Progressive Multi-stage Interactive Training in Mobile Network for Fine-grained Recognition

Zhenxin Wu, Qingliang Chen, Yifeng Liu, Yinqi Zhang, Chengkai Zhu, Yang Yu
2533653096@qq.com

CONTENTS

I Introduction

II Related Work
   II-A Fine-grained visual classification
   II-B Data argumentation for FGVC
   II-C Multi-stage feature fusion and progressive training

III The Proposed Approach
   III-A Overview
   III-B Recursive Mosaic Generator
   III-C Progressive multi-stage interaction
      III-C1 Multi-stage interaction
      III-C2 Progressive training strategy

IV Experiments
   IV-A Dataset and Implementation Details
   IV-B Ablation studies
   IV-C Comparative studies
   IV-D Robustness analysis
   IV-E Visualization

V Conclusion

References

LIST OF FIGURES

1 An overview of performance comparison of baseline and RMG-PMSI in CNN backbones on three datasets. . .
2 The structure of PMSI is illustrated. There are $StageNum + 1$ phases at each iteration. Here, we set $StageNum = 3$ for explanation. We set the increase in the number of feature map channels as the division between this stage and the next stage. There are a total of $N$ stages in the backbone. The Smooth Conv Block consists of two convolutional layers which are used to convert feature maps with different channel numbers generated by different stages into same number of channels. The MLP (multi-layer perceptron) has two fully connected layers. The classifier contains two fully connected layers with a softmax layer. The input image of each phase is different (corresponding to input 1 to input 4). Each phase produces different classification results and the common parts of each phase are marked in gray in the figure.
3 Recursive Mosaic Generator (RMG). In each phase, $2 \times 2$ mosaic operation is performed again based on a random patch from the previous phase.
4 Activation map based on the MobilenetV2 with RMG-PMSI. Columns (a)-(c) are CUB, Air, and Car respectively.

LIST OF TABLES

I Statistics of benchmark datasets
II The performance of the proposed model when interacting at different stages.
III The contribution of each component in RMG-PMSI.
IV Accuracy of different data augmentations.
V Comparison with other state-of-the-art fine-grained recognition models
VI The results on other stronger backbones
VII Comparison of anti-interference ability of baseline and RMG-PMSI.
Progressive Multi-stage Interactive Training in Mobile Network for Fine-grained Recognition

Abstract—Fine-grained Visual Classification (FGVC) aims to identify objects from subcategories. It is a very challenging task because of the subtle inter-class differences. Existing research applies large-scale convolutional neural networks or visual transformers as the feature extractor, which is extremely computationally expensive. In fact, real-world scenarios of fine-grained recognition often require a more lightweight mobile network that can be utilized offline. However, the fundamental mobile network feature extraction capability is weaker than large-scale models. In this paper, based on the lightweight MobilenetV2, we propose a Progressive Multi-Stage Interactive training method with a Recursive Mosaic Generator (RMG-PMSI). First, we propose a Recursive Mosaic Generator (RMG) that generates images with different granularities in different phases. Then, the features of different stages pass through a Multi-Stage Interaction (MSI) module, which strengthens and complements the corresponding features of different stages. Finally, using the progressive training (P), the features extracted by the model in different stages can be fully utilized and fused with each other. Experiments on three prestigious fine-grained benchmarks show that RMG-PMSI can significantly improve the performance with good robustness and transferability.

I. INTRODUCTION

Fine-grained Classification aims to identify different sub-categories of the same category, e.g., different kinds of birds, cars or planes. Early work mainly used the method of strong supervision in which people manually marked specific regions [2], [4], [16], [33], [34], [40]. These methods required a lot of manpower and were prone to errors, which led to performance degradation [19]. Therefore, this field of research has gradually shifted to weak supervision approaches that do not require explicit labeling of regions [5], [8], [12], [19], [36], [41], [44]. These efforts incorporate features from different regions or pair of interactive comparative learning to make features more discriminative, resulting in significant performance improvements.

