SPANet: Frequency-balancing Token Mixer using Spectral Pooling Aggregation Modulation

Guhnoo Yun[1,2] Juhan Yoo[3] Kijung Kim[1,2] Jeongho Lee[1,2] Dong Hwan Kim[1,2]

[1]Korea Institute of Science and Technology
[2]Korea University [3]Semyung University
{doranlyong, plan100day, kape67, gregorykim}@kist.re.kr
unchinto@semyung.ac.kr

Abstract

Recent studies show that self-attentions behave like low-pass filters (as opposed to convolutions) and enhancing their high-pass filtering capability improves model performance. Contrary to this idea, we investigate existing convolution-based models with spectral analysis and observe that improving the low-pass filtering in convolution operations also leads to performance improvement. To account for this observation, we hypothesize that utilizing optimal token mixers that capture balanced representations of both high- and low-frequency components can enhance the performance of models. We verify this by decomposing visual features into the frequency domain and combining them in a balanced manner. To handle this, we replace the balancing problem with a mask filtering problem in the frequency domain. Then, we introduce a novel token-mixer named SPAM and leverage it to derive a MetaFormer model termed as SPANet. Experimental results show that the proposed method provides a way to achieve this balance, and the balanced representations of both high- and low-frequency components can improve the performance of models on multiple computer vision tasks. Our code is available at https://doranlyong.github.io/projects/spanet/.

1. Introduction

In recent years, Vision Transformers (ViTs) have achieved remarkable success and have garnered significant attention in the field of computer vision. As a result, numerous follow-up models based on the ViT [15] have been proposed, making ViTs a dominant architecture and a viable alternative to Convolutional Neural Networks (CNNs) in various computer vision tasks including image classification [55, 67, 34, 58], object detection [3, 80, 76], segmentation [61, 64, 8], and beyond [4, 75, 41, 62].

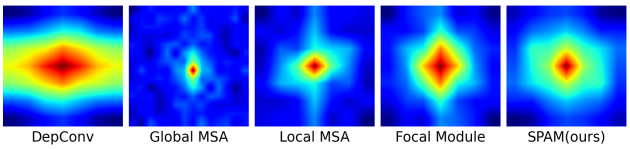

Figure 1: **Fourier spectrum maps of ConvNeXt and other MetaFormers.** The output of the spectrum map from each token mixer is processed for the same input. Depth-wise convolution (DepConv) of ConvNeXt-T [35], Global MSA of ViT-B/16 [15], Local MSA of Swin-T [34], Focal module of FocalNet-T [69], and SPAM of our SPANet-S are shown in order.

The reason for the success of ViT has been explained primarily as the use of Multi-Head Self-Attention (MSA) for token mixing [15]. This commonly held belief has led to the development of numerous variations of MSA [16, 20, 63, 79] aimed at improving the performance of ViTs. Yet some recent works have challenged this belief by demonstrating competitive results without utilizing MSAs. Tolstikhin *et al.* [52] fully replaced the MSAs with a spatial Multi-Layer Perceptron (MLP) and achieves comparable results on image classification benchmarks. Subsequent studies [24, 33, 54, 51] have attempted to reduce the performance gap between MLP-like models and ViTs by utilizing improved data-efficient training and redesigned MLP modules. These endeavors have shown the feasibility of MLP-like models to replace MSAs as token mixers. Moreover, other research lines [29, 38, 39, 46, 21] have explored alternative self-attention-based token mixers and reported encouraging results. For example, GFNet [46] replaces self-attention with Fourier Transform and achieves competitive performance to ViT in image classification tasks.

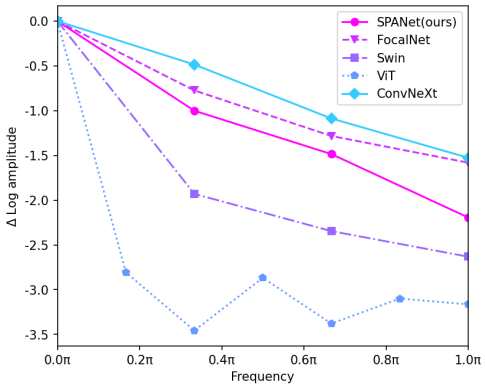

Figure 2: **Relative log amplitude of Fourier transformed feature maps.** DepConv of ConvNeXt-T and Global MSA of ViT-B/16 exhibit low-pass filtering that contains the most and the least high-frequency components, respectively. Conversely, Local MSA of Swin-T and Focal Module of FocalNet-T seem to capture both spectral components in a better balanced way.

There have also been works that aim to understand the fundamental differences between MSAs and convolution operations. The commonly accepted explanation for the efficacy of MSAs is their capacity to effectively capture long-range dependencies without imposing a strong inductive bias [15, 40, 57, 71, 37, 9] in contrast to convolution operations. In a recent study, however, Park *et al.* [43] explored the spectral filtering properties of both MSAs and convolutions and found that MSAs are closer to low-pass filtering, while convolution operations are better suited for filtering high-pass signals. The study also suggests that incorporating both operations in a specific sequence can lead to improved performance. Another study done by Bai *et al.* [2] investigates the adversarial robustness of MSAs and convolution operations by adding frequency perturbation and reached a similar conclusion. Moreover, the study proposes three training schemes to enhance the capture of high-frequency components by MSAs, leading to performance improvement of ViTs. That is, the model performance can be improved by enhancing the weak high-pass filtering capability of MSAs, or by using a token mixer optimized from a spectral filter perspective. Conversely, it can be expected that enhancing the low-pass filter capability of convolutions can also improve performance.

Figures 1 and 2 provide evidence supporting the expectation. Consistent with previous studies, depth-wise convolution (DepConv) is relatively more effective at capturing high-frequency signals compared to local- and global-MSA. On the other hand, the Focal Module [69] demonstrates better low-pass filtering capability, despite utilizing DepConv, and its performance also surpasses that of ConvNeXt [35], ViT [15], and Swin Transformer [34]. Collecting all these results together, we then naturally make such a hypothesis: *utilizing optimal token mixers that capture balanced representations of both high- and low-frequency components can enhance the performance of models*.

