Head-Free Lightweight Semantic Segmentation with Linear Transformer

Bo Dong*, Pichao Wang†, Fan Wang

Alibaba Group
{bo.dong.cst, pichaowang}@gmail.com; fan.w@alibaba-inc.com

Abstract

Existing semantic segmentation works have been mainly focused on designing effective decoders; however, the computational load introduced by the overall structure has long been ignored, which hinders their applications on resource-constrained hardwares. In this paper, we propose a head-free lightweight architecture specifically for semantic segmentation, named Adaptive Frequency Transformer (AFFormer). AFFormer adopts a parallel architecture to leverage prototype representations as specific learnable local descriptions which replaces the decoder and preserves the rich image semantics on high-resolution features. Although removing the decoder compresses most of the computation, the accuracy of the parallel structure is still limited by low computational resources. Therefore, we employ heterogeneous operators (CNN and Vision Transformer) for pixel embedding and prototype representations to further save computational costs. Moreover, it is very difficult to linearize the complexity of the vision Transformer from the perspective of spatial domain. Due to the fact that semantic segmentation is very sensitive to frequency information, we construct a lightweight prototype learning block with adaptive frequency filter of complexity $O(n)$ to replace standard self attention with $O(n^2)$. Extensive experiments on widely adopted datasets demonstrate that AFFormer achieves superior accuracy while retaining only 3M parameters. On the ADE20K dataset, AFFormer achieves 41.8 mIoU and 4.6 GFLOPs, which is 4.4 mIoU higher than Segformer, with 45% less GFLOPs. On the Cityscapes dataset, AFFormer achieves 78.7 mIoU and 34.4 GFLOPs, which is 2.5 mIoU higher than Segformer with 72.5% less GFLOPs. Code is available at https://github.com/dongbo811/AFFormer.

Introduction

Semantic segmentation aims to partition an image into sub-regions (collections of pixels) and is defined as a pixel-level classification task (Long, Shelhamer, and Darrell 2015; Chen et al. 2018; Strudel et al. 2021) since Fully Convolutional Networks (FCN) (Long, Shelhamer, and Darrell 2015). It has two unique characteristics compared to image classification: pixel-wise dense prediction and multi-class representation, which is usually built upon high-resolution features and requires a global inductive capability of image semantics, respectively. Previous semantic segmentation methods (Zhao et al. 2017; Chen et al. 2018; Strudel et al. 2021; Xie et al. 2021; Cheng, Schwing, and Kirillov 2021; Yuan et al. 2021b) focus on using the classification network as backbone to extract multi-scale features, and designing a complicated decoder head to establish the relationship between multi-scale features. However, these improvements come at the expense of large model size and high computational cost. For instance, the well-known PSPNet (Zhao et al. 2017) using light-weight MobilenetV2 (Sandler et al. 2018) as backbone contains 13.7M parameters and 52.2 GFLOPs with the input scale of 512×512. The widely-used DeepLabV3+ (Chen et al. 2018) with the same backbone requires 15.4M parameters and 25.8 GFLOPs. The inherent design manner limits the development of this field

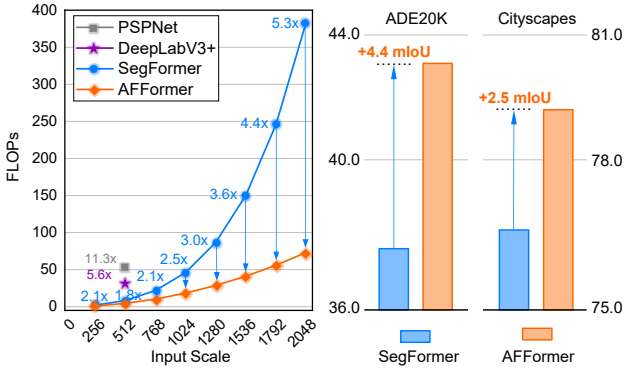

Figure 1: *Left:* Computational complexity under different input scales. Segformer (Xie et al. 2021) significantly reduces the computational complexity compared to traditional methods, such as PSPNet (Zhao et al. 2017) and DeepLabV3+ (Chen et al. 2018) which have mobilenetV2 (Sandler et al. 2018) as backbone. However, Segformer still has a huge computational burden for higher resolutions. *Right:* AFFormer achieves better accuracy on ADE20K and Cityscapes datasets with significantly lower FLOPs.

*Work done during an internship at Alibaba Group.

