PEDTrans: A fine-grained visual classification model for self-attention patch enhancement and dropout

Xuhong Lin[1], Qian Yan[1], Caicong Wu[1,2], and Yifei Chen[1,2⋆]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
glhfei@126.com
[2] Key Laboratory of Agricultural Machinery Monitoring and Big Data Applications, Ministry of Agriculture and Rural Affairs, Beijing 100083, China

Abstract. Fine-grained visual classification (FGVC) is an essential and challenging classification task in computer visual classification, aiming to identify different cars and birds. Recently, most studies use a convolutional neural network combined with an attention mechanism to find discriminant regions to improve algorithm accuracy automatically. However, the discriminant regions selected by the convolutional neural network are extensive. Vision Transformer divides the image into patches and relies on self-attention to select more accurate discriminant regions. However, the Vision Transformer model ignores the response between local patches before patch embedding. In addition, patches usually have high similarity, and they are considered redundant. Therefore, we propose a PEDTrans model based on Vision Transformer. The model has a patch enhancement module based on attention mechanism and a random similar group patch discarding module based on similarity. These two modules can establish patch local feature relationships and select patches that are easier to distinguish between images. Combining these two modules with the Vision Transformer backbone network can improve the fine-grained visual classification accuracy. We employ commonly used fine-grained visual classification datasets CUB-200-2011, Stanford Cars, Stanford Dogs and NABirds to get advanced results.

Keywords: Fine-grained visual classification · Vision Transformer · Self-attention.

1 Introduction

In deep learning, convolutional neural networks have been rapidly developed and used in various computer vision tasks. Fine-grained visual classification has long been a challenging task because it categorizes sub-classes within categories with little difference (for example, similar birds, dogs, and car types). In the early stages of the fine-grained visual classification model, researchers used expensive

⋆ Corresponding Author

location-calibrated data to classify, relying on additional bounding boxes to locate detailed parts of the image [1, 15]. However, with the development of the fine-grained visual classification network, researchers found that this calibration method was not the best because the annotator's notes might be incorrect and time and effort consumed. Thanks to the development of primary convolutional networks, new progress has been made in fine-grained visual classification networks and classification methods. Methods [18, 38] represent weakly supervised

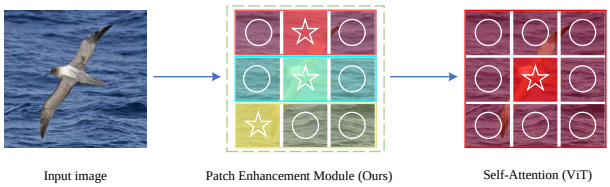

Fig. 1. Repeated and redundant patches are marked with a hollow circle, and local and global enhanced patches are marked with a hollow five pointed star.

learning, which avoids the use of expensive annotation data. Most of the current methods use weak supervision to solve this problem, effectively saving the labelling cost and producing good results. Transformer achieves good results in text tasks [4, 26, 3] and achieves the same results in visual classification tasks as deep convolutional networks [7]. Recently, it has been widely used in this field. Because the input of the visual converter is patch data, it can better find important parts in fine-grained visual classification.

Although the Vision Transformer model adds the position vector in the patch embedding stage, it only increases the relative position between patches and cannot establish the relationship between local patches. In addition, the model divides the input image into small patches, so there are many duplicate parts between patches. Effectively removing these duplicate patches can make the model better distinguish images. These problems are shown in Fig. 1, where the dotted box is the newly added Patch Enhancement Module. It more clearly presents enhanced local patches and redundant patches. Global and local enhanced patches can be more clearly distinguished.

This study proposes a fine-grained visual classification model for self-attention enhancement and random dropout patches to address these issues. We add two separate modules for the transformer to achieve better classification results, one is Patch Enhancement Module, and the other is Dropout Patch Module. The relationships between local patches can be established by the Patch Enhancement Module and make important patch information more prominent. The Dropout Patch Module uses the similarity between patches to select necessary patches and then classifies them using the most effective patches to avoid redundant information. The marked patches will be put into the transformer as new input and classified using the position token. Finally, we conduct extensive experiments on

standard fine-grained Vision classification data sets and get better results than the convolutional neural network.