To verify this hypothesis, we employ the Discrete Fourier Transform (DFT) to decompose visual features into low- and high-frequency components. We then assign weights to tokens corresponding to each frequency band to balance low-frequency and high-frequency components in a way. To accomplish this, we replace the balancing problem with a mask filtering problem in the frequency domain and introduce a novel token-mixer called *spectral pooling aggregation modulation* (SPAM) module, which enables the balance of high- and low-frequency components. Using the SPAM token-mixer, we propose *SPANet* based on the MetaFormer architecture [72]. The performance of SPANet is evaluated on three benchmark computer vision tasks: image classification, object detection, and segmentation, and it demonstrates improved results compared to the previous state-of-the-art.

Our contributions are summarized as three-fold. (1) We handle the balancing problem of high- and low-frequency components of visual features, and show that it can be replaced with a mask filtering problem in the frequency domain. Specifically, we solve this problem by introducing SPAM. (2) Leveraging SPAM, we propose SPANet, which is based on the MetaFormer architecture [72]. (3) Our proposed SPANet is evaluated on multiple vision tasks, including image classification [14], object detection [32], instance segmentation [32], and semantic segmentation [78]. Our results show that SPANet outperforms state-of-the-art models.

2. Related Works

2.1. Transformers

Transformer has been first proposed in [59] for machine language translation which utilizes self-attention to learn representations of the input sequence that capture long-range dependencies and relationships between different language tokens. Thanks to its successful application in many natural language processing (NLP) tasks, the applicability of self-attention has been extended to the computer vision field. For instance, ViT [15] pioneered how to adopt a pure transformer architecture in image classification tasks and achieve excellent performance. Since the success of ViT, many follow-up works have been focusing on improving the MSA-based token mixers of ViTs through various approaches, such as shifted windows [34], relative position encoding [68], anti-aliasing attention map [45], or incorporating convolution [16, 19, 67], *etc*.

2.2. MetaFormers beyond Self-Attentions

Despite the widespread belief that the MSAs play an essential role in the success of ViTs, some recent studies have raised the question of whether it is the crucial element responsible for their high performance. For instance, it was found that MSAs can be entirely substituted with MLPs as token mixers [52, 54], while still achieving competitive performance relative to ViTs. This discovery sparked a discussion in the research community about which token mixer is better [7, 24] and several works challenged the dominance of attention-based token mixers by replacing MSAs with various approaches [29, 38, 39, 46]. Meanwhile, there have been other studies to explore transformers from the aspect of general architecture termed MetaFormer by replacing MSAs with non-parametric token mixers. ShiftViT [60] uses a partial shift operation [30] instead of MSAs, and PoolFormer [72] employs a spatial average pooling operator to replace MSAs. Both models achieve competitive performance on various computer vision tasks, suggesting that utilizing MetaFormer architecture can lead to reasonable performance. Building on this idea, we propose SPANet leveraging the advantage of MetaFormer architecture.

2.3. Frequency Domain Analysis

The frequency domain analysis has been extensively studied in the literature on computer vision. Normally, the low frequencies correspond to global structures and color information while the high frequencies correspond to fine details of objects (*e.g.*, local edges/textures) [11, 13]. According to [43, 2], MSAs highly tend to learn low-frequency representations in visual data but are weak for learning high-frequencies. On the other hand, convolutions exhibit the opposite behavior. Based on these observations, LITv2 [42] proposed a HiLo attention-mixer which captures both high- and low-frequency information with self-attention. Furthermore, Bai *et al.* [2] proposed HAT that enhances the ability of ViTs to capture high-frequency components using adversarial training. To the best of our knowledge, however, there has been no prior work aimed at enhancing CNNs in effectively capturing low-frequency components in visual data. Inspired by this, we introduce a new token-mixer called SPAM, which utilizes convolutional operation to efficiently capture both high- and low-frequency signals in a balanced manner.

3. Background

3.1. Feature Filtering in the Frequency Domain

Typically, there are two types of methods for image filtering. One is to perform a kernel convolution in the spatial domain and the other is to utilize the Discrete Fourier Transform (DFT) for filtering in the frequency domain. According to the convolution theorem [26], the results of visual feature processing in either the spatial domain or the frequency domain are equivalent. Yet transforming the features into the frequency domain allows for direct control of the spectral signals of features. Therefore, we adopt the frequency-based filtering method using the 2D DFT. This process is divided into three steps as follows.

Given a visual feature $\boldsymbol{x} \in \mathbb{R}^{H\times W\times D}$ as input, 2D DFT is used to transform it from the spatial domain to the frequency domain:

$$\boldsymbol{X}_c = \mathcal{F}(\boldsymbol{x}_c) \in \mathbb{C}^{H\times W}, \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes 2D DFT function, $\boldsymbol{x}_c \in \mathbb{R}^{H\times W}$ represents the c-th dimension of visual feature $\boldsymbol{x}$, and $\boldsymbol{X}_c$ is a complex tensor representing the spectrum of $\boldsymbol{x}_c$. We use `torch.fft.fft2` implemented by PyTorch library [44] to apply $\mathcal{F}(\cdot)$ to $\boldsymbol{x}_c$.

The desired frequency band is then modified by applying the Hadamard product (HP) with a weighting matrix to weight the spectrum:

$$\tilde{\boldsymbol{X}}_c = \boldsymbol{M} \odot \boldsymbol{X}_c, \tag{2}$$

where $\odot$ denotes the HP and $\boldsymbol{M}$ is an arbitrary weighting matrix that has the same size as $\boldsymbol{X}_c$.

