Multimedia Tools and Applications (2023) 82:14799–14813
https://doi.org/10.1007/s11042-022-13619-z

Learning enhanced features and inferring twice for fine-grained image classification

Xuan Nie[1] · Bosong Chai[1] · Luyao Wang[1] · Qiyu Liao[2] · Min Xu[2]

Received: 24 February 2021 / Revised: 20 May 2022 / Accepted: 21 July 2022 /
Published online: 7 October 2022

Abstract

Fine-Grained Visual Categorization (FGVC) aims to distinguish between extremely similar subordinate-level categories within the same basic-level category. Existing research has proven the great importance of the discriminative features in FGVC but ignored the contributions for correct classification from other features, and the extracted features always contain more information about the obvious regions but less about subtle regions. In this paper, firstly, a novel module named forcing module is proposed to force the network to extract more diverse features for FGVC, which generates a suppression mask based on the class activation maps to suppress the most distinguishable regions, so as to force the network to extract other secondary distinguishable features as the final features. The forcing module consists of the original branch and the forcing branch. The original branch focuses on the primary discriminative regions while the forcing branch focuses on secondary discriminative regions. Secondly, in order to solve the problem that information of small-scale distinguishable features is lost seriously after multi-layer down-sampling, according to the class activation maps of the first prediction, the object is cropped and scaled as the second input. To reduce the prediction error, the first and second prediction probabilities are fused as the final prediction result. Experimental results indicate that the proposed method not only outperforms the baseline model by a large margin (3.7%, 5.9%, 3.1% respectively) on CUB-200-2011, Stanford-Cars, and FGVC-Aircraft, but also achieves state-of-the-art performance on FGVC-Aircraft.

Keywords Fine-grained visual categorization (FGVC) · Image classification · Convolutional neural networks (CNN)

✉ Min Xu
min.xu@uts.edu.au

Xuan Nie
xnie@nwpu.edu.cn

Bosong Chai
chaibosong@mail.nwpu.edu.cn

Luyao Wang
sf_wly_faith@mail.nwpu.edu.cn

[1] Northwestern Polytechnical University, Xi'an, China

[2] University of Technology Sydney Ultimo, Sydney, Australia

1 Introduction

In the past few years, Convolutional Neural Networks (CNNs) have excelled immensely on image classification for its splendid feature extraction capability. However, different from the traditional image classification task, where categories have a huge difference in morphology. Fine-Grained Visual Categorization (FGVC) mainly focuses on distinguishing between subordinate-level categories within the same basic-level category, e.g., different kinds of birds [29], cars [17], dogs [15], aircraft [20]. FGVC is more challenging than traditional image classification for that the intra-class variances could be much higher than the inter-class variances. The CNN model can not correctly distinguish between extremely similar-looking categories unless it can extract subtle and discriminative features.

As shown in recent works [7, 25, 34], paying attention to multiple discriminative parts plays a vital role in FGVC. In the early work [2, 3, 32], extra manual bounding-box or part annotations are employed to extracting discriminative features in multiple object parts. Recent efforts [34, 39] utilize only class labels to automatically localize the object parts. Ding et al. [7], Sun et al. [25] show that without external interference, CNNs [13, 27, 31] usually excel at extracting the most discriminative feature but ignores the crucial complementary information as well. Recently, the study of translation invariance [1, 24, 36] in CNN indicates that small translation or rescaling on the input image can drastically change the prediction of a deep network, it means that a fixed network will focus on different parts and extract different features when the object is panned or zoomed.

In this paper, a novel framework named “forcing network” is proposed, which is referred to as F-Net to address the challenges of FGVC. The diverse and enhanced features will be obtained in F-Net by the forcing module which is consisted of the original branch and the forcing branch. The original branch generates the class activation maps (CAM) to localize the most discriminative parts. In the forcing branch, the suppressive mask is generated to suppress the primary discriminative and force the network to pay attention to secondary discriminative regions which are usually overlooked due to the network pays the most attention to the primary discriminative. After the back gradient propagation, enhanced features will be extracted for classifiers. To reduce the prediction error, the subtle regions are magnified, according to the CAM, the object is cropped and zoomed as the second input to predict again. The first and second prediction probability are fused as the final results. In the training phase, the most discriminative region on the cropped image is dropped to force the network to pay attention to more regions.

Our main contributions can be summarized as follows:

- We proposed a novel “forcing network” structure. The forcing branch is introduced as an auxiliary branch to force the network to focus on multiple regions. And extract diverse features including primary discriminative features and confusion features for fine-grained visual categorization.
- Based on class activation maps, the object is cropped to the center of the image and the subtle regions will be magnified for the second prediction. The sum of the two prediction probability serves as the final prediction.
- Comprehensive experiments were carried on the widely-used fine-grained benchmarks, including CUB-200-2011, FGVC-aircraft, and Stanford-cars. The comparison results demonstrated that our method outperforms the majority of methods and achieves state-of-the-art performance on FGVC-Aircraft.

The rest of the paper is organized as follows. Section 2 contains the literature review. Section 3 contains the methodology (method). Section 4 contains the results. Section 5 contains the conclusions and policy implications.

2 Related work

In this section, we briefly review the related works of fine-grained visual categorization.

