Neural Computing and Applications (2023) 35:14653–14665
https://doi.org/10.1007/s00521-021-06234-w

S.I: ML4BD SHS

Aggregated decentralized down-sampling-based ResNet for smart healthcare systems

Zhiwen Jiang[1] · Ziji Ma[1] · Yaonan Wang[1] · Xun Shao[2] · Keping Yu[3] · Alireza Jolfaei[4]

Received: 22 February 2021 / Accepted: 13 June 2021 / Published online: 26 June 2021

Abstract
With the rapid growth of the world's population and urbanization, people are increasingly seeking higher-quality medical services to improve their lives. The classification method based on deep convolutional neural networks (CNNs) is widely used in smart healthcare systems along with advancements in communication and hardware technology. Unfortunately, for conventional deep CNN algorithms, most of the regions do not participate in the convolution operation, resulting in the loss of feature information and the correlation of information between the features. To address this issue, this paper proposes a new strategy of aggregation decentralized down-sampling to prevent the loss of feature information. The regions that are not involved in the convolution operation are re-convoluted and stacked onto depth information in the forward propagation layer and the short-circuit layer, ensuring gradual convergence of the feature map and avoiding the loss of feature information. The accuracy of the proposed residual network (ResNet) system for classification tasks showed an average improvement of 2.57% compared with the conventional ResNet strategies.

Keywords Deep convolution neural network · ResNet · Down-sampling · Classification of medical images

1 Introduction

With the development of artificial intelligence (AI) technology, deep convolution neural networks (CNNs) are being adopted in healthcare fields, such as intelligent medical diagnostic systems and intelligent pathologic image analysis systems. Among their many applications, classification systems based on AI techniques have attracted the attention of global researchers. The basis of AI techniques is the birth of the neural network model in 1943. After decades of development, neural networks have gradually developed from a shallow layer to deep neural networks, which can extract higher-level abstract information to obtain better results. The structural diagram of a neural network is shown in Fig. 1, including the input layer, hidden layers and output layer.

As an important branch of neural networks, CNNs have recently played a very important role in classification systems. LeNet [1], proposed in the 1990s, marks the beginning of CNNs, which have been further developed to GoogLeNet [2] and VGGNet [3], the champion and runner-up of the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2014, respectively. Since then, the number of layers in CNNs has been increasing, and they are becoming increasingly complex. The residual network (ResNet) proposed by He et al. [4] in 2015 increased the number of CNN layers from dozens to hundreds, laying a solid foundation for deeper networks. The Inception-ResNet [5] network proposed in 2017 has learned from the ResNet structure, achieving better classification performance.

In this paper, we propose an aggregated decentralized down-sampling (ADD) method that is embedded into the ResNet to enhance its feature extraction ability. The improved ResNet can increase the utilization rate of the

✉ Ziji Ma
zijima@hnu.edu.cn

[1] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China

[2] School of Regional Innovation and Social Design Engineering, Kitami Institute of Technology, Kitami, Japan

[3] Global Information and Telecommunication Institute, Waseda University, Tokyo, Japan

[4] Department of Computing, Macquarie University, Sydney, NSW 2113, Australia

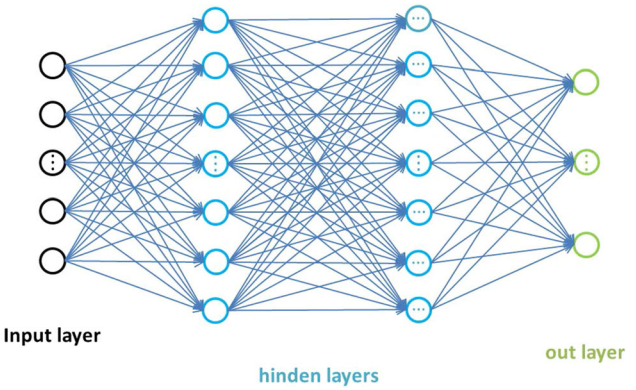

Fig. 1 Structure of neural network

information contained in the feature map, which reduces the loss of high-dimensional information. For medical image processing, high-resolution and large-size images provide more pathological details as well as much more computational complexity. For common ResNet, the larger the step size, the faster the feature graph convergences. The proposed ADD method makes a good trade-off between high recognition accuracy and a fast convergence rate. Besides avoiding the loss of information, the required number of convolution layers reduces with the acceleration of the convergence. This means that the number of parameters in ResNet decreases as well, greatly reducing the total computational cost. The proposed ADD method can also help other CNNs, not only ResNet, to greatly improve their practical performance. In addition, the loss of information caused by using a large step size probably reduces the ability to detect and recognize small-sized objects, such as small-diseased areas, which are crucial for early diagnosis. The proposed ADD ResNet reuses the skipped regions and combines them into depth information. Thus, the information of small targets is reserved to prevent the detection performance for small objects from degrading.

This article is structured as follows. In Sect. 2, we introduce related works on classification systems. In Sect. 3, we introduce several improved schemes for CNNs in detail. In Sect. 4, we describe and demonstrate the improved CNNs on different data sets and provide the results of their analysis compared with some other methods. Finally, in Sect. 5, we discuss the results and conclude the article.

2 Related work

ResNet was created to solve the performance degradation problem of traditional CNNs. It makes traditional neural networks develop in a deeper and higher performance direction. In the residual module, ResNet has one more branch than traditional neural networks. Take the two-layer residual module as an example, as shown in Fig. 2, where x denotes the input of the residual module, and $H(x)$ is the potential expectation mapping:

$$H(x) = F(x) + x \tag{1}$$

Therefore, the mapping that needs to be learned is:

$$F(x) = H(x) - x \tag{2}$$

$F(x)$ is the residual between the expected mapping and the input. The residual module is defined as follows:

$$y = F(x, \{W_i\}) + x \tag{3}$$

where x and y denote input and output, respectively, and $F(x, \{W_i\})$ represents the residual that we need to continuously learn and optimize. For the two-layer convolution example in Fig. 2 below, $F = W_2\sigma(W_1x)$, where W_1 and W_2, respectively, denote the weight parameter of the first and second convolution layer of the residual module; σ is rectified linear unit (ReLU), $\sigma(x) = \max(0, x)$. ReLU function makes the negative axis zero and provides system sparsity and generality. The x added to the back is the output of the short-circuit layer. In order to match the residual output of the learning with the output data dimension of the short-circuit layer, W_s is introduced:

$$y = F(x, W_i) + W_sx \tag{4}$$

Nowadays, the development of deep learning networks is very rapid. ResNet can be used in almost all computer