Finally, the inverse DFT is applied to convert the modulated $\tilde{\boldsymbol{X}}_c$ back into the spatial domain and update the features:

$$\boldsymbol{x}_c \leftarrow \tilde{\boldsymbol{x}}_c = \mathcal{F}^{-1}(\tilde{\boldsymbol{X}}_c). \tag{3}$$

3.2. Focal Modulation

The focal modulation [69] is a new method that exploits depth-wise convolution to mimic the self-attention in a different way. This approach first aggregates context features, then interacts with visual tokens using the HP as:

$$\boldsymbol{y}^k = q(\boldsymbol{x}^k) \odot m(k, \boldsymbol{x}), \tag{4}$$

where $\boldsymbol{x}^k \in \mathbb{R}^D$ is visual token (query) at position k and $\boldsymbol{y}^k \in \mathbb{R}^D$ is refined representation. $q(\cdot)$ and $m(\cdot)$ are functions for query projection and context aggregation, respectively.

By observing Figures 1 and 2, the transformed feature of the focal modulation has a relatively more concentration of low-frequency signals compared to that of the DepConv. This result suggests that modulation with $m(\cdot)$ has a structural advantage for constructing a low-pass filter. Motivated by this, we leverage the focal modulation strategy described in Eq. 4 for our token-mixer design.

4. A Frequency-balancing Token Mixer

In this section, we introduce a novel context aggregation using convolutional modulation. Since convolution operations tend to relatively favor high-pass filtering [43], we aim to modulate the context features to concentrate relatively more on the low-pass signal for balance.

4.1. Spectral Pooling Gate (SPG)

For simplicity of implementation, we decompose a visual feature into a combination of low-pass (*lp*) and high-pass (*hp*) filters. That is, the low- and high-frequency components from the input visual features $\boldsymbol{x}$ are filtered out by pre-defined filters and then blended into one. This can be expressed in the following equation:

$$\tilde{\boldsymbol{x}}_c = \lambda_b f_{lp}(\boldsymbol{x}_c) + (1-\lambda_b) f_{hp}(\boldsymbol{x}_c) \in \mathbb{R}^{H\times W}, \quad (5)$$

where $\lambda_b \in [0,1]$ is a balancing parameter and $\tilde{\boldsymbol{x}}_c$ represents the filtered $\boldsymbol{x}_c$ by the combination of low- and high-pass filters.

Now the balance of the high- and low-frequency components can be controlled by manipulating the spectrum of the visual features by adjusting λ_b. For example, setting λ_b to 0.5, the output after normalization will be the same as the normalized input without any transformation.

4.1.1 Filtering with Spectral Pooling Filter (SPF)

Spectral pooling introduced by Rippel *et al.* [47] is a pooling technique that is used to reduce spatial tensor dimension by applying a low-pass filter. This is based on the inverse power law, which states that the expected power of natural images is statistically concentrated in the low-frequency region [53]. In other words, most of the important visual information in natural images is contained in the low-frequency part of the spectrum. Based on this, we design that low-frequency components are given greater weight compared to high-frequency components for frequency balancing. Also, it is general to preserve the input and output dimensions in traditional token-mixer designs. In the proposed spectral pooling scheme, therefore, filtering is applied while preserving the dimension.

The first step is to apply the 2D DFT to the input feature map and shift it so that the low-frequency components are located at the center(*i.e.*, the origin is set in the middle of the spectral map). For the low pass filter, f_{lp}, we select a low-frequency subset and remove the rest as follows:

$$\boldsymbol{S}_c^{lf} = \begin{cases} \mathcal{G}(\boldsymbol{X}_c)(u,v) & (u,v) \in \mathbf{A}^{lf} \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

where $\mathcal{G}(\cdot)$ is a function for centering the Fourier transform (we use `torch.fft.fftshift` implemented by PyTorch [44] library), (u,v) is a pair of positions for frequency-domain, and $\mathbf{A}^{lf} \in \mathbb{R}^2$ is a selected low-frequency region centered on the origin. Then, we obtain the spectral pooled feature map by applying the inverted shift and the inverse DFT:

$$f_{lp}(\boldsymbol{x}_c) = \mathcal{F}^{-1}(\mathcal{G}^{-1}(\boldsymbol{S}_c^{lf})) \in \mathbb{R}^{H\times W}. \quad (7)$$

The high-pass filter, f_{hp}, acts in the opposite manner to the low-pass filter and can be obtained by blocking or subtracting low-frequency components from the input feature map as follows:

$$\boldsymbol{S}_c^{hf} = \mathcal{G}(\boldsymbol{X}_c) - \boldsymbol{S}_c^{lf}, \quad (8)$$

where $\boldsymbol{S}_c^{hf} \in \mathbb{C}^{H\times W}$ is the high-frequency subset with the low-frequency area $\mathbf{A}^{lf}$ filled with zeros in $\mathcal{G}(\boldsymbol{X}_c)$. Subsequently, the inverse DFT is applied to the inverted shift of the high-frequency subset in a similar fashion as in Eq. 7 to obtain the high-pass filtered outcome:

$$f_{hp}(\boldsymbol{x}_c) = \mathcal{F}^{-1}(\mathcal{G}^{-1}(\boldsymbol{S}_c^{hf})) \in \mathbb{R}^{H\times W}. \quad (9)$$

4.1.2 Implementation of SPF using Mask Filtering

Since $\mathcal{F}$, $\mathcal{G}$, and those inverses are linear systems, they satisfy the superposition property. Therefore, Eq. 5 can be replaced by using Eq. 7 and Eq. 9 as follows:

$$\tilde{\boldsymbol{x}}_c = \mathcal{F}^{-1}(\mathcal{G}^{-1}(\lambda_b \boldsymbol{S}_c^{lf} + (1-\lambda_b)\boldsymbol{S}_c^{hf})). \quad (10)$$

