This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Building Vision Transformers with Hierarchy Aware Feature Aggregation

Yongjie Chen[1,2] Hongmin Liu[1,2*] Haoran Yin[3] Bin Fan[1,2]

[1]School of Intelligence Science and Technology, University of Science and Technology Beijing
[2]Institute of Artificial Intelligence, University of Science and Technology Beijing
[3]Horizon Robotics

yongjie_chen@xs.ustb.edu.cn, hmliu_82@163.com, haoran.yin@horizon.ai, bin.fan@ieee.org

Abstract

Thanks to the excellent global modeling capability of attention mechanisms, the Vision Transformer has achieved better results than ConvNet in many computer tasks. However, in generating hierarchical feature maps, the Transformer still adopts the ConvNet feature aggregation scheme. This leads to the problem that the semantic information of the grid area of image becomes confused after feature aggregation, making it difficult for attention to accurately model global relationships. To address this, we propose the Hierarchy Aware Feature Aggregation framework (HAFA). HAFA enhances the extraction of local features adaptively in shallow layers where semantic information is weak, while is able to aggregate patches with similar semantics in deep layers. The clear semantic information of the aggregated patches, enables the attention mechanism to more accurately model global information at the semantic level. Extensive experiments show that after using the HAFA framework, significant improvements have been achieved relative to the baseline models in image classification, object detection, and semantic segmentation tasks.

1. Introduction

The success of Transformer [37, 7, 18] in natural language processing (NLP) has inspired the research of Vision Transformer in the field of computer vision [9]. Benefiting from the excellent global modeling and asymmetric data processing abilities of Transformer, Vision Transformers have witnessed new state of the arts in various vision tasks like image classification [9, 24, 34, 42, 35], object detection [2, 59, 11, 54] and semantic segmentation [55, 44, 31, 38].

Compared to Convolutional Neural Networks (ConvNets), Transformers inherently have better global model-

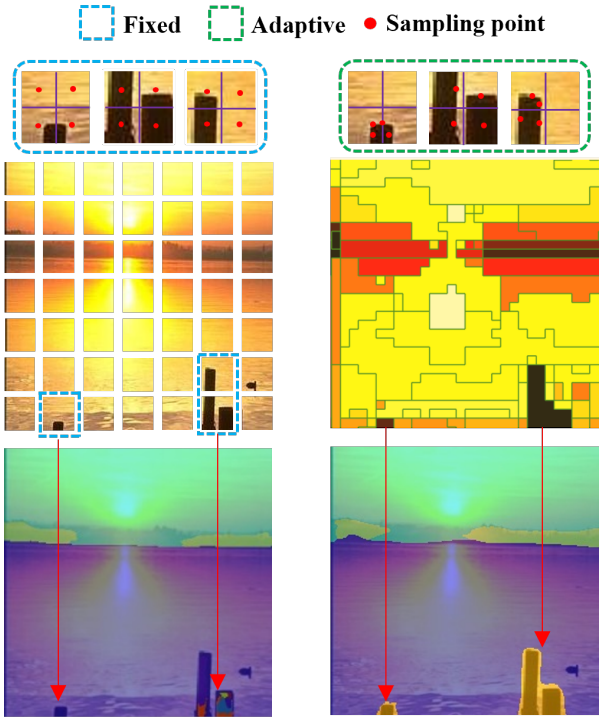

Figure 1. Comparison of feature aggregation based on Fixed Grids and the proposed Hierarchy Aware Feature Aggregation (HAFA). **Left:** An image is conceptualized into multiple fixed grids. In the shallow layers, the sampling center is fixed at the grid center, while in the deep layers, the semantic information is fragmented across multiple grids, leading to inaccurate semantic segmentation. **Right:** In the shallow layers, HAFA adaptively changes the sampling center through learning, enhancing local information perception. In the deep layers, HAFA aggregates semantic similar patches to ensure the integrity of semantic information. Compared to the fixed grids in the left figure, HAFA performs better in semantic segmentation due to its hierarchy aware way of conducting adaptive feature aggregation.

*Hongmin Liu is the corresponding author

ing capabilities. This is because the attention mechanism can directly model global relationships, so the Transformer does not need to establish global relationships through downsampling using sliding windows, as is the case of ConvNets. Nevertheless, in order to directly incorporate the Transformer into existing frameworks for downstream tasks[2, 59, 55, 48], it is still crucial to generate hierarchical feature maps for the Transformer. However, currently almost all Transformers, such as Swin Transformer [24] and PVT [40], still generate hierarchical feature maps following the downsampling method of ConvNets. As shown on the left side of Figure 1, in this paradigm, the image is divided into multiple fixed grids and downsampling is performed using a fixed-size sliding window to generate hierarchical feature maps. It is worth noting that this approach can lead to a serious problem: one object may be separated into multiple grids, results in the complete semantic information of objects being destroyed, in other words, multiple fragmented semantic information of different objects can be contained within a same grid. After downsampling with a fixed-size sliding window, the semantic information within the grid becomes confused, which further leads to inaccuracy in modeling global relationships through attention. As a result, downstream tasks may suffer from inaccurate segmentation or missed segmentation.

