SpectFormer: Frequency and Attention is what you need in a Vision Transformer

Badri Narayana Patro
Microsoft
badripatro@microsoft.com

Vinay P. Namboodiri
University of Bath
vpn22@bath.ac.uk

Vijay Srinivas Agneeswaran
Microsoft
vagneeswaran@microsoft.com

Abstract

*Vision transformers have been applied successfully for image recognition tasks. There have been either multi-headed self-attention based (ViT [14], DeIT, [53]) similar to the original work in textual models or more recently based on spectral layers (Fnet[29], GFNet[47], AFNO[17]). We hypothesize that both spectral and multi-headed attention plays a major role. We investigate this hypothesis through this work and observe that indeed combining spectral and multi-headed attention layers provides a better transformer architecture. We thus propose the novel Spectformer architecture for transformers that combines spectral and multi-headed attention layers. We believe that the resulting representation allows the transformer to capture the feature representation appropriately and it yields improved performance over other transformer representations. For instance, it improves the top-1 accuracy by 2% on ImageNet compared to both GFNet-H and LiT. SpectFormer-S reaches 84.25% top-1 accuracy on ImageNet-1K (state of the art for small version). Further, Spectformer-L achieves 85.7% that is the state of the art for the comparable base version of the transformers. We further ensure that we obtain reasonable results in other scenarios such as transfer learning on standard datasets such as CIFAR-10, CIFAR-100, Oxford-IIIT-flower, and Standford Car datasets. We then investigate its use in downstream tasks such of object detection and instance segmentation on MS-COCO dataset and observe that Spectformer shows consistent performance that is comparable to the best backbones and can be further optimized and improved. Hence, we believe that combined spectral and attention layers are what are needed for vision transformers. The project page is available at this webpage.*https://badripatro.github.io/SpectFormers/.

1. Introduction

Transformers originated in natural language processing with the seminal work by Vaswani *et al.* [57]. Transformers have gone on to revolutionize the language domain in the form of large language models such as GPT-3 and its variants (including chatGPT) [1], and Palm [9]. Subsequent work extended the concept of Transformer to computer vision and other domains. Interestingly, the main ideas explored retain much of the original transformer architecture. Clearly, the different domains could benefit from adapted transformers that are particular to the specific task. Through this work, we aim to specifically analyse the transformer for the image classification task using a vision transformer. We show that the proposed adaptation, Spectformer, can outperform the state-of-the-art for this task.

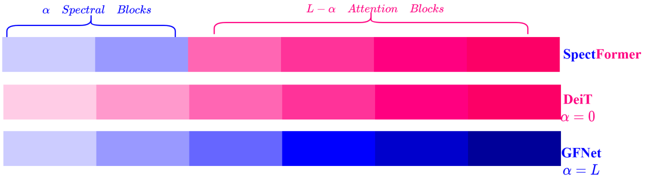


Figure 1. SpectFormer architecture consist of L Blocks, out of which α spectral blocks and $L-\alpha$ attention blocks.

The adaptation of transformers for computer vision was first explored in the Vision Transformer (ViT) [14]. They made the important contribution of developing appropriate patch-based tokenization for images whereby the transformer architecture could be used for images. DeIT [54] further improved the training process. The Fourier domain plays a major role in extracting frequency-based analysis of image information and has been well studied by the community. This is further supported by seminal work by Hubel and Weisel [25] that showed frequency tuned simple cells in the visual cortex. In transformers, it has been shown that the Fourier transforms could replace the multi-headed attention layers and achieve similar performance by Rao *et al.* [47] where they presented GFNet. They suggested that this approach captures fine-grained properties of images. This approach was further extended by AFNO [17], where they treated token mixing as operator learning. We hypothesize that for the image domain, both spectral and multi-headed self-attention plays an important role.

There have also been a number of hierarchical transformer architectures that have been explored in the literature [59, 10, 39]. One of the hierarchical approaches has been LiT [44] that uses less self-attention in the early layers, by using pure MLP (Multi-layer Perceptron) layers. They do use self-attention in deeper layers to capture longer depen-

