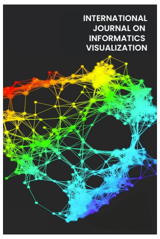
INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



Multi-Head Voting based on Kernel Filtering for Fine-grained Visual Classification

Mutiarahmi Khairunnisa [a,b], Suryo Adhi Wibowo [a,b,*]

[a] School of Electrical Engineering, Telkom University, Bandung, Indonesia
[b] Central of Excellence Artificial Intelligence for Learning and Optimization (CoE AILO), Telkom University, Bandung, Indonesia
Corresponding author: [*]suryoadhiwibowo@telkomuniversity.ac.id

***Abstract*—Research on Fine-Grained Visual Classification (FGVC) faces a significant challenge in distinguishing objects with subtle differences within intra-class variations and inter-class similarities, which are critical for accurate classification. To address this complexity, many advanced methods have been proposed using feature coding, part-based components for modification, and attention-based efforts to facilitate different classification phases. Vision Transformers (ViT) has recently emerged as a promising competitor compared to other complex methods in FGVC applications for image recognition, which are mainly capable of capturing more fine-grained details and subtle inter-class differences with higher accuracy. While these advances have shown improvements in various tasks, existing methods still suffer from inconsistent learning performance across heads and layers in the multi-head self-attention (MHSA) mechanisms that result in suboptimal classification task performance. To enhance the performance of ViT, we propose an innovative approach that modifies the convolutional kernel. Our method considerably improves the method's capacity to identify and highlight specific crucial characteristics required for classification by using an array of kernels. Experimental results show kernel sharpening outperforms other state-of-the-art approaches in improving accuracy across numerous datasets, including Oxford-IIIT Pet, CUB-200-2011, and Stanford Dogs. Our findings show that the suggested approach improves the method's overall performance in classification tasks by achieving more concentration and precision in recognizing discriminative areas inside pictures. Using kernel adjustments to improve Vision Transformers' ability to differentiate somewhat complicated visual features, our strategy offers a strong response to the problem of fine-grained categorization.**

***Keywords*— Fine-grained visual classification; vision transformer; multi-head self-attention.**

Manuscript received 30 Jul. 2024; revised 11 Sep. 2024; accepted 1 Dec. 2024. Date of publication 31 Mar. 2025.

I. Introduction

Due to the requirement for highly detailed object identification, computer vision research on fine-grained visual classification (FGVC) is quite interesting. However, improving performance in fine-grained visual classification remains a significant challenge. FGVC is more complex than coarse classification, as shown in Fig. 1 and Fig. 2. The main challenge lies in two factors: 1) the limitation in acquiring training data and 2) the existence of subtle differences between objects in the same class, while objects in different classes may have striking similarities. FGVC requires recognizing small, specific details of objects in images, which are often difficult to distinguish even by human observers due to their visual similarity, such as classifying various types of birds [1], cats, or dogs [2], [3]. This entails a good knowledge of their specific morphological and textural characteristics. However, different studies have attempted to construct more effective methods to address its complexity. The researchers have also adopted deep learning methods to improve accuracy in this task [4], [5], [6]. While some approaches based on CNN architecture perform reasonably well, they still have their shortcomings. The technique leads to a high computational cost and noisy outcome, especially as the number of networks used increases. There are now three primary types of approaches in use: attention-based methods [4], [12], feature encoding [10], [11], and part-based [7], [8], [9]. Part-based approaches identify discriminative sections of objects and categorize them, whereas feature encoding methods extract high-level features from images for recognition. Meanwhile, attention-based techniques employ attention processes to assess how vital specific object components are concerning one another.

Initially developed for word natural language processing (NLP), transformers [13] have also been modified for use in picture recognition software. Dosovitskiy et al. [14]

introduced Vision Transformer (ViT), a substantial modification of the transformer design for these tasks, which has proven effective in various object identification [15], segmentation [16] and classification task. ViT takes segments of image patches and transforms them into patch tokens. Like character sequences in NLP, these tokens are used in a multi-head self-attention mechanism during training. The self-attention mechanism is an appropriate strategy for FGVC as it effectively extracts and weights information from the full visual map for the classification token. Nonetheless, while applying the ViT method for FGVC, two primary problems need to be resolved. First, when processing all patch tokens at once, the ViT method might not be able to adequately draw attention to crucial locations in complicated datasets or images with crowded backgrounds. Second, ViT's receptive field extension is limited, which may cause the loss of locally significant information.

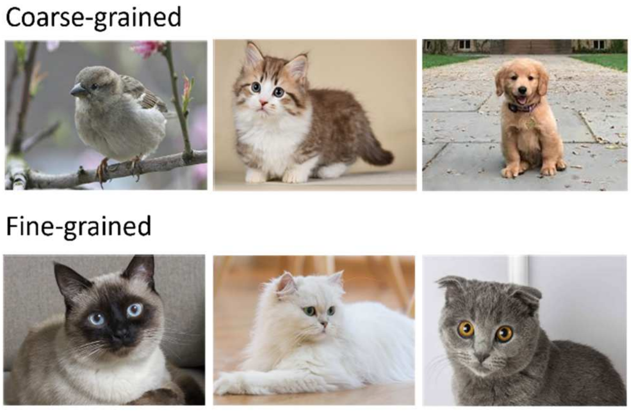


Fig. 1 Example of coarse-grained and fine-grained visual classification.

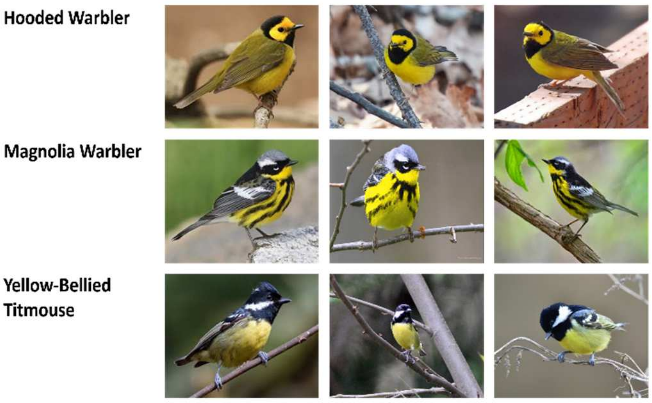


Fig. 2 These images illustrate the challenges of FGVC, where birds of the same species can differ in color variation and individuality (rows 1 and 2), while birds of different species can appear very similar (rows 2 and 3). One of the most important aspects of FGVC is the ability to distinguish bird species based on fine details in their patterns and colors.

