Fine-Grained Visual Classification using Self Assessment Classifier

Tuong Do[1,2], Huy Tran[1], Erman Tjiputra[1], Quang D. Tran[1], Anh Nguyen[2]
[1]AIOZ, Singapore, [2]University of Liverpool, UK

***Abstract*—Extracting discriminative features plays a crucial role in the fine-grained visual classification task. Most of the existing methods focus on developing attention or augmentation mechanisms to achieve this goal. However, addressing the ambiguity in the top-k prediction classes is not fully investigated. In this paper, we introduce a Self Assessment Classifier, which simultaneously leverages the representation of the image and top-k prediction classes to reassess the classification results. Our method is inspired by self-supervised learning with coarse-grained and fine-grained classifiers to increase the discrimination of features in the backbone and produce attention maps of informative areas on the image. In practice, our method works as an auxiliary branch and can be easily integrated into different architectures. We show that by effectively addressing the ambiguity in the top-k prediction classes, our method achieves new state-of-the-art results on CUB200-2011, Stanford Dog, and FGVC Aircraft datasets. Furthermore, our method also consistently improves the accuracy of different existing fine-grained classifiers with a unified setup.**

***Index Terms*—fine-grained classification, self-supervised learning**

I. Introduction

The fine-grained visual classification task aims to classify images belonging to the same category (e.g., different kinds of birds, aircraft, or flowers). Compared to the ordinary image classification task, classifying fine-grained images is more challenging due to three main reasons: *(i)* large intra-class difference: objects that belong to the same category present significantly different poses and viewpoints; *(ii)* subtle inter-class difference: objects that belong to different categories might be very similar apart from some minor differences, e.g., the color styles of a bird's head can usually determine its fine-grained category; *(iii)* limitation of training data: labeling fine-grained categories usually requires specialized knowledge and a large amount of annotation time. Because of these reasons, it is not a trivial task to obtain accurate classification results by using only the state-of-the-art CNN such as VGG [1].

Recent works show that the key solution for fine-grained classification is to find informative regions in multiple object's parts and extract discriminative features [2]–[4]. A popular approach to learn object's parts is based on human annotations [5]–[7]. However, it is time-consuming to annotate fine-grained regions, hence making this approach impractical. Some improvements utilize unsupervised or weakly-supervised learning to locate the informative object's parts [3], [8] or region of interest bounding boxes [9], [10]. Although this is a promising approach to overcome the problem of manually labeling fine-grained regions, these methods have drawbacks such as low accuracy, costly in training phase/inference phase, or hard to accurately detect separated bounding boxes.

In this paper, we introduce a Self Assessment Classifier (SAC) method to address the ambiguity in the fine-grained classification task. Intuitively, our method is designed to reassess the top-k prediction results and eliminate the uninformative regions in the input image. This helps to decrease the inter-class ambiguity and allows the backbone to learn more discriminative features. During training, our method also produces attention maps that focus on informative areas of the input image. By integrating into a backbone network, our method can reduce the wrong classification over top-k ambiguity classes. Note that *ambiguity classes* are the results of uncertainty in the prediction that can lead to the wrong classification. Our contributions can be summarized as follows.

- We propose a new self class assessment method that effectively jointly learns the discriminative features and addresses the ambiguity problem in the fine-grained visual classification task.
- We show that our method can be easily integrated into different fine-grained classifiers to achieve new state-of-the-art results.

II. Related Works

Fine-grained visual classification involves small diversity within the different classes. Typical fine-grained problems, such as differentiating between animal and plant species, drew much attention from researchers. Since background context acted as a distraction in most cases, many pieces of research focus on improving the attentional and localization capabilities of CNN-based algorithms [11], [12]. Besides, to focus on the informative regions that could distinguish the species between any two images, many methods relied on annotations of parts' location or attributes [7]. Specifically, Part R-CNN [5] and extended R-CNN [13] detected objects and localized their parts under a geometric prior. Then, these works predicted a fine-grained category from a pose-normalized representation.