In fact, the process to obtain spectral-pooled subsets ($\boldsymbol{S}_c^{lf}$ and $\boldsymbol{S}_c^{hf}$) by cropping the target band and filling the rest with zeros, can be easily achieved by masking the spectral map $\mathcal{G}(\boldsymbol{X}_c)$ with ideal binary masks using Eq. 2. The binary mask $\boldsymbol{M}^{lf}$ for obtaining $\boldsymbol{S}_c^{lf}$ is filled with ones in $\mathbf{A}^{lf}$ and zeros in the rest as follows:

$$\boldsymbol{M}^{lf} = \begin{cases} 1 & (u,v) \in \mathbf{A}^{lf} \\ 0 & \text{otherwise} \end{cases}. \quad (11)$$

Conversely, the binary mask $\boldsymbol{M}^{hf}$ is obtained with filling zeros in $\mathbf{A}^{lf}$ and ones in the rest:

$$\boldsymbol{M}^{hf} = \begin{cases} 0 & (u,v) \in \mathbf{A}^{lf} \\ 1 & \text{otherwise} \end{cases}. \quad (12)$$

Now the spectral-pooled subsets, $\boldsymbol{S}_c^{lf}$ and $\boldsymbol{S}_c^{hf}$, can be obtained by simple mask operation as follows:

$$\boldsymbol{S}_c^{lf} = \boldsymbol{M}^{lf} \odot \mathcal{G}(\boldsymbol{X}_c), \quad (13)$$

$$\boldsymbol{S}_c^{hf} = \boldsymbol{M}^{hf} \odot \mathcal{G}(\boldsymbol{X}_c). \quad (14)$$

Thus, $\lambda_b \boldsymbol{S}_c^{lf} + (1-\lambda_b)\boldsymbol{S}_c^{hf}$ can be described as below by applying Eq. 13 and Eq. 14:

$$(\lambda_b \boldsymbol{M}^{lf} + (1-\lambda_b)\boldsymbol{M}^{hf}) \odot \mathcal{G}(\boldsymbol{X}_c). \quad (15)$$

Any filter can be described by combining two or more ideal filters. In Eq. 15, $\lambda_b \boldsymbol{M}^{lf}$ means scaling the values in $\mathbf{A}^{lf}$ by λ_b, and $(1-\lambda_b)\boldsymbol{M}^{hf}$ means scaling the values

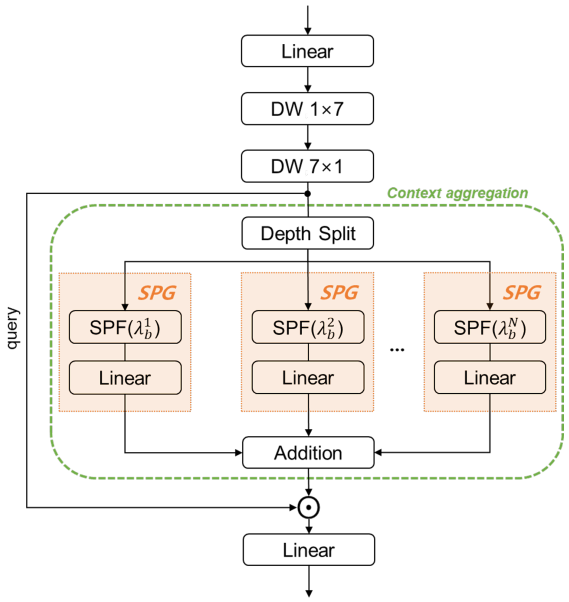

Figure 3: **Overview of the SPAM.** DW represents a depth-wise convolution and the block 'Linear' is implemented using a 1×1 convolution.

except $\mathbf{A}^{lf}$ by $(1-\lambda_b)$. For efficient mask operation, therefore, $\lambda_b \boldsymbol{M}^{lf} + (1-\lambda_b)\boldsymbol{M}^{hf}$ can be combined as a single mask:

$$\boldsymbol{M}^f = \begin{cases} \lambda_b & (u,v) \in \mathbf{A}^{lf} \\ 1-\lambda_b & \text{otherwise} \end{cases}, \tag{16}$$

where $\boldsymbol{M}^f \in \mathbb{R}^{H \times W}$ is the combination of $\boldsymbol{M}^{lf}$ and $\boldsymbol{M}^{hf}$. Therefore, Eq. 10 is simply rewritten as:

$$\tilde{\boldsymbol{x}}_c = \mathcal{F}^{-1}(\mathcal{G}^{-1}(\boldsymbol{M}^f \odot \mathcal{G}(\boldsymbol{X}_c)). \tag{17}$$

Finally, we need to define $\mathbf{A}^{lf}$ of Eq. 6 in detail. In the spectral pooling of Rippel *et al.* [47], $\mathbf{A}^{lf}$ is described as a rectangular shape. Generally, rectangular low-pass filtering, however, can result in artifacts or distortion in the output image. Therefore, we define $\mathbf{A}^{lf}$ as a circular shape:

$$\mathbf{A}^{lf}(u,v) = \{(u,v) | \sqrt{(u-u_0)^2 + (v-v_0)^2} < r\}, \tag{18}$$

where (u_0, v_0) indicates the origin of (u,v) pairs and r is a radius. That is, λ_b is assigned to the locations within radius r and $1-\lambda_b$ is assigned to the rest.

4.1.3 Feature Interaction

Applying the pre-defined filter in Section 4.1.2 uniformly to all feature dimensions is generally simplistic but limits the ability to reliably optimize representations by considering correlations between feature maps. In order to deal with this problem, Qian *et al.* [45] derived various and complex filters from the pre-defined filters with a linear assembling strategy with 1×1 convolutions. In this paper, we also apply the same scheme using Eq. 17 :

$$\boldsymbol{x}_i = \sum_{c=1}^{D} \phi_{i,c} \tilde{\boldsymbol{x}}_c, \tag{19}$$

where $\phi_{i,c} \in \mathbb{R}$ denotes the c-th learnable parameter of i-th kernel of 1×1 convolutions, and $\boldsymbol{x}_i \in \mathbb{R}^{H \times W}$ is a dynamically interacted feature map of $\tilde{\boldsymbol{x}} \in \mathbb{R}^{H \times W \times D}$.