To address the above issue of ConvNets paradigm in building hierarchical vision Transformers, we propose a Hierarchy Aware Feature Aggregation framework (HAFA). Specifically, HAFA consists of two parts: a Semantic Information Aggregation (SIA) module and a Local Adaptive Feature Aggregation (LAA) module. The SIA module utilizes clustering to group patches with similar semantics in the feature space, which will be later aggregated for global relationship modeling using Transformer. Due to the fact that only patches with similar semantic information are aggregated, the semantic information is not confused after aggregation, which allows for accurate modeling of global relationships. Consequently, this approach effectively improves over inaccurate or missed segmentation in downstream tasks. As clustering in the feature space will result in missing of spatial distribution of patches, we propose to preserve the fine-grained spatial distribution of patches in the SIA module by conducting patch merging only on queries in Transformer. On the other hand, the quality of establishing deep semantic information largely depends on the extraction of shallow features, and Transformers tend to learn better local texture information in shallow learning than other layers, it is still necessary to use Transformer in the early stages. However, clustering shallow features is noisy due to the weaker semantic information contained in them. Therefore, we propose a Local Adaptive Feature Aggregation module (LAA) for shallow layers. LAA learns from the local texture information obtained by the model, adaptively changes the sampling center of the patch, and enhances the capture of local, especially edge information. Since the LAA module is feature-adaptive, it can effectively capture local information and avoid the loss of local information, such as edge information, caused by fixed grid segmentation and down-sampling by sliding windows. This ultimately helps to address the problem of incomplete semantic information construction in deep layers. In summary, the proposed HAFA framework uses the LAA module to enhance the learning of local texture information in shallow layers, enabling deeper layers to establish high-quality semantic information. In deep layers, the SIA module enables the model to establish global relationships more accurately. LAA and SIA modules work together in a hierarchy-aware way.

The contributions are summarized as follows:

- A novel hierarchical feature aggregation framework called HAFA is proposed, which can be applied as a plugin to Transformers, resulting in a significant improvement in performance with only a negligible increase in computational cost.
- Two feature-adaptive aggregation modules (LAA and SIA) are introduced in HAFA, which contribute to build hierarchical Vision Transformers with different and dynamic relationship modeling in a hierarchy aware way.
- Extensive experimental results indicate that HAFA consistently improves over various models, especially in downstream dense prediction tasks, with notable improvements on small object detection and semantic segmentation.

2. Related Works

2.1. Vision Transformer

The Transformer is a major architecture in natural language processing and has recently been extended to the computer vision domain. ViT [9] proposed a pure Transformer architecture and demonstrated the enormous potential of Transformers in visual tasks. As ViT heavily relies on large amounts of data, DeiT [34] introduced several training strategies to enable ViT to be trained on smaller ImageNet-1k datasets. Due to the quadratic complexity in ViT, some works were focused on improving the attention computation, with [8, 47, 46] using sparse attention to reduce computational complexity. Inspired by CNN models and the requirement for dealing with dense prediction tasks, [24, 40, 36, 49] and other works explored the feature pyramid structure of Transformers, which is different from the fixed token length in ViT. Subsequently, Transformers towards downstream tasks have also been proposed [4, 12, 19, 20, 58].

2.2. Hierarchical Feature Aggregation

The importance of hierarchical feature maps for downstream dense prediction tasks has been well demonstrated in CNNs [22, 15, 16, 30]. Recently, there has been an increasing number of works on generating hierarchical feature maps in Vision Transformers. Previous works can be divided into two categories: fixed grids based methods and dynamic features based ones. One popular fixed-grid hierarchical feature map generation method is Swin Transformer [24], which combines patches at the same position in adjacent windows into a new one. PVT [40] uses 2D convolution to fuse adjacent small patches into a larger one. However, such methods can result in confusion and fragmentation of semantic information within patches, leading to inaccurate modeling of global information. Therefore, dynamic feature-based hierarchical feature map generation has been proposed. In the LIT [28], MLPs are used instead of Transformer Blocks in the shallow layers. At each stage, the sampling center of the patch is changed by predicting an offset, and the patches with the modified center offset are aggregated to generate hierarchical feature maps. DynamicViT [29] and ViT-Slim [3] adopt a strategy of removing redundant patches in the image and retaining relatively important ones to form hierarchical feature maps. DynamicViT achieves this by using a predictor to select important patches, while ViT-Slim uses network architecture search. EviT [21] selects the top K tokens with the highest average values across all heads, and retains them for the next stage of fusion, while the remaining tokens are fused together. TCformer [52] aggregates redundant patches through clustering algorithms, generating more patches on the target object to capture more information. To incorporate more object information into the hierarchical feature maps, PSViT [50] iteratively moves the center of the patch towards the object during each iteration. DAT [43] incorporates the concept of deformable convolution into each block to generate dynamic attention. Token Merging [1] leverages cosine similarity to assess the similarity of tokens within each block and progressively merges similar tokens, thereby increasing the model's throughput.

The proposed HAFA belongs to the category of dynamic feature-based hierarchical feature map generation. Compared to existing methods, HAFA does not use the same feature aggregation method in all stages. Instead, in the shallow layers, HAFA uses the LAA module to enhance perception of local discriminative information, and in the deeper layers, it adopts the SIA module, which is capable of aggregating patches with similar semantic information. Because the information learned by the model from shallow to deep layers goes from local to global, HAFA could be a more natural and accurate method. Compared to DynamicViT [29], HAFA can be directly applied in downstream dense tasks and show better performance. Compared to DAT [43] and Token Merging [1], our method applies token merging between different stages to generate hierarchical features. It is worth to point out that LIT and TCformer are two extreme cases of HAFA (refer to Section 4.4). In addition, we experimentally found that the feature aggregation scheme of LIT [28] may cause unconvergence in some backbone networks (will be explained in Section 4.4). Compared to TCformer [52], our method can achieve higher accuracy and twice the inference speed with only 40% of the training resources consumed.