Our method is more effective than the existing methods and does not need additional annotation information. The main contributions we create in this paper are summarized as follows:

1. We use a method similar to channel attention to establishing the local relationship between patches. The patch information of effectively distinguishing images can be more significant through the self-attention mechanism, equivalent to the first selection of important feature regions.
2. Although patch information is beneficial to the search of important discrimination areas, there is a large amount of redundant information and high similarity in patches. We propose a method to drop out the repeated patches according to the information similarity, and the remaining patches will be easier to distinguish the images.
3. We conduct extensive testing on fine-grained classification data sets. The results show that the Vision Transformer accuracy is improved and advanced performance is achieved by adding the Patch Enhancement Module and the Dropout Patch Module.

2 Related Work

The second section introduces the related work and methods of fine-grained visual classification, channel attention, dropout method, and Vision Transformer.

2.1 Fine grained visual classification method

Zhang et al. [35] proposed a local-based Part-based R-CNN (Part R-CNN) fine-grained visual classification algorithm in 2014, which directly uses convolutional features for classification. Wei et al. [32] proposed the Mask CNN model, an end-to-end deep convolutional model, which is different from the Part R-CNN algorithm, which evaluates and screens the characteristics of deep convolution. Lin et al. [18] designed an effective Bilinear CNN model framework. Bilinear channel features extract the paired correlation between channels and then distinguish the subtle differences between images. This structure can obtain feature information of different granularity and then improve classification accuracy. Fu et al. [9] proposed a Recurrent Attention Convolutional Neural Network (RA-CNN) based on an attention mechanism that learns and discriminates region attention and region-based feature representation recursively and enhances each other. The progressive Multi-Granularity (PMG) model [8] adopts a progressive learning method and random patch puzzle to make different levels of networks can learn different feature information. Methods [36, 34] rely on the attention mechanism to obtain more effective discrimination areas. The latest TransFG model [11] adds a Part Selection Module (PSM) to the Vision Transformer (ViT) [7] and applies it to fine-grained visual classification. The main fine-grained visual classification method is weakly supervised training based on image-level labeled data.

2.2 Channel attention

Vaswani et al. [28] proposed a self-attention mechanism in 2017. Then, Hu et al. [13] Proposed a Sequence-and-Excitation Network (SENet), first embedded into the Residual Networks (ResNet) [12] model. The channel attention method improves the accuracy of the convolutional neural network and further extracts the effective features. Since then, Park et al. [23] Combined channel attention with spatial attention and proposed Bottleneck Attention Module (BAM) [23] and Convolutional Block Attention Module (CBAM) [33] models. The former is the parallel channel and spatial attention structure, and the latter is the serial structure. Wang et al. [31] designed a more efficient channel attention module, which uses a one-dimensional convolution to connect the features, avoiding using the full connection to reduce the dimension and lose unnecessary features. At the same time, the one-dimensional convolution is used to interact with local channels, which effectively reduces the complexity of the model.

2.3 Dropout

Regularization methods for dropping neural units have been applied in deep convolutional networks. These methods are generally divided into randomly dropping out information and self-attention methods. The initial Dropout method [24] was to suppress neurons with a certain probability, and later methods were to delete the entire feature map or patch. For example, the Spatial Dropout method [25] deletes channels randomly, and the Cutout method [5] deletes patches randomly from the input image. Similarly, the ADCM method [21] uses the method of randomly dropping out channels and location information to improve the performance of the attention method. ADL [2] is an attention-based dropout method in which attention-based drop masks are applied to feature maps to mask most discriminatory components and promote networks to learn important features that are easy to distinguish. In addition, the Channel Drop Block method is to remove a similar set of related channels to break similarities between channels [6].

2.4 Transformer

Transformer was originally applied in natural language processing and text translation fields and has greatly promoted its development [4, 26, 3]. Recently, more and more models based on Vision Transformer have been widely used in other computer vision tasks. Researchers have improved the model's accuracy by improving self-attention mechanisms to detect discriminant regions automatically [36, 34] or to model parts [11, 30]. The representation of computer vision direction is pure Vision Transformer [7], Swin Transformer [20], etc. Vision Transformer is the first time it has been used in the field of vision, followed by local or global connections through different hierarchical networks to extract features. The first Transformer model in the fine-grained visual classification task is the TransFG [11] model, which improves the performance of pure Vision

Transformer models in fine-grained visual classification by adding the Part Selection Module and redesigning the loss function. In our work, we also add part modules to the pure Vision Transformer model and apply them to fine-grained visual classification tasks.

3 Method

In this chapter, we will better explain our approach. The steps to construct the input data are described in section 3.1. The entire PEDTrans model framework, the Patch Enhancement Module, and Module are shown in section 3.2, which uses three different dropout strategies.