For FGVC, the traditional image classification method was used in the earliest stage. The commonly used image feature extraction method is the SIFT method. After the features are lifted by SIFT [21], the features are clustered by K-nearest [22] and other clustering methods. Such methods are computationally complex and time-consuming. There have been a variety of methods proposed for FGVC. In the early work [2, 3, 32, 38], the cumbersome and expensive manual bounding box or part annotations are adopted. Later, part or box annotations were replaced by extracting features of multiple parts or discriminative parts in a weakly supervised way. MA-CNN [39] generated multiple parts by clustering, weighting, and pooling from spatially-correlated channels and then classified an image by each individual part. MA-CNN takes a long time to train and has low accuracy. NTS-Net [34] adopted self-supervision to effectively localize informational regions without the demand of bounding-boxes or part annotations. DCL [5] partitioned the input image into local regions and then shuffles them as another destructed sample. It will pay more attention to discriminative regions to recognize the destructed image. The adversarial learning module is added to the DCL to prevent the network from overfitting to the noisy features caused by random image scrambles. S3N [7] collected peaks from the class response maps to estimate the discriminative and complementary information receptive fields and learn a set of sparse attention for capturing the subtle yet fine-detailed visual evidence as well as preserving content information. DB [25] found subtle differences between similar-looking categories by suppressing the most prominent discriminative regions in class activation maps in the training phase. DB network enables the network to notice multiple regions in the inference stage by randomly suppressing the feature expressions of different regions in the training stage. DB can achieve higher accuracy with fewer parameters.

Bilinear [18] CNN model is another effective stream for FGVC, the output of two CNN branches is multiplied using the outer product at each location of the image and pooled to obtain the bilinear vectors as the features for the classification layer. Following the impressive performance, some improved bilinear models are proposed. TASN [40] proposed trilinear attention sampling to learn subtle feature representations from hundreds of part proposals for FGVC. Gao et al. [9] compacted bilinear pooling with low-dimension and low-rank bilinear pooling [16] by applying a low-rand bilinear classifier was proposed to reduce the consumption in computation time and parameters memory. HBP [35] adapted bilinear pooling between different layers that enabled the inter-layer interaction of features. Xiong et al. [33] proposed an efficient framework for RGB-D scene recognition, which adaptively selects important local features to capture the great spatial variability of scene images. Wang et al. [30] present multiscale representation for scene classification, which is realized by a global–local two-stream architecture.

FGVC is improved by various other methods as well. MAMC [26] leveraged metric learning to learn multiple relevant parts by pulling positive features closer while pushing negative features away. API-Net [41] recognized a pair of fine-grained images by interaction. In MC-loss [4], each class was predicted by a specific number of channels, and each group consists of a discriminative component and a diversity component. GCL [31] proposed a criss-cross Graph propagation sub-network to learn region correlations. MGE-CNN [37] developed several experts to classifier the image, and each expert learns with prior knowledge from the previous expert, in the end, a gating network was used to determine the contribution of each expert. A gradient-boosting loss that seeks to resolve ambiguities among closely related classes is proposed in DB [25] as well.

Our method obtains diverse features that contain the primary discriminative features and confusion features by enhancing the secondary discriminative regions. Compared with random suppression, suppressing primary the discriminative regions in class activation maps that force the network to pay more attention to the confusing regions which are usually overlooked due to the network pays the most attention to the primary discriminative regions. Compare with multiple frameworks, the first and the second prediction in our method share the same framework, and we only increase an extra convolutional layer, based on backbone such as ResNet-50 [13]. Since the object is panned and magnified in the second input, the network will focus on the parts different from the first prediction. In the training phase, we use the average of the first and the second loss as the final loss, it reduces the loss of the oscillation from the first wrong prediction.

3 Methodology

In this section, the F-Net and the CAM-based cropping moudule are described in detail, the overview architectures of the two modules are illustrated in Figs. 1 and 2, respectively. F-Net consists of two components including the feature extracting module and the forcing module. The feature extracting module is the convolutional backbone of Resnet-50 [13]. The forcing module and CAM-based cropping module will be described detailly in this Sections 3.1 and 3.2, respectively. To acquire class activation maps conveniently, the fully connected layer for classification is replaced with a 1 × 1 convolutional layer. The 1 × 1 convolutional layer has an output channel number equaling to the number of classes to acquire class activation maps. Given an input image, the feature maps for classification are produced by the feature extracting module. We denote the extracted feature maps as $F \epsilon R^{N \times W \times H}$, with height H, width W, and the number of channels N.

