

# Deep Multibranch Fusion Residual Network for Insect Pest Recognition

Wenjie Liu, Guoqing Wu, and Fuji Ren, *Senior Member, IEEE*

**Abstract**—Earlier insect pest recognition is one of the critical factors for agricultural yield. Thus, an effective method to recognize the category of insect pests has become significant issues in the agricultural field. In this paper, we proposed a new residual block to learn multi-scale representation. In each block, it contains three branches: one is parameter-free, and the others contain several successive convolution layers. Moreover, we proposed a module and embedded it into the new residual block to recalibrate the channel-wise feature response and to model the relationship of the three branches. By stacking this kind of block, we constructed the Deep Multi-branch Fusion Residual Network (DMF-ResNet). For evaluating the model performance, we first test our model on CIFAR-10 and CIFAR-100 benchmark datasets. The experimental results show that DMF-ResNet outperforms the baseline models significantly. Then, we construct DMF-ResNet with different depths for high-resolution image classification tasks and apply it to recognize insect pests. We evaluate the model performance on the IP102 dataset, and the experimental results show that DMF-ResNet could achieve the best accuracy performance than the baseline models and other state-of-art methods. Based on these empirical experiments, we demonstrate the effectiveness of our approach.

**Index Terms**—Multi-branch Fusion, Insect pest recognition, image classification.

## I. INTRODUCTION

IN the agricultural field, insect pest control has always been one of the critical issues for agricultural productivity for decades. Therefore, seeking an effective method to recognize the category of insect pests will be helpful for early detection and disposal, which can decrease the damage. In the world, there are millions of pest species and the subtle differences among species making pest recognition becoming one of the significant challenges in agriculture pest manual management. In the case of traditional insect pest recognition, it relies on agriculture experts heavily, which is highly expensive and time-consuming. With the development of machine learning and deep learning techniques, many works focus on transferring these technologies to the insect pest recognition task [1]-[5].

Deep learning, especially for computer vision, has been successfully applied in various domains (e.g., image classification, object detection, and segmentation). Researchers proposed many excellent models, which achieved state-of-art

performance in benchmark datasets. These models are also applied to address our daily life problems and bring convenience to us [6]-[9]. In 2015, residual networks (ResNets) [10], as a popular convolutional neural network now, achieved the 1<sup>st</sup> place in several visual tasks, including image classification, object detection, and segmentation. It can be implemented exceeding 1000+ layers with a nice convergence behavior. In order to address the degradation problem, K. He et al. [11] proposed the Pre-activation ResNet. Meanwhile, in GoogleNet [12], it fuses multi-scale features from different branches to construct effective models. Thus, learning multi-scale representation is also considered as one of the effective methods to improve model performance.

The development of deep learning relies on large-scale image datasets massively. It is well known that the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [13] had significantly improved the development of computer vision. Therefore, a large-scale image dataset on a specific domain is essential. In IP102 [1], it released a large-scale dataset for insect pest recognition. To recognize insect pests more effectively, we hypothesis that fusing the feature from different scales can enhance the model performance. Based on this hypothesis, we proposed a new convolutional neural network to perform insect pest recognition task based on ResNet, Pre-ResNet, and multi-scale representation fusion method. Thus, we use ResNet and Pre-ResNet as our baseline models.

In ResNet [10], it proposed two types of residual architecture: basic and deeper bottleneck architectures. Meanwhile, Pre-ResNet [11] modified the building of residual architectures by changing the order of Conv-Bn-Relu to Bn-Relu-Conv and regarding the identity mapping as the skip connection and after-addition activation, as shown in Fig. 1(a) and (b). Therefore, to learn multi-scale representation, we combined the two residual architectures with a parameter-free branch together, as shown in Fig. 1(c). To fusion the multi-scale representation more effectively, we added two  $1 \times 1$  convolution layer. One is used to optimize and balance the input feature from three branches; the other is used to reduce the feature map dimension after feature concatenation operation. We named this block as multi-branch fusion residual block. Moreover, we proposed a module to recalibrate the channel-wise feature response and to model the relationship between the three

W. Liu is with the School of Information Science and Technology, Nantong University, Nantong 226019, China, and also with the School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China. Meanwhile, he is with the Faculty of Engineering, Tokushima University, Tokushima 770-8506, Japan (lwj2014@ntu.edu.cn).

G. Wu is with the School of Information Science and Technology, Nantong University, Nantong 226019, China (e-mail: wgg@ntu.edu.cn).

F. Ren is with the Faculty of Engineering, Tokushima University, Tokushima 770-8506, Japan (e-mail: ren@is.tokushima-u.ac.jp).

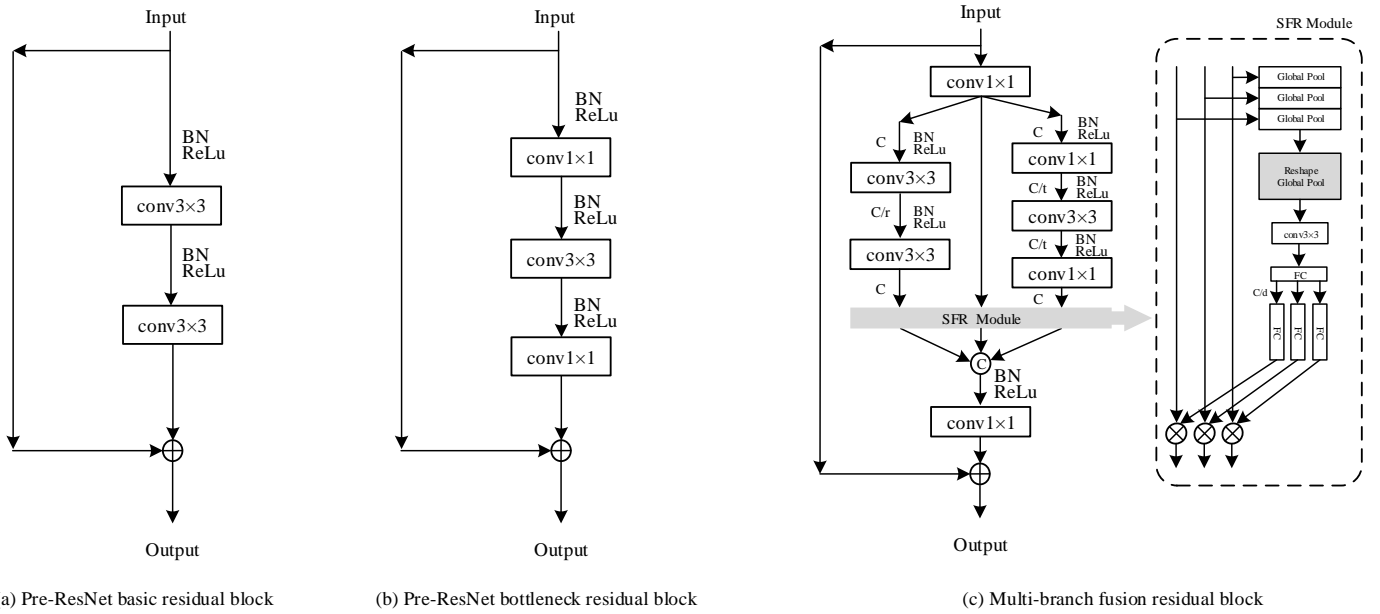


Fig. 1. Different residual block used in this paper.

branches. This module realized this purpose by three operations – Squeeze, Fuse, and Recalibrate. Thus, we named it as SFR module, as shown in Fig. 1(c). Then, we embedded the SFR module into the multi-branch fusion residual block to enhance its capacity. The SFR module actualized its function by the following steps. First, we used a global pooling layer on each branch to squeeze global spatial information and generate channel-wise statistics. Second, we concatenated the squeezed information and reshaped the matrix. Then a  $3 \times 3$  convolutional kernel scanned the matrix to improve the non-linear representation capability. Third, the result is flattened and pass through a fully connected (FC) layer with dimensionality-reduction and relu function. Then the output is connected to three FC layers, which of channel dimensions are increased to match each branch. Last, the final output of the three FC layers is applied to each branch.