As a result, SPG adjusts the high- and low-frequency components of all visual features using SPF of Eq. 17 and expresses complex and rich features utilizing Eq. 19, while optimizing the balance of frequency components. The overview of SPG is included in Figure 3.

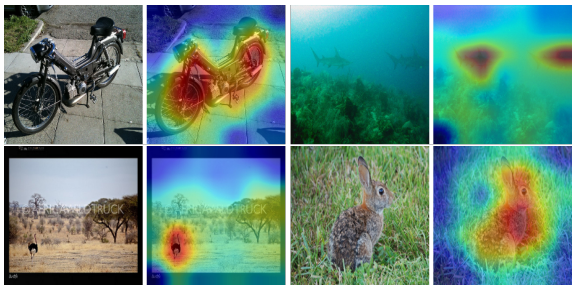

Figure 4: **Visualization of the context map.** The context map derived from the aggregated SPG features appropriately aligns with the object in the given image. This demonstrates that the aggregated context of SPAM can exhibit interpretable contextual features without self-attentions.

4.2. Spectral Pooling Aggregation Modulation

In this section, we propose a novel context aggregation using SPG. We then introduce a new token-mixer called *Spectral Pooling Aggregation Modulation* (SPAM) following the same strategy in Eq. 4. The overall structure is shown in Figure 3. Given a visual feature $\boldsymbol{x}$, it passes through a linear layer and depth-wise convolution for query projection. To reduce the number of parameters, spatial separable convolution [49] is adopted, which decompose $K \times K$ kernel into a pair of $1 \times K$ and $K \times 1$. In the context aggregation phase, N SPGs are utilized to aggregate filtered values by various balancing parameters. Each SPG receives a uniformly split projection map, and its output is aggregated by addition for context. The context map is shown in Figure 4. Then, the aggregated context is applied to the query for modulation. Finally, the modulated feature is passed through a linear layer for interaction.

Table 1: **Model configurations of SPANets.** C, L, and r mean embedding dimension, layer number (as known as depth), and radius of SPF in each stage, respectively. Each row describes each model variant for small, medium, and base denoted as S, M, and B, respectively.

Model	size	C	L	r
SPANet	S	64-128-320-512	4-4-12-4	2-2-1-1
	M	64-128-320-512	6-6-18-6	2-2-1-1
	B	96-192-384-768	6-6-18-6	2-2-1-1

4.3. SPANet Architectures

We adopt the same stage layouts and embedding dimensions as in the MetaFormer baseline [72] but replace the token-mixer parts with the proposed SPAM to construct a series of *SPAM Network* (SPANet) variants. In SPANets, we only need to specify the balancing parameters, λ_b, for each SPG, along with the radius, r, for the low-frequency band at each stage. The detailed configurations for each variant labeled as small, medium, and base are described in Table 1. Following the inverse power law [53], we assume λ_b should be larger than 0.5 to emphasize low-frequency components. Experimentally, we set N to 3, and λ_b of each SPG to 0.7, 0.8, and 0.9, respectively.

5. Experiments

Following common practices [72, 34, 70, 63], we conduct experiments to verify the effectiveness of the proposed SPANet on three tasks: image classification on ImageNet-1K [14], object detection and instance segmentation on COCO [32] and semantic segmentation on ADE20K [78]. Firstly, we evaluate the proposed SPANet architecture against the previous state-of-the-art on three tasks. In addition, the ablation study section analyzes the significance of the design elements of the proposed architecture. All experiments were implemented using PyTorch [44] on Ubuntu 20.04 with 4 NVIDIA RTX3090 GPUs.

5.1. Image Classification on ImageNet-1K

Implementation setup. For image classification, we evaluated SPANet on ImageNet-1K [14] which is one of the most widely cited benchmarks in the computer vision society. It comprises 1.28M training images and 50K validation images from 1K classes. Most of the training strategies are followed in [72] and [55]. The models are trained for 300 epochs at 224^2 resolution by AdamW optimizer [27, 36] with weight decay 0.05 and peak learning rate lr= $1e^{-3} \times \frac{\text{batch size}}{1024}$ (a batch size of 1024 and a learning rate of $1e^{-3}$ are used in this paper). The number of warmup epochs is 5 and a cosine decay learning rate scheduler is used. For data augmentation and regularization, MixUp [74], CutMix [73], CutOut [77], RandAugment [12], Label Smoothing [49] and Stochastic Depth [25] are used. Dropout is disabled but ResScale [48] for the last two stages is adopted to aid in training deep models. We employed Modified Layer Normalization (MLN) [72] to calculate the mean and variance along both visual token and channel dimensions, as opposed to only channel dimension in vanilla Layer Normalization [1]. MLN can be implemented using GroupNorm API in PyTorch [44] by setting the group number as 1. Our code implementation is based on Pytorch-image-models [65] and MetaFormer baseline [72].

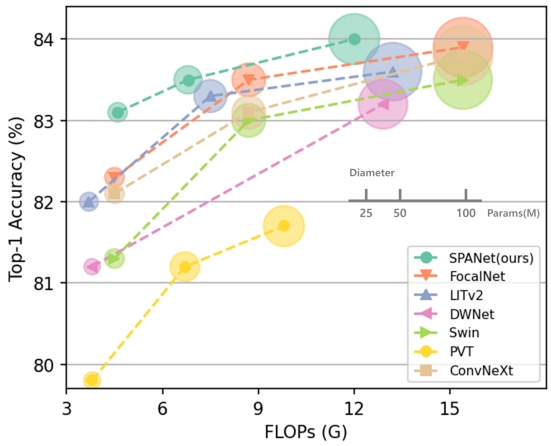

Figure 5: **ImageNet-1K validation accuracy vs. FLOPs/Params for SPANets and other comparative models.** The size of each bubble is proportional to the number of parameters in a variant within a model family.