However, the existing approaches mentioned above mainly focus on using large-scale CNN [13], [14], [29], [35] or visual transformer [7] as feature extractor, which is well-known to be computationally expensive. In fact, fine-grained classification scenarios often require a more lightweight mobile network that can be applied offline. For example, ornithologists conducting field research need to be able to use mobile devices to quickly identify birds they find. In the field of intelligent traffic, traffic police also rely on mobile devices to quickly identify vehicle models when performing tasks. However, the large-scale CNNs has a hefty price tag and makes it difficult to recognize objects in real-time. Therefore, we need new mobile networks to meet the requirements for lightweight models and fast offline recognition in the scenarios mentioned above.

At present, because the feature extraction capability of lightweight mobile network [26], [43] is weaker than the large-scale model, there is no research on a mobile network specifically for fine-grained recognition. However, if we can improve the utilization of the features based on the limited feature extraction capability, the performance of the model will be enhanced. Intuitively, the shallow layer of an end-to-end CNN will extract fine-grained local features of the image. As the network depth increases, the receptive field also expands. CNN will gradually extract the global features and use them for classification. During the process, much of the local information at the shallow layer that is helpful for classification is not taken into account at the end. So, if we can fully combine all of the local and deep global features that are extracted from different stages of CNN, the features of different stages can jointly boost the classification accuracy. As a result, the feature extraction ability of the model can be fully utilized to improve the performance of the lightweight model.

Inspired by the idea above, and based on the lightweight MobilenetV2, we propose a Progressive Multi-Stage Interactive training method for lightweight mobile networks using a Recursive Mosaic Generator (RMG-PMSI). Firstly, we introduce a Recursive Mosaic Generator (RMG) that generates images containing different granularities at different stages. Then, the features of different stages go through a Multi-Stage Interactive module (MSI) to reinforce and supplement the corresponding features of different stages. Finally, using the progressive training (P), the model focuses on learning stable local fine-grained features in the shallower layer and more abstract and large-grained global features in the deeper layer in the next phase. In addition, during the training process of
each phase, different stages produce an interactive supplement instead of being disassociated, as it ensures the consistency in the whole network training process. By using RMG-PMSI as the training mode, the features extracted from the model at different stages can interact and be complementary with each other, which significantly improves the model performance.

The proposed method has been extensively evaluated on three prestigious fine-grained visual classification benchmarks (CUB-200-2011 [30], Stanford Cars [19], Aircraft [24]). An overview of the performance comparison can be seen in Fig. 1. Compared to the baselines, the RMG-PMSI method provides a significant performance improvement. At the same time, we conducted two experiments testing robustness. The results show that the RMG-PMSI has good transferability to models of different scales and can equip them with significantly more anti-interference power. In summary, the main contributions of this paper are as follows:

- A new data augmentation method for FGVC named Recursive Mosaic Generator (RMG) is introduced that can help models focus on features of different granularities at different training phases.
- A novel Progressive Multi-Stage Interactive training method (PMSI) for lightweight mobile networks is proposed for Fine-grained Visual Classification (FGVC), which can make better utilization of features and strengthen the anti-interference ability substantially.
- The proposed RMG-PMSI approach has been implemented and evaluated on standard benchmarks, with performance being significantly improved, and moreover with good robustness and transferability, leading to the potential application on mobile devices.

II. RELATED WORK

In this section, we mainly discuss methods related to FGVC, data argumentation, multi-stage feature fusion and progressive training.

A. Fine-grained visual classification

As visual models continue to evolve, the research on FGVC has shifted from strongly supervised methods with additional bounding boxes [2], [16], [34], [40] to weakly supervised ones with category labels only [10], [11], [20], [32], [36], [41], [42]. Most of the weakly supervised methods aim at locating the most discriminative regions in the image. For example, Fu et al. [10] found that region detection and fine-grained feature learning could reinforce each other and constructed a series of networks during prediction to locate more differentiated regions for the following networks. Lin et al. [23] presented bilinear pooling on the representation of two local patches in the image to learn more representative features. However, these approaches used a large-scale framework to extract high-level abstraction features at the end of an end-to-end network without utilizing shallow local fine-grained features. Due to the mobile network’s limited feature extraction capability, the full utilization of features extracted at different stages remains to be explored.