In practice, it is expensive to acquire pixel-level annotations of the object's parts as ground truth. Thus, methods that require only image-level annotations draw more attention [4], [14], [15]. Lin *et al.* proposed the bilinear pooling [16] and its improved version [17], where two features were combined at each location using the outer product. In [18], the authors introduce the Spatial Transformer Network to achieve accurate classification performance by learning geometric transformations. Yang *et al.* [19] used geographical and temporal

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00117

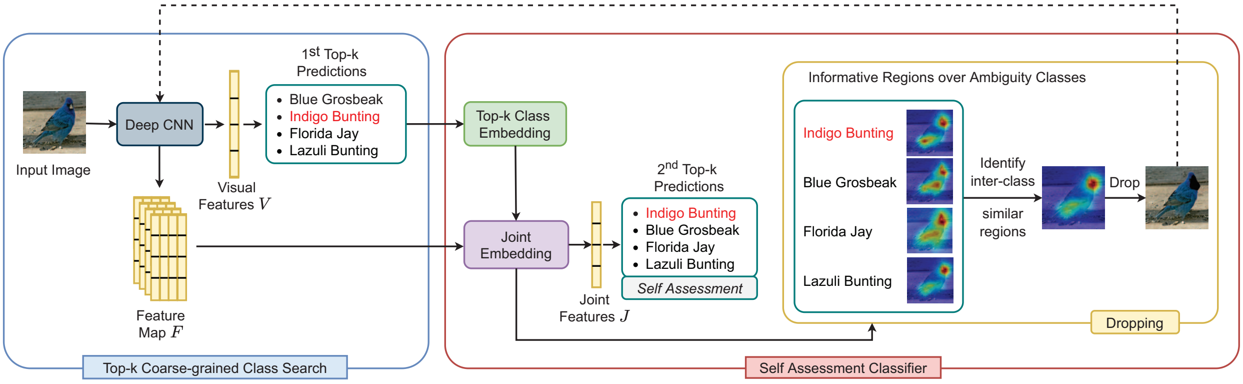

Fig. 1. An overview of our proposed method. The Top-k Coarse-grained Class Search first generates a list of potential top-k prediction candidates using a Deep CNN backbone. The Self Assessment Classifier then reassesses these predictions to improve the fine-grained classification results. The red color class label denotes the ground truth. The black color regions denote dropped regions. Dotted line means the image is used for augmentation. Best viewed in color.

information for improving fine-grained classification results. In [20], a meta-learning framework had been adopted to learn features for fine-grained recognition. Recently, Joung *et al.* [21] leveraged object representation to cope with multiple camera viewpoint problem in the fine-grained classification.

Many works provided a training routine that maximized the entropy of the output probability distribution for training CNNs [14], [22]. Sun *et al.* proposed Multiple Attention Multiple Class loss [23] that pulled positive features closer to the anchor and pushed negative features away. Dubey *et al.* proposed PC [24], which reduced overfitting by combining the cross-entropy loss with the pairwise confusion loss to learn more discriminative features. In [25], by using Maximum-Entropy learning in the context of fine-grained classification, the authors introduced a training routine that maximizes the entropy of the output probability distribution. A triplet loss was used in [26] to achieve inter-class separation. Hu *et al.* [14] proposed a regularization loss to focus on attention regions between corresponding local features. More recently, in [27], diversification block cooperated with gradient-boosting loss had been introduced to maximally separate the highly similar fine-grained classes.

While it is expensive to acquire annotations of object's parts, unsupervised and weakly supervised methods for identifying informative regions have been investigated recently. In SCDA [28], an unsupervised method was introduced to locate the informative regions without using any image label or extra annotation. However, it is less accurate when compared with weakly supervised localization methods, which leveraged image-level super-vision [29]. To locate the whole object, Zhang *et al.* [30] used Adversarial Complementary Learning which could recognize different object's parts and discover complementary regions that belong to the same object. Recently, authors in [31] used Gaussian Mixture Model to learn discriminative regions from the image feature maps for fine-grained classification.

All of the above methods do not focus on the ambiguity prediction classes, which is one of the main reasons that causes wrong classifications. To address this problem, our method is designed to explicitly reduce the effect of the top-k ambiguity prediction classes. Furthermore, our method can effectively learn and produce the attention map in an unsupervised manner. In practice, our method can be easily integrated into different fine-grained classifiers to further improve the classification results.

