and an output layer. The hidden layer includes a convolution layer, a pooling layer, and a fully connected layer. The main function of the convolutional layer is to extract the feature of the picture, which contains several

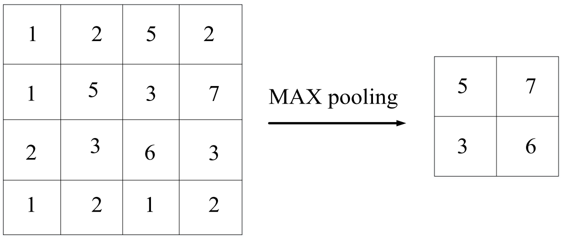


FIGURE 3. Max pooling.

convolution kernels. The convolution kernel includes a matrix composed of weight coefficients and a deviation value. The convolution calculation formula is as follows:

$$G^{l+1}(i,j) = [\{G^l \otimes w^{l+1}](i,j) + b$$
$$(i,j) \in \{0, 1, \ldots, L_{l+1}\}, L_{l+1} = \frac{L_l + 2p - f}{s_0} \quad (1)$$

where b represents the deviation value, G^l represents the input of the convolution at the l+1 layer, G^{l+1} represents the output of the convolution at the $l + 1$ layer, L_{l+1} represents the size of G^{l+1}, G^{l+1} represents the parameters of the convolution layer, s_0 and p respectively represent the size and step size of the Filter.

After a convolution calculation, the output feature map enters the pooling layer after being calculated by the activation function. The pooling layer is used to compress the results, reducing the spatial size of the data and the number of parameters in the network, so that the computing resource consumption is reduced and overfitting is effectively controlled. Pooling is also called downsampling, denoted as $S = down(C)$. Common pooling operations mainly include maximum pooling and average pooling. Among them the most widely used is the maximum pooling [25]. Maximum pooling is shown in Figure 3.

After the multi-layer convolution pooling calculation, the output feature map will be sequentially expanded in rows, connected into a vector to enter the fully connected network and then output through the multi-layer neural network. Like BP neural network, convolutional neural network also uses gradient descent to perform weight update and backpropagate.

B. SHUFFLED FROG LEAPING ALGORITHM

The Shuffled Frog Leaping Algorithm (SFLA) is an emerging and effective sub-heuristic group computing technique in the field of evolutionary computation [26] and a meme calculus algorithm combining genetics [27], which mainly consists of two parts: local Search and global information exchange. Local search and global information exchange continue until the convergence conditions are met [28]. The basic workflow of SFLA: A frogs group is randomly generated, and the i-th frog is represented as $f_i = [x_1, x_2, \ldots\ldots, x_s]$, and s indicates the dimension of each frog. After the frogs are generated, all frogs are sorted in descending order according to the fitness value, and divided into M populations. The division rules are:

$$H_i = \{f_{i+M(l-1)} \in p | 1 \leq l \leq N\} \quad (1 \leq i \leq M) \quad (2)$$

Each population contains N frog, with $F = M \times N$.

After the population is divided, the global optimal frog f_g is recorded, and the local search is carried out in each population. After the j-th local search, the worst adaptive frog in the population is expressed as $f_w(j)$, and the best adaptive frog is expressed as $f_b(j)$. The worst frog in each subpopulation is updated as follows.

$$D(j) = Rand() \times (f_b(j) - f_w(j)) \quad (3)$$
$$D_{new} = f_w(j) + D(j)(-D_{\max} \leq D(j) \leq D_{\max}) \quad (4)$$

where Rand() represents a random number between 0 and 1, and D_{max} represents the maximum frog jumping distance. After the frog position is updated, if there is no improvement, then f_g is substituted for $f_w(j)$. If there is still no improvement, a new frog is randomly generated instead of $f_w(j)$.

After all populations have been searched U_{mem} times, all frogs are mixed together, then reordered and subdivided into subpopulations according to the fitness value, so that the meme information of the frogs in each population can be fully transmitted, and then the local search is continued. Repeat the above steps until the convergence conditions are met and the algorithm ends.

III. CNN MODEL BASED ON IMPROVED SFLA

In view of the complexity of fundus images, this study introduces a single-population frog hop into the convolutional neural network weight initialization and weight update. By constantly updating the position of the worst frog, the global optimal frog is found as the initial weight of the network. During the network iteration process, monitor the loss value calculated each time, and correct the network weights that generate abnormal loss values through the frog jumping algorithm. The algorithm flow is shown in Figure 4.

A. WEIGHT INITIALIZATION BASED ON SINGLE-POPULATION LEAPFROG ALGORITHM

In convolutional neural networks, the convolution kernel and the fully connected layer weights are generally generated in a random manner. Because multiple lesions in the fundus image may exist at the same time, they are irregularly distributed, and the healthy fundus image is not clearly distinguished from the lesion fundus image. As shown in Figure 5. When the weights generated by the network are too bad, the network needs to spend more time and samples to perform gradient descent operation to correct the weights, and it is also easy to make the network fall into local optimization and affect the final result. Aiming at this problem, this paper proposes to introduce a single population leapfrog algorithm to initialize the weights of convolutional neural network.

Shuffled frog leaping algorithm is a sub-heuristic swarm evolution algorithm with excellent global search ability. It can effectively calculate the global optimal solution. However,

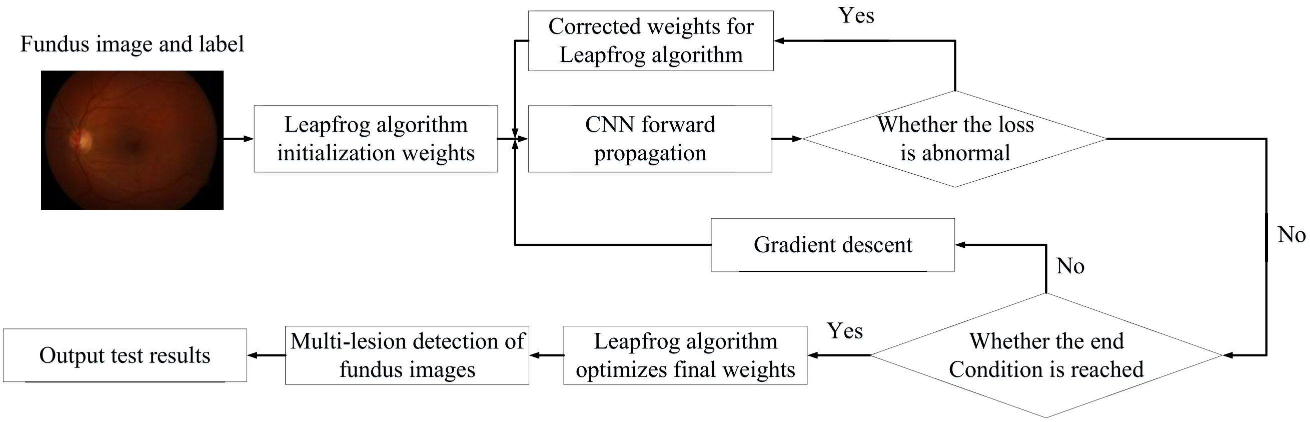

Fundus image and label
Leapfrog algorithm initialization weights
Corrected weights for Leapfrog algorithm
CNN forward propagation
Yes
Whether the loss is abnormal
No
Gradient descent
No
Whether the end Condition is reached
Yes
Leapfrog algorithm optimizes final weights
Multi-lesion detection of fundus images
Output test results

FIGURE 4. Flowchart of SFCNN algorithm.

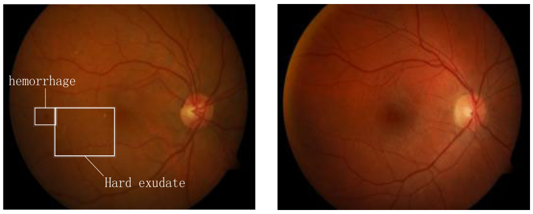


FIGURE 5. Similar fundus image.

directly applying it to the initialization of weights in this study will generate a large number of calculations, increase the time cost of the overall algorithm operation, and affect efficiency. So this study by simplifying hybrid leapfrog algorithm, single populations leapfrog algorithm is proposed. Different from the hybrid leapfrog algorithm, we no longer divide the population of the initial samples, but treat all the frogs as a population and directly conduct global optimization, and make some improvements on the leapfrog rules.