3. Method

In this section, we will first introduce the overall structure of the HAFA framework, followed by detailed introductions to the LAA module and the SIA module, and finally, we will discuss some details of HAFA.

3.1. Framework Overview

Figure 2 illustrates the proposed Hierarchy Aware Feature Aggregation (HAFA) framework built up on the Pyramid Vision Transformer backbone (PVT) [40], although extensions to other Vision Transformers are straightforward (see Table 6). Specifically, the HAFA framework adopts different feature aggregation schemes based on the features learned by the model at different stages. For example, targeted LAA and SIA modules are proposed in the shallow and deep layers of the model, respectively. It is worth noting that there is no additional explicit positional encoding after the SIA module.

3.2. Local Adaptive Feature Aggregation

Directly using a fixed grid to segment images and aggregating features using a sliding window in the shallow layers of the model can disrupt local information, especially edge information, as shown in Figure 1. However, edge information is crucial for establishing high quality semantic information later on. Therefore, We propose a Local Adaptive Feature Aggregation (LAA) module, which can adaptively enhance the capture of local information and preserve more edge information.

As shown in the bottom left of Figure 2, given the input feature map $F = \mathbb{R}^{C\times H\times W}$, where C, H and W are the feature channel dimensions, height, and width, respectively. The initial coordinates of the sampling point are regular grids denoted as $P_I \in \mathbb{R}^{2\times(n\times n)}$, where $(\mathrm{n}\times\mathrm{n})$ represents the number of sampled features, and the produced coordinates for sampling by LAA module are denoted as $P_E \in \mathbb{R}^{2\times(n\times n)}$. Instead of learning the sampled coordinates directly, we choose to learn the offset with respect to the regular grids inspired by the deformable CNN [5], Denoting the learned offset being $O_f \in \mathbb{R}^{2\times(n\times n)}$, the sam-

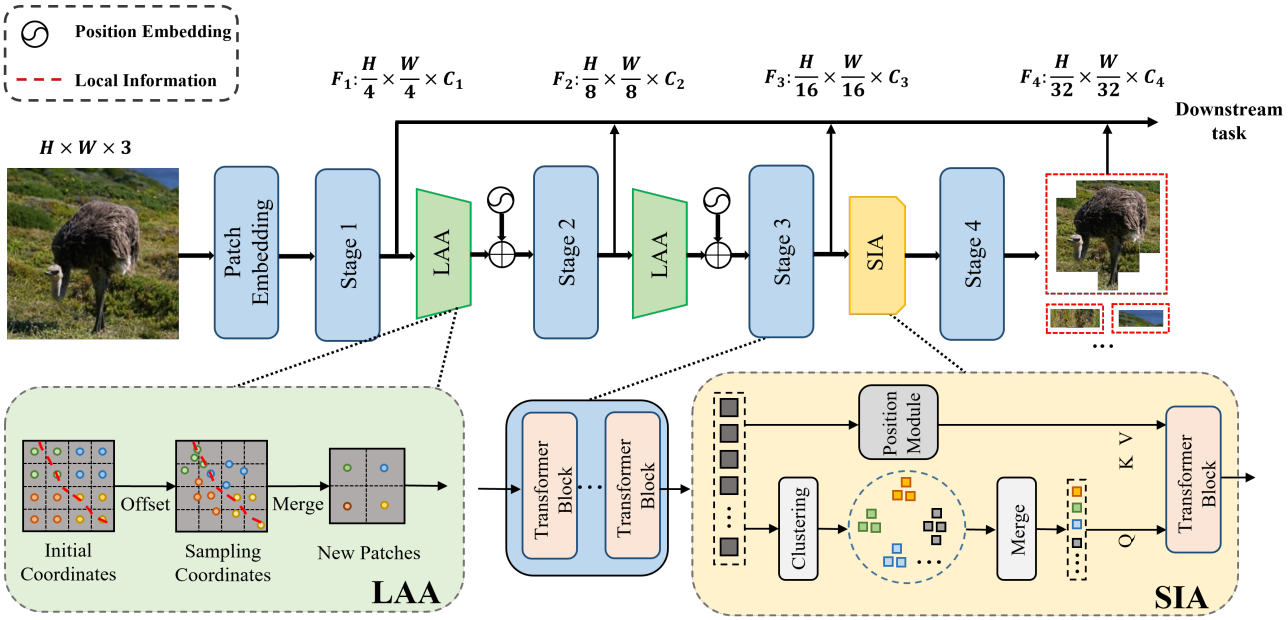

Figure 2. Overview of the proposed Hierarchy Aware Feature Aggregation (HAFA) framework. HAFA first employs the LAA module at the end of the first two stages to enhance perception of local discriminative information. Then, the SIA module is utilized before the final stage to aggregate patches with similar semantic information, in order to better model global relationships.

pling coordinates can be represented as:

$$P_E = P_I + O_f, \tag{1}$$

Based on the coordinates, a new patch is formed using bilinear interpolation based on the sampling coordinates of the patch, and the new patch is finally aggregated. After aggregation, the resulting feature map will have rich and high quality local information, which can help deep models better establish high quality semantic information.