III. Methodology

A. Method Overview

We propose two main steps in our method: Top-k Coarse-grained Class Search (TCCS) and Self Assessment Classifier (SAC). TCCS works as a coarse-grained classifier to extract visual features from the backbone. The Self Assessment Classifier works as a fine-grained classifier to reassess the ambiguity classes and eliminate the non-informative regions. Our SAC has four modules: the Top-k Class Embedding module aims to encode the information of the ambiguity class; the Joint Embedding module aims to jointly learn the coarse-grained features and top-k ambiguity classes; the Self Assessment module is designed to differentiate between ambiguity classes; and finally, the Dropping module is a data augmentation method, designed to erase unnecessary inter-class similar regions out of the input image. Figure 1 shows an overview of our approach.

B. Top-k Coarse-grained Class Search

The TCCS takes an image as input. Each input image is passed through a Deep CNN to extract feature map $\boldsymbol{F} \in \mathbb{R}^{d_f \times m \times n}$ and the visual feature $\boldsymbol{V} \in \mathbb{R}^{d_v}$. m, n, and d_f represent the feature map height, width, and the number of channels, respectively; d_v denotes the dimension of the visual feature $\boldsymbol{V}$. In practice, the visual feature $\boldsymbol{V}$ is usually obtained by applying some fully connected layers after the convolutional feature map $\boldsymbol{F}$.

The visual features $\boldsymbol{V}$ is used by the 1^{st} classifier, i.e., the original classifier of the backbone, to obtain the top-k prediction results. Assuming that the fine-grained dataset has N classes. The top-k prediction results $\mathbb{C}_k = \{C_1, ..., C_k\}$ is a subset of all prediction classes $\mathbb{C}_N$, with k is the number of candidates that have the k-highest confident scores.

C. *Self Assessment Classifier*

Our Self Assessment Classifier takes the image feature $\boldsymbol{F}$ and top-k prediction $\mathbb{C}_k$ from TCCS as the input to reassess the fine-grained classification results.

Top-k Class Embedding. The output of the TCCS module $\mathbb{C}_k$ is passed through the top-k class embedding module to output label embedding set $\mathbb{E}_k = \{E_1, ...E_i, ..., E_k\}, i \in \{1, 2, ..., k\}, E_i \in \mathbb{R}^{d_e}$. This module contains a word embedding layer [32] for encoding each word in class labels and a GRU [33] layer for learning the temporal information in class label names. d_e represents the dimension of each class label. It is worth noting that the embedding module is trained end-to-end with the whole model. Hence, the class label representations are learned from scratch without the need of any pre-extracted/pre-trained or transfer learning.

Given an input class label, we trim the input to a maximum of 4 words. The class label that is shorter than 4 words is zero-padded. Each word is then represented by a 300-D word embedding. This step results in a sequence of word embeddings with a size of 4×300 and denotes as $\hat{E}_i$ of i-th class label in $\mathbb{C}_k$ class label set. In order to obtain the dependency within the class label name, the $\hat{E}_i$ is passed through a Gated Recurrent Unit (GRU) [33], which results in a 1024-D vector representation E_i for each input class. Note that, although we use the language modality (i.e., class label name), it is not extra information as the class label name and the class label identity (for calculating the loss) represent the same object category.

Joint Embedding. This module takes the feature map $\boldsymbol{F}$ and the top-k class embedding $\mathbb{E}_k$ as the input to produce the joint representation $\boldsymbol{J} \in \mathbb{R}^{d_j}$ and the attention map. We first flatten $\boldsymbol{F}$ into $(d_f \times f)$, and $\mathbb{E}_k$ is into $(d_e \times k)$. The joint representation $\boldsymbol{J}$ is calculated using two modalities $\boldsymbol{F}$ and $\mathbb{E}_k$ as follows

$$\boldsymbol{J}^T = (\mathcal{T} \times_1 \text{vec}(\boldsymbol{F})) \times_2 \text{vec}(\mathbb{E}_k) \tag{1}$$

where $\mathcal{T} \in \mathbb{R}^{d_{\boldsymbol{F}} \times d_{\mathbb{E}_k} \times d_j}$ is a learnable tensor; $d_{\boldsymbol{F}} = (d_f \times f)$; $d_{\mathbb{E}_k} = (d_e \times k)$; vec() is a vectorization operator; $\times_i$ denotes the i-mode tensor product.

