Progressive Learning of Category-Consistent Multi-Granularity Features for Fine-Grained Visual Classification

Ruoyi Du, Jiyang Xie, *Student Member, IEEE*, Zhanyu Ma, *Senior Member, IEEE*, Dongliang Chang, Yi-Zhe Song, *Senior Member, IEEE*, and Jun Guo

Abstract—Fine-grained visual classification (FGVC) is much more challenging than traditional classification tasks due to the inherently subtle intra-class object variations. Recent works are mainly part-driven (either explicitly or implicitly), with the assumption that fine-grained information naturally rests within the parts. In this paper, we take a different stance, and show that part operations are not strictly necessary – the key lies with encouraging the network to learn at different granularities and progressively fusing multi-granularity features together. In particular, we propose: (i) a progressive training strategy that effectively fuses features from different granularities, and (ii) a consistent block convolution that encourages the network to learn the category-consistent features at specific granularities. We evaluate on several standard FGVC benchmark datasets, and demonstrate the proposed method consistently outperforms existing alternatives or delivers competitive results. Codes are available at https://github.com/PRIS-CV/PMG-V2.

Index Terms—Fine-grained visual classification, convolutional neural network, progressive training, consistency constraint

1 Introduction

Fine-grained visual classification (FGVC) aims at identifying sub-classes of a given object category, e.g., different species of birds, and models of cars and aircraft. It has attracted much attention due to its diverse applications in intelligent retail, environmental protection, transportation, and other fields [1]. However, unlike traditional classification tasks where holistic features are often sufficient, the subtle differences between fine-grained categories dictates novel designs to overcome the intra-class variations. Most effective solutions to date rely on extracting fine-grained feature representations at local discriminative regions, either by explicitly detecting semantic parts [2], [3], [4], [5], [6] or implicitly via saliency localization [7], [8], [9], [10]. It follows that such locally discriminative features are collectively fused to perform final classification.

Early work mostly finds discriminative regions with the assistance of manual annotations [11], [12], [13], [14], [15]. However, human annotations are difficult to obtain, and can often be error-prone resulting in performance degradations [3]. Research focus has consequently shifted to training models in a weakly-supervised manner given only category labels [3], [4], [7], [9], [16]. Success behind these models can be largely attributed to being able to locate more discriminative local regions for downstream classification. Yet little or no effort has been made to determine (i) at which granularities are these local regions most discriminative, e.g., the finer beak or coarser head of a bird, and (ii) whether the complementary information across different granularities can be cultivated for better fine-grained feature learning, e.g., is there any benefit on synergizing beak and head features.

We first note that information across different granularities often work simultaneously in discriminating fine-grained categories. This is intuitive as experts sometimes need to identify a bird using *both* the overall structure of a bird's head, and the finer details such as the shape of its beak. That is, it is often insufficient to identify standalone discriminative parts, but also how these parts interact with each other in a complementary manner. Very recent research has focused on the "zooming-in" factor [2], [6], i.e., not just identifying parts, but also identifying truly discriminative regions within each part (e.g., the beak, more than the head). Yet these methods mostly focus on a few pre-defined parts and ignore others while zooming in. More importantly, they do not consider how features from different zoomed-in parts can be fused together in a synergistic manner. Different to these approaches, we further argue that, one not only needs to identify parts and their most discriminative granularities, but moreover address how parts at different granularities can be effectively merged.

- *Ruoyi Du, Jiyang Xie, Zhanyu Ma, Dongliang Chang, and Jun Guo are with Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: {beiyoudry, xiejiyang2013, mazhanyu, changdongliang, guojun}@bupt.edu.cn.*
- *Yi-Zhe Song is with SketchX, CVSSP, University of Surrey, GU2 7XH London, U.K. E-mail: y.song@surrey.ac.uk.*

Manuscript received 1 June 2021; revised 4 Nov. 2021; accepted 4 Nov. 2021. Date of publication 9 Nov. 2021; date of current version 3 Nov. 2022.
This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303300 and Subject II under Grant 2019YFF0303302, in part by National Natural Science Foundation of China (NSFC) under Grants 61922015, 61773071, and U19B2036, in part by Beijing Natural Science Foundation Project under Grant Z200002, in part by a scholarship from China Scholarship Council (CSC) under Grant 202006470036, and in part by BUPT Excellent PhD Students Foundation under Grant CX2020105.
(Corresponding author: Zhanyu Ma.)
Recommended for acceptance by O. Russakovsky.
Digital Object Identifier no. 10.1109/TPAMI.2021.3126668

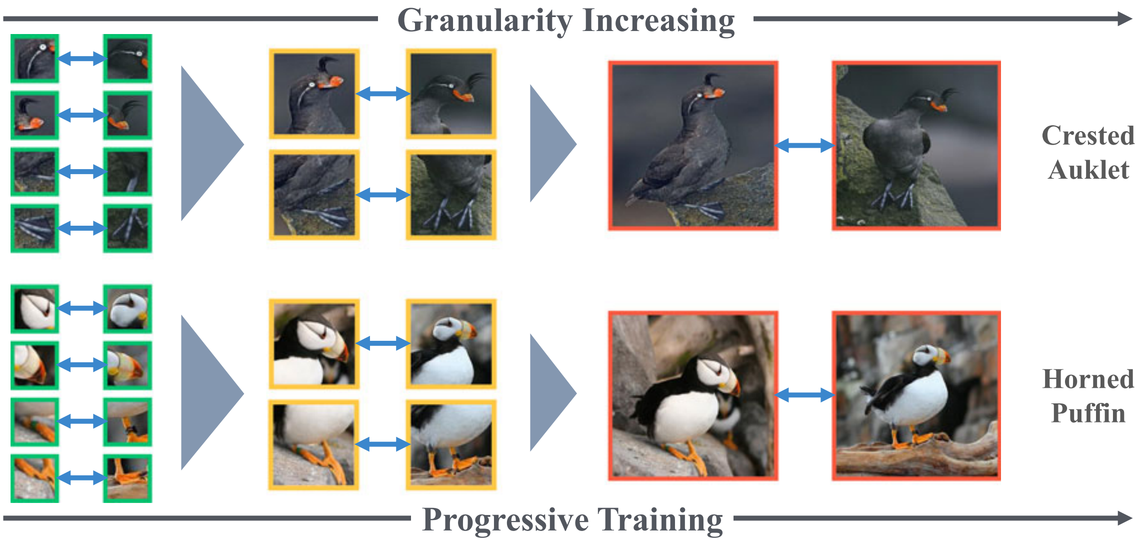


Fig. 1. An example on CUB-200-2011 dataset [17] of multi-granularity information in FGVC. For birds (in the right) to belong to a certain species, large intra-class variations can be observed (e.g., different postures, backgrounds, and shooting angles). As the granularities of semantic parts decrease from right to left, the patterns of their structures become more stable and the intra-class variations of the patterns significantly decrease.

We uniquely work with two key insights, as illustrated in Fig. 1: (i) discriminative patterns often exhibit less visual variance at lower granularity levels, e.g., the bird beaks show smaller intra-class variances than the bird heads, and (ii) coarser granularity features can be progressively reinforced by the learning of finer ones, e.g., knowing the beaks helps with differentiating the heads (i.e., “zooming-out” rather than “zooming-in”). Based on these insights, we take an alternate stance towards fine-grained classification in this paper. We do not explicitly, nor implicitly attempt to mine fine-grained feature representations from parts (or their zoomed-in versions). Instead, we approach the problem with the hypothesis that the fine-grained discriminative information lies *naturally* within different visual granularities – it is all about encouraging the network to learn at different granularities and simultaneously synergizing multi-granularity features.

More specifically, we propose a consolidated framework that accommodates both part granularity learning and cross-granularity feature fusion simultaneously. This is achieved through two components that work synergistically with each other: (i) a progressive training strategy that effectively fuses the features from different granularities, and (ii) a category-consistent block convolution (CCBC) that encourages the network to learn category-consistent features at specific granularities. Note that we refrain from using “scale” since we do not apply Gaussian blur filters on image patches, rather we evenly divide the feature map to form different granularity levels.

As the first contribution, we propose a multi-granularity progressive training framework to learn the complementary information across different granularities. Our progressive framework works in steps during training, where at each step the training focuses on cultivating granularity-specific information. We start with finer granularities which are more stable, and gradually move onto coarser ones. This is akin to a “zooming out” operation, where the network can first focus on a small region, then zoom out to a larger patch surrounding this local region, and finish with the whole image. More specifically, at the end of each training stage, parameters learned at the current stage will be passed onto the next training stage as the parameter initialization. This passing operation essentially enables the network to leverage finer granularity features in learning coarser granularity ones. In the end, prediction results made at all stages are combined to further ensure complementary information are fully-utilized.

However, applying progressive training naively cannot guarantee multi-granularity feature learning. This is because fine-grained information learned via progressive training tends to focus on similar regions. In our early work [18], we tackled this problem by introducing a jigsaw puzzle generator to form different granularity levels at each training step. It essentially forces each stage of the network to focus on local patches rather than holistically across the entire image, thus learning information specific to a given granularity level. Although this was shown to be effective in encouraging the model to focus on fine-grained visual details, further examination reveals that the jigsaw strategy also introduces artificial boundaries that do not agree with the natural image statistics, which is otherwise counter-productive for feature learning – the model will need to simultaneously attend to such “information rich” boundary regions (see Section 4.5.4).

In this paper, we extend our previous approach [18] by addressing these problems with a novel category-consistent block convolution (CCBC) module that consists of a block convolution operation coupled with a category-consistency constraint. Fig. 2 illustrates how these two parts effectively work together. For the block convolution element, the feature map is split into several blocks before being sent to each convolution layer at the training phase. Each of the blocks cannot access information at their adjacent blocks through the convolution operation. This operation keeps the benefit of training with jigsaw patches without introducing artificial boundaries. In order to further capture meaningful regions at each granularity while conducting convolution within feature blocks, a category-consistency constraint is applied to accompany the block convolution operation. In detail, we sample images

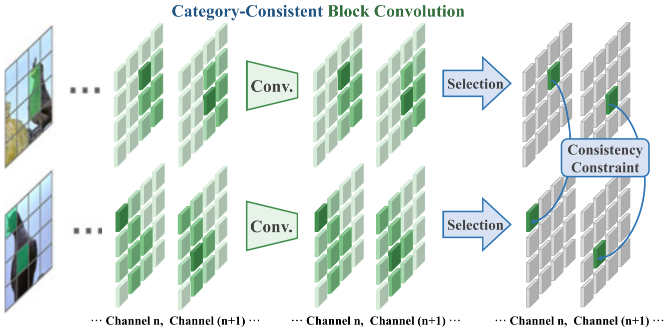

Category-Consistent Block Convolution
Conv.
Selection
Consistency Constraint
Conv.
Selection
··· Channel n, Channel (n+1) ···
··· Channel n, Channel (n+1) ···
··· Channel n, Channel (n+1) ···

Fig. 2. Illustration of the proposed category-consistent block convolution (CCBC) operation. Here, we split a feature map into 4×4 blocks as an example. For the proposed block convolution, the feature map is first split into blocks with the same size. We only conduct the convolution process inside each block, which means all the elements of the output can only receive information from the block that it belongs to. Then, given an image pair belonging to the same category, we select the most salient blocks from each of their channels and correspondingly constrain them to be consistent, which encourage the network to focus on category-relevant regions. Note that the whole process happens at every channel, and we only show the case with two channels for brief illustration.

from each category in pairs, and then constrain their salient blocks to be category-consistent during the block convolution process, which effectively encourage the network to focus on common regions shared within the category (i.e., the foreground). Note that, although reducing intra-class variance by pairwise constraint seems to be an intuitive and general solution, it cannot independently work with our previous framework [18] due to the existence of artificial boundaries (see Section 4.4.2).