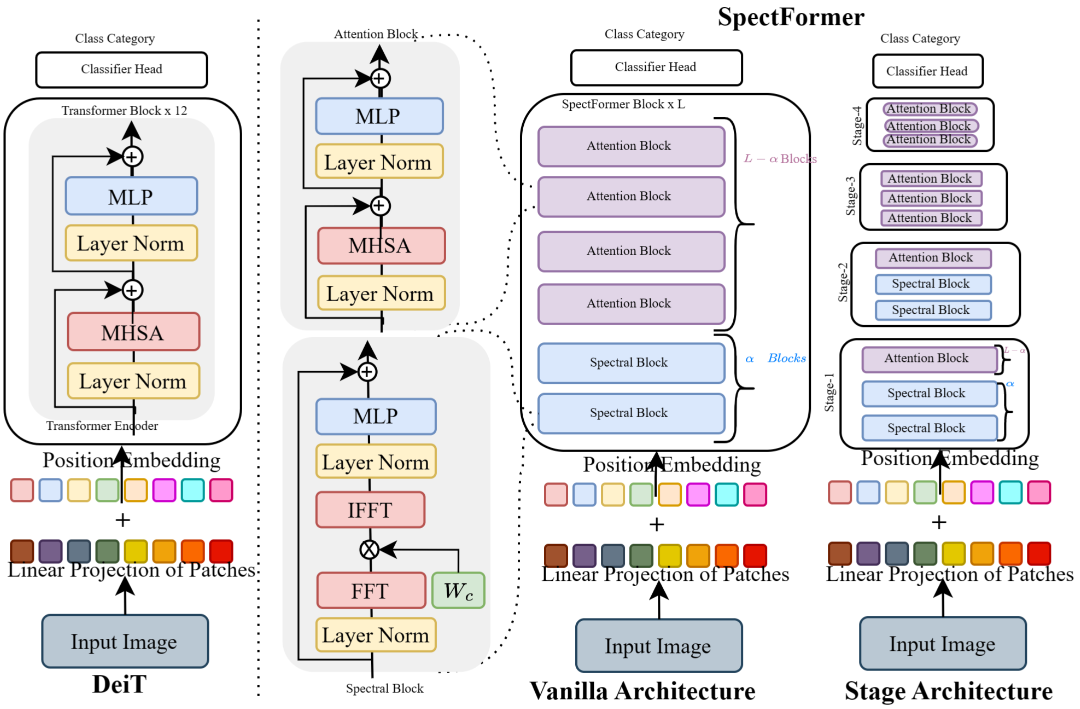


Figure 2. This figure shows Architectural details of SpectFormer. The first part shows the DeiT[53] architecture. The second part shows the vanilla and Stage architecture of the SpectFormer Model. This also shows the layer structure of Spectral and Attention Blocks.

dencies. Motivated by the works related to spectral and also hierarchical transformers, we developed SpectFormer, a new transformer architecture that uses spectral layers implemented with Fourier Transform to capture relevant features in the initial layers of the architecture. Further, we use multi-headed self-attention in the deeper layers of the network. The SpectFormer architecture is simple and transforms the image tokens to the Fourier domain, then applies gating techniques using learnable weight parameters, and finally does the inverse Fourier transform to get the signal back. Our approach combines both spectral and multi-headed attention as shown in figure-1.

Our multi-headed self-attention layer is similar to the original attention paper [57]. We show that SpectFormer achieves state-of-art performance compared to parallel architectures like LiT and outperforms complete spectral architectures like GFNet [47] and AFNO[17]. It also outperforms complete multi-headed attention-based transformers like DeIT on ImageNet 1K dataset. We outline our contributions below:

- We design Spectformer by using initial spectral layers and multi-headed attention in deeper layers. We validate the choice of this architecture through thorough empirical validation. For instance, the visualisation of the learned filters for the spectral layers are more localised as compared to similar fully spectral GFT [47]. The evidence suggests that adopting mixed spectral and later multi-headed attention results in improved results. We further validate this by comparing the proposed SpectFormer to a number of similar transformers such as LiT, vanilla transformers such as DeIT, spectral transformers such as GFNet, AFNO as well as hierarchical transformers such as PVT, Swin on the ImageNet dataset.

- We show that SpectFormer gets reasonable performance when used in transfer learning mode (trained on ImageNet and tested on CIFAR datasets) on CIFAR-10, and CIFAR-100 datasets.

- Further, we show that SpectFormer obtains consistent performance in other tasks such as object detection and instance segmentation by evaluating its performance on the MS COCO dataset.

2. Related Work

Quadratic Complexity of Attention Nets: The Vision Transformer (ViT) [14] model considers the image as a 16x16 word and is used to classify the image into predefined categories. In the ViT model, each image is split into a sequence of tokens of fixed length and then applied to multiple transformer layers to capture the global relationship

Table 1. Detailed architecture specifications for three variants of our SpectFormer with different model sizes, *i.e.*, SpectFormer-S (small size), SpectFormer-B (base size), and SpectFormer-L (large size). E_i, G_i, H_i, and C_i represent the expansion ratio of the feed-forward layer, the spectral gating number, the head number, and the channel dimension in each stage i, respectively.