†Corresponding author; work done at Alibaba Group, and now affiliated with Amazon Prime Video.

and hinders many real-world applications. Thus, we raise the following question: *can semantic segmentation be as simple as image classification?*

Recently vision Transformers (ViTs) (Liu et al. 2021; Xie et al. 2021; Lee et al. 2022) have shown great potential in semantic segmentation, however, they face the challenges of balancing performance and memory usage when deployed on ultra-low computing power devices. Standard Transformers has computational complexity of $O(n^2)$ in the spatial domain, where n is the input resolution. Existing methods alleviate this situation by reducing the number of tokens (Wang et al. 2021; Ren et al. 2022) or sliding windows (Liu et al. 2021; Yuan et al. 2021a), but they introduce limited reduction on computational complexity and even compromise global or local semantics for the segmentation task. Meanwhile, semantic segmentation as a fundamental research field, has extensive application scenarios and needs to process images with various resolutions. As shown in Figure 1, although the well-known efficient Segformer (Xie et al. 2021) achieves a great breakthrough compared to PSPNet and DeepLabV3+, it still faces a huge computational burden for higher resolutions. At the scale of 512×512, although Segformer is very light compared to PSPNet and DeepLabV3+, it is almost twice as expensive as ours (8.4 GFLOPs *vs* 4.6 GFLOPs); at the scale of 2048×2048, even 5x GFLOPs is required (384.3 GFLOPs *vs* 73.2 GFLOPs). Thus, we raise another question: *can we design an efficient and lightweight Transformer network for semantic segmentation in ultra-low computational scenarios?*

The answers to above two questions are affirmative. To this end, we propose a head-free lightweight semantic segmentation specific architecture, named Adaptive Frequency Transformer (AFFormer). Inspired by the properties that ViT maintains a single high-resolution feature map to keep details (Dosovitskiy et al. 2021) and the pyramid structure reduces the resolution to explore semantics and reduce computational cost (He et al. 2016; Wang et al. 2021), AFFormer adopts a parallel architecture to leverage the prototype representations as specific learnable local descriptions which replace the decoder and preserves the rich image semantics on high-resolution features. The parallel structure compresses the majority of the computation by removing the decoder, but it is still not enough for ultra-low computational resources. Moreover, we employ heterogeneous operators for pixel embedding features and local description features to save more computational costs. A Transformer-based module named *prototype learning* (PL) is used to learn the prototype representations, while a convolution-based module called *pixel descriptor* (PD) takes pixel embedding features and the learned prototype representations as inputs, transforming them back into the full pixel embedding space to preserve high-resolution semantics.

However, it is still very difficult to linearize the complexity of the vision Transformer from the perspective of spatial domain. Inspired by the effects of frequency on classification tasks (Rao et al. 2021; Wang et al. 2020), we find that semantic segmentation is also very sensitive to frequency information. Thus, we construct a lightweight adaptive frequency filter of complexity $O(n)$ as prototype learning to replace the standard self attention with $O(n^2)$. The core of this module is composed of frequency similarity kernel, dynamic low-pass and high-pass filters, which capture frequency information that is beneficial to semantic segmentation from the perspectives of emphasizing important frequency components and dynamically filtering frequency, respectively. Finally, the computational cost is further reduced by sharing weights in high and low frequency extraction and enhancement modules. We also embed a simplified depthwise convolutional layer in the feed-forward network (FFN) layer to enhance the fusion effect, reducing the size of the two matrix transformations.

With the help of parallel heterogeneous architecture and adaptive frequency filter, we use only one convolutional layer as classification layer (CLS) for single-scale feature, achieving the best performance and making semantic segmentation as simple as image classification. We demonstrate the advantages of the proposed AFFormer on three widely-used datasets: ADE20K, Cityscapes and COCO-stuff. With only 3M parameters, AFFormer significantly outperforms the state-of-the-art lightweight methods. On ADE20K, AFFormer achieves 41.8 mIoU with 4.6 GFLOPs, outperforming Segformer by 4.4 mIoU, while reducing GFLOPs by 45%. On Cityscapes, AFFormer achieves 78.7 mIoU and 34.4 GFLOPs, which is 2.5 mIoU higher than Segformer, with 72.5% less GFLOPs. Extensive experimental results demonstrate that it is possible to apply our model in computationally constrained scenarios, which still maintaining the high performance and robustness across different datasets.

Related Work

Semantic Segmentation

Semantic segmentation is regarded as a pixel classification task (Xu et al. 2017; Xie et al. 2021). In the last two years, new paradigms based on visual Transformers have emerged, which enable mask classification via queries or dynamic kernels (Zhang et al. 2021; Li et al. 2022; Cheng et al. 2022). For instance, Maskformer (Cheng, Schwing, and Kirillov 2021) learns an object query and converts it into an embedding of masks. Mask2former (Cheng et al. 2022) enhances the query learning with a powerful multi-scale masked Transformer (Zhu et al. 2021). K-Net (Zhang et al. 2021) adopts dynamic kernels for masks generation. MaskDINO (Li et al. 2022) brings object detection to semantic segmentation, further improving query capabilities. However, all above methods are not suitable for low computing power scene due to the high computational cost of learning efficient queries and dynamic kernels. We argue that the essence of these paradigms is to update pixel semantics by replacing the whole with individual representations. Therefore, we leverage pixel embeddings as a specific learnable local description that extracts image and pixel semantics and allows semantic interaction.