By stacking the multi-branch fusion residual block, we obtained the Deep Multi-branch Fusion Residual Network (DMF-ResNet). In order to evaluate the model performance, we first tested the test error performance on CIFAR-10 and CIFAR-100 datasets. The experimental results supported that our approach could improve the capacity of the model. Meanwhile, to explore the characteristics of DMF-ResNet, we also evaluated the impact of depth and width for our models. The empirical experiments demonstrated the model performance could be improved when increasing the model depth or width. Even for extremely deep DMF-ResNet, it can achieve compelling test error performance. Besides, we constructed some ablation experiments to verify the effectiveness of multi-scale representation learning and SFR module, respectively. The results indicated that the multi-scale representation learning could bring benefits to our model, and the SFR module could recalibrate channel-wise feature response and model the relationship of three branches. Then, we applied our model to recognize insect pests. We constructed DMF-ResNet with

different depths for high-resolution image classification tasks and tested the model performance on IP102 dataset. The experimental results showed that our model not only had higher test accuracy than ResNet and Pre-ResNet with fewer parameters but also outperformed other state-of-art methods. Based on these empirical experiments on CIFAR and IP102 datasets, we demonstrated the effectiveness of our approach. Thus, it also verified that learning multi-scale representation and modeling the relationship of different scales can enrich the extracted feature to improve the capacity of image classification.

The main contributions are summarized as follows:

- 1) To enrich the extracted feature for the classification task, we proposed a multi-branch fusion residual architecture to learn multi-scale representation. The experimental results indicate that our approach can enhance model performance effectively.
- 2) To further enhance the capacity of our model, we proposed the SFR module. It can be embedded into multi-branch fusion residual architecture to recalibrate channel-wise feature response and to model the relationship of three branches.

The rest of the paper is arranged as follows. Section II is a literature review that introduces the related works for the development of insect pest recognition, deep convolutional neural networks, and attention mechanism in CNNs. Section III introduces our methodology and some critical principles of our approach. Section IV introduces the experimental results and analysis on CIFAR and IP102 datasets. Section V presents some discussions. Section VI concludes this paper.

## II. RELATED WORK

We will review some related works in this section. First, we present the development of insect pest recognition. Then, the deep convolutional neural networks and some attention

mechanism works in CNNs are presented.

#### A. Development of insect pest recognition

For the traditional insect pest recognition task, the primary solutions are based on the handcrafted feature, such as SIFT [14], HOG [15]. These methods obtain a good result on the low-level feature representations, including color, edge, and texture. However, because of lacking high-level semantic information representation ability, the handcrafted methods can not reach satisfactory results. In recent years, deep learning methods attracted more attention in the research community. Many great convolution neural networks, including VGG [16], ResNet [10], and GoogleNet [12], are proposed and achieve state-of-art results on large-scale benchmark datasets, which exceed the handcrafted methods significantly. The successful application in other domains also facilitated the development of insect recognition. To solve the problem of pests' different scales and attitudes, R. Li et al. [2] proposed an valid data augmentation strategy for CNN-based models. To detect and classify eight insects, K. Dimililer and S. Zarrouk [3] proposed a two-stages method depended on neural networks. Liu et al. [4] released their dataset consisting of about 5,000 training images in 12 categories of paddy field pests and trained a deep CNN model on this dataset. Through reusing the feature in each residual block, F. Ren et al. [5] proposed the FR-ResNet to classify insect pests. Meanwhile, a large-scale dataset will promote the development of insect pest recognition. X. Wu et al. [1] collected a large-scale dataset called IP102 for insect pest recognition, which consists of more than 75,000 images in 102 classes.

#### B. Deep convolutional neural networks

Many excellent convolutional neural networks are emerged in recent years, including VGGNet [16], NiN [17], ResNet [10], GoogLeNet [12], DesnseNet [18], and NASNet [19] etc. Some of these works focus on constructing more deeper models to enhance model performance. By introducing a shortcut conception, ResNet [10] can propagate information in the entire network smoother. Thus, ResNet can be implemented with exceeding 1000+ layers and still have nice convergence behaviors and compelling accuracy on many classification tasks. Besides, ResNet proposed a deeper bottleneck architecture, which can construct a thinner and deeper model than basic architecture for high-resolution image tasks. However, the degradation problem appeared in extremely deep ResNet. To address this problem, Pre-ResNet [11] regarded the identity mapping as the skip connection and after-addition activation, and the author converted the order of Conv-Bn-Relu to Bn-Relu-Conv. The weighted residual network [20] solved the issue of incompatibility between ReLU and element-wise addition to eliminating the degradation problem. More and more deep residual network variants emerged. Instead of sharply increasing the feature map dimension at down-sampling block, D. Han et al. [21] proposed the Pyramid Residual Network by increasing the feature map dimension gradually in the entire network. G. Huang et al. [22] proposed the stochastic depth training method for ResNet variant networks, which

reduces training time substantially and improves the test error significantly. To further dig the optimization ability of Residual networks, K. Zhang et al [23] proposed the RoR to optimize original residual mapping. All these models form a deep residual-networks family. Meanwhile, some other works focus on enriching the feature representation to improve model performance. GoogLeNet [12] enhanced the model performance by concatenating the feature map from several branches with three different sizes of filters (1×1, 3×3, 5×5). This method had been demonstrated that it could achieve high accuracy performance on several benchmark datasets. Then, several enhanced edition models based on GoogLeNet are proposed, including Inception-v2 [24], Inception-v3 [25], and Inception-v4 [26]. Besides, Z. Wang [27] proposed a multi-scale module to the acquisition of complex breast features in a single image. Inspired by linear dynamical systems, D. Anastasios et al. [28] proposed LDS-ResNet, which outperformed several extensions of the original network on several benchmark datasets. Therefore, learning the multi-scale representation is also deemed as a valid method to enhance model performance.