B. Data argumentation for FGVC

In the field of computer vision, there are many popular data augmentation methods. For example, for conventional image recognition, data augmentation based on fusion of different images is widely used [38], [39]. Based on these tricks, a trainable image fusion augmentation is adopted to achieve great performance improvement in end-to-end training for FGVC [15]. In a weak supervision network, Jigsaw Puzzle is often used as an initialization method to achieve better conversion performance [31]. In a recent study, PMG [8] encouraged the generation of different granular inputs through the Jigsaw Puzzle at different stages. However, none of these methods explicitly generated fine-grained local features and large-grained global features in a single image, which are very substantial for FGVC. In this paper, we address the deficit, and propose a Recursive Mosaic Generator (RMG) accordingly by combining the mosaic augmentation and recursion.

C. Multi-stage feature fusion and progressive training

Multi-stage feature fusion is often applied in the field of object detection [21], [22], [25]. By combining information from different stages, we can better distinguish the background and objects. In FGVC, API [44] can well solve the problem of image confusion due to the nuances between fine-grained image classes by fusing two similar fine-grained image features to find distinguishing clues.

Progressive training is widely used in generation tasks [1], [18], [28], which starts with low resolution images and gradually improves the resolution by adding new layers to the network. Rather than learning features from all scales, the strategy allows the network to discover the large-scale structures of the image distribution and shift its attention to finer-scale details. In FGVC, [8] adopted a progressive training method for image classification, which guided the training of the next stage by the training of the previous stage, gradually shifting the focus from local features to global features.

In this paper, by combining multi-stage feature fusion and progressive training, we propose a progressive multi-stage interactive training approach, which can make features of different stages jointly cooperate in the training and ensure the conformity of training goals. Together with the proposed RMG trick, the feature extraction capability of the model is fully utilized, and the model performance can thus be significantly improved.

III. THE PROPOSED APPROACH

In this section, we will present our proposed approach of RMG-PMSI, integrating Progressive Multi-Stage Interactive training (PMSI) and Recursive Mosaic Generator (RMG). The main idea is to enable interactive learning of features at different stages in the training phase, which can make better use of features of different granularities extracted from the network to recognize fine-grained images and improve the robustness of the model.
A. Overview

PMSI consists of two key components: (i) Recursive Mosaic Generator (RMG) for generating multi-granularities inputs; (ii) Progressive Multi-Stage Interaction (PMSI), as shown in Fig. 2.

To be more specific, firstly, a batch of fine-grained images is fed into the RMG to generate inputs of different phases. The granularities of the input images in different phases are different so that each phase focuses on learning features from fine-grained granularities to larger-grained ones. After the input goes through the backbone, the features generated by multiple stages are extracted respectively. Then, the input of different stages generates interaction vectors \( x_m \), and through a gate mechanism, the corresponding features of different stages are strengthened and supplemented. Finally, in each training phase, the network focuses on training the information of the corresponding granularity extracted by one of the stages. Through the above process, the model can be guided to learn stable fine-grained information in the shallow layer. With the progress of training, the model focuses on learning more abstract and larger-grained global information in the deeper layer in the next phase. In addition, in the training process of each phase, different stages are supplemented interactively rather than being separated and disassociated, which ensures the consistency of the goal in the whole network training phase, filtering out the disturbance of other stage features caused by the independent training of different stages.

B. Recursive Mosaic Generator

Mosaic data augmentation \( [3] \) is an effective mean of boosting performance in object detection. Meanwhile, Jigsaw Puzzle solving \( [8], [31] \) was found to be effective in self-supervised learning and progressive training tasks. We get inspiration from the above two techniques and propose accordingly a Recursive Mosaic Generator (RMG) for a picture to adapt to our progressive multi-stage interactive training. The goal of the RMG is to design regions of different granularities and force the network to learn information specific to the corresponding level of granularity in different training phases. The recursive idea is that the generator will randomly select one of the patches based on the mosaic image generated in the previous step and perform the mosaic operation again. Then, the image contains a large granularity of global information that is difficult to learn, and furthermore, the stable and fine-grained local information that is easy to learn.