To summarize, our contributions are three-fold:

1) We introduce a progressive training strategy that works in a “zooming out” manner for fine-grained visual classification tasks. It operates in different training steps to facilitate the feature learning process and ultimately cultivate the inherent complementary properties across different granularities.
2) We propose a category-consistent block convolution (CCBC) that couples a block convolution operation with a feature category-consistency constraint. On the one hand, the block convolution guarantees that finer features are learned at each step by controlling the feature granularity. On the other hand, the category-consistency constraint prevents the over-fitting problem and makes sure the captured multi-granularity regions are meaningful and category-relevant.
3) We conduct experiments on four widely-used FGVC datasets: the CUB-200-2011 [17], NA-Birds [19], Stanford Cars [20], and FGVC-Aircraft [21] datasets. The proposed method achieves state-of-the-art (SOTA) or competitive performance on all of them. Extensive ablation studies and visualizations demonstrate the effectiveness of our design.

2 Related Work

2.1 Fine-Grained Visual Classification

Significant progress has been made in recent years on convolutional neural network (CNN) architectures. With deeper architectures and broader receptive fields, CNNs tend to work better at understanding complex semantics and processing high-resolution images. However, due to the significant gap between FGVC and traditional coarser classification tasks, the benefits of CNNs for fine-grained analysis remain limited.

In search for fine-grained representations, many previous approaches have focused on better and more efficient localization of discriminative object parts. Zhang *et al.* [14] adopted object detection (i.e., R-CNN [22]) to locate local regions for feature extraction. Krause *et al.* [23] argued the limitation of part annotations and applied a co-segmentation approach to crop the object parts in a weakly-supervised manner. Zheng *et al.* [3] further pointed out that detection and fine-grained feature learning can reinforce each other. Instead of simply locating discriminative parts, several methods [2], [6] tried to mine complementary information from parts of multi-granularity. In addition to cropping the located parts, Zheng *et al.* [24] utilized an attention-based non-uniform sampler to highlight the parts of interest. Ding *et al.* [25] also argued the limitations of discarding surrounding context, and applied an inhomogeneous transform to selected informative regions. A bank of filters was carefully designed by Wang *et al.* [7] as part detectors that implicitly capture discriminative patterns. Except for utilizing region features independently, Wang *et al.* [26] aggregated discriminative regions into groups and further mined discriminative potentials from region correlations. Recently, knowledge from other modalities also has been considered. Song *et al.* [27] not only captured multiple discriminative parts stage-by-stage, but also leveraged the knowledge from text description for interactive alignment.

Despite the great strides made, previous localization-based frameworks mostly follow the pipeline where parts are first detected, and later fused in an ad-hoc manner. In this work, instead of explicitly or implicitly locating the object parts, we approach the problem with the hypothesis that the fine-grained discriminative information lies naturally within different visual granularities – it is all about encouraging the network to learn at different granularities and simultaneously fusing multi-granularity features together.

An earlier and preliminary version of this work appeared in [18] where the model was encouraged to mine multi-granularity information from multi-granularity jigsaw patches in a progressive training scheme. However, naively training with jigsaw patches leads to some limitations: (i) jigsaw patches will introduce artificial boundaries during training, and (ii) jigsaw patches complicate the feature learning and the model may fail to capture meaningful regions. Compared with [18], this work differs in that (i) we put forward an alternative solution named category-consistent block convolution (CCBC) for feature granularity controlling without introducing artificial boundaries. Meanwhile, with the category-consistent constraint, it prevents meaningless regions being focused. (ii) Extensive experiments and analysis are added to better demonstrate the effectiveness and generalization ability of the proposed method.

2.2 Progressive Training Scheme

Progressive training was first introduced in [28] for the task of image generation. It was motivated by previous generative adversarial networks (GANs) with multiple generators or discriminators [29], [30]. It tries to reduce training difficulty of the GANs by starting with low resolution images and then progressively increasing image resolutions by adding new layers to the network. Instead of learning to generate the whole image, it enables the network to first learn the global structure and subsequently supply those difficult finer details. In [31], Shaham *et al.* proposed an unconditional single image GAN scheme that learned from the multi-scale distributions of a single image. Recently, Wang *et al.* [32] adopted the progressive idea for the task of super-resolution reconstruction, where the model was progressive in both architecture and training scheme, and can work well with large up-sampling factors. The progressive training scheme was also introduced by Ahn *et al.* in [33] for progressively adding cascading residual blocks in the super-resolution reconstruction task to generate images with higher resolution.

All aforementioned works however only focused on image generation tasks. In this paper, we utilise the general concept of progressive training on the problem of FGVC. The most salient differences are (i) instead of working with low and high resolutions, we perform progression across patches of different granularities, and (ii) we start with finer details (smaller patches) and progressively move onto more global ones (larger patches), rather than going from coarse to fine [29], [30].

3 Methodology

3.1 Overview of The Proposed Method

Thanks to the continuous progress of the CNN architectures (e.g., [34], [35], [36]), training very deep networks becomes feasible, which greatly strengthens CNNs' representation abilities and enlarges their receptive fields for traditional basic-level image classification tasks. It enables the CNNs to process higher-resolution images and capture the patterns in a global view. However, for FGVC tasks, directly focusing on the global structure of images will introduce instabilities caused by large intra-class variations, which increases the training difficulties.

There has been a common view that the key to solving the issues in FGVC tasks is mining discriminative information from local regions [9]. In this work, instead of recognizing the objects in a "zooming-in" manner (e.g., locating the discriminative parts first and then extracting local features from them), we propose a novel "zooming-out" framework under two insights (as illustrated in Fig. 1): (i) the discriminative patterns tend to be more stable when their granularities decrease, and (ii) learning of small-granularity patterns can facilitate the learning of large-granularity patterns.

Specifically, our method can be roughly divided into two parts. (i) First, we tackle the learning of fine-grained information via a progressive training procedure. In general, the shallower stages of a CNN represent local details and the deeper ones mainly indicate more abstract semantics with larger granularities. Hence, as an intuitive choice, we first let the network mine finer discriminative features at shallow stages and then progressively shift attention to the patterns with larger granularities based on the pre-learned knowledge (obtained in the shallower stages). In this case, the drawbacks caused by intra-class variances when we directly learn the whole objects can be effectively prevented. (ii) Second, in order to facilitate the multi-granularity learning during the progressive training process, we propose a category-consistent block convolution (CCBC) module that consists of two highly-coupled components: the block convolution operation and the category-consistency constraint. At each training step, the feature maps are split spatially into several blocks and the convolution layers only operate within each feature block. This encourages the network to focus on a specific granularity by limiting the receptive field of the layer smaller than its theoretical size. Then, the blocks with the highest responses at each channel are selected and aggregated to form a discriminative feature embedding, which represents the discriminative information being mined at a specific granularity. The distance of a feature embedding pair that belongs to the same category is forced to be smaller to make sure they are category-relevant.

In summary, the progressive training strategy enables a local-to-global learning procedure, and the CCBC guarantees that category-consistent multi-granularity features are learned at each training step. The whole framework is illustrated in Fig. 3 and the design details of each component are elaborated in Sections 3.2 and 3.3.1.

3.2 Progressive Training

3.2.1 Network Architecture

Our network design for progressive training is generic and can be implemented on top of any CNN architecture. A CNN's structure can be divided into several stages, and each of them consists of a group of cascaded convolution layers (e.g., ResNet [35] can be divided into five stages: from $Conv(1)$ to $Conv(5)$). For a feature extractor that contains S stages, $X^{Conv(s)} \in R^{T\times W\times H\times C}, s=1,\ldots,S$ (T, W, H, and C are batch size, width, height, and channel number of the feature map, respectively) represents the feature map output by the s^{th} stage. Here, our objective is to impose classification loss on the feature maps extracted at different intermediate stages. Hence, additional convolution blocks $H_{conv}^{Conv(s)}$ and fully connected layers $H_{fc}^{Conv(s)}$ are incorporated to process the features extracted from each stage to work together as a classifier.

Specifically, the classifier consists of two convolution layers and two fully connected layers with BatchNorm [37]. The M-class final output $V^{Conv(s)} \in R^{T\times M}$ of $Conv(s)$ can be expressed as

$$V^{Conv(s)} = H_{fc}^{Conv(s)}\left(H_{conv}^{Conv(s)}\left(X^{Conv(s)}\right)\right). \tag{1}$$

3.2.2 Training Procedure

During the training phase, one iteration contains several steps to optimize the outputs from each stage sequentially. Specifically, the network is trained from the low-level outputs to the high-level ones with decreasing n. Since the receptive field and the representation ability of low-level stages are limited, the network will be forced to first exploit the discriminative information from local details (i.e., object textures). In this way, step-wise incremental training naturally allows the model to mine discriminative information beginning with local details and progressing to more global structures when the features are gradually sent into higher stages.

