SeA: Selective Attention for Fine-grained Visual Categorization

Yajie Chen
chenyajie202103@163.com
Huan Wang
wanghuanphd@njust.edu.cn
Peiwen Pan
121106022690@njust.edu.cn

Nanjing University of Science and Technology China

Abstract

Fine-grained recognition intends to distinguish objects with similar visual signals and has been a challenging problem in computer vision. Vision transformers (ViTs) have recently led the trends of visual representations by a global self-attention mechanism, and exhibit their potential in fine-grained related tasks. Yet, we find that the common ViTs focus on all patches and aggregate spatial features using *shift* operation or down-sampling, tending to overlook locally discernible features. In this paper, we propose a novel scheme, named *selective attention* (SeA), as an alternative to regular self-attention with higher efficiency and conciseness. Specifically, we progressively learn fine-grained features in images by focusing the network on regions with high attention scores via a multi-step training and inference strategy. Also, SeA can be viewed as a plug-and-play module for various hierarchical architectures (e.g., ResNet, Swin) and significantly improves the performance of existing backbones. Extensive experimental results on five fine-grained benchmarks substantiate the effectiveness of our approach, e.g., new SOTA on CUB-200 and Nabird with an accuracy of 93.0% and 93.9%. Code will be made available at link.

1 Introduction

Fine-grained visual classification differs from traditional classification tasks and aims to identify different subclasses within the same broad class. As the differences in objects within such subclasses are often small, this also poses a greater challenge to fine-grained classification. Previous works on fine-grained classification have been devoted to locating the distinguishing parts to extract the fine-grained features. Current localization methods can be broadly classified into the following two types: (1) localization methods based on detection or segmentation (2) localization methods based on attention machine.

In detection or segmentation based localisation methods, most of the early work [13, 26, 27] used manually annotated anchors to extract fine-grained features. However traditional localisation methods require dense manual annotations for training, which is labour intensive work and not conducive to the application of fine-grained classification in life. As a result,

a large number of weakly supervised fine-grained works have emerged, where only image-level labels are used. The main approaches [9, 12, 28] are to generate a large number of proposal parts and then filter regions by clustering or heuristic algorithms. However those approaches can't ensure localisation to fine-grained regions and there are lack of interaction between the localised regions.

In view of this, many works [8, 14, 29] have turned to a localisation approach based on the attention mechanism. Attention mechanism, similar to human perception, selectively focuses on the salient parts of an object or scene. Specifically it works by performing a weighted summation of different patches to achieve a localisation-like effect. This is a good way of modelling the global picture, but it also raises the question that are all patches we need? Although attention learns weights for the different patches to highlight the important ones, the useless ones are also counted. To solve this problem, further work [3, 10, 25] proposed to select patches first, and then perform attention calculation within the selected patches. Such approaches often require the design of a complex token selected module, which increases the complexity of the model and lacks interpretability.

In summary, previous works have been limited by two general problems: (1) The global self-attention contains a large amount of redundant and useless information. (2) The token selected module is accompanied by complex design and inefficient computation.

In order to solve the appeal problem, we propose a simple and efficient selective attention. SeA does not weight the global tokens to sum up, but only calculates some tokens with high attention score. This method does not introduce additional calculation, and the selective principle is based on the attention map generated in the calculation process. In addition, we also borrowed the progressive multi granularity training strategy [7], and improved on it by exploiting the difference between inference and training, so as to promote the network focus on the fine-grained part. We call this method DIT. The main contributions of this paper are:

- We propose a novel attention mechanism, namely selective attention, which unifies selected module and attention mechanism without introducing extra computational effort. And We build a plug-and-play FGVC Head module with this core, applying it to different frameworks to verify the effectiveness for fine-grained feature extraction.
- A more optimal solution named DIT is proposed based on the Progressive Multi-Granularity Training strategy as a generic strategy during training and inference, and its effectiveness is demonstrated by applying it to different backbones in different benchmarks.
- Extensive experiments on fine-grained visual representation tasks suggest the superiority of proposed methods. For instance, we achieved a new SOTA of 93.0% on the CUB-200, 93.9% on the Nabird.