3.1 Patch embedding

The input information of the Vision Transformer model is different from the traditional convolutional neural network because Transformer was originally used to solve text problems. Therefore we design image data as text vectors to be input into the Vision Transformer. The initial image area is marked $S = H * W$ and divided into patches of size $P * P$, so the number N of patches can be calculated from Eq. (1):

$$N = (H/P) * (W/P)\,. \tag{1}$$

3.2 PEDTrans

Generally speaking, the size of the input image is set to a square, $H = W$. Patches are projected into D-dimensional vector space by a learnable linear projection. Since the original image is divided into patches that cannot represent relative location information, a learning position vector E_{pos} is added to x_p. In addition, a token for classification is added before the first token converted from the patch. The final vector is like Eq. (2):

$$Z_0 = \left[x_{class}; E(x_p^1); \cdots ; E(x_p^N)\right] + E_{pos}. \tag{2}$$

where E is the patch embedding projection, $E \in R^{(D*p^2*c)}$, c is the number of channels of the original image, and E_{pos} represents the learning position embedding. The embedded image is transmitted to the interior of the Vision Transformer model. First, a Layer Normalization (LN) process is applied, which is then fed to Multi-head Self-attention (MSA) Module and Multi-layer Perceptron (MLP) Module. For better classification, such encoders are repeated L times and use a shortcut connection structure in the ResNet model. The output from Layer i can be calculated according to Eq. (4).

$$z_l^{'} = MSA\left(LN\left(z_{l-1}\right)\right) + z_{l-1}. \tag{3}$$

$$z_l = MLP\left(LN(\left(z_l^{'}\right)\right) + z_l^{'}. \tag{4}$$

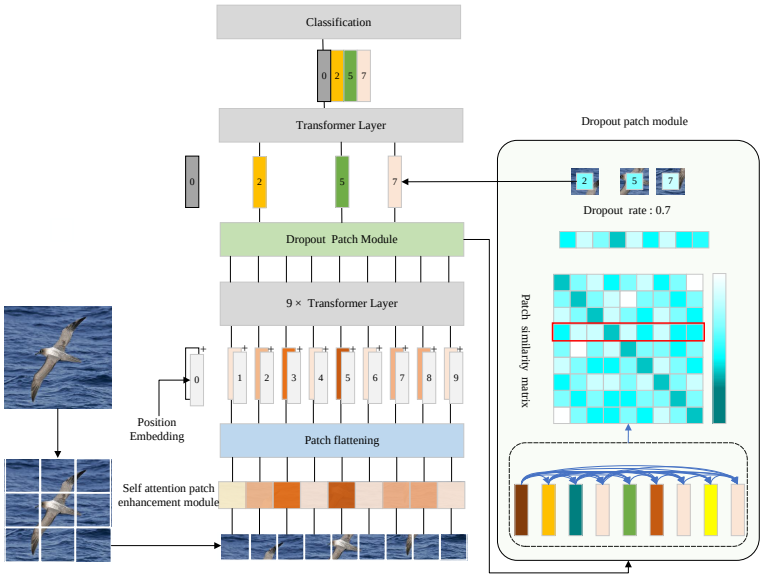

Fig. 2. PEDTrans: An image is divided into patches of the same size and enhanced by self-attention patches. Further, the linear projection embeds the patch into vector space, then combines it with the position embedding by addition. The Dropout Patch Module was designed to remove the duplicate and invalid patches before the last encoder to allow patch selection . The remaining patches are entered into the last encoder.

Following ViT, the original image is divided into N patches. The local relationship between patches helps the model select more effective patches. Although the patches add location information when embedded, the local patches' relationship is insufficient. We propose a Patch Enhancement Module for attention mechanism to enhance this local relationship. In addition, effective patches are a minority in a large number of patches. Redundant and duplicate patches can burden the classification results of models and mask patches that benefit classification. Our Dropout Patch Module can remove similar patches that are not good for classification. This section describes the details of our proposed PEDTrans. The framework of the whole model is shown in Fig. 2.

Patch Enhancement Module (PEM) In deep convolutional neural networks, the interaction between channels is very important. Different channels can get different weights through the attention mechanism, which enhances the characteristics of important channels. In Vision Transformer, patches are divided by the original image, ignoring the connections between local images. In addition, patch characteristics are critical to the final classification of the model, so using self-attention to enhance patch characteristics before converting patches to tokens increases the link between patches and makes important patches more effective in the final classification.