3.3. Semantic Information Aggregation

To address the problem of grid semantic confusion that arises from aggregating features using the ConvNets paradigm and ultimately causes the attention mechanism failing to model global information correctly, we propose a Semantic Information Aggregation (SIA) module. The SIA module can aggregate patches with similar semantic information to avoid the semantic confusion. As shown in the bottom right of Figure 2, the SIA module mainly consists of two parts: patch clustering and merging.

Clustering. The purpose of clustering is to group patches with similar semantic information into a semantic group. When clustering patches, we use a density peak clustering algorithm based on K-nearest neighbors (DPC-KNN) [10, 52]. The reason for utilizing this clustering algorithm will be explained in Section 3.4. DPC-KNN involves two assumptions. Firstly, the local density of a cluster center is higher than that of the surrounding data points. Secondly, the centers of different clusters are far apart. This introduces two concepts, local density and relative distance.

Firstly, for the calculation of local density, we utilized a Gaussian kernel. Given N patches and k-nearest neighboring data points, the Euclidean distance was used to represent the distance between data points:

$$d_{ij} = \| x_i - x_j \|_2, \tag{2}$$

The formula for calculating the local density is as follows:

$$\rho_i = \exp\left(-\frac{1}{k} \sum_{x_j \in KNN(x_i)} (d_{ij})^2 \right), \tag{3}$$

Here, x represents a data point, d_{ij} represents the distance between data point x_i and x_j, ρ_i represents the local density of point x_i, and $KNN(x_i)$ represents the set of k-nearest neighboring data points of x_i, denoted as $x_j \in KNN(x_i)$.

The second concept is relative distance δ_i, which refers to the minimum distance between data point i and any other point with higher local density. Regards to data points with maximum local density, the relative distance is assumed to be the maximum value by default.

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}), i = 1, 2, \cdots, N \tag{4}$$

In order to satisfy two conditions simultaneously, namely large local density ρ_i and large relative distance σ_i,

the score $\rho_i \times \sigma_i$ for every data point can be calculated. The data points with the highest scores are then selected as the cluster centers, and the remaining data points are assigned to the cluster centers with the closest feature distance.

Merging. After clustering, each semantic group may contain a different number of patches, and different patches may contribute differently. Therefore, inspired by [52, 29], we use a predictor to quantify the importance p of each patch.

$$y_i = \sum_{j \in C_i} \frac{p_j}{\sum_{j \in C} p_j} x_j, j = 1, 2, \cdots, m \tag{5}$$

y_i represents the new patch after merging, C_i represents the ith cluster, and m is used to represent the number of patches in cluster.

Spatial distribution Reserving. The clustering algorithm in SIA only categorizes patches into different semantic groups based on semantic information in the feature space. Therefore, patches within the same semantic group are only semantically related and may not necessarily from adjacent regions. This can result in a loss of spatial distribution for the same semantic group after merging. To preserve fine-grained spatial distribution, our SIA module only performs clustering on the Q vector while adopting the implicit positional encoding module proposed by [42] on the K and V vectors. Due to the inclusion of the original spatial distribution information in K and V, the clustered Q is capable of preserving fine-grained spatial distribution through attention calculation based on K and V. For comparison, we will also show the results obtained by clustering Q, K, and V simultaneously in the model analysis section. The attention calculation is as follows:

$$Attention\,(Q, K_p, V_p) = softmax\left(\frac{QK_p^T}{\sqrt{d_k}}\right) V_p, \tag{6}$$

In this equation, Q represents the new patches generated by clustering and merging, while K_p and V_p represent the keys and values respectively, which have undergone positional encoding. d_k represents the channel number of the queries.

3.4. Discussions

Why is DPC-KNN? Firstly, this clustering method was employed in previous works [52] and showed good performance. To make a fair comparison, we also adopted the same clustering method. Secondly, due to the fact that clustering is only used prior to the final stage, the size of the feature maps is relatively small at this stage. After comparing multiple clustering algorithms, we found that there was no obvious difference in the time cost. Finally, DPC-KNN can perform clustering on dense data with arbitrary shapes, especially non-spherical clusters, while many clustering algorithms are only applicable to convex data sets, such as K-means [25].

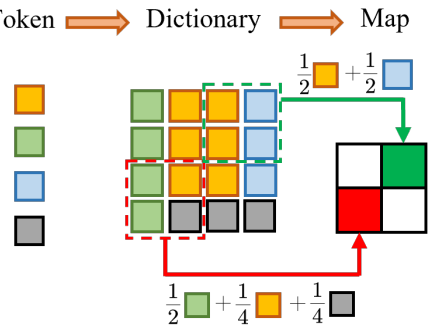

Figure 3. Illustration of Transforming from Vision Tokens to Feature Maps. From left to right are patches after clustering and merging, the dictionary that records the category relationship between tokens before and after clustering, and the transformed feature maps. The tokens after clustering can be converted to feature maps based on the relationship indicated by the dotted box.

Transform between Vision Tokens and Feature Maps. After merging patches from different semantic groups separately, the relative positions between the newly generated patches become relative disordered. Therefore, direct conversion between the feature map and Vision Tokens is not feasible. As shown in Figure 3, to address this issue, we use a dictionary to preserve the category of each token corresponding to the clustering process. It should be noted that the dictionary only records the category of each token and does not include any specific data. Based on the corresponding relationships between categories recorded in the dictionary, the tokens that have been clustered can be converted into the corresponding feature maps. Conversely, to convert a feature map into tokens, the process is reversed.

