IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 24, 2019, accepted August 20, 2019, date of publication August 29, 2019, date of current version September 12, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2938194*

# Feature Reuse Residual Networks for Insect Pest Recognition

**FUJI REN[1], (Senior Member, IEEE), WENJIE LIU[1,2], AND GUOQING WU[2]**
[1]Faculty of Engineering, Tokushima University, Tokushima 770-8506, Japan
[2]School of Information Science and Technology, Nantong University, Nantong 226019, China

Corresponding authors: Fuji Ren (ren@is.tokushima-u.ac.jp) and Guoqing Wu (wgq@ntu.edu.cn)

This work was supported in part by the Nantong Innovation Conditions Construction Plan under Grant CP12015004.

**ABSTRACT** Insect pests are one of the main threats to the commercially important crops. An effective insect pest recognition method can avoid economic losses. In this paper, we proposed a new and simple structure based on the original residual block and named as feature reuse residual block which combines feature from the input signal of a residual block with the residual signal. In each feature reuse residual block, it enhances the capacity of representation by learning half and reuse half feature. By stacking the feature reuse residual block, we obtained the feature reuse residual network (FR-ResNet) and evaluated the performance on IP102 benchmark dataset. The experimental results showed that FR-ResNet can achieve significant performance improvement in terms of insect pest classification. Moreover, to demonstrate the adaptive of our approach, we applied it to various kinds of residual networks, including ResNet, Pre-ResNet, and WRN, and we tested the performance on a series of benchmark datasets: CIFAR-10, CIFAR-100, and SVHN. The experimental results showed that the performance can be improved obviously than original networks. Based on these experiments on CIFAR-10, CIFAR-100, SVHN, and IP102 benchmark datasets, it demonstrates the effectiveness of our approach.

**INDEX TERMS** Insect pest recognition, feature reuse, residual network.

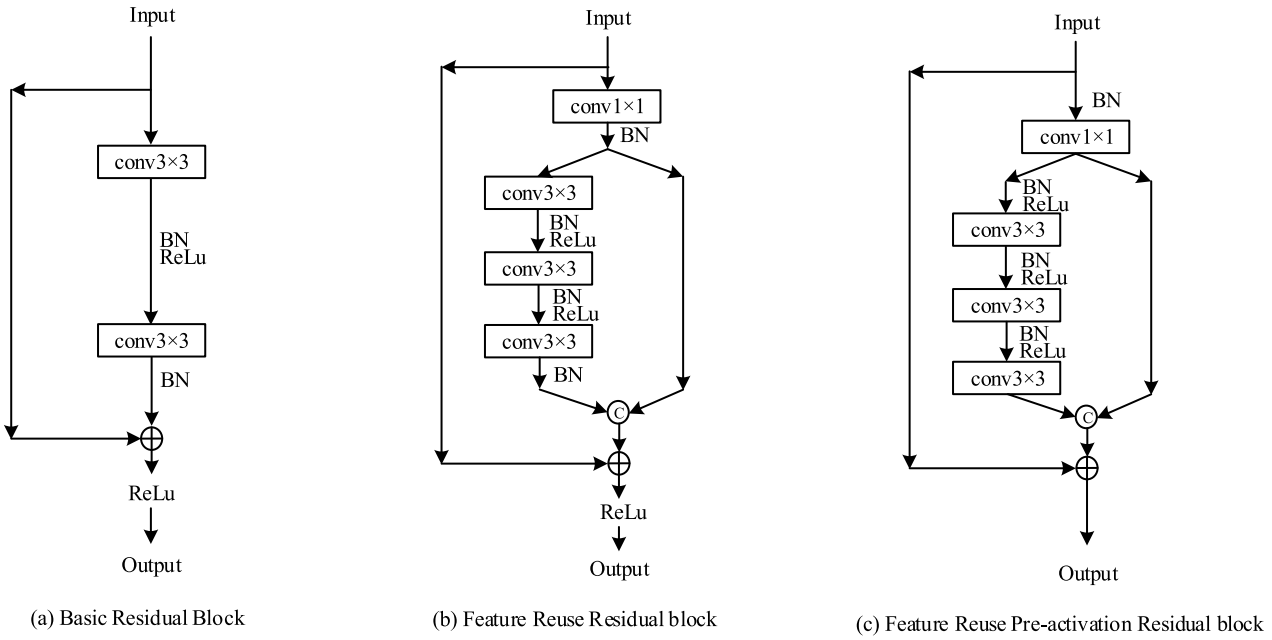
## I. INTRODUCTION

The production of crops is related to many factors, such as climate change, biodiversity, plant diseases and insect pests. Insect pests are regarded as a significant threat to the commercially important crops [1]. The conventional method of insect pests recognition relies on the professional knowledge of agricultural experts, which is a low efficient and expensive cost. In order to promote the development of intelligent agriculture, a new effectively recognition method should be proposed. In recent years, with the development of deep learning techniques, many researchers are attracted to apply this technology into different fields and propose many excellent approaches. With the successful application of deep learning in various areas, it also has been used in the agricultural domain [2]–[8].

The emergence of deep learning technology has made breakthroughs in various fields, including natural language processing [9], emotion computing [10], especially for computer vision tasks, such as image classification [11]–[13], object detection [14], [15], image segmentation [16], [17], facial expression recognition [18], [19]. Since LeNet [20] introduced the use of deep neural network architectures for computer vision tasks, the advanced architecture AlexNet [21] acquired ground-breaking victory at the ImageNet competition in 2012 by a large margin over traditional methods. Subsequently, many excellent neural networks, such as ZF-net [22], VGG [23], GoogleNet [24], Residual Networks [12], [25], and Inception Residual Networks [26], are proposed and obtain better performance on ImageNet and other benchmark datasets. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015, Residual Networks (ResNets) [12] win the 1st places on ImageNet classification, detection, localization, COCO detection as well as segmentation tasks. The shortcut connections concept inside a proposed residual unit for residual learning makes it possible to train much deeper network architectures.

In order to acquire an effective classifier to recognize the insect pest, we train the models on IP102 dataset, which is a large-scale benchmark dataset for insect pests recognition. Because the classification task of IP102 belongs to a fine-grained visual classification, we hypothesize that reuse

The associate editor coordinating the review of this article and approving it for publication was Paolo Napoletano.




