Multi-scale Progressive Complementary Fusion Network for Fine-Grained Visual Classification

Jingsheng Lei¹, Xinqi Yang¹ and Shengying Yang¹
¹Zhejiang University of Science and Technology, Hangzhou, 310023, Zhejiang, China

Corresponding author: Shengying Yang (e-mail: syyang@zust.edu.cn).

This work was supported by Natural Science Foundation of China (No.61672337, 61972357), Medical Science and Technology Project of Zhejiang Province (No.2022KY104) and Natural Science Foundation of Xinjiang Uygur Autonomous Region (No.2022D01C349).

ABSTRACT In fine-grained visual classification (FGVC), the small inter-class variations and the large intra-class variations are always inherent attributes, so it is much more challenging than traditional classification tasks. Recent works mainly tackle this problem by employing attention mechanisms to locate the most discriminative parts. However, these methods tend to neglect other inconspicuous but distinguishable parts, and can not effectively fuse the features information of different scales and different degrees of discrimination. In this work, we propose a multi-scale progressive complementary fusion network (MPCF-Net) to tackle these problems. In particular, we propose: (i) A three-step multi-scale progressive training method, which employs the image slicer to generate puzzle images at different scales followed by multi-step progressive training. This enables the network to capture multi-granularity local feature information and gradually expand its attention to the global structural information as the training progresses for multi-granularity information fusion. (ii) A plug-and-play feature complementary enhancement module (FCEM) that explicitly enhances the features extracted by the current layer of the network, while also enabling the next layer of the network to extract potential complementary feature information to diversify the features. Our experiments were conducted on four FGVC benchmark datasets and yielded state-of-the-art or competitive results.

INDEX TERMS Complementary enhancement, Fine-grained visual classification, Feature fusion, Progressive training

I. INTRODUCTION

Fine-grained visual classification (FGVC), also known as sub-category recognition, is a very popular research topic in the fields of computer vision and pattern recognition in recent years. Its purpose is to perform a more detailed sub-category classification of a given object category. Fine-grained images have more precise category accuracy and more subtle differences between categories. Often, different categories can be distinguished only with the help of small local differences. As shown in Figure 1. Therefore, fine-grained visual classification is a very challenging research task.

In order to address the aforementioned challenges, some early solutions [1-7] relied mainly on predefined bounding boxes and part annotations to capture distinguishable regions. Although these methods are efficient, the workload of collecting extra annotated information is very huge and requires professional knowledge, which makes them less practical. Therefore, researchers have recently focused more on weakly-supervised FGVC methods that only need image labels as supervision [8-16], which eliminate the need for expensive annotations and enable the construction of localization sub-networks to locate the most discriminative parts employing attention mechanisms, channel clustering, etc.

However, most of the methods localize only the most discriminative parts but ignore the low-level texture information and other inconspicuous but discriminative parts. Our study shows that because of the high intra-class variance and low inter-class variance in the FGVC tasks, low-level feature information such as texture details, edge junctions, and color overlays of objects and high-level feature information with different distinguishability are necessary for the FGVC tasks. Specifically, as the CNN advances from shallow to deep layers, the activation region of the image gradually shifts from low-level surface texture to high-level deep semantics. However, during the advancement process, detailed information from small areas is lost.
discriminative regions is inevitably lost, which is helpful for the FGVC tasks because it can reflect the subtle differences between different sub-categories. Meanwhile, numerous methods based on attention mechanisms only locate the most discriminative regions, and inconspicuous but equally discriminative regions are often ignored or not effectively captured. These regions, however, are important in the FGVC tasks because they are complementary and can effectively improve the classification ability and robustness of the model.

![FIGURE 1. In FGVC, apparent intra-class variation is illustrated with subtle inter-class variation and distracting factors such as pose, illumination, and occlusion. Images with large differences in each row belong to the same class. However, the images with smaller differences in each column belong to different classes and are accompanied by environmental disturbance situations such as illumination and occlusion as if in the second column of images, making fine-grained visual classification more challenging.](image)

The purpose of this paper is to verify the effect of fusion of low-level feature information with high-level feature information of different distinguishability on fine-grained visual classification. We argue that fusing low-level feature information and complementary high-level feature information can not only avoids losing useful low-level information during training but also capture more discriminative high-level information to enhance the classification ability and robustness of the model. To this end, we propose a new multi-scale progressive complementary fusion network (MPCF-Net), which applies a progressive training method in a weakly-supervised learning manner to enable the network to capture multi-level and multi-scale complementary feature information and perform information fusion. As shown in Figure 2. Specifically, we propose a three-step multi-scale progressive training method, which applies an image slicer to uniformly slice and shuffle the input images, after which images of different slice scales are progressively trained in the order of small to large scales so that different training steps can learn local feature information of different scale sizes. Our method of progressive training from small to large scale can effectively avoid the confusion caused by large intra-class variations. More specifically, this approach enables the network to capture feature information starting from the smallest scale local region in the image until the whole image features are captured. At the end of each training step, the parameters of the current step are passed to the next training step as initialization parameters. This enables the next training step to capture feature information at a larger scale based on the features learned in the previous training step. Until the completion of all training steps, the feature information captured from all steps is fused and output at the last layer of the network. However, although the problem of low-level information loss and multi-scale feature fusion can be effectively solved by using only the multi-scale progressive training method, the inconspicuous yet discriminative high-level semantic information cannot be adequately captured. To this end, we propose a feature complementary enhancement module (FCEM) with two branches, the first of which is able to further enhance the most salient features captured by the current network layer, while the second branch masks the salient features at the current layer and sends them to the subsequent network, forcing the subsequent network layer to capture other complementary features. By inserting FCEM into multiple intermediate layers of the backbone network, we can obtain multiple complementary high-level semantic feature representations in the same training step.