#### C. Attention mechanism in CNNs

Attention mechanism has been approved their effectiveness in many tasks, including sequence learning [29], [30], localization [31], [32], and image captioning [33] etc. Meanwhile, soft attention can be trained end-to-end for convolutional neural networks. Therefore, some works combined the soft attention method with the existing models in an innovative way to construct new models. Wang et al. [34] proposed the attention residual learning, which helped to train very deep residual attention networks. They used the mixed attention to capture different types of attention guiding feature learning and encoded top-down attention mechanism into trunk-and-mask architecture based on hourglass modules [35]. To strengthen the model representational power, J. Hu et al. [36] proposed a light-weight squeeze-and-excitation block to adaptively recalibrate channel-wise feature responses. It can significantly promote the model performance for existing state-of-art CNN models. M. Luo et al. [37] proposed a stochastic region pooling module to improve the capacity of channel-wise attention network. This module made the channel descriptors more diversity and representative through generating more or wider important feature response. To realize the adaptive receptive field sizes of neurons, X. Li et al. [38] proposed a selective kernel convolution to aggregate information from multiple kernels.

Activated by these works, we chiefly focus on seeking a valid approach to promote the model capability of insect pest recognition. In other words, we try to construct a more effective model based on learning multi-scale representation, recalibrating channel-wise response, and modeling the relationship of these branches to acquire better model performance on IP102 and other benchmark datasets.

### III. DEEP MULTI-BRANCH FUSION RESIDUAL NETWORK

First, we will present the methodology of deep multi-branch

fusion residual network in this section. Then, to maxing the model performance, some critical optimization principles should be presented, including parameter proportion of different branches on model performance and the impact of model width.

### A. Methodology

Learning multi-scale representation is deemed to be a valid method to improve model performance. Therefore, in order to further improve the capacity of the model, we combine two types of residual architectures with a parameter-free branch to learn multi-scale representation, as shown in Fig. 1(c), which is named as multi-branch fusion residual block. We use  $x_l \in R^{H \times W \times C}$  denotes the  $l$ -th layer input feature map. Then, the following computation can perform it:

$$\begin{aligned} u &= g_1(x_l, \hat{w}_{1 \times 1}) \\ B_1 &= u \\ B_2 &= F_{basic}(u, w_{basic}) \\ B_3 &= F_{bottleneck}(u, w_{bottleneck}) \end{aligned} \quad (1)$$

Here the function  $g_1$  denotes to the  $1 \times 1$  convolution layer with the parameter of  $\hat{w}_{1 \times 1}$ .  $B_1$ ,  $B_2$ , and  $B_3$  refer to the extracted feature from three branches, and  $F_{basic}$  and  $F_{bottleneck}$  refer to the residual functions of basic and bottleneck residual branches, respectively.  $w_{basic}$  and  $w_{bottleneck}$  refer to the parameter of two residual branches. However, if we concatenated three branches directly, model performance can not achieve optimal results. The parameter proportion of different branches and the width of the model have significant impact on model performance. The comparison of these matters will be continued in the following section.

In order to further improve the capacity of multi-branch fusion residual block, we proposed a module to recalibrate channel-wise feature response adaptively and to model the relationship of the three branches, as shown in Fig. 1(c). This module realized this purpose by three operations - Squeeze, Fuse and Recalibrate, thus we named it as SFR module.

**Squeeze.** Following the setting in SENet [36], we also adopted the global average pooling to generate global information  $b_k \in R^c, k = 1, 2, 3$  from each branch  $B_k, k = 1, 2, 3$  in its spatial dimensions  $H \times W$ , and the  $c$ -th element of  $b_k$  can be calculated by:

$$b_k^c = F_{gp}(B_k^c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W B_k^c(i, j), k = 1, 2, 3 \quad (3)$$

Where  $F_{gp}$  denotes the global average function,  $B_k = [B_k^1, B_k^2, \dots, B_k^c], k = 1, 2, 3$ .

**Fuse.** As shown in Fig. 1(c), the squeezed signals are concatenated as  $\hat{s} = [b_1^T, b_2^T, b_3^T] \in R^{C \times 3}$ . Then,  $\hat{s}$  is reshaped to generate the folded feature map  $\tilde{s} \in R^{\frac{C}{m} \times 3m}$ , where the fold-ratio of  $m$  is used to control the shape of feature map. Subsequently, a  $3 \times 3$  convolution kernel scans the folded feature map to enhance the nonlinear representation capacity,

$$\check{s} = F_w(\tilde{s}, w_{3 \times 3}) \quad (4)$$

Where  $\check{s} \in R^{C \times \frac{C}{m} \times 3m}$ ,  $F_w$  denotes to  $3 \times 3$  convolution function, and  $w_{3 \times 3}$  denotes to the parameter of  $3 \times 3$  convolution layer. Then, we obtain the mean result in channel dimension,

and the flatten layer is used to reshape the convolution results for subsequent FC layers, as  $s = (F_{flatten}(\check{s}))^T \in R^{3C}$ . Further, a compact feature  $z \in R^{\frac{C}{d}}$  is implemented to reduce the model complexity:

$$z = \tilde{F}_{fc}(s, W_1) = \delta(W_1 s) \quad (5)$$

Where  $W_1 \in R^{\frac{C}{d} \times 3C}$ ,  $d$  is the reduction ratio to control the bottleneck structure, and  $\delta$  refers to the relu function.

**Recalibrate.** As stated in previous section, our goal is to rescale the value for each channel and to model the relationship of three branches. Therefore, we implement three soft attention vectors  $M_1, M_2, M_3 \in R^{C \times d}$  for  $B_1, B_2, B_3$ , respectively. Note that  $M_k^c$  is the  $c$ -th row of  $M_k$ .

$$\begin{aligned} M_k &= \hat{F}_{fc}(z, W_2) = \sigma(W_2 z) \\ \tilde{B}_k^c &= M_k^c \cdot B_k^c \end{aligned} \quad (6)$$

Here  $W_2 \in R^{C \times \frac{C}{d}}$ ,  $\tilde{B}_k = [\tilde{B}_k^1, \tilde{B}_k^2, \dots, \tilde{B}_k^c], k = 1, 2, 3$ . Then,  $\tilde{B}_1, \tilde{B}_2, \tilde{B}_3$  are concatenated together, as  $\tilde{B} = [\tilde{B}_1, \tilde{B}_2, \tilde{B}_3]$ . The concatenated feature pass through a  $1 \times 1$  convolution layer to reduce the feature map dimension.

$$x_{l+1} = h(x_l) + g_2(\tilde{B}, \tilde{w}_{1 \times 1}) \quad (7)$$

Where  $x_{l+1}$  refers to the output of the  $l$ -th residual block in the network, and  $g_2$  refers to the function of the  $1 \times 1$  convolution layer with the parameter of  $\tilde{w}_{1 \times 1}$ . The function  $h(x_l)$  is an identity mapping:  $h(x_l) = x_l$ .

### B. Optimization of deep multi-branch fusion residual network

For the sake of maxing DMF-ResNet model performance, some critical principles should be determined, including parameter proportion of different branch, and the width of the model. We test the model performance on the CIFAR-100 benchmark dataset to evaluate these principles.