(a) Basic Residual Block (b) Feature Reuse Residual block (c) Feature Reuse Pre-activation Residual block

**FIGURE 1.** Various residual blocks used in the paper.

feature from previous can enhance the accuracy performance. The method of feature reuse is widely used in computer vision tasks. In particular, DenseNets [13] allow layers connect to feature-maps from all of its preceding layers. In this way, feature-maps of all preceding layers are used as inputs and are input into all subsequent layers. It can not only effectively alleviate the difficulty of training very deep networks, but also strongly enhance the performance by reusing features from the initial layers in all subsequent layers. Therefore, we design a feature reuse residual block based on the basic residual unit, as shown in Fig. 1(b), to enhance the performance of the network. By stacking Feature Reuse Residual block, we can construct Feature Reuse Residual Network (FR-ResNet). First, compared to the original residual block used in ResNets, as shown in Fig. 1(a), we add extra connections from the input of basic residual block and concatenate the output of two branches, as seen in Fig. 1(b), then the concatenated features are feed into the next feature reuse residual block after the operation of summation. Second, in order to keep the original feature dimension and reduce complexity, the $1\times1$ convolutional layer halves the number of filters. Third, we used three serial convolutional layers instead of two convolutional layers.

In this paper, we evaluated FR-ResNet performance on IP102 dataset and compared with several state-of-art models. The experimental results show that FR-ResNet can achieve the best performance on IP102 dataset. Consequently, to demonstrate the adaptive of our method, we explored it on various residual networks, including Pre-ResNets and WRNs, and found that it is not only suitable for original ResNets, but also fit for other residual networks nicely on several benchmark datasets: CIFAR-10, CIFAR-100 [27], and SVHN [28]. The experimental results show that FR-ResNets outperform in accuracy and efficiency compared with original models, by merely reusing features from the input signal of residual blocks.

The rest of this paper is organized as follows. Section II summarizes the related works. Section III describes the feature reuse residual block and the optimization principles. In section IV, the experimental results are presented. In section V, we make some discussion. Finally, section VI concludes our paper.

## II. RELATED WORKS

In this section, we will review some related works. We first introduce the deep learning application in agriculture. Then, we present the deep convolutional neural networks, residual network variants, and feature reuse networks.

### A. DEEP LEARNING APPLICATIONS IN AGRICULTURE

The deep learning technology was applied in agriculture in recent years, and most of applications focused on identification of weed [2], plant recognition [3], [4], fruits counting [5] and crop type classification [6]. The PlantVillage, containing 54,306 images with 14 crop species and 26 kinds of diseases, is a large-scale plant disease classification dataset. And Mohanty *et al.* [7] trained AlexNet and GoogleNet via transfer learning on PlantVillage. Lin *et al.* [8] proposed a novel CNN-Fourier Dense Network and evaluated this method on their self-build dataset which is based on the optical images captured using an unmanned aerial vehicle. Lin *et al.* [29] improved the accuracy performance with a tolerable addition of parameters on the matrix-based convolutional neural network to increase neurons, data streams, and link channel of the model based. S. Chouhan *et al.* [30]

introduced a method of bacterial foraging optimization to identify and classify the plant leaf diseases automatically.

### B. DEEP CONVOLUTIONAL NEURAL NETWORKS

Since the network of AlexNet [21] was proposed in 2012, many different deeper and deeper convolutional neural networks emerged, such as VGGNet [23], NiN [31], GoogLeNet [24], ResNet [12] and DenseNet [13]. With depth going deep, the accuracy has continued to increase. However, very deep CNNs have to face the vanishing gradients problem [12]. Initialization methods and layer-wise training were adopted to address this problem in earlier works. Moreover, the activation function of ReLU and its variants were also used to prevent vanishing gradients, such as ELU [32], PReLU [33], and PELU [34]. Then batch normalization (BN) [35] could largely address this problem through standardizing the mean and variance of hidden layers for each mini-batch, and MSR initialized the weights with more reasonable variance. Meanwhile, a degradation problem has emerged, and several methods were proposed to resolve this problem. Motivated by Long Short-Term Memory recurrent networks and by using adaptive gating units to regulate the information flow, Highway Networks [36] can be trained with simple gradient descent. ResNets [12] introduced a shortcut conception to propagate information to deeper layers of networks, which achieved better performance than Highway Networks in a simpler way. Consequently, ResNets constructed deep residual network with layers exceeding 1000+ and still have competitive accuracy and superior convergence performance on many computer vision tasks. Therefore, it attracts many researchers, and more and more residual network variants have been proposed as a family of extremely deep residual architectures.

### C. RESIDUAL NETWORK VARIANTS

ResNets achieved significant success in computer vision. However, ResNets become difficult to converge when the depth goes very deep. Pre-ResNets [25] reduced the training difficulties by introducing a BN-ReLU-Conv order residual block with identity mappings as the skip connections and after-addition activation. Weighted Residual Networks [37] found the original residual networks have the incompatibility between ReLU and element-wise addition and deep network initialization problem. Therefore, they proposed the weighted residual networks, which enjoy a consistent improvement over accuracy when depths increase from 100+ layers to 1000+ layers. More residual network variants try to improve performance by constructing deeper residual networks, while the problem of diminishing feature reuse for very deep residual networks makes these networks very slow to train. To conduct these problems, WRNs [39] generated residual networks by increasing width and decreasing depth of residual networks. WRNs improved accuracy and reduced the training time compared with thin and very deep counterparts. Feng and Ren [19] proposed a stochastic depth drop-path approach which randomly drops a subset of layers and bypasses them with identity function. Their experiments showed that their method shortened training time substantially and reduced the test errors. RoR [38] added shortcut connections upon original residual networks to further dug the optimization ability of residual networks. Pyramidal Residual Network [40] enhanced the generalization ability by increasing the feature map dimension gradually instead of increasing the feature map dimension sharply at the downsampling location.

### D. FEATURE REUSE NETWORKS

Feature reuse is also regarded as an effective method to improve the performance of networks. In DenseNets [13], feature-maps from all preceding layers are used as inputs and feed into all subsequent layers directly. This method alleviated the difficulty of training very deep networks and enhanced the performance by reusing features from the initial layers in all subsequent layers. DSOD [14] trained an object detector which learned half of the feature maps from the previous scale with a series of convolutional layers and the remaining half feature maps downsampled from the contiguous high-resolution feature maps. Wang *et al.* [41] proposed a regularization method which stochastically reused feature by randomly dropping a set of feature maps for each mini-batch during the training phase to address the problem of overfitting.

In this paper, we mainly focus on models with feature reuse residual networks for insect pest recognition. We mean to explore how to construct a new effective residual network with feature reuse method to achieve better performance on IP102 dataset and evaluate the effectiveness of our approach on other residual networks and datasets.