2 Related Work

2.1 Backbone

As AlexNet [17] achieving the 2012 ImageNet [4] classification championship, CNNs have once become the most popular feature extraction method within the past few years. A large number of deep neural networks based on CNN have emerged. He *et al*. [11] proposed ResNet with residual connections, which solves the problem of gradient disappearance or

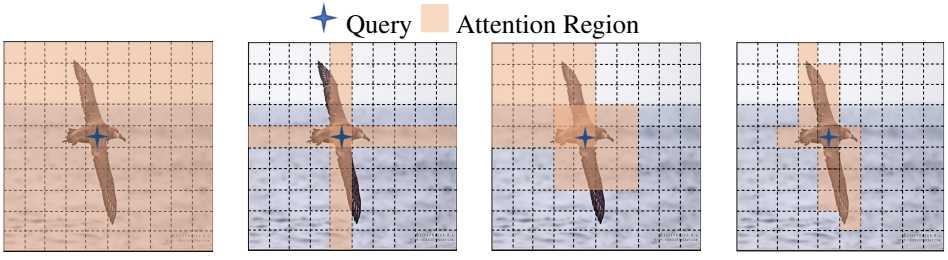

Figure 1: An overview of different attention mechanism. (a) is the standard global self-attention. (b) is the cross self-attention that computes in a cross shape.(c) is the shift-window attention and only perform self-attention in each window. We propose SeA given in (d) which can select part tokens adaptively by attention score.

gradient explosion in deep neural networks. As the network deepens, the number of parameters inevitably increases. In view of this, Howard al. [11] proposed a lightweight MobileNet that uses depthwise convolution instead of vanilla convolution, greatly reducing the parameters of the convolution operation and still achieving a competitive performance. Later Tan *et al*. [22] proposed an efficient EfficientNet that balances the width and depth of the network with the resolution of the input image to achieve higher accuracy with a limited number of parameters.

It was not until the excellent performance of ViT [6], that lots of works turned to the transformer. ViT divides images into different patches and adds a classification token with a learnable position embedding, achieving an accuracy of 90.7% on the ImageNet after a large amount of data pre-training. The Swin transformer [18] was proposed to reduce the quadratic complexity to linear and achieved SOTA on a variety of downstream tasks. Interestingly, the proposal of Convnext [19] brought a renewed focus on CNNs at a time when the transformer was rapidly evolving. The main body of Convnext is much like ResNet in that it references the block numbers and normalization methods of the transformer and uses depthwise convolution to achieve top1 accuracy on ImageNet.

2.2 Attention Mechanism

With the rise of the transformer, there has also been an explosion of improvements on the attention mechanism, of which the main attention variants are briefly shown in Fig1. First is the standard self-attention in Fig1.(a), which is a strategy for modeling the global by weighted summation. To reduce the computational complexity of self-attention, two main variants of attention are proposed:(1)hand-craft self-attention region as shown in Fig1.(b). (2)window-based self-attention as shown in Fig1.(c). Cross shape self-attention was proposed in [5], which artificially delineated attention zones as crosses. While shift window attention divides the image into non-overlapping windows to calculate attention within the window, then move the window to recalculate attention. Neither window-based attention nor hand-craft attention region can avoid global information loss, while global self-attention contains too much useless information. Our proposed SeA solves the appeal problem by extracting the key patch information better, although not reducing the computational complexity. Our work is largely inspired by self-attention.