3.1 Forcing module

The proposed forcing module is inspired by DB [25]. The forcing module aims to force the network to extract more diverse features for the classifier, it consists of the original branch and forcing branch. The original branch and forcing branch share the same feature extract model, but the inputs of the two branches are different. The destination of the original branch is to generate the class activation maps and to localize the primary discriminative regions. After the feature maps F is convoluted by 1 × 1 convolutional layer, class activation maps $M' \epsilon R^{C \times W \times H}$ is obtained, where W, H, and C represent the feature's width, height, and the number of classes, respectively. Then we perform global average pooling, in order to obtain the predicted class activation maps $M'_p \epsilon R^{W \times H}$, where p is the index of maximum in predicted vector $V \epsilon R^C$. Here, C refers to the number of classes.

$$V = g(M'), \tag{1}$$

where $g(\cdot)$ is Global Average pooling. For forcing branch, M'_p is utilized to generate a mask to suppress top-k discriminative positions of F. Since the top-k positions are suppressed, the forcing branch has to pay attention to other confused positions. Here, we describe the procedure to generate the input for the forcing branch in detail. Firstly, M'_p is reshaped to a vector of size W times H, i.e. WH and sorted in descending order, then, the k-th values T

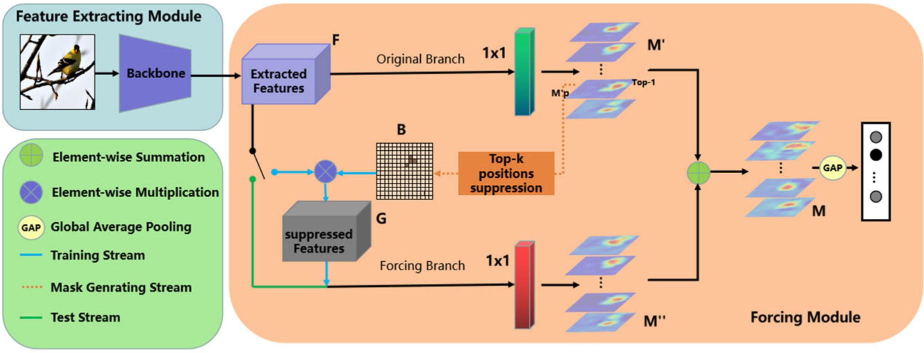

Fig. 1 Overview of F-Net. F-Net consists of the feature extracting module and the forcing module The feature extracting module is convolutional layers that extract features. The forcing module contains the original branch and forcing branch

is obtained as the threshold value.

$$T = Sort(M'_p)[k], \tag{2}$$

where $Sort(\cdot)$ denotes sorting in descending order,$[\cdot]$ represents getting value from vector and k is a hyperparameter that denotes the number of suppressive positions. Let B be the suppressive mask derived from M'_p such that:

$$B(i, j) = \begin{cases} \alpha & M'_p(i, j) >= T \\ 1 & M'_p(i, j) < T \end{cases}, \tag{3}$$

where i and j represent row and column of the feature's position respectively and α is a hyperparameter that denotes suppressing factor. Finally, the input of forcing branch $G \epsilon R^{C \times W \times H}$ is obtained, which is generated as follows:

$$G = B \odot F, \tag{4}$$

where $\odot$ denotes the element-wise multiplication of the two tensors. After the classification convolution is performed, the output of the forcing branch $M'' \epsilon R^{C \times W \times N}$ is obtained. Let M be the output of the forcing module, M is obtained as :

$$M = M' + M'' \tag{5}$$

The confidence scores will be obtained after M is fed to global average pooling.

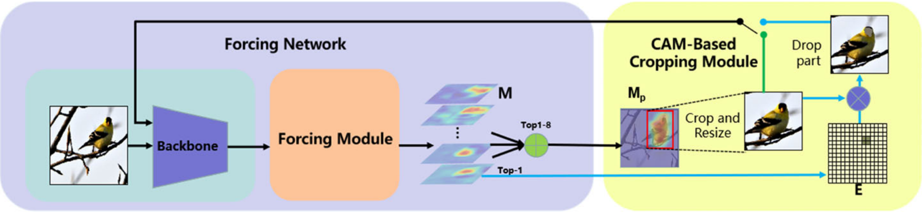

Fig. 2 Overview of CAM-based cropping module. This module crops the object region as the center of the input image for the second prediction. The first and second prediction probabilities are fused as the final result

3.2 CAM-based cropping module

CAM-based cropping module is proposed to crop the object into the image's center and infer again. The first prediction always focuses on the obvious regions while the second prediction may pay attention to some subtle regions which will be amplified after the CAM-based cropping module. The summation of raw prediction and the second prediction is as to the final prediction. Here, we explain the procedure to crop the object. In the forcing module, we have described the generation of class activation maps $M \epsilon R^{C \times W \times H}$. Since the top-1 map usually highly responds on part of the object and the high response regions from other maps are other parts of the object, instead of using the top-1 map to localize the discriminative region, we utilize top-8 maps to crop the whole object.

Denote $M_p \epsilon R^{W \times H}$ as the element-wise summation of top-8 maps. M_p consists of the object and backgrounds. The threshold value t is set to distinguish between the object and backgrounds. t is generated as follows:

$$m = max(M_p), \tag{6}$$

$$t = m \times d, where\ d \sim rand(0.4, 0.6), \tag{7}$$

where m is the maximum of M_p. Because of the diversity of samples, we generate a random number d from the uniform distribution between 0.4 and 0.6 in the training phase. d is set to the minimum value of 0.4 in the random number in the test phase to ensure the whole object will be cropped. Then, crop mask B_2 is obtained as follows:

$$B_2(i, j) = \begin{cases} 1 & M_p(i, j) >= t \\ 0 & M_p(i, j) < t \end{cases}, \tag{8}$$