**Parameter proportion of different branches on model performance.** In each multi-branch fusion residual block, the three branches represent different scale representation features. In these branches, one is parameter-free, and the other two branches contain several successive convolutional layers. However, as experimental results showed, simply concatenating these branches together can not achieve the superior effect. Because the parameter proportion of two residual signal branches have a significant on model performance, which means the feature extracted from the two residual branches need to be adjusted to max the capacity of the block. In the original ResNet, deeper bottleneck architecture has similar time complexity compared with basic residual architecture. The deeper bottleneck architecture uses a stack of  $1 \times 1, 3 \times 3$ , and  $1 \times 1$  convolution, where  $1 \times 1$  layers are used to reduce and then increase dimension. Meanwhile, the input and output dimensions are expanded four times compared with the feature map dimension of basic residual architecture. In order to learn semantic information equally from each scale, we construct each branch having the same output feature map size and dimension, which leads to the bottleneck branch having fewer parameters compared with the basic residual architecture branch. In order to max the model performance, we should balance the number of parameters between the two residual

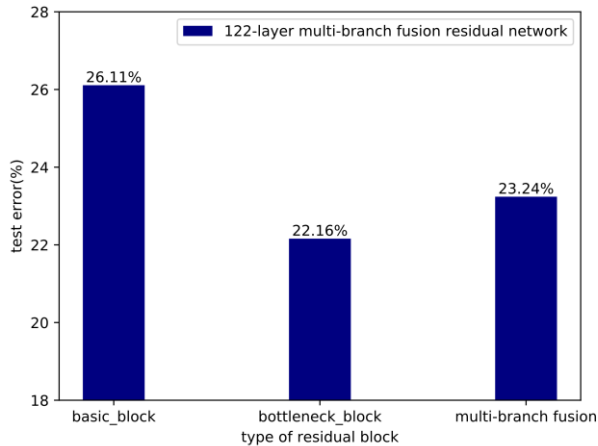


Fig. 2. Test error (%) on CIFAR-100. Multi-branch fusion architecture achieves lower test error than basic architecture, but still higher than bottleneck architecture.

TABLE I  
THE TEST ERROR (%) ON CIFAR-10 WITH DIFFERENT WIDTH UNDER A SIMILAR TOTAL NUMBER OF PARAMETERS.

Width	Params	Error (%)
1	2.54M	23.24
2	2.45M	21.38
3	2.76M	22.22

TABLE II  
**COMPARISON** THE TEST ERROR (%) OF DMF-RESNET WITH DIFFERENT REDUCING RATIO OF  $r$  AND  $t$  UNDER A SIMILAR TOTAL NUMBER OF PARAMETERS ON CIFAR-100. THE MODEL ACHIEVES THE BEST RESULT WHEN  $r = 2$  AND  $t = 4$

$r \backslash t$	1	2	4	8
1	22.66	22.37	22.61	22.41
2	22.69	21.77	21.38	21.82
4	22.37	21.72	21.80	21.92

signal branches. Let us introduce the factor  $r$  and  $t$ , as shown in Fig. 1(c), where  $r$  and  $t$  are ratios of reducing the feature map dimension for basic and bottleneck residual architecture branch, respectively. The experimental results indicate that DMF-ResNets achieve the best performance when  $r=2$  and  $t=4$ , which means decrease the parameter proportion of the basic residual branch can improve the capacity of the block.

**The impact of model width.** Adjusting the parameter proportion from different branches can learn each scale representation effectively. However, as shown in Fig. 2, we test these models with a similar number of parameters. The test error on CIFAR-100 of DMF-ResNet is lower than Pre-ResNet with basic architecture, while it is still higher than Pre-ResNet with bottleneck architecture. We conjectured that the mismatch between model width and model performance leads to this problem. In the original ResNet, to keep similar compute complexity, the feature map dimension of  $1 \times 1$  convolution layer are expanded. Thus, the feature extracted capacity of the bottleneck branch in multi-branch fusion residual block is not maximized. In order to address this problem, we constructed the model with different width under a similar total number of

parameters, and Table I shows the experimental results. As the results showed, the model achieves the best result when the model width is 2. Therefore, we adopt width = 2 for DMF-ResNet in the following experiments (except where otherwise stated).

#### IV. EXPERIMENT AND ANALYSIS

We first reported the influence of hyper-parameters on model performance in this section. Then, we empirically demonstrated the effectiveness of our approach on CIFAR datasets and investigated the impact of model depth and width. Subsequently, we implemented some ablation experiments to verify the validness of multi-branch fusion and SFR module. Based on these explorations, we further constructed DMF-ResNets for high-resolution image classification tasks and evaluated these models' accuracy performance on IP102 dataset.

##### A. Influence of hyper-parameters

For the sake of exploring the impact of the two hyper-parameters (reducing ratio  $r$  and  $t$ ) on model performance, we constructed some ablation experiments on CIFAR-100 datasets under a similar total number of parameters. Meanwhile, to eliminate the interference, we constructed these models without the SFR module, and the experimental results are shown in Table II. As the results showed, increasing the value of  $r$  and  $t$  at the same time can bring benefits to our model, and the model achieves the best test error result as  $r = 2$  and  $t = 4$ . It means the multi-branch fusion residual block **learns** multi-scale representation more **effectively** with  $r = 2$  and  $t = 4$ . Under this condition, we calculate the number of parameters for each residual branch. The result shows that the bottleneck branch extracts feature with fewer parameters than the basic branch. Thus, it demonstrated that the parameter effectiveness of the bottleneck branch is higher than the basic branch.

##### B. Implementations on CIFAR datasets

To demonstrate the validness of our method, we first evaluated our models on CIFAR-10 and CIFAR-100 benchmark datasets. For CIFAR datasets, the weights are initialized by Kaiming Xavier algorithm [39], and all models adopted SGD algorithm. The momentum is 0.9, with a min-batch size of 128. The learning rate is set to be 0.1, and it drops to 1/10 and 1/100 at 150th and 225th, ending at 300 epochs. All experiments are constructed on Pytorch platform. Meanwhile, in multi-branch fusion residual block, the final settings of model performance depended on four hyper-parameters: the fold-ratio of  $m$ , the feature dimension reducing factor of  $r$  and  $t$ , and the reduction ratio of  $d$ . In our experiments, the model performance is not very sensitive to the fold-ratio of  $m$ . As the experimental results showed, the model obtains the best result when  $m = 8$ . Thus, we select  $m = 8$  for CIFAR-10 and CIFAR-100 datasets in the following experiments. Following the setting of the reduction ratio in SENet [36], we set  $d = 4$  for CIFAR-10 and CIFAR-100 datasets. The effect of the reduction ratio of  $r$  and  $t$  has been discussed in the previous section.