B. WEIGHT OPTIMIZATION METHOD BASED ON SINGLE-POPULATION LEAPFROG ALGORITHM

The traditional CNN backpropagation is essentially a gradient descent method, which updates the network parameters by the loss value calculated by forward propagation to find the optimal solution. However, the fundus image has problems such as dense blood vessels, overlapping interlaced lesions, and some lesions are very small. In addition, the illumination is severely affected by the shooting. Therefore, it is difficult to identify various lesions in the fundus image. The traditional gradient descent algorithm is prone to abnormal situations with large changes in the loss value, which affects the execution efficiency of the algorithm, and even causes problems such as the algorithm's convergence to a local optimum. In view of the above problems, this study uses a single population frog jumping algorithm to correct the poor gradient of the back propagation of the convolutional neural network.

During the network iteration process, the loss value calculated by each forward propagation is monitored. If the absolute value of the difference between the i-th calculated loss value and the $i-1$th loss value is greater than the threshold δ, that is $|l_i - l_{i-1}| > \delta$. It is considered the weight in the i-th forward propagation is an invalid weight. At this time, the single population frog jumping algorithm will be used to find the optimal weight. After the convolutional neural network meets the end condition, the weighted value f_{qb} is obtained, and the algorithm is executed again. The single-population frog leaping calculation can directly avoid the local optimum through the last single-population frog leaping optimization.

The flow chart of the weight optimization is shown in Figure 7:

IV. SFCNN ALGORITHM BASED ON SPARK PLATFORM

With the informatization construction of medical hygience, the scale and type of medical data are also growing at a very fast speed. In the medical big data, medical image and audio-visual data account for a large proportion. As a basic and noninvasive method of retina examination, fundus image has a huge amount of data. Inmedical image and audio-visual data account for a large proportion. As a basic and noninvasive method of retina examination, fundus image has a huge amount of data. In this chapter, we mainly focus on the classification of big data and multiple lesions of fundus image, and propose a distributed solution based on spark platform in combination with SFCNN algorithm above, which can realize the weight training of SFCNN in big data environment.

A. GROUP OPTIMIZATION TRAINING METHOD BASED ON SPARK

The training process of neural network has always been a long process, because to train the optimal weight, it needs a large number of training samples and thousands of steps of calculation. Convolution neural network is no exception. In the big data environment, a large number of sample training can improve the recognition accuracy of SFCNN algorithm, but its time consumption will be very huge. In order to solve the

Algorithm 1 Weight Initialization Algorithm

Step 1: Parameter initialization. Generate frog colony according to Gauss formula:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \quad (\mu = 0, \sigma = 1) \qquad (5)$$

where m is the number of frogs, and each frog p contains all weight parameters in the network.

Step 2: Sorting. All the frogs without calculated loss values are brought into the CNN model as shown in Figure 7. Some random images are selected from the training image library as reference images to carry out CNN forward propagation and calculate the loss value of each frog.

$$loss = \sum\nolimits_{k=1}^{b} (-\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{s} t_{ij} \ln(p_{ij}))/1000 \qquad (6)$$

where p represents the network output value, t represents the true value, s represents the dimension of lesion labels in each group, and b represents the number of types of retinopathy that need to be detected simultaneously.

Step 3: Search and position update within the population. According to step 2, the optimal frog f_b and the worst frog f_w can be obtained. Then Update the position of the worst frog with the position update function. For frog position update, an offset to the traditional frog jump formula and appropriately is increased by the random interval of Rand(). The formula is as follows:

$$D = (f_b + f_p - f_w) \times Rand(0, 1.2) \qquad (7)$$

$$f_{new} = f_w + D \qquad (8)$$

$$f_{pi} = \frac{1}{\exp(f_b - f_a) + 1} \times Rand(-1, 1) \times 0.0001 \qquad (9)$$

Here f_p represents the offset. Its dimension is the same as the dimension of each frog. The f_{pi} represents the value in the i-th dimension of f_p, and f_{new} represents the updated frog. By adding an offset, the performance of the leapfrog can be effectively improved, and the random interval is increased to make the algorithm easier to find the optimal solution. The jumping mode mentioned in this step is mapped to 2D coordinates as shown in Figure 6:

Step 4: Verify the calculation of the stop condition. Determine whether the algorithm satisfies the convergence condition. If it is satisfied, the algorithm will be stopped and the value of the optimal frog will be used as the initial weight of the convolutional neural network. Otherwise, go to step 3.

Algorithm 2 Weight Update Algorithm

Step 1: Input the samples. Enter the fundus images and perform one-hot encoding on the label.

Step 2: Convolution and pooling. Multiple convolution and pooling operations are performed on the input fundus image. At the end of each pooling calculation, the output result is activated by using the ReLU function.

Step 3: Full connection calculation. The predicted value is obtained through multi-layer full connection calculation. Then bring the predicted value into the soft max() function to calculate the result, and the loss value is calculated by formula (6).

Step 4: Error value comparison. It is determined whether the absolute value of the difference between the loss value of the current iteration and the previous loss value is greater than the threshold value δ. If it is not greater than the threshold value, skip to step 10. Otherwise the step 5 is performed.

Step 5: Initialization parameters. Record the weight w_b of the $(i-1)$-th forward propagation and set the number of frogs c.

Step 6: Generate a frog. Unlike the above mentioned method of generating frogs using Gaussion distribution, the frogs here are mainly generated around w_b. The formula is as follows:

$$w_{ij} = w_{bj} + 0.0001 \times \text{Rand}(-1, 1) \quad (1 < i \leq c-1) \qquad (10)$$

The w_{ij} represents the value in the j-th dimension of the generated i-th frog, and n represents the total number of frogs. At this time, in the frog group, the frog w_g must be satisfied to satisfy the condition: $l_g \leq l_b$. Here l_g and l_b are the loss values of w_g and w_b, respectively.

Step 7: Sorting. Randomly extract a few number of training pictures as input values. Then bring all the frogs in c into the CNN model shown in Figure 2. Calculate the loss value according to formula (6) and sort all the frogs in descending order of loss values.

Step 8: Search and location update within the population. This step is the same as step 3 in weight initialization algorithm.

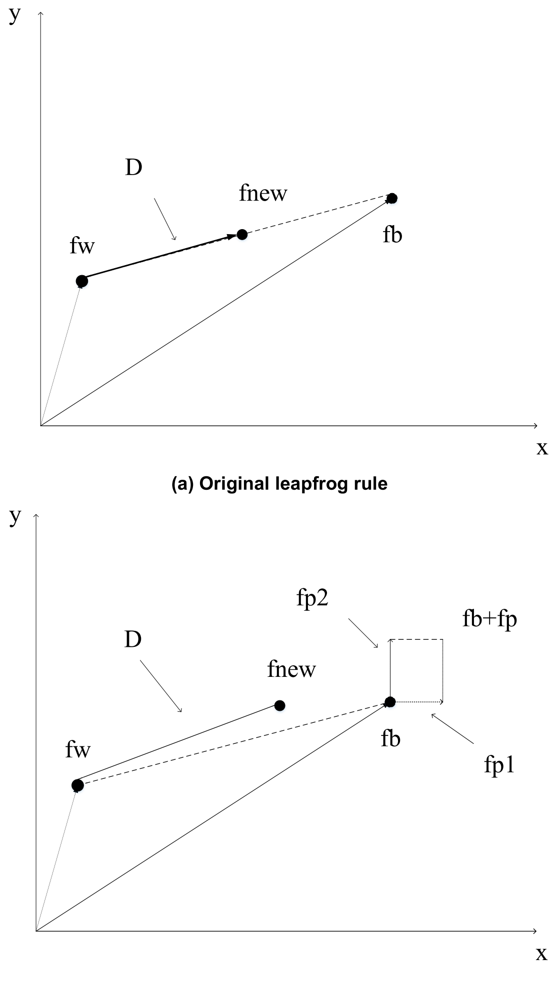
Step 9: Judge whether the leapfrog meets the stop condition. It is judged whether the algorithm satisfies the convergence condition. If it is satisfied, the algorithm is stopped, and the value of the optimal frog is updated to the network weight. Otherwise, the process proceeds to step 8.