We combine the three-step multi-scale progressive training method with FCEM to form our MPCF-Net model. Different from PMG [17], the number of our progressive training steps is three, and each step incorporates a feature complementary enhancement module to ensure the extraction and fusion of features at different scales and different granularities. In contrast, PMG divides the training step into four steps and only extracts the output of a single stage in each step. Different from CAMF [18], we use a simpler global feature enhancement operation. The enhanced output is obtained by the feature complementary enhancement module while the suppressed output is used as input to the subsequent network for further complementary feature mining, thus allowing feature diversification. CAMF, on the other hand, uses only one enhancement and suppression, which, if under the same feature extraction network, would have fewer parameters but would also greatly reduce the feature diversity. Our method does not need bounding box/part annotations and yields state-of-the-art or competitive results on four FGVC benchmark datasets. Our main contributions can be summarized as follows:

1. We propose a three-step multi-scale progressive training method, which can capture feature information at multiple scales at different stages of different training steps, effectively improving the ability to capture subtle features and fuse feature information at different scales, and our network can be
trained end-to-end.

2. We propose a feature complementary enhancement module (FCEM) that enhances the features captured by the current layer while enabling the next network layer to capture mutually complementary regional features. We combine it with the multi-scale progressive training method to further enhance the network's ability to capture potentially high-level complementary information with distinguishing properties.

3. We effectively fuse the above two methods to form a multiscale progressive complementary fusion network, and our method achieves state-of-the-art or competitive performance on all four standard FGVC benchmark datasets.

![FIGURE 2. An overview of our method, in which Step1, Step2, and Step3 are different training steps. We carry out progressive training in the order of labels. Where $i$ in $y_i$ is the step sequence number, and $j$ is the network layer currently output. We take the enhanced output of stage4 in step1 as $y^i_1$, obtain $y_{concat1}$ after concatenating the complementary enhanced output features of stage4 and stage5 in step2 and obtain $y_{concat2}$ after concatenating the complementary enhanced output features of stage3, stage4, and stage5 in step3. Finally, the outputs of the three steps are summed by the cross-entropy loss function to obtain the maximum prediction as the classification result of our model.]

II. RELATED WORK

In this section, we will describe the most representative methods related to our method.

A. FINE-GRAINED FEATURE LEARNING

In the FGVC task, finding the distinguishing local regions in different categories is beneficial to establish the relationship between regions and object instances, and also to eliminate the influence of distracting factors such as different pose transformations of objects and complex backgrounds. Therefore, the methods based on the localization-classification sub-network are widely studied and applied. The one using more manual annotation information is called the localization-classification sub-network methods with strongly-supervised information. With the further development of deep learning technology, more methods start to use less annotation information to accomplish the task of localization and classification, and these methods are called localization-classification sub-network methods with weakly-supervised information.

B. STRONGLY-SUPERVISED CLASSIFICATION METHODS

Strongly-supervised information classification methods need to use extra labeled bounding boxes or key part points to make the model more focused on feature differences between local regions. Zhang et al. proposed Part R-CNN [1] to train to capture block features from different parts of the image and then concatenate them together as feature vectors to predict fine-grained categories. Wei et al. proposed Mask-CNN [20], which uses a full convolutional network (FCN) to learn a part-based segmentation model (PBSM), combined with part annotation training to obtain the minimum outer rectangle of different parts, cropped and extracted features as the feature representation of the whole image. Although the classification model based on strongly-supervised information achieves satisfactory classification accuracy, the cost of obtaining annotation information is very expensive, so it also limits the practical application of such algorithms to some extent. Therefore, weakly-supervised classification methods, which do not use extra annotation information and can achieve comparable accuracy to strongly-supervised classification methods, are an obvious trend of current research.

C. WEAKLY-SUPERVISED CLASSIFICATION METHODS
Among the weakly-supervised information classification methods, many recent works have focused on the research and application of attention mechanisms. For example, He et al. [25] proposed a multi-scale and multi-granularity deep reinforcement learning approach, which learns multi-granularity discriminative region attention and multi-scale region-based feature representation. Zheng et al. [45] proposed a trilinear attention sampling network: 1) Using self-attention to locate distinguishable image blocks; 2) Enlarge the image with high attention weight to extract more details; 3) The enlarged image block is used to distill the original model and optimize the details. Zhang et al. [46] proposed an attentional convolutional binary neural tree architecture, which uses an attentional transformation module to force the network to capture discriminative features, exhibiting a hierarchical feature learning process from coarse to fine. He et al. [48] proposed a weakly supervised part selection method with spatial constraints. A whole-object detector is learned to automatically localize the target by jointly using saliency extraction and co-segmentation, followed by the selection of distinguishable parts using box constraint and part constraint. He et al. [22] proposed a two-stream model combining vision and language to learn potential semantic representations using the complementary nature of the two streams. Tan et al. [23] proposed a multi-scale selective hierarchical biquadratic pooling method to interact intra and inter-layer features to capture complementary information within features in a multi-scale interaction structure. Zhang et al. [24] proposed a multi-scale erasure and confusion method to perform erasure, segmentation, and confusion operations on sub-regions with different confidence scores to generate multi-scale information images, and finally extract features in the images through a backbone network. Our proposed feature complementary enhancement module (FCEM) designed to enhance features while being the input to output the attention feature map and forces the network to shift its attention to regional features that complement the features of the current layer, which is a significant difference from previous approaches.

