Fully Convolutional Neural Network Structure and Its Loss Function for Image Classification

QIUYU ZHU, (Member, IEEE), AND XUEWEN ZU
School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
Corresponding author: Qiuyu Zhu (zhuqiuyu@staff.shu.edu.cn)

ABSTRACT The overall structure of a convolutional neural network classifier includes multiple convolutional layers and one or more linear layers. Due to the fully connected characteristics of linear layer networks, there are usually many parameters, which may easily lead to local optimization and over-fitting of the model. Therefore, its indispensability in Convolutional Neural Networks (CNNs) classifier is questionable. At the same time, the excessive number of output features of the convolutional layer will also lead to the curse of dimensionality. After verifying the redundancy characteristics of the final activation function, linear layer, and the number of channels by eigen-values analysis of latent features and experiments, we propose a Fully Convolutional Neural Network (FCNN) classifier architecture, which removes the linear layers and the corresponding activation functions from the conventional CNN classifiers. By modifying the number of output channels of the last convolutional layer, the network can be trained directly through Softmax Loss. Furthermore, a softmax-free loss (POD Loss) based on Predefined Optimal-Distribution of latent features is adopted instead of Softmax Loss to obtain better recognition performance. Experiments on multiple commonly used datasets and typical networks have proved that the network structure not only reduces the amount of parameters and calculation, but also improves the recognition rate. In the meanwhile, the adoption of POD Loss further improves the classification accuracy and robustness of the model, making them play a better synergy. Code is available in https://github.com/TianYuZu/Fully-Convolutional-Network.

INDEX TERMS Fully convolutional neural network (FCNN), loss function, image classification, fully connected layer, curse of dimensionality, ReLU function.

I. INTRODUCTION
In the past few years, Convolutional Neural Networks (CNNs) [1] have achieved excellent performance in the field of classification, such as image classification and face recognition. To improve the prediction performance of CNNs, researchers have proposed many effective solutions, including data enhancement, batch normalization, improving the loss function and improving the network structure. Among them, data enhancement focuses on the processing of datasets, and batch normalization pays attention to the distribution of input data and weight parameters in the network. The improvement direction of loss function [2]–[4] tends to constrain the differences between classes and similarities within classes, which means how to make the features of images of the same class more similar and the features of images of different classes more distant. The development of the network structure is also very rapid, starting from AlexNet [5] to VGGNet [6], and then to the deeper ResNet [7], DenseNet [8], and so on. CNNs contain more and more layers, leading to more and more parameters and calculation requirement of the network model, as well as higher training time and training costs.

The nonlinear function is introduced as the activation function in the neural network, which makes the deep neural network more powerful. Compared with other activation functions, the Rectified Linear Unit (ReLU) [9] activation function has incomparable advantages: the network training speed is fast and the vanishing gradient problem is solved, so the ReLU function is currently the most commonly used activation function, and it is generally preferred to try when building a neural network. However, the value of the negative semi-axis of the ReLU function is 0, which means that some of the information is lost.

CNN classifiers generally include convolutional layers and linear layers. Various convolutional layers extract features from complex data, and then the linear layer performs dimensionality reduction and classification. If the linear layer has
multiple layers and there is an activation function between layers, it is essentially a nonlinear transformation. What is the difference between its nonlinearity and that of the convolutional layer? If there is only one linear layer, then the linear layer is a linear transformation of hidden features, and convolutional layer and the pooling layer can also play a similar role. In this way, the indispensability of the linear layer is questionable.

In this paper, for the CNN used for classification, we propose to remove the ReLU activation function of the last convolutional layer to make the training information more complete, modify the last convolutional layer and discard linear layers, which turns the network to be a Fully Convolutional Neural Network (FCNN). The structure of CNN classifier is shown in FIGURE 1, and the structure of modified FCNN classifier is shown in FIGURE 2. Furthermore, the loss function is changed from Softmax Loss to a softmax-free loss (POD Loss) [10] based on Predefined Optimal-Distribution of latent features, as shown in FIGURE 3. The details of the method are given in Section III.