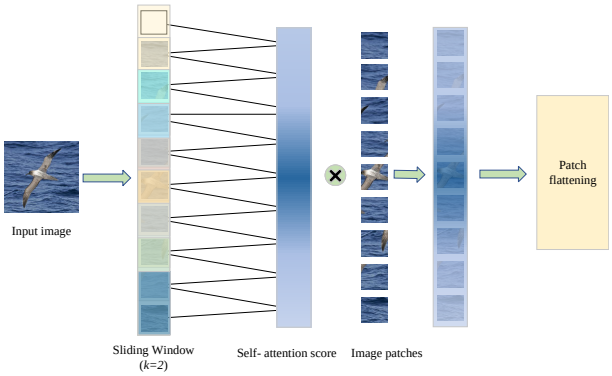

Fig. 3. This is a feature enhancement module with a sliding window length of 2. Each patch in the graph performs local information interaction through self-attention. Through this self-attention mechanism, different patches can obtain weights multiplied by the original feature map to obtain new enhanced feature patches.

We propose a sliding window to establish local information connections. The framework of this module is shown in Fig.3. Before the sliding window, we use global average pooling to process patches, and then use shared weights to learn the importance of local patches. Local attention weight is shown in Eq. (5).

$$\omega_i = s\left(\sum_{i=i-(k-1)/2}^{i+(k-1)/2} \omega_i * x_p^i\right). \tag{5}$$

$$x_p^i \in \Omega = \left\{x_p^{i-(k-1)/2}, \cdots, x_p^i, \cdots, x_p^{i+(k-1)/2}\right\}. \tag{6}$$

where ω_i is the shared weight, s is the $Softmax$ function, and the set Ω consists of k patches are connected before and after the i-th patch. This method can be realized by fast one-dimensional convolution with kernel k, as shown in Eq. (7) :

$$\omega_i = s\left(Conv1D_k(mean(x_p^i))\right). \tag{7}$$

where $Conv1D$ is a one-dimensional convolution, which can change the relationship between the upper and lower patches by adjusting the size of the k value to adjust the length of the sliding window. We do not use a structure similar to the ResNet model because we find that using a shortcut connection structure here does not increase performance very well. The general channel attention expression is represented by Eq. (8):

$$X_{out} = X_{in}^T W + X_{in}. \tag{8}$$

$$X_{out} = s\left(\frac{QK^T}{\sqrt{k}}\right) X_{in}. \tag{9}$$

However, we use a similar approach to self-attention in PEM to achieve feature enhancement, as shown in Eq. (9) where Q is obtained by the linear mapping of x^p and K is obtained by the global average pooling of x^p. k is the size of the sliding window. It can be understood more simply in Eq. (10).

$$X_{out} = X_{in}^T W. \tag{10}$$

More detailed results will be given in Chapter 4 of the ablation study and the reasons for the performance difference between the two will be analyzed.

Dropout Patch Module (DPM) This part contains three dropping strategies, namely, dropping out based on similarity, dropping out randomly and dropping out based on attention weight.

The Dropout Patch Module is implemented based on the correlation between patches to remove the most redundant patch and retain the most effective patches for final classification. We proposed that the Dropout Patch Module establishes a discarded patch combination by calculating the correlation coefficient matrix between patches. Fig.2 shows the details of the Dropout Patch Module in the PEDTrans module. Specifically, the input to Dropout Patch Module is a feature embedded $X_{in} \in R^{(N*P^2)}$ where N is the number of patches, and P^2 is the characteristic length of each embedded patch. Refer to Eq. (11) for the calculation process. We calculate the correlation matrix $M \in R^{(N*N)}$ between each patch, then drop out a patch by random selection and drop out patches that are too similar to the patch based on the calculated similarity matrix M and dropout rate γ.

The similarity matrix M of the Dropout Patch Module is calculated using the correlation measure in the bilinear pooling [18] algorithm, which measures the similarity between patches using the normalized cosine distance. Input features are first normalized, then similar matrices are constructed by matrix multiplication to obtain similar relationships between patches.

$$M = \mathcal{N}\left(X_{in}\right) \bullet \mathcal{N}\left(X_{in}^T\right) = \begin{bmatrix} 1 & M_{12} & \cdots & M_{1p} \\ M_{21} & 1 & \cdots & M_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ M_{p1} & M_{p2} & \cdots & 1 \end{bmatrix}. \tag{11}$$

In Eq. (11), $\mathcal{N}$ is a normalization function, and the similarity matrix M is a symmetric matrix. That is, the diagonal element $M_{ii} = 1$. M_{ij} indicates how similar the i-th patch is to the j-th patch.