## III. FEATURE REUSE OF RESIDUAL NETWORK

In this section, we first introduce the methodology of feature reuse residual network. Then, we present some important optimization principles.

### A. METHODOLOGY

The original residual block with identity mapping can be expressed by the following computation:

$$y_l = h(x_l) + F(x_l, w_l) \tag{1}$$

$$x_{l+1} = f(y_l) \tag{2}$$

where $x_{l+1}$ and $x_l$ are output and input of the $l$-th residual block in the network, $F$ denotes a residual function and $w_l$ are parameters of the $l$-th residual block. The function $h(x_l)$ denotes an identity mapping: $h(x_l) = x_l$, and f denotes a ReLU function. Fig. 1(a) shows the original residual basic block with branched residual signal consisting of two successive 3×3 conv layers. Sequentially stacked residual blocks construct residual networks.

We try to explore the effects of feature reuse for the original residual block by an adding extra connection from the input signal of the residual block. In order to match feature

size and dimension, the input feature maps pass through a 1×1 conv layer without ReLU function shown in Fig. 1 (b). To maximize the performance of the network, we experiment with kinds of residual block size and analysis results in the following section. Therefore, the Feature Reuse Residual block can be denoted by the following formulations:

$$y_l = h(x_l) + F(g(x_l), w_l)\, og(x_l) \tag{3}$$

$$x_{l+1} = f(y_l) \tag{4}$$

where g is a function to transform feature map size and dimensions, which is realized by a 1×1 conv layer without ReLU. Meanwhile, the location of reducing feature size also have a significant impact on test error, and our experiments empirically show that using average pooling layer before 1×1 convolutional layer in down-sampling block can achieve better performance than other options. The comparison of this matter is continued in the following section.

### B. OPTIMIZATION OF FR-RESNET

In order to optimize FR-ResNet, we must determine some important principles, such as residual block size, and location of reducing feature map size. We tested these principles on CIFAR-10 benchmark dataset.

In the case of original ResNets, the basic residual unit consists of a stack of two 3×3 convolutional layers in [12] as shown in Fig. 1(a). In order to compare the performance of different residual block sizes, we explored several types of convolutions in every residual block with a similar total number of parameters. We use $B(M)$ to denote residual block structure, which is used by WRN [39], and $M$ means a list with the kernel sizes of the convolutional layers in a residual block. For example, $B(1, 3, 3, 3)$ denotes a residual block with one 1×1 and three 3×3 convolutional layers, as shown in Fig. 1(b). We experiment with these types of convolutions on CIFAR-10 dataset, and the results are reported in Table 1. The experimental results show that $B(1, 3, 3, 3)$ had the best performance when the epoch number was 164 or 500.

**TABLE 1. Test error (%) on CIFAR-10 with different types of convolutions for FR-ResNet.**

| Block Type | Depth | Params | 164 epoch | 500 epoch |
|---|---|---|---|---|
| *B(1,3,3)* | 98 | 1.7M | 6.12 | 5.23 |
| *B(1,3,3,3)* | 135 | 1.7M | 5.17 | 4.76 |
| *B(1,3,3,3,3)* | 122 | 1.7M | 6.77 | 5.52 |

The experiments show that the performance can vary depending on the location of reducing feature map size in down-sampling block. We consider four variants in this paper: (A) adding a 2×2 average pooling layer before 1×1 conv; (B) adding a 3×3 max pooling layer with the stride of 2 before 1×1 conv; (C) 1×1 conv with the stride of 2; (D) first 3×43 conv with the stride of 2. The test results are shown in Fig. 2, which shows type A can achieve lower test error than other variants on CIFAR-10 dataset. Therefore, in our paper, we choose type A in our structures.

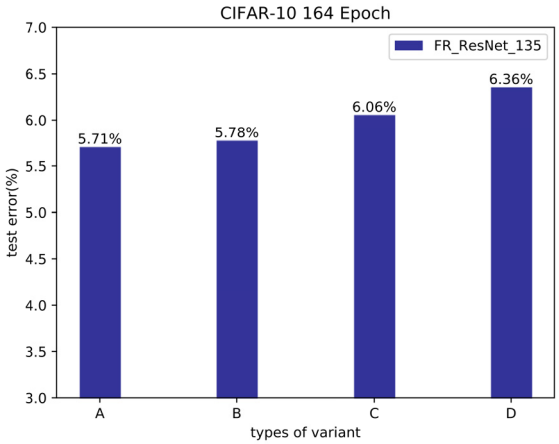



**FIGURE 2. Comparison of FR-ResNet with different location of reducing feature map size on CIFAR-10. Using type A can achieve best performance than others.**


**FIGURE 3. Example images from the IP102 dataset.**

## IV. EXPERIMENTS AND ANALYSIS

We empirically demonstrated the effectiveness of Feature Reuse Residual Network (FR-ResNet) on IP102 dataset. In order to investigate the effectiveness of our approach generalize, we combined our method with various residual networks: ResNet, Pre-ResNet, and WRN and evaluate the performance on a series of benchmark datasets: CIFAR-10, CIFAR-100, and SVHN.

### A. IP102 CLASSIFICATION RESULTS

IP102 [1] is a large-scaled insect pest dataset covered 102 species of common crop insect pests. The dataset contains 45,095 images in the training set, 7,508 images in the validation set, and 22,619 images in the testing set for classification task. Fig. 3 shows some example images from IP102. As illustrated in [1], several factors affect classification performance. First, the pests are difficult to be distinguished, because the colors are similar between object and background. Second, IP102 contains the image throughout the

**TABLE 2.** Test performance on IP102 by FR-ResNets.

| IP102 | Depth | Params | F1 | Acc (%) |
|---|---|---|---|---|
| FR-ResNet | 34 | 20.67M | 53.58 | 54.73 |
| | 50 | 30.78M | 54.18 | 55.24 |

**TABLE 3.** Test performance on IP102 by state-of-art methods.

| IP102 | Params | F1 | Acc (%) |
|---|---|---|---|
| AlexNet | 57.42M | 48.22 | 49.41 |
| ResNet-50 | 23.72M | 52.93 | 54.19 |
| ResNet-101 | 42.63M | 52.00 | 53.07 |
| Googlenet | 10.24M | 51.24 | 52.17 |
| VGG-16 | 134.68M | 51.20 | 51.84 |
| DenseNet-121 | 7.06M | 52.97 | 54.59 |

pests life cycle, and it is hard to classify, especially in the larval period. Third, the pests are often similar to each other. Due to these factors, it is more challenging for classification on IP102 dataset.