Efficient Vision Transformers

The lightweight solution of vision Transformer mainly focuses on the optimization of self attention, including following ways: reducing the token length (Wang et al. 2021; Xie

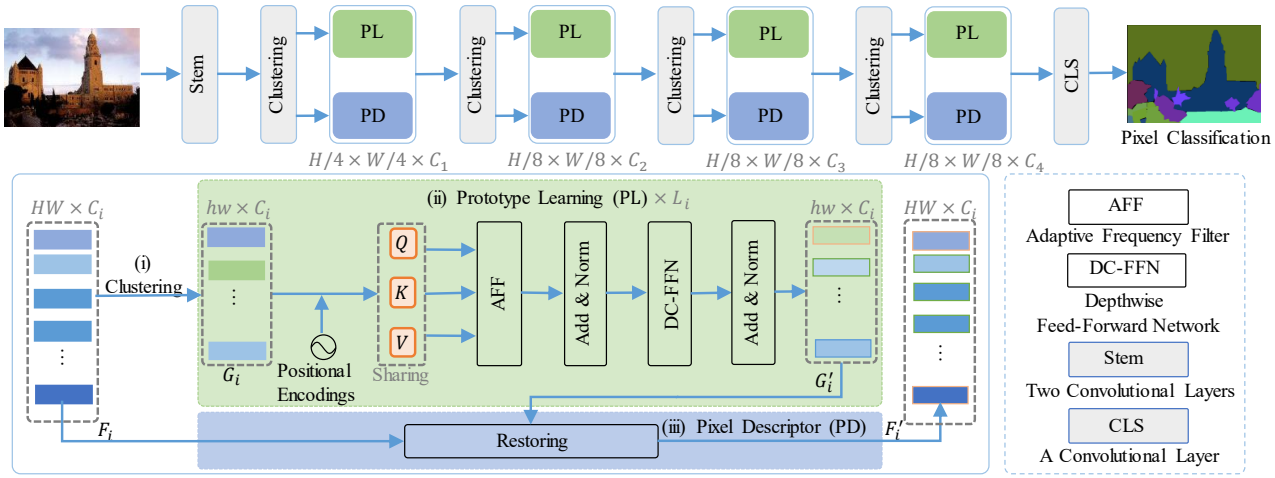

Figure 2: An Overview of Adaptive Frequency Transformer (AFFormer). We first displays the overall structure of parallel heterogeneous network. Specifically, the feature $\boldsymbol{F}$ after patch embedding is first clustered to obtain the prototype feature $\boldsymbol{G}$, so as to construct a parallel network structure, which includes two heterogeneous operators. A Transformer-based module as prototype learning to capture favorable frequency components in $\boldsymbol{G}$, resulting prototype representation $\boldsymbol{G}'$. Finally $\boldsymbol{G}'$ is restored by a CNN-based pixel descriptor, resulting $\boldsymbol{F}'$ for the next stage.

et al. 2021) and using local windows (Liu et al. 2021; Yuan et al. 2021a). PVT (Wang et al. 2021) performs spatial compression on keys and values through spatial reduction, and PVTv2 (Wang et al. 2022) further replaces the spatial reduction by pooling operation, but many details are lost in this way. Swin (Liu et al. 2021; Yuan et al. 2021a) significantly reduce the length of the token by restricting self attention to local windows, while these against the global nature of Transformer and restrict the global receptive field. At the same time, many lightweight designs (Chen et al. 2022; Mehta and Rastegari 2022) introduce Transformers in MobileNet to obtain more global semantics, but these methods still suffer from the square-level computational complexity of conventional Transformers. Mobile-Former (Chen et al. 2022) combines the parallel design of MobileNet (Sandler et al. 2018) and Transformer (Dosovitskiy et al. 2021), which can achieve bidirectional fusion performance of local and global features far beyond lightweight networks such as MobileNetV3. However, it only uses a very small number of tokens, which is not conducive to semantic segmentation tasks.

Method

In this section, we introduce the lightweight parallel heterogeneous network for semantic segmentation. The basic information is first provivided on the replacement of semantic decoder by parallel heterogeneous network. Then, we introduce the modeling of pixel descriptions and semantic frequencies. Finally, the specific details and the computational overhead of parallel architectures are discussed.

Parallel Heterogeneous Architecture

The semantic decoder propagates the image semantics obtained by the encoder to each pixel and restores the lost details in downsampling. A straightforward alternative is to extract image semantics in high resolution features, but it introduces a huge amount of computation, especially for vision Transformers. In contrast, we propose a novel strategy to describe pixel semantic information with prototype semantics. For each stage, given a feature $\boldsymbol{F} \in \mathbb{R}^{H \times W \times C}$, we first initial a grid $\boldsymbol{G} \in \mathbb{R}^{h \times w \times C}$ as a prototype of the image, where each point in $\boldsymbol{G}$ acts as a local cluster center, and the initial state simply contains information about the surrounding area. Here we use a $1 \times C$ vector to represent the local semantic information of each point. For each specific pixel, because the semantics of the surrounding pixels are not consistent, there are overlap semantics between each cluster centers. The cluster centers are weighted initialized in its corresponding area α^2, and the initialization of each cluster center is expressed as:

$$\boldsymbol{G}(s) = \sum_{i=0}^{n} w_i x_i \tag{1}$$

where $n = \alpha \times \alpha$, w_i denotes the weight of x_i, and α is set to 3. Our purpose is to update each cluster center s in the grid G instead of updating the feature $\boldsymbol{F}$ directly. As $h \times w \ll H \times W$, it greatly simplifies the computation.