The response values greater than or equal to t belong to the object, otherwise belong to the backgrounds. We generate a bounding box that can cover all positions of 1 in B_2 and crop the object from the raw image as the second input. In the training phase, the discriminative parts are dropped on the second input. It should be noted that discriminative parts do not drop in the test phase. The drop mask is obtained as follows:

$$m_1 = max(M_1), \tag{9}$$

$$E(i, j) = \begin{cases} 0 & M_1(i, j) >= m_1 \times 0.75 \\ 1 & M_1(i, j) < m_1 \times 0.75 \end{cases}, \tag{10}$$

where M_1 is the top-1map of M and m_1 is the maximum of M_1. As the training progresses, the size of the high response area changes all the time, so the threshold value is set to a fixed value of 0.75 instead of random values. The position of 0 on E will be dropped in the second input.

3.3 Multi-prediction model

The multi-prediction model is to make two predictions for the same image, the first and the second share the same network, but the input is different. The second input of the multi-prediction model is also different in the training phase and the test phase. The multi-prediction model process is shown in Fig. 2. The multi-prediction model training phase process is shown by the blue arrows in Fig. 2. In the training phase, the original image I is input to the network for the first time, and the prediction result $Prop_1$ is obtained. According to the first predicted class activation map, the I target is first clipped and expanded by linear interpolation, the clipping factor d is a random number between 0.4 and 0.6, and then the main feature area is discarded to obtain the second predicted Input image I_2. After going through the same network model, the second prediction result $Prob_2$ is obtained. Both $Prob_1$ and $Prob_2$ are predicted probability values through softmax, and the final prediction result P is calculated as follows:

$$P = (Prob_1 + Prob_1) \div 2, \tag{11}$$

In this classification task, by using the cross-entropy function as the loss function, the cross-entropy loss is calculated as follows:

$$L = -\sum_i y_i \log \hat{y}_i, \tag{12}$$

where y_i is the prediction result, $\hat{y}_i$ is the true label, and i is the category subscript and takes values from 0 to $c - 1$. During multi-prediction model training, the final prediction result P is not used to calculate the loss value, but the first prediction result $Prob_1$ and the second prediction result $Prob_2$ are used to calculate the loss value, The calculation process is as follows:

$$Loss = (-\sum_i Prob_{1i} \log \hat{y}_i + -\sum_i Prob_{2i} \log \hat{y}_i) \div 2, \tag{13}$$

The final loss is the average of the cross-entropy of the first prediction and the cross-entropy of the second prediction. The flow of the testing phase is shown by the green arrows in Fig. 2. During the testing phase, inputting the original image I into the network to get the first prediction result $Prob_1$ firstly. According to the first predicted class activation map, only the I target is clipped and expanded by linear interpolation, the clipping factor d is 0.4, and the main feature loss module is not performed, and the second predicted input image I_2 is obtained. After the same network model, the second prediction result $Prob_2$ is obtained. The final prediction result P is calculated in the same way as in the training phase.

4 Experiments

In this section, we show comprehensive experiments to verify the effectiveness of F-Net. Firstly, three datasets used to verify our method and the implementation details will be described in Sections 4.1 and 4.2. Then we compare our model with other methods among the three common fine-grained visual classification datasets in Section 4.3. Finally, we analyze the contribution of each component in the proposed framework in Section 4.4.

4.1 Datasets

We comprehensively evaluate our method with three challenging fine-grained datasets, including CUB-200-2011 [29], Stanford Cars [17], and FGVC Aircraft [20]. The detailed statistics with category numbers and data splits are shown in Table 1.

Table 1 Three common fine-grained visual classification datasets

Dataset	Class	Train	Test
CUB-200-2011 [29]	200	5994	5794
Standford Cars [17]	196	8144	8041
FGVC Aircraft [20]	100	6667	3333

4.2 Implementation details

In the following experiments, ResNet-50 [13] implemented in Pytorch [23] is adopted as the backbone and the fully connected layer is replaced with a 1 × 1 convolutional layer which has the same output channel as the number of classes. The feature extracting convolutional layers is initialized by pre-trained ResNet-50 weights from ImageNet [6], and the classification layer is initialized by Xavier initialization [11].

In the training phase, the images are resized to 515 × 512 and then randomly cropped to 448 × 448 with random horizontal flipping. The cropped threshold d is randomly selected from 0.4 to 0.6 for every sample and the dropped threshold is set to 0.75, as described in Section 3.2. We train our network using Stochastic Gradient Descent (SGD) with the momentum of 0.9, epoch number of 100, weight decay of 0.0001, and a mini-batch of 6 on GTX-2080ti(11G) GPU. The initial learning rate is set to 0.001 and decayed on the 30th epoch with a decay rate of 0.1. Source code is released at https://github.com/boxyao/Forcing-Network.

4.3 Quantitative results

We do not use any manual annotations except for the class labels. For fair comparisons, our method is compared with methods without human-defined bounding boxes or part annotations. The comparisons with the various recent and top-performing methods on three challenging datasets, including CUB-200-2011, FGVC aircraft, and Stanford-cars. Table 2 illustrates the results of three datasets.