In recent years, FGVC research employing ViT has concentrated on optimizing the ViT architecture to use local and global information most. To detect essential patch tokens, for example, TransFG [17] advocated using attention weights in the ViT method and multiplying them before going to the final transformer layer. While helpful, this method might not work well for complicated datasets or low-resolution images when combining particular tokens with the general categorization token.

Furthermore, TransFG uses the attention weights that ViT has built to try to remove extraneous inputs from the final transformer layer. Still, it does not entirely use the attention from all transformer levels. Zhang et al. [18] presented the AFTrans technique as a solution to this problem. This technique employs a Siamese design to offer a selective attention module with the same weight parameters. Nevertheless, this approach has limitations in that it takes attention away from highly identifiable local places, which leads to uncertainty in the training phase over the trustworthiness of the attention map. Mutual Attention Weight Selection (MAWS) is a token selection strategy that Wang et al. [19] suggested being used to choose the most informative tokens. FFVT aims to improve feature representation by merging information from several locations and levels in an image. However, applying fixed-size patches introduces noise, which makes the final class token emphasize global information instead of local features across layers. SIM-Trans [20] presented the Structure Information Learning (SIL) module. It leverages the Multilevel Feature Boosting (MFB) module with self-attention weights to enable contrastive learning and extract robust features. Xu et al. [21] introduced the Internal Ensemble Learning Transformer (IELT) to overcome the uneven learning performance in FGVC. This method selects necessary tokens, considers each center of attention a lousy learner, and assists in cross-layer feature learning. However, despite improving the model's ability to process image details, IELT still faces redundancy and noise problems, where irrelevant information reduces the model's efficiency.

In this paper, we propose a modified method by changing the convolutional kernel of the Internal Ensemble Learning Transformer (IELT). This change is motivated by previous research [21] to overcome the redundancy and noise problems and improve the method's capacity to identify good and essential characteristics for classification. We explored and analyzed various kernels to find the one that performs best.

This paper is structured as follows: Section II discusses related works, some literature on the Vision Transformer, and the adopted method. Section III presents experimental results and extensive analysis. Finally, the conclusion is in Section IV.

II. Materials and Method

A. Related Works

Fine-grained visual classification research focuses on two areas: local identification and global identification. Local identification selects essential parts of the object and creates intermediate-level representations for final classification [22]. Depending on how bounding box/part annotations are incorporated into the technique, the local identification method can be either strong or weak. Intense supervised learning requires part annotations [23], [24], while weak supervision only uses image labels [25], [26], [27], [28]. More recently, studies have concentrated on identifying discriminative areas and extracting features for more in-depth visual categorization [25], [29]. Nevertheless, an essential source of classification mistakes is the disregard for the holistic structural information of an item by many current approaches, which is crucial for precisely localizing the complete object.

In contrast, global discrimination methods employ specific distance metrics to learn deep feature embeddings for an entity. Other examples include bilinear methods [24] for learning interaction features between two independent Convolutional Neural Networks (CNNs). The process allows each global feature extracted from the whole input image to interact, helping learn better or more separable representations to perform fine-grained classification. Global methods prioritize a holistic understanding of objects or images and rarely involve precise localization steps. This differs from local identification methods, which focus on understanding the detailed parts of an object.

Recent research in FGVC has established Vision Transformers (ViT) as a front-runner technique through image segmentation and transformer architecture. While ViTs work well for various tasks, they fall short in capturing crucial local features necessary for in-depth classification. To overcome this constraint, various techniques have been devised, such as employing attention maps to mitigate background noise [17], integrating features from various layers using cross-layer filters [21], [19], and strengthening feature robustness through fusion techniques that employ graph networks and contrastive learning [20].

B. Vision Transformer Backbone

The vision transformer (ViT) is a computer vision method based on the Transformer method [14], which was initially designed for natural language processing (NLP). Patch embedding transforms the picture $x \in R^{H\times W\times C}$ into a 2D patch sequence $xp \in R^{N\times(P^2\cdot C)}$. The number of parts separated results in the Eq. (1):

$$N = \frac{HW}{P^2} \qquad (1)$$

where H is the image height, W is the image width and P^2 is the resolution of each image patch. This equation describes how the original image x is divided into smaller pieces (N patches). Each patch has a resolution of $(P \times P)$ and some channels (C). A sequence of xp The output of this operation is patches to be used in the following process. Patch embedding stores and input information into the transformer encoder in ViT using a trainable linear projection. Patch embeddings, such as cls tokens, are modified with learnable position embeddings to encode positional information. This approach is based on the BERT paper, which only utilizes the final representation associated with it (the output of the transformer L) in the classification layers. The procedure is shown in formula Eq. (2):

$$\boldsymbol{Z_0} = \left[\boldsymbol{x}_{\text{cls}}; \boldsymbol{x}_p^1\boldsymbol{E}; \boldsymbol{x}_p^2\boldsymbol{E}; \dots; \boldsymbol{x}_p^N\boldsymbol{E}\right] + \boldsymbol{E}_{\text{pos}} \qquad (2)$$

where $\boldsymbol{E} \in R^{(P^2.C)}$is the patch embedding projection, D is the token dimension, and $\boldsymbol{E}_{\text{pos}} \in R^{(N+1)\times D}$ is the position embedding. The transformer encoder receives the patch embeddings after this stage. The encoder transformer in ViT is made up of a normalizing layer, skipping connections between blocks, a multi-layer perceptron (MLP) block, and a multi-head self-attention block. The multi-head attention function is an essential part of the transformer encoder that allows it to store and process large volumes of data effectively. This technique allows the transformer encoder to recognize and emphasize significant picture characteristics. Additionally, the input from the previous stage is normalized using a normal distribution through layer normalization. The transformer encoder's MLP block consists of two wholly linked layers. The output of the l-th layer can be expressed as follows:

$$\boldsymbol{Z}_{\mathbf{l}}' = MHSA\big(LN(\boldsymbol{z}_{\mathbf{l-1}})\big) + \boldsymbol{z}_{\mathbf{l-1}} \qquad (3)$$

$$\boldsymbol{Z}_{\mathbf{l}}^{\boldsymbol{out}} = MLP\big(LN(\boldsymbol{z}_{\mathbf{l}})\big) + \boldsymbol{z}_{\mathbf{l}} \qquad (4)$$

Layer normalization (LN), multi-head self-attention (MHSA), and multi-layer perceptrons (MLP) are integral to the transformer-encoder process. The MLP head is responsible for the final categorization stage. This is completed at the MLP head layer, which produces the transformer encoder's output.