Step 10: Verify the network stop condition. Determine whether the network meets the convergence condition. If it is satisfied, stop iteration, otherwise jump to step 1.

Step 11: Optimize the final weight. After the algorithm training is finished, the weight in the network is the final weight. Take the final weight as the initial frog w_b. The frog group is generated by formula (10), and then step 7, step 8 and step 9 are performed in sequence. Finally, the global optimal frog w_{qb} is obtained as the weight of the final training of the algorithm.

above problems, this paper combines the SFCNN algorithm with spark platform to effectively reduce the execution time of the algorithm and allocate the computing pressure through distributed computing.

This paper mainly solves the problem of the unity and conflict of the network weights between master server and slave server in the distributed gradient descent. The classic data parallel gradient descent algorithm is realized by synchronous



(a) Original leapfrog rule

（b）Improved leapfrog rule

FIGURE 6. Improvement of leapfrog rule.

communication between master and slave, but it is bound to have a lot of synchronous communication delay, which directly leads to the confusion of network weight training, a lot of synchronous operations make the algorithm unable to achieve real parallelism, and cannot play the advantages of spark parallel computing. In order to avoid the problem that the weights training of different servers cannot be effectively unified, this paper studies and proposes the weights parameter grouping optimization training method based on spark.

According to a large number of fundus image data acquired by the master node, it is divided into N subsets, then the fundus image training data set is: $S = \{M_1, M_2, M_3 \ldots\ldots, M_N\}$. Firstly, obtain the first fundus image training data M_1, take M_1 as an RDD, and divide the RDD into n partitions, then the data in the i-th partition is M_{1i}, as shown in Figure 8.

In the first grouping training, the initial weights are generated by the weight generation algorithm, and saved in the local or HDFS. Then the main node distributes the parameters, configuration and network updater status to all slave nodes. Thus, each training weight is used to realize distributed parallel training. When the training data of the last RDD partition is trained, the master node can get n different weight results. Different from the idea of average parameters, this paper uses the single population leapfrog optimization algorithm to optimize the aggregated weights. The aggregated n different weights are taken as the initial frogs, and the weights carried by the optimal green frogs are selected as the initial weights for the next group training by continuously updating

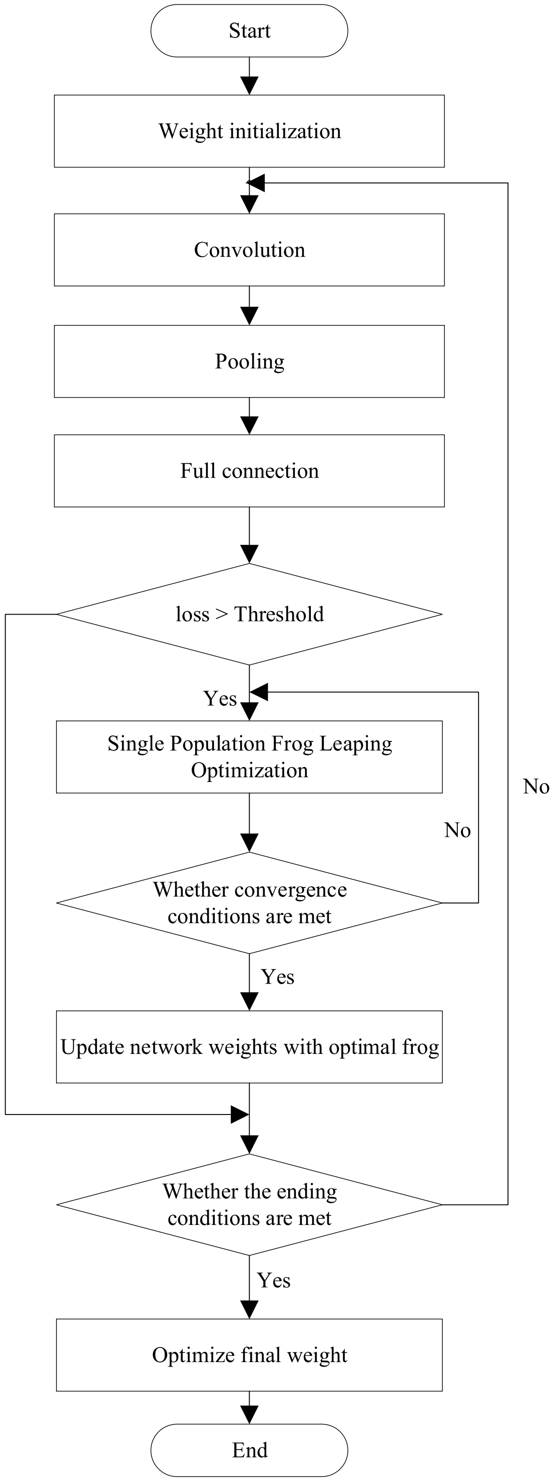

Start
Weight initialization
Convolution
Pooling
Full connection
loss > Threshold
Yes
Single Population Frog Leaping Optimization
No
Whether convergence conditions are met
Yes
Update network weights with optimal frog
No
Whether the ending conditions are met
Yes
Optimize final weight
End

FIGURE 7. Flow chart of weight optimization.

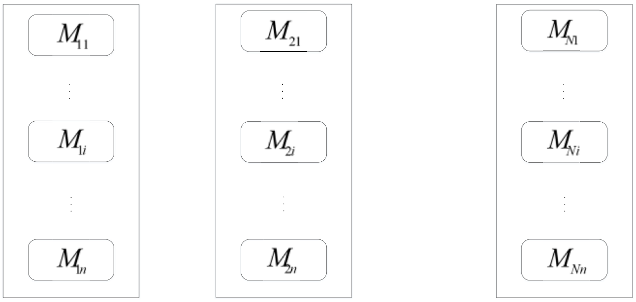


FIGURE 8. Grouped diagram of training sets.

the position of the worst frogs. Compared with the traditional average parameter method, the weight obtained by optimization calculation has stronger stability and no uncertainty due to average. Moreover, the parameters obtained by averaging are likely to cause the network to fall into the local optimum, and even the excellence of the weights is worse than the sum of weights. However, the weights of single population leapfrog can avoid these problems. The algorithm structure is shown in Figure 9.

B. MAIN STEPS OF SPARK-SFCNN ALGORITHM

Spark-SFCNN can effectively solve the problem of multi lesion classification of fundus image in big data environment, and realize the efficient training of SFCNN algorithm. The main algorithm steps are as follows.

Algorithm 3 Spark-SFCNN Algorithm

Step 1: Initialization parameters. It can be seen from the total operation steps S entered. After the second grouping, the calculation formula of each group's running steps s is as follows:

$$s = S/\lfloor (2w + 10)/10t \rfloor \tag{11}$$

where w represents the number of servers used for distributed computing, and t represents the total times.

Step 2: Network weight initialization. The optimal initial network weights are generated by the above-mentioned single population leapfrog algorithm based on spark and saved in the specified directory.

Step 3: Training set secondary grouping. According to the formula (11) set, the data of fundus images participating in this group of training are divided into n groups. This n meets the following formula:

$$n = 2w + 10 \tag{12}$$

where $w \geq 2$.

Step 4: Group training. The weights of convolutional neural network are trained in groups. Through independent training, n different network weights are finally obtained, and these n network weights are summarized to the main node.

Step 5: Frog leaping optimization. Taking these n networks as the initial frogs, the global optimal frog f_b is found by leaping in the same way as step (2), and the optimal frog is saved in the specified directory, covering the file saved in step (2).

Step 6: Judge the end condition. Judge whether the current grouping times t' is satisfied. If $t' \leq t$ is satisfied, skip to step 3. Otherwise, the algorithm ends.

V. EXPERIMENTED RESULTS AND ANALYSIS

The proposed algorithm code is implemented with the help of Python3.5. The experimental environment was Windows10/2.6GHz/8GB PC. The colored retinal fundus images obtained from DRIVE (Digital Retinal Images for Vessel Extraction) Database and DIAREDB1(Diabetic Retinopathy Image Database) [29] are analyzed and examined to obtain the classification results of the proposed system.