Table 2 Comparison with the state-of-the-art on the CUB-200-2011, Stanford Cars, and FGVC Aircraft benchmarks

Method	Backbone	Resolution	Parameters	Datasets		
				Bird	Aircraft	Cars
ResNet-50 [13]	ResNet-50	448	23.9M	85.4	88.5	91.7
NTS-Net [34]	ResNet-50	448	25.5M	87.5	91.4	93.9
DCL [5]	ResNet-50	448	24.7M	87.8	93.0	94.5
S3N [7]	ResNet-50	448	>101.5M	88.5	92.8	94.7
MGE-CNN [37]	ResNet-50	448	>25.1M	88.5	–	93.9
DB [25]	ResNet-50	448	23.9M	88.6	93.5	94.9
Stacked-LSTM [10]	GoogleNet+ ResNet-50	short-side 800	–	90.4	–	–
API-Net [41]	DenseNet-161	448	30.3M	90.0	93.9	95.3
AttNet&AffNet [12]	ResNet-50	448	23.8M	88.9	94.1	95.6
MC-Loss [4]	B-CNN	448	67.1M	86.4	92.9	94.4
EfficientNet-B3 [28]	EfficientNet-B3	448	22.6M	89.8	93.4	94.4
ConvNext-B [19]	ConvNext-B	448	–	90.7	93.9	94.4
Ours	ResNet-50	448	24.3M	88.4	93.3	94.5
Ours	DenseNet-161	448	27.3M	89.1	94.4	94.8
Ours	EfficientNet-B3	448	23.3M	90.3	94.7	95.2
Ours	ConvNext-B	448	27.8M	90.0	94.2	95.0

On the CUB-200-2011, the baseline based on ResNet-50 achieves 85.4%. our method further outperforms the baseline by 3.0%. A further improvement of another 0.7% can be observed when we use DenseNet-161 [14] as the backbone. Compared with MGE-CNN [37] based on ResNet-50, which used multi-experts, we acquire almost the same accuracy by adding an auxiliary classifier. Both our approach and the DB [25] extract diverse features by feature suppression. Despite DB method outperform our method by 0.2%, our forcing module outperforms DB without Gradient-boosting loss by 1%. And our method is based on the latest backbone EfficientNet and ConvNext to further improve the accuracy. Compared with API-Net based on DenseNet-161, EfficientNet and AttNet&AffNet, our method when we use EfficientNet-B3 [28] as the backbone, has 0.3%, 1.4%, 0.5% improvement, respectively.

On the FGVC-aircraft, the proposed F-Net on ResNet-50 and DenseNet-161 [14] achieves 93.3%, 94.4% respectively. Compared with methods based on ResNet-50, our methods outperform most of the methods except DB and AttNet&AffNet. Our method based on DenseNet-161 achieves state-of-the-art performance, which further outperforms API-Net [41] based on DenseNet-161 by 0.5%.

On the Stanford-cars, our method based on ResNet-50 obtains 94.5%, which is 2.8% better than the baseline 91.7%. A further improvement of another 0.7% can be observed when we use EfficientNet as the backbone. Compared with API-Net, our proposed method based on EfficientNet-B2 is very competitive.

In Fig. 3, We visualize the experimental results of the forced module. the high-response regions in the second column are marked by red boxes, and the high-response regions in the third column are marked by black boxes. In the first row, the highest response area in the original branch is the bird's head, and the most distinguishable area is the bird's head, but in the forcing branch when the bird's head features are suppressed to a certain extent, the forcing branch puts more attention on the bird's tail and claws. In the output of the forced module in the fourth column, the classification basis of the network is not only the head of the most important distinguishable area bird, but also the tail and claws of the second important distinguishable area bird. In the second row, the primordial branch judges that the main distinguishable area of the bird is the bird's head, and in the forcing branch, the bird's feathers are judged to be the secondary distinguishable area because the head is suppressed. In the third row, the original branch judges the bird's wings and tail as the most distinguishable regions, and in the forcing branch, the bird's beak and neck are judged as the secondary distinguishable regions because the wings and tail are suppressed. Finally, in the forcing module, the beak and the neck are judged to be more important distinguishable regions than the tail and wings, which indicates that the forcing branch corrects the misclassified results in a certain possibility.

In Fig. 4, we visualize the results of our method. The results show that the proposed structure is activated to different parts of the raw input and the cropped input. The case that the original prediction is wrong while the cropped prediction is correct indicates that the two-step strategy can reduce the loss of prediction once.

4.4 Ablation study

To sufficiently analyze the contribution of different components in our method. we conduct various experiments respectively on CUB-200-201, Stanford-Cars, and FGVC-Aircraft using ResNet-50. Tables 3, 4 and 5 illustrate the detailed contribution of each key component. It shows both forcing branch and crop inference are effective to improve the