The ResNets models for ImageNet contain four residual block groups. The basic residual block required 64, 128, 256, 512 filters, and the bottleneck residual block needed 256, 512, 1024, 2048 filters. More planes will increase the number of parameters and introduce the overfitting problem for IP102. Therefore, to limit the number of parameters, we only conducted feature reuse residual network based on the basic residual block, as discussed in Section III. For IP102 dataset, we follow the implementation in [1]. SGD was adopted with a mini-batch size of 64. The learning rate is initialized by 0.01 and is divided by 10 every 40 epochs. The weight decay is 0.0005, and momentum is 0.9. The data augmentation strategies were adopted in our experiments for training: first, the image is resized into 256×256 square image; second, a rectangular region is randomly cropped with aspect ratio randomly in [3/4, 4/3] and area randomly sampled in [0.08, 1]; third, the cropped region resized into a 224×224 square image; last, mean and standard deviation normalization is also applied. During testing, we followed the processing of training except for random augmentation and cropped out the 224×4224 regions in the center of the resized image during validation. We trained our models on the training set and evaluated the performance on the test set. Our implementations are based on Pytorch 1.0 with one Nvidia Titan X.

We constructed FR-ResNet with different depths and evaluated accuracy performance on IP102 compared with ResNet baseline models. The results are reported in Table 2. Moreover, we compared FR-ResNet with several state-of-art models: AlexNet, ResNet-50, ResNet-101, Googlenet, VGG-16, and DeseNet121 to demonstrate their performance on IP102 dataset, and the results are reported in Table 3. As can be observed, compared with Table 2 and Table 3, 34-layer FR-ResNet had a test accuracy of 54.73% and 53.58 F1 score on the test set, which outperformed all model's performance in Table 3. 50-layer FR-ResNet can acquire

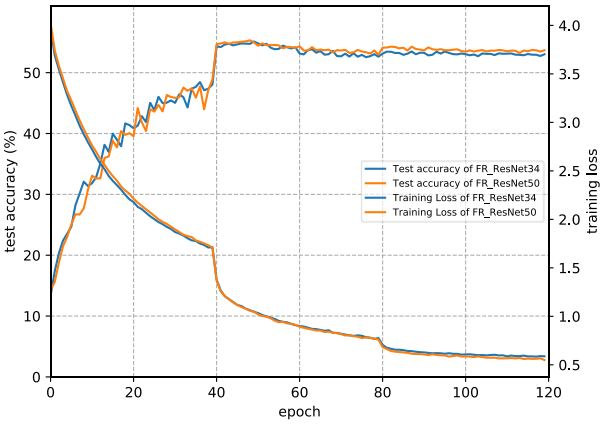



**FIGURE 4.** Test accuracy on IP102 by 34-layer and 50-layer FR-ResNets, corresponding to results in table 2.

better performance than 34-layer FR-ResNet. As Fig. 5 and Table 3 shown, ResNet-101 can achieve lower training loss than ResNet-50, while its test accuracy is worse than ResNet-50 because the increased parameters led to overfitting. Through these experiments, it demonstrated the effectiveness of our method on IP102 dataset.

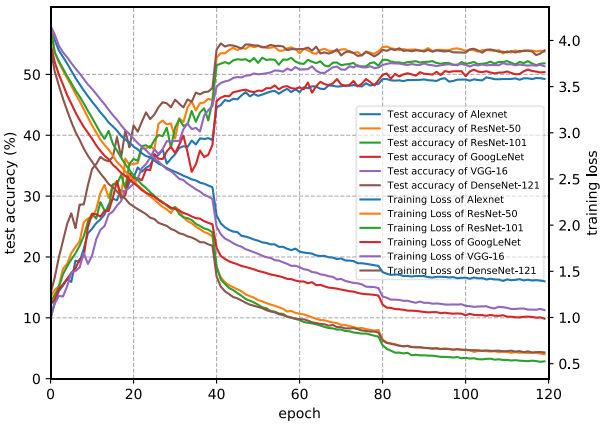



**FIGURE 5.** Test accuracy on IP102 by several state-of-art methods, corresponding to results table 3.

### B. IMPLEMENTATION ON CIFAR-10, CIFAR-100 AND SVHN DATASETS

We combined our method with various residual networks: ResNet, Pre-ResNet, and WRN and evaluate the performance on CIFAR-10, CIFAR-100, and SVHN datasets to demonstrate the effectiveness of our approach. We compared the results of FR-ResNets and the original ResNets baseline with a similar total number of parameters. In the case of CIFAR-10 and CIFAR-100, we used the 135-layer and 194-layer FR-ResNets compared with 100-layer and 164-layer ResNets, respectively. The original ResNets contained three groups which had16 filters, 32 filters and 64 filters of residual blocks, and the size of feature map are 32, 16 and 8, respectively. As shown in Fig. 1(b), we adopted

**TABLE 4. Test error (%) on CIFAR-10 by ResNets and FR-ResNets.**

| CIFAR-10 500 Epoch | Depth | Params | Error (%) | Error (%) +SD |
|---|---|---|---|---|
| ResNets | 110 | 1.7M | 5.79 | 4.84 |
| | 164 | 2.6M | 5.59 | 4.70 |
| FR-ResNets | 135 | 1.7M | 4.76 | 4.57 |
| | 194 | 2.5M | 4.96 | 4.15 |

each convolution following batch normalization and activation (ReLU). In FR-Pre-ResNet and FR-WRN experiments, we used BN-ReLU-Conv order. For CIFAR datasets, we initialize the weights with Kaiming Xavier algorithm [33] and use SGD with a batch size of 128 for 500 epochs as in [38]. The learning rate is initialized by 0.1 and is divided by 10 at the 250th and 375th. For SVHN dataset, we adopted SGD with a batch size of 128 for 50 epochs. The learning rate is initialized by 0.1 and is divided by 10 at the 30th and 35th as in [38]. The weight decay is 0.0001, and momentum is 0.9 on all datasets. According to [42], the stochastic drop-path method can alleviate overfitting and enhance the test performance, so we also adopted this method in this paper for more a comprehensive comparison. We set $p_l$ with the linear decay rule of $p_0 = 1$ and $p_l = 0.5$ when depth exceeds 100 layers, and we set $p_0 = 1$ and $p_l = 0.8$ with the linear decay when the depth is less than 100 layers.