Our main contributions are as follows:

- The redundancy characteristics of the final activation function, linear layer, and the number of channels is verified by the eigen-value analysis of latent features and experiments.
- A FCNN classifier structure is proposed, which uses the last convolutional layer and global average pooling layer to perform feature dimensionality reduction. The performance, number of parameters and computational complexity of CNN structure and FCNN structure are compared. Experiments show the effectiveness of the proposed FCNN structure.
- The FCNN is combined with POD Loss [10]. Experiments show the adoption of POD Loss improves the accuracy and robustness of our FCNN structure.

II. RELATED WORK

A. STRUCTURE OF CONVOLUTIONAL NEURAL NETWORK

CNN is first proposed by Lecun et al. [1], and has been developed rapidly in recent years. AlexNet [5] proposed in 2012 includes 5 layers of convolutional layers and 3 layers of fully connected layers. The total number of parameters reaches 60M. The non-linear activation function ReLU is used in the network structure to improve the learning rate, and dropout and data augmentation are proposed to prevent overfitting.

VGGNet [6] proposed in 2014 pushes the depth to 16 — 19 weighted layers. The network uses a smaller receptive window size to add more convolutional layers to stabilize the depth of the network. The network contains 3 fully connected layers, with a total parameter amount of 140M.

However, as the network depth increases, the accuracy decreases rapidly after reaching saturation, and more layers to the appropriate depth model will lead to higher training errors. ResNet [7] proposes that introducing a deep residual learning framework to solve the degradation problem. Compared with VGGNet [6], ResNet network model has fewer convolution kernels and lower complexity. For example, the 34-layer benchmark of ResNet34 has 3.6 billion flops (multiplication and addition operations), which is only 18% of VGG19 (19.6 billion flops). DenseNet [8] proposes dense connection, which alleviates the problem of vanishing gradient and has a certain inhibitory effect on overfitting.

Although the recognition performance of above networks is quite good, the amount of parameters and calculation of the models are very large, and it is not suitable for running on mobile terminals and embedded devices. The MobileNet [11] proposes to use depthwise separable convolutions and introduced hyperparameters to reduce the amount of parameters and calculation without affecting the accuracy rate, while the core of ShuffleNet [12] is using ShuffleNet unit to reduce the amount of calculation and improve the accuracy.

The improvement of the performance of deep learning models relies on manual fine adjustments. Neural Architecture Search automates this parameter adjustment process to obtain the optimal network structure. NASNet [13] is a network structure based on Neural Architecture Search, which surpasses the classical structure of manual design in terms of accuracy and speed.

In all the networks described above, features are mostly extracted from convolutional layers, and dimensionality reduction and classification are carried out by fully connected layer, which is indispensable. Some researchers have proposed the network structure of fully convolutional neural network for object detection, image segmentation, image denoising and so on. He et al. [14] discard the non-convolutional portion of detection nets to make a feature extractor, which combines proposals and spatial pyramid pooling to yield a localized, fixed-length feature for classification. Although it is fast and efficient, the model cannot be learned end-to-end. Long et al. [15] once proposed a fully convolutional neural network for semantic segmentation. Since the input and output resolutions of the fully connected layer remain unchanged, the fully connected layer can be removed. The fully convolutional neural network improves the accuracy by transferring the weight of the classifier, fusing the representations of different layers and learning the entire image.
Q. Zhu, X. Zu: FCNN Structure and Its Loss Function for Image Classification

end-to-end. In the task of image denoising, Zhang et al. [16] proposed a fast and flexible denoising convolutional neural network, namely FFDNet. The FFDNet, without fully connected layer, works on downscaled sub-images, achieving a good trade-off between inference speed and denoising performance.