	OP Size	SpectFormer-H-S	SpectFormer-H-B	SpectFormer-H-L
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} E_1 = 8 \\ G_1 = 1 \\ C_1 = 64 \end{bmatrix} \times 2$, $\begin{bmatrix} E_1 = 8 \\ H_1 = 2 \\ C_1 = 64 \end{bmatrix} \times 1$	$\begin{bmatrix} E_1 = 8 \\ G_1 = 1 \\ C_1 = 64 \end{bmatrix} \times 2$, $\begin{bmatrix} E_1 = 8 \\ H_1 = 2 \\ C_1 = 64 \end{bmatrix} \times 1$	$\begin{bmatrix} E_1 = 8 \\ G_1 = 1 \\ C_1 = 96 \end{bmatrix} \times 2$, $\begin{bmatrix} E_1 = 8 \\ H_1 = 3 \\ C_1 = 96 \end{bmatrix} \times 1$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} E_2 = 8 \\ G_2 = 1 \\ C_2 = 128 \end{bmatrix} \times 2$, $\begin{bmatrix} E_2 = 8 \\ H_2 = 4 \\ C_2 = 128 \end{bmatrix} \times 2$	$\begin{bmatrix} E_2 = 8 \\ G_2 = 1 \\ C_2 = 128 \end{bmatrix} \times 2$, $\begin{bmatrix} E_2 = 8 \\ H_2 = 4 \\ C_2 = 128 \end{bmatrix} \times 2$	$\begin{bmatrix} E_2 = 8 \\ G_2 = 1 \\ C_2 = 192 \end{bmatrix} \times 2$, $\begin{bmatrix} E_2 = 8 \\ H_2 = 6 \\ C_2 = 192 \end{bmatrix} \times 4$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	$\begin{bmatrix} E_3 = 4 \\ H_3 = 10 \\ C_3 = 320 \end{bmatrix} \times 6$	$\begin{bmatrix} E_3 = 4 \\ H_3 = 10 \\ C_3 = 320 \end{bmatrix} \times 12$	$\begin{bmatrix} E_3 = 4 \\ H_3 = 12 \\ C_3 = 384 \end{bmatrix} \times 18$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	$\begin{bmatrix} E_4 = 4 \\ H_4 = 14 \\ C_4 = 448 \end{bmatrix} \times 3$	$\begin{bmatrix} E_4 = 4 \\ H_4 = 16 \\ C_4 = 512 \end{bmatrix} \times 3$	$\begin{bmatrix} E_4 = 4 \\ H_4 = 16 \\ C_4 = 512 \end{bmatrix} \times 3$

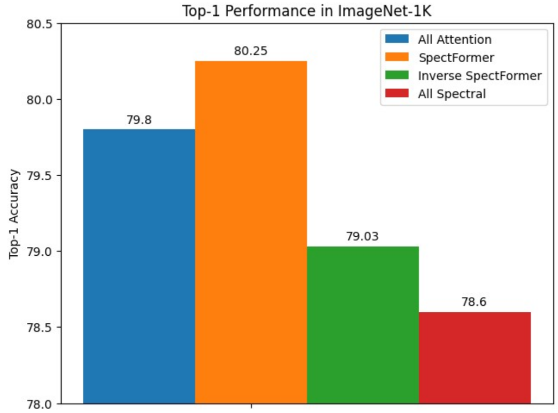


Figure 3. Initial comparison of attention vs spectral combinations.

across the token for the classification task. Touvron et al. [53] proposed an efficient transformer model based on distillation technique (DeiT). It uses a teacher-student strategy that relies on a distillation token to ensure that a student learns from a teacher through attention. Bao et al. [2] have proposed a masked image model task for a pretrained vision transformer. The vanilla transformer architecture which uses multi-headed self-attention (MSA) for efficient token mixing includes papers such as Tokens-to-token ViT [69], Transformer iN Transformer (TNT) [20], Cross-ViT [5], Class attention image Transformer(CaiT) [55] etc. The architectural complexity of most of the above transformers is O(N^2). Attempts at alleviating this include the Uniformer [30] which brings the best of convolutional nets and transformers by using multi-headed relation aggregation and RegionViT [4] as well as Token Pyramid Vision Transformer (TopFormer) [73]. A recent effort to address this complexity was by the paper [8] which used scaled element-wise embedding and an adapted-mask self-attention and enabled transformers to solve error-correcting codes. The complexity has also been mitigated by using spectral transformers, which typically have O(Nlog$N$) complexity. They also reduce the parameter count significantly compared to vanilla transformers.

Spectral Transformers Inspired by an MLP-mixer-based token mixing technique, recent work carried out using a spectral mixing technique in which the self-attention layer of the transformer is replaced by a non-parameterized Fourier transformation (Fnet) [29], which is then followed by a non-linearity and feed-forward network. This was followed by the Global Filter network (GFNet) [47], which uses a depth-wise global convolution for token mixing. Guibias et al. [17] formulated the token mixing task as an operator-learning task that learns mapping among continuous functions in infinite dimensional space using Fourier Neural Operator (FNO) [34]. In Wave-ViT [67], the author has discussed the quadratic complexity of the self-attention network of the transformer model with input patch numbers. They have proposed a wavelet vision transformer to perform lossless down-sampling using wavelet transform over keys and values of the self-attention network. The model obtains state-of-the-art results on image recognition, object detection, and instance segmentation tasks. Recently, another work [41] proposes a Fourier integral theorem characterize attention as non-parametric kernel regression and approximate key-query distributions. We compare the performance of the above methods with the proposed SpectFormer model.

Hierarchical Transformers Hierarchical architecture-based transformers are widely used to improve the performance of the transformer. It is a four-stage architecture first proposed by Wang et al. [59] in Pyramid vision Transformer (PVT). Later the stage-based architecture is used by SwinT [39], Twins [10] transformers. Through the work on LiT [44], Pan *et al.* have proposed an attention-based hierarchical transformer architecture that used to pay less attention in vision transformers using MLP layers. It must be noted that SpectFormer, which uses Fourier-based spectral layers is a more generalized and efficient mechanism to capture localized features that are fine-grained components of the image. The initial spectral layers improve the performance and achieve state-of-art compared to WaveViT [67] as mentioned in the recent efficient360 work[45]. We compare the performance of SpectFormer with recent works and observe that the proposed work outperforms all these other works.