D. PROGRESSIVE TRAINING

The progressive training approach was originally proposed in PGGAN [26], which starts with a low-resolution image and gradually adds new layers allowing the model to better refine the details and thus increase the resolution during the training process. This approach allows the network to discover feature information progressively from large scales to small scales, instead of learning information at all scales. In recent years, progressive training methods have been widely used for image generation tasks in hyper-segmentation [27] and generative adversarial networks [28], as it simplifies the information propagation within the network through intermediate supervision. In the FGVC task, Wang et al. [29] proposed a cross-layer progressive attention bilinear fusion method, including the combination of a cross-layer attention module and a cross-layer bilinear fusion module, to represent the features of the distinguished regions by a progressive training approach. Zhao et al. [30] proposed channel attention and progressive multi-granularity training network, which explores the correlation between channels and thus mines meaningful feature maps through the channel attention module, and captures multi-granularity features through the progressive multi-granularity training module. While PMG [17] provides a good control of detailed features, it neglects the effective capture of larger-scale complementary regions.

For FGVC, the fusion of multi-scale information is crucial to the performance of the model. In this work, we employ the idea of multi-scale progressive training to design a single network with shared parameters in three steps, which learns feature information at different scales through three training steps and can fuse the output features from different stages in the feature extraction network. During the training process, the scale of the input puzzle gradually increases, and higher layers are added and trained accordingly to progressively learn information from local details to the global structure of the image.

III. METHOD

In this section, we will describe our proposed method in detail. An overview of our model framework is shown in Figure 2. Our model consists of a three-step multi-scale progressive training framework and a lightweight feature complementary enhancement module: (1) a three-step multi-scale progressive training framework designed to learn multi-granularity feature information at different steps and different scales. (2) A feature complementary enhancement module (FCEM) designed to enhance features while forcing the network to learn more complementary and discriminative regions.

A. MULTI-SCALE PROGRESSIVE TRAINING FRAME

Our network is end-to-end and universal and can be easily implemented on various convolutional neural networks, such as Resnet [42]. Our model is implemented on Resnet50, take it as an example, let’s F is our backbone feature extractor, it has five stages, the deeper the network layers, the richer the semantic information captured by the network. Therefore, we represent the feature maps of the intermediate layer as $F_i \in \mathbb{R}^{H_i \times W_i \times C_i}$, where $H_i, W_i, C_i$ are the height, width, and the number of channels of the feature maps in stage $i$. In our network, $i \in \{3,4,5\}$, i.e., our feature extraction process is performed at stage3, stage4, and stage5. During training, we need to calculate the classification loss of the feature map extracted at each stage:

$$A_i = B_{\text{conv}}^i(F_i) \quad (1)$$
$$A_{\text{concat}1} = \text{concat}[A_3, A_5] \quad (2)$$
$$A_{\text{concat}2} = \text{concat}[A_3, A_4, A_5] \quad (3)$$
$$z_i = \text{softmax}(C_{\text{cls}}(A_i)) \quad (4)$$

Where, $B_{\text{conv}}^i$ is the attention convolution layer, with $F_i$ being the input to output the attention feature map $A_i$. $C_{\text{cls}}$ is
the classifier of stage $i$, $A_{\text{concat}i}$ is the concatenation of stage 4 and stage 5 outputs, $A_{\text{concat}2}$ is the concatenation of stage 3, stage 4, and stage 5 outputs, $z^i$ is the predicted score vector obtained by the classifier through the softmax function.

In order to train the output of each stage and the output of each stage concatenation, we use the cross-entropy loss function $L_{CE}$ between ground truth label $y$ and predicted probability distribution for loss calculation, and the final optimization objectives are:

$$L_{CE}(y, y^i) = - \sum_{c=1}^{M} y^i_c \log (z^i_c)$$  \hspace{1cm} (5)

$$L_{CE}(y, y^{\text{concat}}) = - \sum_{c=1}^{M} y^{\text{concat}}_c \log (z^{\text{concat}}_c)$$ \hspace{1cm} (6)

Where $M$ is the number of object categories and $x = \{1, 2\}$. $y^i$ with $y^{\text{concat}}$ is the predicted value, $y$ is the true value, $y^i_c$ with $y^{\text{concat}}_c$ is the label value and $z^i_c$ with $z^{\text{concat}}_c$ is the predicted score vector for the $c$-th category at stage $i$. We will describe $y^{\text{concat}}$ in detail in part C of this section.

We invoke the strategy of combining progressive training with multi-scale input, and first, we apply a simple image slicer to slice the input images in a random puzzle style with the numbers $P_1$, $P_2$, and $P_3$, respectively, and $P_n$ is given by:

$$P_n = 4^{3-n}$$  \hspace{1cm} (7)

When we generate the puzzle, we reduce the puzzle by $n$ times one-half of the original edge length, so the number of our puzzle is $4^{3-n}$, $n \in \{1, 2, 3\}$, where $n$ is the training step number. Setting the number of puzzle bases in this way makes it easier to calculate while not creating gaps between puzzles. After that, the sliced images are progressively trained in three steps from the low stage to the high stage (the settings of the training steps will be explained in detail in subsection C of Section 4). The low stage and high stage here refer to the small size puzzle step and the large size puzzle step. The low stage is the training step corresponding to $P_1$, while the high stage is the training step corresponding to $P_3$ (the original image), as shown in Figure 3. Due to the limited receptive domain and feature representation capabilities at low stages, the network will be forced to utilize discriminative information from local details (i.e., object textures). In contrast to training the entire network directly, this incremental feature allows the model to incrementally locate feature information from local details to global structure as features are gradually transferred from lower to higher stages, rather than learning all the granularity information simultaneously.