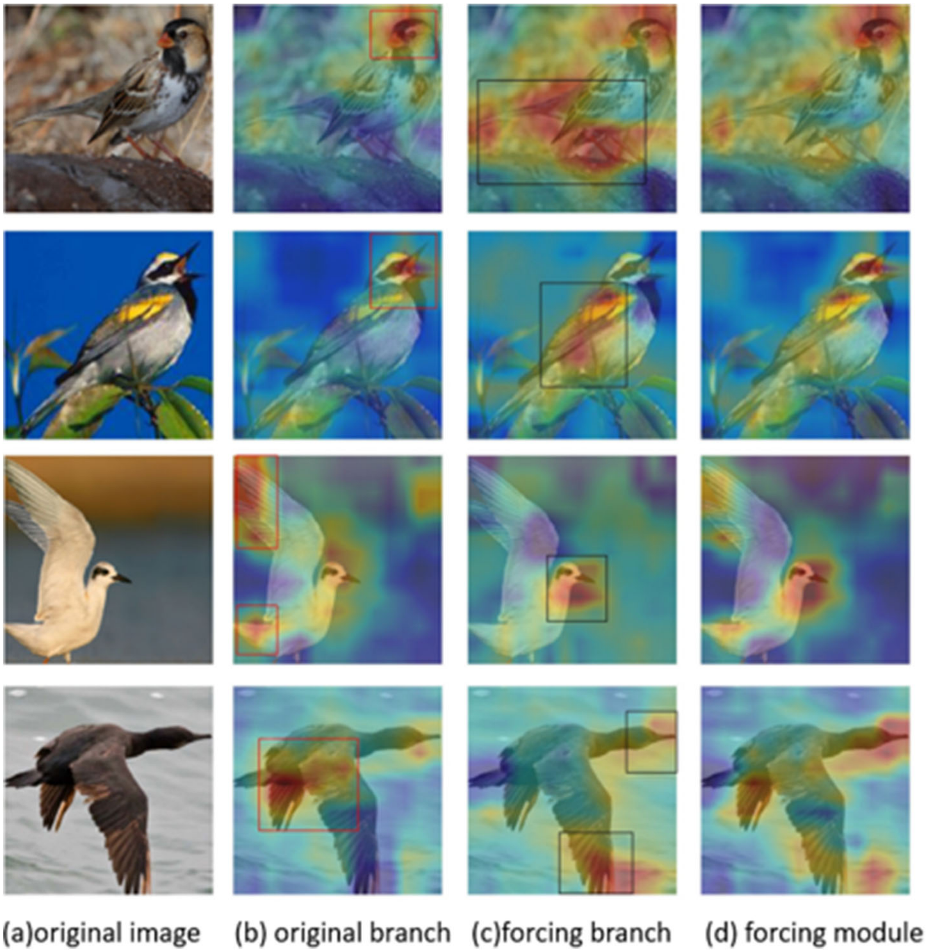

Fig. 3 The experimental results of the forced module are shown visually. From left to right, each column is the original image, the class activation map of the original branch, the class activation map of the forcing branch, and the class activation map of the output of the final forcing module that fuses the original branch and the forcing branch. Among them, the high-response regions in the second column are marked by red boxes, and the high-response regions in the third column are marked by black boxes

performance of FGVC. In the analysis of the results of the three datasets, the CAM-based Copping Module improves the accuracy more significantly.

Impact of forcing branch Basic ResNet-50 with forcing branch achieves 87.3%, 91.7% and 88.5% top-1 accuracy on the CUB-200-201, Stanford-Cars, and FGVC-Aircraft respectively. Since the primary discriminative of the extracted features for forcing branch classifier is suppressed, the network has to focus on other equally important parts rather than the primary discriminative part. It also means we enhanced the weight of the secondary discriminative regions in the extracted features. In the inference phase, the diverse feature will be acquired by CNN and the classifiers of each branch will pay attention to different parts.

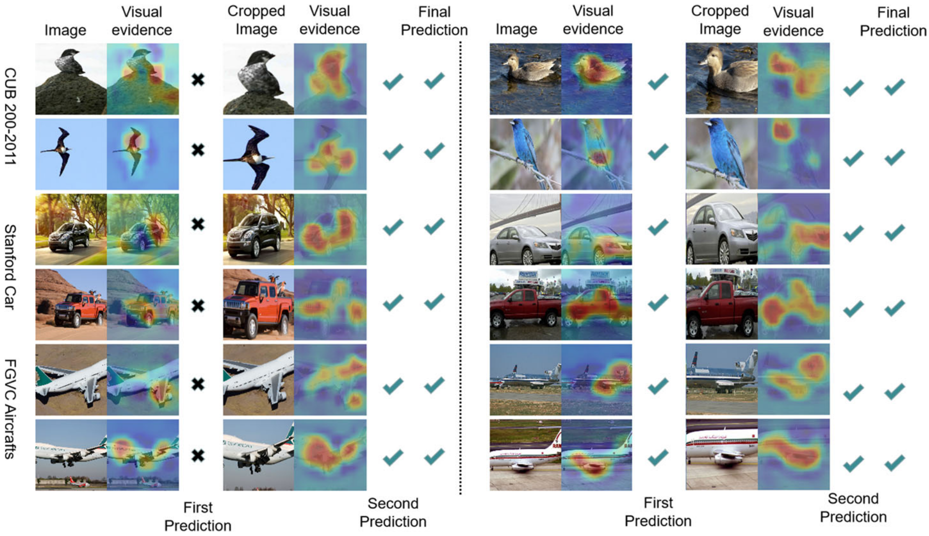

Fig. 4 Visualization of our method. The one to the left of the dotted line is examples where the first prediction was wrong and the second prediction was right and the final prediction was right. The one to the right of the dotted line is examples where the first, the second, and final predictions are right. Each of these examples from left to right is the original image, top-1 class activation map of the original image, prediction of original images, cropped image, top-1 class activation map of the cropped image, prediction of the cropped image, the summation of original image prediction, and cropped image prediction