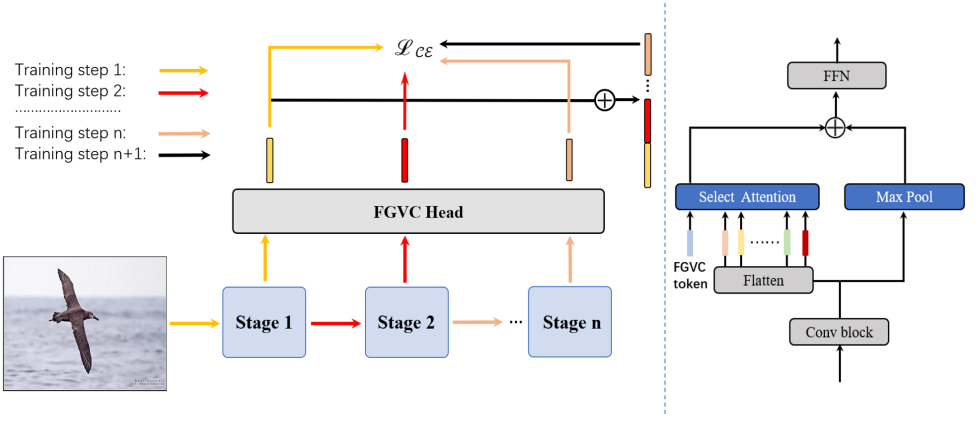

Figure 2: Overall structure diagram of the model with DIT strategy. The left part represents the training process of the model using the DIT strategy, and the right part shows the structure of our proposed FGVC Head. The stage can be any hierarchical down sampling network model, E.g.Resnet,Convnext,Swin transformer.

2.3 Fine-grained Classification

Identifying the distinguished parts is the most vital step in fine-grained classification. Localization methods can be broadly divided into those based on segmentation or detection [9, 12, 26, 27, 28] and those based on attention mechanisms. Our work falls into the latter paradigm. In this paradigm, [8] first applied attention to fine-grained recognition. Specifically, it uses an iterative visual attention model to select a series of attention regions, using previous predictions as a reference to generate a regional attention map iteratively. [3] proposed a generic feature selection module that keeps the features with classification probability above a threshold. [10] integrates the original attention weights into the attention map to guide the network to select distinguishing patches and calculate the relationships between them. Our work unifies part selected and attention mechanism and proposes a new paradigm, which significantly simplifies the fine-grained classification steps.

3 Approach

3.1 Multi-stage Training and Inference

In simple terms, DIT is a multi-stage training and inference strategy. The feature map of the lower stage is first trained, and then gradually progresses to the higher stage. As the network progresses to the higher stage, the focus of the model shifts from local details to global structural discriminative information, rather than learning all the granular information simultaneously. Specifically, the overall structure of DIT is shown in the left part of Fig2. After stage 1, the low-level representation of image is connected to a FGVC Head to obtain the fine-grained feature, and calculate the cross entropy loss with the ground truth as follows:

$$F_i = \text{head}\left(\text{stage}_i\left(\text{input}_{i-1}\right)\right) \tag{1}$$

$$\text{pre}_i = \text{softmax}\left(\text{linear}\left(F_i\right)\right) \tag{2}$$

$$\text{loss}_i = -\sum(GT \log pre_i + (1-GT)\log(1-pre_i)) \tag{3}$$

$$F_{n+1} = \texttt{Concat}(F_1, F_2 \ldots\ldots F_n) \tag{4}$$

Where $i \in (1,2,\ldots\ldots,n)$ denotes the number of stages of backbone; F_i denotes the final feature vector and pre_i denotes the probability distribution of each category after softmax.

Then the parameters of $stage_1$ and $head_1$ are updated according to the backward of $loss_1$, completing the training step1. At the same time, the feature map from $stage_1$ is downsampled through $stage_2$ and the appeal operation is repeated. After n stages, the features of each stage are cancated to obtain a feature that fuses multi-stage as Eq4, and then backward is done again to update the parameters of the whole network. For the specific application, we will select the last s stages for training instead of all of them participating in the training. The stages are: $\{$ stage $n-s+1$, stage $n-s+2,\ldots\ldots,$ stage $n\}$.

For the inference stage, we similarly integrate the prediction for the last n stages, and it is worth noting that here both n and s are adjusted as hyperparametric. We believe that if the predictions of low stages are integrated it may suppress the classification effect because the classification of the low stages is not very good in the first place. Our proposed DIT discusses the difference between training and inference in more depth, separating the training stage number s and inference stage number n as two hyperparametric where $n \leq s$. We explore in detail the optimal inference stages for different training stages, and the optimal number of training stages and inference stages for the global case.