In each iteration, each batch of data will be used for all stages of the three steps. We train the output of stage 4, the last two stages, and the last three stages respectively in each step according to the image segmentation (the smaller the puzzle scale, the fewer the output stages). It should be clear that all parameters used in the current prediction will be iteratively optimized, even if they may have been updated in the previous steps, which can help each network stage in the model work together, that is, the three steps share a CNN to reduce the number of training parameters.

B. FEATURE COMPLEMENTARY ENHANCEMENT MODULE

Our module is divided into two parts. The first part enhances the salient features captured by the current layer, and the second part covers the salient features of the current layer and forces the subsequent networks to capture potential complementary features. The two parts are relatively independent, so they can be explained separately. Given the feature map $F \in R^{H \times W \times C}$ from a specific layer Where $H, W, \ C$ represents height, width, and the number of channels respectively. In the feature enhancement part, inspired by [44], we first need to capture the salient region of the feature map and then conduct element-wise multiplexing with the feature map $F$ to obtain the enhanced feature map. Specifically, the feature map $F$ is used as the input of this module, and the channel-based global max pooling and global average pooling operations are performed to obtain two $H \times W \times 1$ feature maps, respectively.

$$F_{avg} = \text{AvgPool}(F)$$  \hspace{1cm} (8)

$$F_{max} = \text{MaxPool}(F)$$ \hspace{1cm} (9)

Then the two feature maps are concatenated based on the channels. After that, the number of channels is reduced to 1 by a $7 \times 7$ convolution operation, i.e., $H \times W \times 1$. Finally, the spatial attention map $M$ is generated through sigmoid, which represents the most distinguishing region in the feature map $F$. After getting the spatial attention map, we do element-wise multiplication with the input feature map $F$ of this module to get the final generated enhanced feature map $X_1$.

$$X_1 = M(F) = \sigma(f^{7 \times 7}(\text{concat}[F_{avg}, F_{max}]))$$  \hspace{1cm} (10)

where $\sigma$ is a sigmoid function and $f^{7 \times 7}$ indicates a convolution operation with the filter size of $7 \times 7$.

In the feature suppression part, firstly, we divide $F$ into $K$ parts along the $W$ dimension and represent each part as $F_i(0 \in R^{H \times \frac{W}{K}}, i \in [1, K])$ (here $i$ is the $i$-th part of the $K$ parts, and $i$ in subsection $A$ is the $i$-th stage in the backbone network). Then we perform a convolution operation using a convolution kernel of size $1 \times 1$, with the aim of obtaining the importance of each part and changing the number of

![FIGURE 3. Our three-step multi-scale progressive training strategy. Among them, ✠ is the sixteenth puzzle, ✡ is the quarter puzzle and ✢ is the original. The three training steps are progressively trained by sequencing the puzzle sizes from small to large, where the small size puzzle step we call the low stage and the large size puzzle step we call the high stage. And the three steps share a single convolutional neural network.](Image 302x613 to 535x718)
channels of $F(i)$ to 1. Immediately afterward, the negative activation values are removed by the nonlinear function Relu.

$$a_i = \text{Relu}(\text{Conv}_{1 \times 1}(F(i)))$$

(11)

$$a_i = [a_1, a_2, \ldots, a_K] \in \mathbb{R}^{H \times (W/8) 	imes 1},$$ indicating the importance factor of Group $K$ features. $\text{Conv}_{1 \times 1}$ is shared among different parts to judge the importance of different parts. Then, we perform the global average pooling operation on $a_i$ and then perform the softmax function to normalize it to obtain the importance factor $b_i$ of $F(i)$, that is:

$$b_i = \frac{\exp(GAP(a_i))}{\sum_{j=1}^{K} \exp(GAP(a_j))}$$

(12)

At this point, the part most in need of inhibition can be determined immediately, so that the inhibitory factor $b'_i$ can be further obtained:

$$b'_i = \begin{cases} (1 - \beta) b_i, & \text{if } b_i = \text{max}(B) \\ b_i, & \text{otherwise} \end{cases}$$

(13)

$b'_i$ represents the inhibition factor of part $i$ features, and $\beta$ is a hyper-parameter, indicating the degree of inhibition. The higher the value of $\beta$, the greater the degree of inhibition. Finally, the normalized characteristic inhibition factor $B = [b'_1, b'_2, \ldots, b'_K] \in \mathbb{R}^{1 \times (W/8) \times 1}$ are obtained. By inhibiting the most salient part, we can obtain the inhibition feature $X_2$:

$$X_2 = B \otimes F$$

(14)

We emphasize the weight of the sub salient local region by suppressing the weight of the most salient local region. In short, the function of FCEM can be expressed as $FCEM(F) = (X_1, X_2)$. Given feature map $F$, FCEM outputs enhanced feature map $X_1$ and potential feature map $X_2$. Since $X_2$ inhibits the most salient part at the current layer, other potential features will stand out and complement the features captured from the previous layer after $X_2$ enters the next layer. The diagram of FCEM is shown in Figure 4.