Table 3 Ablation analysis on the CUB-200-2011

Method	Accuracy
ResNet-50	85.4
ResNet-50+Forcing module	87.3
ResNet-50+CAM-based copping module	88.1
ResNet-50+Forcing module+CAM-based copping module	88.4

Table 4 Ablation analysis on the Standford-Cars

Method	Accuracy
ResNet-50	91.7
ResNet-50+Forcing module	92.5
ResNet-50+CAM-based copping module	93.6
ResNet-50+Forcing module+CAM-based copping module	94.5

Table 5 Ablation analysis on the Fgvc-Aircraft

Method	Accuracy
ResNet-50	88.5
ResNet-50+Forcing module	89.7
ResNet-50+CAM-based copping module	90.7
ResNet-50+Forcing module+CAM-based copping module	93.3

Table 6 Ablation study on the number of suppressing position *k*

k	0	1	2	3	4	5	6
Accuracy	86.9	87.0	87.1	86.8	**87.3**	87.2	87.0

For this, the forcing branch can improve the accuracy of the backbone network respectively by 1.9%, 0.8% and 1.2%.

CAM-based cropping module Because we have conducted panning and rescaling of the images, the prediction of the cropped image is different from the raw image. The Visualization of the result in Fig. 4 shows that the network always pays attention to different parts when the object is panned or zoomed. Analyzing the results on the CUB-200-2011, double prediction improves the result from 85.6% to 88.1%. The 2.7% improvement shows the second prediction can reduce the loss. Compared with double prediction, the combination of forcing model and double prediction leads to an improvement of 0.3%. When we crop the object to the center of the image, since the object is clearer than the first to the network, the network pays attention to more object parts, but the forcing module still forces the network to focus on other confusing parts and improve results from 88.1% to 88.4%. Compared with the Forcing Module, the CAM-based Copping Module improves the accuracy more significantly.

Hyperparameters suppressing factor α and the number of suppressing positions k The accuracy of different k and α setting is shown in Tables 6 and 7. Because we suppress the top-k positions based on class activation maps which is probably vital for classification, suppressing too many positions or setting an over small α will result in lower accuracy. We first fix α to 0.5 and compare the performance of different k. Specifically, k =4 provides the best performance. Then we fix k to 4 and compare the performance of different α. The experiments indicate that α=0.5 promises the best performance on CUB-200-2011.

5 Conclusion

In this paper, we proposed a forcing network to focus on multiple regions as well as extract diverse features for fine-grained visual categorization and we combined the first prediction and second prediction whose input is cropped based on class activation maps from the first prediction as the final prediction to reduce the prediction errors. The forcing network does not require bounding boxes or part annotations and can be trained end-to-end. Our method outperforms the majority of methods of FGVC among datasets of CUB-200-2011, FGVC-Aircraft, Stanford-Cars and achieves state-of-the-art performance on FGVC-Aircraft. Although our method has improved the accuracy greatly, the suppressed region is highly dependent on hyperparameters. Then we try to use hybrid model [8] of the ensemble learning-based method to further improve the accuracy. Our future work will

Table 7 Ablation study on suppressing factor α

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	87.0	87.1	86.7	86.9	**87.3**	86.9	86.9	87.0	86.9	86.9

try to use hyperparameters as model trainable parameters to reduce the dependence on hyperparameters while maintaining high accuracy.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This work is supported by the 2020 Key research and development Plan of Shaanxi Province under project (no: 2020ZDLSF04-02).

Declarations

Conflict of Interests We have no conflict of interests to disclose.

References

1. Azulay A, Weiss Y (2019) Why do deep convolutional networks generalize so poorly to small image transformations? J Mach Learn Res 20:1–25
2. Berg T, Belhumeur PN (2013) POOF: part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In: 2013 IEEE conference on computer vision and pattern recognition, Portland, OR, USA, pp 955–962
3. Chai Y, Lempitsky V, Zisserman A (2013) Symbiotic segmentation and part localization for fine-grained categorization. In: Proceedings of the 2013 IEEE international conference on computer vision, IEEE
4. Chang D, Ding Y, Xie J, Bhunia AK, Li X, Ma Z, Wu M, Guo J, Song Y-Z (2020) The devil is in the channels: mutual-channel loss for fine-grained image classification. IEEE Trans Image Process 29:4683–4695
5. Chen Y, Bai Y, Zhang W, Mei T (2019) Destruction and construction learning for fine-grained image recognition. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), Long Beach, CA, USA, pp 5152–5161
6. Deng J, Dong W, Socher R, Li L, Li K, Li F-F (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, Miami, FL, USA, pp 248–255
7. Ding Y, Zhou Y, Zhu Y, Ye Q, Jiao J (2019) Selective sparse sampling for Fine-Grained image recognition. In: 2019 IEEE/CVF international conference on computer vision (ICCV), Seoul, Korea (South), pp 6598–6607
8. Fan G-F, Yu M, Dong S-Q, Yeh Y-H, Hong W-C (2021) Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling. Util Policy 73:101294
9. Gao Y, Beijbom O, Zhang N, Darrell T (2016) Compact bilinear pooling. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, pp 317–326
10. Ge W, Lin X, Yu Y (2019) Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), Long Beach, CA, USA, 2019, pp 3029–3038
11. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
12. Hanselmann H, Ney H (2020) ELOPE: fine-grained visual classification with efficient localization, pooling and embedding. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, pp 1236–1245