Expressly, assume that the input image \( p \in W \times H \times C \) is randomly divided into \( 2 \times 2 \) patches in the first recursion, and the size of each patch is \( \frac{1}{2} W \times \frac{1}{2} H \times C \). In the second recursion, we will randomly divide one of the four patches generated in the first recursion into \( 2 \times 2 \) patches, while the remaining three patches are unchanged. Similarly, one of the patches generated based on the previous recursion will be randomly selected to enter the following recursion in each recursion. Finally, we combine all the generated patches from the recursion into a new graph \( G \left( p, r \right) \). Here, the number of recursions - the granularity contained in the image - is controlled by the hyperparameters \( r \).

For the selection of \( r \) in each stage, it is necessary to meet
use the MLP as the mapping function. Since \( x_m \in \mathbb{R}^c \) is adaptively summarized from features of different stages, it often contains information of different granularities.

After learning the mutual vector \( x_m \), we proceed to expand it to \( x_{N-\text{StageNum}+1}, \cdots, x_{N-1}, x_N \). The main reason is that, by interacting the mutual vector with the feature vectors of different stages, the features generated from each stage can be supplemented with different granularities from other stages, so as to make better advantage of features of different levels. In particular, we perform channel-wise product between \( x_m \) and \( x_n \). Then, we add a sigmoid function \( \sigma \) to generate supplemental gate vectors \( g_n \in \mathbb{R}^c \) corresponding to different stages as follows:

\[
g_n = \sigma \left( x_m \odot x_n \right).
\]

where \( n \in \{ N - \text{StageNum} + 1, \cdots, N - 1, N \} \). Finally, we introduce an interactive supplement mechanism for the features extracted from each stage via residual connections as follows:

\[
m_n = x_n + g_n \odot x_n,
\]

where \( n \in \{ N - \text{StageNum} + 1, \cdots, N - 1, N \} \), and \( m_n \in \mathbb{R}^c \) is the output of multi-stage interactive module.

2) Progressive training strategy: The proposed RMG-PMSI model adopts the progressive training strategy to effectively capture discriminative information of different granularities. It could be implemented on any popular CNN feature extractor, such as the MobilenetV2 [26] and ResNet [13].

The model gets the feature vector \( m_n \) after multi-stage interactions. Thereafter, a classification module \( F_{\text{classifier}}^n \) consisting of two fully connected layers with batch normalization [17] and the activation function ELU [6] corresponding to the \( n \)-th stage, predicts the probability distribution over the classes as

\[
\hat{y}_n = F_{\text{classifier}}^n (m_n).
\]

Here, we consider the last \( \text{StageNum} \) stages with \( n \in \{ N - \text{StageNum} + 1, \cdots, N - 1, N \} \). Finally, we concatenate the outputs from the last \( \text{StageNum} \) stages as

\[
m_{\text{concat}} = \text{concat} \left[ m_{N-\text{StageNum}+1}, \cdots, m_{N-1}, m_N \right].
\]

This is followed by an additional classification module

\[
\hat{y}_{\text{concat}} = F_{\text{classifier}}^{\text{concat}} (m_{\text{concat}}).
\]

During training, each iteration contains \( \text{StageNum} + 1 \) phases. For training, we compute the cross entropy loss \( L_{CE} \) between the ground truth label \( y \) and the predicted output from each phase. At each iteration, a batch of data \( d \) is used for \( \text{StageNum} + 1 \) phases, and we only focus on training one stage’s output \( y_n \) at each phase. It needs to be emphasized that, except for the classifier, all parameters that are used in the current phase prediction will be optimized, even they might have been updated in previous phases, and the strategy helps all stages in the model collaborate together. The detailed implementation of the training algorithm of our RMG-PMSI model is shown in Algorithm 1.
Algorithm 1 RMG-PMSI