B. LOSS FUNCTION OF CONVOLUTIONAL NEURAL NETWORK

Loss function is an indispensable part of CNN model. The traditional Softmax loss function is composed of softmax plus cross-entropy loss function. Because of its advantages of fast learning speed and good performance, it is widely used in image classification. The Softmax loss function is as follows:

$$L_{\text{Softmax}} = \frac{1}{N} \sum_{i=1}^{N} -\log \frac{e^{z_{y_i}}}{\sum_{j} e^{z_j}}$$  \hspace{1cm} (1)

where $N$ represents the number of samples, $z_{y_i}$ represents the output value of the last fully connected layer of the correct class $y_i$, and $z_j$ is the output value of the last fully connected layer of the $j$-th class.

However, Softmax loss function adopts an inter-class competition mechanism, only cares about the accuracy of the prediction probability of the correct label, and ignores the difference of the incorrect label. Some subsequent loss functions improve softmax, such as L-Softmax [17], AM-Softmax [4], etc. These loss functions improve the classification accuracy in the field of face recognition (the inter-class distance is relatively close), but in general image classification (the inter-class distance is farther), the effect of Softmax Loss is still the best. POD Loss [10] discards the constraints on the posterior probability in the traditional loss function, and only restricts the extracted sample features to achieve the optimal distribution of latent features, including the cosine distance between sample feature vector and predefined evenly-distributed class centroids (PEDCC) [18], [19], the decorrelation mechanism between sample features, and finally the classification through the solidified PEDCC layer. The final POD Loss is written as:

$$L_{\text{POD}} = \frac{1}{N} \sum_{i=1}^{N} (1 - \cos \theta_{y_i})^2 + \lambda \sum_{i}^{n} \sum_{j \neq i}^{n} R_{ij}^2$$  \hspace{1cm} (2)

where $N$ represents the number of samples, and $\cos \theta_{y_i}$ is the cosine value of the angle between the sample feature and its own predefined central feature. $n$ represents feature dimension, $R$ represents the self-correlation matrix of the difference matrix formed by the difference between the sample features and the predefined central features. The element in the $i$-th row and $j$-th column of the self-correlation matrix is the correlation coefficient between the $i$-th column and the $j$-th column of the difference matrix. $\lambda$ is the weighting coefficient.

III. METHOD

A. THE ROLE OF THE ACTIVATION FUNCTION OF THE LAST CONVOLUTIONAL LAYER

ReLU activation function is the most commonly used activation function. Compared to Sigmoid activation function and Tanh activation function, because of its simple derivative form, it can make network training faster. On the other hand, when the value is too large or too small, the derivatives of the other two activation functions are close to 0. Since ReLU is an unsaturated activation function, this phenomenon does not exist. However, the part of the ReLU function where the input value is negative is equivalent to the death of the neuron and will not resurrect. Therefore, some of the information is lost.

Based on the above considerations, when the non-linearity in the network convolutional layer is enough, the removal of the activation function ReLU of the last convolutional layer should enable the fully connected layer to obtain more information, which may improve the recognition performance of the entire network. For the three representative network structures VGG16, ResNet50 and MobileNetV2, we remove the last ReLU activation function and test the classification performance on four widely used datasets (which will be...
TABLE 1. Comparison of classification accuracy(%) of VGG16, ResNet50 and MobileNetV2 before and after removing the last ReLU on different datasets.

| Network structure | Dataset         | Accuracy | Accuracy (Remove ReLU) |
|-------------------|-----------------|----------|------------------------|
| VGG16             | CIFAR10         | 92.43    | 92.97 (0.54↑)          |
|                   | CIFAR100        | 71.10    | 71.67 (0.57↑)          |
|                   | Tiny ImageNet   | 57.82    | 58.35 (0.53↑)          |
|                   | FaceScrub       | 94.05    | 94.56 (0.51↑)          |
| ResNet50          | CIFAR10         | 93.06    | 93.43 (0.37↑)          |
|                   | CIFAR100        | 74.19    | 74.95 (0.76↑)          |
|                   | Tiny ImageNet   | 61.76    | 62.09 (0.33↑)          |
|                   | FaceScrub       | 89.88    | 93.42 (3.54↑)          |
| MobileNetV2       | CIFAR10         | 85.18    | 84.24 (0.94↓)          |
|                   | CIFAR100        | 51.89    | 48.31 (3.58↓)          |
|                   | Tiny ImageNet   | 45.77    | 44.66 (1.11↓)          |
|                   | FaceScrub       | 87.84    | 88.35 (0.51↑)          |