Here, we use a Transformer-based module as prototype learning to update each cluster center, which contains L layers in total, and the updated center is denoted as $\boldsymbol{G}'(s)$. For each updated cluster center, we recover it by a pixel descriptor. Let F_i' denote the recovered feature, which contains not only the rich pixel semantics from F, but also the prototype semantics collected by the cluster centers $\boldsymbol{G}'(s)$. Since the cluster centers aggregate the semantics of surrounding pixels, resulting in the loss of local details, PD first models local details in F with pixel semantics. Specifically, F is projected to a low-dimensional space, establishing local relationships

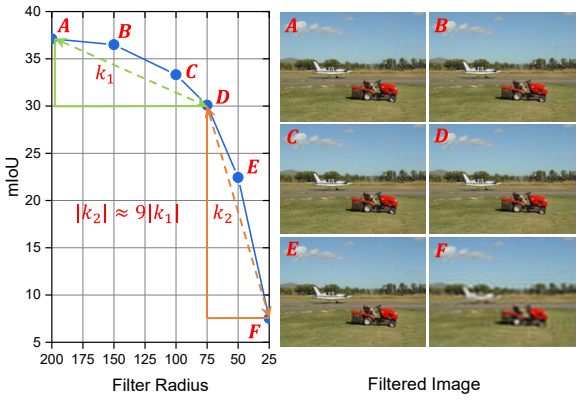

Figure 3: The effect of different frequency components on semantic segmentation. We use the cut-edge method Segformer (Xie et al. 2021) to evaluate the impact of frequency components on semantic segmentation on the widely used ADE20K dataset (Zhou et al. 2017). The image is transformed into the frequency domain by a fast Fourier transform (Heideman, Johnson, and Burrus 1984), and high-frequency information is filtered out using a low-pass operator with a radius. Removing high-frequency components at different levels results the prediction performance drops significantly.

between pixels such that each local patch keeps a distinct boundary. Then $\boldsymbol{G}'(s)$ is embedded into F to restore to the original space feature F' through bilinear interpolation. Finally, they are integrated through a linear projection layer.

Prototype Learning by Adaptive Frequency Filter

Motivation Semantic segmentation is an extremely complex pixel-level classification task that is prone to category confusion. The frequency representation can be used as a new paradigm of learning difference between categories, which can excavate the information ignored by human vision (Zhong et al. 2022; Qian et al. 2020). As shown in Figure 3, humans are robust to frequency information removal unless the vast majority of frequency components are filtered out. However, the model is extremely sensitive to frequency information removal, and even removing a small amount would result in significant performance degradation. It shows that for the model, mining more frequency information can enhance the difference between categories and make the boundary between each category more clear, thereby improving the effect of semantic segmentation.

Since feature $\boldsymbol{F}$ contains rich frequency features, each cluster center in the grid $\boldsymbol{G}$ also collects these frequency information. Motivated by the above analysis, extracting more beneficial frequencies in grid $\boldsymbol{G}$ helps to discriminate the attributes of each cluster. To extract different frequency features, the straightforward way is to transform the spatial domain features into spectral features through Fourier transform, and use a simple mask filter in the frequency domain to enhance or attenuate the intensity of each frequency component of the spectrum. Then the extracted frequency features are converted to the spatial domain by inverse Fourier transform. However, Fourier transform and inverse transform bring in additional computational expenses, and such operators are not supported on many hardwares. Thus, we design an adaptive frequency filter block based on the vanilla vision Transformer from the perspective of spectral correlation to capture important high frequency and low frequency features directly in the spatial domain. The core components are shown in Figure 4 and the formula $\boldsymbol{F}(\cdot)$ is defined as:

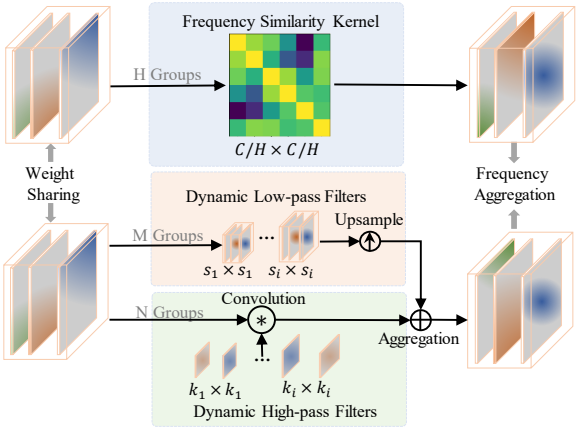

Figure 4: Structure of the adaptive frequency filter in prototype learning. The prototype as learnable local description utilizes frequency component similarity kernel to enhance different components while combining efficient and dynamic low-pass and high-pass filters to capture more frequency information.

$$\boldsymbol{F}(X) = \underbrace{||\boldsymbol{D}_h^{fc}(X)||_H}_{corr.} + \underbrace{||\boldsymbol{D}_m^{lf}(X)||_M + ||\boldsymbol{D}_n^{hf}(X)||_N}_{dynamic\ filters}, \tag{2}$$

where $\boldsymbol{D}_h^{fc}$, $\boldsymbol{D}_m^{lf}(X)$ and $\boldsymbol{D}_n^{hf}(X)$ denote the frequency similarity kernel with H groups to achieve frequency component correlation enhancement, dynamical low-pass filters with M groups and dynamical high-pass filters with N groups, respectively. $||\cdot||$ denotes concatenation. It is worth noting that these operators adopt a parallel structure to further reduce the computational cost by sharing weights.