In practice, the preceding $\mathcal{T}$ is too large and infeasible to learn. Thus, we apply decomposition solutions that reduce the size of $\mathcal{T}$ but still retain the learning effectiveness. Inspired by [34] and [35], we rely on the idea of the unitary attention mechanism. Specifically, let $\boldsymbol{J}_p \in \mathbb{R}^{d_j}$ be the joint representation of p^{th} couple of channels where each channel in the couple is from a different input. The joint representation $\boldsymbol{J}$ is approximated by using the joint representations of all couples instead of using fully parameterized interaction as in Eq. 1. Hence, we compute $\boldsymbol{J}$ as

$$\boldsymbol{J} = \sum_p \mathcal{M}_p \boldsymbol{J}_p \tag{2}$$

Note that in Eq. 2, we compute a weighted sum over all possible couples. The p^{th} couple is associated with a scalar weight $\mathcal{M}_p$. The set of $\mathcal{M}_p$ is called the attention map $\mathcal{M}$, where $\mathcal{M} \in \mathbb{R}^{f \times k}$.

There are $f \times k$ possible couples over the two modalities. The representation of each channel in a couple is $\boldsymbol{F}_i, (\mathbb{E}_k)_j$, where $i \in [1, f], j \in [1, k]$, respectively. The joint representation $\boldsymbol{J}_p$ is then computed as follows

$$\boldsymbol{J}_p^T = (\mathcal{T}_u \times_1 \boldsymbol{F}_i) \times_2 (\mathbb{E}_k)_j \tag{3}$$

where $\mathcal{T}_u \in \mathbb{R}^{d_f \times d_e \times d_j}$ is the learning tensor between channels in the couple.

From Eq. 2, we can compute the attention map $\mathcal{M}$ using the reduced parameterized bilinear interaction over the inputs $\boldsymbol{F}$ and $\mathbb{E}_k$. The attention map is computed as

$$\mathcal{M} = \text{softmax}\left((\mathcal{T}_{\mathcal{M}} \times_1 \boldsymbol{F}) \times_2 \mathbb{E}_k\right) \tag{4}$$

where $\mathcal{T}_{\mathcal{M}} \in \mathbb{R}^{d_f \times d_e}$ is the learnable tensor.

By integrating Eq. 3 and Eq. 4 into Eq. 2, the joint representation $\boldsymbol{J}$ can be rewritten as

$$\boldsymbol{J}^T = \sum_{i=1}^{f} \sum_{j=1}^{k} \mathcal{M}_{ij} \left((\mathcal{T}_u \times_1 \boldsymbol{F}_i) \times_2 (\mathbb{E}_k)_j \right) \tag{5}$$

It is also worth noting from Eq. 5 that to compute $\boldsymbol{J}$, instead of learning the large tensor $\mathcal{T} \in \mathbb{R}^{d_F \times d_{\mathbb{E}_k} \times d_j}$ in Eq. 1, we now only need to learn two smaller tensors $\mathcal{T}_u \in \mathbb{R}^{d_f \times d_e \times d_j}$ in Eq. 3 and $\mathcal{T}_{\mathcal{M}} \in \mathbb{R}^{d_f \times d_e}$ in Eq. 4.

Self Assessment. The joint representation $\boldsymbol{J}$ from the Joint Embedding module is used as the input in the Self Assessment step to obtain the 2^{nd} top-k predictions $\mathbb{C}'_k$. Note that $\mathbb{C}'_k = \{C'_1, ..., C'_k\}$. Intuitively, $\mathbb{C}'_k$ is the top-k classification result after self-assessment. This module is a fine-grained classifier that produces the 2^{nd} predictions to reassess the ambiguity classification results.