TABLE 2. Ratio of the sum of the largest first n (number of class) eigenvalues to the sum of all eigenvalues.

| Dataset          | CIFAR10 | CIFAR100 | Tiny ImageNet | FaceScrub |
|------------------|---------|----------|---------------|-----------|
| Ratio(%)         | 96.77   | 97.23    | 97.72         | 97.81     |

introduced in Section IV). The results are shown in TABLE 1.

TABLE 1 shows that for VGG16, the removal of the last ReLU improves the recognition accuracy overall. For ResNet50, for regular image classification task, the removal slightly improves the recognition accuracy; for face recognition task, the removal significantly improves the recognition accuracy. When the network structure is MobileNetV2, for regular image classification tasks, the removal reduces the recognition accuracy; while for face recognition task, the removal slightly improves the recognition accuracy. It can be seen that, for the deeper network, the removal of the activation function of the last layer ReLU optimizes the recognition performance, while for the shallower network, the removal lead to insufficient non-linearity, which may reduce the recognition accuracy in some case.

**B. THE EFFECT OF THE NUMBER OF LINEAR LAYERS**

When the structure of shortcut [7] is adopted, the more the number of convolutional layers, the higher the recognition accuracy. However, the linear layer is different. An increase in the number of linear layers will easily fall into local optimum and overfitting in learning. For example, from VGG to ResNet, the fully connected layer is changed from three layers to one layer, and the ReLU function between the three-layer fully connected layer is also removed, which indicates that the activation function between linear layers has a negative impact on the entire network, and the reduction in the number of fully connected layers can improve the recognition accuracy. This is because the multi-layer linear layer + activation function is essentially a non-linear transformation. Since there is no effective means to solve the local optimization and overfitting solution, the multi-layer will often have a counterproductive effect, while its nonlinear effect can be completely achieved through convolutional layers. So the latest network structures such as ResNet and DenseNet retain only one linear layer, which plays a role of linear dimension reduction to adapt to the number of classes.

Furthermore, we calculate the eigenvalues of the latent features before the linear layer of the trained network, sort the eigenvalues from large to small, as shown in FIGURE 4. It can be seen that a precipitous decline in eigenvalues. In this case, we calculate the ratio of the sum of eigenvalues of number of classes (for example, select the top ten eigenvalues for CIFAR10) to the sum of all eigenvalues. The results are shown in TABLE 2. On CIFAR10, CIFAR100, Tiny ImageNet and FaceScrub datasets, it is found that most of the energy (more than 95%) is in the key eigenvalues. This shows that most of the dimensions in the features before the linear layer are invalid, so it is a feasible research direction to directly obtain the features of number of classes, and there is no need for the linear layer to reduce the dimension.

**C. CHANNEL NUMBER OF THE LAST CONVOLUTIONAL LAYER AND CURSE OF DIMENSIONALITY**

The increase in the number of features in neural networks will bring about the sparseness of high-dimensional spatial data, which leads to the problem of overfitting [20]. In a certain range, increasing the number of feature dimensions will result in better classification performance, but when the number of feature dimensions exceeds a certain scale, the performance of the classifier will decline, which can be described in FIGURE 5.