The complexity of the fundus images generates the influence of illumination and the interlacing of lesions. In the experiment, we first use the single population leapfrog algorithm to find the optimal initial weight, and then the loss value calculated by the forward propagation of the convolutional neural network is monitored. When the abnormal loss value is found, the single-population frog hopping algorithm is used to correct the weight of the abnormal loss value. To improve the efficiency of network execution and avoid the network falling into local optimum, the experimental process is shown in Figure 10.

A. CROSS ENTROPY LOSS CALCULATION METHOD

The experiment detected four retinopathies simultaneously, including hard exudate, bleeding, microaneurysm and soft exudate. Firstly, each fundus image was labeled and one-hot encoded to a two-dimensional array containing 8 elements, such as $y\_true = [[1, 0], [0, 1], [1, 0], [1, 0]]$. The eight elements are divided into four groups. The first value of each group of element is 1 to indicate no lesion, and the second value is 1 to indicate the occurrence of lesion. The predicted value $y\_predict$ calculated by the forward propagation of the convolutional neural network is also a two-dimensional array containing eight elements. Each of which will be calculated by soft max(), and then the loss value is calculated by the formula (6). In the experiment, it is found that the loss value of the individual weight calculation is too large, which exceeds the maximum value range of the programming language variable, resulting in an error. Therefore, we divide each loss value by 1000 to control the maximum loss value in a certain range in the actual experiments.

B. SFCNN EXPERIMENTAL RESULTS

The single-population frog leaping algorithm can find the optimal initial weight by global optimization, which can effectively reduce the initial loss value of the network, improve the network computing efficiency and even have a greater impact on the final classification result. As shown

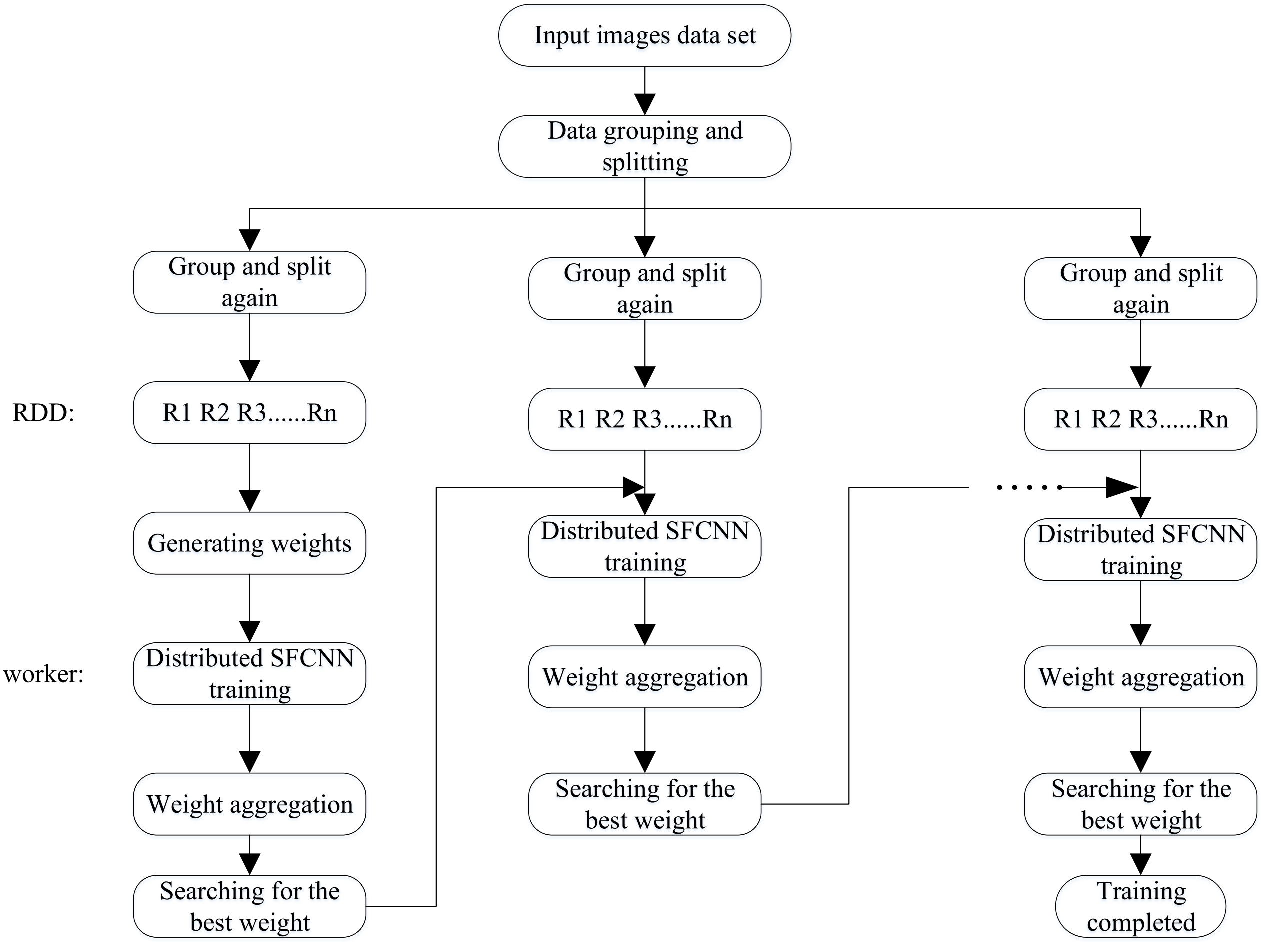


FIGURE 9. Structure of training set grouping algorithm.

in TABLE 1, it can be seen that compared with the traditional Gaussian distribution randomly generated initial value, the initial loss value calculated by optimized frog leaping algorithm is much lower. The images of different lesions may have certain similarities, which makes the initial network weights generated by the traditional methods have poor stability and the calculated loss values have large gaps. The poor initial parameters require more step gradient descent calculations to optimize, and may also lead the algorithm to falling into local optimum. Therefore, the initial value can be optimized by the single-popping leapfrog algorithm, which can effectively ensure the stability of the initial value generation of the convolutional neural network in multi-label classification of fundus images, and the high quality initial network weight can effectively reduce the number of network iterations, improve the efficiency of network execution, and avoid the network falling into local optimum to some extent.

With the increase of the number of iterations, the loss value of the network shows a tortuous downward trend. However, due to the high similarity of the fundus image and it is seriously affected by the light and tiny lesions. The traditional convolutional neural network error will have a large fluctuation value. This will affect the efficiency of the algorithm's execution, and even lead to local optimality, as shown in Figure 11.

TABLE 1. Comparison of initial loss.

Items	Related loss	SFCNN	CNN
Best	First loss	691.5060	12145.06
	Second loss	426.5983	32318.79
Worst	First loss	1975.8852	52076.46
	Second loss	1549.8188	38816.58

In this study, the comparison of error value and threshold value is monitored, and the weight of the abnormal loss value is corrected by the single-population leapfrog algorithm. The weight comparison method is as follows:

$$\mathrm{abs}(l_{i-1} - l_i) > o_1 \quad \text{or} \quad \mathrm{abs}(l_{i-1} - l_i) > o_2,$$
$$o_1 = 300 + 2 \times l_{i-1}, \quad o_2 = 300 + 6 \times l_i \tag{13}$$

where o_1 and o_2 are the two defined thresholds, l_i is the loss value of the i-th iteration, and abs() is the absolute value function. The loss value line graph optimized by the single-population leapfrog algorithm is shown in Figure 12, and it