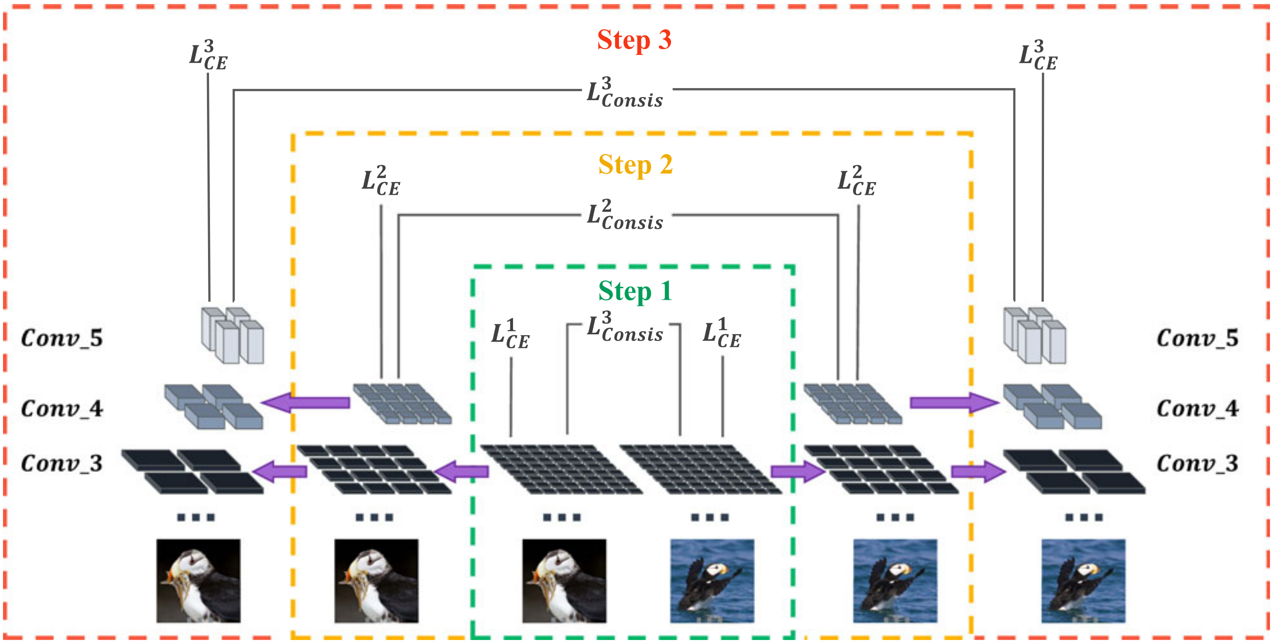


Fig. 3. Illustration of our training pipeline. With the training process going on, the trained stage comes deeper and the sizes of feature blocks become larger in different steps. Here, we apply a progressive training strategy on the last three stages as an example, which means the number of training steps $P = 3$. The hyper-parameter n for each step is set as 8, 4, or 2, respectively. Feature maps with the same colors indicate sharing the weights of the corresponding convolution kernels in the stages, and the purple arrow shows the knowledge transfer from a smaller-granularity level to a larger-granularity level between two steps. Additional convolution layers and fully connected layers including classifiers are omitted in the figure.

Note that according to the experimental results in Section 4.4, instead of utilizing all S stages for the progressive training, taking the last P stages can lead to better performance. This is because low-level stages mainly represent class-irrelevant patterns that will be harmed by the aforementioned supervisions for FGVC. Equal to the number of stages that take part in progressive training, there are P steps at one iteration corresponding to P outputs as $[V^{Conv(S-P+1)}, \dots, V^{Conv(s)}, \dots, V^{Conv(S)}]$. Each output is optimized by cross-entropy (CE) loss as

$$\begin{aligned}\mathcal{L}^p_{\text{CE}} &= \mathcal{L}_{\text{CE}}(V^{Conv(S-P+p)}, Y) \\ &= -\frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M} Y_{t,m}\log\left(V^{Conv(S-P+p)}_{t,m}\right),\end{aligned} \tag{2}$$

where each row of Y ($T \times M$-dimension) is the one-hot label of the corresponding sample in this batch of inputs.

3.3 Consistent Block Convolution

3.3.1 Block Convolution

With the progressive training strategy, we hope the network can mine multi-granularity information from finer- to coarser-level according to the network receptive fields changing. However, the effect is limited due to the wide receptive fields of the advanced deep convolution neural networks (e.g., for ResNet50, $Conv(4)$ and $Conv(5)$ with theoretical receptive fields of 291×291, and 483×483 are both able to cover the main parts of a target object when input image size is set as 448×448.) In order to tackle this drawback for FGVC, we propose a block convolution operation that encourages the network to focus on the local features and simultaneously keep the great representation abilities of very deep architectures. The block convolution operation can be implemented on any widely-used backbone network without changing network architectures or introducing additional parameters.

Let $X^l \in \mathbb{R}^{T\times W\times H\times C}$, $l \in [1, L]$ be the output of the l^{th} convolution layer, where L is the total number of convolution layers. Then a general convolution operation (with paddings) can be expressed as

$$X^{l+1} = \mathcal{F}(X^l, \mathcal{W}^{l+1}), \tag{3}$$

where X^{l+1} are the output feature maps, $\mathcal{W}^{l+1}$ are weights of the $l+1^{st}$ convolution layer.

For the proposed block convolution, the input feature maps are first split into an array of $n \times n$ blocks with equal size as

$$\begin{aligned}&\mathcal{B}(X^l) = [X^l_{1,1}, \dots, X^l_{i,j}, \dots, X^l_{n,n}],\\ &X^l_{i,j} \in R^{T\times\frac{W}{n}\times\frac{H}{n}\times C},\end{aligned} \tag{4}$$

where $i, j \in \{1, \dots, n\}$ are the vertical and horizontal indices of feature block $X^l_{i,j}$ and the hyper-parameter n defines the number of feature blocks in each dimension which in turn controls the granularities of the patterns that can be learned at the current step. We apply a convolution inside each feature block with shared parameters $\mathcal{W}^{l+1}$ and then restore them back into $X^{l+1} \in R^{T\times W\times H\times C}$ as

$$X^{l+1}_{i,j} = \mathcal{F}(X^l_{i,j}, \mathcal{W}^{l+1}), \tag{5}$$

$$X^{l+1} = \mathcal{R}(X^{l+1}_{1,1}, \dots, X^{l+1}_{i,j}, \dots, X^{l+1}_{n,n}). \tag{6}$$

Here $\mathcal{R}(\cdot)$is the restoration operation. In this way, we limit the size of information regions that can be captured by the network to be no larger than $\frac{W}{n} \times \frac{H}{n}$ and the network is forced to mine the discriminative patterns from local regions.

Note that although the block convolution guarantees that the granularities of all the kept patterns are smaller than the block size, not all the eligible patterns can be always kept, as they might be split by the block boundaries. This phenomenon will harm the model training when the block boundaries are always constructed in the same positions because it makes some patterns never be learned. Hence, we require to split the feature maps at different positions in each iteration. Fortunately, a routine data augmentation process, named random cropping, can undertake this role by providing the input images with a random shifting.

The proposed block convolution can be easily implemented by splitting $T \times W \times H \times C$-dimensional feature maps into the blocks and concatenate them to feature maps with size of $(n^2T) \times \frac{W}{n} \times \frac{H}{n} \times C$. Then, in practice, the general convolution operation can be applied on them to implement the block-wise convolution strategy. In addition, instead of repetitively splitting the feature maps before each convolution layer, at each stage, we can split them at the beginning and restore them after all convolution operations are conducted. This implementation can be adopted for better efficiency when there are no other additional components that require global information (e.g., non-local network [38]).

3.3.2 Category-Consistency Constraint

The block convolution weakens the influence of intra-class variations by highlighting stable features within each sample. Here, we further apply a pairwise category-consistency constraint to reduce the dissimilarity among samples from the same category. In detail, we modify the data loading strategy to load images in a batch in pairs during training, which means $Y_{2i-1} = Y_{2i}, i \in [1, \frac{T}{2}]$. Then we minimize the euclidean distances between the feature embeddings in the pairs.

The pairwise category-consistency constraint can cooperate well with our proposed framework. At the p^{th} step, given feature maps $F^{Conv(S-P+p)} \in R^{(n_p)^2 \times \frac{W}{n_p} \times \frac{H}{n_p} \times C}$ extracted from the last block convolution layer of $Conv(S-P+p)$, and n^p is the hyper-parameter n set for the p^{th} step. We first apply a block-wise average pooling layer $AvgPool(\cdot)$ to aggregate the features from each block, and then select the maximum response values at each channel (with $Max(\cdot)$) as the feature embedding $E^{Conv(S-P+p)}$ as

$$E^{Conv(S-P+p)} = Max(AvgPool(X^{Conv(S-P+p)})) \in R^C. \tag{7}$$

And then we take the euclidean distance between a pair of the aforementioned embedding that belongs to the same category as the loss function for the category-consistency constraint

$$\mathcal{L}^p_{Consis} = \sum_{i=1}^{\frac{T}{2}} \left| E^{Conv(S-P+p)}_{2i-1} - E^{Conv(S-P+p)}_{2i} \right|^2. \tag{8}$$

The total loss for the p^{th} stage can be expressed as

$$\mathcal{L}^p_{Total} = \mathcal{L}^p_{\mathrm{CE}} + \lambda_p \mathcal{L}^p_{Consis}, \tag{9}$$

where λ_p is a hyper-parameter to balance the weights of the two losses.

TABLE 1
Details of the FGVC Datasets Used in Our Experiments

Dataset	#Categories	#Training	#Test
CUB-200-2011 [17]	200	5,994	5,794
NA-Birds [19]	555	23,929	24,633
Stanford Dogs [39]	120	12,000	8,559
Stanford Cars [20]	196	8,144	8,041
FGVC-Aircraft [21]	100	6,667	3,333

3.4 Inference

At the inference phase, all of the aforementioned components can be removed. The block convolution operations can be simply replaced by the general convolution operations, since the model has already learned to recognize local patterns and there is no need to keep any constraint on the feature maps. In this case, although our framework extends the training time, the inference efficiency will not be affected. To utilize the complementary multi-granularity knowledge from various stages, we combine all the outputs together by taking their average values and obtain the final results V as

$$V = \frac{1}{P} \sum_{p=1}^{P} V^{Conv(S-P+p)}. \tag{10}$$

The analysis on the performance of each output and their different combination can be found in Section 4.4, where we show the performance gained by using the complementary nature of the multi-granularity information.

4 Experimental Results and Discussions

In this section, we conduct experiments on four widely-used FGVC datasets discussed in Section 4.1; detailed statistics about these datasets are listed in Table 1. The implementation details of the proposed method are provided in Section 4.2. Subsequently, the model performance comparisons with other state-of-the-art methods are shown in Section 4.3. In order to illustrate the effectiveness of different components and the design choices in our method, comprehensive ablation studies and discussions are provided in Section 4.4. In addition, visualizations of models under different settings are shown in Section 4.5 for better understanding of how our model works.

4.1 Datasets

We evaluate the proposed method on the CUB-200-2011 [17], NA-Birds [19], Stanford Dogs [39], Stanford Cars [20], and FGVC-Aircraft [21] datasets, which are briefly introduced as follows:

CUB-200-2011 (CUB). one of the most challenging and popular FGVC dataset. The diversity in pose (e.g., flying birds, swimming birds, and birds standing on branches), growth stages (e.g., nestlings and adult birds), and the diverse background environments (e.g., forest, sea, and sky) lead to large intra-class variance of their global structures. In addition, for birds belonging to a given family (e.g., different species of sparrow or of seagull), they are visually indistinguishable even for many experts, which makes it challenging to obtain further improvement based on current advanced techniques.