Therefore, in a certain range, the reduction of the number of features will not only improve the recognition accuracy, but also reduce the amount of parameters and calculation. For FaceScrub [21] dataset (number of classes is 100), we test the influence of different number of output features on ResNet50, whose original number of output features of the
TABLE 3. Classification accuracy(%) under different number of output features on FaceScrub dataset (number of classes is 100) and ResNet50 (original number of output features is 2048).

| No. of Features | 2048 | 1024 | 512 | 300 | 100 | 50 | 10 |
|-----------------|------|------|-----|-----|-----|----|----|
| Accuracy        | 89.88| 92.74| 93.56| 94.02| 94.71| 93.69| 92.14 |

TABLE 4. Classification accuracy(%) under different number of output features on Tiny ImageNet dataset (number of classes is 200) and MobileNetV2 (original number of output features is 1280).

| No. of Features | 1280 | 640 | 320 | 200 | 100 | 50 | 10 |
|-----------------|------|-----|-----|-----|-----|----|----|
| Accuracy        | 45.77| 46.19| 46.56| 46.71| 46.06| 45.17| 40.24 |

Further, the loss function in the network training is changed from Softmax Loss to POD Loss. In order to adapt to POD Loss, the number of output channels of the merged convolutional layer is set as the feature dimension of PEDCC. The change of the loss function further improves the accuracy and robustness of CNN model.

IV. EXPERIMENTS AND RESULTS

The experiment is implemented using Pytorch 1.4.0 [23]. Three network structures of VGG16, ResNet50 and MobileNetV2, four widely used datasets of CIFAR10 [24], CIFAR100 [24], ImageNet [22] (including Tiny ImageNet and miniImageNet) and FaceScrub [21] are selected to show the advantages of FCNN. Subsequently, NASNet [13], obtained by network architecture search, and DenseNet [8], the best network in typical convolutional neural networks, are also verified. All the above experiments use Softmax Loss as the loss function. Finally, the loss function is changed from Softmax Loss to POD Loss, and experiments are carried out on several data sets on ResNet50. Each result is the average of three identical experiments.

A. EXPERIMENTAL DATASETS AND DETAILS

CIFAR10 dataset contains 10 classes of RGB color pictures. The size of the pictures is $32 \times 32$. CIFAR100 dataset contains 100 classes of pictures, the picture size is $32 \times 32$. Tiny ImageNet dataset contains 200 classes of pictures, the picture size is $64 \times 64$. FaceScrub dataset contains 100 classes of pictures, the size of the pictures is $64 \times 64$. For these four datasets, standard data augmentation [17] is performed, that is, the training images are filled with 4 pixels, randomly cropped to the original size, and horizontally flipped with a probability of 0.5, and the test images are not processed.

In the training phase, the SGD optimizer are used, the weight decay is 0.0005, and the momentum is 0.9. The initial learning rate is 0.1, a total of 100 epochs (CIFAR100 dataset is 200 epochs) are trained. For CIFAR10 dataset, Tiny ImageNet dataset and FaceScrub dataset, the learning rate is reduced to one-tenth of the original at the 30th, 60th, and 90th epoch. For CIFAR100 dataset, the learning rate is reduced to one-tenth of the original at the 50th, 100th, and 150th epoch. For the batchsize, CIFAR10 dataset, CIFAR100 dataset and FaceScrub dataset are set to 200, and Tiny ImageNet dataset is set to 256.

ImageNet dataset contains 1000 classes of pictures. The size of the pictures is not fixed, and the height and width are both greater than 224. MiniImageNet is a subset of Imagenet, and contains 100 classes of pictures. For these two datasets, the training pictures are randomly cropped to different sizes and aspect ratios, scaled to $224 \times 224$ (miniImageNet is $84 \times 84$), and flipped horizontally with a probability of 0.5. The test pictures scale the small side length to 256 (miniImageNet is 96) proportionally, and the center crops to $224 \times 224$ (miniImageNet is $84 \times 84$). In the training phase, the SGD optimizer is used, the weight decay is 0.0001 (miniImageNet is 0.0005) and the momentum is 0.9. The batchsize is set...
TABLE 5. Comparison of classification accuracy, calculation and parameter on CNN structure and FCNN structure of VGG16, ResNet50 and MobileNetV2 on different datasets.