Frequency Similarity Kernel (FSK) Different frequency components distribute over in $\boldsymbol{G}$, and our purpose is to select and enhance the important components that helps semantic parsing. To this end, we design a frequency similarity kernel module. Generally, this module is implemented by the vision Transformer. Given a feature $\boldsymbol{X} \in \mathbb{R}^{(hw)\times C}$, with relative position encoding on $\boldsymbol{G}$ through a convolution layer (Wu et al. 2021). We first use a fixed-size similarity kernel $\boldsymbol{A} \in \mathbb{R}^{C/H\times C/H}$ to represent the correspondence between different frequency components, and select the important frequency components by querying the similarity kernel. We treat it as a function transfer that computes the keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ of frequency components through a linear

layer, and normalizes the keys across frequency components by a Softmax operation. Each component integrates a similarity kernel $\boldsymbol{A}_{i,j}$, which is computed as:

$$\boldsymbol{A}_{i,j} = e^{\boldsymbol{k}_i \boldsymbol{v}_j^\top} / \sum_{j=1}^{n} e^{\boldsymbol{k}_i}, \tag{3}$$

where $\boldsymbol{k}_i$ represents the i-th frequency component in $\boldsymbol{K}$, $\boldsymbol{v}_j$ represents the j-th frequency component in $\boldsymbol{V}$. We also transform the input $\boldsymbol{X}$ into the query $\boldsymbol{Q}$ through a linear layer, and obtain the component-enhanced output through interactions on the fixed-size similarity kernel.

Dynamic Low-Pass Filters (DLF) Low-frequency components occupy most of the energy in the absolute image and represent most of the semantic information. A low-pass filter allows signals below the cutoff frequency to pass, while signals above the cutoff frequency are obstructed. Thus, we employ typical average pooling as a low-pass filter. However, the cutoff frequencies of different images are different. To this end, we control different kernels and strides in multi-groups to generate dynamic low-pass filters. For m-th group, we have:

$$\boldsymbol{D}_m^{lf}(\boldsymbol{v}^m)) = \boldsymbol{B}(\Gamma_{s\times s}(\boldsymbol{v}^m)), \tag{4}$$

where $\boldsymbol{B}(\cdot)$ represents bilinear interpolation and $\Gamma_{s\times s}$ denotes the adaptive average pooling with the output size of $s \times s$.

Dynamic High-Pass Filters (DHF) High-frequency information is crucial to preserve details in segmentation. As a typical high-pass operator, convolution can filter out irrelevant low-frequency redundant components to retain favorable high-frequency components. The high-frequency components determine the image quality and the cutoff frequency of the high-pass for each image is different. Thus, we divide the value $\boldsymbol{V}$ into $\boldsymbol{N}$ groups, resulting $\boldsymbol{v}^n$. For each group, we use a convolution layer with different kernels to simulate the cutoff frequencies in different high-pass filters. For the n-th group, we have:

$$\boldsymbol{D}_n^{hf}(\boldsymbol{v}^n)) = \Lambda_{k\times k}(\boldsymbol{v}^n), \tag{5}$$

where $\Lambda_{k\times k}$ denotes the depthwise convolution layer with kernel size of $k \times k$. In addition, we use the Hadamard product of query and high-frequency features to suppress high frequencies inside objects, which are noise for segmentation.

FFN helps to fuse the captured frequency information, but owns a large amount of calculation, which is often ignored in lightweight designs. Here we reduce the dimension of the hidden layer by introducing a convolution layer to make up for the missing capability due to dimension compression.

Discuss For the frequency similarity kernel, the computational complexity is $\mathcal{O}(hwC^2)$. The complexity of each dynamic high-pass filter is $\mathcal{O}(hwCk^2)$, which is much smaller than that of frequency similarity kernel. Since the dynamic low-pass filter is implemented by adaptive mean pooling of each group, its complexity is about $\mathcal{O}(hwC)$. Therefore, the complexity of a module is linear with the resolution, which is advantageous for high resolution in semantic segmentation.

Model	#Param.	FLOPs	mIoU
FCN-8s	9.8M	39.6G	19.7
PSPNet (MV2)	13.7M	52.2G	29.6
DeepLabV3+ (MV2)	15.4M	25.8G	38.1
DeepLabV3+ (EN)	17.1M	26.9G	36.2
DeepLabV3+ (SV2)	16.9M	15.3G	37.6
Lite-ASPP	2.9M	4.4G	36.6
R-ASPP	2.2M	2.8G	32.0
LR-ASPP	3.2M	2.0G	33.1
HRNet-W18-Small	4.0M	10.2G	33.4
HR-NAS-A	2.5M	1.4G	33.2
HR-NAS-B	3.9M	2.2G	34.9
PVT-v2-B0	7.6M	25.0G	37.2
TopFormer	5.1M	1.8G	37.8
EdgeViT-XXS	7.9M	24.4G	39.7
Segformer (LVT)	3.9M	10.6G	39.3
Swin-tiny	31.9M	46G	41.5
Xcit-T12/16	8.4M	21.5G	38.1
ViT	10.2M	24.6G	37.4
PVT-tiny	17.0M	33G	36.6
Segformer	3.8M	8.4G	37.4
AFFormer-tiny	1.6M(-58%)	2.8G(-67%)	38.7(+1.3)
AFFormer-small	2.3M(-41%)	3.6G(-61%)	40.2(+2.8)
AFFormer-base	3.0M(-21%)	4.6G(-45%)	41.8(+4.4)

Table 1: Comparison to state of the art methods on ADE20K with resolution at 512×512. Here we use the Segformer as the baseline and report the percentage growth. MV2=MobileNetV2, EN=EfficientNet, SV2=ShuffleNetV2.