C. NETWORK DESIGN AND INFECTION

Our network combines the multi-scale progressive training framework with FCEM to form our MPCF-Net. Specifically, we insert our FCEM into the end of stage3, stage4, and stage5 of Resnet50 in each training step, and extract the feature enhanced output $y^j_i$ at specific steps and network stages, $i$ is the training step, $i \in \{1, 2, 3\}$, and $j$ is the number of stages of the feature extraction network Resnet50. In our network design, $j \in \{3, 4, 5\}$. Due to the limited acceptance domain and representation ability of the low stage, in step1, we only obtain the $y^1_1$ output by feature enhancement after the end of stage4. In step2, due to the further improvement of local acceptance domain and representation ability, we will extract the feature enhanced outputs $y^2_2$ and $y^2_3$ after the end of stage4 and stage5 in this training step, and obtain $y_{concat1}$ after concatenating the two outputs. In step3, because the original image can best represent the global structure of the object, in this training step, we will extract the feature enhanced outputs $y^3_3$, $y^3_4$, and $y^3_5$ after stage3, stage4, and stage5, and obtain $y_{concat2}$ after concatenating the three outputs. As shown in the second half of Figure 2 (Since $y^c_{concat} \times$ in Eq. (6) involves the number of object categories $c$, $concatx$ is the superscript of $y$; here, $y_{concatx}$ is $y^{concatx}$ in Eq. (6)).

In the inference step, we only input the original image into the trained model without using an image slicer. If we only use $y_{concat2}$ for prediction, we can delete the full connection (FC) layer of the other two steps, resulting in less computing budget. In this case, the final result $C_1$ can be expressed as:

$$C_1 = \text{argmax}(y_{concat})$$

(15)

where $y_{concat}$ represents the output of step3, i.e., the step where the original image is the input. The output of the argmax function, $\hat{C}_1$, represents the category ID corresponding to the maximum of the predicted values of $y_{concat}$. The prediction of a single-stage based on specific granularity information is unique and supplementary, which will achieve better performance when we combine all outputs with the same weight. Multi-output combined prediction $C_2$ can be written as:

$$C_2 = \text{argmax}(y^1_1 + y_{concat1} + y_{concat2})$$

(16)

$C_2$ represents the category ID corresponding to the maximum value obtained after summing the predicted values from all three steps.

IV. EXPERIMENTS

In this section, we will show some comprehensive experiments to verify the performance of MPCF-Net. First, the four fine-grained benchmark data sets and implementation details we used are introduced in subsection A. Then, in subsection B we compare our model with the latest method on four published fine-grained visual classification data sets. Then in subsection C, we discuss the contribution of the proposed module and the performance comparison of different network structures. Finally, in
subsection D, we provide FCEM and MPCF-Net model visualization experiments.

A. DATASETS AND IMPLEMENTATION DETAILS

1) DATASETS

Our experiments were conducted on four public FGVC benchmark datasets, including CUB-200-2011, FGVC-Aircraft, Stanford Cars, and Stanford Dogs. The specific information of each data set is shown in Table I.

2) IMPLEMENT DETAILS

We performed all experiments on GTX1080ti GPU using Pytorch with a version greater than 1.5. Our method was evaluated with Resnet50 as the backbone. In order to obtain the best performance, we set the number of training steps \(i = 3\), the number of output stages \(j = 3\), and \(\beta = 0.5\). The category label of the image is the only annotation for training. During training, the input images are resized to 550 × 550 and randomly cropped to 448 × 448. We apply random horizontal flips to augment the trainset. During testing, the input images are resized to 550 × 550 and cropped from the center into 448 × 448. All the above settings are standard settings in the literature.

We use stochastic gradient descent (SGD) optimizer and batch normalization as regularizers. At the same time, the learning rates of the newly added convolution layers and FC layers are initialized to 0.002 respectively and reduced according to the cosine annealing plan during the training process. The learning rate of the pre-trained convolution layer is 0.0002. For all the above models, we train them up to 100 epochs, the batch size is 16, and the weight attenuation is 0.00001 and the momentum is 0.9.

B. COMPARISONS WITH STATE-OF-THE-ART METHODS

The most advanced comparison between our method and other most advanced methods on CUB-200-2011, FGVC-Aircraft, Stanford Cars, and Stanford Dogs are shown in Table II.

1) CUB-200-2011

CUB-200-2011 is the most challenging benchmark measurement dataset in FGVC, and our Resnet50-based model achieves competitive results on this dataset. Compared with DeepLAC[31] and Part-RCNN[1] using predefined bounding boxes and part annotations, our method outperforms them by 8.8% and 7.5%, respectively. Compared with RA-CNN[12], NTS-Net[10], MGE-CNN[36], S3N[21], and FDL[37], which are two-stage methods, our model outperforms them by 3.8%, 1.6%, 0.6%, 0.6%, and 0.5%, respectively. Compared with ISQRT-COV[2] and DTB-Net[35], which explore higher-order information to capture subtle features, our approach outperforms them by 1% versus 1.6%. Compared with API-Net[33] and Cross-X[34], which distinguish features by pairwise interactions, our model gets a 1.4% improvement. Compared with MA-CNN[9], CIN[47], and LIO[16], our method outperforms them by 2.6%, 1.7%, and 1.1%, respectively. However, it lags behind PMG by 0.2% due to the fact that CUB-200-2011 has more complex and smaller objects, and our model is slightly less sensitive to higher-order complementary information for small-scale objects than for large-scale objects. PMG[17] makes more use of small-scale detail information and thus is more robust to small-scale objects, but this also makes it lack higher-order complementary information on other datasets with more significant objects, with loss of classification accuracy.

2) FGVC-AIRCRAFT

On this task, our method achieves state-of-the-art performance. Compared with RA-CNN and MA-CNN, our method outperforms them by 5.5% and 3.7%, respectively. Using the Resnet50 backbone, our model outperforms WS-DAN[39], DFL[38], DCL[19], Cross-X, and API-Net by 0.6%, 0.5%, 0.6%, 1.0%, and 0.6%, respectively. Our one-stage method outperforms the two-stage methods FDL and S3N by 0.2% and 0.8%, respectively, demonstrating that our model has better performance with a smaller number of parameters.