Training data set \( D \).
Training data for a batch \( x \),
Training label for a batch \( y \).

\[
\text{for } \text{epoch} \in \{0, \text{epochs}\} \text{ do} \\
\quad \text{for } b \in \{0, D/batch\text{Size}\} \text{ do} \\
\quad \quad x, y \leftarrow \text{batch } b \text{ of } D \\
\quad \quad \text{for } n \in \{N - \text{StageNum} + 1, N\} \text{ do} \\
\quad \quad \quad r \leftarrow N - n + 1 \\
\quad \quad \quad X \leftarrow G(x, r) \\
\quad \quad \quad \hat{y}_n \leftarrow F_{n}^{\text{classifier}} (m_n) \\
\quad \quad \quad \text{Backprop} (L_{\text{CE}} (\hat{y}_n, y)) \\
\quad \quad \text{end for} \\
\quad m_{\text{concat}} \leftarrow \text{concat}\{m_{N-\text{StageNum}+1}, \ldots, m_{N-1}, m_N\} \\
\quad \hat{y} \leftarrow F_{\text{concat}}^{\text{classifier}} (m_{\text{concat}}) \\
\quad \text{Backprop} (L_{\text{CE}} (\hat{y}_{\text{concat}}, y)) \\
\text{end for} \\
\text{end for}
\]

In the testing phase, if we only use \( \hat{y}_{\text{concat}} \) as the prediction, the final result of our module can be expressed as

\[
P_{\text{concat}} = \arg\max (\hat{y}_{\text{concat}}).
\]

Nevertheless, the predictions of each phase are complementary. Hence, we combine all outputs to get the final result which can be calculated as

\[
P_{\text{mix}} = \arg\max \left( \sum_{n=N-\text{StageNum}+1}^{N} \hat{y}_n + \hat{y}_{\text{concat}} \right).
\]

IV. EXPERIMENTS

In this section, we will introduce the datasets and implementation details of the experimental studies. Firstly, a series of ablation studies has been conducted to demonstrate the contributions of each module to the performance of our model. Next, we compare our model’s performance to other state-of-the-art fine-grained classification counterparts. Moreover, we have carried out two experiments to test the robustness of RMG-PMSI. Finally, we visualize and qualitatively analyze the interior of the model.

| Dataset       | #Classes | #Train | #Test |
|---------------|----------|--------|-------|
| CUB-200-2011  | 200      | 5994   | 5794  |
| Stanford Cars | 196      | 8144   | 8041  |
| FGVC Aircraft | 100      | 6667   | 3333  |

A. Dataset and Implementation Details

We have evaluated the performance of the proposed method on three prestigious fine-grained benchmarks: CUB-200-2011 (CUB) \cite{30}, Stanford Cars (Car) \cite{19} and FGVC-Aircraft (Air) \cite{24}, with details shown in Table I. It is worth emphasizing that in all experiments, the category labels of the images are the only annotations used for training.

We run all the experiments on the GTX 2080Ti GPU cluster using PyTorch with a version higher than 1.8.0. The method we proposed was evaluated on the widely used mobile network of MobileNetV2. During the training phase, we adjusted the input image to 512 × 512 and randomly cropped it to 448 × 448 after a random horizontal flip. During the test, we resized the image to 512 × 512 and cropped it from the center to 448 × 448 as input. We used Stochastic Gradient Descent (SGD) to optimize our network and we applied the pre-trained model on ImageNet. For pre-trained convolution layers, the initial learning rate was 0.0001 and reduced by the cosine annealing schedule. For newly added layers, the initial learning rate was 0.001. For all the aforementioned models, we trained them for up to 150 epochs with batch size as 32 and used a weight decay as 0.0005 and a momentum of 0.9. Besides, during training phase, we froze the backbone and only trained the newly-added layers in the first 5 epochs.

B. Ablation studies

The impact of StageNum. To demonstrate the efficacy of progressive interaction training, we conducted experiments without a Recursive Mosaic Generator (RMG) in CUB dataset.