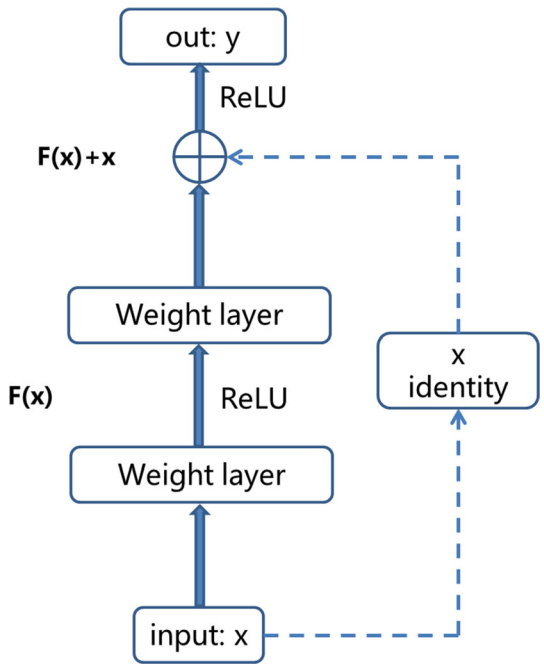

Fig. 2 Two-layer residual structure block

vision tasks. For example, for object detection, Haque et al. [6] improved the classic object detection network Single Shot MultiBox Detector (SSD) and replaced the visual geometry group (VGG) module inside the SSD with a ResNet module. For detecting different dangerous targets, the accuracy and learning rate of the network have been also improved to some extent. Lu et al. [7] proposed a combination of the classical network VGG and ResNet to solve the problem, whereby the VGG network cannot detect small objects; this combination improved the training effect. The application of object detection in the traffic [8] or the Internet of Things [9] field has helped to alleviate many traffic problems. Jung et al. [10] used ResNet to extract vehicle position and category information from surveillance videos, and they combined this information with joint fine-tuning technology (JF) to improve the accuracy of identification. Hu et al. [11] proposed an improved-depth ResNet and used this network to establish a model to predict the crowd flow on urban roads. This model can reduce the training and prediction time on the data set but does not improve the accuracy. Lu et al. [12] combined ResNet with You Only Live Once (YOLO) and used ResNet to improve YOLO's darknet feature extraction ability, which greatly improved the accuracy of the combined network in multi-object detection. Ou et al. [13] presented a network model based on ResNet that combines an encoder and a decoder. For moving objects, ResNet extracts feature information and encodes it, and after decoding it with a decoder, it makes it correspond to different targets one by one. This model has improved target positioning, but its accuracy has not improved as much. Li et al. [14] introduced ResNet to transfer learning when identifying ships in synthetic aperture radar (SAR) images, which improved the detection accuracy slightly. ResNet is also very good at virus detection in the field of computers. Rezende et al. [15] expressed a virus software with the gray value of the image and visualized it using the t-distributed stochastic neighbor embedding (t-SNE) algorithm. Then, they used the pre-trained ResNet network to identify the samples. It shows better performance than of manual feature extraction. Yu et al. [16] improved the recognition rate by introducing a ResNet network for detecting the presence of a virus in integrated chips. In faster R-CNN, a two-step detection method, Oztel [17] used the ResNet network to extract feature information and identify objects under partial occlusion. Atliha et al. [18] compared VGG and ResNet when generating image subtitles and found that the performance of the ResNet model was better than that of VGG. In semantic segmentation, the use of ResNet has achieved good results [19]. The use of ResNet in classification problems has been more common. Li et al. [20] developed a conditional random fields (CRF) model by integrating ResNet into maximum a posteriori (MAP) for hyperspectral image classification in order to obtain better neighbor class boundaries. Medical images have been classified using a network based on the ResNet structure, and good results have been obtained [21–23]. Recently, homecare robotic systems (HRS) [24, 25] based on cyber-physical systems (CPS) have entered ordinary families, providing more intelligent medical services and mobile-edge computing services [26] after applying deep learning. Zahisham et al. [27] used a framework based on ResNet-50 to achieve a certain effect on food classification on multiple food data sets. Song et al. [28] used a 41-layer ResNet to classify railway shelling and achieved better results compared with other networks. In the field of agriculture, the ResNet network structure has been used to classify crop diseases, and some good effects have been achieved [29, 30]. Firdaus et al. [31] used ResNet-50 to classify tourist attractions, and the results were close to the expected results. In text processing, ResNet has been used to extract feature information and to recognize and correct text in a number of different scenes [32, 33]. The application of deep learning in 3D reconstruction [34] has also become common. For resource management and scheduling, He et al. [35] provided a new deep learning model that can help in hierarchical scheduling, effectively increasing its efficiency in various environments. In [36], the authors used the deep learning method to allocate and schedule frequency bands to improve the system's efficiency. In the field of wireless communication, Chen et al. [37] proposed a reinforcement learning (RL) algorithm in the distributed framework to improve the performance of machine translation with a relatively long training time. Wang et al. [38] used a so-called ascending learning code to identify reflected Wi-Fi signals of human movement.

Nowadays, there are many novel ResNet applications that have been introduced in the literature, including feature extractors, pre-trained models and classifiers. However, many studies do not improve the network structure itself but rather a kind of pieced application. For most networks, the core work involves extracting feature information. Because of the strong extraction ability of ResNet, it has been widely adopted for many practical applications. The ResNet network structure contains a large number of residual modules that usually use 1*1 filters and 3*3 filters to perform convolution operations with a step size of 2 on the output feature map of the previous layer. This makes that the features output in the previous layer, i.e., approximately 75% of the pixels in the picture, is not involved in the calculation process of the convolution kernel. Therefore, the information carried by these pixels and their correlation information with other pixels in the neighborhood cannot be transmitted forward through the short connection line of the residual block, resulting in information loss. One of the most important advantages of

ResNet is realizing a deeper network. As the number of network layers increases, these feature maps have a larger part of the information missing each time they pass through the residual block, which decreases the system's accuracy. The proposed ADD is a decentralized down-sampling mechanism that re-enables the regions not involved in the convolution operation and makes them convolute with 1*1 filters or 3*3 filters. The upgraded feature map is stacked and merged with the feature map obtained by the convolution operation to avoid loss of information. Certainly, training data are also a hot research point of neural networks. In order to reduce the loss caused by repeated compression as the training data spread, Li et al. [39] used adaptive adjustment of discrete cosine transform (DCT) coefficients to generate data. The bit error rate and stronger anti-compression performance are conducive to data transmission.

3 Proposed work

In most CNNs, with an increase in the number of convolutional layers, down-sampling is adopted to reduce the feature map accordingly; that is, the step size of the filters is normally set to be greater than 1. Taking the step size of 2 as an example, normal down-sampling is shown in Fig. 3.

There are two ways to calculate the size of a feature graph in convolution, which are "valid" and "same". The "valid" method does not calculate the boundary data. This means that the original size of the feature map is not filled. The relationship between the size of the feature map newly obtained with "valid" and the size of the input feature map is as follows:

$$W = (M - F)/S + 1 \tag{5}$$

In this paper, the "same" method is adopted by default, so the convolution result of the boundary can be retained. As the step size is set to 1, the size W of the newly obtained feature graph remains unchanged. The size of the new output feature map can be obtained from the following equation:

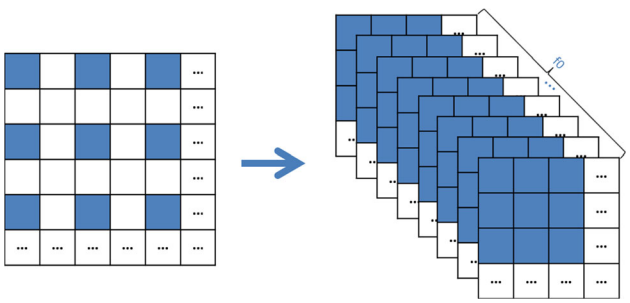

Fig. 3 Normal down-sampling of feature maps with a step size of 2

$$W = (M - F + 2 * \text{padding})/S + 1 \tag{6}$$

where M is the size of the input feature map, F is the size of the convolution kernel, padding is the number of fillings in the feature map of the "same" method, and S is the convolution step size. Assuming that the size of the feature map is $M * M$, the size of the filters is $3 * 3$, and the stride is equal to 2. As M is even, the length and width of the new output feature map become half of the original, and the overall size is only 1/4 of the original. In other words, nearly 3/4 of the associated information of the original has been lost. Meanwhile, as M is odd, the length and width of the new output feature graph are $(M + 1)/2$ of the original, and the overall size is $(M + 1)^2/4$ of the original. With the increase in the step size, the lost association information increases sharply.

In this article, we propose an improved solution called aggregated decentralized down-sampling-based ResNet (ADD-ResNet) convolution, which reuses the convolution kernel to perform convolution operations in the regions where some elements of the feature map have been skipped because the convolution step size of the filters is greater than one. After the convolution operation of multiple filters, the feature map is stacked to form a new feature map. Then, the feature map's output from the previous layer is added and fused with the new feature map to continue the forward propagation.