3.2 FGVC Head

The overall idea of the FGVC Head is to extract fine-grained and coarse-grained features of the image separately. Specifically, to facilitate the subsequent processing, we first transform the feature map by a 1×1 convolution to adjust the feature map of different stages to a uniform dimension. Next, we interact with the information around the different channels and spaces of the feature map through a 3×3 convolution. Referring to the convblock design paradigm, batch normalization and activation function relu are added after the convolution to improve the training speed and non-linearity of the network. After pre-processing the feature map, we extract its fine-grained features and coarse-grained features in parallel through two branches as shown in the right part of Fig2. For coarse-grained feature extraction, we do not need the information of each point of the feature map, but only the position of the point with the largest response. We use global max pooling to ignore regions with low scores. For fine-grained feature extraction, we refer to ViT's design idea of adding a learnable FGVC token as a fine-grained feature. We let it do selective attention with other tokens and learn a feature that contains local fine-grained information. Finally, we stitch the two granularity features together and pass it through a feed forward network to obtain the classification result.

The main advantage of our FGVC Head is that it extracts coarse-grained features and fine-grained features for the same feature map in parallel. For fine-grain feature extraction, there is no additional localization sub-network for token selection, which greatly simplifies the steps of fine-grain feature extraction. Overall, our proposed FGVC Head is a simple, efficient and plug-and-play fine-grain classification head.

3.3 Selective Attention

The original design motivation for SeA was that are all patches we need for computing self-attention? That is clearly NO. For example, the background part of an image often contain a lot of useless information. These will affect our feature extraction of the subject and play a negative role. Although all tokens are given a relevance weight when calculating self-attention to reduce the impact of useless tokens, negative tokens are still taken into account in a low weight. For this reason, our SeA is born to incorporate only the information of the positive tokens and discard the negative tokens. This design idea is similar to fine-grain feature extraction in FGVC, so we apply it to FGVC as a new fine-grain feature extraction method.

SeA is used to selectively compute the key tokens by masking others with low attention scores. Specifically, borrowing from multi-head self-attention, features are first mapped to Q, K and V. Then the dimensions are divided equally among multi-heads, prompting features within different heads to learn different semantic information. For a single head, we first compute the matrix multiplication of Q and K^T to obtain an n×n attention score mat. We define $score_{i,j}$ as the relevance of the ith token to the jth token. For the ith row of the attention score mat, we select the high relevance scores to keep them and the other low relevance scores set to zero. In the specific implementation process, we first sort the index of the ith row of score in descending order and record the $(n \times select\ ratio)$th index. We find the watershed score by selected index and do not disrupt the ordering of the original score. Then compare $score_{i,j}$ with the watershed score, those greater than it are kept and those less than are set to zero as follow:

$$\text{score} = Q \cdot K^T / \sqrt{d_k} \tag{5}$$

$$\text{index} = \text{argsort}(-\text{score}_i)[\text{ratio} \times n] \tag{6}$$

$$\text{score}_i = \text{where}(\text{score}_i > \text{score}_i[\text{index}], \text{score}_i, 0) \tag{7}$$

where $i \in (1, 2, \ldots\ldots, n)$ denotes the number of all tokens; d_k denotes the dimension in a single head of K; $ratio$ controls the select number.

We then normalize the newly obtained attention score by a softmax function. It is worth noting that we perform softmax on the attention score after the selection to strengthen the role of the selective tokens. The normalized attention score is then weighted and summed over V by calculating the matrix multiplication of the attention score and V. The above is the SeA calculation within a single head. We do the calculation within each head individually and then stitch together the output of all the heads as the final select attention output as follows:

$$att^h = \text{softmax}(\text{score}) \cdot V \tag{8}$$

$$att = \text{concat}\left(att^1, att^2, \cdots\cdots, att^h\right) \tag{9}$$

4 Experiment

4.1 Experiment Setup

Dataset. For this experiment, we used five popular fine-grained benchmarks to demonstrate the generality of our model, including CUB-200 [24], Nabird [23], FGVC Aircraft [20],

Stanford Cars [16], and Stanford Dogs [15].