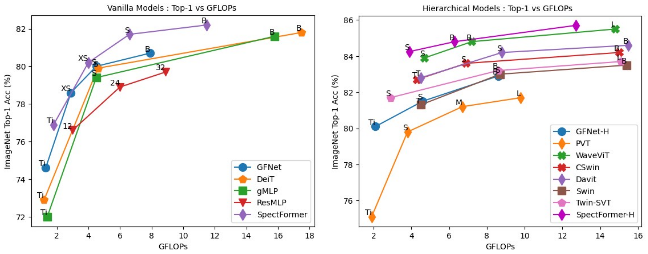


Figure 4. Comparison of ImageNet Top-1 Accuracy (%) vs GFLOPs of various models in Vanilla and Hierarchical architecture.

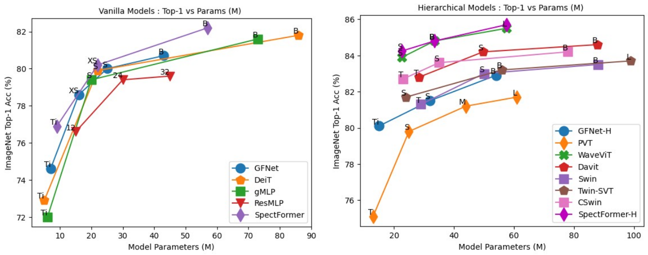


Figure 5. Comparison of ImageNet Top-1 Accuracy (%) vs Parameters (M) of various models in Vanilla and Hierarchical architecture.

Table 2. In this table, we present detailed configurations of various versions of SpectFormer for the vanilla transformer architecture. The table provides information on the number of heads, embedding dimensions, the number of layers in each variant, and the training resolution. For hierarchical SpectFormer-H models, we provide information in Table-1 of the main paper, which includes details for four stages. The FLOPs (floating-point operations) are calculated for both 224 × 224 and 384 × 384 input sizes. For the vanilla SpectFormer architecture, we use four spectral layers with $\alpha = 4$, while the remaining attention layers are equal to $(L - \alpha)$.

Model	#Layers	#heads	#Embedding Dim	Params (M)	Training Resolution	FLOPs (G)
SpectFormer-Ti	12	4	256	9	224	1.8
SpectFormer-XS	12	6	384	20	224	4.0
SpectFormer-S	19	6	384	32	224	6.6
SpectFormer-B	19	8	512	57	224	11.5
SpectFormer-XS	12	6	384	21	384	13.1
SpectFormer-S	19	6	384	33	384	22.0
SpectFormer-B	19	8	512	57	384	37.3

3. Method: Spectformer

3.1. Need for mixed spectral transformer

In order to validate whether there is a difference in performance in terms of representation if we consider a mixture of spectral and multi-headed attention layers, we did a study by considering the difference between basic all-attention, all-spectral and mixed spectral-attention layers with the spectral layers being placed initially or finally (that we term inverse SpectFormer). This analysis is provided in the following figure 3. As can be observed, the particular configuration where we use spectral layers initially followed by multi-headed attention layers is more beneficial. Thus, the need for Spectformer is evident. We therefore propose an architecture that includes initial spectral layers followed by multi-headed attention layers.

3.2. SpectFormer Architecture

The key idea of the SpectFormer architecture is illustrated in figure 2. As can be inferred from the figure, the SpectFormer architecture comprises a patch embedding layer, followed by a positional embedding layer, followed by a transformer block, and followed by a classification head (MLP - dim to 1000 projection). The transformer block comprises a series of spectral layers followed by attention layers. The image is split into a sequence of patches and we obtain a patch embedding using a linear projection layer. Our positional embedding uses a standard positional encoding layer. The transformer block will be explained below in two parts.

3.3. Spectral Block

The objective of the spectral layer is to capture the different frequency components of the image to comprehend localized frequencies. This can be achieved using a spectral gating network, that comprises a Fast Fourier Transform (FFT) layer, followed by a weighted gating, followed by an inverse FFT layer. The spectral layer converts physical space into the spectral space using FFT. We use a learnable weight parameter to determine the weight of each frequency component so as to capture the lines and edges of an image appropriately. The learnable weight parameter is specific to each layer of SpectFormer and is learnt using back-propagation techniques.

The spectral layer uses an inverse Fast Fourier Transform (IFFT) to bring back the spectral space back to the physical space. Following the IFFT, the spectral layer has layer normalization and Multi-Layer Perceptron (MLP) block for channel mixing, while token mixing is done using the spectral gating technique. Note that while we use the FFT/IFFT, the method can also be implemented using wavelet/inverse wavelet transform.

3.4. Attention Block

The attention layer of SpectFormer is a standard attention layer comprising layer normalization, followed by multi-headed self-attention (MHSA), followed by layer normalization and is followed by an MLP. The MHSA architecture is similar to DeIT attention architecture in that MHSA is used for token mixing and MLP is used for channel mixing in the attention layer.