NA-Birds (NAB). a recently proposed FGVC dataset with high-quality annotation and large-scale images. Compared with the CUB-200-2011 dataset, it contains three times the number of images with category labels and bounding box annotations. They also provide a baseline with AlexNet [51] as the feature extractor and utilize all annotations, which achieves the accuracy of 75.0%. Here, we also evaluate our method on NA-Birds with only category labels during the training and the test phases.

Stanford Dogs (DOG). a large-scale dataset consists of dogs belonging to 120 species. Similar to other FGVC datasets, it is also faced with small inter-class variance and relatively large intra-class variance. For example, these images could contain dogs of different ages, poses, and colors. In addition, as a friendly partner of humans, these images mainly come from photos of daily life, which provides us with a complementary scenario to evaluate our method.

Stanford Cars (CAR). a dataset consisting of 196 classes of cars that are typically different at the level of Make, Model, and Year. The data are collected from both natural images and rendered images, which leads to diverse backgrounds and various proportions of the main objects. Although the shapes and structures of cars are more rigid than birds, this dataset is still challenging due to large intra-class variance caused by appearance (e.g., color) and also represents a complementary scenario.

FGVC-Aircraft (AIR). a dataset aimed at distinguishing different models of airplanes. Different from categorizing animals like birds, classifying aircraft with rigid structures is faced with intra-class variations due to other causes (e.g., variable aircraft paintings of the different airline companies). Hence, it is widely used by many researchers to show the generalization abilities of their methods with any type of FGVC tasks.

4.2 Implementation Details

We performed all experiments using PyTorch [52] with version higher than 1.4 on a cluster of GTX2080Ti GPUs. The total number of stages is five for these networks. For the best performance, we let the last three stages undergo progressive training, which means the number of training steps $P = 3$. For the three training steps, we set $n = 8$, 4, or 2 and $\lambda_p = 0.01$, 0.05, or 0.1, respectively. During the training phase, only the category labels were introduced. The input images were resized to a fixed size and then randomly cropped to 448×448. A horizontal flip was randomly applied for data augmentation when we trained the networks. During the inference phase, the input images were resized and then center-cropped into 448×448. All of the above settings are standard in the literature [3].

We used stochastic gradient descent (SGD) with the momentum optimizer and batch normalization as the regularizer. Meanwhile, similar to [10], [47], we set the optimal learning rate for each dataset. The learning rates of the newly-added convolution and FC layers were initialized as 0.005, and the learning rates of the pre-trained backbones were maintained as $1/10$ of those of the newly added layers. For all of the aforementioned models, we trained them for up to 200 epochs with a batch size of 32 and used weight decay as 0.0005 and momentum as 0.9. For all the aforementioned models, the learning rates were multiplied by 0.1 at the 100th and 150th epochs.

4.3 Comparisons With State-of-the-Art Methods

The proposed method was evaluated on aforementioned datasets with three popular backbone architectures: VGG16 [34], ResNet50, and ResNet101 [35]. The comparison results are shown in Table 2. According to their backbone network, the table is divided into three groups. In addition to the top-1 accuracy, we also list the p-values of one-sample Student's t-tests between our methods and the state-of-the-art methods with the null hypothesis that the means of two populations are equal under the same backbone network. The proposed method has a statistically significant performance gain from the referred techniques as all the corresponding p-values are all smaller than 0.05. Detailed analyses are carried out as follows.

4.3.1 Performance on CUB-200-2011

The comparison results are shown in Table 2. According to the backbone network being used, the table is separated into three groups. Our method outperforms all the state-of-the-art methods with different backbone networks,demonstrating its superiority and robustness. Especially, compared to the popular part-based methods [4], [7], [25], the proposed method achieves an average improvement of 2.2% without explicitly nor implicitly locating discriminative parts. Even compared to M-CNN [42] which leveraged part annotations and PMA [27] which introduced text descriptions, our method still surpasses them while training with only category labels. MetaFGNet [45] argued the limitation of fine-grained data and proposed a meta learning based method to obtain optimal network parameters for specific fine-grained tasks. However, our method outperforms it by 2.3% with ResNet50 as the base model, showing our method's generalization ability with limited training data.

4.3.2 Performance on NA-Birds

As a large-scale dataset, NA-Birds has more than twice the number of categories in CUB-200-2011, which makes it much more challenging. However, as listed in Table 2, our method can still obtain state-of-the-art results on it, demonstrating the robustness of our model across dataset scales and category amounts. While Cross-X [10] also utilizes multi-stage features to leverage the relationship between them, we still outperform it with a margin of 1.4%, which indicates the superiority of learning multi-granularity features in a progressive manner. In addition, the proposed method achieves an improvement of 0.6% when we replace VGG16 by ResNet50 as the base model, and obtains further improvement of 0.8% when equipped with ResNet101. This suggests that the proposed method can benefit more from deeper CNN architectures when the scale of the dataset is significantly large.

4.3.3 Performance on Stanford Dogs

As a dataset that has been in existence for over a decade, Stanford Dogs has largely been overlooked as a common benchmark for FGVC, with only very few papers reporting results

TABLE 2
Comparisons With Other State-of-the-Art Methods on CUB-200-2011 (CUB), NA-Birds (NAB), Stanford Dogs (DOG), Stanford Cars (CAR), and FGVC-Aircraft (AIR) Dataset

Method	**Backbone**	**Additional Anno./Data**	**Accuracy (%) / p-value**				
			CUB	**NAB**	**DOG**	**CAR**	**AIR**
FT-VGG (Baseline) [7]	VGG16	–	$77.8/9.23 \times 10^{-7}$	–	–	$84.9/2.01 \times 10^{-6}$	$84.8/4.25 \times 10^{-6}$
B-CNN (ICCV15) [40]	VGG16	–	$84.1/4.78 \times 10^{-6}$	–	–	$84.1/1.72 \times 10^{-6}$	$91.3/5.29 \times 10^{-5}$
BoT (CVPR16) [41]	VGG16	Anno.	–	$92.5/3.48 \times 10^{-5}$	–	–	$88.4/1.16 \times 10^{-5}$
M-CNN (PR18) [42]	VGG16	Anno.	$85.7/1.05 \times 10^{-5}$	–	–	–	–
MaxEnt (NIPS18) [43]	VGG16	–	$77.0/8.05 \times 10^{-7}$	$72.6/5.06 \times 10^{-6}$	$65.4/1.95 \times 10^{-6}$	$83.9/1.67 \times 10^{-6}$	$78.1/1.40 \times 10^{-6}$
PMG (ECCV20) [18]	VGG16	–	$88.8/2.01 \times 10^{-3}$	–	–	$94.3/4.11 \times 10^{-4}$	$92.7/2.55 \times 10^{-4}$
AENet (TOMM21) [44]	VGG16	–	$86.9/2.55 \times 10^{-5}$	–	–	$92,4/3.02 \times 10^{-5}$	$92.3/1.41 \times 10^{-4}$
Ours	VGG16	–	89.0±0.02	87.0±0.08	88.1 ± 0.08	95.0±0.04	93.9±0.05
FT-ResNet (Baseline) [7]	ResNet50	–	$84.1/3.46 \times 10^{-6}$	–	–	$91.7/1.27 \times 10^{-5}$	$88.5/3.36 \times 10^{-6}$
M-CNN (PR18) [42]	ResNet50	Anno.	$87.3/1.72 \times 10^{-5}$	–	–	–	–
PC (ECCV18) [8]	ResNet50	–	$80.2/1.23 \times 10^{-6}$	$68.2/2.18 \times 10^{-6}$	$83.0/8.53 \times 10^{-6}$	$93.4/4.42 \times 10^{-5}$	$83.4/9.22 \times 10^{-7}$
MetaFGNet (ECCV18) [45]	ResNet50	Data	$87.6/2.21 \times 10^{-5}$	–	–	–	–
NTS-Net (ECCV18) [4]	ResNet50	–	$87.5/2.02 \times 10^{-5}$	–	–	$93.9/7.93 \times 10^{-5}$	$91.4/1.45 \times 10^{-5}$
MaxEnt (NIPS18) [43]	ResNet50	–	$80.4/1.29 \times 10^{-6}$	$69.2/2.42 \times 10^{-6}$	$73.6/1.31 \times 10^{-6}$	$93.9/7.93 \times 10^{-5}$	$83.9/1.01 \times 10^{-6}$
DFL (CVPR18) [7]	ResNet50	–	$87.4/1.86 \times 10^{-5}$	–	–	$93.1/3.33 \times 10^{-5}$	$91.7/1.83 \times 10^{-5}$
TA-FGVC (ACM MM18) [46]	ResNet50	Data	$88.1/3.60 \times 10^{-5}$	–	–	$88.9/4.10 \times 10^{-6}$	–
Cross-X (ICCV19) [10]	ResNet50	–	$87.7/2.41 \times 10^{-5}$	$86.2/4.16 \times 10^{-4}$	$88.9/8.54 \times 10^{-3}$	$94.6/2.87 \times 10^{-4}$	$92.6/4.71 \times 10^{-5}$
S3N (ICCV19) [25]	ResNet50	–	$88.5/5.59 \times 10^{-5}$	–	–	$94.7/3.79 \times 10^{-4}$	$92.8/6.27 \times 10^{-5}$
MGE-CNN (ICCV19) [6]	ResNet50	–	$88.5/5.59 \times 10^{-5}$	–	–	$93.9/7.93 \times 10^{-5}$	–
DCL (CVPR19) [9]	ResNet50	–	$87.8/2.64 \times 10^{-5}$	–	–	$94.5/2.25 \times 10^{-5}$	$93.0/8.77 \times 10^{-5}$
API-Net (AAAI20) [47]	ResNet50	–	$87.7/2.41 \times 10^{-5}$	$86.2/4.16 \times 10^{-4}$	$88.3/5.06 \times 10^{-4}$	$94.8/5.23 \times 10^{-4}$	$93.0/8.77 \times 10^{-5}$
MC-Loss (TIP20) [48]	ResNet50	–	$87.3/1.72 \times 10^{-5}$	–	–	$93.7/6.14 \times 10^{-5}$	$92.6/4.71 \times 10^{-5}$
PMA (TIP20) [27]	ResNet50	Anno.*	$87.7/2,41 \times 10^{-5}$	–	–	$93.1/3.33 \times 10^{-5}$	$90.8/9.71 \times 10^{-6}$
PMG (ECCV20) [18]	ResNet50	–	$89.6/1.29 \times 10^{-3}$	–	–	$\underline{95.1}/2.28 \times 10^{-3}$	$93.4/2.17 \times 10^{-4}$
AENet (TOMM21) [44]	ResNet50	–	$87.6/2.21 \times 10^{-5}$	–	–	$93.6/5.47 \times 10^{-5}$	$93.3/1.66 \times 10^{-6}$
DP-Net (AAAI21) [49]	ResNet50	–	$89.3/3.23 \times 10^{-4}$	–	–	$94.8/5.23 \times 10^{-4}$	$\underline{93.9}/2.70 \times 10^{-5}$
Chang *et al.* (CVPR21) [50]	ResNet50	–	$\underline{89.9}/4.99 \times 10^{-1}$	–	–	$\underline{95.1}/2.28 \times 10^{-3}$	$93.6/4.27 \times 10^{-6}$
Ours	ResNet50	–	$\underline{89.9}$±0.03	$\underline{87.6}$±0.07	89.1 ± 0.04	**95.4**±0.03	**94.1**±0.03
MGE-CNN (ICCV19) [6]	ResNet101	–	$89.4/2.41 \times 10^{-4}$	–	–	$93.6/5.47 \times 10^{-5}$	–
API-Net (AAAI20) [47]	ResNet101	–	$88.6/4.49 \times 10^{-5}$	$86.6/2.64 \times 10^{-5}$	$\underline{90.3}/1.76 \times 10^{-3}$	$94.9/7.66 \times 10^{-4}$	$93.4/2.17 \times 10^{-4}$
AENet (TOMM21) [44]	ResNet101	–	$88.6/4.49 \times 10^{-5}$	–	–	$93.7/6.14 \times 10^{-5}$	$93.8/1.19 \times 10^{-3}$
Ours	ResNet101	–	**90.0**±0.02	**88.4**±0.03	**90.7** ± 0.04	**95.4**±0.03	**94.1**±0.03