3) STANFORD CARS

Our method achieves state-of-the-art performance in this dataset as well. Our method outperforms KP[32], RA-CNN, and MA-CNN using VGG as the backbone by a large margin. Using the Resnet50 backbone, the results of our method outperform the ISQRT-COV, Cross-X, DCL, LIO, S3N, MC-Loss[40], WS-DAN, FBSD[41], and PMG methods to varying degrees. It proves that our model has superior performance on more rigid datasets.

4) STANFORD DOGS

Since the training cost of this dataset is the highest in the four benchmark datasets, many methods do not report the results on this dataset. Our method obtains a competitive result on this dataset and far outperforms RA-CNN, MAMC[43], and S3N, but lags behind by 0.6% compared to Cross-X. Cross-X samples inter-class and intra-class images by designing a non-trivial data sampler and uses the relationship between different images for multi-feature learning, which makes it more robust on such complex dataset. In contrast, our model lacks in the interaction between images, but our model does not need to design an additional data sampler, so we have a lower training cost.

In conclusion, due to the simplicity and validity of our model, it scales well to all four benchmark datasets. Using the Resnet50 backbone, PMG obtains the best results on CUB-200-2011 but performs poorly on FGVC-Aircraft. Cross-X obtains the best results on Stanford Dogs, but the results on the other three datasets need to be improved. Our model showed the best performance on both Stanford Cars and FGVC-Aircraft, and the results on the other two datasets were relatively competitive.
TABLE II
COMPAred with state-of-the-art methods on four fine-grained Benchmark datasets.

| Methods | Backbone | CUB-200-2011 | FGVC-Aircraft | Stanford Cars | Stanford Dogs |
|---------|----------|--------------|---------------|---------------|---------------|
| DeepLAC | VGG      | 80.3         | -             | -             | -             |
| Part-RCNN | VGG     | 81.6         | -             | -             | -             |
| KP      | VGG      | 86.2         | 86.9          | 92.4          | -             |
| RA-CNN  | VGG      | 85.3         | 88.1          | 92.5          | 87.3          |
| MA-CNN  | VGG      | 86.5         | 89.9          | 92.8          | -             |
| MAMC    | Resnet50 | 86.2         | -             | 92.8          | 84.8          |
| NTS-Net | Resnet50 | 87.5         | 91.4          | 93.3          | -             |
| API-Net | Resnet50 | 87.7         | 93.0          | 94.8          | 88.3          |
| Cross-X | Resnet50 | 87.7         | 92.6          | 94.5          | 88.9          |
| DCL     | Resnet50 | 87.8         | 93.0          | 94.5          | -             |
| DTB-Net | Resnet50 | 87.5         | 91.2          | 94.1          | -             |
| CIN     | Resnet50 | 87.5         | 92.6          | 94.1          | -             |
| LIO     | Resnet50 | 88.0         | 92.7          | 94.5          | -             |
| ISQRT-COV | Resnet50 | 88.1         | 90.0          | 92.8          | -             |
| MGE-CNN | Resnet50 | 88.5         | -             | 93.9          | -             |
| S3N     | Resnet50 | 88.5         | 92.8          | 94.7          | 85.0          |
| FDL     | Resnet50 | 88.6         | 93.4          | 94.3          | 88.2          |
| FT-ResNet | Resnet50 | 84.1         | 91.7          | 88.5          | -             |
| DFL     | Resnet50 | 87.4         | 93.1          | 91.7          | -             |
| WS-DAN  | Resnet50 | 89.0         | 93.0          | 94.3          | 88.4          |
| MC-Loss | Resnet50 | 87.3         | 92.6          | 93.7          | -             |
| PMG     | Resnet50 | 89.3         | 92.8          | 94.9          | -             |
| FBSD    | Resnet50 | 88.9         | 92.7          | 94.4          | 88.2          |
| Ours    | Resnet50 | 89.1         | **93.6**      | **95.0**      | **88.3**      |

C. ABLATION STUDIES
We conducted an ablation study to understand the contribution of each training step and the multiscale progressive training approach as well as our proposed FCEM. We chose the Stanford Cars dataset for our experiments, using Resnet50 as the backbone. The results are shown in Table III. First, we divided the experiments into four groups, with step3 placed in the first group as the baseline, after which each group was sequentially overlaid with one step in the order of the puzzle scale from largest to smallest. From Eq. (7) in subsection A of Section 3, the number of puzzles corresponds to step0 $p_0 = 4^3$. After that, we qualitatively analyzed the contribution of the multiscale progressive training method with FCEM in each group again. In particular, since the first group has only one training step, there is no progressive training method. To obtain multiply differentiated and complementary partially specific feature representations, we inserted FCEM into the last three stages of the backbone network. Our experiments can be easily compared horizontally and vertically to verify the contribution of each part of our model in different dimensions. As can be seen in Table III, we obtain a performance improvement for each additional step. Specifically, for the first group, we obtain 0.8%, 1.2%, and 1.4% improvement after adding three steps in sequence. In the second, third, and fourth sets we obtained 0.3%, 0.5%, and 0.8% improvement using progressive training, respectively. On top of the above, we obtained 0.4%, 0.3%, and 0.6% gains in the first, second, and third groups, respectively, after inserting FCEM, but a 0.1% decrease in the fourth group. We believe that the detailed information fused by step0 will slightly hinder FCEM from mining higher-order complementary information when it reaches the downstream step, so we finally set the specific training steps in the model as step1+step2+step3, as shown in Figure 2.