In this paper, we take the ResNet-18 network as an example to improve the performance of the modified ResNet model by adjusting its residual module. Moreover, three improvement schemes are used for ResNet-18. In scheme 1, only the short-circuit layer, namely Path-b, is modified; in scheme 2, only the forward propagation path, namely Path-a, is modified; and in scheme 3, two paths, Path-a and Path-b, are merged after modification; that is, out-a and out-b are added and merged. The structure of the ResNet-18 residual module is shown in Fig. 4. A comparison between the proposed ADD method and the common down-sampling method is shown in Fig. 5.

To ensure that the stacked feature map can be added and fused with the original feature map of forward propagation smoothly, the following problems need to be solved for the proposed schemes.

3.1 Depth of the feature map

When $n * n$ convolution kernels are used to perform convolution operations in all regions of the original feature map with a step size of S, the depth of the new feature map generated after stacking needs to be the same as the depth of the original feature map.

Scheme 1: In the short-circuit layer (path-b), when the transverse and longitudinal step size is set to S, there is no

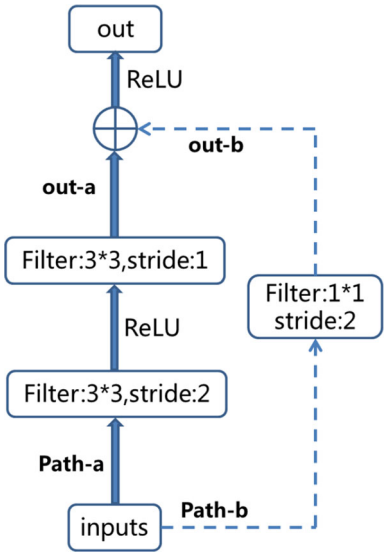

Fig. 4 The structure of each path in ResNet-18 network residual module

difference between the square feature map and the rectangular feature map using the "same" method. The number of filters in different regions is set to $1/S^2$ of the original number. Regardless of how big the original feature map is, the newly added convolution area is $S^2 - 1$ times the original one. Therefore, if the number of original filters is unchanged, the depth of the stacked feature map is S^2 times that of the original feature map. It is further concluded that when the transverse step size is S and the longitudinal step size is T, the number of filters in different regions is $1/(S * T)$ of the original, the newly added convolution area is the original $S * T - 1$ times, and the depth of the feature map after stacking is $S * T$ times the depth of the original feature map. To combine the new feature map with the original feature map, this paper proposes that for $S * T$ blocks, each block only uses $1/(S * T)$ of the original number of filters so that the total number of filters is the same as the original, and no additional calculation is included. We use a $1 * 1$ convolution and 2 as the step size as an example. The implementation is shown in Fig. 6, and the network is named ADD-ResNet-18 I. In the short-circuit layer, the convolution step size of the $1 * 1$ filter is 2, and the unconvolved area on Path-b of ADD-ResNet-18 I is calculated using the $1 * 1$ size filter from the A, B, C and D of the original feature map to start the convolution. The four results of the convolution are stacked in the depth direction to get out-b and then added and merged with out-a. The number of filters is taken as the original number 1/4. As shown in Fig. 12, $f1 = f2 = f3 = f4 = 1/4 * F$, where

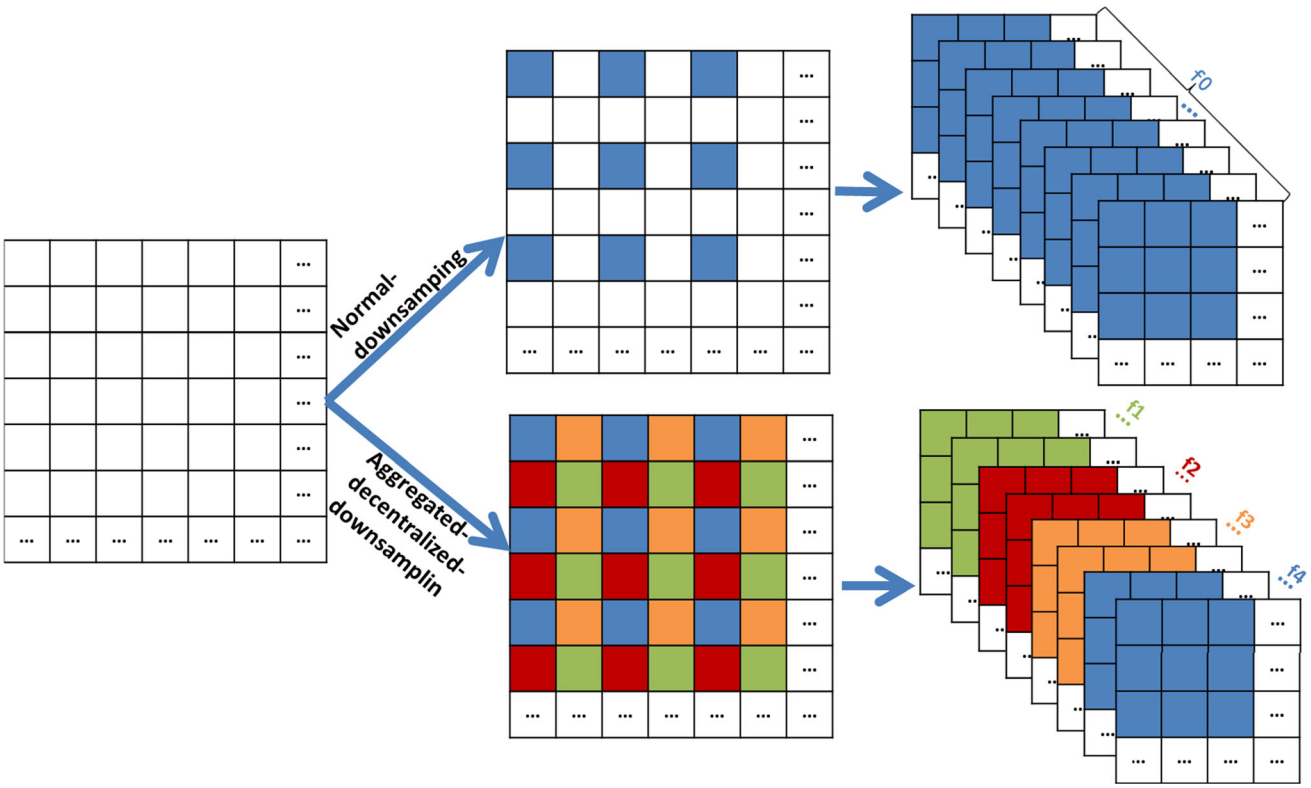

Fig. 5 Compare the aggregated decentralized down-sampling of the feature map with traditional down-sampling, where $f0$, $f1$, $f2$, $f3$ and $f4$ are the number of convolution kernels

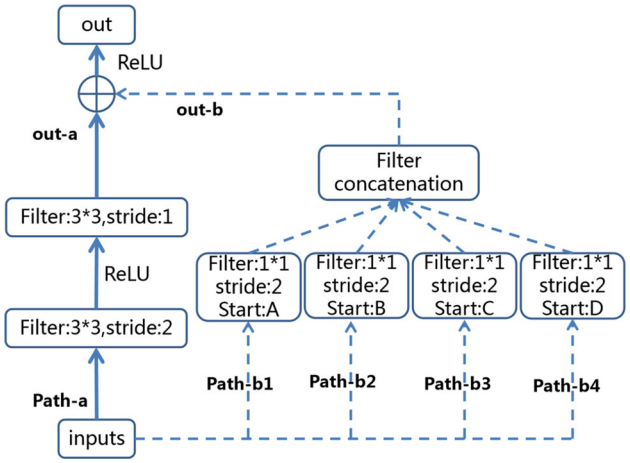

Fig. 6 ADD-ResNet-18 I, where StratA, StratB, StratC and StratD are the starting positions of the $1 * 1$ convolution kernel, as shown in Fig. 12

F is the number of filters in the corresponding layer of the ResNet and is unchanged.