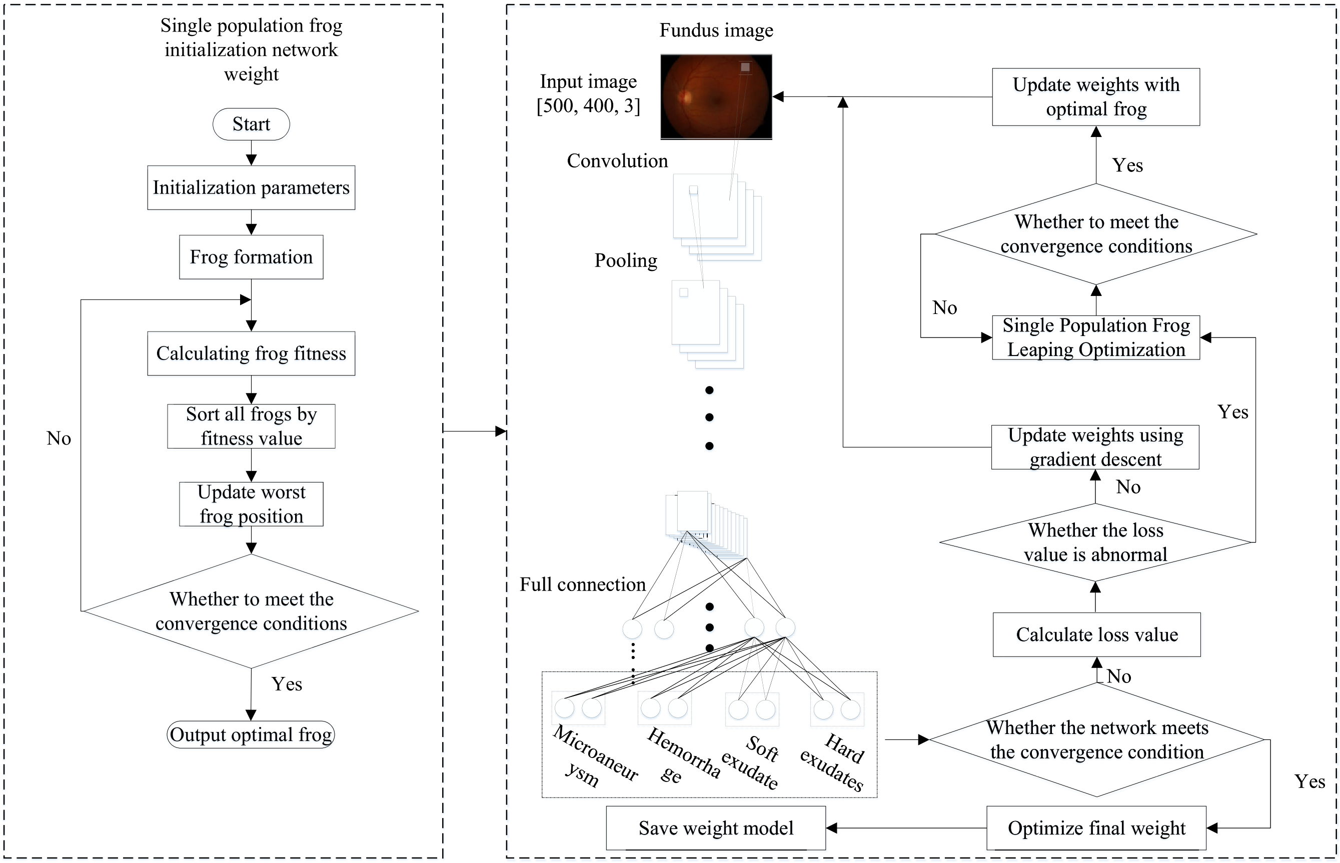

Single population frog initialization network weight
Start
Initialization parameters
Frog formation
Calculating frog fitness
Sort all frogs by fitness value
Update worst frog position
Whether to meet the convergence conditions
No
Yes
Output optimal frog
Fundus image
Input image [500, 400, 3]
Convolution
Pooling
Full connection
Microaneurysm
Hemorrhage
Soft exudate
Hard exudates
Update weights with optimal frog
Yes
Whether to meet the convergence conditions
No
Single Population Frog Leaping Optimization
Yes
Update weights using gradient descent
No
Whether the loss value is abnormal
Calculate loss value
No
Whether the network meets the convergence condition
Yes
Save weight model
Optimize final weight

FIGURE 10. Flowchart of SFCNN algorithm experiment.

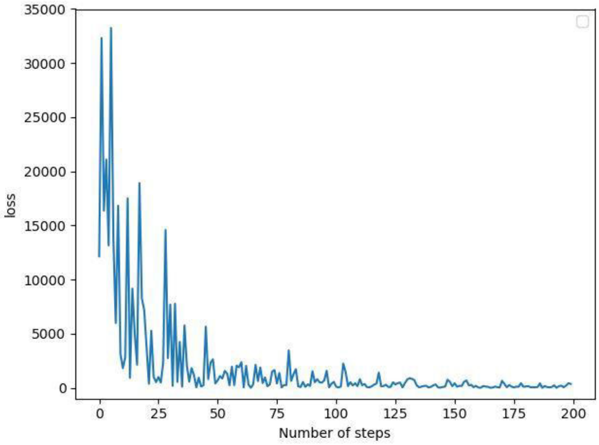


FIGURE 11. CNN loss reduction chart.

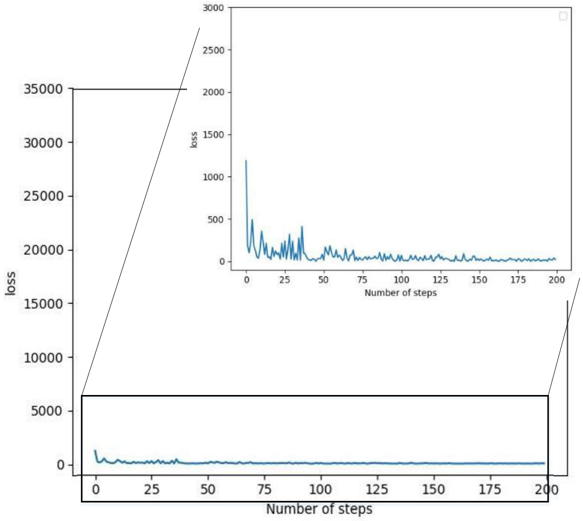


FIGURE 12. Loss reduction chart of improved algorithm.

can be seen that the fluctuation after optimization is smaller and the excessive fluctuation is less than that before optimization. Thus, the possibility of falling into local optimum is effectively reduced.

After the completion of the convolutional neural network, a single-population frog hopping optimization is performed on the final weight of the network, which can directly and effectively avoid falling into local optimization, and further optimize the network weight, so that the detection of lesions in complex fundus images such as darkness of light and overlapping of lesions is more accurate. Figure 13 shows the multi-lesion detection effect of CNN and the algorithm on three kinds of complex fundus images affected by different

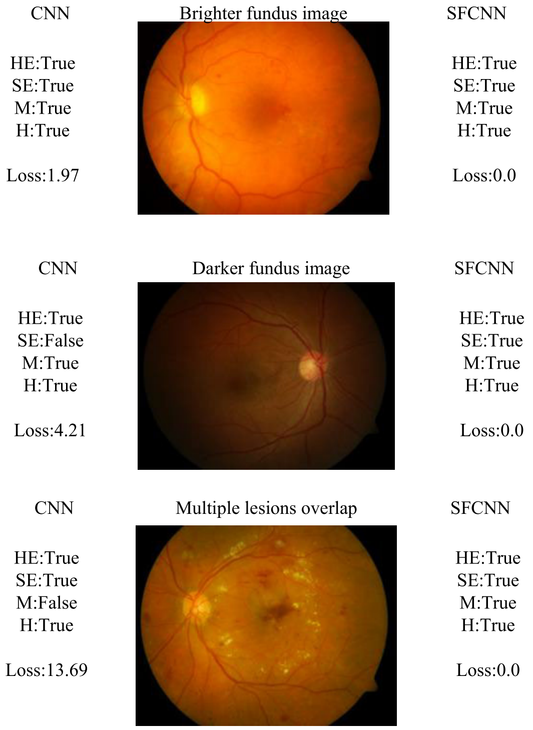


FIGURE 13. Process of complex fundus image detection.