The algorithm 1 description indicates that the i-th patch is randomly selected for dropout, and if $M_{ij} > min(A)$, the j-th patch will be dropout where A is the most similar set to patch i, and its size is $N * \gamma$. In other words, the similarity between every two patches will be arranged in descending order, leaving only patches that are similar to $m_{(N*\gamma)}$ where $m_{(N*\gamma)}$ is the $N * \gamma$-th similarity. Dropout Patch Module contains only one super parameter: γ, which controls the percentage of similar patches discarded.

Algorithm 1: Dropout Patch Module

Data: Patches feature map X_{in}; Dropout rate γ
Result: The index of the reserved patch
Computing the correlation matrix M;
Randomly select a patch i;
Dropout patch i;
while $j \leq N$ **do**
  if $M_{ij} > \min(A)$ **then**
    Dropout patch j;
  else
    Retain patch j;
  end
end

The above is the dropout strategy based on patch similarity. The following describes the random dropout strategy and the dropout strategy based on attention weight.

According to the description in algorithm 1, the random dropout strategy is to randomly select $N * \gamma$ patches from N patches and return the remaining indexes. The self-attention weight dropout strategy relies on the self-attention score of each layer of patches to drop out, discards the $N * \gamma$ patches with the smallest weight, and returns the reserved index.

4 Experiments

This chapter introduces the detailed settings of the experiment in section 4.1, including dataset, training hyperparameter settings, and hardware device information. Quantitative experimental analysis is given in section 4.2. Ablation study is performed in subsequent section 4.3. Qualitative analysis and experimental visualization are further given in section 4.4.

4.1 Experiments Setup

Datasets We tested the effectiveness of PEDTrans on several widely used fine-grained datasets, CUB-200- 2011 [29], Stanford Dogs [16], Stanford Cars [17], and NABirds [27] are datasets tested by general fine-grained visual algorithms to assess the performance of fine-grained visual algorithms very well, and they are challenging. Specific dataset information is shown in Table 1.

Experimental details We used most of the data enhancement methods in our experiments (clipping for training, central clipping for testing). First, we adjusted the image to $600 * 600$ and then clipped it to $448 * 448$. We use the pre-training model of the official ViT-B_16 in ImageNet21K. The parameters are set as follows: Batch size is 8, SGD optimizer is used, momentum is 0.9,

Table 1. Detailed statistics for CUB-200-2011, Stanford Dogs and Stanford Cars.

Datasets	Category	Training	Testing
CUB	200	5994	5794
Dog	120	12000	8580
Car	196	8144	8041
NABirds	555	23929	24633

learning rate is initialized to 0.02, cosine annealing is used as the scheduler of the optimizer. The loss function uses a cross-entropy function and contrast loss function. The Patch Enhancement Module sliding window super parameter k is 3, and the Dropout Patch Module dropout rate γ is 0.5. The experiments were trained with four Nvidia RTX 3080 GPUs using the deep learning framework Pytorch toolbox and Apex with FP16 training. In order to avoid the error caused by the experimental equipment, we conducted the experiment again according to the open source code and marked it with $*$.

Table 2. Comparison results of different methods on CUB-200- 2011, Stanford Dogs and Stanford Cars.

Method	Backbone	CUB	Cars	Dogs
RA-CNN [9]	VGG-19	85.3	92.5	87.3
MA-CNN [37]	VGG-19	86.5	92.8	-
P-CNN [10]	VGG-19	87.3	93.3	90.6
ResNet50 [12]	ResNet-50	84.4	-	-
Cross-X [22]	ResNet-50	87.7	94.6	88.9
PMG [8]	ResNet-50	89.6	<u>95.1</u>	-
FDL[19]	DenseNet-161	89.1	84.7	84.9
API-Net [38]	DenseNet-161	90.0	**95.3**	90.3
ViT [7]	ViT-B_16	90.7	93.7	91.7
FFTV * [30]	ViT-B_16	<u>91.4</u>	-	91.5
RAMS-Trans [14]	ViT-B_16	91.3	-	**92.4**
TransFG * [11]	ViT-B_16	91.3	94.8	<u>92.3</u>
PEDTans(ours)	ViT-B_16	**91.7**	<u>95.1</u>	**92.4**