In addition, for networks with more than 50 layers, the corresponding improvements in this article are shown in Fig. 7.

Scheme 2: In the main path (Path-a), because of the two layers of the convolution operation, the convolution layer with a step size greater than 1 is not the final output feature layer. Therefore, the final output depth of the new feature map is determined by the number of filters in the second layer, and the number of filters used in the convolutional layer with a step size large than 1 can theoretically be unlimited. The number of filters represents the depth of the output feature map. When the number of filters used in the first layer is large, the depth of the output feature map will be very large. In the second layer of convolution, the depth of the convolution kernel needs to be set to the depth of the output feature map of the previous layer. We use a $3 * 3$ size filter with a step size of 2 as an example and name the network ADD-ResNet-18 II. On Path-a, the unconvolved area is convolved with a $3 * 3$ size filter, the number of filters is the same as the unimproved ResNet and then stacked onto out-a; subsequently, out-a and out-b are added and fused. The implementation is shown in Fig. 8.

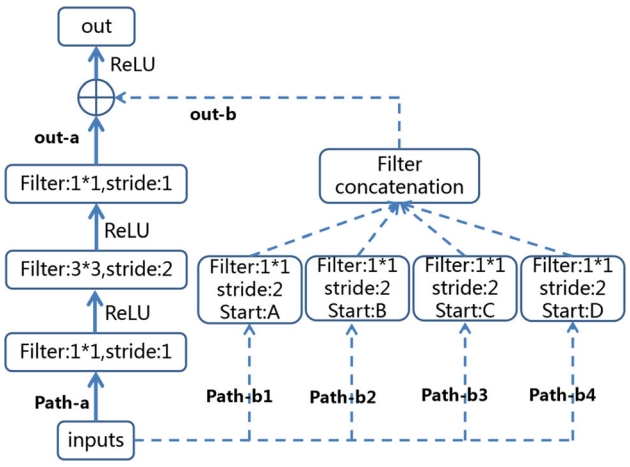

Fig. 7 The residual structure of the ResNet network with more than 50 layers, among which StratA, StratB, StratC and StratD are the starting positions of the $1 * 1$ filters, as shown in Fig. 12

And for networks with more than 50 layers, the corresponding improvements in this article are shown in Fig. 9.

Scheme 3: By combining scheme 1 and scheme 2, it is easier to obtain scheme 3, that is, to make modifications on both Path-a and Path-b. We use a $1 * 1$ size filter with a step size of 2 as an example and name the network ADD-ResNet-18 III. On Path-a, we use $3 * 3$ size filters to calculate the unconvolved area and stack them onto out-a. On Path-b, we use $1 * 1$ size filters to calculate the unconvolved area and stack them onto out-b; then, out-a and out-b are added and merged. The implementation is shown in Fig. 10.

Similarly, for networks with more than 50 layers, the corresponding improvements in this article are shown in Fig. 11.

3.2 Size of the feature map

To ensure that the size of the stacked feature map after area convolution is the same as that of the original feature map, 0 is used to complement the feature map. There are several different situations for different size feature maps and step-

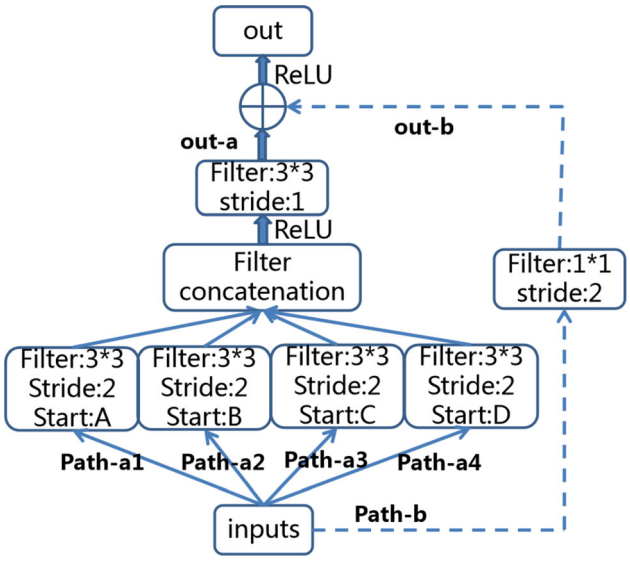

Fig. 8 ADD-ResNet-18 II, where StratA, StratB, StratC and StratD are the starting positions of the $3 * 3$ convolution kernel, as shown in Fig. 12

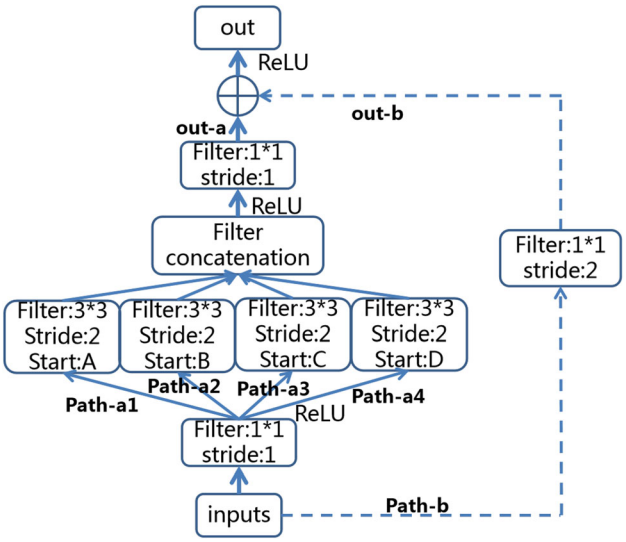

Fig. 9 The residual structure of the ResNet network with more than 50 layers, among which StratA, StratB, StratC and StratD are the starting positions of the $3 * 3$ filters, as shown in Fig. 12

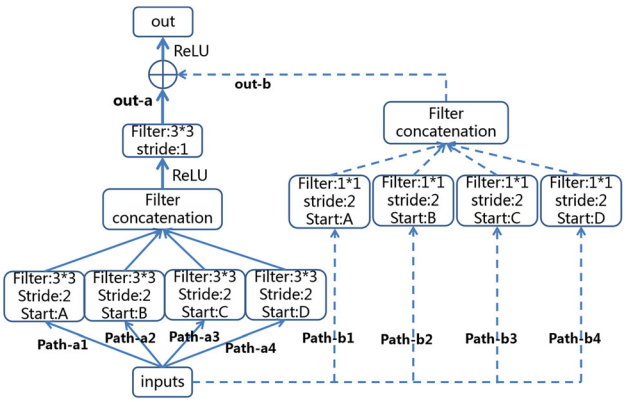

Fig. 10 ADD-ResNet-18 III, where StratA, StratB, StratC and StratD are the starting positions of the convolution kernel, as shown in Fig. 12

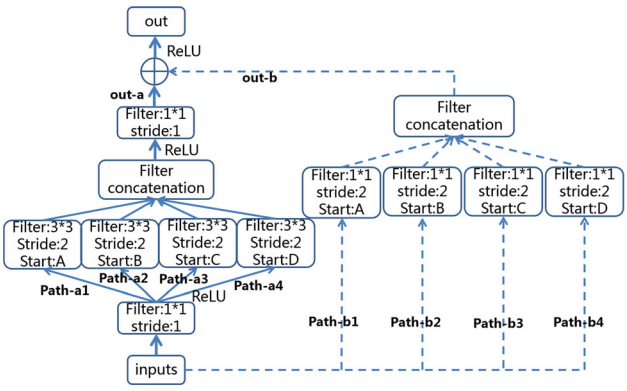

Fig. 11 Figure 11 ResNet network has more than 50 layers of residual structure, among which StratA, StratB, StratC and StratD are the starting positions of the convolution kernel, as shown in Fig. 12