| Network Structure | Dataset      | Accuracy(%) (CNN) | Accuracy(%) (FCNN) | FLOPs /GMac (CNN) | FLOPs /GMac (FCNN) | Params /M (CNN) | Params /M (FCNN) |
|-------------------|--------------|-------------------|--------------------|------------------|-------------------|----------------|-----------------|
| VGG16             | CIFAR10      | 92.43             | 93.25 (0.82†)      | 0.316            | 0.305             | 16.308         | 12.408          |
|                   | CIFAR100     | 71.10             | 72.31 (1.21†)      | 0.316            | 0.307             | 16.401         | 12.823          |
|                   | Tiny ImageNet| 57.82             | 58.42 (0.60†)      | 1.259            | 1.235             | 16.503         | 13.285          |
|                   | FaceScrub    | 94.05             | 94.73 (0.68†)      | 1.259            | 1.227             | 16.401         | 12.823          |

TABLE 6. Comparison of classification accuracy on CNN structure and FCNN structure of DenseNet121 on different datasets.

| Dataset       | Accuracy(%) (CNN) | Accuracy(%) (FCNN) | FLOPs /GMac (CNN) | FLOPs /GMac (FCNN) | Params /M (CNN) | Params /M (FCNN) |
|---------------|-------------------|--------------------|------------------|-------------------|----------------|-----------------|
| CIFAR10       | 93.06             | 94.44 (1.38†)      | 0.0841           | 0.0830            | 23.529         | 22.461          |
| CIFAR100      | 74.19             | 75.74 (1.55†)      | 0.0843           | 0.0807            | 23.713         | 22.507          |
| Tiny ImageNet | 61.76             | 62.65 (1.09†)      | 0.3367           | 0.3325            | 23.918         | 22.558          |
| FaceScrub     | 89.88             | 94.23 (4.35†)      | 0.3365           | 0.3323            | 23.713         | 22.507          |

FIGURE 6. CNN and FCNN structure of ResNet50.

TABLE 7. Changes in accuracy, calculation amount and parameter amount on different datasets plus network structures.

| Dataset and network structure | Performance comparison | CNN | FCNN |
|-------------------------------|-------------------------|-----|------|
| ImageNet                      | Accuracy(%)             | 76.05 | 77.56 (1.51†) |
| + FLOPs(GMac)                 | 4.122                   | 4.093 |
| ResNet50                      | Params(M)               | 25.557 | 22.969 |
| CIFAR100                      | Accuracy(%)             | 78.20 | 76.72 (1.48†) |
| + FLOPs(GMac)                 | 0.682                   | 0.638 |
| NASNet                        | Params(M)               | 5.222 | 4.273 |

B. CONVOLUTIONAL NEURAL NETWORK STRUCTURE AND FULLY CONVOLUTIONAL NEURAL NETWORK STRUCTURE

For VGG16, the three fully connected layers and the ReLU activation function between them are removed, the output channel of the last convolutional layer (convolution kernel is 3 × 3) is modified from 512 to the number of classes, and the ReLU activation function after the convolutional layer is also removed. For ResNet50, the last ReLU activation function and a fully connected layer are removed, and the output channel of the last convolutional layer (convolution kernel is 1 × 1) and the shortcut connection are modified from 2048 to the number of classes, as shown in FIGURE 6, which select top class-number of features to connect to last layer. For MobileNetV2, the last ReLU activation function and a fully connected layer are removed, and the output channel of the last convolutional layer (convolution kernel is 1 × 1) is modified from 2048 to the number of classes. For DenseNet121, the last DenseBlock, the last ReLU activation function and a fully connected layer are removed, and a convolutional layer (convolution kernel is 3 × 3) is add, whose output channel is the number of classes. For NASNet, the last ReLU activation function and a fully connected layer are removed, and the output channel 1056 (the sum of 6 output channels 176) after concatenating is modified to the number of classes 100 (the sum of 4 output channels 17 and 2 output channels 16). The network structures of VGG16, MobileNetV2, DenseNet121 and NASNet are given in APPENDIX A.