3.5. SpectFormer Block

SpectFormer block has been illustrated in the figure 1, in the staged architecture. We introduce an alpha factor in the SpectFormer block, which controls the number of spectral layers and attention layers. If α=0, SpectFormer comprises all attention layers, similar to DeIT-s, while with an α value of 12, SpectFormer becomes similar to GFNet, with all spectral layers. It must be noted that all attention layers have the disadvantage that local features cannot be captured accurately. Similarly, all spectral layers have the disadvantage that global image properties or semantic features cannot be handled accurately. SpectFormer gives the flexibility to vary the number of spectral and attention layers, which helps in capturing both global properties as well as local features accurately. Consequently, as also ratified by our performance studies, SpectFormer considers local features, which help to capture localized frequencies in initial layers as well as global features in the deeper layers, which help capture long-range dependencies.

The above explanation is mainly for the vanilla SpectFormer architecture. We have also come up with a staged architecture, which comprises four stages, with each stage having a varying number of SpecfFormer blocks. Stage 1 has 3 SpectFormer blocks, while Stage 2 has 4, Stage 3 has 6 and Stage 4 has 3 SpectFormer blocks, as shown in table 1. In the stage of SpectFormer-s, there are 2 spectral layers and 1 attention layer, while Stage 2 comprises 2 spectral and 2 attention layers, to capture the local information. Stages 3 and 4 comprise only attention layers, to capture the semantic information. The details of SpectFormer-s architecture were explained above, while SpectFormer-B and SpectFormer-L are depicted in the table. We came up with several variants of the spectral layer including using FNet, FNO, GFNet and AFNO. We also provide a details SpectFormer architecture of the vanilla transformer model, presented in table-2.

4. Experiments and Results

Our proposed SpectFormer, is evaluated through various empirical evidence on a range of mainstream computer vision tasks, including image recognition, object detection, and instance segmentation. To compare the quality of learned feature representations obtained from SpectFormer, we conduct the following evaluations: (a) Conducting ablation studies that support each variant in our SpectFormer block and select best α value for it; (b) Training from scratch for image recognition task on ImageNet1K; (c) Transfer learning on CIFAR10, CIFAR-100, Oxford-IIIT flower, Standford Car dataset for Image recognition task using the SpectFormer (pre-trained on ImageNet1K) model; (d) Fine-tuning the SpectFormer (pre-trained on ImageNet1K) for downstream tasks such as object detection and instance segmentation on COCO; and (e) Visualizing the learned visual representation by SpectFormer. Through these evaluations, we demonstrate the effectiveness of our proposed SpectFormer model for computer vision tasks.

Table 3. This table shows the ablation analysis of various spectral layers in SpectFormer architecture such as the Fourier Network (FN), the Fourier Gating Network (FGN), the Wavelet Gating Network (WGN), and the Fourier Neural Operator (FNO). We conduct this ablation study on the small-size networks in stage architecture. This indicates that FGN performs better than other kinds of networks.

Model	Params (M)	FLOPs (G)	Top-1 (%)	Top-5 (%)
SpectFormer_FN	21.17	3.9	84.02	96.77
SpectFormer_FNO	21.33	3.9	84.09	96.86
SpectFormer_WGN	21.59	3.9	83.70	96.56
SpectFormer_FGN	22.22	3.9	**84.25**	**96.93**

4.1. Ablation analysis on spectral architectures

We conduct an experiment on the spectral network for the spectral layer in SpectFormer architecture as shown in figure-2. In the first study, we compare various spectral architectures to develop SpectFormer, such as the Fourier Network (FN), Fourier Gating Network (FGN), Fourier Neural Operator (FNO), and Wavelet Gating Network (WGN) as shown in table 3. The Fourier transform network indicates the spectral layer contains just a Fourier transform instead of a multi-headed self-attention network. Similarly, the Fourier gating network uses a Fourier transform and its contribution is controlled by learnable weight parameters, followed by the inverse Fourier transform. We use neural operator

Table 4. Ablation Analysis with different variants of SpectFormer architecture with variying alpha.

Model	Params (M)	FLOPs (G)	Top-1 (%)	Top-5 (%)
SpectFormer_α_0	22.00	4.6	79.80	-
SpectFormer_α_2	21.03	4.3	79.87	94.69
SpectFormer_α_4	20.02	4.0	**80.21**	94.76
SpectFormer_α_6	19.01	3.7	80.14	**94.85**
SpectFormer_α_8	18.00	3.4	79.55.	94.59
SpectFormer_α_{10}	16.99	3.1	79.06	94.62
SpectFormer_α_{12}	16.00	2.9	78.60	94.20
iSpectFormer_α_4	20.02	4.0	79.03	94.30

Table 5. This table shows the SpectFormer performance based on different model size. The first part shows results on vanilla architecture for Fourier Gating Network based model. The second part shows results for Hierarchical architecture indicated by 'H'. These results are for $\alpha = 4$.

Model	Params (M)	FLOPs (G)	Top-1 (%)	Top-5 (%)
SpectFormer-T	9.15	1.8	76.89	93.38
SpectFormer-XS	20.02	4.0	80.21	94.76
SpectFormer-S	32.56	6.6	81.70	95.64
SpectFormer-B	57.15	11.5	**82.12**	**95.75**
SpectFormer-H-FN-S	21.17	3.9	84.02	96.77
SpectFormer-H-FN-B	31.99	6.3	85.04	97.37
SpectFormer-H-WGN-S	22.59	3.9	83.7	96.56
SpectFormer-H-WGN-B	33.42	6.3	84.57	96.97
SpectFormer-H-S	22.22	3.9	84.25	96.93
SpectFormer-H-B	33.05	6.3	85.05	97.30
SpectFormer-H-L	54.67	12.7	**85.7**	**97.52**

techniques for channel mixing and Fourier transform techniques for token mixing similar to FNO paper [17]. Wavelet gating network uses a wavelet transform followed by learnable weight parameters to control the wavelet decomposition. We observe that the Fourier gating network outperforms all other architectures as it uses a gating technique to control the Fourier features. We have done ablation studies on the ImageNet-1K dataset to evaluate the performance of the SpectFormer architecture. We conduct this ablation study on the small-size networks in stage architecture as mentioned in table-1.