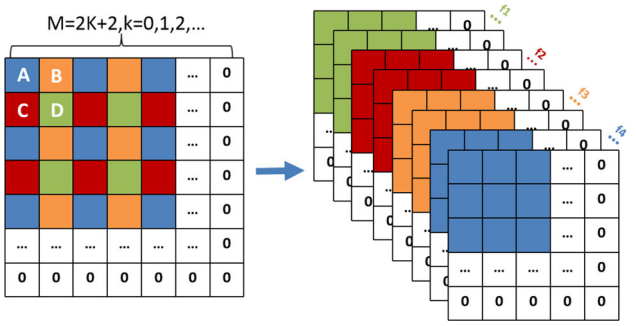

Fig. 12 When the pixel length and width of the feature map are even numbers, for strides=2, $1 * 1$ convolution kernel, the starting positions are A, B, C and D, respectively

size filters. For the feature map with size $M * H$, when the horizontal step size is S and the vertical step size is T, the 0 to be supplemented is the P_r row and the P_c column, where:

$$P_r = S - M/S \tag{7}$$

$$P_c = T - H/T \tag{8}$$

The core idea is to make up the deficiency with 0, as adding 0 will not result in new noise. If we use a $1 * 1$ size filter with a step size of 2 as an example, there will be two situations:

(1) When the length and width of the original feature map are even numbers, as shown in Fig. 12, it can be directly stacked and added for fusion without additional operations.

(2) When the original feature map has an odd length and width, as shown in Fig. 13, this paper proposes to add 0 at the end when the length or width is insufficient so that the size of the feature map after convolution of the four-part $1 * 1$ size filters is the same.

The parameters and calculations of the network are important for evaluating the performance of the network, and the calculation of the CNN is mainly reflected in the convolution operation. Here, two-dimensional convolution is used as an example.

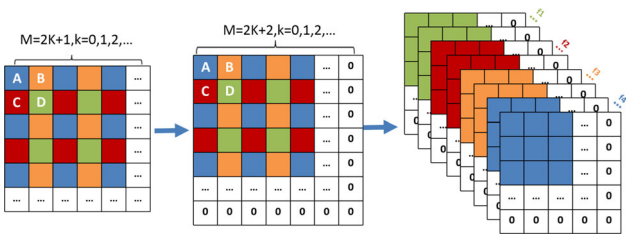

Fig. 13 When the feature map pixel length and width are odd numbers, use 0 to fill the original feature map to an even number. For stride = 2, $1 * 1$ convolution kernel, the starting position is still A, B, C and D

$$P(m,n) = f(\sum_{m=0}\sum_{n=0} w(i,j)x(m+i,n+j) + b_w) \tag{9}$$

where $p_(m,n)$ is the element in the $m-th$ row and nth column of the output feature map; f represents the activation function; $x_(m,n)$ is the element in the $i-th$ row and $j-th$ column of the input feature map; $w_(i,j)$ represents the weight in the first column of the $i-th$ row, and b_w is the bias term. We assume that the input channel is M, the feature map output channel is N, the convolution kernel is $k*k$, the bias is not used, and the number of parameters Pa is:

$$Pa = k*k*M*N \tag{10}$$

The number of parameters of the network optimized in this article and the original ResNet-18. Here, only the optimized residual module is calculated as follows:

$$\text{ResNet}-18:$$

$$Pa_0 = 3*3*M_1*N_1 + 3*3*M_2*N_2 + 1*1*M_s*N_s \tag{11}$$

$$\text{ADD}-\text{ResNet}-18\ I:$$

$$Pa_1 = 3*3*M_1*N_1 + 3*3*M_2*N_2 + 1*1*((M_s-A)/4 + (M_s-B)/4 + (M_s-C)/4 + (M_s-D)/4)*N_s \tag{12}$$

$$\text{ADD}-\text{ResNet}-18\ II:$$

$$Pa_2 = 3*3*((M_1-A)/4 + (M_1-B)/4 + (M_1-C)/4 + (M_1-D)/4)*N_1 + 3*3*M_2*N_2 + 1*1*M_s*N_s \tag{13}$$

$$\text{ADD}-\text{ResNet}-18\ III:$$

$$Pa_3 = 3*3*((M_1-A)/4 + (M_1-B)/4 + (M_1-C)/4 + (M_1-D)/4)*N_1 + 3*3*M_2*N_2 + 1*1*M_s*N_s + 1*1*((M_s-A)/4 + (M_s-B)/4 + (M_s-C)/4 + (M_s-D)/4)*N_s \tag{14}$$

where M_i and N_i represent the input channels and output channels of the $i-th$ layer of the residual module, respectively, M_s and N_s are the input and output channels of the short-circuit layer, M_i-K and N_i-K are the different positions of the input channel and the output channel, as shown in Fig. 12. $(M_i-K)/4$ indicates that the number of channels in different positions is set to 1/4 of the unimproved ResNet. The required calculation FLOPs of the optimized network and original ResNet-18 are as follows:

$$\text{FLOPs} = \text{number\_of\_parameters} * (H*W) \tag{15}$$

where $H*W$ represents the size of the output feature map, here $H = W$, and the value is Equation 6:

$$\text{FLOPs} = Pa_i * (W*W) \tag{16}$$

Algorithm 1 classification algorithm for proposed aggregated decentralized down-sampling

Input: Training data set D,a batch of images from a data set X;validation data set V,a batch of images from a data set Y

Output: Image classification results

```
Input data preprocessing and enhancement
initialization
for X in T do
    if residual block is Trure then
        if size of feature map is odd then
            padding the feature map with 0;
        end if
        The first convolution layer:
            O_A ← convolution on X_A;
            O_B ← convolution on X_B;
            O_C ← convolution on X_C;
            O_D ← convolution on X_D;
            O_1 ← stack O_A, O_B, O_C and O_D;
        The second convolution layer:
            O_2 ← convolution on O_1;
        The residual layer:
            R_A ← convolution on X_A;
            R_B ← convolution on X_B;
            R_C ← convolution on X_C;
            R_D ← convolution on X_D;
            R_1 ← stack R_A, R_B, R_C and R_D;
        The out of residual block:
            R_O ← R_1 add O_2;
        return result R_O
    else
        The first convolution layer:
            O_1 ← convolution on X;
        The second convolution layer:
            O_2 ← convolution on O_1;
        The short-cut layer:
            S_1 ← X;
        The out of short-cut block:
            S_O ← O_2 add S_1;
        return result S_O
    end if
end for
```

The ADD-ResNet-18 III algorithm is described as algorithm 1, where X_A, X_B, X_C *and* X_D represent the convolution positions, respectively, as shown in Fig. 12.

When scheme 1 proposed in this paper is used in the residual block, the number of filters used in different regions is reduced. So the total number of filters does not change, and the number of parameters and calculation excess does not change. As Scheme 2 is used in the residual block, the number of parameters increases slightly. Since the convolution layer with a step size of 2 does not reduce the number of filters for different regions of the feature

map in this paper. In addition, there are only a few modules linked by residuals in the whole network, and thus, the increased calculational complexity over all calculation amount is very limited on one iteration process. In Scheme 3, the number of parameters and the amount of calculation are the same with that of Scheme 2, as the short-circuit layer is the same as in Scheme 1.