4.2 Quantitative analysis

Our specific experimental results are shown in Table 2 and Table 3. PEDTrans achieved 91.7% accuracy on CUB datasets. It is 1.0% higher than the original Vision Transformer (ViT) model, and it does not use the overlapping strategy in the TransFG model to improve accuracy. The overlap strategy improves accuracy by increasing the number of patches. Our method is 0.3% more accurate than TransFG on the Stanford Cars dataset. It implements state-of-the-art (SOTA) on the data sets we tested and it is superior to most methods and has higher accuracy than the original Vision Transformer model. We obtained 92.4% accuracy on the Stanford dogs dataset, which is higher than 92.3% of the TransFG model and better than other convolutional neural networks. On the NABirds dataset, our model achieves an accuracy of 90.7%. In the experimental results, we use bold to represent the optimal results and underline to represent the suboptimal results.

Table 3. Comparison results of different methods on NABirds.

Method	Backbone	Acc
Cross-X [22]	ResNet-50	86.4
API-Net [38]	DenseNet-161	88.1
ViT [7]	ViT-B_16	89.3
TPSKG [14]	ViT-B_16	90.1
FFTV * [30]	ViT-B_16	89.5
TransFG * [11]	ViT-B_16	<u>90.2</u>
R^2-Trans [11]	ViT-B_16	<u>90.2</u>
PEDTans(ours)	ViT-B_16	**90.7**

4.3 Ablation study

PEM: We performed an ablation study on PEDTrans to see how our framework affects classification accuracy. All the experimental results are implemented on the CUB dataset. We test these framework parts, including the Patch Enhancement Module and Dropout Patch Module. The impact of the Patch Enhancement Module is shown in Table 4.

A quantitative comparison between adding PEM and deleting PEM shows that the model's accuracy is improved by adding PEM. Specifically, the accuracy of the ViT results increased from 90.7% to 91.2%. We think that building the correlation of local information can increase the interaction between patches, which will help the model increase the ability to select different patches, enhance the characteristics of some patches, and improve the model's accuracy.

Table 4. Ablation study on PEM and DPM on CUB-200-2011 dataset.

Datasets	Accuracy (%)
ViT	90.7
+PEM	91.2
+DPM	91.2
PEDTrans(ours)	91.7

Table 5. Ablation study on value of sliding window k on CUB-200-2011 dataset.

Value of k	Accuracy (%)
1	91.2
3	91.7
5	91.5
7	91.3

In the Patch Enhancement Module, we also tested the size of the sliding window and found that the best results at $k = 3$. The test results of sliding windows of different sizes are shown in Table 5. Sliding windows are too small to establish a good relationship between local patches and interact effectively, but too large windows can cause some valid patches to be masked by the best patches.

Table 6. Ablation study on shortcut connection on CUB-200-2011 dataset.

Method	Accuracy (%)
PEDTrans($XW^T + X$)	91.3
PEDTrans(XW^T)	91.7

We try to use shortcut connections in this section, but we can not get good results. Typically, channel attention is a shortcut connection after a series of feature extraction, but a multilayer convolution does not characterize our model. So, it is critical to multiply patches by the self-attention enhancement factor and get good results. The test results of these two connection modes are shown in Table 6.

DPM: We remove redundant patches from the last input level by adding a dropout patch module before the last encoder. The performance of ViT and ViT+DPM will be compared in this part, and compare three different dropout strategies in DPM. The test results are shown in Table 4. Specifically, the accuracy of ViT is 90.7%, and after adding DPM, the accuracy is improved by 0.5% to 91.2%. We believe that removing a large number of invalid, duplicate patches will make it easier for the model to obtain the most recognizable patches and thus improve the accuracy.

We also test the dropout rate γ in the dropout patch module with different 3 strategies and found that 50% dropout rate in DPM based on similarity is the best value, $\gamma = 0.5$. The input position information of the last encoder

is shown in Fig. 4. The random dropout strategy (Blue Border) can get more scattered patches, while the self-attention score strategy (Yellow Border) can get more focused patches centered on the classification object. The similarity based dropout strategy (Red Border) is between the two. This strategy can achieve the best results. It not only drops out part of the background, but also eliminates some unimportant prospects. It has a certain regularization effect. It is calculated based on similarity, so even if the first patch selected randomly selects the foreground object, the module will keep the parts with differences because the parts of the object are different. More detailed results are shown in Table 7. Discarding too many patches will easily lose categorized patches, but discarding too few patches will not highlight important ones.