### C. CIFAR-10 CLASSIFICATION BY FR-RESNET

CIFAR-10 is a dataset comprising a collection of 50k training images and 10k testing 32×32 pixel RGB images in 10 classes of natural scene objects. The standard data augmentation strategies were adopted in our experiments for training: 4 pixels are padded on each side, then a random 32×32 crop is sampled from the padded image; mean and standard deviation normalization is also applied or its horizontal flip.

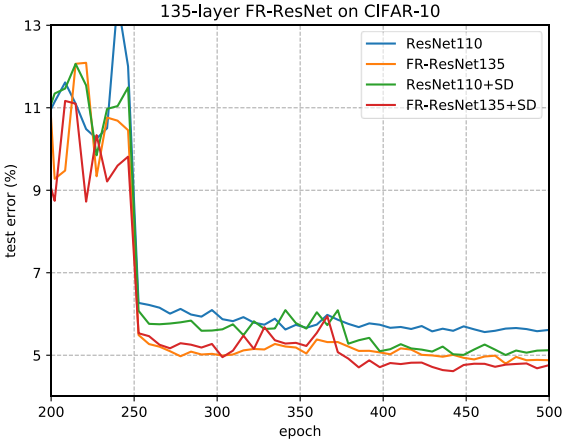



**FIGURE 6. Smoothed test errors on CIFAR-10 by 110-layer ResNets, 135-layer FR-ResNet, 110-layer ResNets+SD and 135-layer FR-ResNet+SD during training, corresponding to results in Table 4. Either FR-ResNet without SD (the orange curve) or FR-ResNet+SD (the red curve) is shown yielding a lower test error than ResNets.**

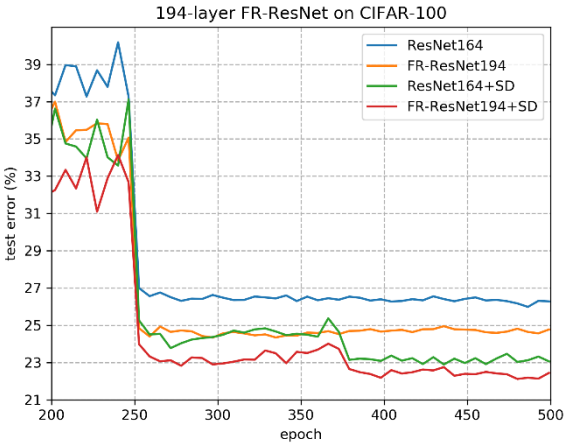



**FIGURE 7. Smoothed test error on CIFAR-100 by 164-layer ResNets, 194-layer FR-ResNet, 164-layer ResNets+SD and 194-layer FR-ResNet+SD during training, corresponding to results in Table 5. FR-ResNet+SD (the red curve) has lower test errors than other curves.**

**TABLE 5. Test error (%) on CIFAR-100 by ResNets and FR-ResNets.**

| CIFAR-100 500 Epoch | Depth | Params | Error (%) | Error (%) +SD |
|---|---|---|---|---|
| ResNets | 110 | 1.7M | 26.21 | 23.45 |
| | 164 | 2.6M | 25.96 | 22.78 |
| FR-ResNets | 135 | 1.7M | 24.88 | 22.19 |
| | 194 | 2.5M | 24.26 | 21.83 |

We construct 135-layer and 194-layer FR-ResNets to compare with 110-layer and 164-layer ResNets with a similar total number of parameters, respectively. The Table 4 and Fig. 6 show that the 110-layer ResNet and 135-layer FR-ResNet without SD achieved 5.79% and 4.76% error on the test set and the 135-layer FR-ResNets without SD outperformed the 110-layer ResNets without SD by 17.8% on CIFAR-10. Consequently, the 135-layer FR-ResNets without SD can also slightly outperform the 4.92% test error of the 1001-layer Pre-ResNets with the same batch size [25]. The 194-layer FR-ResNets without SD achieved a 4.96% error on the test set, which outperforms the 164-layer ResNets without SD by 11.3%. As can be observed, we found that the 194-layer FR-ResNets performance is worse than 135-layer FR-ResNet while the result changed when we add SD. We conjectured that the incompatibility between ReLU and element-wise addition degraded the accuracy and feature reuse method made the situation worse, and the problem was resolved in FR-Pre-ResNets which treat both $x_l$ and $f$ as identity mappings.

### D. CIFAR-100 CLASSIFICATION BY FR-RESNET

CIFAR-100 is a dataset comprising a collection of 50k training images and 10k testing 32×32 pixels RGB images, similar to CIFAR-10, but the number of classes is extended to 100. Due to each class only consists of 600 images, it is more challenging for classification on CIFAR-100 dataset. We adopt the same augmentation and preprocessing techniques as on CIFAR-10. As shown in Table 5 and Fig. 7, the 110-layer and 164-layer ResNets without SD achieved a compelling 26.21% and 25.96% error on the test set,

**TABLE 6.** Test error (%) on CIFAR-10 and CIFAR-100 by Pre-ResNets and FR-Pre-ResNets.

| 500 Epoch | Depth | Params | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|---|---|
| | | | Error (%) | Error (%) +SD | Error (%) | Error (%) +SD |
| Pre-ResNets | 110 | 1.7M | 5.22 | 4.71 | 25.93 | 23.09 |
| | 164 | 2.6M | 4.75 | 4.69 | 25.06 | 22.98 |
| FR-Pre-ResNets | 135 | 1.7M | 4.41 | 4.35 | 23.38 | 21.53 |
| | 194 | 2.5M | 4.36 | 3.90 | 22.39 | 20.73 |

**TABLE 7.** Test error (%) on CIFAR-10 and CIFAR-100 by WRNs and FR-WRNs.

| 500 Epoch | Depth | Params | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|---|---|
| | | | Error (%) | Error (%) +SD | Error (%) | Error (%) +SD |
| WRNs | 40-2 | 2.2M | 4.63 | 4.25 | 24.42 | 22.21 |
| | 40-4 | 8.9M | 4.07 | 3.98 | 21.85 | 20.28 |
| FR-WRNs | 49-2 | 2.2M | 4.50 | 4.18 | 23.21 | 21.45 |
| | 49-4 | 8.7M | 3.99 | 3.73 | 20.92 | 19.16 |

and the 135-layer FR-ResNets and 194-layer FR-ResNets without SD had 24.88% and 24.26% error on the test set. Unlike on CIFAR-10, FR-ResNets without SD outperformed their counterparts obviously on CIFAR-100 even when depth becomes deep. FR-ResNets perform better performance on a more challenging dataset. It is gratifying that the 135-layer FR-ResNets+SD and 194-layer FR-ResNets+SD achieved a 22.19% and 21.83% error on the test set which outperformed the 110-layer ResNets, 110-layer ResNets+SD, 164-layer ResNets and 164-layer ResNets+SD by 15.3%, 5.3%, 15.9% and 4.2%, respectively on CIFAR-100.