C. Multi-head Voting

The multi-head self-attention functions help to learn complicated correlations between characters in an input sequence efficiently. During multi-head self-attention, each attention 'head' learns its representation of the input and evaluates the relevance of each token concerning the others in the sequence. Each head in the multi-headed self-attention generates an attention map. This strategy allows the method to obtain more precise and complex information about the relationships between the tokens. However, the effectiveness of each attention head in identifying discriminative regions can exhibit variability. A novel technique introduced by Xu [21] is used to address this variability and enhance method reliability. Inspired by ensemble learning, notably the bagging algorithm [30], the method treats each attention head in a multi-head self-attention (MHSA) mechanism as a weak learner. The module aims to selectively collect tokens from various attention heads to improve the detection of unique areas inside each layer. Assume the $l-th$ layer of the transformer (where l is between 1 and $L-1$) has input and output tokens $\boldsymbol{Z}_{in}^{l}$ and $\boldsymbol{Z}_{out}^{l}$ respectively, we can refer to the collection of attention scores for class tokens as $\boldsymbol{G}$. $G^k \in \mathbb{R}^N$ represents the attention score of the $k-th$ head from the MHSA-generated attention map.

To select valuable tokens based on the attention score $\boldsymbol{G}^{k'}$, a score map $\boldsymbol{N}^k$ is generated using Eq. (5):

$$\boldsymbol{N}^k(i,j) = \begin{cases} 1, & if\ \boldsymbol{G}^k(i,j)\ is\ top-v\ value \\ 0, & otherwise \end{cases} \qquad (5)$$

where v is a hyperparameter that controls the number of votes per head. To produce the total score map $\boldsymbol{N}' \in \mathbb{R}^{n_1\times n_2}$, combine the score maps of each head as follows:

$$\boldsymbol{N}' = \sum_{k=0}^{K}\boldsymbol{N}^k \qquad (6)$$

A convolution kernel K is used to enhance discrimination on the total score map $\boldsymbol{N}'$. Eq. (7) determines the increased score map $\boldsymbol{N}^* \in \mathbb{R}^{n_1\times n_2}$:

$$\boldsymbol{N}^* = \boldsymbol{N}' * \boldsymbol{K_c} \qquad (7)$$

where $*$ denotes the convolution operator. In this study, we evaluated many convolutions kernel $\boldsymbol{K_c}$ for the best outcome**,** where c indicates kernel variations, is defined as follows:

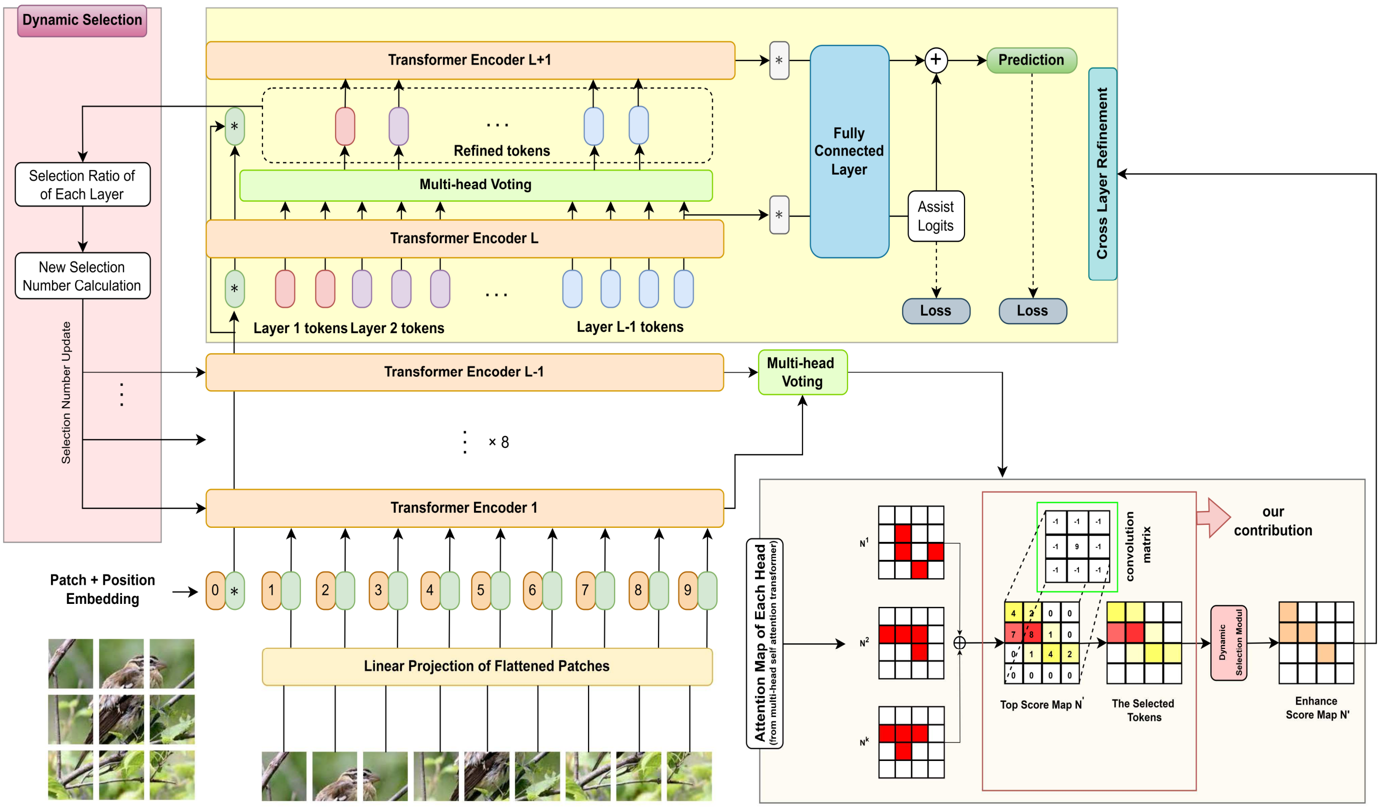


Fig. 3 Overview of the proposed innovative approach. In this approach, we alter the kernel (convolution matrix) within the multi-head voting module. The image depicts a detailed flow starting from patch and position embedding, leading through multiple transformer encoder layers with multi-head voting and dynamic selection to a fully connected layer for prediction.