Experiments

Implementation Details

We validate the proposed AFFormer on three publicly datasets: ADE20K (Zhou et al. 2017), Cityscapes (Cordts et al. 2016) and COCO-stuff (Caesar, Uijlings, and Ferrari 2018). We implement our AFFormer with the PyTorch framework base on MMSegmentation toolbox (OpenMMLab 2020). Follow previous works (Cheng, Schwing, and Kirillov 2021), we use ImageNet-1k to pretrain our model. During semantic segmentation training, we employ the widely used AdamW optimizer for all datasets to update the model parameters. For fair comparisons, our training parameters mainly follow the previous work (Xie et al. 2021). For the ADE20K and Cityscapes datasets, we adopt the default training iterations 160K in Segformer, where mini-batchsize is set to 16 and 8, respectively. For the COCO-stuff dataset, we set the training iterations to 80K and the mini-batch to 16. In addition, we implement data augmentation during training by random horizontal flipping, random resizing with a ratio of 0.5-2.0, and random cropping. We evaluate the results with mean Intersection over Union (mIoU) metric.

Comparisons with Existing Works

Results on ADE20K Dataset. We compare our AFFormer with top-ranking semantic segmentation methods. Following the inference settings in (Xie et al. 2021), we test FLOPs at 512×512 resolution and show the single scale

Model	#Param.	FLOPs	mIoU
FCN	9.8M	317G	61.5
PSPNet (MV2)	13.7M	423G	70.2
DeepLabV3+ (MV2)	15.4M	555G	75.2
SwiftNetRN	11.8M	104G	75.5
EncNet	55.1M	1748G	76.9
Segformer	3.8M	125G	76.2
AFFormer-tiny	1.6M(-58%)	23.0G(-82%)	76.5(+0.3)
AFFormer-small	2.3M(-41%)	26.2G(-79%)	77.6(+1.4)
AFFormer-base	3.0M(-21%)	34.4G(-73%)	78.7(+2.5)

Table 2: Comparison to state of the art methods on Cityscapes *val* set. The FLOPs are test on the resolution of 1024×2048. Meanwhile, we also report the percentage increase compared to Segformer.

Model	size	FLOPs	mIoU
Segformer	512×1024	17.7G	71.9
AFFormer-base	512×1024	8.6G(-51.4%)	73.5(+1.6)
Segformer	640×1280	31.5G	73.7
AFFormer-base	640×1280	13.4G(-57.5%)	75.6(+1.9)
Segformer	768×1536	51.7G	75.3
AFFormer-base	768×1536	19.4G(-62.5%)	76.5(+1.2)
Segformer	1024×2048	125G	76.2
AFFormer-base	1024×2048	34.4G(-72.5%)	78.7(+2.5)

Table 3: Speed-accuracy tradeoffs at different scales on Cityscapes.

Model	#Param.	FLOPs	mIoU
PSPNet (MV2)	13.7M	52.9G	30.1
DeepLabV3+ (MV2)	15.4M	25.9G	29.9
DeepLabV3+ (EN)	17.1M	27.1G	31.5
LR-ASPP (MV3)	–	2.37G	25.2
AFFormer-base	3.0M	4.6G	35.1

Table 4: Comparison to state of the art methods on COCO-stuff. We use a single-scale results at the input resolution of 512×512. MV3=MobileNetV3.

results in Table 1. Our model AFFormer-base improves by 5.2 mIoU under the same computing power consumption as Lite-ASPP, reaching 41.8 mIoU. Meanwhile, by reducing the number of layers and channels, we obtain AFFormer-tiny and AFFormer-small versions to adapt to different computing power scenarios. For the lightweight and efficient Segformer (8.4 GFLOPs),our base version (4.6 GFLOPs) also gain 4.4 mIoU using half the computing power and the tiny version (2.4 GFLOPs) with only 33% the computing power improving 1.3 mIoU. Only 1.8 GFLOPs are needed for the lighter topformer, but our base version has 2.1M less parameters (5.1M vs 3M) with 4.0 higher mIoU.

Results on Cityscapes Dataset. Table 2 shows the results of our model and the cutting-edge methods on Cityscapes. Although the Segformer is efficient enough, due to its square-level complexity, we only use 30% of the computational cost to reach 78.7 mIoU, which is 2.5 mIoU improvement with a 70% reduction in FLOPs. Meanwhile, we report the results at different high resolutions in Table 3. At the short side of {512, 640, 768, 1024}, the computational cost of our model is 51.4%, 57.5%, 62.5% and 72.5% of that of Segformer, respectively. Meanwhile, the mIoU are improved by 1.6, 1.9, 1.2 and 2.5, respectively. The higher the input resolution, the more advantageous of our model in both computational cost and accuracy.