Inspired by [36], [37], the contribution of the coarse-grained and fine-grained classifier is calculated by

$$\Pr(\rho = \rho_i) = \alpha \Pr_1(\rho = \rho_i) + (1 - \alpha) \Pr_2(\rho = \rho_i) \tag{6}$$

where α is the trade-off hyper-parameter $(0 \leq \alpha \leq 1)$. $\Pr_1(\rho = \rho_i), \Pr_2(\rho = \rho_i)$ denotes the prediction probabilities for class ρ_i, from the coarse-grained and fine-grained classifiers, respectively.

IV. Experiment

A. *Experimental Setup*

Dataset	Target	# Cate	# Train	# Test
CUB-200-2011 [38]	Bird	200	5,994	5,794
Stanford Dogs [39]	Dog	120	12,000	8,580
FGVC-Aircraft [40]	Aircraft	100	6,667	3,333

TABLE I

Fine-grained classification datasets in our experiments.

Dataset. We evaluate our method on three popular fine-grained datasets: CUB-200-2011 [38], Stanford Dogs [39] and FGVC Aircraft [40] (Table I).

Implementation. All experiments are conducted on an NVIDIA Titan V GPU with 12GB RAM. The model is trained using Stochastic Gradient Descent with a momentum of 0.9. The maximum number of epochs is set at 80; the weight decay equals 0.00001, and the mini-batch size is 12. Besides, the initial learning rate is set to 0.001, with exponential decay of 0.9 after every two epochs. Based on validation results, the number of top-k ambiguity classes is set to 10, while the parameters d_ϕ, α are set to 0.1 and 0.5, respectively.

Baseline. To validate the effectiveness and generalization of our method, we integrate it into 7 different deep networks, including two popular Deep CNN backbones, Inception-V3 [41] and ResNet-50 [42]; and five fine-grained classification methods: WS [36], DT [43], WS_DAN [14], MMAL [11], and the recent transformer work ViT [44]. It is worth noting that we only add our Self Assessment Classifier into these works, other setups and hyper-parameters for training are kept unchanged when we compare with original codes.

B. Results

Table II summarises the contribution of our Self Assessment Classifier (SAC) to the fine-grained classification results of different methods on three datasets CUB-200-2011, Stanford Dogs, and FGVC Aircraft. This table clearly shows that by integrating SAC into different classifiers, the fine-grained classification results are consistently improved. In particular, we observe an average improvement of +1.3, +1.2, and +1.2 in the CUB-200-2011, Stanford Dogs, and FGVC Aircraft datasets, respectively.

C. Ablation Study

Coarse vs. Fine-grained Classifier. In this work, we consider that both the coarse-grained classifier and the fine-grained classifier are equally important. In practice, we can control the contribution of each classifier by changing the parameter α in Eq. 6. Table III is provided to validate the effect of this parameter using ResNet-50 and WS_DAN on the CUB-200-2011 dataset. This table demonstrates that by fine-tuning the α parameter, the results can be slightly improved. However, we can see that the final classification results do not depend too much on this α parameter.

Complexity Analysis. Table IV shows the efficiency of each module of SAC and its complexity indicated by the GPU speed and the number of parameters during the inference process, when we integrate SAC into ResNet-50 [42] and WS_DAN [14] on the CUB-200-2011 dataset. These results show that SAC increases the performance without affecting the computational cost of backbones.

Language Modality Contribution. In this experiment, we analyze the contribution of language labels. Two experiments are conducted: *(i)* we keep using the labels from the dataset and *(ii)* we replace them by their indexes. Table V shows the experimental results. We can see that the class labels also contribute additional information for the model to learn more effectively. The results confirm that the fine-grained dataset itself contains potential additional information that can be wisely leveraged to improve the learning of the classifier.