We set the increase in the number of feature map channels as the division between this stage and the next stage. In order to obtain the best performance and prevent excessive noise being introduced by too shallow layer features, we use the last 5 stages as our experimental setup. The StageNum increases from 1 to 5. We use Top-1 Accuracy (Acc) as the evaluation criterion. The results of the experiment are shown in Table II, where Concat is the result of the \( P_{\text{concat}} \) (it is worth noting that when \( S=5 \), Concat is equivalent to the concatenation of the output of stage 5 and its output after MLP) and Mix represents a mixed classification result \( P_{\text{mix}} \). As the results show, when the number of stages (S) involved in interactive training is less than 4, the increase in StageNum improves the model’s performance. Both Concat Accuracy and Mix Acc Accuracy began to decline when the StageNum=4. This might be caused by the low-stage layer focusing on class-independent features. However, the additional supervision forces the low-stage layer (S=1,2) to focus prematurely on the features associated with classification, thus introducing too much low-stage noise to the high-stage classification through progressive interaction, resulting in a decrease in accuracy. In addition, using multi-stages of progressive interaction can lead to increased training costs. To sum up, we use the last three phases (S=5,4,3, StageNum=3) as the optimal choice for progressive interactive training.

The contribution of different components. Specifically, RMG-PMSI can be split into three sub-modules, recursive mosaic generator (R), progressive training (P), and multi-stage interaction (M). To justify their contributions and joint effort, we validate all of their possible combinations in the CUB dataset\(^1\). The results of the experiment are shown in Table III.

\(^1\)It should be noted beforehand that R must work with P.
We can see from the results that both multi-stage interaction (M) and progressive training (P) can significantly improve the model. Combining them further drives accuracy forward. This shows that progressive training (P) and multi-stage interaction (M) can effectively benefit rather than offset each other. Recursive mosaic generator (R) works very well with progressive training (P), helping the model pay more attention to the fine-grained local information in shallow and global features in deep stages. The above experiments demonstrate the power of the components of our RMG-PMSI model and their good collaboration.

C. Comparative studies

Methods for augmenting fine-grained data. We compared our RMG with other data augmentation methods that are widely used in fine-grained classifications on CUB200, including cutmix [38], mixup [39], snapmix [15], and jigsaw [8]. As can be seen in Table IV stitching different pictures does not bring much improvement or even decrease the model performance. We believe that it happens because progressive training and multi-stage interaction encourage the fusion of different features of an image from the local level to the global level. The stitching-based image augmentation greatly disturbs the process of learning features of different stages, which makes it difficult to boost the model performance. On the other hand, the puzzle method jigsaw performs well. Compared to jigsaw, our method can make the image contain both fine-grained local features and large-grained global features, which promote better collaborative learning at different stages and thus achieve optimal performance.

Models for Fine-grained recognition. To further illustrate the advantages of our RMG-PMSI, we compare it with the widely used state-of-the-art fine-grained approaches on MobileNetV2 in CUB dataset. The results of the experiment are shown in Table V. From the table, we can see that because B-CNN, HBP, PC, and API rely more on the abstract features extracted by the model in deeper stages, their performances are limited by the feature extraction capability of the MobileNetV2. For PMG and RMG-PMSI, the performance of the model can be greatly improved. The key to improving performance on a lightweight mobile network is to take full advantage of the features that the model extracts at different stages. Finally, the RMG-PMSI offers better performance improvement than PMG because RMG and multi-stage interaction are better equipped to help different stages work together during the progressive training.

D. Robustness analysis

Stronger backbones. In order to demonstrate the robustness and transferability of our proposed approach on different scales, in addition to MobileNetV2, we applied RMG-PMSI on ResNet-18 and ResNet-34 as well. For RMG-PMSI, we
used Mix Acc (%) as the evaluation metrics. The experimental results are shown in Table VII.

On the three backbones, RMG-PMSI has a significant performance improvement compared to the baseline, which shows that RMG-PMSI is extremely effective on backbones of different scales. At the same time, compared to ResNet-18 and ResNet-34, RMG-PMSI brings more significant scores on MobilenetV2, implying that RMG-PMSI is more suitable for lightweight mobile networks.