Results. The performance of SPANets on ImageNet classification is presented in Table 2 and Figure 5. Our SPANets outperform others in terms of top-1 accuracy for small, medium, and base models when compared to the CNNs and other MetaFormers based on convolutions or self-attentions. In the case of the small model, SPANet-S achieves better performance than two state-of-the-art MetaFormers, namely LITv2-S and FocalNet-T, despite having a similar number of parameters and FLOPs. Specifically, it outperforms LITv2-S, which uses an attention-based mixer to handle both low and high frequencies, by $1.1\%p$, and FocalNet-T, which utilizes a modulated convolution-based mixer, by $0.8\%p$. In the medium model case, SPANet-M achieves the highest accuracy with the lowest number of FLOPs and parameters. Even compared to LITv2-M, it gains $0.2\%p$ top-1 accuracy. For CNNs, similar to comparison results on small and medium models, SPANets outperform ConvNeXts by $1.0\%p$, and $0.4\%p$, respectively. Similar results to those of the small and medium cases can also be observed for the base model case.

Table 2: **Performance comparison on ImageNet-1K [14] classification.** All models are trained from scratch on the ImageNet-1K training set and the accuracy on the validation set is reported. The numbers of FLOPs for input size 224^2 are counted by `fvcore` [17] library. The results of RSB-ResNet are from "ResNet Strikes Back" [66] which improves the ResNet model [23] with an optimized procedure for 300 epochs.

Model	General Arch.	Token Mixer	Params (M)	FLOPs (G)	Top-1 (%)
RSB-ResNet-50 [23, 66]	CNN	-	26	4.1	79.8
ConvNeXt-T [35]			29	4.5	82.1
PoolFormer-S24 [72]	MetaFormer	Pooling	21	3.4	80.3
PVT-Small [63]		Attention	25	3.8	79.8
Swin-T [34]			29	4.5	81.3
LITv2-S [42]			28	3.7	82.0
GFNet-H-S [46]		Convolution	32	4.6	81.5
DWNet-tiny [21]			24	3.8	81.2
FocalNet-T [69]			29	4.5	82.3
SPANet-S (ours)			29	4.6	**83.1**
RSB-ResNet-101 [23, 66]	CNN	-	45	7.9	81.3
ConvNeXt-S [35]			50	8.7	83.1
PoolFormer-M36 [72]	MetaFormer	Pooling	56	8.8	82.1
PVT-Medium [63]		Attention	44	6.7	81.2
Swin-S [34]			50	8.7	83.0
LITv2-M [42]			49	7.5	83.3
GFNet-H-B [46]		Convolution	54	8.6	82.9
FocalNet-S [69]			50	8.7	83.5
SPANet-M (ours)			42	6.8	**83.5**
RSB-ResNet-152 [23, 66]	CNN	-	60	11.6	81.8
ConvNeXt-B [35]			89	15.4	83.8
PoolFormer-M48 [72]	MetaFormer	Pooling	73	11.6	82.5
ViT-B/16 [15]		Attention	86	17.6	79.7
PVT-Large [63]			61	9.8	81.7
Swin-B [34]			88	15.4	83.5
LITv2-B [42]			87	13.2	83.6
DWNet-base [21]		Convolution	74	12.9	83.2
FocalNet-B [69]			89	15.4	83.9
SPANet-B (ours)			76	12.0	**84.0**

5.2. Object Detection and Instance Segmentation on COCO

Implementation setup. SPANet is evaluated based on COCO benchmark [32] which includes 118K training images (`train2017`) and 5K validation images (`val2017`). The models are trained on the training set, and the performance is reported on the validation set. SPANet is used as the backbone for two widely adopted detectors, namely RetinaNet [31] and Mask R-CNN [22]. ImageNet pre-trained weights are used to initialize the backbones, while Xavier initialization [18] is utilized to initialize the added layers. All models are trained using AdamW [27, 36] with an initial learning rate of $1e^{-4}$ and batch size of 8. Following common practices [31, 22], we adopted $1\times$ training schedule, which involves training the detection models for 12 epochs. The training images are resized to have a shorter side of 800 pixels, while the longer side is constrained to be at most 1,333 pixels. For testing, the shorter side of the images is also resized to 800 pixels. The implementation is based on the `mmdetection` [5] codebase.

Results. As shown in Table 3, SPANets equipped with RetinaNet [31] show competitive performances compared to their counterparts. For example, SPANet-S achieves 43.3AP, surpassing ResNet50 (36.3 AP), PVT-Small (40.4 AP), and Swin-T (41.5 AP), while obtaining competitive result to LITv2-S (43.7 AP). Similar results are also observed for SPANet-M. Moreover, these similar results also hold when equipped with Mask R-CNN [22].

5.3. Semantic Segmentation on ADE20K

Implementation setup. Following previous studies [63, 72], ADE20K [78] is selected to benchmark semantic segmentation, which requires an understanding of fine-grained details as well as an ability to analyze long-range interactions. The dataset consists of 20K training and 2K validation images, covering 150 fine-grained categories. We follow the evaluation approach of by employing SPANets as backbones equipped with Semantic FPN [28] and measuring model performance in terms of mIoU. ImageNet pre-trained weights are adopted to initialize the backbones, while Xavier [18] is used to initialize the newly added layers. Following common practices [28, 6], models are trained for 80K iterations with a batch size of 16. We employed the AdamW [27, 36] with an initial learning rate of $2e^{-4}$ that will decay following a polynomial decay schedule with a power of 0.9. Images are randomly resized and cropped into 512×512 for training and are rescaled on the shorter side of 512 pixels for testing. Our code implementation is based on the `mmsegmentation` [10] codebase.

Results. As shown in Table 4, equipped with Semantic FPN [28] for semantic segmentation, SPANet consistently outperforms other existing models. For instance, using nearly identical numbers of parameters and FLOPs, SPANet-S exhibits a $3.9\%p$ and $1.1\%p$ improvement in mIoU over Swin-T and LITv2-S, respectively. Similar results are also observed for the medium model case.

5.4. Ablation

This section presents ablation studies conducted on SPANet using ImageNet-1K [14]. The results of these studies are presented in Table 5 and are discussed below according to the following aspects.