The best results are marked in bold, while second best result are marked using underline. With the null-hypothesis that the means of two populations are equal, the p-values of the one-sample Student's t*-test between the accuracy of our method and accuracies of other state-of-the-art method are listed. The significance level was set as 0.05. Extra A./D.: Extra Annotation/Data.* * *[27] only utilizes additional data (text descriptions) on the CUB dataset.*

on it. Because of the lack of prior efforts, it is not a surprise that we already obtain state-of-the-art performances with all three backbone networks. Furthermore, similar to results on NA-Birds, when equipped with deeper CNN architectures the proposed method delivers greater performance gains when compared with the other three commonly used datasets. We attribute this to larger scale datasets dictates networks with better representation learning capabilities. This may suggest that large-scale datasets like NA-Birds and Stanford Dogs should not be neglected in the FGVC literature for that they offer different perspectives on the problem.

4.3.4 Performance on Stanford Cars

When taking VGG16 as the backbone network, the proposed method surprisingly surpasses FT-VGG (i.e., the baseline model) by 10.1%, since the progressive training strategy makes it easier for the gradients to propagate to the shallow layers and plays the role of skip-connections. It also achieves an improvement of 3.4% compared with the fine-tuned ResNet50 which has already worked well. We obtain state-of-the-art performance with all these backbone networks (VGG16, ResNet50, and ResNet101), which further shows that our progressive multi-granularity training strategy can be equipped by any advanced network architecture for performance boosting. While BoT [41] leverages object bounding box annotations, our method still surpass it by 2.5% with the same backbone network. Note that, compared to our previous version [18], the newly proposed method obtains an average improvement of 0.5%, which signifies the benefit brought by removing artificial boundaries and introducing the category-consistency constraint.

4.3.5 Performance on FGVC-Aircraft

The proposed method also obtains better performances with at least a margin of 0.7%, even when comparing with other state-of-the-art methods including [46] and [41], which utilized additional text descriptions or box annotations. Similar to the results on the Stanford Cars dataset, the newly proposed method also significantly outperforms our previous version [18], with improvement of 1.2% and 0.7% by VGG16 and ResNet50, respectively. This is because these rigid objects (e.g., cars and aircrafts) are more vulnerable to the influence of artificial boundaries which are also with rigid shape.

TABLE 3
Ablation Studies

Components		Accuracy (%)					Avg
PT	CCBC	CUB	NAB	DOG	CAR	AIR	Improv.
×	×	87.9	86.1	87.2	93.3	92.4	–
✓	×	88.6	86.9	88.4	94.2	93.1	+0.86%
×	✓	89.1	86.8	88.2	94.5	93.4	+1.02%
✓	✓	**89.9**	**87.6**	**89.1**	**95.4**	**94.1**	+1.84%

The effectiveness of the progressive training (PT) strategy and the category-consistent block convolution (CCBC) are listed and discussed. The best results are marked in bold.

TABLE 4
Comparison of the Model Performance Trained With the Jigsaw Patches (JP) and the Block Convolution (BC)

CC	JP/BC	Accuracy (%)					Avg
		CUB	NAB	DOG	CAR	AIR	Improv.
×	**JP**	89.4	87.3	88.6	94.5	93.4	–
×	**BC**	89.5	87.3	88.8	95.2	93.7	+0.26%
✓	**JP**	89.4	87.3	88.5	94.7	93.5	+0.04%
✓	**BC**	**89.9**	**87.6**	**89.1**	**95.4**	**94.1**	+0.58%

Results with/without the category-consistency (CC) constraint are both listed. The best results are marked in bold

4.4 Ablation Studies and Discussions

In this section, we first perform ablation studies on all five datasets to verify the contributions of each key component (Section 4.4.1), and demonstrate the superiority over our prior work [18] (Section 4.4.2). Then, a series of detailed experiments are conducted in an attempt to spell out the reasons behind our design choices (Please refer to Sections 4.4.4, 4.4.5, 4.4.6, and 4.4.7).

4.4.1 Effectiveness of Each Component

As shown in Table 3, we conducted experiments on five datasets with ResNet50 as the backbone network to verify the effectiveness of each component. When both the progressive training strategy and the category-consistent block convolution are removed, the results are simply obtained by training a model with multi-stage outputs and combining their predictions at inference time. While simply combining multi-stage outputs has achieving a good performance, the progressive training strategy improves model accuracy with an average margin of 0.86%, demonstrating that a simple local-to-global learning procedure can facilitate fine-grained feature learning. Moreover, even the accuracy of our model with only the progressive training strategy (e.g., 88.6% on the CUB-200-2011 dataset) has surpassed most of the popular methods, the CCBC can further boost the model performance and obtain an additional average improvement of 0.98% on all five datasets. This further improvement indicates the limitation of only training in a progressive manner and signifies the effectiveness of highlighting multi-granularity and category-relevant features.

4.4.2 Jigsaw Patches versus Category-Consistent Block Convolution

In this section, we perform ablation studies comparing the jigsaw patches and the block convolution to demonstrate the benefit brought by removing the artificial boundaries. We then further study the coupled relationship between the category-consistency constraint and the newly-proposed block convolution. ResNet50 was used as the backbone network and experimental results are listed in Table 4. Without the pairwise category-consistency constraint, we can boost the model performance to a certain extend with an average improvement of 0.26% by replacing the jigsaw patches with the block convolution. The performance gain demonstrates the model learning can benefit slightly from removing the artificial boundaries caused by jigsaw patches. Note that the improvements on Stanford Cars and FGVC-Aircraft datasets are much more significant than the two bird datasets. We attribute this to the rigid shapes of cars and aircraft being more vulnerable to the effect of artificial boundaries which are also rigid in shape.

Moreover, we introduce the category-consistency constraint during the training step in order to encourage the model to focus on the consistent discriminative features within each category, and the model with the block convolution achieves further improvement. However, with jigsaw patches for feature granularity control, the category-consistency constraint results in limited benefit and performance even drops on the CUB-200-2011 dataset. We argue that this is caused by the artificial boundaries introduced by jigsaw patches are highly consistent, and the consistency constraint will mistakenly force the model to focus on these meaningless boundaries, which ultimately weakens its effectiveness. In contrast, the block convolution not only keeps the ability of controlling the granularities of learned features but also enables the category-consistency constraint component to further boost the model performance. Hence, within the proposed CCBC, the progressive training strategy and the category-consistency constraint are considerably coupled.

4.4.3 Efficiency of Category-Consistent Block Convolution

It is worth noting that we only use CCBC at the training phase, and it does not affect the model's inference efficiency. As introduced in Section 3.3.1, there are two implementations for block convolution: (i) splitting and restoring the feature map at each convolution layer, and (ii) splitting and restoring the feature map once at each network stage. The first implementation requires $(2 \times n \times L)$ times of splitting-and-restoring operation and the second one requires $(2 \times n \times S)$ times of such operations, where L is the number of convolution layers and S is the number of network stages ($L \gg S$). The computational complexity of both implementations, regarding n, are $O(n)$. In this section, we further conduct ablation studies about the computational costs in the realistic training process of these two implementations with different block numbers n, which is shown in Fig. 6. When $n = 1$, block convolution is equivalent to the conventional convolution, and the increase of computational cost along with the increase of hyper-parameter n indicates the time budget brought by block convolution. It can be observed that the additional time costs of two implementations increase linearly, echoing their time complexity of $O(n)$. Moreover, compared with "Imp. 1", the time budget brought by "Imp. 2" is negligible and acceptable.

TABLE 5
Performance With Different Hyper-Parameter P

P	Accuracy (%)
2 ($\mathbf{Conv(4) - Conv(5)}$)	89.4
3 ($\mathbf{Conv(3) - Conv(5)}$)	**89.9**
4 ($\mathbf{Conv(2) - Conv(5)}$)	89.0

The best results are marked in bold.

TABLE 6
Performance With Different Output Combinations With ResNet50 as the Backbone Network

Combination	Accuracy (%)				
	CUB	NAB	DOG	CAR	AIR
$\mathbf{Conv(3)}$	82.1	77.9	77.4	77.3	80.0
$\mathbf{Conv(4)}$	85.6	86.0	87.7	90.6	89.4
$\mathbf{Conv(5)}$	87.5	87.1	88.5	93.5	89.0
$\mathbf{Conv(4) - Conv(5)}$	89.5	87.4	89.0	95.2	94.0
$\mathbf{Conv(3) - Conv(5)}$	**89.9**	**87.6**	**89.1**	**95.4**	**94.1**

The best results are marked in bold.

TABLE 7
Performance With Different n for Each Step

$\mathbf{Conv(3)}$ (n_1)	$\mathbf{Conv(4)}$ (n_2)	$\mathbf{Conv(5)}$ (n_3)	Accuracy (%)
1	1	1	88.9
2	2	2	89.0
4	4	4	89.5
8	8	8	89.1
4	2	1	89.2
8	4	2	**89.9**
16	8	4	88.5

The best results are marked in bold.