4. Experiments

4.1. Image Classification on ImageNet-1K

Setting. We first evaluate the proposed HAFA on the ImageNet-1k dataset [6], which includes 1.28 million training images and 50k validation images from 1,000 categories. To ensure a fair comparison, all of our models were trained on the training set and the top-1 accuracy was reported on the validation set. We apply random cropping, random horizontal flipping [32], label smoothing regularization [33], mixup [53], CutMix [51], and random erasing [56] as data augmentations. During training, we use a mini-batch size of 128, and employ AdamW [27] with momentum of 0.9 and weight decay of 5×10^{-2}. The initial learning rate is set to 1×10^{-3} and decreases following the cosine schedule [26]. All models are trained for 300 epochs from scratch on eight A40 GPUs. According to benchmark,

Method	#Param (M)	GFLOPs	Top-1 Acc (%)
ResNet18 [15]	11.7	1.8	68.5
DeiT-Tiny/16 [34]	5.7	1.3	72.2
PVT-Tiny [40]	13.2	1.9	75.1
PVT-Tiny(HAFA)	**14.6**	**1.9**	**77.5** (+2.4)
ResNet50 [15]	25.6	4.1	78.5
ResNeXt50-32x4d [45]	25.0	8.0	79.5
DeiT-Small/16 [34]	22.1	4.6	79.9
HRNet-W32 [39]	41.2	8.3	78.5
PVT-Small [40]	24.5	3.8	79.8
PVT-Small(HAFA)	**25.8**	**3.8**	**80.1** (+0.3)
ResNeXt101-64x4d [45]	83.5	15.6	81.5
ViT-Base/16 [9]	86.6	17.6	81.8
DeiT-Base/16 [34]	86.6	17.6	81.8
PVT-Large [40]	61.4	9.8	81.7
PVT-Large(HAFA)	**62.7**	**9.8**	**82.2** (+0.5)

Table 1. **Image classification performance on the ImageNet validation set.** "#Param" refers to the number of parameters. "GFLOPs" is calculated under the input size of 224 × 224.

we apply a center crop on the validation set, i.e., a 224×224 patch is cropped to evaluate the classification accuracy.

Result. In Table 1, we observe that inserting HAFA into PVT only adds a small number of parameters and does not significantly affect GFLOPs relative to the original PVT. This is because clustering itself does not participate in end-to-end training and does not have additional learnable parameters. Additionally, clustering is only used in the last stage, where the feature map is relatively small and does not contribute much to computational cost. The final experimental results demonstrate that the accuracy of models of different sizes has been improved, particularly the Tiny model. The latency of PVT-HAFA is as follows: Tiny: 7.0ms (+1.4), Small: 11.0ms (+1.5), Large: 24.8ms (+1.3). The numbers in the brackets represent the increase compared to the latency without HAFA (i.e., PVT). Due to the additional operations induced in the proposed HAHA framework, HAHA needs about 1.5ms more for inference. Notwithstanding, the HAFA framework has demonstrated its capacity to achieve substantial improvements in downstream tasks. For instance, in semantic segmentation tasks, the performance of PVT-Tiny-HAFA even surpasses that of the original PVT-Small model, as depicted in Table 2.

4.2. Semantic Segmentation on ADE20K

Setting. ADE20K [57] is a widely used semantic segmentation dataset, consisting of 150 categories with 20210, 2000, and 3352 images allocated for training, validation, and testing, respectively. All the compared methods was evaluated using the Semantic FPN [17] framework. The backbone network of our method was initialized with the pre-trained Imagenet-1k model, and the newly added layers were initialized using the Xavier method. The initial learning rate was set to 0.0001 and the model was optimized using the AdamW optimizer. we train our models for

Backbone	Semantic FPN		
	#Param(M)	GFLOPs	mIoU(%)
ResNet18 [15]	15.5	32.2	32.9
PVT-Tiny [40]	17.0	33.2	35.7
PVT-Tiny(HAFA)	**18.7**	**33.2**	**40.1** (+4.4))
ResNet50 [15]	28.5	45.6	36.7
PVT-Small [40]	28.2	44.5	39.8
PVT-Small(HAFA)	**29.9**	**44.5**	**43.8** (+4.0)
ResNeXt101-64x4d [45]	86.4	103.9	40.2
PVT-Large [40]	65.1	79.6	42.1
PVT-Large(HAFA)	**66.8**	**79.6**	**43.6** (+1.5)

Table 2. **Semantic segmentation performance of different backbones on the ADE20K validation set.** "GFLOPs" is calculated under the input scale of 512 × 512.

40k iterations with a batch size of 16 on eight A40 GPUs. The learning rate is decayed following the polynomial decay schedule with a power of 0.9. We randomly resize and crop the images to 512 × 512 for training, and rescale to have a shorter side of 512 pixels during testing.