1) Laplacian Kernel: The Laplacian kernel is a kernel used in image processing to detect edges and abrupt changes in picture intensity. This kernel is sometimes referred to as the Laplacian filter. The Laplacian kernel is typically represented mathematically as a 3 × 3 or 5 × 5 matrix, as shown in Eq. (8) and Eq. (9), respectively.

$$\boldsymbol{K_1} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{8}$$

$$\boldsymbol{K_2} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{9}$$

2) Modified Laplacian Kernel: This kernel, known as the modified Laplacian kernel, serves two functions. It has greater core value and negative values surrounding it. Primarily, it helps to reduce picture noise, resulting in cleaner and more accurate results. Second, this kernel enhances the image's edges. The positive value in the center highlights pixels with high intensity, while the negative values surrounding it assist decrease the impact of extremely sharp edges, resulting in smoother and more defined results. The kernel is often expressed as a 3 × 3 or 5 × 5 matrix, as shown in Eq. (10) and Eq. (11).

$$\boldsymbol{K_3} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{10}$$

$$\boldsymbol{K_4} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 2 & 16 & 2 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{11}$$

3) Box Linear Kernel: A box linear kernel is a spatial domain linear filter in which each pixel in the output image receives the average value of its nearby pixels in the input image. This approach works as a low-pass filter, producing a blurred image. Eq. (12) and Eq. (13) represent 3 × 3 and 5 × 5 box blurs, respectively.

$$\boldsymbol{K_5} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{12}$$

$$\boldsymbol{K_6} = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{13}$$

4) Sharpening Kernel: Sharpening kernels are used in image processing to improve picture sharpness. This kernel works by accentuating intensity changes surrounding each pixel in the image, making edges and details more visible and distinct. Sharpening kernels often have greater values in the center, serving as strong weights to highlight pixels with considerable intensity shifts. The negative numbers around it serve to limit noise effects and guarantee that the sharpening result isn't too sharp. Thus, sharpening kernels is an important tool in image processing for increasing picture clarity and detail. The kernel is often expressed as a 3 × 3 or 5 × 5 matrix, as shown in Eq. (14) and Eq. (15).

$$\boldsymbol{K_7} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{14}$$

$$\boldsymbol{K_8} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 25 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{15}$$

D. Cross Layer Refinement

The Cross-Layer Refinement (CLR) module is introduced by Xu [20]. The purpose of the CLR module is to enhance the utilization of cross-layer features by suppressing noise, thus improving the feature representation for final prediction. In this module, the process starts with the formation of input features ($\boldsymbol{Z}_L^{in}$), consisting of class tokens from the previous layer ($\boldsymbol{Z}_{L-1}^{class}$), and refined cross-layer feature tokens ($\boldsymbol{Z}_l^{sel}$), selected through the MHV and DS modules. The formation of the selected token sequence ($\boldsymbol{Z}_l^{sel}$) is based on their discriminative strength. This process is described by the Eq. (8):

$$\boldsymbol{Z}_L^{in} = [\boldsymbol{Z}_{L-1}^{class}\,;\, \boldsymbol{Z}_1^{sel};\, \boldsymbol{Z}_2^{sel};\, \dots;\, \boldsymbol{Z}_{L-1]}^{sel} \tag{16}$$

Equation 2 describes how the L-th layer's output ($\boldsymbol{Z}_L^{out}$) is obtained using established transformation processes. The cross-layer features are retrieved from the class tokens in this output, indicated by $\boldsymbol{Z}_L^{in} = \boldsymbol{Z}_{L,1}^{out}$. These features include valuable information from several layers. To get refined features, we process $\boldsymbol{Z}_L^{out}$ using the MHV module without size-altering operations or increased convolutions. Token indices are indicated as $in' \in \mathbb{R}^t$, where t is the number of refined tokens.

Furthermore, an additional transformer layer, the $(L + 1)$-th layer, is used to extract refined features from refined tokens. However, to minimize noise effects or quality degradation, the refined tokens are not immediately fed into the $(L - 1)$-th layer. Instead, the class tokens from the $(L - 1)$-th layer are used as inputs to the $(L + 1)$-th layer, rather than the class tokens from the L -th layer. Thus, the input to the $(L + 1)$-th layer represented by Eq. (17):

$$\boldsymbol{Z}_{L+1}^{in} = [\boldsymbol{Z}_{L-1}^{class}\,;\, \boldsymbol{Z}_{L,in'(1),:}^{out};\, \boldsymbol{Z}_{L,in'(2),:}^{out};\, \dots;\, \boldsymbol{Z}_{L,in'(t),:}^{out}] \tag{17}$$

Here, $\boldsymbol{Z}_{L+1}^{in}$ combines class tokens from the $(L - 1)$-th layer with refined tokens, resulting in cleaner and more informative input for the following layer.

To improve the final prediction results, a logit assistance operation is utilized, which takes use of earlier predictions. The preceding prediction results, $\mathbf{p} \in \mathbb{R}^c$, are computed based on cross-layer features as input to the Fully Connected (FC) layer, followed by a SoftMax operation, as indicated by Eq. (18):

$$\mathbf{p} = \text{softmax}\,(\text{FC}\left(\text{LN}(\boldsymbol{Z}_L^{class})\right)) \tag{18}$$

where C denotes the number of categories. To calculate the cross-layer logits $\mathbf{y} \in \mathbb{R}^c$, previously predicted results $\mathbf{p}$ and weights W are used in the FC layer, as given in Eq. (19):

$$y = \mathbf{p} \odot \sum_{\boldsymbol{i=0}}^{D} \mathrm{w}^i \tag{19}$$

The symbol $\odot$ represents a Hadamard operation, which is defined as element-by-element multiplication. Then, the final prediction, denoted by $\mathbf{p}$, is calculated by combining the cross-layer logits, represented by y, and the corrected features, represented by $\boldsymbol{Z}_{L+1}^{class}$ as follows:

$$\mathbf{p}' = \text{softmax}\,(\text{FC}\left(\text{LN}(\boldsymbol{Z}_L^{class})\right) + \boldsymbol{y}) \tag{20}$$

During training, the cross-entropy loss function is used for $\mathbf{p}$ and $\mathbf{p}'$, which are modified using the ground-truth labels $\mathbf{z}$ and balance parameters. The loss function L is defined as follows:

$$L = \lambda \text{CrossEntropy}(\mathbf{p}, \mathbf{z}) + (1 - \lambda)\text{CrossEntropy}(\mathbf{p}';\, \mathbf{z}) \tag{21}$$

The CLR module's integration of cross-layer and refined features helps to minimize noise and improve feature representation capabilities for efficient classification.