C. EXPERIMENTS ON FULLY CONVOLUTIONAL NETWORK PLUS SOFTMAX LOSS

VGG16, ResNet50 and MobileNetV2 are selected for experiments. Four small and medium-sized datasets, CIFAR10, CIFAR100, Tiny ImageNet and FaceScrub, are used. The
TABLE 8. Comparison of classification accuracy(%) on ResNet50 on different datasets.

| Dataset            | CNN+Softmax Loss | FCNN+Softmax Loss | CNN+POD Loss | FCNN+POD Loss |
|--------------------|------------------|------------------|--------------|---------------|
| CIFAR100           | 74.19±0.35       | 75.74±0.23(1.55↑) | 77.70±0.54(3.51↑) | 77.82±0.33(3.63↑) |
| FaceScrub          | 89.88±0.31       | 94.23±0.33(4.35↑) | 93.01±0.28(3.13↑) | 94.25±0.18(4.37↑) |
| ImageNet(100)      | 78.25±0.27       | 81.07±0.17(2.82↑) | 82.06±0.06(3.81↑) | 82.06±0.14(3.81↑) |
| minImageNet        | 75.44±0.39       | 75.97±0.34(0.53↑) | 77.01±0.15(1.57↑) | 77.56±0.57(2.12↑) |

TABLE 6 shows that FCNN still performs better than CNN for the changes in the parameters and calculation before and after the modification of the three network structures, the results are shown in TABLE 5. For VGG16, the amount of calculation before and after the modification has been reduced slightly, and the number of parameters has been greatly reduced. For the deeper network ResNet50, the amount of calculation has been reduced slightly, and the number of parameters are reduced by a larger proportion.

In order to compare the convergence speed of CNN and FCNN, we chose ResNet50 and Tiny ImageNet for the experiment. The variation of classification accuracy of CNN and FCNN in the training process was record, which is shown in Figure 7. Figure 7 shows that FCNN is better than CNN in convergence speed and classification accuracy in this case.

On the basis of the successful verification of these four widely used datasets, ImageNet dataset is used to experiment on ResNet50, and the changes in the amount of parameters and calculation before and after the network structure are calculated. We use the pre-trained model of ResNet50 and fine-tune it on this basis. The results are shown in TABLE 7. When the combination is ImageNet + ResNet50, the FCNN has improved the recognition accuracy, and the amount of parameters and calculation have also been reduced. NASNet is the optimal structure obtained through network architecture search, and we also modify it to a FCNN. The network is based on the network architecture search of CIFAR100. The results are shown in TABLE 7. When the combination is CIFAR100 + NASNet, the modification of FCNN has reduced the accuracy, but the amount of parameters and calculation have also decreased significantly. The reason is that NASNet [13] is obtained by network architecture search, the variable redundancy of the network structure is not as high as VGG16, ResNet50 and MobileNetV2, which are designed artificially. At the same time, NASNet is different from the previous two networks in that the convolutional layer adopts a splicing method, which includes deep-separable convolution, pooling layers,
etc., and is different from general convolution, so the effect of improving the full convolution is not obvious.

D. EXPERIMENTS ON FCNN PLUS POD LOSS

ResNet50 is used for the experiments on FCNN plus POD Loss. In order to adapt to POD Loss, some adjustments have been made to the FCNN structure of ResNet50 in FIGURE 6, as shown in FIGURE 8. The output channel of the last convolutional layer (convolution kernel is $1 \times 1$) is modified from the number of classes to the feature dimension of PEDCC, and the shortcut connection is modified to a separate convolutional layer (convolution kernel is $1 \times 1$). In order to meet the zero mean requirements of POD Loss on output features, the bias of them is both set to True.

Several datasets are selected for comparative experiments on ResNet50. We report the “mean±std” values in TABLE 8. TABLE 8 shows that the combination of FCNN plus POD Loss performed the best across all tested conditions, which indicates that the combination improves the classification accuracy and robustness of the model, making them play a better synergy.