13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, pp 770–778
14. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, vol 2017. pp 2261–2269
15. Khosla A, Jayadevaprakash N, Yao B, Fei-Fei L (2012) Novel dataset for fine-grained image categorization: stanford dogs
16. Kong S, Fowlkes C (2017) Low-rank bilinear pooling for fine-grained classification. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, pp 7025–7034
17. Krause J, Stark M, Deng J, Fei-Fei L (2013) 3D object representations for fine-grained categorization. In: 2013 IEEE international conference on computer vision workshops, Sydney, NSW, Australia, pp 554–561
18. Lin T-Y, RoyChowdhury A, Subhransu M (2015) Bilinear CNN Models for Fine-grained Visual Recognition. In: Proceedings of the IEEE international conference on computer vision, pp 1449–1457
19. Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S (2022) A ConvNet for the 2020s. arXiv:2201.03545
20. Maji S, et al. (2013) Fine-grained visual classification of aircraft hal inria
21. Ng PC, Henikoff S (2003) SIFT: predicting amino acid changes that affect protein function. Nucleic Acids Res 31(13):3812–3814
22. Onyema EM, Elhaj MAE, Bashir SG, Abdullahi I, Hauwa AA, Hayatu AA, Edeh MO, Abdullahi I (2020) Evaluation of the performance of K-nearest neighbor algorithm in determining student learning styles. Int J Innov Sci Eng Technol 7(1):91–102
23. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
24. Semih Kayhan O, van Gemert JC (2020) On Translation Invariance in CNNs: Convolutional Layers Can Exploit Absolute Spatial Location. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), Seattle, WA, USA, pp 14262–14273
25. Sun G, Cholakkal H, Khan S, Khan F, Shao L (2020) Fine-grained recognition, accounting for subtle differences between similar classes. In: Proceedings of the AAAI conference on artificial intelligence, pp 12047–12054
26. Sun M, Yuan Y, Zhou F, Ding E (2018) Multi-attention multi-class constraint for fine-grained image recognition. In: Proceedings of the european conference on computer vision (ECCV), pp 805–821
27. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, USA, pp 1–9
28. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR, pp 6105–6114
29. Wah C, Branson S, Welinder P, Perona P, Belongie S (2011) The caltech-ucsd birds-200-2011 dataset. California Institute of Technology
30. Wang Q, Huang W, Xiong Z et al (2020) Looking closer at the scene: multiscale representation learning for remote sensing image scene classification. IEEE Trans Neural Netw Learn Syst 33:1414–1428
31. Wang Z, Wang S, Li H, Dou Z, Li J (2020) Graph-propagation based correlation learning for weakly supervised fine-grained image classification. In: Proceedings of the AAAI conference on artificial intelligence, pp 12289–12296
32. Xie L, Tian Q, Hong R, Yan S, Zhang B (2013) Hierarchical part matching for fine-grained visual categorization. In: 2013 IEEE international conference on computer vision, pp 1641–1648
33. Xiong Z, Yuan Y, Wang Q (2021) ASK: adaptively selecting key local features for RGB-d scene recognition. IEEE Trans Image Process 30:2722–2733
34. Yang Z, Luo T, Wang D, Hu Z, Gao J, Wang L (2018) Learning to navigate for fine-grained classification. In: Proceedings of the european conference on computer vision (ECCV) pp 420–435
35. Yu C, Zhao X, Zheng Q, Zhang P, You X (2018) Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition. In: Proceedings of the European conference on computer vision (ECCV), pp 574–589
36. Zhang R (2019) Making convolutional networks shift-invariant again. ICML
37. Zhang L, Huang S, Liu W, Tao D (2019) Learning a mixture of granularity-specific experts for fine-grained categorization. In: 2019 IEEE/CVF international conference on computer vision (ICCV), Seoul, Korea (South), pp 8330–8339
38. Zhang N, et al. (2014) Part-based r-CNNs for Fine-Grained Category Detection. In: European conference on computer vision. Springer, Cham
39. Zheng H, Fu J, Mei T, Luo J (2017) Learning multi-attention convolutional neural network for fine-grained image recognition. In: 2017 IEEE international conference on computer vision (ICCV), Venice, Italy, pp 5219–5227

40. Zheng H, Fu J, Zha Z-J, Luo J (2019) Looking for the devil in the details, learning trilinear attention sampling network for fine-grained image recognition. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR), IEEE
41. Zhuang P, Wang Y, Qiao Y (2020) Learning attentive pairwise interaction for fine-grained classification. In: Proceedings of the AAAI conference on artificial intelligence, pp 13130–13137

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.