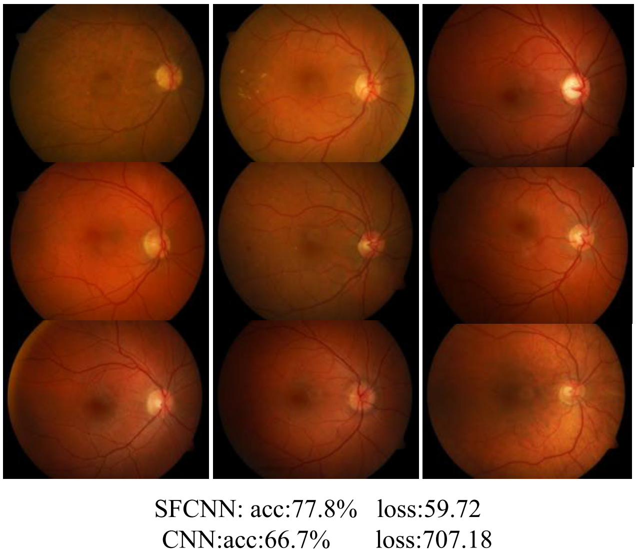


FIGURE 14. Similar fundus images.

factors. In the figure, HE stands for hard exudates, SE stands for soft exudates, M stands for microaneurysms, H stands for hemorrhages and Loss represents the total loss value of the network for four lesions (accurate to one decimal place). Figure 14 shows the accuracy and the sum of losses of two algorithms for multi-lesion recognition of similar fundus images. It can be seen from the figures that the traditional CNN algorithm does not accurately detect individual lesions on part of the fundus image, while the SFCNN accuracy rate is higher on such fundus images, and the loss value is lower.

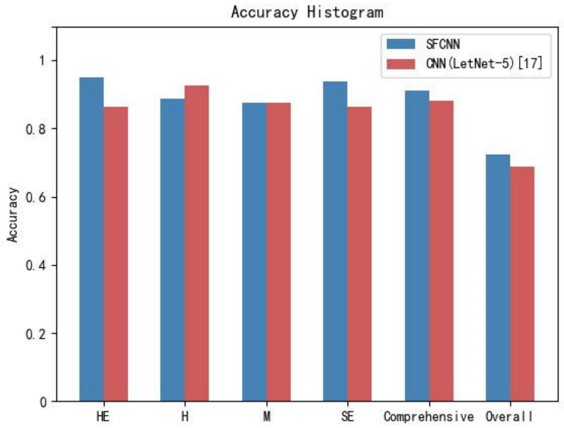


FIGURE 15. Accuracy contrast histogram of multiple lesions.

The convolutional neural network part of the experiment mainly performs multiple lesion classification detection on the fundus image through two layers of convolution, two pooling layers and one fully connected layer. The algorithm is evaluated mainly from the aspects of accuracy and sensitivity. And comprehensive recognition rate and overall recognition rate are added as reference. Comprehensive recognition rate: $p = (c_1 + c_2 + c_3 + c_4)\big/n$, where c_1, c_2, c_3 and c_4 represent the correct number of four lesions, respectively, and n represents the total number of samples in the test set. The overall recognition rate is the correct percentage of the four lesions detected simultaneously for each sample. It can be seen from TABLE 2 and Figure 15 that compared with SFLA-CNN, AlexNet neural network and LetNet-5 neural network, SFCNN has been improved in varying degrees. Figure 16 compares the running time of SFCNN and SFLA-CNN. It can be seen from the figure that the running time increases with the number of steps, which has certain randomness. However, the running time of SFLA-CNN with the same number of runs is generally longer than that of SFCNN. So the improved SFLA is more suitable for the improvement of this paper.

The experiment also compares the SFCNN algorithm proposed in this paper with the traditional CNN algorithm, SVM [30] algorithm and KPCA+SVM [31] fusion algorithm to compare the accuracy of the classification of the hard exudate of the fundus image. The detection accuracy is represented by histogram as shown in Figure 17. From the classification effect histogram, it can be seen that the SFCNN algorithm improves the accuracy of the multi-lesion detection and single lesion detection compared with the existing algorithms, which proves that the single-population leapfrog optimization convolutional neural network has certain superiority and effectiveness.

C. SPARK-SFCNN EXPERIMENTAL RESULTS

We first compare the single-machine single-population leapfrog algorithm with the single-population leapfrog

TABLE 2. Comparison of experimental results.

Lesion category	SFCNN		SFLA-CNN		CNN(AlexNet)[14]		CNN(LetNet-5)[17]	
	Accuracy	Sensitivity	Accuracy	Sensitivity	Accuracy	Sensitivity	Accuracy	Sensitivity
Hard exudate	95.0%	96.9%	87.6%	73.6%	86.2%	84.2%	86.3%	87.9%
Hemorrhages	88.7%	88.2%	89.9%	90.1%	88.5%	88.7%	92.5%	93.3%
Microaneurysm	87.5%	88.9%	87.3%	88.3%	84.9%	86.3%	87.5%	88.8%
Soft exudate	93.7%	94.5%	88.7%	83.3%	90.7%	91.5%	86.3%	83.4%

algorithm based on spark. In the experiment, 100 eye ground images were randomly selected as the reference set. The test frog group is set as 50 frogs. After the updated frogs are distributed adaption calculation, 100 fundus images are stored as one RDD, and ten images are stored in each RDD partition. The simulation experiment recorded the time consumption (/s) of 50 steps, 100 steps, 150 steps, 200 steps and 250 steps of leapfrog update. The time comparison line chart is shown in Figure 18. It can be seen from the figure that the training time of the two methods is very close when the leapfrog update is selected at 50 steps, and then the time consumption of the two algorithms has a certain expansion trend with the increase of the number of leapfrog update steps. However, in the 200 steps experiment, the two algorithms are close again. As a whole, with the increase of running steps, the advantages of spark based single population leapfrog algorithm will be more obvious.

In order to simulate the training experiment of Spark-SFCNN algorithm in big data environment, this study conducted 10000 training iterations for SFCNN and Spark-SFCNN, and compared the time efficiency. As mentioned above, the sample training set was expanded to about 5000 fundus images training sets, but it failed to reach 10000. In this experiment, 10000 iterations were realized through circular training.

Firstly, the number of training steps of each group was calculated by formula (11). In order to facilitate the calculation, t is taken as 5, and $w = 2$, then s is taken as 200, and the number of frog update steps in the leapfrog algorithm is 50. The time comparison is shown in Figure 19. It can be seen from the figure that the time consumed by SFCNN algorithm based on spark is less than that of the single SFCNN, and the execution efficiency is higher, but the reduced time is not too obvious. TABLE 3 shows the comparison of accuracy and sensitivity between SFCNN and Spark-SFCNN. It can be seen from the table that the accuracy of Spark-SFCNN in the classification and detection of bleeding and soft exudates is higher than that of SFCNN, while the accuracy of exudates and microaneurysms is lower than that of SFCNN. However, from the overall point of view, the accuracy difference between the two algorithms is not large, while the time efficiency of Spark-SFCNN is better. It can be seen that the weight parameter grouping optimization based on spark has certain advantages.

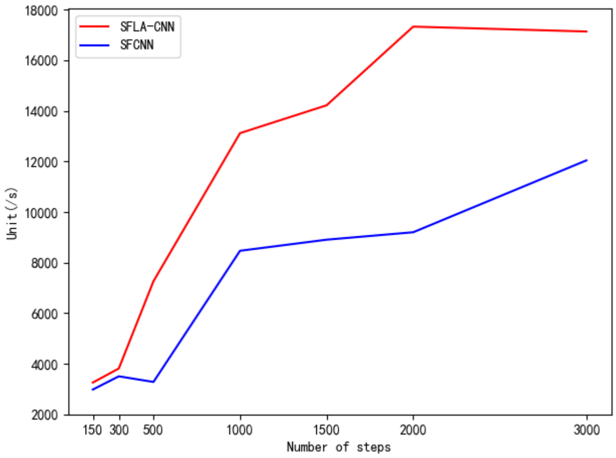


FIGURE 16. Line chart of time comparison.

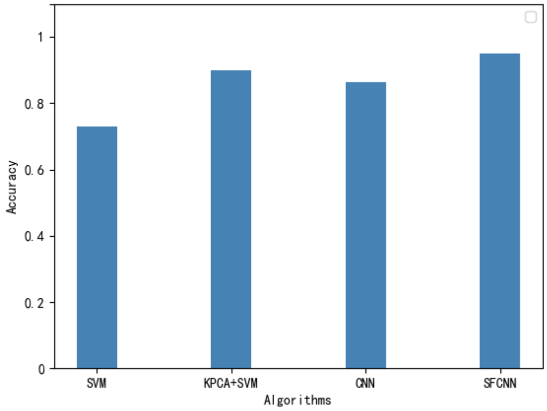


FIGURE 17. Histogram of hard exudation detection accuracy.