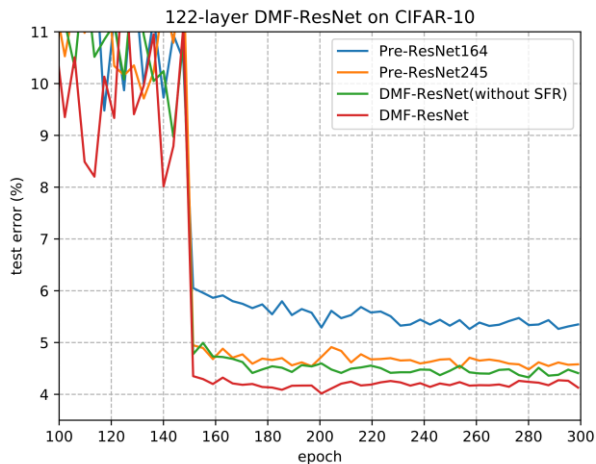


Fig. 3. Test error curves on CIFAR-10 by 164-layer Pre-ResNet, 245-layer Pre-ResNet, 122-layer DMF-ResNet without SFR module and 122-layer DMF-ResNet during training, corresponding to results in Table III. The 122-layer DMF-ResNet (the red curve) is shown yielding a lower test error than other models.

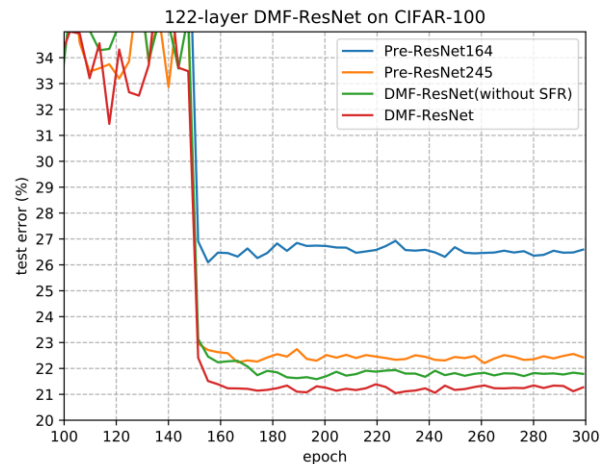


Fig. 4. Test error curves on CIFAR-100 by 164-layer Pre-ResNet, 245-layer Pre-ResNet, 122-layer DMF-ResNet without SFR module and 122-layer DMF-ResNet during training, corresponding to results in Table III. The 122-layer DMF-ResNet (the red curve) is shown yielding a lower test error than other models.

TABLE III  
TEST ERROR (%) ON CIFAR-10 AND CIFAR-100 DATASETS.

	Params	CIFAR-10 Error (%)	CIFAR-100 Error (%)
164-layer Pre-ResNet (basic architecture)	2.6M	5.23	26.11
245-layer Pre-ResNet (bottleneck architecture)	2.5M	4.42	22.16
122-layer DMF-ResNet (without SFR)	2.4M	4.32	21.38
122-layer DMF-ResNet	2.7M	4.01	20.85

### C. CIFAR classification by DMF-ResNet

CIFAR-10 and CIFAR-100 datasets consist of 50K training and 10K testing colored natural scene images, with  $32 \times 32$  pixels each. CIFAR-10 dataset contains 10 classes, each of which consists of 6000 images. CIFAR-100 dataset contains 100 classes, each of which consists of 600 images. We adopted the standard data augmentation strategy, which is widely used for these datasets. For each image, 4 pixels are padded on each side to form an image with a size of  $40 \times 40$ . Then, a random  $32 \times 32$  crop is applied to produce  $32 \times 32$  images with horizontally mirroring half of the image. Mean and standard deviation normalization are also adopted.

We experimented with four models on CIFAR datasets: 164-layer Pre-ResNet with basic residual block, 245-layer Pre-ResNet with bottleneck residual block, 122-layer DMF-ResNet without SFR module, and 122-layer DMF-ResNet. All these models have a similar total number of parameters, and the test error performance and training curves on CIFAR datasets are showed in Fig. 3, Fig. 4, and Table III. As the results shown, the 164-layer Pre-ResNet with basic residual block had a 5.23% and 26.11% test error on CIFAR-10 and CIFAR-100, respectively. The 245-layer Pre-ResNet with bottleneck residual block achieved a competitive 4.42% and 22.16% test error on CIFAR-10 and CIFAR-100, respectively, which is better than Pre-ResNet with basic residual block. The 122-layer

TABLE IV  
TEST ERROR (%) ON CIFAR-10 AND CIFAR-100 WITH DIFFERENT DEPTH

Depth	Params	CIFAR-10 Error (%)	CIFAR-100 Error (%)
122	2.7M	4.01	20.85
182	4.1M	3.97	20.77
302	6.8M	3.82	20.16
1052 (batch-size=32)	23.7M	3.63	18.73

TABLE V  
TEST ERROR (%) ON CIFAR-10 AND CIFAR-100 WITH DIFFERENT WIDTH.

Depth and Width	Params	CIFAR-10 Error (%)	CIFAR-100 Error (%)
122-2	2.7M	4.01	20.85
122-4	10.8M	3.77	19.12
122-8	42.9M	3.52	17.51
302-4	26.9M	3.53	18.26
122-8+mixup	42.9M	2.60	16.88

DMF-ResNet without SFR module outperformed 245-layer Pre-ResNet with bottleneck residual block by 0.1% on CIFAR-10 and 0.78% on CIFAR-100, which demonstrated that fusing the extracted feature from three branches can bring benefits for model performance. Meanwhile, the 122-layer DMF-ResNet with the SFR module achieved better test error performance than without the SFR module, and it achieved 4.01% and 20.85% test error on CIFAR-10 and CIFAR-100, respectively. Therefore, the results demonstrate that the SFR module also can bring improvements to our model.

### D. The impact of depth and width

In order to explore the effect of depth and width on DMF-ResNet, we implemented the following experiments. These experiments are evaluated on CIFAR datasets, and Table IV and Table V report the experimental results. As results showed,