TABLE 8
Demonstrations of the Theoretical Receptive Fields (RFs) With the General Convolution for Each Stage and the Corresponding RFs With the Proposed Block Convolution (BC)

Stage	n	Theoretical RF	RF with BC
$\mathbf{Conv(3)}$	8	99×99	56×56
$\mathbf{Conv(4)}$	4	291×291	112×112
$\mathbf{Conv(5)}$	2	483×483	224×224

4.4.4 Number of Stages for Progressive Training

Here, we discuss the number of stages P used for progressive training. Experiments are conducted on the CUB-200-2011 dataset with other settings remaining the same. ResNet50 is used as the backbone network, which means the total number of stages $S = 5$ and $P \in [1, 5]$. Since the first stage of ResNet50 only consists of one convolution layer, and since there is no progressive training when $P = 1$, Table 5 only shows the performances of $P \in [2, 3, 4]$. We can observe that the progressive training strategy continuously boosts model performance when $P = 2$ and 3 as we desired. However, when we set $P = 4$, which means $Conv(2)$ participates the progressive training process, the test accuracy significantly drops by 0.9%. We argue that the main propose for shallow layers in $Conv(2)$ is to recognize some basic patterns (e.g., some geometric shapes) that are class-irrelevant. However, the additional intermediate supervision will force these layers to mine the class-relevant features, which affects the overall model learning. Hence, we adopt $P = 3$ in our final design.

4.4.5 Effectiveness of Output Combinations

To better understand the effectiveness of the proposed method, we show the test accuracy of each stage classifier and their different combinations in Table 6. For all of the datasets, the test accuracy of the single classifier from $Conv(5)$ surpasses the baseline (FT-ResNet50) and some state-of-the-art methods listed in Table 2, which further demonstrates the block convolution with progressive training strategy can improve the fine-grained feature learning. When we take the multi-output ensemble results as the final predictions, the model performance reaches a new level due to the utilization of complementary information from different granularities. Even though the classifier at $Conv(3)$ shows poor performance, it also provides useful information and boosts the test accuracy on CUB-200-2011 and Stanford Cars datasets. For the aircraft in FGVC-Aircraft, their rigid structure and huge size make small granularity parts less effective, and the best performance can be achieved with only $Conv(4)$ and $Conv(5)$.

4.4.6 Effectiveness of n at Each Stage

We discuss the effect of the hyper-parameter n at each stage. When we keep the same n for all the steps, as shown in the top half of Table 7, the test accuracy continuously increases as n increases until $n = 8$. It verifies the hypothesis that finer granularity leads to more stable patterns that can enhance the representation learning. And we also observe that the model performance drops at $n = 8$. One possible reason is that the discriminative abilities of very fine-grained regions are limited, which indicates the sufficiency of utilizing features of multi-granularity.

In the bottom half of Table 7, we set n in an exponentially declining manner, which means that n is different when we train different stages of the network. It enables the network to learn coarser patterns based on finer ones that are already learned. According to the experimental results, the best performance can be obtained when we set $n_i, n_2, n_3 = \{8, 4, 2\}$ for $Conv(3)$, $Conv(4)$, and $Conv(5)$, respectively. Table 8 lists the theoretical receptive fields and the receptive fields with block convolution when $n = \{8, 4, 2\}$ with ResNet50 as the backbone and input resolution of 448×448. It is clear that our block convolution operation constrains the receptive fields to roughly a quarter of their original size for every stage.

4.4.7 Would Block Convolution Destroy Useful Features?

For block convolution without block-overlapping, it can only be guaranteed that all features at the current step are smaller than a specific granularity, but it cannot guarantee that all features smaller than the specific granularity will be retained. Thus we try to discuss its negative effect and possible solution in this section. We conducted ablation studies with two possible solutions: (i) stochastic splitting that applies a random

TABLE 9
Ablation Studies of Block Convolution, Stochastic Splitting, and Image Random Cropping

Block Conv.	Stochastic Split.	Random Crop.	Accuracy (%)
×	×	×	87.4
✓	×	×	86.3
✓	✓	×	88.1
✓	×	✓	**89.9**
✓	✓	✓	89.7

The best results are marked in bold.

jitter to the position of the split line (Ensuring that the block size between adjacent steps is still gradually increased. The horizontal and vertical jitters are within intervals of $[0, \frac{w}{4n}]$ and $[0, \frac{h}{4n}]$, respectively), and (ii) random cropping that widely used for data augmentation. The experiments were implemented on CUB-200-2011 dataset with ResNet50 as the base model. As shown in Table 9, without either the stochastic splitting or the random cropping, block convolution even leads to performance degradation, which verifies our concern. And when we separately applied the stochastic splitting and the random cropping alone with block convolution, we obtained performance boosting of 0.7% and 1.5%, respectively. Both these two techniques make the position on which the image is split be different at each iteration and prevent accuracy degradation. In addition, when we applied the stochastic splitting and the random cropping simultaneously, it does not yield further improvement. this is most likely due to stochastic splitting making block sizes in two adjacent steps to be similar, rather than a stable four-fold increase, which consequently weakens the effect of granularity control for progressive training

4.4.8 Would the Learned Multi-Granularity Features Work for Object Across Various Scales?

The various object scale relative to the full image is one of main cause of large intra-class variance of FGVC. Thus, some may wonder whether the proposed method learning with fixed block sizes can still work with various object scales. For birds in CUB-200-2011, their scales are mainly decided by their categories, e.g., pelicans are generally larger than sparrows in terms of object scale. Therefore, to further evaluate how does our model work with different scales of objects across a dataset, in this section, we discuss the performance gain for each category on CUB-200-2011. We use ResNet50 as the backbone network, and FT-ResNet50 is the baseline model. As shown in Fig. 9, the 200 categories of birds in CUB-200-2011 are organized in ascending-order according to their average relative scales and then evenly divided into 20 groups. On average, the birds only account for about 25.3% pixels in the whole image in the first group, while accounting 47.6% in the last one. Our method consistently boosts all the 20 groups and does not show any significant biases along with object scales.

4.5 Visualizations

4.5.1 Activation Maps of All the Datasets

As shown in Fig. 4, we generated the activation map of our method on five datasets using the Grad-CAM [53] algorithm. We used ResNet50 [35] as the feature extractor and its last

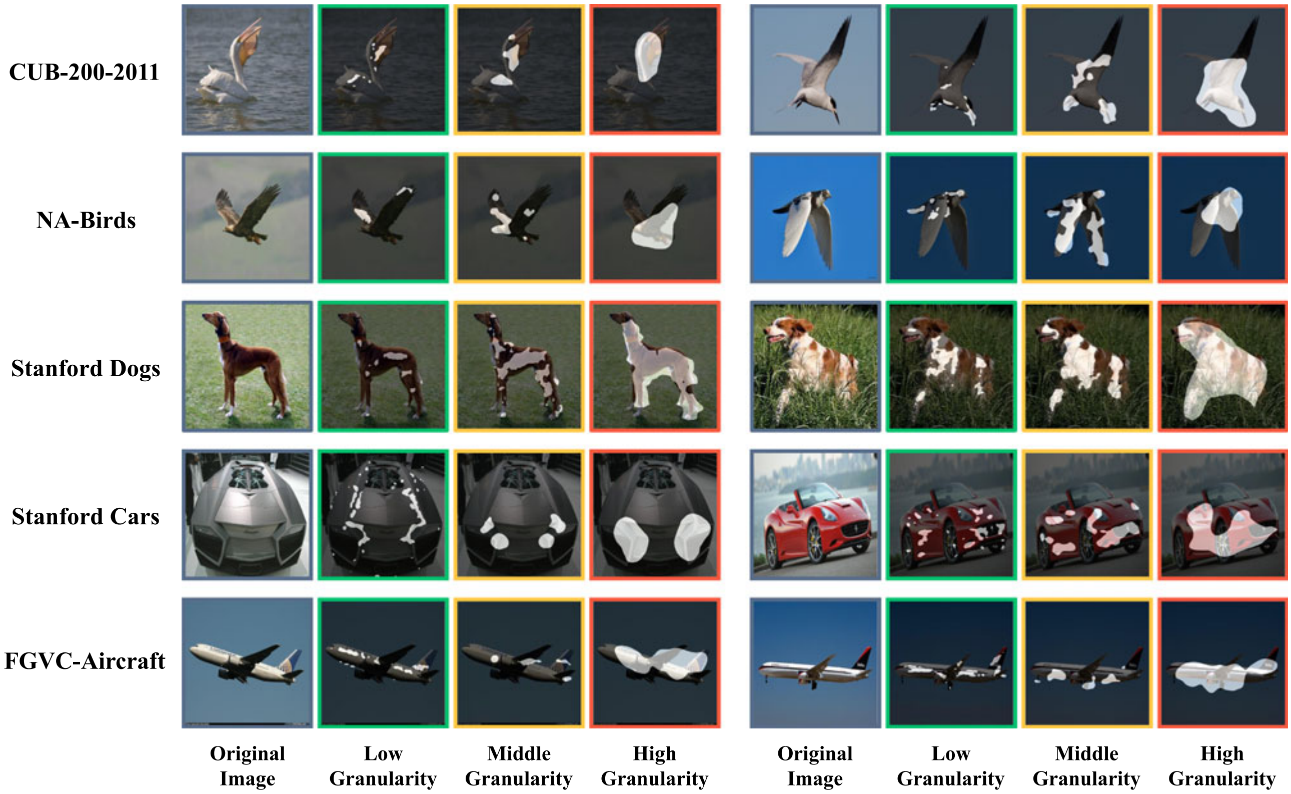


Fig. 4. Activation maps of the images from the CUB-200-2011, NA-Birds, Stanford Dogs, Stanford Cars, and FGVC-Aircraft datasets. The visualization results are obtained via the Grad-CAM algorithm [53] with ResNet50 [35] as the backbone network. “Low-Granularity”, “Middle-Granularity”, and “High-Granularity” means activation maps of model attentions at $Conv(3)$, $Conv(4)$, and $Conv(5)$, respectively. The figure is best viewed digitally.