Implementation details. To be fair for comparison with other methods, the parameters of our implementation are set in line with most methods. First, we resize the image to 550×550 size, then random crop to 448×448 size, and use the centercrop in the test with the same size. We just use the random horizontal flip as the data augmentation. During training, the cosine decay is used; The weight decay is set to 0.0005; SGD is used as the optimizer, and the batch size is set to 16; A total of 100 epochs are trained. It is worth noting that we set different initial learning rates for different parameters. For backbone, we set 0.0001 as its initial learning rate, and for FGVC Head we set 0.001. We aim is to fine-tune the weighting of the backbone and increase the weighting variation of the FGVC Head. For all backbones we loaded pre-trained weights on ImageNet 21k. All experiments are completed on a single Nvidia GeForce RTX 3090, and the Pytorch toolbox is used as the main implementation substrate. If not otherwise specified, all experiments were parameterized as shown above.

Table 1: Comparison of various methods on five benchmarks, namely CUB-200, Nabird, Stanford Car, FGVC Aircraft, and Stanford dog.

Method	Backbone	CUB-200	Nabird	Car	Aircraft	Dog
API-Net [30]	Densenet-161	90.9%	88.1%	95.3%	93.9%	90.3%
PMG [7]	Resnet-50	89.6%	-	95.1%	93.4%	-
FFVT [25]	ViT-B_16	91.6%	-	-	-	91.5%
TransFG [10]	ViT-B_16	91.7%	90.8%	94.8%	-	**92.3**%
PIM [3]	Swin-T	92.8%	92.8%	-	-	-
CAL [21]	Resnet-101	90.6%	-	95.5%	94.2%	-
CAP [2]	Xception	91.8%	91.0%	**95.7**%	**94.9**%	-
SeA(ours)	-	**93.0**%	**93.9**%	95.3%	94.4%	90.9%

4.2 Compare With State-Of-The-Art Approach

We compared the performance of SeA with some other state-of-the-art methods on five popular fine-grained benchmarks, and the results are shown in Table 1. Overall, most previous methods have only shown SOTA or competitive performance on a portion of benchmarks and very few have been experimentally analysed on the full benchmarks, e.g., PIM is a previous SOTA on CUB-200 and Nabird, but it has no relevant experiments on other benchmarks; CAP is a SOTA method on Stanford Car and FGVC Aircraft, but it does not work very well on CUB-200 and Nabird. Our approach achieves the highest performance on the CUB-200 and Nabird with 93.0% and 93.9%, and very competitive performance on the Stanford Car, FGVC Aircraft, and Stanford Dog.

Specifically, The previous best method on CUB-200 and Nabird was PIM, which achieved 92.8% performance on both benchmarks. However PIM uses a supervised approach to select the fine-grained token, which is not suitable for model scaling because of the fusion of multiple losses. In contrast, our SeA improves 0.2% on CUB-200 and 1.1% on Nabird compared with the PIM, and we use only one type of loss as the direction of model optimization, which is more scalable.

In the Stanford Car and FGVC Aircraft, CAP achieves the best performance on them based on anchors. While our approach does not divide any anchors in advance and only extracts fine-grained features through feature maps, which greatly simplifies the fine-grained classification step but is slightly less performance than CAP. We believe that is because the

shape and pose of cars and aircrafts are more fixed and the features that need to be extracted focus on texture. That makes these benchmarks more friendly to the anchor based method. However, for the birds, its shapes and poses are more variable so that we need to extract general features. At this point the anchor limits the expression of the pose, whereas our SeA does not require any anchors to determine the fine-grained part, and is more suitable for the general case. For the Stanford Dog, TransFG achieves top-1 accuracy with 92.3%. TransFG designs a token selected module first, and then does self-attention in these tokens. Although our method is inferior to TransFG and FFVT on this benchmark, we unify the selected module and attention mechanism. And we have a significantly higher overall performance than the other methods.

Table 2: The accuracy under different n and s in CUB-200 with Swin-B. *s* denotes the number of last stages for training; *n* represents the number of last stages for inference.

s \ n	0	1	2	3
1	**92.5**%	92.4%	-	-
2	92.6%	92.8%	**93.0**%	-
3	**92.6**%	92.4%	92.5%	92.4%

Table 3: The accuracy of different selection ratio in CUB-200 with Swin-B.