4 Results and discussion

The following is the evaluation of the three improvement schemes proposed in this paper. The original network ResNet-18 and the three improved networks were all tested on the CIFAR10 [40] dataset. After using data enhancement for CIFAR10, we set the network batch size to 128 and used $3 * 3$ filters with a step size of 1 to replace the $7 * 7$ filters and pooling layer of the input layer. In order to increase the training speed, 32 initial initialization filters are used, which is half less than the traditional ResNet. Although this will reduce the accuracy of some networks, this article wants to show the improvement of network performance after aggregated decentralized down-sampling. The final fully connected layer uses L1 regularization and performs 300 rounds of training on the data set CIFAR10. As the number of training epochs increased, the accuracy of the validation set stabilized gradually.

We use accuracy (Acc) to describe the performance of the network; the calculation method is as follows:

$$\text{Acc} = \frac{\text{Correct validation data}}{\text{All validation data}} * 100\% \tag{17}$$

The final result is activated by the softmax function and output with L2 regularization, and the loss function uses the cross-entropy loss function:

$$\text{softmax:} \quad S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{m} e^{x_j}} \tag{18}$$

where x_i is the output of the i-th category in the previous layer, m is the total number of categories, e^{x_i} is the output value of the i-th neuron, and $\frac{e^{x_i}}{\sum_{i=1}^{m} e^{x_i}}$ is the probability value of the output of the i-th neuron.

$$L2: \quad \Omega(w) = \|w\|_2^2 = \sum_i w_i^2 \tag{19}$$

$$\text{crossentropy loss} = -\frac{1}{k}\sum_{i=1}^{k} p_i log(q_i) \tag{20}$$

From the above, the total loss function has been calculated as:

$$\text{all loss} = -\frac{1}{k}\sum_{i=1}^{k} p_i log(q_i) + \lambda R(w) \tag{21}$$

where $q_i = S(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{m} e^{x_i}}$, $R(w) = \Omega(w)$, λ is the configuration coefficient, and k is the number of samples.

The average value of the training data in the last 30 epochs was taken to reduce the error. After multiple trainings, the average value was taken as the final result, as shown in Table 1.

Table 1 The performance comparison of four kinds of networks on CIFAR10

Networks	Average Acc (%)	Average loss
ResNet18	81.59	0.7227
ADD − ResNet − 18 I	82.02	0.6947
ADD − ResNet − 18 II	83.46	0.6775
ADD − ResNet − 18 III	84.16	0.6448

Table 1 shows that compared with the original unimproved ResNet-18 network, the performance of the three schemes gradually improved. Compared with the unimproved ResNet-18, the accuracy of the improved ADD-ResNet-18 I verification set increased by 0.43%. The improved network extracted the information lost by the unimproved network in the short-circuit layer (Path-b) and reduced the loss from 0.73 to 0.69. In the forward propagation down-sampling block (Path-a), the performance of the distributed down-sampling network ResNet-18 II proposed in this paper was further improved. The accuracy of the validation set increased by 1.87% compared with the original ResNet-18, and the loss value was also further reduced. This is because, at this time, scattered down-sampling extracts more interrelated information between pixels, which reflects the importance of various features of the picture; therefore, the improvement is greater. To make the effect clearer, we made a histogram, which is shown in Figs. 14, 15.

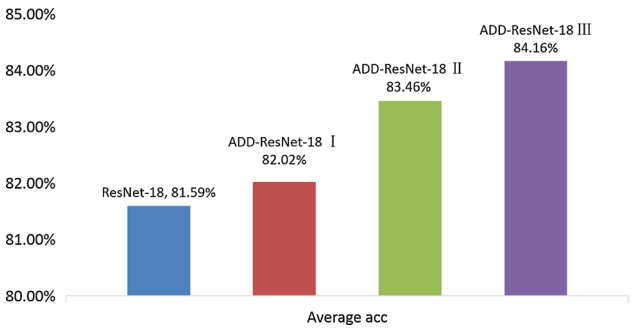

Fig. 14 Performance comparison of average ACC of varying networks on CIFAR10

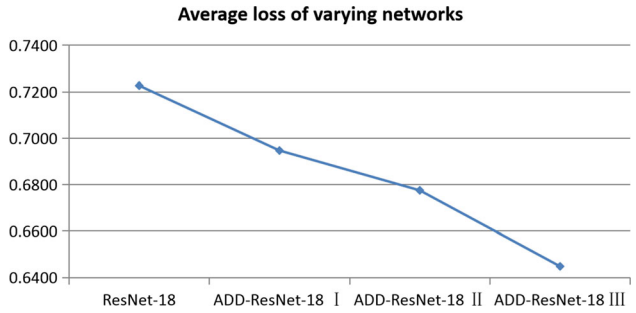

Fig. 15 Performance comparison of average loss of various networks on CIFAR10

Taking into account the characteristics of ResNet, the previous information can be forwarded through the short-circuit module. Therefore, a third solution was proposed to make the improved network ResNet-18 III continue to improve the effect. Compared with the unimproved ResNet-18, the validation set accuracy increased by 2.57%, which is 0.7% higher than that of ResNet18 II, and the loss was reduced to 0.6448. Based on ResNet-18 II, ResNet-18 III receives more comprehensive information from the short-circuit module so that the final characteristic information is further strengthened and the effect is better. The loss value also reflects the improvement in the performance of the optimized network in this paper.

Considering that medical diseases are sometimes not only classified as benign and malignant but also into many types and periods, there may be more classification requirements. Therefore, we also conducted experiments on the data set CIFAR100 [40], which contains 100 different categories. The results are shown in Table 2 below.

It can be seen from Table 2 that the effects of the three improved schemes proposed in this paper are better than these without improvement. Among the three improvement schemes, the third one shows the most improvement (2.60%), and its loss value is also the smallest, as shown in Fig. 17. To visualize the accuracy of the validation set, we constructed a bar chart, which is shown in Fig. 16.

The line chart of each network loss value is shown in Fig. 17.

Table 2 Performance comparison of four kinds of networks on CIFAR100

Networks	Average Acc (%)	Average loss
ResNet − 18	60.34	2.1494
ADD − ResNet − 18 I	60.77	2.1070
ADD − ResNet − 18 II	61.88	2.0885
ADD − ResNet − 18 III	62.94	2.0056

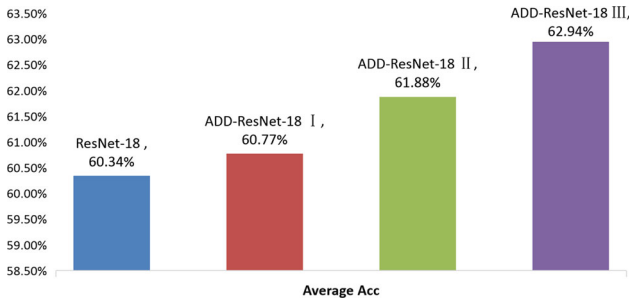

Fig. 16 Performance comparison of average ACC of various networks on CIFAR100

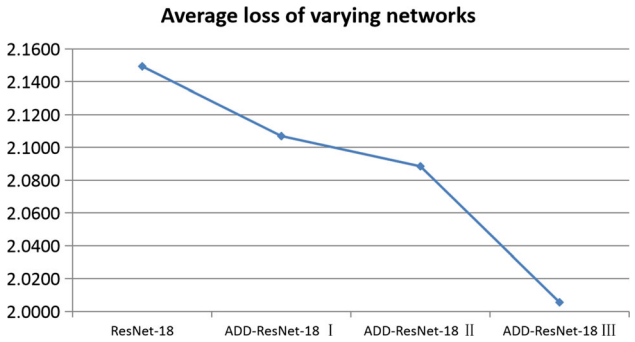

Fig. 17 Performance comparison of average loss of various networks on CIFAR100

Table 3 The performance of the ResNet D and ADD-ResNet D networks was compared on the CIFAR10 data sets

Networks	Average Acc (%)	Average loss
Resnet − 18 *D*	81.90	0.7301
ADD − ResNet − 18 *D*	83.06	0.7168
Resnet − 34 *D*	84.52	0.6458
ADD − Resnet − 34 *D*	85.31	0.6257

Table 4 The performance of the ResNet D and ADD-ResNet D networks was compared on the CIFAR100 data sets

Networks	Average Acc (%)	Average loss
Resnet − 18 D	60.82	2.1416
ADD − ResNet − 18 D	62.70	2.0533
Resnet − 34 D	62.56	2.0708
ADD − Resnet − 34 D	63.63	2.0055

In addition, we further introduced an improved network (ResNet D) in [41]. We used the aggregation decentralized down-sampling proposed in this paper on this improved network to further improve its performance. The results are shown in Tables 3 and 4 below.