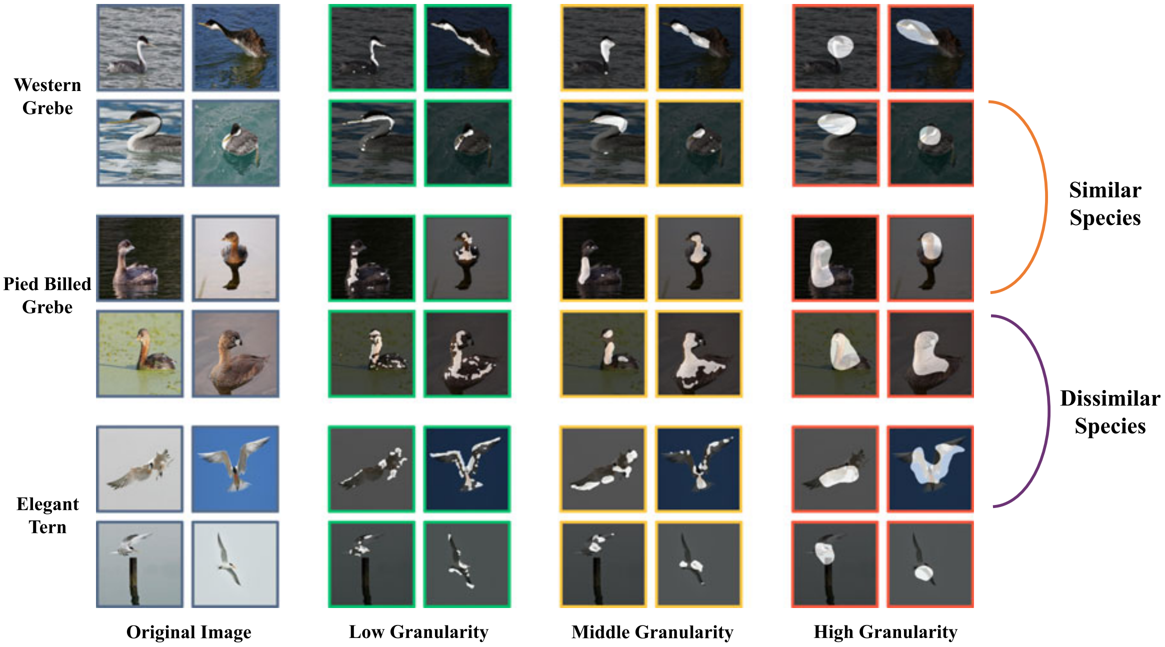


Fig. 5. Activation maps for images from three bird species in CUB-200-2011 for further comparisons. The visualization results are obtained via the Grad-CAM algorithm and ResNet50 is used as the backbone network. It is clear that our model shows consistency on different samples within the same category at each granularity level. For the two similar species we selected, the Grebes, the model mainly focuses on their necks which is the most significant characteristic to distinguish them. As for "Elegant Tern", which is not similar to the other two species, the model tends to concentrate on their feathers. "Low-Granularity", "Middle-Granularity", and "High-Granularity" means activation maps of model attentions at $Conv(3)$, $Conv(4)$, and $Conv(5)$, respectively. The figure is best viewed digitally.

three stages of progressive training. Hence, here we visualize the concentration of feature maps from $Conv(3)$, $Conv(4)$, and $Conv(5)$ which represent low-level granularity, middle-level granularity, and high-level granularity discriminative parts, respectively. The bright regions indicate where our model focuses on.

It can be observed that, with the supervision of category labels at each stage, all stages show attentions on target objects and ignore the background noise. As the network becomes deeper, the network will first focus on some local texture information and then progressively shift attention to more abstract object parts. For the birds in CUB-200-2011 and NA-birds, the network mainly pays attention to their unique feathers and beaks at the low-level stage and focus on their body parts at deeper layers. For the aircraft in the FGVC-Aircraft dataset, the network tends to concentrate on their windows and wheels at shallow layers, which is reasonable due to some aircraft of the same model but different years of manufacturing can only be distinguished by counting the number of their windows. For most of the images, their activation maps from different stages follow a basic rule that they focus on the same positions with different granularities. This indicates the achievement of learning multi-granularity discriminative parts progressively.

4.5.2 Feature Category-Consistency

In Fig. 5, we show visualization results of different samples from two similar species, "Western Grebe" and "Pied Billed Grebe", in the CUB-200-2011 dataset. It can be observed that the network attention shows significant consistency among all the samples within the same category at each granularity level. For the low-granularity parts, the model mainly focuses on some feather textures of bird necks and bird chests. For the middle-granularity parts, the model tends to concentrate on some body parts like bird heads or chests. And for the high-granularity parts, the model consistently focuses on the whole upper parts of birds. Even with different bird postures, light conditions, and shooting angles, the proposed model shows great attention consistency against these intra-class variations.

In addition, when humans distinguish these two categories, the neck of the "Western Grebe", which contains a significant boundary of black and white, is considered a reliable identification mark. Hence, the model attention shows great effectiveness by concentrating on bird necks at each granularity level.

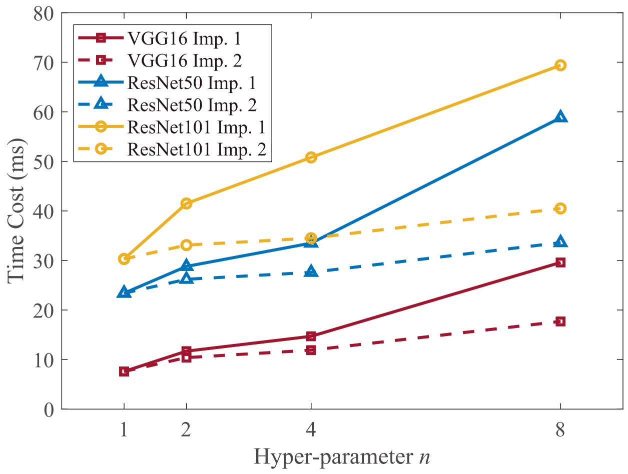


Fig. 6. Illustration of the time cost brought by block convolution increasing with the hyper-parameter n. "Imp. 1" and "Imp.2" represent two implementations.

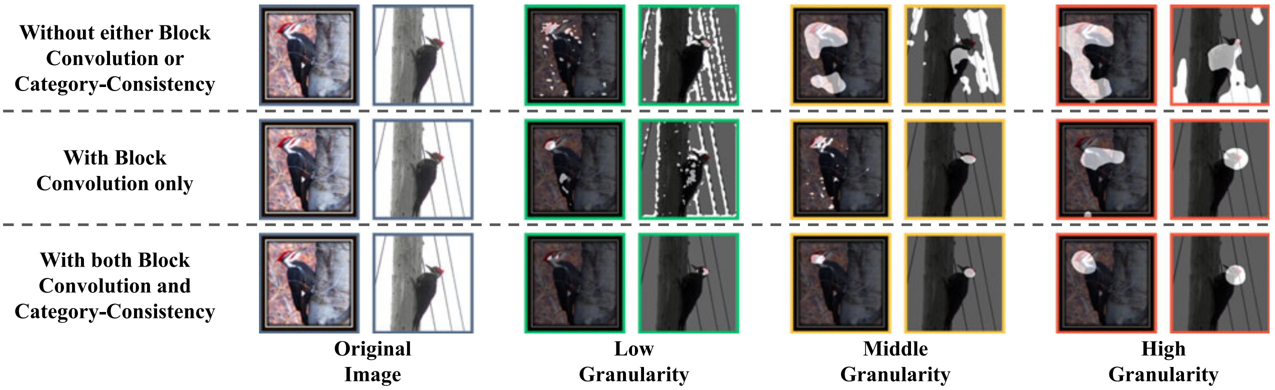

Without either Block Convolution or Category-Consistency
With Block Convolution only
With both Block Convolution and Category-Consistency
Original Image
Low Granularity
Middle Granularity
High Granularity

Fig. 7. Ablation study via visualization of model attention map. The visualization results are obtained by the Grad-CAM algorithm and ResNet50 is used as the backbone network. Here we show the activation maps of a pair of images from the same bird species "Pileated Woodpecker" with/without the block convolution and the category-consistency component. When the category-consistency constraint is removed, the model fails to focus on a consistent part. When the block convolution operation is removed, the model fails to focus on multi-granularity local parts. "Low-Granularity", "Middle-Granularity", and "High-Granularity" means activation maps of model attentions at $Conv(3)$, $Conv(4)$, and $Conv(5)$, respectively. The figure is best viewed digitally.

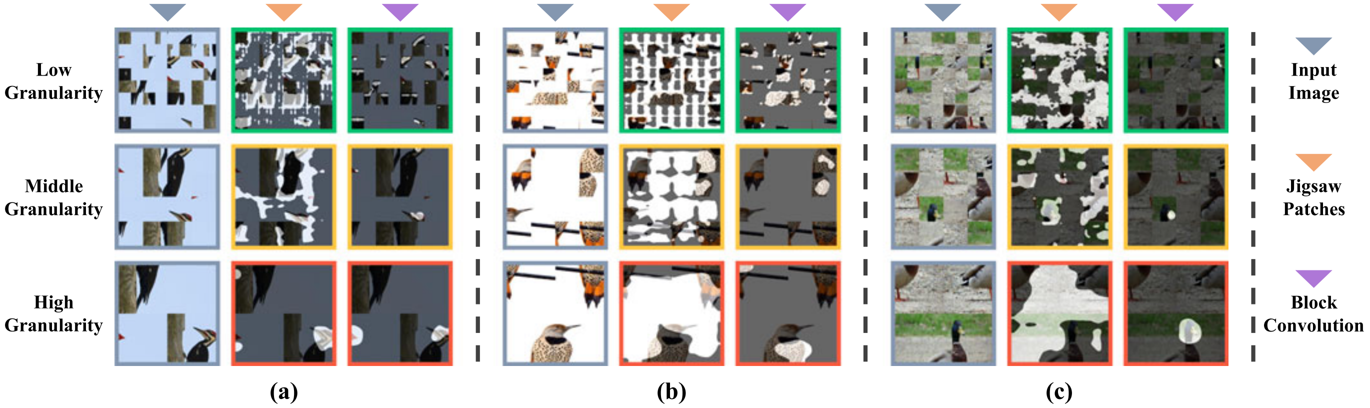


Fig. 8. Selected attention maps from the CUB-200-2011 dataset with different part-granularity control techniques. In order to show the superiority of the proposed block convolution, images that are shuffled into multi-granularity jigsaw patches are used for inputs during visualization. For each sub-figure, the first column lists input images, the second column shows the attention maps of a model trained with jigsaw patches, and the third column shows the attention maps of a model trained with block convolution. The attention maps of different granularities are obtained from their corresponding network stages. "Low-Granularity", "Middle-Granularity", and "High-Granularity" means activation maps of model attentions at $Conv(3)$, $Conv(4)$, and $Conv(5)$, respectively. The figure is best viewed digitally.