D. ANALYSIS OF EXPERIMENTAL RESULTS

Through above-mentioned experimental results, we find that the proposed SFCNN algorithm has the following characteristics:

(1) In the exudate detection, the proposed algorithm in this paper has some advantages over other algorithms, as shown in TABLE 1 and Figure 14.

(2) In the detection of multiple lesions at the same time, the accuracy of this algorithm is also improved, but it still needs to be further improved.

TABLE 3. Accuracy and sensitivity comparison of SFCNN and Spark-SFCNN.

Lesion category	SFCNN		Spark-SFCNN	
	Accuracy	Sensitivity	Accuracy	Sensitivity
Hard exudate	96.7%	96.9%	95.9%	96.3%
Hemorrhages	92.1%	93.5%	94.2%	95.1%
Microaneurysms	90.5%	91.1%	90.3%	90.9%
Soft exudate	92.6%	93.5%	95.1%	94.9%

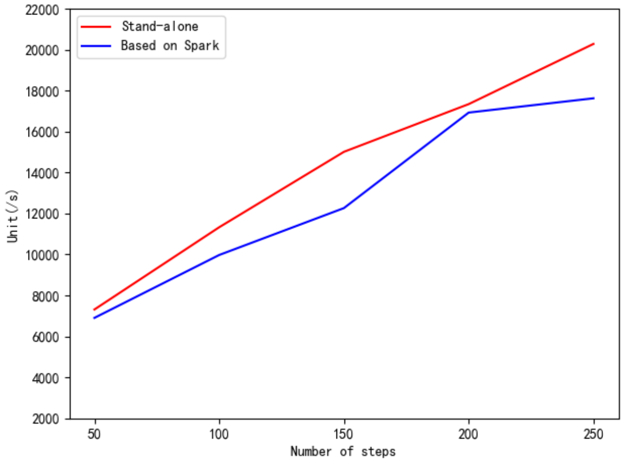


FIGURE 18. Line chart of leapfrog time comparison.

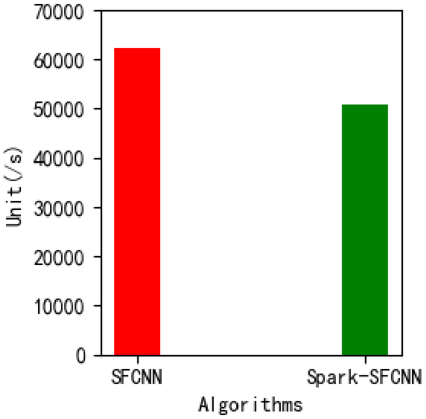


FIGURE 19. Histogram of time comparison.

(3) In the detection of fundus lesions, the weights trained by this algorithm are more robust than those trained by traditional CNN algorithm.

(4) In the single population leapfrog algorithm based on spark. It can be seen from Figure 18 that the time consumption of the single machine leapfrog algorithm is higher than that of the single population leapfrog algorithm based on spark, but the difference between the two is not large, and the time consumption of the two is very close at step 50 and step 200, but the overall time consumption of the two increases with the number of execution steps. For the situation that the gap is too small in the experiment, this study also makes some analysis. Firstly, Due to the poor performance of the local installed virtual machine used in the simulation test, and due to the limitation of computing resources, there are only two virtual machines for spark computing, so the distributed advantages cannot be well reflected. Secondly, due to the limitation of conditions, this test only selects the time result of one operation, and does not use the average value of multiple operations. In the calculation process, it will be affected by the physical machine where the virtual machine is located, and there are other programs or systems that occupy the calculation resources, resulting in the situation of long operation time. But from the overall results, the implementation efficiency of the leapfrog algorithm combined with spark platform is better than the traditional way.

(5) In the weight training method based on spark. It can be seen from Figure 19 that the consumption time of SFCNN is about 10000 seconds longer than that of Spark-SFCNN, but it does not reflect the advantages of spark distributed computing platform. Through the analysis of algorithm code, we find that the reason is that the framework of algorithm code is not optimized in detail. In the process of distributed computing, it is necessary to create an operation chart every time the training calculation of grouping or leapfrog fitness calculation is started, while SFCNN only needs to create an operation chart in the main operation, and the experiment consumes a lot of computing resources in the aspects unrelated to the algorithm. However, this experiment is also sufficient to prove that SFCNN trained by combining spark framework group optimization is more efficient in time efficiency. From TABLE 3, it can be seen that there is little difference in accuracy between the two, and all this method has certain feasibility and research significance.

VI. CONCLUSION

In this paper the simplified SFLA algorithm is used to treat the global as a population for leapfrog calculation, and a single-population leapfrog algorithm is proposed to optimize CNN for the recognition of multiple lesions in fundus images. Due to the uncertainty of CNN initial weight selection, it is easy to affect the network execution efficiency. CNN own backpropagation calculation is easy to fall into local optimum. Then this paper proposes the single-population leapfrog algorithm to optimize the weight of the loss value anomaly, correct the parameter value without falling into local optimum, improve the recognition rate of the lesion and reduce the training cost.

The accuracy of multi-category is further improved. In the future, the training sample library needs to be expanded to improve the quality of samples and labels. At the same time, in terms of execution time, both the latest leapfrog algorithm research results and more efficient improvement methods can be used to improve the training timeliness of the proposed algorithm. In the follow-up study, the application of convolutional neural network in the fundus image segmentation and lesion localization will be explored to solve the knowledge mining problems of electronic health records that are very difficult to solve by traditional methods.

REFERENCES

[1] T. Walter, P. Massin, A. Erginay, R. Ordonez, C. Jeulin, and J.-C. Klein, "Automatic detection of microaneurysms in color fundus images," *Med. Image Anal.*, vol. 11, no. 6, pp. 555–566, Dec. 2007.

[2] F. Grassmann, J. Mengelkamp, C. Brandl, S. Harsch, M. E. Zimmermann, B. Linkohr, A. Peters, I. M. Heid, C. Palm, and B. H. F. Weber, "A deep learning algorithm for prediction of age-related eye disease study severity scale for age-related macular degeneration from color fundus photography," *Ophthalmology*, vol. 125, no. 9, pp. 1410–1420, Sep. 2018.

[3] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: A review," *J. Med. Syst.*, vol. 42, no. 11, p. 226, Oct. 2018.

[4] A. Agarwal, S. Gulia, S. Chaudhary, M. K. Dutta, R. Burget, and K. Riha, "Automatic glaucoma detection using adaptive threshold based technique in fundus image," in *Proc. 38th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2015, pp. 416–420.

[5] Z. Zhang, B. H. Lee, J. Liu, D. W. K. Wong, N. M. Tan, J. H. Lim, F. Yin, W. Huang, H. Li, and T. Y. Wong, "Optic disc region of interest localization in fundus image for glaucoma detection in ARGALI," in *Proc. 5th IEEE Conf. Ind. Electron. Appl.*, Taichung, Taiwan, Jun. 2010, pp. 1686–1689.

[6] S. Ganguly, S. Ganguly, K. Srivastava, M. K. Dutta, M. Parthasarathi, R. Burget, and K. Riha, "An adaptive threshold based algorithm for detection of red lesions of diabetic retinopathy in a fundus image," in *Proc. Int. Conf. Med. Imag., M-Health Emerg. Commun. Syst. (MedCom)*, Greater Noida, India, Nov. 2014, pp. 91–94.

[7] T. Balasubramanian, S. Krishnan, M. Mohanakrishnan, K. R. Rao, C. V. Kumar, and K. Nirmala, "HOG feature based SVM classification of glaucomatous fundus image with extraction of blood vessels," in *Proc. IEEE Annu. India Conf. (INDICON)*, Bengaluru, India, Dec. 2016, pp. 1–4.