### E. FEATURE REUSE FOR PRE-RESNET AND WRN

Pre-ResNets [25] changed the order of Conv-BN-ReLU to BN-ReLU-Conv to reduce vanishing gradients, and WRN [39] can achieve a dramatic performance improvement by increasing width and decreasing depth of residual networks. We replaced the residual blocks of the original FR-ResNet in a BN-ReLU-Conv order, as shown in Fig. 1(c). We did the same experiment by FR-Pre-ResNet on CIFAR datasets, and the results are reported in Table 6, where FR-Pre-ResNet is compared with Pre-ResNet. As can be observed, the 135-layer and 194-layer FR-Pre-ResNets with SD achieved 4.35% and 3.90% test error, and they outperformed the 110-layer Pre-ResNet, 110-layer Pre-ResNet+SD, 164-layer Pre-ResNet and 164-layer Pre-ResNet+SD by 16.7%, 7.6%, 17.9%, and 16.8%, respectively on CIFAR-10. Consequently, a similar phenomenon appeared on CIFAR-100.

In the case of WRN, we found *B(1, 3, 3)* can achieve better performance. Therefore, *B(1, 3, 3)* was adopted in FR-WRN. We did the same experiment by 40-layer WRN and 49-layer FR-WRN of a width of 2 and 4 on CIFAR-10 and CIFAR-100 with a similar total number of parameters. Table 7 shows the results of our FR-WRN compared with WRN. Fig. 8 and Fig. 9 show the test error curves on CIFAR-10 and CIFAR-100 at different training epochs. The experimental results show that FR-WRNs are better than WRNs on CIFAR-10 and CIFAR-100. FR-WRN49-4+SD achieved 3.73% test error on CIFAR-10 and 19.16% test error

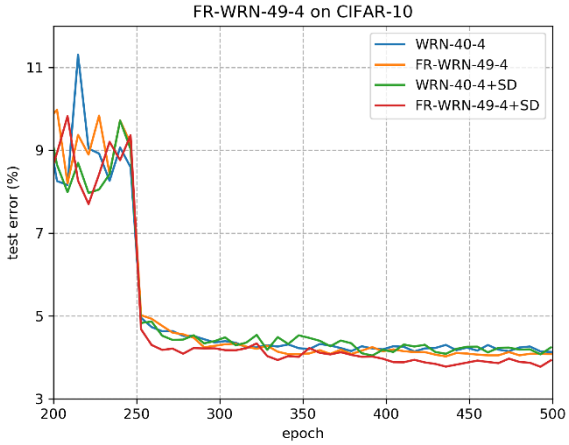



**FIGURE 8.** Smoothed test error on CIFAR-10 by WRN40-4, WRN40-4+SD, FR-WRN49-4 and FR-WRN49-4+SD during training, corresponding to results in Table 7. FR-WRN49-4+SD (the red curve) has lower test errors than the other curves.

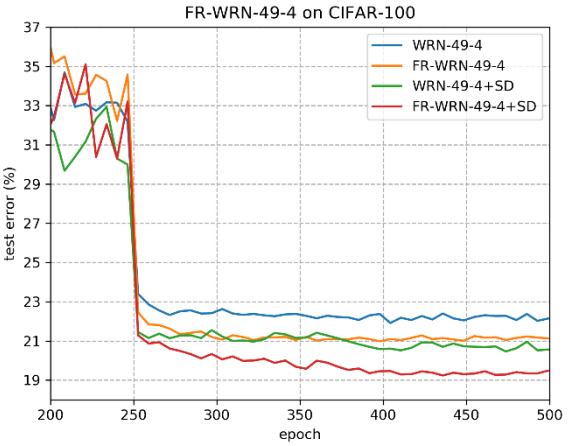



**FIGURE 9.** Smoothed test error on CIFAR-100 by WRN40-4, WRN40-4+SD, FR-WRN49-4 and FR-WRN49-4+SD during training, corresponding to results in Table 7. FR-WRN49-4+SD (the red curve) has lower test errors than the other curves.

on CIFAR-100, and it outperformed WRN-40-4+SD by 6.3% on CIFAR-10 and 5.5% on CIFAR-100. Based on these experiments and analysis, we can conclude that our Feature Reuse

Residual architecture can also improve the performance of other residual networks, including Pre-ResNets and WRNs.

### F. EFFECT OF FEATURE REUSE, DEPTH AND WIDTH

Based on preceding experiments in this section, we can conclude that increasing width or depth can improve the performance. In order to investigate the effect of width and depth to feature reuse residual network, we explored the following experiments.

The FR-ResNets derive from the original ResNets, and the vanishing gradients problems appear when depth goes deep. As shown in Table 8, the test error gradually increases from 135-layer to 194-layer, then to 314-layer on CIFAR-10. Interesting, in the case of CIFAR-100, FR-ResNets present consistent improvement. These experiments indicated that the vanishing problem still exists in deep FR-ResNet, and feature reuse method is more effective in the competitive dataset, for example, CIFAR-100.

**TABLE 8. Test error (%) on CIFAR-10 and CIFAR-100 by FR-ResNet with different depths.**

| Depth | CIFAR-10 FR-ResNet | CIFAR-100 FR-ResNet |
|---|---|---|
| 135-layer | 4.76 | 24.88 |
| 194-layer | 4.97 | 24.26 |
| 314-layer | 5.09 | 23.25 |

In the case of Pre-ResNet, it reduced the vanishing problem. We constructed FR-Pre-ResNet based on Pre-ResNet and experimented with different depth, as shown in Table 9. As can be observed, the test error gradually reduced as the depth increased. For the 1202-layer FR-Pre-ResNet with a batch size of 32, it had the 3.74% test error on CIFAR-10 and 17.85% test error on CIFAR-100. So, we conclude that the vanishing gradients can be alleviated, even on very deep FR-Pre-ResNet.