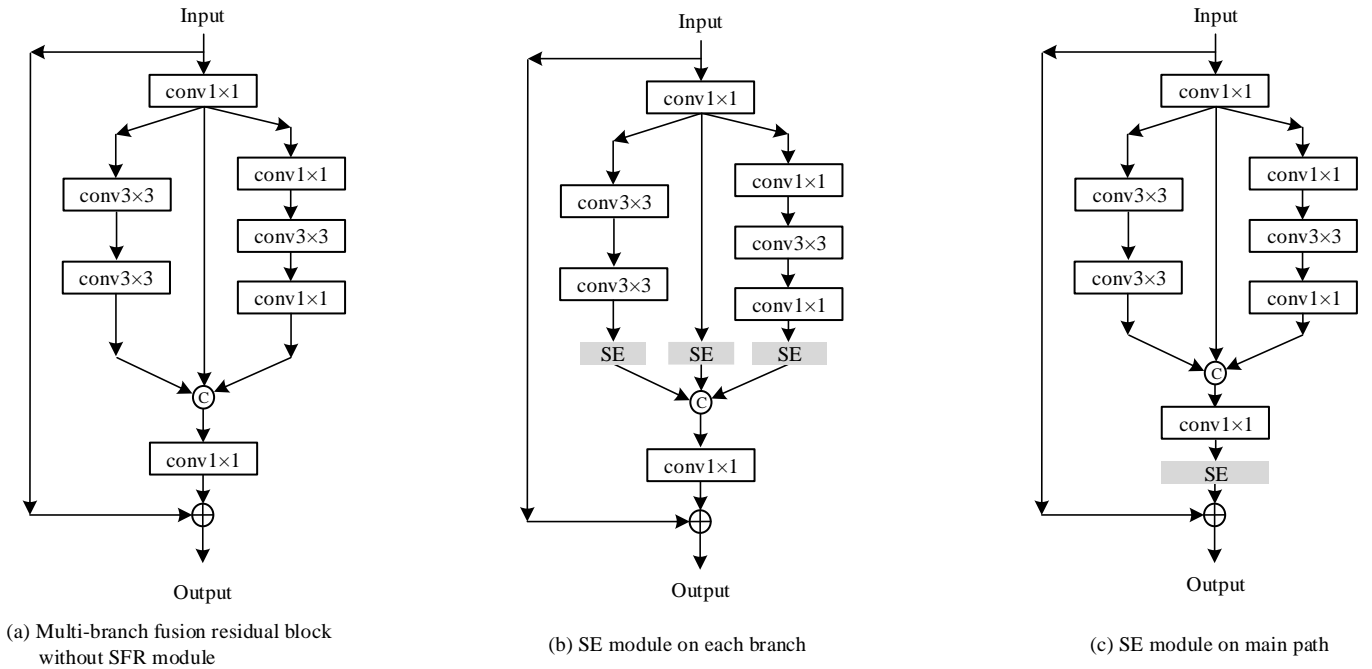


Fig. 5. Structure of multi-branch fusion residual block with the SE module in different localization for ablation study, corresponding to results in Table VII.

increasing depth or width could bring benefits to our models.

In terms of depth, we implemented 122-layer, 182-layer, 302-layer, and 1052-layer DMF-ResNet to explore the influence of depth for our model. As results are showed in Table IV, the test error gradually decreased on CIFAR-10 and CIFAR-100 datasets when depth increased. The 302-layer DMF-ResNet had a 3.82% test error on CIFAR-10 test set and a 20.16% test error on CIFAR-100 test set. For extremely deep DMF-ResNet, due to the limited resource, the 1052-layer model was trained with a batch-size of 32, and it can still achieve a 3.63% test error on CIFAR-10 test set and 18.73% test error on CIFAR-100 test set. Based on these experimental results, we can conclude that increasing the depth can bring benefits for our model performance.

In terms of width, we implemented DMF-ResNet with a different width to explore the influence of width for our model. As the results are showed in Table V, the test error gradually decreased on CIFAR-10, and CIFAR-100 datasets as the width increased on 122-layer DMF-ResNet. The DMF-ResNet-122-8 had a 3.52% test error on CIFAR-10 and 17.51% test error on CIFAR-100. Meanwhile, we also constructed the DMF-ResNet-302-4, and it had a 3.53% test error on CIFAR-10 test set and 18.26% test error on CIFAR-100 test set. Thus, the result indicates that increasing the depth and width at the same time can also improve model performance. On the other hand, augmentation methods, such as mixup [40], can further improve model performance. The DMF-ResNet122-8+mixup had the 2.60% test error and 16.88% test error on CIFAR-10 and CIFAR-100, respectively.

#### E. Ablation Study

For the sake of demonstrating the effectiveness of the multi-branch fusion and SFR module, we constructed some ablation experiments as follows. These experiments are evaluated on

TABLE VI  
TEST ERROR (%) ON CIFAR-100 TO DEMONSTRATE THE EFFECTIVENESS OF MULTI-BRANCH FUSION.

Model	Params	CIFAR-100 Error (%)
A (basic)	2.6M	26.11
B (basic, $r=2$ )	2.5M	25.62
C (basic, $r=2, w=2$ )	2.5M	25.04
D (bottleneck)	2.6M	22.16
122-layer DMF-ResNet (without SFR)	2.4M	21.38

TABLE VII  
TEST ERROR (%) ON CIFAR-100 TO DEMONSTRATE THE EFFECTIVENESS OF SFR MODULE.

Model	Params	CIFAR-100 Error (%)
(a) without SFR module	2.5M	21.38
(b) SE on each branch	2.7M	21.12
(c) SE on main path	2.5M	21.66
122-layer DMF-ResNet	2.7M	20.85

CIFAR-100 dataset.

**Multi-branch Fusion.** In DMF-ResNet, we fused three branches between two  $1 \times 1$  convolutional layer. One branch is parameter-free, and the other two are residual signals with basic residual block or bottleneck residual block. In order to demonstrate the effectiveness of the multi-branch fusion approach, we constructed four models to compare with 122-layer DMF-ResNet. Model A is the 164-layer Pre-ResNet with basic residual block. Model B is the 164-layer Pre-ResNet with basic residual block and channel reduction ratio of 2 ( $r=2$ ). Doubling the width of model B, we obtained model C. Model D is the 254-layer Pre-ResNet with bottleneck residual block.

TABLE VIII  
DEEP MULTI-BRANCH FUSION RESIDUAL NETWORK ARCHITECTURES CONFIGURATION.

Layer name	Output size	77-layer	97-layer	117-layer
Conv1_x	112×112	7×7, 64, stride 2		
		3×3, max pool, stride 2		
Conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 128 \end{bmatrix} \times 5$ , $\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 128 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 128 \end{bmatrix} \times 5$ , $\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 128 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 128 \end{bmatrix} \times 5$ , $\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 128 \end{bmatrix} \times 5$
Conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ , $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 256 \end{bmatrix} \times 10$ , $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 10$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 256 \end{bmatrix} \times 14$ , $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 14$
Conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 512 \end{bmatrix} \times 4$ , $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 512 \end{bmatrix} \times 4$ , $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 512 \end{bmatrix} \times 4$ , $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
	1×1	Average pool, 102-d fc, softmax		

The results are reported in Table VI. All models are constructed under a similar total number of parameters. Compared to model A with model B, channel reduction between two 3×3 convolution layers make the model deeper under a similar number of parameters, which improves the model performance. Compared to model B with model C, increasing width has the same effect. However, model D had the lowest test error than the other three models. DMF-ResNet without SFR module contains the residual signal in models C and D, and it has a 21.38% test error on CIFAR-100 test set. The result outperforms model D by 0.78%. Therefore, it demonstrated the effectiveness of our multi-branch fusion approach.

**Impact of SFR module.** For the sake of verifying the effectiveness of the SFR module, we implemented some ablation experiments. We constructed three models to compare with DMF-ResNet, as shown in Fig. 5. To simplify, we omit the batch normalization and relu layer. The first residual block is the multi-branch fusion residual block without the SFR module. The second residual block adds a SE block in each branch. The third residual block only adds a SE block in the main residual signal. All these models are constructed under a similar total number of parameters and tested model performance on CIFAR-100 dataset. Table VII shows the experimental results. As the results showed, compared (a) with (c), adding a SE block in the main signal can not improve the model performance. Compared (a) with (b), adding a SE block in each branch can bring benefits to model performance. However, model (b) is short of modeling the relationship between three branches. The 122-layer DMF-ResNet had a 20.85% test error on CIFAR-100 test set, which outperforms model (b) by 0.27%. Based on these analyses, we can conclude that the SFR module can enhance our model performance effectively.