Results on COCO-stuff Dataset. COCO-stuff dataset contains a large number of difficult samples that collected in COCO. As show in Table 4, although complex decoders (*e.g.*, PSPNet, DeepLabV3+) can achieve better results than LR-ASPP (MV3), they bring a lot of computational cost. Our model achieves an accuracy of 35.1 mIoU while only taking 4.5 GFLOPs, achieving the best trade-off.

Setting	#Param.	FLOPs	mIoU
w/o PD	2.78G	2.98G	39.2
w/o PL	0.42G	1.65G	19.5
Parallel	3.0G	4.6G	41.8

Table 5: Ablation studies on the parallel structure.

Setting	#Param.	FLOPs	mIoU
All PD	0.6M	1.85G	27.4
All PL	3.6M	7.0G	41.6
Heterogeneous	3.0M	4.6G	41.8

Table 6: Ablation studies on heterogeneous architecture.

Ablation Studies

All the ablation studies are conducted on ADE20K dataset with AFFormer-base unless otherwise specified.

Rationalization of Parallel Structures. Parallel architecture is the key to removing the decoder head and ensuring accuracy and efficiency. We first adjust the proposed structure to a naive pyramid architecture (denoted as "w/o PD") and a ViT architecture (denoted as "w/o PL") to illustrate the advantages of the parallel architecture. Specifically, the "w/o PD" means removing PD module and keeping only PL module, while the "w/o PL" does the opposite. As shown in Table 5, the setting "w/o PD" reduces 2.6 mIoU due to the lack of high-resolution pixel semantic information. The "w/o PL" structure without the pyramid structure has a significant reduction in accuracy due to few parameters and lack of rich image semantic information. It also demonstrates that our parallel architecture can effectively combine the advantages of both architectures.

Advantages of Heterogeneous Structure. The purpose of the heterogeneous approach is to further reduce the computational overhead. The PL module is adopted to learn the prototype representation in the clustered features, and then use PD to combine the original features for restoration, which avoids direct calculation on the high-resolution original features and reduce the computational cost. It can be seen from Table 6 that when the parallel branch is adjusted to the pixel description module (denote as "All PD"), which means

that the prototype representation is learned by PD module. The model size is only 0.6M, and the FLOPs are reduced by 2.5G, but the accuracy is reduced by 14.3 mIoU. This is due to the PD lacks the ability to learn great prototype representations. In contrast, after we replace the PD module with the PL module (denote as "All PL"), the FLOPs are increased by 2.4G, but there is almost no difference in accuracy. We believe that the PD module is actually only a simple way to restore the learned prototype, and the relatively complex PL module saturates the model capacity.

Advantages of Adaptive Frequency Filter. We use two datasets with large differences, including ADE20K and Cityscapes, to explore the core components in adaptive frequency filter module. The main reason is that the upper limit of the ADE20K dataset is only 40 mIoU, while the upper limit of the Cityscapes is 80 mIoU. The two datasets have different degrees of sensitivity to different frequencies. We report the benefits of each internal component in the Table 7. We find that DHF alone outperforms DLF, especially on the Cityscapes dataset by 2.6 mIoU, while FSK is significantly higher than DLF and DHF on ADE20K. This shows that ADE20K may be more inclined to an intermediate state between high frequency and low frequency, while Cityscapes needs more high frequency information. The combined experiments show that the combination of the advantages of each component can stably improve the results of ADE20K and Cityscapes.

Frequency Statistics Visualization. We first count the characteristic frequency distribution of different stages, as shown in Figure 5. It can be found that the curves of G_2 and F_2 almost overlap, indicating that the frequencies after clustering are very similar to those in the original features. The same goes for G_3 and F_3. Whereas, the learned prototype representation after frequency adaptive filtering significantly improves the contained frequency information. After PD restoration, different frequency components can be emphasized in different stages. As shwon in Figure 6, we also analyze the frequency effects of the core components in the AFF module. As expected, DLF and DHF show strong low-pass and high-pass capabilities, respectively, as FSK does. At the same time, we also found that the important frequency components screened and enhanced by FSK are mainly concentrated in the high frequency part, but the frequency signal is more saturated than that of DHF. This also shows that the high-frequency component part is particularly important in the semantic segmentation task, because it emphasizes more on the boundary details and texture differences between objects. Meanwhile, according to the analysis in Table 7 (the effects of ADE20K and Cityscapes have been steadily improved), each core component has its own advantages, and the AFF module shows strong robustness in various types and complex scenes.

Speed and Memory Costs. Meanwhile, we report the speed on the Cityscapes in Table 8. We can find that the proposed model improves by 10 FPS and performs much better than Segformer on such high-resolution Cityscapes images.