E. Dynamic Selection

Inspired by the boosting algorithm, Xu [20] proposed a dynamic selection (DS) module, using each transformer layer as a "weak learner". DS module is a crucial part of the Internal Ensemble Learning Transformer (IELT). It controls which tokens to keep from each transformer layer considering importance of that layer for final feature quality. In DS module, the contribution of each layer to the final prediction is determined by comparing the number of tokens chosen from each layer in the CLR module. This module starts by calculating the contribution of each layer with comparing the number of tokens selected in the Cross-Layer Refinement (CLR) module. For the l-th layer, the number of selected tokens is recorded in the vector $q(l)$, which represents the contribution of each layer to the refined feature.

$$r'(l) = \frac{q(l)}{t} \tag{22}$$

where $r'(l)$ is the incremental selection ratio for the l-th layer, $q(l)$, is the number of tokens selected from that layer, and t is the total number of tokens selected across all layers. Layers that make a large contribution will have a higher selection ratio, while layers that make a small contribution will have a lower selection ratio.

To determine the contribution of each layer, the selection ratio is updated based on how much the layer contributes to generating useful tokens. The selection ratio for the l-th layer (where $l \in 1, 2, 3, \dots, L - 1$) is calculated by updating the previous selection ratio $r(l)$ using the latest contribution $r'(l)$ with the following Eq. (23):

$$r(l) \leftarrow (1 - \theta)r(l) + \theta r'(l) \tag{23}$$

where $r(l)$ is the selection ratio of the l-th layer before it is updated, $r'(l)$ is the newly calculated contribution ratio of the tokens selected from that layer, θ is the moving rate that governs how much the selection ratio is updated. The total number of tokens selected from each layer, $\mathbf{m}(l)$, is then calculated using the updated selection ratio:

$$\mathbf{m}(l) = s.\mathbf{r}(l) \tag{24}$$

where s is the total number of tokens to be selected from all layers. The tokens selected from each layer are based on a certain interval. This interval ensures that each layer selects a specific number of tokens and that there is no overlap between layers. This interval is determined by the initial index $\mathbf{a}(l)$ and the final index $b(l)$, which is calculated by the formula:

$$\mathbf{a}(l) = \begin{cases} 0, & l = 1 \\ \sum_{i=1}^{l-1} \boldsymbol{m}(i), & l > 1 \end{cases} \tag{25}$$

$$\mathbf{b}(l) = \mathbf{a}(l) + \boldsymbol{m}(l) \tag{26}$$

where $\mathbf{a}(l)$ is defined as 0 for $l = 1$, or as the number of tokens from the previous layer for $l > 1$, while $\mathbf{b}(l)$ is the sum of $\mathbf{a}(l)$ and $\mathbf{m}(l)$, with l being in the range of 1 to $(L - 1)$.

With dynamic selection, the system automatically adjusts the number of tokens selected from better-performing layers, improving the resulting feature representation while reducing the influence of noise or less relevant features.

III. Result and Discussion

A. Datasets

The fine-grained benchmark used in this study consists of three datasets: CUB-200-2011, Stanford Dogs, and Oxford-IIIT Pet. These three datasets refer to datasets commonly used in testing fine-grained classification algorithms and datasets utilized in the IELT approach [21]. Furthermore, all three datasets provide a range of visual challenges that may be used to assess the system's robustness, ultimately leading to a more precise and reliable visual method. CUB-200-2011, a fine-grained dataset created exclusively for bird classification, includes not only bird labels but also bounding boxes and part annotations, which are critical for accurate classification. The Stanford Dogs dataset contains images of 120 dog breeds, including 12,000 training and 8,580 testing images. The Oxford-IIIT Pet dataset comprises images of cats and dogs from 37 distinct breeds, with around 200 images per class. Table I includes detailed information on these datasets.

TABLE I
Fine-Grained Dataset

Dataset	Class	Training	Testing
CUB-200-2011 [1]	200	5994	5794
Stanford Dogs [2]	120	12000	8580
Oxford-IIIT Pet [3]	37	3680	3669

B. Implementation Details

The image was resized to 448 × 448 pixels using the ViT-B-16 backbone network, which was pretrained on the ImageNet21K dataset. The image underwent random cropping, horizontal flipping, and color adjustments for training, while central cropping was used for testing. The method was optimized using stochastic gradient descent (SGD) with a momentum of 0.9 and cosine annealing for learning rate scheduling. The initial learning rate was set to 0.002 for the Stanford Dogs dataset and 0.02 for the other three datasets. The training method spanned 50 epochs with a batch size of 8 for all datasets. Implementation was carried out using PyTorch on an NVIDIA DGX100 server, with top-1 accuracy as the evaluation metric for all experiments. The experiment conditions for the IELT approach remain the same as in the original paper [21], to observe any changes in the outcomes that arise from testing with various kernel modifications.

The MHV module sets 24 as the maximum number of votes for each head. The loss proportion is set to 0.4, and the number of upgraded tokens is 24 in the CLR module. The DS module's selection ratio per layer is set initially at $1/(L-1)$ and the total number of selections is set at 126. Because of the domain gap between the training dataset and the more specialized dataset, the DS module was not employed during the first 10 epochs, which made low-level characteristics more helpful for classification.