**TABLE 9. Test error (%) on CIFAR-10 and CIFAR-100 by FR-Pre-ResNet with different depths.**

| Depth | CIFAR-10 FR-ResNet | CIFAR-100 FR-ResNet |
|---|---|---|
| 135-layer | 4.35 | 21.53 |
| 194-layer | 3.90 | 20.73 |
| 242-layer | 3.85 | 20.01 |
| 1202-layer (15.7M, bs=32) | 3.74 | 17.85 |

In the case of WRN, the vanishing problem is not apparent for the shallow network, but adding more feature planes and parameters introduce overfitting. We explored experiments with FR-WRN with various widths and depths on CIFAR-10 and CIFAR-100, as reported in Table 10. The experiments show that the network performance can be improved as both depth and width increasing. But when we widened the FR-WRN, the problem of overfitting appeared. So, we need to reduce it by SD. As Table 10 shown, FR-WRN-94-4+SD achieved a 3.34% test error on CIFAR-10 and a 17.99% test error on CIFAR-100. However, we found that the 1202-layer FR-Pre-ResNet+SD compared with FR-WRN-94-4+SD can make lower test error on CIFAR-100 with fewer parameters. It demonstrates that increasing depth is more effective for feature reuse residual networks on CIFAR-100.

**TABLE 10. Test error (%) on CIFAR-10 and CIFAR-100 by FR-WRN with different widths and depths.**

| Depth and Width | CIFAR-10 FR-ResNet | CIFAR-100 FR-ResNet |
|---|---|---|
| FR-WRN49-2 | 4.18 | 21.45 |
| FR-WRN49-4 | 3.73 | 19.16 |
| FR-WRN76-2 | 3.88 | 20.21 |
| FR-WRN76-4 | 3.39 | 18.64 |
| FR-WRN94-4 | 3.34 | 17.99 |

Based on these experiments and analysis, we conclude that both adding depth and width of feature reuse residual networks are effective for model learning capability. We have to carefully choose the tradeoff between the depth and width to achieve satisfying results.

### G. SVHN CLASSIFICATION RESULTS

The Street View House Number (SVHN) dataset is also a well-known benchmark dataset in computer vision. The dataset contains 73,257 digits in the training set, 26,032 in the test set, and 531,131 additional training images respectively.

We used all the training samples but without performing data augmentation. Mean and standard deviation normalization is also applied to preprocess the data. To demonstrate the effectiveness of our method on SVHN, we used WRN40-4 and FR-WRN49-4 with a similar total number of parameters to train SVHN, and the results are reported in Table 11. As can be observed, FR-WRN49-4+SD outperformed WRN40-4+SD by 7.4% on SVHN, and Fig. 10

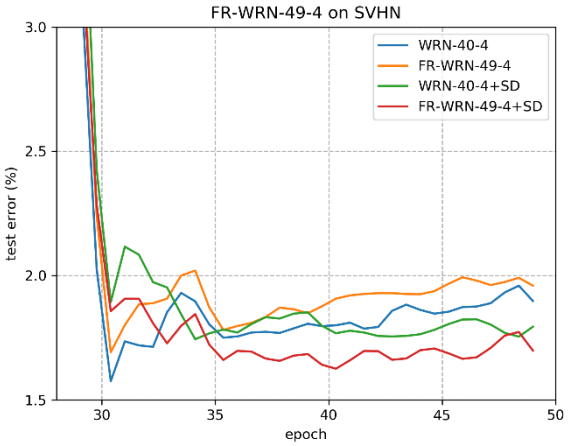



**FIGURE 10. Smoothed test error on CIFAR-10 by WRN40-4, WRN40-4+SD, FR-WRN49-4 and FR-WRN49-4+SD during training, corresponding to results in Table 11. FR-WRN49-4+SD (the red curve) has lower test errors than the other curves.**

showed the testing curves. These experiments showed our approach could achieve improvement on SVHN dataset too.

## V. DISSCUSSION

We found the impact of Feature Reuse Residual unit is twofold. First, the feature from previous layers is used for subsequent residual blocks. Second, feature reuse residual unit has a stronger capacity of representation than the original residual unit.

### A. FEATURE REUSE

From Eqn. 3 the branched residual signal and the input signal of a residual block are concatenated before summation, and we realized the Feature Reuse Residual block based on the original residual bl ock. BN and ReLU are omitted for simplifying. We can split the identity mapping into two parts. Therefore, as illustrated in Fig. 11, we could conjecture that each output of Feature Reuse Residual block contains two parts: one is from a residual block, and the other is from the input directly. Through this method of learning half and reusing half, we enhanced the capacity of the network and intensified the relationship of the adjacent residual block. The results of Table 2, Table 3, Table 4, and Table 5 show our models outperformed the baseline models significantly with a similar total number of parameters or fewer parameters and demonstrate our suppose that reuse feature from previous layers in the residual block can improve the performance on IP102 dataset.

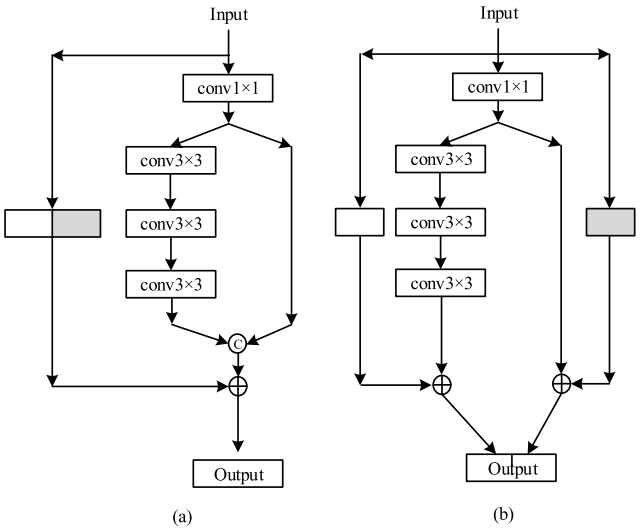



**FIGURE 11.** Structure of feature reuse residual unit (a), (b) unraveled view of (a) showing that the output contains two parts: One is from a residual block and, the other is from the input directly.

### B. STRONGER CAPACITY OF REPRESENTATION

In the case of original ResNet and Pre-ResNet, each residual block contains two continuous 3×3 convolutional layers. While Feature Reuse Residual block contains three continuous 3×3 convolutional layers, as shown in Fig. 1(b), and the results in Table 4, Table 5 and Table 6 show this structure can achieve a better performance with a similar total number of parameters. For each feature reuse residual block, the increased 3×3 convolutional layer with ReLU function enhance nonlinearity. However, the total number of ReLU unit in Feature Reuse Residual networks is less than the counterpart residual networks with a similar total number of parameters. In the case of WRN, the capacity of each wide residual unit is stronger than the pre-activation residual block [39]. After combined WRN with feature reuse method, the performance further enhanced with less ReLU unit, as shown in Table 7, Table 10, and Table 11. So, we can conclude that Feature Reuse Residual network has a stronger capacity of representation than their counterparts with less ReLU unit.