Select Ratio	0.1	0.3	0.5	0.7	0.9	1.0
Accuracy	92.7%	92.8%	**93.0**%	92.5%	92.6%	92.6%

4.3 Ablation Study

Hyperparametric analysis. For the DIT strategy, we conducted adequate experiments on the training stage number s as well as the inference stage number n. The specific experimental results are shown in Table2, where $n \leq s$. It can be seen that it achieves the highest accuracy with 93.0% when $s = 2$ and $n = 2$. The optimal results can be achieved when we progressively train the last two stages and integrate the last two stages when inferring. Moreover, when $s = 1$ and $s = 3$, both are not optimal when the training stage number is directly taken as the inference stage number, which also proves our conjecture for the proposed DIT.

The select ratio is the most important and only hyperparametric in SeA, which represents the number of tokens we want to select. Since each stage has a different number of tokens, we use a ratio to unify the representation of each layer. The SeA turns into self-attention when $select\ ratio = 1$. As shown in Table3, the performance reaches a maximum of 93.0% when $select\ ratio = 0.5$. This also proves the motivation of SeA that not all tokens have a positive effect on classification. Interestingly, when we keep reducing the select ratio, the accuracy decreases but is still higher than when the self-attention, demonstrating that fusing global information is not as useful as fusing some of the key patches.

Influence of DIT and FGVC Head. The performance of adding DIT strategy and FGVC Head on different backbones is shown in Table4. In each backbone, the first row shows the original performance of the model, the second row shows the performance after adding the DIT strategy, and the third row shows the performance after adding the FGVC Head to the DIT strategy. Our method improves the performance of the original model by more than 2% in most cases. And It is even greater than 4% improvement on the aircraft for Swin-B, and

Table 4: Ablation study on DIT and FGVC Head in five datasets with different backbone.

Backbones	Method	CUB-200	Nabird	Car	Aircraft	Dog
Resnet-50	Original	84.5%	84.3%	91.5%	90.3%	86.1%
	+DIT	86.8%	86.5%	94.2%	90.9%	87.0%
	+FGVC Head	88.4%	87.5%	94.5%	92.1%	88.2%
Convnext-B	Original	91.7%	91.9%	93.0%	91.0%	89.9%
	+DIT	91.9%	92.4%	94.3%	91.5%	90.3%
	+FGVC Head	92.5%	92.7%	95.1%	93.2%	**90.9**%
Swin-B	Original	92.2%	92.8%	93.7%	90.3%	89.3%
	+DIT	92.6%	93.1%	94.9%	93.3%	89.6%
	+FGVC Head	**93.0**%	**93.9**%	**95.3**%	**94.4**%	90.0%

have 0.7% in the worst case. The extensive experiments demonstrate that our method is not only effective, but also versatile.

Table 5: Ablation study on SeA in FGVC Head in five datasets with different backbone.

Backbones	Method	CUB-200	Nabird	Car	Aircraft	Dog
Resnet-50	Self-Attention	87.7%	87.4%	94.3%	90.4%	88.0%
	SeA	88.4%	87.5%	94.5%	92.1%	88.2%
Convnext-B	Self-Attention	92.2%	92.6%	94.5%	93.0%	90.4%
	SeA	92.5%	92.7%	95.1%	93.2%	90.9%
Swin-B	Self-Attention	92.6%	93.2%	94.9%	93.5%	89.3%
	SeA	93.0%	93.9%	95.3%	94.4%	90.0%

Influence of Selective Attention. Further to the ablation study in Table4, we focus on the role of SeA in the FGVC Head. We compare it with self-attention, and the results are shown in Table5. For most cases, we can obtain a large improvement by simply replacing self-attention with SeA. In the FGVC Aircraft under Resnet-50, we achieve the highest improvement of 1.7%, while in the Nabird it was only 0.1%. We believe this is due to the high resolution of the Nabird, where a random crop of 448 × 448 would lose a lot of information. Once We crop the image to 672×672 in Swin-B, it achieves a boost of 0.7%. It is worth noting that these improvements are only stacking one layer of SeA.

4.4 Qualitative Analysis

We show the visualization results of proposed SeA and standard self-attention on CUB-200 in Fig3. In the first row we give the visualisation of the attention map under SeA. As a comparison, we show the results of the standard self-attention visualisation in the second row. As can be seen, the visualization of self-attention is coarser and the parts that are focused on are more scattered. It even focuses on some background parts. Our selective attention, on the other hand, focuses more on objects, such as the head or beak of a bird. This is in line with our motivation for proposing SeA.