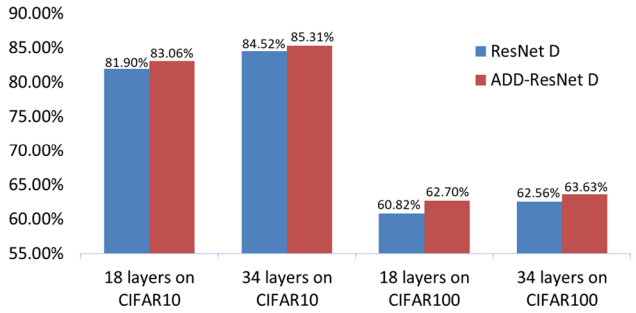

Fig. 18 The results of 18-layer and 34-layer ResNet D and Add-ResNet D on CIFAR10 and CIFAR100, respectively

The ResNet D uses a $2 * 2$ size pooling layer of strides $= 2$ to replace the $1 * 1$ size convolution layer of path-b, which improves the performance of the network. However, it does not operate on path-a, so on the basis of ResNet D, we can further use aggregation decentralized down-sampling on path-a of ResNet D, named Add-ResNet D. We tested the 18-layer and 34-layer networks on CIFAR10 and CIFAR100, respectively, and the results are shown in Table 3. It can be seen that Add-ResNet D further improves the effect of ResNet D, and its histogram is shown in Fig. 18 below.

Finally, we conducted experiments on the medical image data set Ocular Disease Intelligent Recognition (ODIR-2019) [42]. In ophthalmology, there are almost no symptoms of ocular disease in the early stages. Fundus screening is a cost-effective method for early detection of ocular diseases and prevention of visual impairment caused by other diseases. The dataset contains color photographs of the left and right fundi of 5000 patients, divided into eight labels. However, due to the large number of patients suffering from two or more eye diseases, there is a large number of images corresponding to two or more labels, and the direct use of classification 8 is of little significance. This experiment will focus on cataract, an eye disease, by first selecting all the pictures labeled as "cataract" in the data set and then randomly selecting a certain number of pictures from the remaining seven labels to form the data set. The experimental results are shown in Table 5.

Table 5 The performance of the ResNet D and ADD-ResNet D networks compared on the CIFAR100 data sets

Networks	Average Acc (%)	Average loss
Resnet − 18	91.96	0.2896
ADD − ResNet − 18 I	93.86	0.2321
ADD − ResNet − 18 II	93.34	0.2472
ADD − ResNet − 18 III	94.32	0.2163

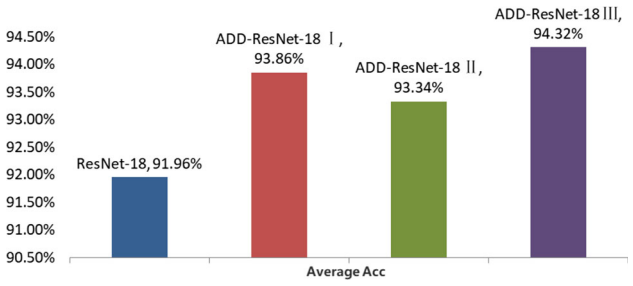

Fig. 19 Performance comparison of average ACC of various networks on ODIR

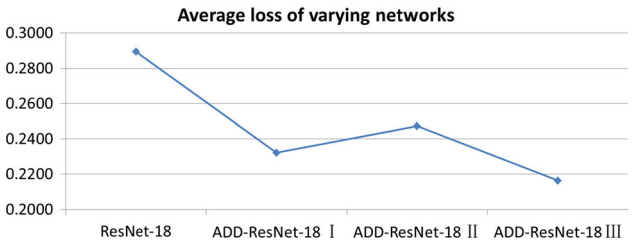

Fig. 20 Performance comparison of average loss of various networks on ODIR

As can be seen from Table 5, the recognition accuracy of the three schemes proposed in this paper is still higher than that of the unimproved ResNet network with the medical image data set, but the three proposed schemes have their own advantages with different data sets. The histogram is shown in Fig. 19, and the line chart of the loss value of each network on the ODIR data set is shown in Fig. 20.

In general, compared with the original ResNet-18, the three improved schemes achieved the same level of accuracy several epochs earlier. From ResNet-18 I to ResNet-18 III, the overall accuracy of the verification set fluctuates gradually. The loss of the verification set gradually declines more rapidly, and finally, ResNet-18 III has the highest accuracy and the smallest loss. For ADD-ResNet II and ADD-ResNet III, using the method proposed in this article in the down-sampling layer does not reduce the filters while increasing the sampling area, so there will be a slight increase in the amount of calculation, and ResNet itself has only a small number of down-sampling layers. Therefore, the actual experiment running time barely changes.

5 Conclusion and future works

This paper uses ResNet as an example to illustrate the loss of image feature information caused by the use of a step size greater than 1 in CNNs to reduce the feature map. Each pixel in the image does not exist independently. They all contain information about their relationship with the

surrounding pixels, which is important. If the information is skipped directly, the network will lose an increasing amount of information as the number of layers increases. Xie et al. [43] used dozens of similar filters to extract features from the same graph in order to extract more information. In this paper, we propose to reuse these skipped areas because of the large step size and transform them into the depth dimension of the feature map, which can improve the feature extraction and reduce the feature map at the same time. In addition, although the method in this paper increases the area of down-sampling, the number of filters used in different regions is reduced, so the total number of convolutions can be guaranteed to be the same, and thus, no additional calculation is required. This can have an important impact in medical image classification. This article only uses ResNet as an example to show the impact of the method proposed in this article on model performance. The proposed method can be applied to all networks that use down-sampling. In the future, we will also evaluate our method using high-resolution medical images and attempt to maintain classification accuracy with medical images while rapidly shrinking the feature map.

Acknowledgements This work was supported in part by the National Nature Science Foundation of China under Grants 61971182 and 61771191 and in part by Changsha City Science and Technology Department Funds under Grants CSKJ2019-08 and CSKJ2020-12.

Declarations

Conflict of interest The authors declared that they have no conflicts of interest to this work.