V. CONCLUSION

Through the study on the effect of the activation function of the last convolutional layer, the effect of the number of linear layers, the number of channels of the last convolutional layer and the curse of dimensionality, this paper proposes a fully
convolutional deep learning classification network, and combines it with POD Loss. Experimental results show that the network structure not only reduces the amount of parameters and calculation, but also improves the recognition accuracy, while the addition of POD Loss further improves the accuracy and robustness of the model. For deeper networks, although the FCNN reduces limited amount of parameters and calculation, the recognition accuracy is improved. For the shallower networks, although the improvement in recognition accuracy is limited, the reduction in the amount of parameters and calculation accounts for a relatively large amount. In the follow-up, we will further study the interaction among the number of convolutional layer channels, the number of classes, and the final recognition result, find a better network structure and corresponding optimal loss function.

APPENDIX A

NETWORK STRUCTURES

CNN structure and FCNN structure of NASNet, VGG16, DenseNet121 and MobileNetV2 are shown below.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), Dec. 2015, pp. 815–823.

[3] Y. Wen, K. Zhang, Z. Li, and Q. Yu, “A discriminative feature learning approach for deep face recognition,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 499–515.

[4] F. Wang, J. Cheng, W. Liu, and H. Liu, “Additive margin softmax for face verification,” IEEE Signal Process. Lett., vol. 25, no. 7, pp. 926–930, Jul. 2018.

[5] 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. 1–7.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” Sep. 2014, arXiv:1409.1556.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. CVPR, 2016, pp. 770–778.

[8] G. Huang, Z. Liu, L. Van, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. Conf. Comput. Vis. Pattern Recognition (CVPR), 2017, pp. 4700–4708.

[9] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proc. 14th Int. Conf. Artif. Intel. Stat., Fort Lauderdale, FL, USA, vol. 15, Aug. 2011, pp. 315–323.

[10] Q. Zhu and X. Zu, “A softmax-free loss function based on predefined optimal-distribution of latent features for CNN classifier,” 2021, arXiv:2111.15449.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4510–4520.

[12] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 6848–6856.

[13] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8697–8710.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[15] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 640–651, Apr. 2017.

[16] K. Zhang, W. Zuo, and L. Zhang, “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” IEEE Trans. Image Process., vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[17] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” 2016, arXiv:1612.02295.

[18] Q. Zhu and R. Zhang, “A classification supervised auto-encoder based on predefined evenly-distributed class centroids,” 2019, arXiv:1902.00220.

[19] Q. Zhu, P. Zhang, Z. Wang, and X. Ye, “A new loss function for CNN classifier based on predefined evenly-distributed class centroids,” IEEE Access, vol. 8, pp. 10888–10895, 2020.

[20] A. K. Jain, R. P. W. Duin, and J. C. Mao, “Statistical pattern recognition: A review,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 4–37, Jan. 2000.

[21] H.-W. Ng and S. Winkler, “A data-driven approach to cleaning large face datasets,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 343–347.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Dec. 2009, pp. 248–255.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in Proc. NIPS Workshop, 2017, pp. 1–4. [Online]. Available: https://openreview.net/forum?id=BJJsrmcZ

QIUYU ZHU (Member, IEEE) received the bachelor’s degree from Fudan University, in 1985, the master’s degree from the Shanghai University of Science and Technology, in 1988, and the Ph.D. degree in information and communication engineering from Shanghai University. He is currently a Professor with Shanghai University. He has coauthored approximately 100 academic papers and the Principal Investigator for more than ten governmental funded research projects and more than 30 industrial research projects, many of which have been widely applied. His research interests include image processing, computer vision, machine learning, smart city, and computer application.

XUEWEN ZU received the bachelor’s degree from the School of Communication and Information Engineering, Shanghai University, in 2020, where he is currently pursuing the master’s degree. His research interest includes computer vision.

* * *