According to the above analysis, the multi-scale input reflects the advantage of fusing feature information at different scales. The progressive training demonstrates the need for the network to progressively mine small-scale detail information to large-scale global structures. FCEM also verifies its ability to significantly enhance the current output features and enable the network to effectively capture potential higher-order complementary feature information.

D. VISUALIZATION
In this subsection, we apply the Grad-CAM visualization model to perform visualization experiments of FCEM and MPCF-Net model on four datasets, CUB-200-2011, FGVC-Aircraft, Stanford Cars, and Stanford Dogs, using Resnet50 as the backbone network.

To be able to see more intuitively the role played by our feature complementary enhancement module, we will use the output feature maps from the last stage of the network for our visualization experiments. Specifically, the class activation heat map is the region where the magnitude of the output class probability for a given feature map is most sensitive to the pixel values in the input image. We will show in turn the activation map of the direct output of the last stage, the activation map after feature enhancement, and the activation...
map after feature suppression. This is shown in Figure 5. We can see that the activation values of the feature maps after feature enhancement have increased (the higher the activation value, the darker the color temperature, and in areas with darker color temperatures, the difference may not be easily visible visually after the activation value increase, but the actual effect is significant), and the activation value of the high activation areas of the feature map after feature suppression is significantly reduced.

**TABLE III**

| Training strategy | Step3 | Step2 | Step1 | Step0 | FCEM | Acc(%) |
|-------------------|-------|-------|-------|-------|------|--------|
| -                 | ✓     | ✓     | ✓     | ✓     |      | 92.7   |
| Non- Progressive  | ✓     | ✓     | ✓     | ✓     |      | 92.8   |
| Progressive       | ✓     | ✓     | ✓     | ✓     |      | 94.1   |
| Non- Progressive  | ✓     | ✓     | ✓     | ✓     |      | 93.9   |
| Progressive       | ✓     | ✓     | ✓     | ✓     |      | 94.4   |
| Progressive       | ✓     | ✓     | ✓     | ✓     | ✓    | 95.0   |
| Non- Progressive  | ✓     | ✓     | ✓     | ✓     | ✓    | 94.1   |
| Progressive       | ✓     | ✓     | ✓     | ✓     | ✓    | 94.9   |
| Progressive       | ✓     | ✓     | ✓     | ✓     | ✓    | 94.7   |

In order to visualize the superiority of our model more, we will perform experiments to visualize the output feature maps in the baseline model and our model in the last three stages separately (i.e., stage3, stage4, and stage5 mentioned in Section 3). This is shown in Figure 6. We selected one original image on each dataset, and the activation maps in the first to third columns below each image correspond to the visualization results of the baseline model and our model for stage3 to stage5, respectively. As can be seen from the first column of each image, at the stage3, the baseline model is not yet able to capture the detailed texture part of the object effectively and even shifts the focus of attention to irrelevant regions. In contrast, our model has captured many effective small-scale detailed feature information at the stage3. In the stage4, we further extend the attentional focus on the object. By the stage5, the region of interest of our model can cover almost the whole object. In contrast, the baseline model can only focus on a certain part of the object, such as the head of a bird or the rearview mirror of a car.

**FIGURE 5.** The visualization experiments of our feature complementary enhancement module. The first column of each set of experiments is the original image of the data set, the second column is the feature activation map output at the last stage of the network, the third column is the feature enhancement activation map, and the fourth column is the feature suppression activation map.
In summary, in the FCEM visualization experiments, we can well observe the feature enhancement and feature suppression effects, which proves the effectiveness of FCEM. In MPCF-Net model visualization experiments, compared with the activation map of the baseline model, the attention of our model can cover the whole object better, which also reflects the progressive exploration process of attention from small-scale regions to the whole object, validating the effectiveness of the three-step multi-scale progressive training method. In addition, fusing FCEM enables our model to show more different attention regions on the target object to enable the attention of the final stage convolutional layer to cover the whole object, demonstrating the ability of FCEM to enable the network to extract potentially higher-order complementary features.

V. CONCLUSION
In this paper, we apply a progressive training method and propose a three-step multi-scale progressive complementary fusion network model. The model has two main components: (i) A three-step multi-scale progressive training method that aims to learn and fuse multi-grained feature information at different steps and different scales. (ii) A feature complementary enhancement module (FCEM) designed to explicitly augment the output features extracted by the current layer and enable the lower layer networks to learn potentially complementary multiple significant region-specific representations. Our method allows end-to-end training without the need for manual annotation other than category labels and requires only a single propagated network during testing. We conducted experiments on four widely used fine-grained datasets, obtaining state-of-the-art performance on two datasets, FGVC-Aircraft and Stanford Cars, and competitive results on CUB-200-2011, and Stanford Dogs. Our comparative experiments and results demonstrate the superiority of our proposed idea of fusing low-level detailed feature information with high-level complementary feature information. Our ablation experiments further demonstrate the effectiveness of our multi-scale progressive training approach and FCEM.