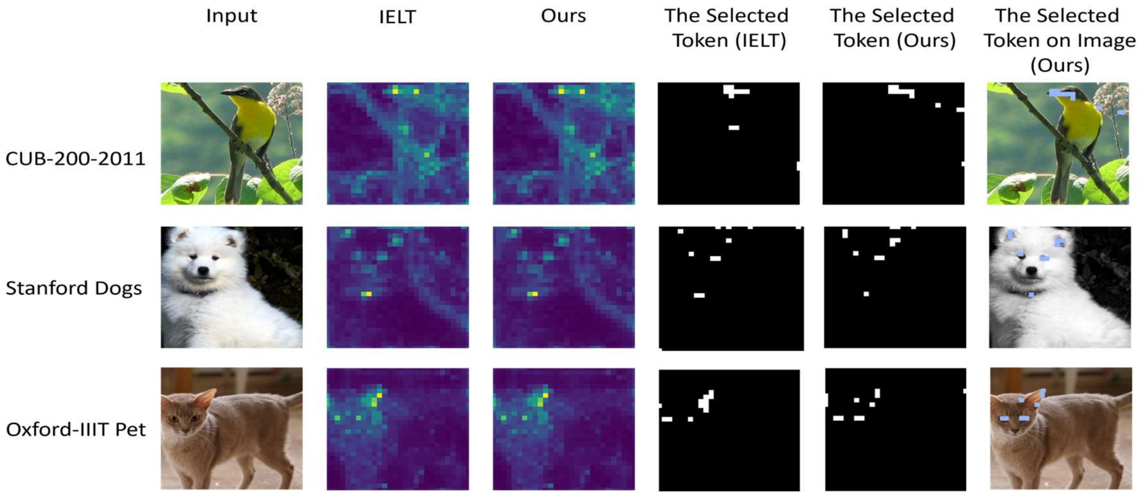

Input
IELT
Ours
The Selected Token (IELT)
The Selected Token (Ours)
The Selected Token on Image (Ours)
CUB-200-2011
Stanford Dogs
Oxford-IIIT Pet

Fig. 4 Visualization results from our method on each dataset. The first column shows the input image. The second and third columns display the attention maps generated by [20] and the modified kernel we proposed, respectively. The fourth and fifth columns show the tokens selected by [20] and the modified kernel we proposed, respectively. The sixth column indicates the locations where the MHV module selects tokens in the image.

C. Comparison of the Type of Enhanced Convolution Kernel

This study conducted experiments using various convolution kernels in the MHV module. Table II shows the experimental results using several kernels tested on the Stanford Dogs dataset. The experimental results show that using convolution kernels with different types and sizes also gives different classification accuracy results. For instance, the 3 × 3 Gaussian kernel gives an accuracy of 91.818%, while the 5 × 5 shows a slightly lower accuracy of 91.770%. This suggests that using a Gaussian kernel with a smaller size is more effective in improving classification accuracy on this dataset.

Based on Table II, it can also be seen that some kernel types, such as Laplacian, Sharpening, and Modified Laplacian, managed to achieve higher accuracy than other kernels. Compared to the Gaussian kernel, using the sharpening kernel here resulted in superior accuracy by 0.213% and 0.165%, reaching 92.031% for the 3 × 3 size and 91.935% for the 5 × 5 size. This shows that the sharpening kernel's ability to improve image clarity and contrast significantly contributed to the improved performance and made it easier for the method to distinguish features important for classification.

TABLE II
COMPARISON RESULT ON THE TYPE OF ENHANCED CONVOLUTION KERNEL ON STANFORD DOGS DATASET

Kernel Type	Kernel Size	Accuracy (%)
Gauss-like [21]	3 × 3	91.818
	5 × 5	91.770
Laplacian	3 × 3	91.958
	5 × 5	91.963
Box Linear	3 × 3	91.946
	5 × 5	91.923
Sharpening	3 × 3	**92.031**
	5 × 5	91.935
Modified Laplacian	3 × 3	91.923
	5 × 5	91.872

Thus, methods trained using kernel sharpening have achieved higher classification accuracy, as they can better spot important patterns or object attributes in the picture. The improvement in accuracy achieved by kernel sharpening corroborates the effectiveness of this approach, reinforcing the method's ability to deal with spatially complex image perturbations and reducing the effect of noise encountered frequently in images. Therefore, the MHV module is more effective when kernel sharpening is used as its convolution kernel.

D. Comparison with the State-Of-The-Art

In this section, we use the IELT method with a 3×3 sharpening kernel based on the accuracy comparison results in Table II.

TABLE III
COMPARISON RESULT ON CUB-200-2011 DATASET

Method	Backbone	Accuracy (%)
ResNet-50 [31]	ResNet-50	84.5
DCL [32]	ResNet-50	87.8
GaRD [33]	ResNet-50	89.6
StackedLSTM [27]	GoogleNet	90.4
CAL [4]	ResNet-101	90.6
ViT [14]	ViT-B_16	91.0
AFTrans [18]	ViT-B_16	91.5
FFVT [19]	ViT-B_16	91.6
TransFG [17]	ViT-B_16	91.7
SIM-Trans [20]	ViT-B_16	91.8
HAVT [34]	ViT-B_16	91.8
MP-FGVC [35]	ViT-B_16	91.8
IELT [21]	ViT-B_16	91.8
KR-MHV **(ours)**	ViT-B_16	**91.9**

TABLE IV
COMPARISON RESULT ON STANFORD DOGS DATASET

Method	Backbone	Accuracy (%)
ResNet-50 [31]	ResNet-50	82.7
FDL [6]	DenseNet-161	84.9
API-Net [36]	ResNet-101	90.3
ViT [14]	ViT-B_16	90.2
TransFG [17]	ViT-B_16	90.6
HAVT [34]	ViT-B_16	91.0
MP-FGVC [35]	ViT-B_16	91.0
FFVT [19]	ViT-B_16	91.5
IELT [21]	ViT-B_16	91.8
KR-MHV **(ours)**	ViT-B_16	**92.0**

TABLE V
COMPARISON RESULT ON OXFORD-IIIT PET DATASET

Method	Backbone	Accuracy (%)
SEER [37]	RG-10B	85.3
NAC [38]	VGG-19	93.8
OPAM [39]	VGG-19	93.8
VIT [14]	ViT-B_16	93.8
CvT [40]	ViT-B_16	94.7
TNT-B [41]	ViT-B_16	95.0
Bamboo [42]	ViT-B_16	95.1
IELT [21]	ViT-B_16	95.2
KR-MHV **(ours)**	ViT-B_16	**95.4**

1) Result on CUB-200-2011: The results of comparing the classification accuracy of the different techniques applied to the CUB-200-2011 dataset are shown in Table III. The data clearly shows that the suggested approach outperforms existing state-of-the-art (SOTA) techniques in performance. With kernel sharpening, the proposed approach obtains the most remarkable accuracy of 91.9%. This is a noteworthy 0.9% improvement over the Vision Transformer (VIT) approach. Compared to TransFG, FFVT, and AFTrans, our method's accuracy improvement is 0.2%, 0.3%, and 0.4%, respectively. Furthermore, it is 0.1% higher than the HAVT, MP-FGVC, and SIM-Trans approaches. This increase is comparable to other sophisticated methods, even though the accuracy gains over the approach given in reference [20] is just 0.1%. Improvements were achieved only by adjusting the convolution kernel (the whole method architecture remained unchanged). This implies that little improvements, like fine-tuning the convolutional kernels, might significantly influence the method's performance without requiring significant structural adjustments. On the CUB-200-2011 dataset, the kernel sharpening method helps improve classification accuracy. Kernel sharpening enables the technique to produce more accurate predictions by clarifying and refining the features created by convolution processes. This indicates that on the CUB-200-2011 dataset, applying kernel sharpening methods improves classification accuracy and enhances method performance.