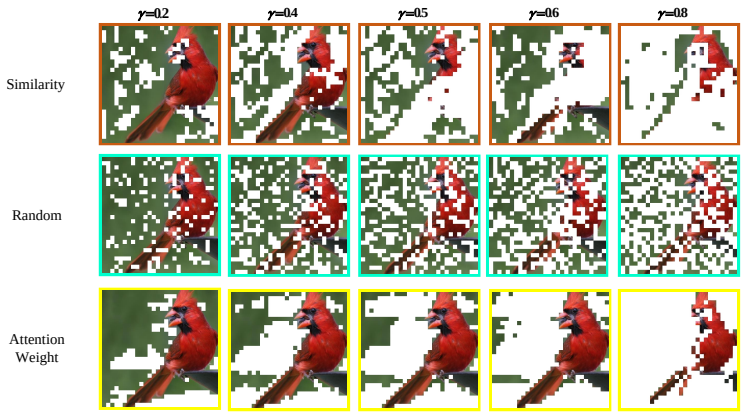

Fig. 4. Illustration of dropout rate. The visual images with different dropout rate are displayed, and the values of γ are 0.2, 0.4, 0.5, 0.6 and 0.8 respectively.

We only added DMP during training, with an accuracy of 91.3%. DPM was added during training and testing, and the accuracy was 91.7%. Its more important role is to select the discrimination region, which is perfectly combined with patch in ViT.

4.4 Analyze visualization results

We show the visualization of PEDTrans on four benchmarks in Fig. 5. We randomly selected two images from the four standard datasets we tested for visualization, and we drew an overall attention image based on the corresponding weight scales $(1, 2, 4, 8)$ of the last four layers of the network. The transparent white areas in the image are important areas from which we can see that our model can accurately capture the most easily distinguishable parts of the object under test, such as the mouth and eyes of a bird; The logo of the car, the lamp of the car, and the intake grille that BMW can recognize most easily; Dog's ears, eyes, etc.

Table 7. Ablation study on value of dropout rate γ on CUB-200-2011 dataset.

Value of γ	Similarity	Random	Attention-Weight
0.2	91.3	91.3	91.2
0.4	91.4	91.3	91.2
0.5	91.7	91.2	91.4
0.6	91.5	91.2	91.3
0.8	91.2	91.2	91.5

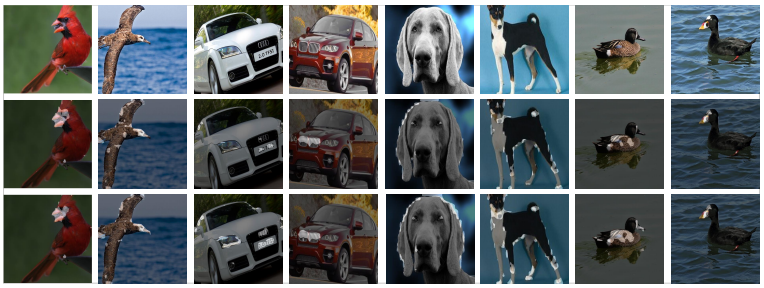

Fig. 5. Visualization results of PEDTrans in CUB-200-2011, Stanford Dogs, Stanford Cars and NABirds, where the first line are input images, while the second and third lines are partial attention maps generated by the ViT and PEDTrans. Best viewed in colour.

5 Conclusion

This work proposes a fine-grained visual classification network based on the Vision Transformer model and has achieved advanced results on standard fine-grained visual classification datasets. We build relationships between patches through self-attention mechanisms like channel interactions in convolutional neural networks. Through this connection, we enhanced patches that help distinguish between images. Fine-grained visual classification networks are most important in finding patches that are easy to distinguish between images, so the Vision Transformer model is more helpful in choosing really effective patches, but most of the small patches are redundant. Duplicate patches can be effectively removed and play a certain role in regularization through our proposed Drop Patches module. The final visualization fully demonstrates the validity of our proposed model.

Acknowledgement This work is partly supported by national precision agriculture application project (constructio number: JZNYYY001).