Methods	CUB-200 -2011	Stanford Dogs	FGVC Aircraft
MAMC [23]	86.5	85.2	–
PC [24]	86.9	83.8	89.2
MC [45]	87.3	–	92.9
DCL [46]	87.8	–	93.0
ACNet [47]	88.1	–	92.4
DF-GMM [31]	88.8	–	93.8
API-Net [48]	90.0	90.3	93.9
GHORD [49]	89.6	–	94.3
CAL [50]	90.6	–	94.2
Parts Models [15]	90.4	93.9	–
ViT + DCAL [22]	91.4	–	91.5
P2P-Net [12]	90.2	–	94.2
Inception-V3 [41]	83.7	85.1	87.4
Inception-V3 [41]+SAC	85.3(+1.6)	86.8(+1.7)	89.2(+1.8)
ResNet-50 [51]	86.4	86.1	90.3
ResNet-50 [51]+SAC	88.3(+1.9)	87.4(+1.3)	92.1(+1.8)
WS [36]	88.8	91.4	92.3
WS [36]+SAC	89.9(+1.1)	92.5(+1.1)	93.2(+0.9)
DT [43]	89.2	88.0	90.7
DT [43]+SAC	90.1(+0.9)	88.8(+0.8)	91.9(+1.2)
MMAL [11]	89.6	90.6	94.7
MMAL [11]+SAC	90.8(+1.2)	91.6(+1.0)	**95.5**(+0.8)
WS_DAN [14]	89.4	92.2	93.0
WS_DAN [14]+SAC	91.1(+1.7)	93.1(+0.9)	93.9(+0.9)
ViT [44]	91.0	93.2	92.1
ViT [44]+SAC	**91.8**(+0.8)	**94.5**(+1.3)	93.1(+1.0)
Avg. Improvement	+1.3	+1.2	+1.2

TABLE II
CONTRIBUTION (% ACC) OF OUR SELF ASSESSMENT CLASSIFIER (SAC) ON FINE-GRAINED CLASSIFICATION RESULTS.

α (coarse-grained)	$(1-\alpha)$ (fine-grained)	ResNet-50 + SAC	WS_DAN + SAC
0.5	0.5	88.3	91.1
0.7	0.3	88.1	91.0
0.3	0.7	88.3	91.2
0.9	0.1	88.0	91.0
0.1	0.9	87.8	90.9

TABLE III
THE EFFECT OF PARAMETER α, WHICH CONTROLS THE CONTRIBUTION OF COARSE-GRAINED CLASSIFIER AND FINE-GRAINED CLASSIFIER.

Number of Top-k Classes. The accuracy of our proposed method depends on the top-k prediction classes extracted dynamically by the coarse-grained classifier. If the coarse-grained classifier has poor performance and the top-k value is set at a small number, there may be no ground truth class in any top-k predictions. In this case, our fine-grained classifier only penalizes the wrong cases. Therefore, the fine-grained classifier can not improve the accuracy of the network.

Backbone		✓	✓	✓	✓	✓
Auxiliary Classifier			✓		✓	✓
Localization						✓
ResNet-50 Backbone	*#Params(M)*	25.6	25.6	25.6	25.6	38.4
	GPU Time (s/sample)	0.009 ±0.0013	0.009 ±0.0013	0.009 ±0.0013	0.009 ±0.0013	0.017 ±0.0022
WS_DAN Backbone	*#Params(M)*	29.8	29.8	29.8	29.8	49.0
	GPU Time (s/sample)	0.121 ±0.0110	0.121 ±0.0110	0.121 ±0.0110	0.121 ±0.0110	0.201 ±0.0120

TABLE IV
PERFORMANCE AND COMPLEXITY OF EACH MODULE OF SAC.

Method		CUB-200 -2011	Stanford Dogs	FGVC Aircraft
ResNet-50		86.4	86.1	90.3
ResNet-50 + SAC	indexes	87.8	86.9	91.7
	labels	88.3	87.4	92.1
WS_DAN		89.4	92.2	93.0
WS_DAN + SAC	indexes	90.6	92.7	93.6
	labels	91.1	93.1	93.9

TABLE V
THE EFFECTIVENESS OF SAC WITH AND WITHOUT USING CLASS LABELS.

Table VI shows the effect of the number of top-k ambiguity classes on the classification results in our method. From this table, we can see that if the number of top-k classes is set to a small number, our improvement is minimal. In practice, we recommend setting this parameter to a relatively big number to avoid this problem. We choose $k = 10$ in all of our experiments with different methods and datasets.

#Top-k classes	ResNet-50 + SAC	WS_DAN + SAC
2	87.2	89.4
5	88.4	89.6
10	88.3	91.1
20	87.7	90.7
50	85.9	87.2

TABLE VI
THE EFFECT OF DIFFERENT NUMBERS OF TOP-K CLASSES.