**Anti-interference analysis.** After justifying the robustness of the RMG-PMSI on models of different scales. We further test the anti-interference capabilities of RMG-PMSI. We used Color-Jitter to generate interference data on the testset of the three datasets. We set the Jitter coefficient to be 1, i.e., the image’s brightness, contrast, and saturation will be randomly adjusted to 0% to 200% of the original image. In addition, we also generated interference images with Gaussian-Noise (mean=0, variance=0, amplitude=5) on the testset of the three datasets. And then we test these two interference data respectively on the standard MobilenetV2 (baseline) and MobilenetV2 with RMG-PMSI. For RMG-PMSI, we used Mix Acc (%) as the evaluation metrics and the experimental results are shown in Table VII. It can be seen that the accuracy degradation of RMG-PMSI on the two types of interference data on the three testsets is less than the baseline. Especially on CUB200, the anti-interference ability of RMG-PMSI is far better than baseline. This may be because the distinction of birds mainly comes from some important local parts, such as eyes, feathers, and beaks. Adding interference will have a greater impact on global information, and the introduction of local information through RMG-PMSI can effectively alleviate this global interference.

E. Visualization

In order to demonstrate more insights into our approach, we applied GradCAM [27] to visualize the convolutional layers of the last three stages of our method. The visualization results in Fig. 4 show that RMG-PMSI can help the model gradually shift from a fine-grained local feature to a large-grained global feature. For example, in bird pictures(a), the focus is on multiple local features on feathers in the shallow layer (Stage N-2) of the model. In the deeper layer (Stage N), besides focusing on the eye, which is the most distinguishing part, the model also pays attention to feathers, beaks, and other features that play a supporting role in classification. The visualization demonstrates that RMG-PMSI helps the model capture more distinguishing features from both the local and global perspectives, enhancing the performance and anti-interference of the model.

| | CUB | Car | Air |
|---|---|---|---|
| Origin | Baseline | +RMG-PMSI | Baseline | +RMG-PMSI | Baseline | +RMG-PMSI |
| 81.5 | 87.0 | 90.9 | 86.5 | 91.6 |
| + Color-Jitter | 14.1 (-77.4) | 59.9 (-27.1) | 69.7 (-21.2) | 79.3 (-14.7) | 64.7 (-24.2) | 73.9 (-17.7) |
| + Gaussian-Noise | 22.2 (-59.3) | 78.6 (-18.4) | 75.0 (-15.9) | 86.5 (-7.5) | 82.6 (-6.3) | 86.6 (-5.0) |

Fig. 4. Activation map based on the MobilenetV2 with RMG-PMSI. Columns (a)-(c) are CUB, Air, and Car respectively.

V. CONCLUSION

In this paper, we propose a Progressive Multi-Stage Interactive training method together with a Recursive Mosaic Generator (RMG-PMSI) for mobile networks of Fine-grained Visual Classification (FGVC). By progressively capturing local to global features, RMG-PMSI effectively helps the model identify and integrate more distinguishing features and makes better use of the features extracted by mobile networks. Experiments on three prestigious fine-grained benchmarks prove that our method can significantly improve the performance compared to state-of-the-art lightweight models, with good robustness and transferability. For the future work, RMG-PMSI can incorporate more tricks such as attention mechanism.

REFERENCES

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Image super-resolution via progressive cascading residual network. In CVPR Workshops, pages 791–799. Computer Vision Foundation / IEEE Computer Society, 2018.

[2] Thomas Berg and Peter N. Belhumeur. POOF: part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In CVPR, pages 955–962. IEEE Computer Society, 2013.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. CoRR, abs/2004.10934, 2020.

[4] Steve Branson, Grant Van Horn, Serge J. Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. CoRR, abs/1406.2952, 2014.

[5] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In CVPR, pages 5157–5166. Computer Vision Foundation / IEEE Computer Society, 2019.