Setting	#Param.	FLOPs	ADE20K	Cityscapes
DLF	2.4M	3.6G	38.7	75.7
DHF	2.6M	3.9G	39.3	78.3
FSK	2.9M	4.2G	40.5	75.3
DLF + DHF	2.7M	3.9G	41.1	77.8
DLF + FSK	2.8M	4.2G	40.0	76.2
DHF + FSK	2.9M	4.3G	41.2	77.3
Whole	3.0M	4.6G	41.8	78.7

Table 7: Ablation studies on frequency aware statistics.

Model	FPS	mIoU
Segformer	12	76.2
AFFormer	22	78.7

Table 8: The FPS is tested on a V100 NVIDIA GPU with a batch size of 1 on the resolution of 1024x2048.

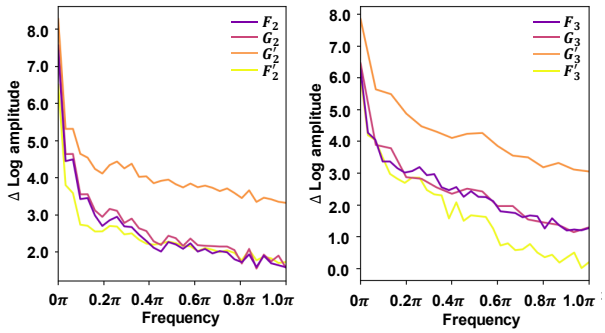

Figure 5: Frequency analysis of stage-2 (left) and stage-3 (right).

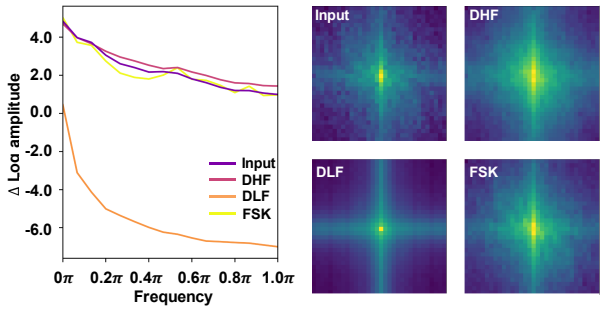

Figure 6: Frequency analysis of the core components in PL module.

Conclusion

In this paper, we propose a head-free lightweight semantic segmentation specific architecture. The core is to learn the local representation of the clustered prototypes from the frequency perspective, instead of directly learning all the pixel embedding features. It removes the complex decoder while having linear complexity Transformer and realizes semantic segmentation as simple as regular classification. Experiments demonstrate that the AFFormer owns powerful accuracy and great stability at low computational cost.

Acknowledgements

This work was supported by Alibaba Group through Alibaba Research Intern Program.

References

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1209–1218.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Chen, Y.; Dai, X.; Chen, D.; Liu, M.; Dong, X.; Yuan, L.; and Liu, Z. 2022. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5270–5279.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1290–1299.

Cheng, B.; Schwing, A.; and Kirillov, A. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34: 17864–17875.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heideman, M.; Johnson, D.; and Burrus, C. 1984. Gauss and the history of the fast Fourier transform. *IEEE ASSP Magazine*, 1(4): 14–21.

Lee, Y.; Kim, J.; Willette, J.; and Hwang, S. J. 2022. MPViT: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7287–7296.

Li, F.; Zhang, H.; Liu, S.; Zhang, L.; Ni, L. M.; Shum, H.-Y.; et al. 2022. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. *arXiv preprint arXiv:2206.02777*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Mehta, S.; and Rastegari, M. 2022. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *ICLR*.

OpenMMLab. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation.

Qian, Y.; Yin, G.; Sheng, L.; Chen, Z.; and Shao, J. 2020. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, 86–103. Springer.

Rao, Y.; Zhao, W.; Zhu, Z.; Lu, J.; and Zhou, J. 2021. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34: 980–993.

Ren, S.; Zhou, D.; He, S.; Feng, J.; and Wang, X. 2022. Shunted Self-Attention via Multi-Scale Token Aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10853–10862.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Strudel, R.; Garcia, R.; Laptev, I.; and Schmid, C. 2021. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7262–7272.

Wang, H.; Wu, X.; Huang, Z.; and Xing, E. P. 2020. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8684–8694.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 568–578.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2022. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3): 415–424.

Wu, K.; Peng, H.; Chen, M.; Fu, J.; and Chao, H. 2021. Rethinking and Improving Relative Position Encoding for Vision Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10033–10041.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34: 12077–12090.

Xu, N.; Price, B.; Cohen, S.; and Huang, T. 2017. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2970–2979.

Yuan, Y.; Fu, R.; Huang, L.; Lin, W.; Zhang, C.; Chen, X.; and Wang, J. 2021a. Hrformer: High-resolution vision transformer for dense predict. *Advances in Neural Information Processing Systems*, 34: 7281–7293.

Yuan, Y.; Huang, L.; Guo, J.; Zhang, C.; Chen, X.; and Wang, J. 2021b. OCNet: Object context for semantic segmentation. *International Journal of Computer Vision*, 129(8): 2375–2398.

Zhang, W.; Pang, J.; Chen, K.; and Loy, C. C. 2021. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34: 10326–10338.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhong, Y.; Li, B.; Tang, L.; Kuang, S.; Wu, S.; and Ding, S. 2022. Detecting Camouflaged Object in Frequency Domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4504–4513.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*.