SPAM components. To investigate the significance of the components that make up SPAM, we conduct experiments that involve altering the operators. In the first step, it is confirmed the SPF as an important element of SPG. The analysis reveals that the removal of this component results in a significant performance decrease, with accuracy dropping to 82.2%. Finally, we find the addition operator is better for context aggregation in SPAM. Our experimental result, shown in Table 5, indicates that replacing it with the HP leads to a decrease in performance to 82.7%.

Radius for low-pass band in each stage. The radius of the low-pass region for each stage is also an important factor affecting performance. As presented in Table 5, using $[1, 1, 1, 1]$ and $[4, 4, 1, 1]$ decrease the performances in $-0.1\%_p$ and $-0.2\%_p$, respectively. Therefore, [2,2,1,1] is adopted by default. However, it may not be optimal for

Table 3: **Performance of object detection with RetinaNet [31], and object detection and instance segmentation with Mask R-CNN [22] on COCO val2017 [32].** For training detection models, $1\times$ training schedule is adopted consisting of 12 epochs. The performance is reported in terms of bounding box AP and mask AP, denoted by AP^b and AP^m, respectively.

Backbone	RetinaNet 1×							Mask R-CNN 1×						
	Param (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Param (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 [23]	38	36.3	55.3	38.6	19.3	40.0	48.8	44	38.0	58.6	41.4	34.4	55.1	36.7
PVT-Small [63]	34	40.4	61.3	43.0	25.0	42.9	55.7	44	40.4	62.9	43.8	37.8	60.1	40.3
Swin-T [34]	39	41.5	62.1	44.2	25.1	44.9	55.5	48	42.2	64.6	46.2	39.1	61.6	42.0
LITv2-S [42]	38	**43.7**	-	-	-	-	-	47	**44.7**	-	-	**40.7**	-	-
SPANet-S (ours)	38	43.3	**63.7**	**46.5**	**25.8**	**47.7**	**57.0**	48	**44.7**	**65.7**	**48.8**	40.6	**62.9**	**43.8**
ResNet101 [23]	57	38.5	57.8	41.2	21.4	42.6	51.1	63	40.4	61.1	44.2	36.4	57.7	38.8
PVT-Medium [63]	54	41.9	63.1	44.3	25.0	44.9	57.6	64	42.0	64.4	45.6	39.0	61.6	42.1
Swin-S [34]	60	44.5	**65.7**	**47.5**	**27.4**	**48.0**	**59.9**	69	44.8	**66.6**	48.9	40.9	63.4	**44.2**
LITv2-M [42]	59	**45.8**	-	-	-	-	-	68	**46.5**	-	-	**42.0**	-	-
SPANet-M (ours)	51	44.0	64.3	47.0	25.9	**48.0**	58.7	61	45.2	66.3	**49.6**	41.0	**63.5**	44.0

Table 4: **Performance of semantic segmentation with Semantic FPN [28] on ADE20K [78].** The FLOPs are measured at the resolution of 512×512.

Backbone	Params (M)	FLOPs (G)	mIoU(%)
ResNet50 [23]	29	46	36.7
PVT-Small [63]	28	45	39.8
Swin-T [34]	32	46	41.5
LITv2-S [42]	31	41	44.3
SPANet-S (ours)	32	46	**45.4**
ResNet101 [23]	48	65	38.8
PVT-Medium [63]	48	61	41.6
Swin-S [34]	53	70	45.2
LITv2-M [42]	52	63	45.7
SPANet-M (ours)	45	57	**46.2**

Table 5: **Ablation for SPANet on ImageNet-1K [14] classification benchmark.** The number of parameters and FLOPs for all variants are the same, 29 and 4.6 respectively.

Ablation	Variant	Top-1(%)
-	SPANet-S-baseline	82.8
SPAM components	SPG with SPF → without SPF	82.2 (-0.6)
	aggregation with addition → with HP	82.7 (-0.1)
Radius for low-pass band in each stage	[2,2,1,1] → [1,1,1,1]	82.7 (-0.1)
	[2,2,1,1] → [4,4,1,1]	82.6 (-0.2)
Kernel size for spatial separable convolution	3 → 5	82.7 (-0.1)
	3 → 7	83.1 (+0.3)
Branch output scaling	ResScale [48] → None	82.7 (-0.1)
	ResScale [48] → LayerScale [56]	82.6 (-0.2)

SPANet and it is needed to explore optimal parameters to further improve performance in future work.

Kernel size for spatial separable convolution. To examine the kernel size of spatial separable convolution [50], we conducted an ablation study using kernels of sizes 3, 5, and 7. Our results indicate that increasing the kernel size from 3 to 7 improves the performance of SPANet from 82.8% to 83.1% while keeping the FLOPs and number of parameters roughly the same. However, we observed that enlarging the kernel from 3 to 5 leads to a decrease in performance. This can be explained by the fact that not all kernels can be split into two separate kernels, which restricts the exploration of all possible kernels and leads to sub-optimal during training. Consequently, we set the kernel size to 7 based on the outcomes of our experiments, *i.e.*, a pair of 1×7 and 7×1 convolutions is used by default.

Branch output scaling. The evaluation in the branch output scaling indicates that ResScale [48] is the most effective for SPANet. Notably, when using LayerScale [56], SPANet exhibits the lowest performance. In other words, we observed that LayerScale [56] has a negative impact on the training of SPANet.

6. Conclusion and Future Works

Discussion. In this work, we point out that existing effective token mixers show performance improvements by enhancing either the high- or low-pass filtering capabilities. Based on this, we show that models can be improved using a token mixer that balances of the high- and low-frequency components of the feature map.

To accomplish this, we replace the balancing problem with a mask filtering in the frequency domain and propose SPAM, a novel context aggregation mechanism that enables the optimal balance of high- and low-frequency components for visual features. With SPAM, we build a series of SPANets and evaluate them on three vision tasks. Our experimental results demonstrate that SPANets outperform the state-of-the-art CNNs and MetaFormers based on convolutions or self-attentions for image classification and semantic segmentation. Additionally, SPANets show competitive

performances for object detection and instance segmentation.