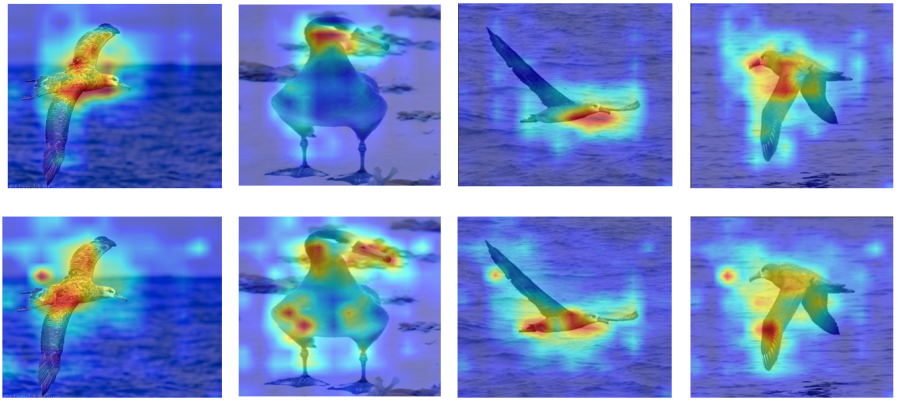

Figure 3: Visualization results of attention maps trained under different attention mechanisms on CUB-200. The first row represents the visualization of selective attention, the second row represents the standard self-attention.

5 Conclusion

In this work, we propose a novel attention mechanism SeA, which adaptively selects high score tokens based on an attention map. In addition, we exploit a general training inference strategy DIT, which is trained in a stepwise progressive manner and integrates predictions from multiple stages during inference. The method substantially improves the existing backbones and achieves the state-of-the-art performance on CUB-200 and Nabird and shows very competitive performance on the other three benchmarks.

SeA achieves satisfactory results for fine-grained visual classification, and we are looking forward to its future performance on other tasks. We plan to build a backbone with SeA as the core to test its performance on small target detection. In addition, selective attention is still quadratic in complexity, which is a future direction for improvement.

References

[1] G Andrew, Zhu Menglong, et al. Efficient convolutional neural networks for mobile vision applications. *Mobilenets*, 2017.

[2] Ardhendu Behera, Zachary Wharton, Pradeep RPG Hewage, and Asish Bera. Context-aware attentional pooling (cap) for fine-grained visual classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 929–937, 2021.

[3] Po-Yung Chou, Cheng-Hung Lin, and Wen-Chung Kao. A novel plug-in module for fine-grained visual classification. *arXiv preprint arXiv:2202.03822*, 2022.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In *European Conference on Computer Vision*, pages 153–168. Springer, 2020.

[8] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017.

[9] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3034–3043, 2019.

[10] Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang. Transfg: A transformer architecture for fine-grained recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 852–860, 2022.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Xiangteng He and Yuxin Peng. Weakly supervised learning of part selection model with spatial constraints for fine-grained image classification. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[13] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1173–1182, 2016.

[14] Ruyi Ji, Longyin Wen, Libo Zhang, Dawei Du, Yanjun Wu, Chen Zhao, Xianglong Liu, and Feiyue Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10468–10477, 2020.

[15] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[19] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[20] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[21] Yongming Rao, Guangyi Chen, Jiwen Lu, and Jie Zhou. Counterfactual attention learning for fine-grained visual categorization and re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1025–1034, 2021.

[22] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[23] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

[24] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[25] Jun Wang, Xiaohan Yu, and Yongsheng Gao. Feature fusion vision transformer for fine-grained visual categorization. *arXiv preprint arXiv:2107.02341*, 2021.

[26] Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognition*, 76:704–714, 2018.

[27] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[28] Yu Zhang, Xiu-Shen Wei, Jianxin Wu, Jianfei Cai, Jiangbo Lu, Viet-Anh Nguyen, and Minh N Do. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Transactions on Image Processing*, 25(4):1713–1725, 2016.

[29] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5012–5021, 2019.

[30] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13130–13137, 2020.