We also illustrate the performance differences in using the number of spectral layers(α) in the SpectFormer architecture as shown in figure-2. We select a vanilla transformer architecture similar to DeIT and we replace the number of attention layers with spectral layers. We choose Deit-Small [53] that has 12 layer architecture with a hidden dimension of 384 and a similar architecture in GFNet[47] is GFNet-XS. We characterize the study using a hyper-parameter α. We select the α value zero for the DeiT-S network and a value of twelve for the GFNet-XS transformer. We fine-tune the α value on ImageNet-1K dataset and find that the ideal α value is four. This result is captured in the table 4. We started with different α values such as 2, 4, 6, 8, and 10 for validating the DeIT small network where α_2 indicates two layers of spectral and ten (12-4) layers of attention in the architecture, while α_4 indicates four layers of spectral and eight (12-4) layers of attention network.

4.2. Comparison with Similar Architectures

We compared the vanilla architecture of SpectFormer to the hierarchical architecture of SpectFormer in two parts of the table-5. In the vanilla architecture, we developed tiny (SpectFormer-T), extra small (SpectFormer-XS), small (SpectFormer-S), and base (SpectFormer-B) models that are similar in layer count and hidden dimensions to GFNet [47], while the attention blocks are similar to Deit [53]. Similarly, in the hierarchical architecture, we developed small (SpectFormer-H-S), base (SpectFormer-H-B), and large (SpectFormer-H-L) models using the Fourier gating network. We also developed small and base models using the Fourier and wavelet gating networks, as shown in table-5. We observed that all the hierarchical models (SpectFormer-H-S, SpectFormer-H-B, and SpectFormer-H-L) performed better than the vanilla architecture and are state-of-the-art, as shown in table-7. We compared the performance of SpectFormer to similar architectures on the ImageNet-1k dataset as shown in table-6. We first compared SpectFormer to vanilla vision transformers such as DeiT [53], Fnet [29], FourierFormer [41], and GFNet [47], as well as hybrid trans-

Table 6. This shows a performance comparison of SpectForm with similar Transformer Architecture with different sizes of the networks on ImageNet-1K. $\star$ indicates additionally trained with the Token Labeling objective using MixToken[26].

Network	Params	GFLOPs	Top-1	Top-5
Vanilla Transformer Comparison				
DeiT-Ti [53]	5M	1.2	72.2	91.1
FourierFormer [41]	-	-	73.3	91.7
GFNet-Ti [47]	7M	1.3	74.6	92.2
SpectFormer-T	9M	1.8	**76.9**	**93.4**
DeiT-S [53]	22M	4.6	79.8	95.0
Fnet-S [29]	15M	2.9	71.2	-
GFNet-XS [47]	16M	2.9	78.6	94.2
GFNet-S [47]	25M	4.5	80.0	94.9
SpectFormer-XS	20M	4.0	**80.2**	**94.7**
SpectFormer-S	32M	6.6	**81.7**	**95.6**
DeiT-B [53]	86M	17.5	81.8	95.6
GFNet-B [47]	43M	7.9	80.7	95.1
SpectFormer-B	57M	11.5	**82.1**	**95.7**
Hierarchical Transformer Comparison				
PVT-S [59]	25M	3.8	79.8	-
Swin-T [39]	29M	4.5	81.3	-
GFNet-H-S [47]	32M	4.6	81.5	95.6
LIT-S [44]	27M	4.1	81.5	-
SpectFormer-H-S	21.7M	3.9	83.1	96.3
SpectFormer-H-S*	22M	3.9	**84.2**	**96.9**
PVT-M [59]	44M	6.7	81.2	-
Swin-S [39]	50M	8.7	83.0	-
GFNet-H-B [47]	54M	8.6	82.9	96.2
LIT-M [44]	48M	8.6	83.0	-
SpectFormer-H-B*	33M	6.3	**85.0**	**97.3**
PVT-L [59]	61M	9.8	82.3	-
Swin-B [39]	88M	15.4	83.3	-
LIT-B [44]	86M	15.0	83.4	-
SpectFormer-H-L*	55M	12.7	**85.7**	**97.5**

Table 7. The table shows the performance of various vision backbones on the ImageNet1K[11] dataset for image recognition tasks. $\star$ indicates additionally trained with the Token Labeling objective using MixToken[26] and a convolutional stem (conv-stem) [58] for patch encoding. We have grouped the vision models into three categories based on their GFLOPs (Small, Base, and Large). The GFLOP ranges:Small (GFLOPs$<$6), Base (6$\leq$GFLOPs$<$10), Large (10$\leq$GFLOPs$<$30).