[8] J. Liu, F. S. Yin, D. W. K. Wong, Z. Zhang, N. M. Tan, C. Y. Cheung, M. Baskaran, T. Aung, and T. Y. Wong, "Automatic glaucoma diagnosis from fundus image," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Boston, MA, USA, Aug. 2011, pp. 3383–3386.

[9] F. Bokhari, T. Syedia, M. Sharif, M. Yasmin, and S. L. Fernandes, "Fundus image segmentation and feature extraction for the detection of glaucoma: A new approach," *Current Med. Imag. Rev.*, vol. 14, no. 1, pp. 77–87, Jan. 2018.

[10] S. Sridhar, K. J. D. S. Rao, N. Hemanth, and M. K. Nath, "An efficient blood vessel segmentation from color fundus image," *Int. J. Comput. Appl.*, vol. 119, no. 2, pp. 25–28, Jun. 2015, doi: 10.5120/21041-3597.

[11] J. G. Zilly, J. M. Buhmann, and D. Mahapatra, "Boosting convolutional filters with entropy sampling for optic cup and disc image segmentation from fundus images," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Cham, Switzerland: Springer, 2015, pp. 136–143.

[12] R. Samad, M. S. F. Nasarudin, M. Mustafa, D. Pebrianti, and N. R. H. Abdullah, "Boundary segmentation and detection of diabetic retinopathy (DR) in fundus image," *J. Amer. Soc. Horticultural Sci.*, vol. 77, no. 6, pp. 25–28. Nov. 2015.

[13] N. I. Hossain and S. Reza, "Blood vessel detection from fundus image using Markov random field based image segmentation," in *Proc. 4th Int. Conf. Adv. Electr. Eng. (ICAEE)*, Dhaka, Bangladesh, Sep. 2017, pp. 123–127.

[14] X. Li, T. Pang, B. Xiong, W. Liu, P. Liang, and T. Wang, "Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification," in *Proc. 10th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Shanghai, China, Oct. 2017, pp. 1–11.

[15] P. F. Ordóñez, C. M. Cepeda, J. Garrido, and S. Chakravarty, "Classification of images based on small local features: A case applied to microaneurysms in fundus retina images," *J. Med. Imag.*, vol. 4, no. 4, p. 1, Nov. 2017.

[16] A. Diaz-Pinto, S. Morales, V. Naranjo, T. Köhler, J. M. Mossi, and A. Navea, "CNNs for automatic glaucoma assessment using fundus images: An extensive validation," *Biomed. Eng. OnLine*, vol. 18, no. 1, p. 29, Mar. 2019.

[17] K. Xu, D. Feng, and H. Mi, "Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image," *Molecules*, vol. 22, no. 12, p. 2054, Nov. 2017.

[18] P. Prentasic and S. Loncaric, "Detection of exudates in fundus photographs using convolutional neural networks," in *Proc. 9th Int. Symp. Image Signal Process. Anal. (ISPA)*, Zagreb, Croatia, Sep. 2015, pp. 188–192.

[19] G. Quellec, S. R. Russell, T. E. Scheetz, E. M. Stone, and M. D. Abramoff, "Computational quantification of complex fundus phenotypes in age-related macular degeneration and Stargardt disease," *Investigative Ophthalmol. Vis. Sci.*, vol. 52, no. 6, pp. 2976–2981, May 2011.

[20] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, no. 2, pp. 1097–1105.

[21] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[22] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Isgum, "Automatic segmentation of MR brain images with a convolutional neural network," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1252–1261, May 2016.

[23] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, Jan. 1995.

[24] X. Du, Y. Cai, S. Wang, and L. Zhang, "Overview of deep learning," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Automat. (YAC)*, Nov. 2016, pp. 159–164.

[25] C. Li, S. X. Yang, Y. Yang, H. Gao, J. Zhao, X. Qu, Y. Wang, D. Yao, and J. Gao, "Hyperspectral remote sensing image classification based on maximum overlap pooling convolutional neural network," *Sensors*, vol. 18, no. 10, p. 3587, Oct. 2018.

[26] A. Alghazi, S. Z. Selim, and A. Elazouni, "Performance of shuffled frog-leaping algorithm in finance-based scheduling," *J. Comput. Civil Eng.*, vol. 26, no. 3, pp. 396–408, May 2012.

[27] J. Li, H. Wang, and B. Yan, "Application of velocity adaptive shuffled frog leaping bat algorithm in ICS intrusion detection," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, Chongqing, China, May 2017, pp. 3630–3635.

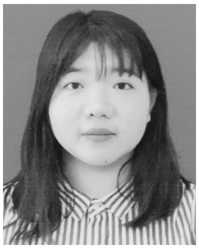
[28] C. L. Antunes, R.-A. Naghizadeh, B. Vahidi, and S. H. Hosseinian, "Parameter identification of Jiles-Atherton model using SFLA," *Compel Int. J. Comput. Math. Elect. Electron. Eng.*, vol. 31, no. 4, pp. 1293–1309, Jul. 2012.

[29] A. D. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 203–210, Mar. 2000.

[30] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Scholkopf, and A. J. Smola, "Support vector machine," *Comput. Sci.*, vol. 1, no. 4, pp. 1–28, 2002.

[31] Z. Li and X. Tian, "Study of soft sensor modeling method based on KPCA-SVM," in *Proc. 6th World Congr. Intell. Control Autom.*, Dalian, China, 2006, pp. 4876–4880.

WEIPING DING (Senior Member, IEEE) received the Ph.D. degree in computation application from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2013. He was a Visiting Scholar with the University of Lethbridge (UL), Lethbridge, AB, Canada, in 2011. From 2014 to 2015, he was a Postdoctoral Researcher with the Brain Research Center, National Chiao Tung University (NCTU), Hsinchu, Taiwan. In 2016, he was a Visiting Scholar with the National University of Singapore (NUS), Singapore. From 2017 to 2018, he was a Visiting Professor with the University of Technology Sydney (UTS), Ultimo, NSW, Australia. He has published more than 50 research peer-reviewed journal and conference papers in these fields, including the IEEE Transactions on Fuzzy Systems (TFS), the IEEE Transactions on Neural Networks and Learning Systems (TNNLS), the IEEE Transactions on Cybernetics (TCYB), the IEEE Transactions on Industrial Informatics (TII), the IEEE Transactions on Systems, Man, and Cybernetics (TSMC), and the IEEE Transactions on Biomedical Engineering (TBME), and has held 15 approved invention patents till now. His main research directions involve data mining, machine learning, granular computing, evolutionary computing, and big data analytics. He currently serves on the Editorial Advisory Board of Knowledge-Based Systems and Editorial Board of Information Fusion, Applied Soft Computing. He serves/served as an Associate Editor for the IEEE Transactions on Fuzzy Systems, *Information Sciences*, *Swarm and Evolutionary Computation*, IEEE Access, and the *Journal of Intelligent and Fuzzy Systems*. He has delivered more than ten keynote speeches at international conferences and has co-chaired several international conferences and workshops in the area of fuzzy decision-making, data mining, and knowledge engineering.

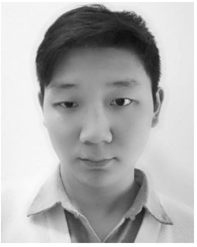


YING SUN is currently pursuing the master's degree with the School of Information Science and Technology, Nantong University. Her main research interests include rough set and deep learning.

LONGJIE REN is currently pursuing the master's degree with the School of Information Science and Technology, Nantong University. His main research interests include data mining and image recognition.

HENGRONG JU received the B.Sc. and M.Sc. degrees in computer science and technology from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2012 and 2015, respectively, and the Ph.D. degree in management science and engineering from Nanjing University, Nanjing, China, in 2019. He is currently a Lecturer with the School of Information Science and Technology, Nantong University. From 2017 to 2018, he worked as a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He has authored or coauthored more than ten scientific articles in international journals and conferences. His current research interests include knowledge discovery and granular computing.

ZHIHAO FENG is currently pursuing the master's degree with the School of Information Science and Technology, Nantong University. His main research interests include data mining and image recognition.



MING LI is currently pursuing the master's degree with the School of Information Science and Technology, Nantong University. His main research interest includes big data analysis and processing.