D. Qualitative Results

Attention Maps. Figure 2 illustrates the visualization of attention maps between image feature maps and each ambiguity class. The visualization indicates that by employing our Self Assessment Classifier, each fine-grained class focuses on different informative regions.

Prediction Results. Figure 3 illustrates the classification results and corresponding localization areas of different methods. In all samples, we can see that our SAC focuses on different areas based on different hard-to-distinguish classes. Thus, the method can focus on more meaningful areas and also ignore unnecessary ones. Hence, SAC achieves good predictions even with challenging cases.

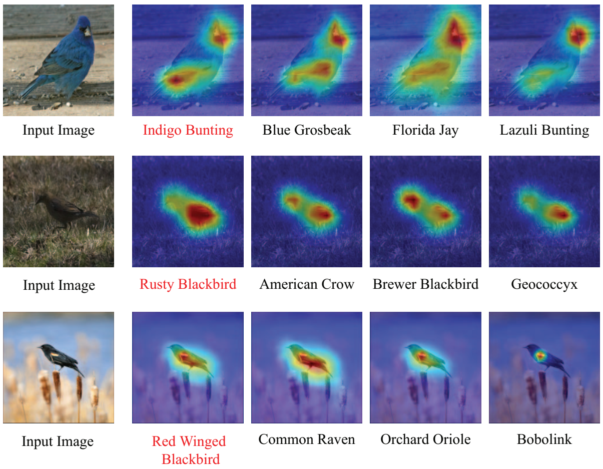

Fig. 2. The visualization of the attention map between image feature maps and different ambiguity classes from our method. The red-colored class label denotes that the prediction is matched with the ground-truth.

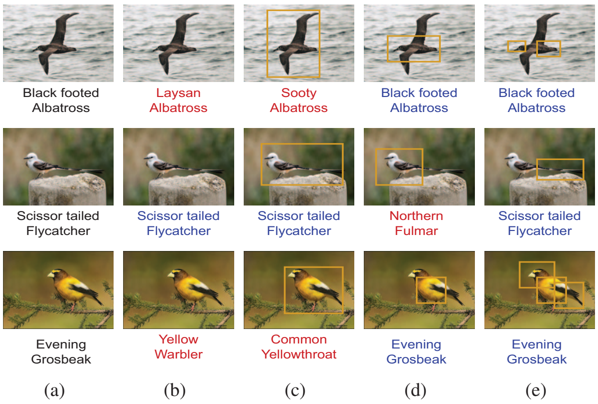

Fig. 3. Qualitative comparison of different classification methods. (a) Input image and its corresponding ground-truth label, (b) ResNet-50 [42], (c) WS_DAN [14], (d) MMAL [11], and (e) Our SAC. Boxes are localization areas. Red color indicates wrong classification result. Blue color indicate correct predicted label. Best viewed in color.

V. CONCLUSION

We introduce a Self Assessment Classifier (SAC) which effectively learns the discriminative features in the image and resolves the ambiguity from the top-k prediction classes. Our method generates the attention map and uses this map to dynamically erase unnecessary regions during the training. The intensive experiments on CUB-200-2011, Stanford Dogs, and FGVC Aircraft datasets show that our proposed method can be easily integrated into different fine-grained classifiers and clearly improve their accuracy.

References

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

[2] 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 *CVPR*, 2019.

[3] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *ICCV*, 2017.

[4] S. Huang, X. Wang, and D. Tao, "Snapmix: Semantically proportional mixing for augmenting fine-grained data," in *AAAI*, 2021.

[5] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *ECCV*, 2014.

[6] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *ICCV*, 2013.

[7] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *ICCV*, 2013.

[8] 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 Recognition*, 2018.

[9] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016.

[10] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *CVPR*, 2017.

[11] F. Zhang, M. Li, G. Zhai, and Y. Liu, "Multi-branch and multi-scale attention learning for fine-grained visual categorization," in *International Conference on Multimedia Modeling*, 2021.