4.5.3 Ablation Study of CCBC

In order to further demonstrate the effectiveness of each part of the proposed CCBC, (i.e., the block convolution operation and the pairwise category-consistency constraint), we conducted visualization on three different models that were trained with/without these two components. A pair of images belong to "Pileated Woodpecker" were selected from the CUB-200-2011 dataset, and the visualization results are shown in Fig. 7. With both of these components, the model clearly focuses on consistent discriminative local parts among these two images. When we remove the category-consistency constraint, the model fails to locate the same high-granularity part even it though still correctly focuses on the target object. And at the low-granularity level, the model is disturbed by some background noises. Here we deduce that the category-consistency constraint can force the network to mine common patterns shared within the category, which facilitates the model robustness against the meaningless regions that do not consistently appear. When we replace the block convolution layers with normal convolution layers, the model does not focus on the local parts anymore and fails to mine multi-granularity features (e.g., the activated regions of the middle-granularity level and the high-granularity level show similar granularities.).

4.5.4 Comparison of Jigsaw Patches and The Block Convolution

To better illustrate the superiority of the proposed block convolution, we visualize the attention map of models trained with the jigsaw patches and the block convolution. While generating the attention map for each stage, the input images are first shuffled into $n \times n$ jigsaw patches, where n is the hyper-parameter set for the respective training stage. In this way, we can investigate the reactions of the two models to the artificial boundaries introduced by jigsaw patches. The images shown are sampled from CUB-200-2011 dataset, and ResNet50 is used as the backbone network.

As shown in Fig. 8, the negative effect of jigsaw patches is clearly illustrated in these attention maps where the artificial boundaries invoke dense responses. In the second column of each sub-figure, the attention maps of the model trained with jigsaw patches show significant patterns of the mesh shapes at the low-level granularity, and the attention maps at the middle-level granularity also show messy distributions. In

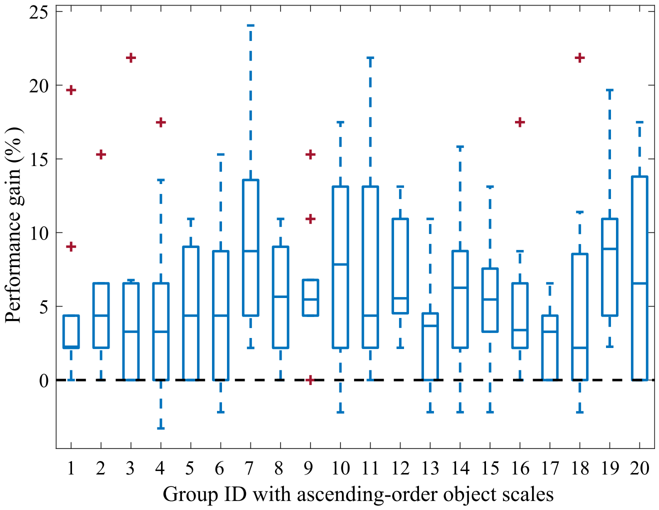


Fig. 9. Illustration of category accuracy difference between our model and a naive FT-ResNet50 on CUB-200-2011. The 200 categories of birds in CUB-200-2011 are organized in ascending-order according to their average relative scale and then are evenly divided into 20 groups.

contrast, the model trained with the block convolution can correctly locate discriminative parts within each jigsaw patch at all granularity levels, even it has never seen shuffled images during training. The visualization results indicate the superiority of the proposed block convolution, especially when it works with the pairwise category-consistency constraint.

5 Conclusion

In this paper, we approached the problem of fine-grained visual classification from a rather unconventional perspective – we do not explicitly nor implicitly mine for object parts; instead we show fine-grained features can be extracted by learning across granularities and effectively fusing multi-granularity features. Our method can be trained without additional manual annotations other than category labels, and only needs one network with one feed-forward pass during testing. We conducted experiments on three widely used fine-grained datasets, and obtained state-of-the-art performance on two of them while being competitive on the other.

References

[1] X.-S. Wei, J. Wu, and Q. Cui, "Deep learning for fine-grained image analysis: A survey," 2019, *arXiv: 1907.03069*.

[2] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4438–4446.

[3] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5209–5217.

[4] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, "Learning to navigate for fine-grained classification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 420–435.

[5] W. Ge, X. Lin, and Y. Yu, "Weakly supervised complementary parts models for fine-grained image classification from the bottom up," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3029–3038.

[6] L. Zhang, S. Huang, W. Liu, and D. Tao, "Learning a mixture of granularity-specific experts for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8331–8340.

[7] Y. Wang, V. Morariu, and L. Davis, "Learning a discriminative filter bank within a CNN for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4148–4157.

[8] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell, and N. Naik, "Pairwise confusion for fine-grained visual classification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 71–88.

[9] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5152–5161.

[10] W. Luo *et al.*, "Cross-x learning for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8242–8251.

[11] T. Berg and P. Belhumeur, "Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 955–962.

[12] J. Lei, J. Duan, F. Wu, N. Ling, and C. Hou, "Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3d-HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 706–718, Mar. 2016.

[13] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1641–1648.

[14] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[15] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked CNN for fine-grained visual categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1173–1182.

[16] Z. Ma *et al.*, "Fine-grained vehicle classification with channel max pooling modified cnns," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3224–3233, Apr. 2019.

[17] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200–2011 dataset," California Inst. Technol., Tech. Rep. CNS-TR-2011-001, 2011.

[18] R. Du *et al.*, "Fine-grained visual classification via progressive multi-granularity training of jigsaw patches," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 153–168.

[19] G. VanHorn *et al.*, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 595–604.

[20] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 554–561.

[21] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[23] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5546–5555.

[24] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5012–5021.

[25] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, and J. Jiao, "Selective sparse sampling for fine-grained image recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6599–6608.

[26] Z. Wang, S. Wang, H. Li, Z. Dou, and J. Li, "Graph-propagation based correlation learning for weakly supervised fine-grained image classification," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12 289–12 296.

[27] K. Song, X.-S. Wei, X. Shu, R.-J. Song, and J. Lu, "Bi-modal progressive mask attention for fine-grained recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 7006–7018, 2020.

[28] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations*, 2018.

[29] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," *Spectrochimica Acta Part B*, vol. 107, pp. 1–10, 2015.

[30] A. Ghosh, V. Kulharia, V. Namboodiri, P. Torr, and P. Dokania, "Multi-agent diverse generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8513–8521.

[31] T. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4570–4580.

[32] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 864–873.

[33] N. Ahn, B. Kang, and K.-A. Sohn, "Image super-resolution via progressive cascading residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 791–799.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

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

[36] G. Huang, Z. Liu, L. VanDerMaaten, and K. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[38] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[39] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop Fine-Grained Vis. Categorization*, vol. 2, no. 1, 2011.

[40] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1449–1457.

[41] Y. Wang, J. Choi, V. Morariu, and L. Davis, "Mining discriminative triplets of patches for fine-grained classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1163–1172.

[42] X.-S. Wei, C.-W. Xie, J. Wu, and C. Shen, "Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization," *Pattern Recognit.*, vol. 76, pp. 704–714, 2018.

[43] A. Dubey, O. Gupta, R. Raskar, and N. Naik, "Maximum-entropy fine grained classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 635–645.

[44] Y. Hu, X. Liu, B. Zhang, J. Han, and X. Cao, "Alignment enhancement network for fine-grained visual categorization," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1s, pp. 1–20, 2021.

[45] Y. Zhang, H. Tang, and K. Jia, "Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 233–248.

[46] J. Li, L. Zhu, Z. Huang, K. Lu, and J. Zhao, "I read, I saw, I tell: Texts assisted fine-grained visual classification," in *Proc. ACM Int. Conf. Multimedia*, 2018, pp. 663–671.

[47] P. Zhuang, Y. Wang, and Y. Qiao, "Learning attentive pairwise interaction for fine-grained classification," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13 130–13 137.

[48] D. Chang *et al.*, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 29, pp. 4683–4695, 2020.

[49] S. Wang, H. Li, Z. Wang, and W. Ouyang, "Dynamic position-aware network for fine-grained image recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, 2021, pp. 2791–2799.

[50] D. Chang, K. Pang, Y. Zheng, Z. Ma, Y.-Z. Song, and J. Guo, "Your" flamingo" is my" bird": Fine-grained, or not," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 476–11 485.

[51] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[52] A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.

[53] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

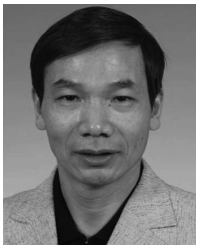
Ruoyi Du received the BE degree in telecommunication with management in 2020 from the Beijing University of Posts and Telecommunications (BUPT), China, where he is currently working toward the PhD degree. His research interests include pattern recognition and computer vision.

Jiyang Xie received the BE degree in information engineering in 2017 from the Beijing University of Posts and Telecommunications (BUPT), China, where he is currently working toward the PhD degree. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in image processing, data mining, and deep learning.

Zhanyu Ma (Senior Member, IEEE) received the PhD degree in electrical engineering from the KTH-Royal Institute of Technology, Sweden, in 2011. Since 2019, he has been a professor with the Beijing University of Posts and Telecommunications, Beijing, China. From 2012 to 2013, he was a postdoctoral research fellow with the School of Electrical Engineering, KTH-Royal Institute of Technology. From 2014 to 2019, he has been an associate professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing, and data mining.

Dongliang Chang received the ME degree in Internet of Things engineering from the Lanzhou University of Technology, China, in 2019. He is currently working toward the PhD degree with the Beijing University of Posts and Telecommunications. His research interests include the intersection of deep learning and computer vision.

Yi-Zhe Song (Senior Member, IEEE) received the bachelor's degree (First Class Honours) from the University of Bath in 2003, the MSc degree (with Best Dissertation Award) from the University of Cambridge in 2004, the PhD degree in computer vision and machine learning from the University of Bath in 2008. He is currently a chair professor of computer vision and machine learning, and director of SketchX Lab, Centre for Vision Speech and Signal Processing (CVSSP), University of Surrey. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, and *Frontiers in Computer Science – Computer Vision*. He is a program chair for British Machine Vision Conference (BMVC) 2021, and is an area chair for flagship computer vision and machine learning conferences, including CVPR'22 and ICCV'21. He is a fellow of the Higher Education Academy and a member of the EPSRC review college.



Jun Guo received the BE and ME degrees from the Beijing University of Posts and Telecommunications (BUPT), China, in 1982 and 1985, respectively, and the PhD degree from the Tohuku Gakuin University, Japan, in 1993. He is currently a professor and a vice president with BUPT. He has authored more than 200 papers in journals and conferences, including *Science*, *Nature Scientific Reports*, *IEEE Transactions on PAMI*, *Pattern Recognition*, AAAI, CVPR, ICCV, and SIGIR. His research interests include pattern recognition theory and application, information retrieval, content-based information security, and bioinformatics.