REFERENCES
[1] Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based r-cnn for fine-grained category detection. In: ECCV (2014)
[2] Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: CVPR. pp. 1449–1457 (2015)
[3] Zhang, H., Xu, T., Elhoseiny, M., Huang, X., Zhang, S., Elgammal, A., Metaxas, D.: Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1143–1152 (2016)
[4] S. Huang, Z. Xu, D. Tao, and Y. Zhang, “Part-stacked cnn for fine-grained visual categorization,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016.
[5] Wei, X.S., Xie, C.W., Wu, J.: Mask-cnn: Localizing parts and selecting descriptors for fine-grained image recognition. arXiv preprint arXiv:1605.06878 (2016)
[6] Lei, J., Duan, J., Wu, F., Ling, N., Hou, C.: Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3d-hevc. IEEE Trans-actions on Circuits and Systems for Video Technology 28(3), 706–718 (2016)
[7] Lam, M., Mahasseni, B., Todorcevic, S.: Fine-grained recognition as instance search for informative image parts. In: CVPR. pp. 2520–2529 (2017)
[8] Zhao, B., Wu, X., Feng, J., Peng, Q., Yan, S.: Diversified visual attention networks for fine-grained object classification. IEEE Transactions on Multimedia 19(6), 1245–1256 (2017)
[9] Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: ICCV. pp. 5209–5217 (2017)
[10] Yang, Z., Luo, T., Wang, D., Hu, Z., Gao, J., Wang, L.: Learning to navigate for fine-grained classification. In: ECCV. pp. 420–435 (2018)
[11] Zheng, H., Fu, J., Zha, Z.J., Luo, J.: Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In: CVPR. pp. 5012–5021 (2019)
[12] J. Fu, H. Zheng, and T. Mei, “Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4476–4484.
[13] C. Liu, H. Xie, Z. Zha, L. Ma, L. Yu, and Y. Zhang, “Filteration and distillation: Enhancing region attention for fine-grained visual categorization,” in Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press, 2020, pp. 11 555–11 562.
[14] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, “Learning to navigate for fine-grained classification,” Lecture Notes in Computer Science, p. 438–454, 2018.
[15] L. Zhang, S. Huang, W. Liu, and D. Tao, “Learning a mixture of granularity-specific experts for fine-grained categorization,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8330–8339.
[16] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, and J. Jiao, “Selective sparse sampling for fine-grained image recognition,” in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6598–6607.
[17] Du R, Chang D, Bhunia A K , et al. Fine-Grained Visual Classification via Progressive Multi-Granularity Training of Jigsaw Patches[J]. 2020.

[18] Miao Z, Zhao X, Wang J, et al. Complementary Attention Multi-Feature Fusion Network for Fine-Grained Classification[J]. IEEE Signal Processing Letters, 2021, 28: 193-197.

[19] Zhu Y, Bai Y, Zhang W, et al. (2019) Destruction and construction learning for fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5157–5166.

[20] Wei, Xiu-Shen, Xie, et al. Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization[J]. Pattern Recognition: The Journal of the Pattern Recognition Society, 2018.

[21] Ding Y, Zhou Y, Zhu Y, et al. Selective Sparse Sampling for Fine-Grained Image Recognition[C]// 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.

[22] He X, Peng Y. Fine-grained image classification via combining vision and language[C]/Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 5994-6002.

[23] Tan M, Yuan F, Yu J, et al. Fine-grained Image Classification via Multi-scale Selective Hierarchical Biquadratic Pooling[J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(1s): 1-23.

[24] Zhang Y, Sun Y, Wang N, et al. MSEC: Multi-Scale Erasure and Confusion for fine-grained image classification[J]. Neurocomputing, 2021, 449: 1-14.

[25] He X, Peng Y, Zhao J. Which and how many regions to gaze: Focus discriminative regions for fine-grained visual categorization[J]. International Journal of Computer Vision, 2019, 127(9): 1235-1255.

[26] Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of discriminative regions for fine-grained image recognition. In: Proceedings of the AAAI conference on artificial intelligence. 2017.

[27] Wang, Y., Perazzi, F., McWilliams, B., Sorkine-Hornung, A., Sorkine-Hornung, O., Schroers, C.: A fully progressive approach to single-image super-resolution. In: CVPR workshops (2018)

[28] Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: ICCV (2019)g

[29] Wang C, Qian Y, Gong W, et al. Cross-layer progressive attention bilinear fusion method for fine-grained visual classification[J]. Journal of Visual Communication and Image Representation, 2022, 82: 103414.

[30] Zhao P, Miao Q, Yao H, et al. CA - PMG: Channel attention and progressive multi-granularity training network for fine-grained visual classification[J]. IET Image Processing, 2021, 15(4): 3718-3727.

[31] D. Lin, X. Shen, C. Lu, and J. Jia, “Deep lac: Deep localization, alignment and classification for fine-grained recognition,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1666–1674.

[32] Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y., Belongie, S.: Kernel pooling for convolutional neural networks. In: CVPR (2017)

[33] P. Zhuang, Y. Wang, and Y. Qiao, "Learning attentive pairwise interaction for fine-grained classification," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, p. 13130–13137, Apr 2020.

[34] W. Luo, X. Yang, X. Mo, Y. Lu, L. S. Davis, and S.-N. Lim, “Cross-x learning for fine-grained visual categorization,” in ICCV, 2019.

[35] H. Zheng, J. Fu, Z-J. Zha, and J. Luo, "Learning deep bilinear transformation for fine-grained image representation," in Advances in Neural Information Processing Systems, 2019, pp. 4277–4286.

[36] Zhang L, Huang S, Liu W, et al. Learning a Mixture of Granularity-Specific Experts for Fine-Grained Categorization[C]/ 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.

[37] Liu C, Xie H, Zha Z J, et al. Filtration and Distillation: Enhancing Region Attention for Fine-Grained Visual Categorization[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7):11555-11562.
Shengying Yang, PhD, lecturer. He graduated with a PhD in Electronic Science and Technology from Hangzhou Dianzi University in 2020, and worked as a lecturer in Zhejiang University of Science and Technology. He mainly engaged in research on big data analysis and power electronic device design.