References

1. Chai, Y., Lempitsky, V., Zisserman, A.: Symbiotic segmentation and part localization for fine-grained categorization. Proceedings of the IEEE International Conference on Computer Vision pp. 321–328 (2013)
2. Choe, J., Shim, H.: Attention-based dropout layer for weakly supervised object localization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 2219–2228 (2019)
3. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R.: Transformer-XL: Attentive language models beyond a fixed-length context. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics pp. 2978–2988 (2019). https://doi.org/10.18653/v1/P19-1285, https://aclanthology.org/P19-1285
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the conference. Association for Computational Linguistics. Meeting pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
5. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
6. Ding, Y., Dong, S., Tong, Y., Ma, Z., Xiao, B., Ling, H.: Channel dropblock: An improved regularization method for fine-grained visual classification. arXiv preprint arXiv:2106.03432 (2021)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (2021)
8. Du, R., Chang, D., Bhunia, A.K., Xie, J., Ma, Z., Song, Y.Z., Guo, J.: Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. European Conference on Computer Vision pp. 153–168 (2020)
9. Fu, J., Zheng, H., Mei, T.: Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 4438–4446 (2017)
10. Han, J., Yao, X., Cheng, G., Feng, X., Xu, D.: P-cnn: Part-based convolutional neural networks for fine-grained visual categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(2), 579–590 (2022). https://doi.org/10.1109/TPAMI.2019.2933510
11. He, J., Chen, J.N., Liu, S., Kortylewski, A., Yang, C., Bai, Y., Wang, C., Yuille, A.: Transfg: A transformer architecture for fine-grained recognition. arXiv preprint arXiv:2103.07976 (2021)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 770–778 (2016)
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 7132–7141 (2018)
14. Hu, Y., Jin, X., Zhang, Y., Hong, H., Zhang, J., He, Y., Xue, H.: Rams-trans: Recurrent attention multi-scale transformer for fine-grained image recognition. Proceedings of the 29th ACM International Conference on Multimedia pp. 4239–4248 (2021)
15. Huang, S., Xu, Z., Tao, D., Zhang, Y.: Part-stacked cnn for fine-grained visual categorization. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1173–1182 (2016)

16. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford dogs. Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC) **2**(1) (2011)
17. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. Proceedings of the IEEE international conference on computer vision workshops pp. 554–561 (2013)
18. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. Proceedings of the IEEE international conference on computer vision pp. 1449–1457 (2015)
19. Liu, C., Xie, H., Zha, Z.J., Ma, L., Zhang, Y.: Filtration and distillation: Enhancing region attention for fine-grained visual categorization. Proceedings of the AAAI Conference on Artificial Intelligence **34**(7), 11555–11562 (2020)
20. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 10012–10022 (2021)
21. Liu, Z., Du, J., Wang, M., Ge, S.S.: Adcm: attention dropout convolutional module. Neurocomputing **394**, 95–104 (2020)
22. Luo, W., Yang, X., Mo, X., Lu, Y., Davis, L.S., Li, J., Yang, J., Lim, S.N.: Cross-x learning for fine-grained visual categorization. Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 8242–8251 (2019)
23. Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module. British Machine Vision Conference p. 147 (2018)
24. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
25. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 648–656 (2015)
26. Tsai, Y.H.H., Bai, S., Liang, P.P., Kolter, J.Z., Morency, L.P., Salakhutdinov, R.: Multimodal transformer for unaligned multimodal language sequences. Proceedings of the conference. Association for Computational Linguistics. Meeting p. 6558 (2019)
27. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 595–604 (2015). https://doi.org/10.1109/CVPR.2015.7298658
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
29. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
30. Wang, J., Yu, X., Gao, Y.: Feature fusion vision transformer for fine-grained visual categorization. arXiv preprint arXiv:2107.02341 (2021)
31. Wang, Q., Wu, B., Zhu, P., Li, P., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
32. Wei, X.S., Xie, C.W., Wu, J., Shen, C.: Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. Pattern Recognition **76**, 704–714 (2018)

33. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. Proceedings of the European conference on computer vision (ECCV) pp. 3–19 (2018)
34. Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., Zhang, Z.: The application of two-level attention models in deep convolutional neural network for fine-grained image classification. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 842–850 (2015)
35. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based r-cnns for fine-grained category detection. European conference on computer vision pp. 834–849 (2014)
36. Zhao, B., Wu, X., Feng, J., Peng, Q., Yan, S.: Diversified visual attention networks for fine-grained object classification. IEEE Transactions on Multimedia **19**(6), 1245–1256 (2017)
37. Zheng, H., Fu, J., Tao, M., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. 2017 IEEE International Conference on Computer Vision (ICCV) (2017)
38. Zhuang, P., Wang, Y., Qiao, Y.: Learning attentive pairwise interaction for fine-grained classification. Proceedings of the AAAI Conference on Artificial Intelligence **34**(07), 13130–13137 (2020)