2) Result on Stanford Dogs: The KR-MHV method outperformed the reference [21] method by 0.2%, achieving the most remarkable accuracy of 92.0% on the Stanford Dogs dataset. The KR-MHV method exhibits significant performance increases over other top methods, indicating its supremacy in fine-grained image categorization. In particular, the KR-MHV method shows a 0.5% increase in accuracy compared to the FFVT method, which had the most fantastic accuracy at 91.5%. In the same way, the KR-MHV method exhibits a noteworthy 1.4% accuracy gain over the TransFG method, which attained a 90.6% accuracy. Additionally, KR-MHV performs 1.7% better than the API-Net method and 1.0% better than the HAVT and MP-FGVC methods, each with an accuracy of 91.0%. These results show the KR-MHV method's improved performance in fine-grained image classification and highlight its cutting-edge methods, including kernel sharpening, which support its increased accuracy.

3) Results on Oxford-IIIT: A comprehensive analysis of the classification accuracy results obtained from applying different methods on the Oxford-IIIT Pet dataset is presented in Table V. It is evident from the results that other state-of-the-art (SOTA) methods were not as successful on this dataset as the KR-MHV method, which achieved the highest accuracy of 95.4%. This remarkable performance underscores the effectiveness of the KR-MHV method in improving classification performance on the Oxford-IIIT Pet dataset. This notable achievement underscores the efficacy of the KR-MHV approach in enhancing classification performance. The suggested approach shows a 0.2% improvement over KR-MHV compared to the IELT method, which obtained accuracy somewhat lower than KR-MHV's. Even though this difference might not seem like much, it matters in the context of high-performance methods since little improvements can have a significant impact. Our method's accuracy improvement over Bamboo, TNT-B, and SEER is 0.3%, 0.4%, and 10.1%, respectively. Moreover, the KR-MHV method achieves an astounding 1.5% greater accuracy, demonstrating a considerable performance advantage over the NAC, OPAM, and ViT methods. This difference highlights the KR-MHV approach's efficacy and kernel sharpening methods, which help explain why it performed so well in the classification challenge.

E. Visualization

The visualization outcomes of many distinct methods are shown in Fig. 4, which has five primary columns. The original image that served as the method's input is shown in the first column. The attention map produced by the reference approach [21] is displayed in the second column. It highlights the regions of the image that the baseline method deemed significant for prediction purposes. The attention map produced by the modified approach is shown in the third column, with more focus on discriminative regions. The voting selection results from the MHV-IELT [21] and our KR-MHV modules are displayed in the fourth and fifth columns. These pictures show variations in the tokens chosen: the KR-MHV approach chose fewer tokens that are more concentrated on significant features.

The KR-MHV selection results on the image are shown in the fifth column. These three results can be compared to identify many significant features. The attention map produced by the approach from [21] is not as concentrated, which implies that the baseline method could have considered less critical information or noise in the image. On the other hand, the attention map generated by the KR-MHV method is more defined and concentrated. This suggests that KR-MHV is more successful in focusing on regions crucial for categorization. With more accurate and concentrated attention mappings on discriminative regions, our suggested KR-MHV method outperforms the reference method. More relevant tokens can be chosen by the KR-MHV module for classification, improving prediction accuracy and dependability. Overall, the KR-MHV approach (our approach) results show a more concentrated attention map on significant regions within the image when using our proposed method, which is based on the reference method [21].

IV. Conclusion

This study shows that by changing the convolution kernel on the multi-head voting module, the Internal Ensemble Learning Transformer (IELT) method can improve the performance of the Fine-Grained Visual Classification (FGVC) task. In particular, kernel sharpening can improve classification accuracy for the FGVC task. In some data sets, the proposed KR-MHV method outperforms other innovative methods while obtaining the highest accuracy. However, compared to the original method, the accuracy improvement is small due to only the modification of the convolution kernel. The visualization results demonstrate how KR-MHV can improve prediction reliability, minimize noise, and focus attention on critical discriminative regions. This approach improves accuracy and strengthens the method's ability to handle complex image variations. To evaluate the generality and transferability of the proposed approach, we will conduct more tests on various datasets and explore the integration of advanced combining strategies with other techniques in FGVC in future research. To enhance training methods and overall interpretability, a better knowledge of the method's mechanism and internal representation will be obtained through the extension of the visualization analysis.

Acknowledgment

This work was supported by the RIIM Batch 3 program grant funded by LPDP, Ministry of Finance, Rep. of. Indonesia, and BRIN (Grant No. 81/IV/KS/05/2023) and was supported by Telkom University.

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," *Tech. Rep. CNS-TR-2011-001, Calif. Inst. Technol.*, 2011.

[2] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, 2011.

[3] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V Jawahar, "Cats and dogs," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3498–3505. doi:10.1109/CVPR.2012.6248092.

[4] Y. Rao, G. Chen, J. Lu, and J. Zhou, "Counterfactual Attention Learning for Fine-Grained Visual Categorization and Re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Aug. 2021, pp. 1025–1034. doi:10.1109/ICCV48922.2021.00106.

[5] J. Song and R. Yang, “Feature Boosting, Suppression, and Diversification for Fine-Grained Visual Classification,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8. doi: 10.1109/ijcnn52387.2021.9534004.

[6] C. Liu, H. Xie, Z.-J. Zha, L. Ma, L. Yu, and Y. Zhang, “Filtration and distillation: Enhancing region attention for fine-grained visual categorization,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 11555–11562.

[7] S. Yang, S. Liu, C. Yang, and C. Wang, “Re-rank Coarse Classification with Local Region Enhanced Features for Fine-Grained Image Recognition,” *arXiv Prepr.*, Feb. 2021, [Online]. Available: http://arxiv.org/abs/2102.09875

[8] M. Liu, C. Zhang, H. Bai, R. Zhang, and Y. Zhao, “Cross-Part Learning for Fine-Grained Image Classification,” *IEEE Trans. Image Process.*, vol. 31, pp. 748–758, 2022, doi: 10.1109/TIP.2021.3135477.