Result. As shown in Table 2, the insertion of HAFA resulted in a significant improvement in the semantic segmentation task. It improved the performance of the Tiny, Small, and Large models by 4.4, 4.0, and 1.5 points, respectively. Surprisingly, after inserting HAFA, the Tiny and Small models exceeded the performance of the Small and Large models without HAFA, achieving a significant improvement across model scales. This is because HAFA aggregates feature maps based on the semantic information of patches during feature map generation. At the same time, it better captures local information, especially edge information, in the shallow layers. Therefore, HAFA has a significant advantage in semantic segmentation tasks. Based on the visualization results in Figure 4, we can observe a significant improvement in segmentation accuracy using our method.

4.3. Object Detection on COCO

Setting. Object detection and instance segmentation are performed on the COCO2017 dataset [23]. All of our models are trained on the training set with 118k images and evaluated on the validation set with 5k images. We validate the effectiveness of different backbones with Mask R-CNN [14]. We use the model pre-trained on ImageNet-1k to initialize the backbone and Xavier [13] initialization for the newly added layers. Our models are trained with a batch size of 16 on eight A40 GPUs and AdamW [27]. with an initial learning rate of 0.0001 and a weight decay of 0.0001. The training duration is set to 12 epochs.

Result. As shown in Table 3, the insertion of HAFA resulted in an improvement of 3.1, 1.4, and 0.9 points in the Tiny, Small, and Large models, respectively, in the object detection task. It also achieved a good improvement in the instance segmentation task. It is worth noting that

Backbone	#Param (M)	Mask R-CNN								
		AP^b	AP^b_{50}	AP^b_{75}	AP^b_s	AP^b_m	AP^b_l	AP^m	AP^m_{50}	AP^m_{75}
ResNet18 [15]	31.2	34.0	54.0	36.7	-	-	-	31.2	51.0	32.7
PVT-Tiny [40]	32.9	36.7	59.2	39.3	21.6	39.2	49.0	35.1	56.7	37.3
PVT-Tiny(HAFA)	**34.5**	**39.8** (+3.1)	**62.6**	**43.3**	**23.3**	**42.7**	**53.3**	**37.1**(+2.0)	**59.4**	**39.3**
ResNet50 [15]	44.2	38.0	58.6	41.4	-	-	-	34.4	55.1	36.7
PVT-Small [40]	44.1	40.4	62.9	43.8	22.9	43.0	55.4	37.8	60.1	40.3
PVT-Small(HAFA)	**45.8**	**41.8** (+1.4)	**64.4**	**45.7**	**26.0**	**44.6**	**56.1**	**38.9** (+1.1)	**61.5**	**41.9**
ResNeXt101-64x4d [45]	101.9	42.8	63.8	47.3	-	-	-	38.4	60.6	41.3
PVT-Large [40]	81.0	42.9	65.0	46.6	24.7	45.4	59.4	39.5	61.9	42.5
PVT-Large(HAFA)	**82.7**	**43.8** (+0.9)	**65.6**	**48.0**	**26.1**	**46.2**	**59.8**	**40.1** (+0.6)	**62.8**	**43.2**

Table 3. **Object detection and instance segmentation performance on COCO val2017.** AP^b and AP^m denote bounding box AP and mask AP, respectively.

by analyzing the detection results of models with different sizes, we can make several observations. The Tiny model shows a significant improvement in detecting medium-sized and large objects. This is because the global modeling capability of the Tiny model is relatively poor, while HAFA merges patches with similar semantic information, enabling the model to more accurately model global relationships. On the other hand, the Small and Large models show a greater improvement in detecting small objects. Although relatively larger models can model better global relationships, the destruction of local information in the shallow layers may prevent small objects from establishing complete semantic information in the deep layers, resulting in missed detections. However, HAFA enhance perception of local discriminative information in shallow layers, enabling smaller models to establish complete semantic information in deeper layers. The visualization results will be presented in the supplementary materials.

4.4. Model Analysis

In this section, we first explain why HAFA consists of two LAAs and one SIA module. Secondly, we demonstrate the importance of the proposed position encoding module in SIA. Finally, we will present the performance of our framework on other backbones.

Why two LAA modules and one SIA module? We conducted a series of experiments on ImageNet-1k to demonstrate the superiority of our method by replacing modules at different stages. To ensure a fair comparison, we employed the same PVT-v2 backbone as the previous work [52], which was a special case of our method. As shown in Table 4, we first replaced all stages with SIA modules, as was done in [52]. Subsequently, we gradually replaced SIA modules with LAA modules until all modules were LAA. We observed that the highest Top-1 accuracy could be achieved when the first two stages used LAA modules, while the last stage used SIA. Notably, when all stages used LAA, the model failed to converge by directly employing the PVT-v2-B1 training parameters. The discussion of how to alter the training strategy is beyond the scope of our paper.

We believe that in the shallow layers, the model primarily learns local information rather than semantic information. Therefore, using the SIA module in the shallow layers may not achieve the desired effect of aggregating patches with similar semantics; instead, it may introduce additional noise. According to our experiments, we found that the clustering results of SIA modules in the first stage had over 95% similarity with those of direct convolution. After replacing the SIA module in the first stage with LAA, the model achieved a 0.3% improvement. After further replacing the SIA module in the second stage with the LAA module, we observed that the difference in accuracy between the two approaches was relatively modest. To conduct a more comprehensive comparison between the two strategies, we measure their latency as detailed in Table 4. To balance the accuracy and inference speed, we selected a solution that used LAA module in the first two stages and SIA module in the last stage. To further show the effectiveness of this approach, we gradually replaced stage 3 and stage 2 with Conv2d layers, resulting in a decrease in the Top-1 accuracy by 0.5% and 0.7%, respectively. In addition, gradually replacing stage 1 and stage 2 with Conv2d layers both led to a reduction in the Top-1 accuracy by 0.5%.