Method	Params	GFLOPs	Top-1	Top-5	Method	Params	GFLOPs	Top-1	Top-5
Small					Large				
ResNet-50 [22]	25.5M	4.1	78.3	94.3	ResNet-152 [22]	60.2M	11.6	81.3	95.5
BoTNet-S1-50 [49]	20.8M	4.3	80.4	95.0	ResNeXt101 [64]	83.5M	15.6	81.5	-
Cross-ViT-S [5]	26.7M	5.6	81.0	-	gMLP-B [37]	73.0M	15.8	81.6	-
Swin-T [39]	29.0M	4.5	81.2	95.5	DeiT-B [53]	86.6M	17.6	81.8	95.6
ConViT-S [15]	27.8M	5.4	81.3	95.7	SE-ResNet-152 [24]	66.8M	11.6	82.2	95.9
T2T-ViT-14 [69]	21.5M	4.8	81.5	95.7	Cross-ViT-B [5]	104.7M	21.2	82.2	-
RegionViT-Ti+ [4]	14.3M	2.7	81.5	-	ResNeSt-101 [71]	48.3M	10.2	82.3	-
SE-CoTNetD-50 [33]	23.1M	4.1	81.6	95.8	ConViT-B [15]	86.5M	16.8	82.4	95.9
Twins-SVT-S [10]	24.1M	2.9	81.7	95.6	PoolFormer-M48 [68]	73.0M	11.8	82.5	-
CoaT-Lite Small [65]	20.0M	4.0	81.9	95.5	T2T-ViTt-24 [69]	64.1M	15.0	82.6	95.9
PVTv2-B2 [60]	25.4M	4.0	82.0	96.0	TNT-B [20]	65.6M	14.1	82.9	96.3
LITv2-S [43]	28.0M	3.7	82.0	-	CycleMLP-B4 [7]	52.0M	10.1	83.0	-
MViTv2-T [32]	24.0M	4.7	82.3	-	DeepViT-L [74]	58.9M	12.8	83.1	-
Wave-ViT-S [67]	19.8M	4.3	82.7	96.2	RegionViT-B [4]	72.7M	13.0	83.2	96.1
CSwin-T [13]	23.0M	4.3	82.7	-	CycleMLP-B5 [7]	76.0M	12.3	83.2	-
DaViT-Ti [12]	28.3M	4.5	82.8	-	ViP-Large/7 [23]	88.0M	24.4	83.2	-
SpectFormer-H-S	21.7M	3.9	83.1	96.3	CaiT-S36 [55]	68.4M	13.9	83.3	-
iFormer-S[48]	20.0M	4.8	83.4	96.6	AS-MLP-B [35]	88.0M	15.2	83.3	-
CMT-S [18]	25.1M	4.0	83.5	-	BoTNet-S1-128 [49]	75.1M	19.3	83.5	96.5
MaxViT-T [56]	31.0M	5.6	83.6	-	Swin-B [39]	88.0M	15.4	83.5	96.5
Wave-ViT-S* [67]	22.7M	4.7	83.9	96.6	Wave-MLP-B [51]	63.0M	10.2	83.6	-
SpectFormer-H-S*	**22.2M**	**3.9**	**84.3**	**96.9**	LITv2-B [43]	87.0M	13.2	83.6	-
Base					PVTv2-B4 [60]	62.6M	10.1	83.6	96.7
ResNet-101 [22]	44.6M	7.9	80.0	95.0	ViL-Base [72]	55.7M	13.4	83.7	-
BoTNet-S1-59 [49]	33.5M	7.3	81.7	95.8	Twins-SVT-L [10]	99.3M	15.1	83.7	96.5
T2T-ViT-19 [69]	39.2M	8.5	81.9	95.7	Hire-MLP-Large [19]	96.0M	13.4	83.8	-
CvT-21 [62]	32.0M	7.1	82.5	-	RegionViT-B+ [4]	73.8M	13.6	83.8	-
GFNet-H-B [47]	54.0M	8.6	82.9	96.2	Focal-Base [66]	89.8M	16.0	83.8	96.5
Swin-S [39]	50.0M	8.7	83.2	96.2	PVTv2-B5 [60]	82.0M	11.8	83.8	96.6
Twins-SVT-B [10]	56.1M	8.6	83.2	96.3	SE-CoTNetD-152 [33]	55.8M	17.0	84.0	97.0
SE-CoTNetD-101 [33]	40.9M	8.5	83.2	96.5	DAT-B [63]	88.0M	15.8	84.0	-
PVTv2-B3 [60]	45.2M	6.9	83.2	96.5	LV-ViT-M* [26]	55.8M	16.0	84.1	96.7
LITv2-M [43]	49.0M	7.5	83.3	-	CSwin-B [13]	78.0M	15.0	84.2	-
RegionViT-M+ [4]	42.0M	7.9	83.4	-	HorNet-B_{GF} [46]	88.0M	15.5	84.3	-
MViTv2-S [32]	35.0M	7.0	83.6	-	DynaMixer-L [61]	97.0M	27.4	84.3	-
CSwin-S [13]	35.0M	6.9	83.6	-	MViTv2-B [32]	52.0M	10.2	84.4	-
DaViT-S [12]	49.7M	8.8	84.2	-	DaViT-B [12]	87.9M	15.5	84.6	-
VOLO-D1* [70]	26.6M	6.8	84.2	-	CMT-L [18]	74.7M	19.5	84.8	-
CMT-B [18]	45.7M	9.3	84.5	-	MaxViT-B [56]	120.0M	23.4	85.0	-
MaxViT-S [56]	69.0M	11.7	84.5	-	VOLO-D2* [70]	58.7M	14.1	85.2	-
iFormer-B[48]	48.0M	9.4	84.6	97.0	VOLO-D3* [70]	86.3M	20.6	85.4	-
Wave-ViT-B* [67]	33.5M	7.2	84.8	97.1	Wave-ViT-L* [67]	57.5M	14.8	85.5	97.3
SpectFormer-H-B*	**33.1M**	**6.3**	**85.1**	**97.3**	**SpectFormer-H-L***	**54.7M**	**12.7**	**85.7**	**97.5**