References

1. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791
2. Szegedy C, Wei L, Yangqing J, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015:1–9. https://doi.org/10.1109/CVPR.2015.7298594
3. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Computer Science. https://arxiv.org/abs/1409.1556
4. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016:770–778. https://doi.org/10.1109/CVPR.2016.90
5. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: 31st AAAI Conference on Artificial Intelligence 4278–4284
6. Haque MF, Lim H, Kang D (2019) Object detection based on VGG with ResNet network. In: 2019 International Conference on Electronics, Information, and Communication (ICEIC) 1–3. https://doi.org/10.23919/ELINFOCOM.2019.8706476
7. Lu X, Kang X, Nishide S, Ren F (2019) Object detection based on SSD-ResNet. In: 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS) 89–92. https://doi.org/10.1109/CCIS48116.2019.9073753
8. Yu K, Lin L, Alazab M et al (2020) Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2020.3042504
9. Zhen L, Bashir AK, Yu K et al (2020) Energy-efficient random access for LEO satellite-assisted 6g internet of remote things. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2020.3030856
10. Jung H, Choi M, Jung J, Lee J, Kwon S, Jung WY (2017) ResNet-based vehicle classification and localization in traffic surveillance systems. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2017:934–940. https://doi.org/10.1109/CVPRW.2017.129
11. Hu X, Dai G, Ge Y, Ning Z, Liu Y A (2018) Simplified deep residual network for citywide crowd flows prediction. In: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG) 60–67. https://doi.org/10.1109/SKG.2018.00016
12. Lu Z, Lu J, Ge Q, Zhan T (2019) Multi-object detection method based on YOLO and ResNet hybrid networks. In: 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM) 827-832. https://doi.org/10.1109/ICARM.2019.8833671
13. Ou X, Yan P, Zhang Y, Tu B, Zhang G, Wu J, Li W (2019) Moving object detection method via ResNet-18 with encoder-decoder structure in complex scenes. IEEE Access 7:108152–108160. https://doi.org/10.1109/ACCESS.2019.2931922
14. Li Y, Ding Z, Zhang C, Wang Y, Chen J (2019) SAR ship detection based on resnet and transfer learning. In: IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium 1188–1191. https://doi.org/10.1109/IGARSS.2019.8900290
15. Rezende E, Ruppert G, Carvalho T, Ramos F, Geus Pd (2017) Malicious software classification using transfer learning of ResNet-50 deep neural network. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) 1011–1014. https://doi.org/10.1109/ICMLA.2017.00-19
16. Yu WZ, Wen YM, Huang XQ (2019) ResNet-based Trojan detection methodology for protected ICs. Electron Lett 55(21):1116–1118. https://doi.org/10.1049/el.2019.2225
17. Oztel I (2020) Human detection system using different depths of the Resnet-50 in Faster R-CNN. In: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 1–5. https://doi.org/10.1109/ISMSIT50672.2020.9255109
18. Atliha V, Šešok D, (2020) Comparison of VGG and ResNet used as encoders for image captioning. In: 2020 IEEE Open Conference of Electrical. Electronic and Information Sciences (eStream) 1–4. https://doi.org/10.1109/eStream50540.2020.9108880
19. Cui Z, Zhang Q, Geng S, Niu X, Yang J, Qiao Y (2017) Semantic segmentation with multi-path refinement and pyramid pooling dilated-resnet. IEEE Int Conf Image Process (ICIP) 2017:3100–3104. https://doi.org/10.1109/ICIP.2017.8296853
20. Li K, Ma Z, Xu L, Chen Y, Ma Y, Wu W, Wang F, Liu Z (2020) Depthwise separable ResNet in the MAP framework for hyperspectral image classification. IEEE Geosci Remote Sens Lett. https://doi.org/10.1109/LGRS.2020.3033149
21. Jeyaraj PR, Nadar ERS, Panigrahi BK (2019) ResNet convolution neural network based hyperspectral imagery classification for accurate cancerous region detection. IEEE Conf Inform Commun Technol 2019:1–6. https://doi.org/10.1109/CICT48419.2019.9066215

22. Gouda N, Amudha J (2020) Skin cancer classification using ResNet. In: 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA) 536–541. https://doi.org/10.1109/ICCCA49541.2020.9250855
23. Reddy ASB, Juliet DS (2019) Transfer learning with ResNet-50 for malaria cell-image classification. Int Conf Commun Signal Process (ICCSP) 2019:0945–0949. https://doi.org/10.1109/ICCSP.2019.8697909
24. Yang G, Pang Z et al (2020) Homecare robotic systems for healthcare 4.0: visions and enabling technologies. IEEE J Biomed Health Inform 24(9):2535–2549. https://doi.org/10.1109/JBHI.2020.2990529
25. Yu K, Tan L, Lin L, Chen X, Zhang and Sato T (2011) Deep learning empowered breast cancer auxiliary diagnosis for 5GB remote E-Health. IEEE Wireless Communications
26. Chen X, Zhang H, Wu C, Mao S, Ji Y, Bennis M (2019) Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. IEEE Internet Things 6(3):4005–4018. https://doi.org/10.1109/JIOT.2018.2876279
27. Zahisham Z, Lee CP, Lim KM (2020) Food recognition with ResNet-50. In: 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) 1–5. https://doi.org/10.1109/IICAIET49801.2020.9257825
28. Song X, Chen K, Cao Z (2020) ResNet-based image classification of railway shelling defect. In: 2020 39th Chinese Control Conference (CCC) 6589–6593. https://doi.org/10.23919/CCC50068.2020.9189112
29. Kumar V, Arora H, Harsh Sisodia J (2020) ResNet-based approach for detection and classification of plant leaf diseases. Int Conf Electron Sustain Commun Syst (ICESC) 2020:495–502. https://doi.org/10.1109/ICESC48915.2020.9155585
30. Hu W, Fan J, Du Y, Li B, Xiong N, Bekkering E (2020) MDFC-ResNet: an agricultural IoT system to accurately recognize crop diseases. IEEE Access 8:115287–115298. https://doi.org/10.1109/ACCESS.2020.3001237
31. Firdaus NM, Chahyati D, Fanany MI (2018) Tourist attractions classification using ResNet. Int Conf Adv Comput Sci Inform Syst (ICACSIS) 2018:429–433. https://doi.org/10.1109/ICACSIS.2018.8618235
32. Zhu X, Jiang Y, Yang S, Wang X, Li W, Fu P, Wang H, Luo Z (2017) Deep residual text detection network for scene text. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) 807–812. https://doi.org/10.1109/ICDAR.2017.137
33. Huang Z, Zhang Q (2019) Skew correction of handwritten chinese character based on ResNet. Int Conf High Perform Big Data Intell Syst (HPBD&IS) 2019:223–227. https://doi.org/10.1109/HPBDIS.2019.8735469
34. Zhang J, Yu K, Wen Z et al (2020) 3D reconstruction for motion blurred images using deep learning-based intelligent systems. CMC Comput Mater Contin 66(2):2087–2104. https://doi.org/10.32604/cmc.2020.014220
35. He L, Ota K, Dong MX (2019) Deep reinforcement scheduling for mobile crowdsensing in fog computing. Acm Trans Internet Technol 19(2):1–18
36. Chen X, Wu C, Chen T et al (2020) Age of information aware radio resource management in vehicular networks: a proactive deep reinforcement learning perspective. IEEE Trans Wirel Commun 19:2268–2281. https://doi.org/10.1109/TWC.2019.2963667
37. Chen X, Wu C , Liu Z et al (2020) Computation offloading in beyond 5g networks: a distributed learning framework and applications. IEEE Wireless Commun. arXiv.org/abs/2007.08001
38. Wang FX, Gong W, Liu JC et al (2020) Channel selective activity recognition with WiFi: a deep learning approach exploring wideband information. IEEE Trans Netw Sci Eng 7(1):181–192. https://doi.org/10.1109/tnse.2018.2825144
39. Li FY, Wu K, Qin C et al (2020) Anti-compression JPEG steganography over repetitive compression networks. Signal Process 170:12. https://doi.org/10.1016/j.sigpro.2020.107454
40. Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, (2009)
41. He T, Zhang Z, Zhang H, Zhang Z, Xie J, Li M (2019) Bag of tricks for image classification with convolutional neural networks. IEEE/CVF Conf Comput Vis Pattern Recognit (CVPR) 2019:558–567
42. https://odir2019.grand-challenge.org/dataset/
43. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. IEEE Conf Comput Vis Pattern Recognit (CVPR) 2017:5987–5995. https://doi.org/10.1109/CVPR.2017.634