Spatial distribution Reserving. In order to address the issue of missing spatial distribution of images after clustering, we perform clustering only on Q in the SIA module and use implicit positional encoding in K and V. We reserved the spatial distribution of images after clustering through the attention mechanism. As shown in Table 5, we discuss two different experimental setups: 1) clustering applied to Q, K, and V, and 2) clustering applied only to Q with implicit positional encoding inserted into K and V.

We found that clustering only applied to Q yielded better reserved of the spatial distribution of images and resulted in the best performance. This method resulted in a Top-1

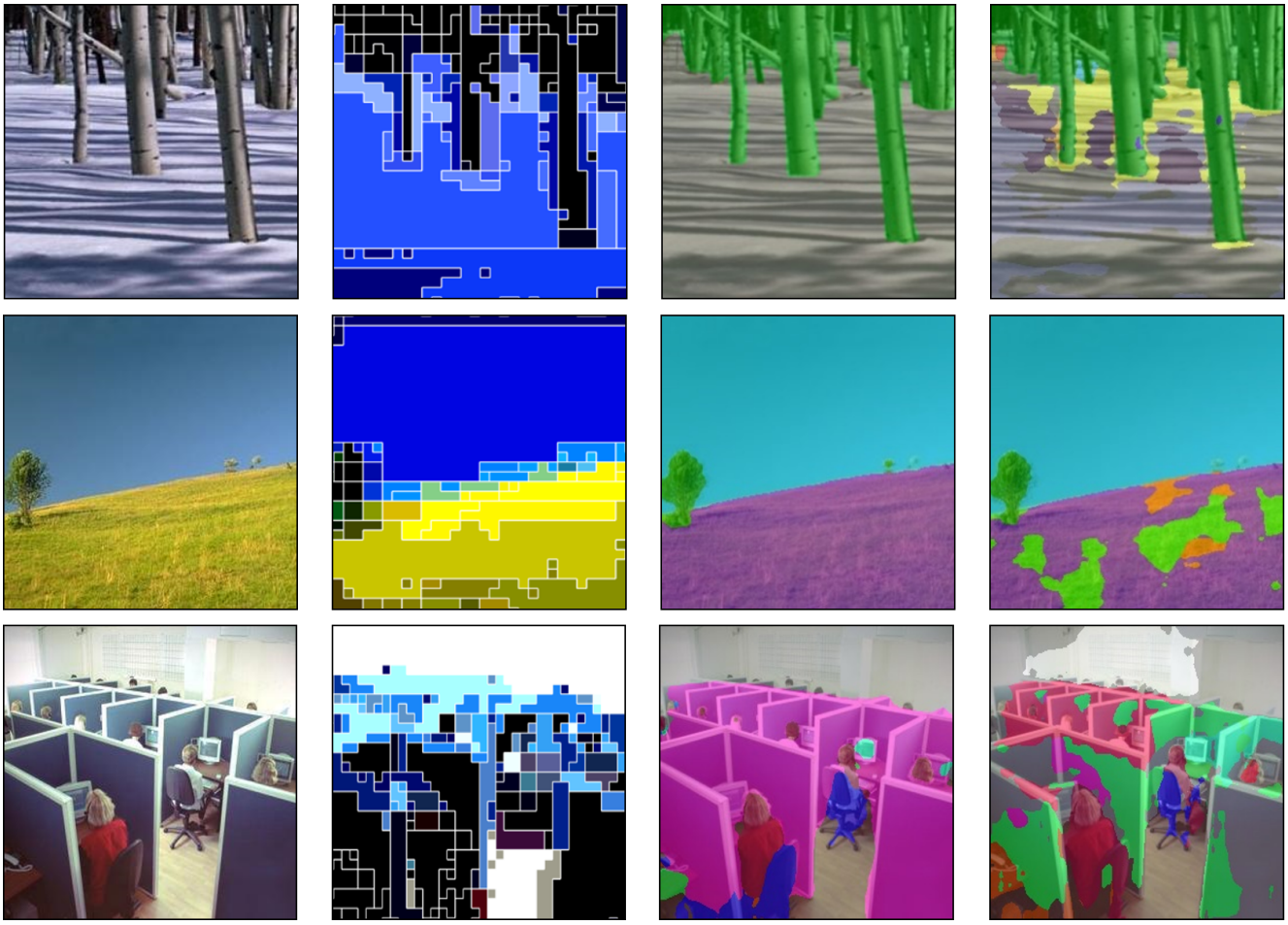

Figure 4. Visualization results of semantic segmentation. It includes the original image (the 1st column), the semantic clustering map produced by SIA (the 2nd column), the segmentation results with the HAFA framework (the 3rd column) or not (the 4th column).

Backbone	Stage1	Stage2	Stage3	ImageNet-1k Top-1 Acc(%)	Latency (ms)
PVT-v2-B1 [41]	SIA	SIA	SIA	79.4	38.5
	LAA	SIA	SIA	79.7	23.8
	LAA	**LAA**	**SIA**	**79.8**	**16.3**
	LAA	LAA	LAA	—	—

Table 4. Abalation experiments were conducted by using different modules at different stages. We conducted tests on an RTX3090 with an image resolution of 224x224 to compare the inference speeds of different solutions. "—" indicates the occurrence of training collapse.

accuracy increase of 0.4%, an improvement of 0.4 box AP and 0.2 mask AP on COCO, and an increase of 0.3% mIoU on ADE20K. This indicates that clustering only on Q, with the insertion of implicit positional encoding in K and V, can reserved the spatial distribution prior to clustering to some extent, thus verifying the effectiveness of our method.