**TABLE 11.** Test error (%) on SVHN by WRNs and FR-WRNs with different depths and widths.

| SVHN | Depth | Params | Error (%) | Error (%) +SD |
|---|---|---|---|---|
| WRN | 40-4 | 8.9M | 1.69 | 1.75 |
| FR-WRN | 49-4 | 8.7M | 1.78 | 1.62 |

## VI. CONCLUSION

In our work, we proposed the feature reuse residual network (FR-ResNet) for insect pest recognition. The central idea of the structure was described in this paper involves learning half and reuse half feature in each Feature Reuse Residual block. Based on the simple structure, we constructed the FR-ResNet and evaluated the classification performance on IP102 dataset, which is a challenging insect pest recognition benchmark dataset. The experimental results on IP102 showed that FR-ResNet could achieve better accuracy recognition performance compared with the baseline models. We also demonstrated that our approach could be used by other residual networks and outperform the original networks on CIFAR-10, CIFAR-100, and SVHN datasets. Through these empirical studies, the effectiveness of our approach was demonstrated, and this approach can be easily implemented in other residual networks.

In future work, we will try to extend the proposed network to different technical fields and explore more effective feature reuse network structure.

## REFERENCES

[1] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8787–8796.
[2] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," *Biosyst. Eng.*, vol. 151, pp. 72–80, Nov. 2016.
[3] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, "Fine-tuning deep convolutional networks for plant recognition," in *Proc. CLEF (Work. Notes)*, Sep. 2015, p. 1391.
[4] H. Zhang, G. He, J. Peng, Z. Kuang, and J. Fan, "Deep learning of path-based tree classifiers for large-scale plant species identification," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 25–30.

[5] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 781–788, Apr. 2017.

[6] S. Ji, C. Zhang, A. Xu, Y. Shi, and Y. Duan, "3D convolutional neural networks for crop classification with multi-temporal remote sensing images," *Remote Sens.*, vol. 10, no. 1, p. 75, Jan. 2018.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016.

[8] C.-W. Lin, Q. Ding, W.-H. Tu, J.-H. Huang, and J.-F. Liu, "Fourier dense network to conduct plant classification using UAV-based optical images," *IEEE Access*, vol. 7, pp. 17736–17749, 2019.

[9] F. Ren and J. Deng, "Background knowledge based multi-stream neural network for text classification," *Appl. Sci.*, vol. 8, no. 12, p. 2472, Dec. 2018.

[10] F. Ren, Y. Dong, and W. Wang, "Emotion recognition based on physiological signals using brain asymmetry index and echo state network," *Neural Comput. Appl.*, vol. 2018, pp. 1–11, 2018.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.

[14] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "DSOD: Learning deeply supervised object detectors from scratch," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1919–1927.

[15] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2874–2883.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[17] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: https://arxiv.org/abs/1511.07122

[18] F. Ren and Z. Huang, "Automatic facial expression learning method based on humanoid robot XIN-REN," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 6, pp. 810–821, Dec. 2016.

[19] D. Feng and F. Ren, "Dynamic facial expression recognition based on two-stream-CNN with LBP-TOP," in *Proc. 5th IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2018, pp. 355–359.

[20] 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.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[22] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning,"in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 4278–4284.

[27] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, ON, Canada, CiteseerX, Princeton, NJ, USA, Tech. Rep. TR-2009, 2009.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, Granada, Spain, vol. 2011, 2011, p. 5.

[29] Z. Lin, S. Mu, F. Huang, K. A. Mateen, M. Wang, W. Gao, and J. Jia, "A unified matrix-based convolutional neural network for fine-grained image classification of wheat leaf diseases," *IEEE Access*, vol. 7, pp. 11570–11590, 2019.

[30] S. S. Chouhan, A. Kaul, U. P. Singh, and S. Jain, "Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology," *IEEE Access*, vol. 6, pp. 8852–8863, 2018.

[31] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014, *arXiv:1312.4400*. [Online]. Available: https://arxiv.org/abs/1312.4400

[32] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*. [Online]. Available: https://arxiv.org/abs/1511.07289

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[34] L. Trottier, P. Giguère, and B. Chaib-Draa, "Parametric exponential linear unit for deep convolutional neural networks," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 207–214.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[36] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, *arXiv:1505.00387*. [Online]. Available: https://arxiv.org/abs/1505.00387

[37] F. Shen, R. Gan, and G. Zeng, "Weighted residuals for very deep networks," in *Proc. 3rd Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2016, pp. 936–941.

[38] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: https://arxiv.org/abs/1605.07146

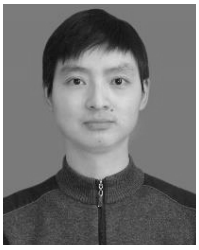
[39] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: Multilevel residual networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1303–1314, Jun. 2018.

[40] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6307–6315.

[41] M. Wang, J. Zhou, W. Mao, and M. Gong, "Multi-scale convolution aggregation and stochastic feature reuse for densenets," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 321–330.

[42] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 646–661.

**FUJI REN** received the Ph.D. degree from the Faculty of Engineering, Hokkaido University, Japan, in 1991. From 1991 to 1994, he was a Chief Researcher with CSK. In 1994, he joined the Faculty of Information Sciences, Hiroshima City University, as an Associate Professor. Since 2001, he has been a Professor with the Faculty of Engineering, Tokushima University. His current research interests include natural language processing, artificial intelligence, affective computing, and emotional robot. He is the Academician of The Engineering Academy of Japan and EU Academy of Sciences. He is an Editor-in-Chief of *International Journal of Advanced Intelligence* and the Vice President of CAAI. He is a Fellow of The Japan Federation of Engineering Societies, IECE, and CAAI. He is the President of the International Advanced Information Institute, Japan.



**WENJIE LIU** was born in Hunan, China, in 1989. He received the B.S. degree in information engineering from Nanhang Jincheng College, China, in 2011, and the M.S. degree in information and communication engineering from Nantong University, China, in 2014. He is currently pursuing the double Ph.D. degree with Nantong University and Tokushima University. His research interests include image analysis, computer vision, and artificial intelligence.

**GUOQING WU** received the B.S. and M.S. degrees in mechatronics from Jiangsu University, China, in 1983 and 1993, respectively, and the Ph.D. degree in mechanical design and theory from Shanghai University, China, in 2006. He is currently a Professor of sciences with Nantong University, China. His research interests include mechanical engineering, laser technology application, and artificial intelligence.