[9] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, “P-CNN: Part-Based Convolutional Neural Networks for Fine-Grained Visual Categorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 579–590, Feb. 2022, doi: 10.1109/tpami.2019.2933510.

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

[11] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, “Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition,” *Computer Vision – ECCV 2018*, pp. 595–610, 2018, doi: 10.1007/978-3-030-01270-0_35.

[12] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, and J. Jiao, “Selective Sparse Sampling for Fine-Grained Image Recognition,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, doi:10.1109/iccv.2019.00670.

[13] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16×16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2021. [Online]. Available: https://arxiv.org/abs/2010.11929.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[16] S. Zheng et al., “Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, doi: 10.1109/cvpr46437.2021.00681.

[17] J. He et al., “TransFG: A Transformer Architecture for Fine-Grained Recognition,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, pp. 852–860, Jun. 2022, doi:10.1609/aaai.v36i1.19967.

[18] Y. Zhang et al., “A free lunch from ViT: adaptive attention multi-scale fusion Transformer for fine-grained visual recognition,” *ICASSP 2022 -2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3234–3238, May 2022, doi:10.1109/icassp43922.2022.9747591.

[19] J. Wang, X. Yu, and Y. Gao, “Feature fusion vision transformer for fine-grained visual categorization,” *arXiv preprint arXiv:2107.02341*, 2022. [Online]. Available: https://arxiv.org/abs/2107.02341

[20] H. Sun, X. He, and Y. Peng, “SIM-Trans: Structure Information Modeling Transformer for Fine-grained Visual Categorization,” Proceedings of the 30th ACM International Conference on Multimedia, Oct. 2022, doi: 10.1145/3503161.3548308.

[21] Q. Xu, J. Wang, B. Jiang, and B. Luo, “Fine-Grained Visual Classification via Internal Ensemble Learning Transformer,” *IEEE Trans. Multimed.*, vol. 25, pp. 9015–9028, 2023, doi:10.1109/TMM.2023.3244340.

[22] X. Liu, L. Wang, and X. Han, “Transformer with peak suppression and knowledge guidance for fine-grained image recognition,” *Neurocomputing*, vol. 492, pp. 137–149, Jul. 2022, doi:10.1016/j.neucom.2022.04.037.

[23] S. Huang, Z. Xu, D. Tao, and Y. Zhang, “Part-Stacked CNN for Fine-Grained Visual Categorization,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1173–1182, Jun. 2016, doi: 10.1109/cvpr.2016.132.

[24] J. Donahue *et al.*, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*, PMLR, 2014, pp. 647–655.

[25] X. He, Y. Peng, and J. Zhao, “Fast Fine-Grained Image Classification via Weakly Supervised Discriminative Localization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1394–1407, May 2019, doi: 10.1109/TCSVT.2018.2834480.

[26] J. Wang, N. Li, Z. Luo, Z. Zhong, and S. Li, “High-Order-Interaction for weakly supervised Fine-Grained Visual Categorization,” *Neurocomputing*, vol. 464, pp. 27–36, 2021, doi:10.1016/j.neucom.2021.08.108.

[27] W. Ge, X. Lin, and Y. Yu, “Weakly Supervised Complementary Parts Models for Fine-Grained Image Classification From the Bottom Up,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3029–3038, Jun. 2019, doi:10.1109/cvpr.2019.00315.

[28] S. Jiang, W. Min, L. Liu, and Z. Luo, “Multi-Scale Multi-View Deep Feature Aggregation for Food Recognition,” *IEEE Trans. Image Process.*, vol. 29, pp. 265–276, 2020, doi: 10.1109/TIP.2019.2929447.

[29] X. He, Y. Peng, and J. Zhao, “Which and How Many Regions to Gaze: Focus Discriminative Regions for Fine-Grained Visual Categorization,” *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1235–1255, Sep. 2019, doi: 10.1007/s11263-019-01176-2.

[30] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/bf00058655.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, doi: 10.1109/cvpr.2016.90.

[32] Y. Chen, Y. Bai, W. Zhang, and T. Mei, “Destruction and construction learning for fine-grained image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5157–5166. doi: 10.1109/cvpr.2019.00530.

[33] Y. Zhao, K. Yan, F. Huang, and J. Li, “Graph-based high-order relation discovery for fine-grained recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15079–15088.

[34] X. Hu, S. Zhu, and T. Peng, “Hierarchical attention vision transformer for fine-grained visual classification,” Journal of Visual Communication and Image Representation, vol. 91, p. 103755, Mar. 2023, doi: 10.1016/j.jvcir.2023.103755.

[35] X. Jiang, H. Tang, J. Gao, X. Du, S. He, and Z. Li, “Delving into multimodal prompting for fine-grained visual classification,” in *Proceedings of the AAAI conference on artificial intelligence*, 2024, pp. 2570–2578.

[36] P. Zhuang, Y. Wang, and Y. Qiao, “Learning attentive pairwise interaction for fine-grained classification,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 13130–13137.

[37] P. Goyal, Q. Duval, I. Seessel, M. Caron, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski, “Vision models are more robust and fair when pretrained on uncurated images without supervision,” *arXiv preprint arXiv:2202.08360*, 2022. [Online]. Available: https://arxiv.org/abs/2202.08360.

[38] M. Simon and E. Rodner, “Neural activation constellations: Unsupervised part model discovery with convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015.

[39] Y. Peng, X. He, and J. Zhao, “Object-Part Attention Method for Fine-Grained Image Classification,” *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1487–1500, 2018, doi: 10.1109/TIP.2017.2774041.

[40] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “CvT: Introducing convolutions to vision transformers,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[41] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in transformer,” in *Advances in Neural Information Processing Systems*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, Eds. Curran Associates, Inc., 2021, pp. 15908–15919.

[42] Y. Zhang, Q. Sun, Y. Zhou, Z. He, Z. Yin, K. Wang, L. Sheng, Y. Qiao, J. Shao, and Z. Liu, “Bamboo: Building mega-scale vision dataset continually with human-machine synergy,” *arXiv preprint arXiv:2203.07845*, 2022. [Online]. Available: https://arxiv.org/abs/2203.07845