Effects of HAFA on other Backbones. We inserted

Clustering	ImageNet	COCO		ADE20K
	Top-1 Acc(%)	AP^b	AP^m	mIoU(%)
Q K V	77.1	39.4	36.9	39.8
Q	**77.5** (+0.4)	**39.8** (+0.4)	**37.1** (+0.2)	**40.1** (+0.3)

Table 5. The results of using fine-grained spatial distribution reserving in the SIA module on different tasks using the PVT-Tiny backbone network.

HAFA into other mainstream backbones and trained them on ImageNet-1k. The Top-1 accuracies are shown in Table 6, from which we can see that the improvement is universal and so the proposed HAFA generalizes well to other Transformer backbones. Concurrently, we perform a comparative analysis between HAFA and the closely related DynamicViT [29]. As evidenced by Table 6, when DynamicViT [29] and HAFA employ models with comparable parameter quantities, PVT-v2-B2-HAFA achieves superior performance. Additionally, when employing an identical Swin backbone and retaining an equivalent 25% token

Method	#Param (M)	GFLOPs	Top-1 Acc (%)
PVT-v2-B1	14.0	1.9	78.7
PVT-v2-B1(HAFA)	**14.6**	**1.9**	**79.8** (+1.1)
PVT-v2-B2	25.4	4.0	82.0
DynamicViT-LV-S/0.5	26.9	3.7	82.0
PVT-v2-B2(HAFA)	**26.0**	**4.0**	**82.5** (+0.5)
Swin-T	30.8	4.5	81.3
Dynamic-Swin-T/0.25	29.8	4.0	77.8
Swin-T(HAFA)	**29.1**	**4.5**	**81.7** (+0.4)

Table 6. Integrating HAFA with other backbones and comparing with similar works. "#Param" refers to the number of parameters. "GFLOPs" is calculated under the input scale of 224 × 224. The value of 0.5 represents that each stage retains 50% of the total number of tokens, while HAFA defaults to 0.25 (standard feature pyramid ratio)

within each stage, HAFA demonstrates a higher level of accuracy.

5. Conclusion

In this paper, we propose a Hierarchy Aware Feature Aggregation framework (HAFA) that can adopt different feature aggregation schemes based on the features learned by the model at different stages. We employ the Local Adaptive Feature Aggregation (LAA) module in the shallow layers of the model to enhance perception of local discriminative information. In the deep layers of the model, we use the Semantic Information Aggregation (SIA) module to aggregate patches with similar semantic information, helping the attention mechanism to model global relationships more accurately. Experimental results show that the baseline model achieved significant improvements in multiple tasks after integrating into the HAFA framework.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1313002; in part by the National Natural Science Foundation of China under Grant 62222302, Grant U22B2055 and Grant U2013202; in part by the Fundamental Research Funds for the Central Universities under Grant FRFTP-22-003C1. Thanks to Jiaqi Ma, Yuefeng Cai, and Jinglin Xu for valuable discussions.

References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[3] Arnav Chavan, Zhiqiang Shen, Zhuang Liu, Zechun Liu, Kwang-Ting Cheng, and Eric P Xing. Vision transformer slimming: Multi-dimension searching in continuous optimization space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4931–4941, 2022.

[4] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13608–13618, 2022.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[6] 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.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] 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.

[9] 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.

[10] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99:135–145, 2016.

[11] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021.

[12] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*, 2022.

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[15] 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.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[17] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019.

[18] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[19] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13147–13156, 2022.

[20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022.

[21] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] 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.

[25] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[26] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] Zizheng Pan, Bohan Zhuang, Haoyu He, Jing Liu, and Jianfei Cai. Less is more: Pay less attention in vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2035–2043, 2022.

[29] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[31] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[35] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 459–479. Springer, 2022.

[36] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 459–479. Springer, 2022.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Jian Wang, Chenhui Gou, Qiman Wu, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Rtformer: Efficient design for real-time semantic segmentation with transformer. *arXiv preprint arXiv:2210.07124*, 2022.

[39] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[40] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[41] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

[42] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[43] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4794–4803, 2022.

[44] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[45] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[46] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. *arXiv preprint arXiv:2207.02202*, 2022.

[47] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021.

[48] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *European Conference on Computer Vision*, pages 288–307. Springer, 2022.

[49] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L Yuille, and Wei Shen. Glance-and-gaze vision transformer. *Advances in Neural Information Processing Systems*, 34:12992–13003, 2021.

[50] Xiaoyu Yue, Shuyang Sun, Zhanghui Kuang, Meng Wei, Philip HS Torr, Wayne Zhang, and Dahua Lin. Vision transformer with progressive sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 387–396, 2021.

[51] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[52] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11101–11111, 2022.

[53] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[54] Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. Vit-yolo: Transformer-based yolo for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2799–2808, 2021.

[55] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[56] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[57] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[58] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. Pttr: Relational 3d point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8531–8540, 2022.

[59] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.