#### F. Classification results on IP102

Based on the previous experiments and analyses, to further demonstrate the effectiveness of our approach, we constructed DMF-ResNet for high-resolution image classification tasks. Then we applied it in a specific domain to recognize insect pests.

For the high-resolution image classification task, we implemented DMF-ResNet with different depths, which of the overall architectures are listed in Table VIII. As described in the previous section, doubling the width of the model can reduce the test error significantly under a similar total number of parameters. However, increasing the width will introduce



Fig. 6. The example images from IP102. Each image belongs to a different Category of insect pests.

more parameters. Therefore, in order to control the number of parameters, we constructed these models with three stages, which is different from the original ResNet with four stages. We increased the number of multi-branch residual blocks in the second stage to construct models with different depths, including 77-layer, 97-layer, and 117-layer DMF-ResNet. Meanwhile, in our experiments, we select  $m = 16$  in the following experiments. Following the setting of the reduction ratio in SENet [36], we set  $d = 16$  for IP102 dataset.

As a large-scale insect pest dataset, IP102 consists of 45,095 training images, 7,508 validation images, and 22,619 testing images with 102 classes of common crop insect pests for the classification task. Fig. 6 shows some examples from IP102 dataset. The dataset involves several properties, which bring difficulties for insect pest classification tasks. First, due to the similar color of pests and background, it is difficult to be distinguished. Second, different growth forms of the same type of insect pests are collected into the same category. Third, the similarity between different types of insect pests is high. Therefore, the rich extracted feature from insect pest images will help distinguish similar pest species. So, we hypothesis that learning multi-scale representation will be conducive to model classification performance. The experimental results also demonstrate this hypothesis, as shown in the following section.

We evaluated our models with different depths on IP102 dataset. We adopt 0.01 as the initial learning rate, and it drops



to 1/10 and 1/100 at 40th and 80th, ending at 120 epochs. SGD is used with a batch-size of 32. The weight decay is set to be 0.0005, and the momentum is set to be 0.9. For data augmentation strategies, we adopt the following common approaches in the training phase. First, the image is randomly cropped a rectangular region, which is randomly sampled in [3/4, 4/3] and area randomly sampled in [0.08, 1] from resized 256×256 squared image. Second, the cropped region is resized into the size of 224×224. Third, randomly horizontal flips and standard deviation normalization are also applied in these experiments. During the evaluation period, the cropped 224×224 region from the center of the resized 256×256 image is used to classify. In these experiments, the model is trained on the training set and evaluated on the validation set to obtain the optimization model. Then we acquire the accuracy performance on the test set.

We implement three DMF-ResNets with different depths to compare with ResNet, Pre-ResNet, and other state-of-art methods. The accuracy performance on the test set is reported in Table IX, and Fig. 7 shows the test accuracy curves on the evaluation set during training. As the results showed, 77-layer DMF-ResNet achieved a 57.67 F1 scores and 58.48% test accuracy on the test set surpassing 50-layer ResNet 1.26 and 1.09% respective with fewer parameters. Meanwhile, the 77-layer DMF-ResNet achieved better model performance than other state-of-art methods. As the depth going deep, the DMF-ResNet test accuracy and F1 score increased. The 117-layer DMF-ResNet had a 58.37 F1 score and 59.22% test accuracy on the test set. Based on these experiments, we empirically demonstrated the validness of our approach to IP102 dataset.

For the sake of visualizing the effect of our approach, we use the technique of Grad-Cam [41] to highlight the important regions in the image for our insect pest classification task. To evaluate the effect of multi-fusion method and SFR module more clearly, we compare results from three models, including ResNet-50, DMF-ResNet without SFR module, and DMF-ResNet. We randomly select some images from IP102 dataset, and Table X presents the results. Compared ResNet-50 with DMF-ResNet without SFR module columns, the highlighted region in DMF-ResNet without SFR moduel column is wider than ResNet-50. It indicates that multi-scale learning obtained more abundant extracted features for the classification task. Compared DMF-ResNet without SFR module with DMF-ResNet columns, we can observe that the highlighted regions are finetuned to acquire more precise information for our task. Therefore, based on these analyses, we further demonstrate the effectiveness of our approach by visualizing the important regions for our task.

## V. DISSCUSSION

In this section, we further discuss the effectiveness of our approach in two folds. First, we will discuss the effectiveness of the multi-branch fusion and SFR module. Second, we will discuss the parameter efficiency.

TABLE IX  
THE F1 SCORE AND TEST ACCURACY (%) ON IP102 DATASET BY DMF-RESNET AND OTHER STATE-OF-ART METHODS.

P102	Depth	Params	F1	Acc (%)
AlexNet [42]	8	57.42M	48.08	49.63
ResNet-50 [10]	50	23.72M	56.41	57.39
ResNet-101 [10]	101	42.63M	55.37	56.02
Pre-ResNet-50 [11]	50	23.70M	55.18	55.86
VGG-16 [16]	16	134.68M	53.18	54.43
Densenet-121 [18]	121	7.06M	56.81	57.73
DMF-ResNet	77	22.10M	57.67	58.48
	97	25.96M	58.06	59.11
	117	29.70M	58.37	59.22

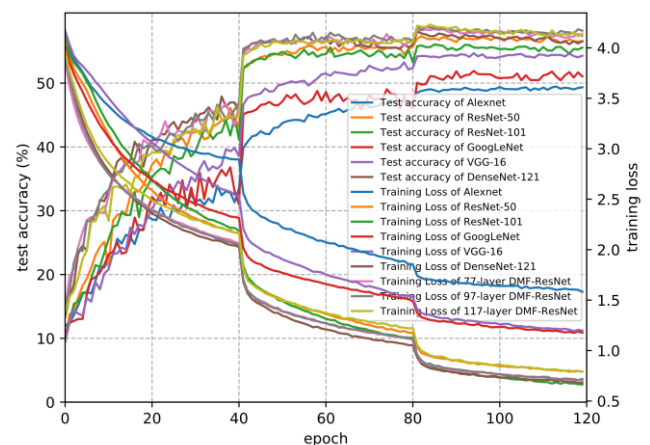


Fig. 7. The evaluation curves on IP102 dataset by DMF-ResNet and other state-of-art methods during training period.