Limitations. SPANets exhibit limited performance improvements when applied to object detection and instance segmentation tasks. In such dense prediction tasks, identifying the fine-grained details of objects is important and this necessitates utilizing local edges and textures, which correspond to high-frequency components. However, the SPANet backbones, which are pre-trained with ImageNet-1K [14], relatively prioritize low-frequency components to balance frequency components following the Inverse Power Law [53]. Consequently, this design choice leads to sub-optimal performance.

In future work, we will further evaluate SPANets under more different vision tasks which require fine-grained features, such as pose estimation and fine-grained image classification. Moreover, it also requires the development of frequency-balancing token mixers tailored to task-specific characteristics.

Acknowledgment

This work was supported by the KIST Institutional Program (Project No. 2E32280 and 2E32282), and by the Technology Innovation Program and Industrial Strategic Technology Development Program (20018256, Development of service robot technologies for cleaning a table).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6

[2] Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving vision transformers by revisiting high-frequency components. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 1–18. Springer, 2022. 2, 3

[3] 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. 1

[4] Shuning Chang, Pichao Wang, Fan Wang, Hao Li, and Jiashi Feng. Augmented transformer with adaptive graph for temporal action proposal generation. *arXiv preprint arXiv:2103.16024*, 2021. 1

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 7

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 7

[7] Shoufa Chen, Enze Xie, Chongjian GE, Runjian Chen, Ding Liang, and Ping Luo. CycleMLP: A MLP-like architecture for dense prediction. In *International Conference on Learning Representations*, 2022. 3

[8] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1

[9] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021. 2

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 7

[11] James W Cooley, Peter AW Lewis, and Peter D Welch. The fast fourier transform and its applications. In *IEEE Transactions on Education*, volume 12, pages 27–34. IEEE, 1969. 3

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 6

[13] Guang Deng and LW Cahill. An adaptive gaussian filter for noise reduction and edge detection. In *1993 IEEE conference record nuclear science symposium and medical imaging conference*, pages 1615–1619. IEEE, 1993. 3

[14] 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. 2, 6, 7, 8, 9

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 7

[16] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021. 1, 2

[17] fvcore Contributors. fvcore. https://github.com/facebookresearch/fvcore, 2021. 7

[18] 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. 7

[19] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022. 2

[20] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 1

[21] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *International Conference on Learning Representations*, 2022. 1, 7

[22] 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. 7, 8

[23] 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. 7, 8

[24] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 2022. 1, 3

[25] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016. 6

[26] Yitzhak Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004. 3

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 7

[28] 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. 7, 8

[29] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States, July 2022. Association for Computational Linguistics. 1, 3

[30] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 3

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 7, 8

[32] 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. 2, 6, 7, 8

[33] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. In *NeurIPS*, 2021. 1

[34] 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 (ICCV)*, 2021. 1, 2, 6, 7, 8

[35] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 1, 2, 7

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6, 7

[37] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. 2

[38] André Martins, António Farinhas, Marcos Treviso, Vlad Niculae, Pedro Aguiar, and Mario Figueiredo. Sparse and continuous attention mechanisms. In *NeurIPS*, 2020. 1, 3

[39] Pedro Henrique Martins, Zita Marinho, and André FT Martins. ∞-former: Infinite memory transformer. In *Proc. ACL*, 2022. 1, 3

[40] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *NeurIPS*, 2021. 2

[41] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021. 1

[42] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast vision transformers with hilo attention. In *NeurIPS*, 2022. 3, 7, 8

[43] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2022. 2, 3

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 3, 4, 6

[45] Shengju Qian, Hao Shao, Yi Zhu, Mu Li, and Jiaya Jia. Blending anti-aliasing into vision transformer. In *NeurIPS*, 2021. 2, 5

[46] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *NeurIPS*, 2021. 1, 3, 7

[47] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 4, 5

[48] Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021. 6, 8

[49] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 2818–2826, 2016. 5, 6

[50] 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 (CVPR)*, June 2016. 8

[51] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. In *CVPR*, 2022. 1

[52] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021. 1, 3

[53] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: computation in neural systems*, 14(3):391, 2003. 4, 6, 9

[54] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 2022. 1, 3

[55] 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. 1, 6

[56] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021. 8

[57] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021. 2

[58] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 1

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017. 2

[60] Guangting Wang, Yucheng Zhao, Chuanxin Tang, Chong Luo, and Wenjun Zeng. When shift operation meets vision transformer: An extremely simple alternative to attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2423–2430, 2022. 3

[61] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5463–5474, 2021. 1

[62] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1571–1580, 2021. 1

[63] 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. 1, 6, 7, 8

[64] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[65] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 6

[66] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 7

[67] 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. 1, 2

[68] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041, 2021. 2

[69] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *NeurIPS*, 2022. 1, 2, 3, 7

[70] 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. In *NeurIPS*, 2021. 6

[71] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. Rethinking token-mixing mlp for mlp-based vision backbone. *arXiv preprint arXiv:2106.14882*, 2021. 2

[72] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 2, 3, 6, 7

[73] 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. 6

[74] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 6

[75] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. Token shift transformer for video classification. In *Proceedings of the*

29th ACM International Conference on Multimedia, pages 917–925, 2021. 1

[76] Minghang Zheng, Peng Gao, Renrui Zhang, Kunchang Li, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. In *arXiv preprint arXiv:2011.09315*, 2020. 1

[77] 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. 6

[78] 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. 2, 6, 7, 8

[79] Daquan Zhou, Yujun Shi, Bingyi Kang, Weihao Yu, Zihang Jiang, Yuan Li, Xiaojie Jin, Qibin Hou, and Jiashi Feng. Refiner: Refining self-attention for vision transformers. *arXiv preprint arXiv:2106.03714*, 2021. 1

[80] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1