[12] X. Yang, Y. Wang, K. Chen, Y. Xu, and Y. Tian, "Fine-grained object classification via self-supervised pose alignment," in *CVPR*, 2022.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[14] T. Hu and H. Qi, "See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification," *arXiv*, 2019.

[15] W. Ge, X. Lin, and Y. Yu, "Weakly supervised complementary parts models for fine-grained image classification from the bottom up," in *CVPR*, 2019.

[16] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *ICCV*, 2015.

[17] T.-Y. Lin and S. Maji, "Improved bilinear pooling with cnns," in *BMVC*, 2017.

[18] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *NIPS*, 2015.

[19] L. Yang, X. Li, R. Song, B. Zhao, J. Tao, S. Zhou, J. Liang, and J. Yang, "Dynamic mlp for fine-grained image classification by leveraging geographical and temporal information," in *CVPR*, 2022.

[20] H. Tang, C. Yuan, Z. Li, and J. Tang, "Learning attention-guided pyramidal features for few-shot fine-grained recognition," *Pattern Recognition*, 2022.

[21] S. Joung, S. Kim, M. Kim, I.-J. Kim, and K. Sohn, "Learning canonical 3d object representation for fine-grained recognition," in *ICCV*, 2021.

[22] H. Zhu, W. Ke, D. Li, J. Liu, L. Tian, and Y. Shan, "Dual cross-attention learning for fine-grained visual categorization and object re-identification," in *CVPR*, 2022.

[23] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *ECCV*, 2018.

[24] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell, and N. Naik, "Pairwise confusion for fine-grained visual classification," in *ECCV*, 2018.

[25] A. Dubey, O. Gupta, R. Raskar, and N. Naik, "Maximum-entropy fine grained classification," in *NIPS*, 2018.

[26] Y. Wang, J. Choi, V. Morariu, and L. S. Davis, "Mining discriminative triplets of patches for fine-grained classification," in *CVPR*, 2016.

[27] G. Sun, H. Cholakkal, S. Khan, F. Khan, and L. Shao, "Fine-grained recognition: Accounting for subtle differences between similar classes," in *AAAI*, 2020.

[28] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *TIP*, 2017.

[29] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *CVPR*, 2019.

[30] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, "Adversarial complementary learning for weakly supervised object localization," in *CVPR*, 2018.

[31] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, and Z. Li, "Weakly supervised fine-grained image classification via guassian mixture model oriented discriminative learning," in *CVPR*, 2020.

[32] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[33] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[34] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016.

[35] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *NIPS*, 2018.

[36] T. Hu, J. Xu, C. Huang, H. Qi, Q. Huang, and Y. Lu, "Weakly supervised bilinear attention network for fine-grained visual classification," *arXiv*, 2018.

[37] T. Gebru, J. Hoffman, and L. Fei-Fei, "Fine-grained recognition in the wild: A multi-task domain adaptation approach," in *ICCV*, 2017.

[38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[39] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *CVPRW*, 2011.

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

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[43] Y. Cui, Y. Song, C. Sun, A. Howard, and S. J. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *CVPR*, 2018.

[44] M. V. Conde and K. Turgutlu, "Exploring vision transformers for fine-grained classification," in *CVPRW*, 2021.

[45] D. Chang, Y. Ding, J. Xie, A. K. Bhunia, X. Li, Z. Ma, M. Wu, J. Guo, and Y. Song, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *TIP*, 2020.

[46] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *CVPR*, 2019.

[47] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, and F. Huang, "Attention convolutional binary neural tree for fine-grained visual categorization," in *CVPR*, 2020.

[48] P. Zhuang, Y. Wang, and Y. Qiao, "Learning attentive pairwise interaction for fine-grained classification." in *AAAI*, 2020.

[49] Y. Zhao, K. Yan, F. Huang, and J. Li, "Graph-based high-order relation discovery for fine-grained recognition," in *CVPR*, 2021.

[50] Y. Rao, G. Chen, J. Lu, and J. Zhou, "Counterfactual attention learning for fine-grained visual categorization and re-identification," in *ICCV*, 2021.

[51] K. Yue, M. Sun, Y. Yuan, F. Zhou, E. Ding, and F. Xu, "Compact generalized non-local network," *NIPS*, 2018.