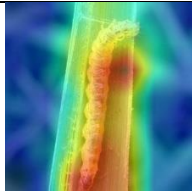
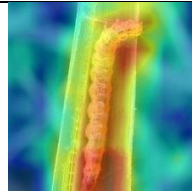
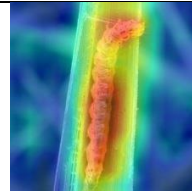

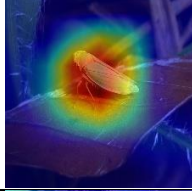
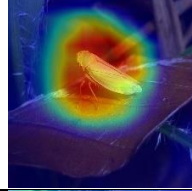
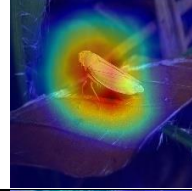

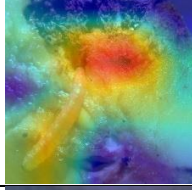
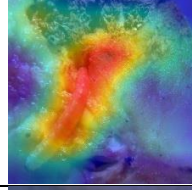
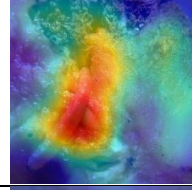

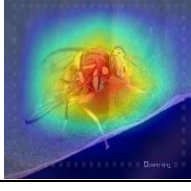
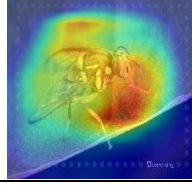
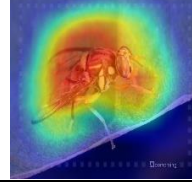
### A. The effectiveness of the multi-branch fusion and SFR module

Compared with the original ResNet, DMF-ResNet combined the extracted feature from three branches to learn the multi-scale representation. The experimental results in Table III and Table VI supported that learning multi-scale representation can enrich the feature for the classification task. Furthermore, the SFR module can further improve model performance. Through visualizing the effect of these approaches on some images, as shown in Table X, the wider highlighted regions indicate that the receptive field is enlarged by learning the multi-scale representation. Meanwhile, the SFR module can finetune the response region by recalibrating channel-wise feature responses and modeling the relationship of these branches. Based on the result shown in Table X, the highlight region is wider and precise. Therefore, it can be used to extract features in two-stage fine-grained image classification models to localize the object more accurately.

### B. Parameter efficiency

In each multi-branch fusion residual block, we explored the different branches of parameter proportion on model performance, and the test error performance is reported in Table

TABLE X  
THE HIGHLIGHTED IMPORTANT REGION.

ID	Original image	ResNet-50	DMF-ResNet (without SFR)	DMF-ResNet
1				
2				
3				
4				

II. As the results showed, the model achieves the best model performance as  $t=2$  and  $t=4$ . Therefore, compared to the bottleneck branch with the basic branch, the ratio of the number of parameters in two branches is 1:9. Because these branches have the same feature dimension, thus each branch provided the same semantic information for model performance. Based on these analyses, we can obtain that the extracting feature capacity of the bottleneck branch is more effective than basic branch with fewer parameters. So, we can conclude that a more effective convolution branch can further enhance the model performance, and the results also provide our direction to improve our model performance in the future.

## VI. CONCLUSION

In our work, to learn the multi-scale representation to improve the model performance, we fused the extracted feature from three branches in each residual block. Moreover, we proposed the SFR module to recalibrate channel-wise feature responses and to model the relationship between these branches. The experimental results verified the effectiveness of our approach on CIFAR-10 and CIFAR-100 datasets. Even for extremely deep DMF-ResNet, our model can achieve compelling results. Then, we constructed our model with different depths and tested the F1 score and model accuracy on IP102 dataset. Compared to the baseline models and other state-of-art methods, our model can obtain the best model performance on IP102 dataset, which had proved the validness of our approach for the high-resolution image classification task.

Through visualizing the highlighted regions on images, we can further explain the effect of our approach for the image classification task. Therefore, based on these empirical studies, we have verified the effectiveness of our approach.

In the future, we will try to construct a more effective feature fusion method and apply the model to perform the fine-grained image classification task.

## 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8787–8796.
- [2] R. Li, R. Wang, J. Zhang, C. Xie, L. Liu, F. Wang, H. Chen, T. Chen, H. Hu, X. Jia et al., "An effective data augmentation strategy for cnn-based pest localization and recognition in the field," IEEE Access, vol. 7, pp. 160274–160283, 2019.
- [3] K. Dimililer and S. Zarrouk, "Icspi: Intelligent classification system of pest insects based on image processing and neural arbitration," Applied Engineering in Agriculture, vol. 33, no. 4, p. 453, 2017.
- [4] Z. Liu, J. Gao, G. Yang, H. Zhang, and Y. He, "Localization and classification of paddy field pests using a saliency map and deep convolutional neural network," Scientific reports, vol. 6, p. 20410, 2016.
- [5] F. Ren, W. Liu, and G. Wu, "Feature reuse residual networks for insect pest recognition," IEEE Access, vol. 7, pp. 122758–122768, 2019.
- [6] Y. Liu, X. Zhang, Y. Lin, and H. Wang, "Facial expression recognition via deep action units graph network based on psychological mechanism," IEEE Transactions on Cognitive and Developmental Systems, 2019.
- [7] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "Gs3d: An efficient 3d object detection framework for autonomous driving," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1019–1028.

- [8] F. Ren, Y. Dong, and W. Wang, "Emotion recognition based on physiological signals using brain asymmetry index and echo state network," *Neural Computing and Applications*, vol. 31, no. 9, pp. 4491–4501, 2019.
- [9] X. Kang, F. Ren, and Y. Wu, "Exploring latent semantic information for textual emotion recognition in blog articles," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 204–216, 2017.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
- [12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.
- [16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [17] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
- [18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [19] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [20] F. Shen, R. Gan, and G. Zeng, "Weighted residuals for very deep networks," in *2016 3rd International Conference on Systems and Informatics (ICSAI)*. IEEE, 2016, pp. 936–941.
- [21] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5927–5935.
- [22] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*. Springer, 2016, pp. 646–661.
- [23] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: Multilevel residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1303–1314, 2017.
- [24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [26] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [27] Z. Wang, L. Zhang, X. Shu, Q. Lv, and Z. Yi, "An end-to-end mammogram diagnosis: a new multi-instance and multi-scale method based on single-image feature," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.
- [28] A. Dimou, D. Ataloglou, K. Dimitropoulos, F. Alvarez, and P. Daras, "Lds-inspired residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2363–2375, 2018.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [30] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Advances in Neural Information Processing Systems*, 2016, pp. 838–846.
- [31] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu et al., "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2956–2964.
- [32] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2219–2228.
- [33] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5659–5667.
- [34] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.
- [35] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.
- [36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [37] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.
- [38] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 510–519.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [40] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
- [41] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations of deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.
- [42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.



**Wenjie Liu** was born in Hunan province, China, in 1989. He received the B.S. degree in information engineering from Nanhang Jincheng college, China, in 2011, the M.S. degree in information and communication engineering from Nantong University, China, in 2014. He is currently pursuing the double Ph.D. degree from 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. degree 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 Sciences, Nantong University, China. His research interests are in the area of Mechanical Engineering, Laser Technology Application, and Artificial Intelligence.



**Fuji Ren** received his Ph. D. degree in 1991 from the Faculty of Engineering, Hokkaido University, Japan. From 1991 to 1994, he worked at CSK as a chief researcher. In 1994, he joined the Faculty of Information Sciences, Hiroshima City University, as an Associate Professor. Since 2001, he has been a Professor of the Faculty of Engineering, Tokushima

University. His current research interests include Natural Language Processing, Artificial Intelligence, Affective Computing, Emotional Robot. He is the Academician of The Engineering Academy of Japan and EU Academy of Sciences. He is a senior member of IEEE, Editor-in-Chief of International Journal of Advanced Intelligence, a vice president of CAAI, and a Fellow of The Japan Federation of Engineering Societies, a Fellow of IEICE, a Fellow of CAAI. He is the President of International Advanced Information Institute, Japan.