formers such as PVT [59] and Swin [39] transformers. Compared to attention-based models like DeiT [53], SpectFormer performed better than DeiT in all size models (T, XS, S, and B). Compared to spectral-based models like Fnet [29], FourierFormer [41], and GFNet [47], SpectFormer performed better than all of them in all sizes (T, XS, S, and B). Then, we compared SpectFormer to hierarchical attention architectures such as PVT [59], Swin [39], LiT [44], and LiTv2[43]

Table 8. **Results on transfer learning datasets**. We report the top-1 accuracy on the four datasets as well as the number of parameters and FLOPs.

Model	CIFAR 10	CIFAR 100	Flowers 102	Cars 196
ResNet50 [22]	-	-	96.2	90.0
ViT-B/16 [14]	98.1	87.1	89.5	-
ViT-L/16 [14]	97.9	86.4	89.7	-
Deit-B/16 [53]	**99.1**	**90.8**	98.4	92.1
ResMLP-24 [52]	98.7	89.5	97.9	89.5
GFNet-XS [47]	98.6	89.1	98.1	92.8
GFNet-H-B [47]	99.0	90.3	98.8	93.2
Spectformer-B	98.9	90.3	**98.9**	**93.7**

and spectral architecture GFNet-H-S/B[47]. We have observed that SpectFormer outperforms vanilla transformers, hybrid transformers, other spectral transformers, and even other weighted attention transformers. SpectFormer performs 2% better than the latest similar model LiTv1 [44] for small architecture and 3% better than the best spectral architecture GFNet-H-S[47]. When compared to DeiT [53], SpectFormer also performs better.

4.3. Image Classification task on ImageNet-1K

Dataset and Training Setups We describe the training process of the image recognition task using the ImageNet1K benchmark dataset, which includes 1.28 million training images and 50K validation images belonging to 1,000 categories. The vision backbones are trained from scratch using data augmentation techniques like RandAug, CutOut, and Token Labeling objectives with MixToken. The performance of the trained backbones is evaluated using both top-1 and top-5 accuracies on the validation set. The optimization process involves using the AdamW optimizer with a momentum of 0.9, 10 epochs of linear warm-up, and 310 epochs of cosine decay learning rate scheduler. The batch size is set to 128 and is distributed on 8 A100 GPUs. The learning rate and weight decay are fixed at 0.00001 and 0.05, respectively.

Table-7 presents a comparison of the performance of the state-of-the-art vision models and our SpectFormer variants. The ViT backbones with the best performance, VOLO-D1*, VOLO-D2*, and VOLO-D3*, are trained using additional strategies such as Token Labeling objective with MixToken and convolutional stem for better patch encoding. We also use these strategies to train our SpectFormer variants in each size, which are denoted as SpectFormer-S*, SpectFormer-B*, and SpectFormer-L*. For a fair comparison, we degraded the version of SpectFormer in Small size without token labeling objective and which is called SpectFormer-S and its top-1 accuracy is 83.1 which is better than Wave-ViT-s (82.7, without extra token).

The table shows that our Wave-ViT variants consistently outperform existing vision models, including ResNet, SE-ResNet, Vanilla ViTs (TNT, CaiT, CrossViT), and hierarchical ViTs (Swin, Twins-SVT, PVTv2, VOLO), under similar GFLOPs for each group. In particular, under the Base size, the Top-1 accuracy score of SpectFormer-B* can reach 85.1%, which leads to the absolute improvement of 0.3% against the best competitive Wave-ViT-B* (Top-1 accuracy: 84.8%). Under the Large size, when compared to ResNet-152 and SE-ResNet-152, which solely rely on CNN architectures, vanilla ViTs (TNT-B, CaiT-S36, and CrossViT) capture long-range dependencies through Transformer structure and outperform them. However, the performances of CaiT-S36 and CrossViT are still lower than most hierarchical ViTs (PVTv2-B5, VOLO-D3*, CMT-L, MaxViT, DaViT and Wave-ViT-L) that aggregate multi-scale contexts. Moreover, unlike PVTv2-B5, which uses irreversible down-sampling for self-attention learning, Wave-ViT uses invertible down-sampling with wavelet transforms for self-attention learning. In particular, under the large size, the Top-1 accuracy score of SpectFormer-L* can reach 85.7%, which leads to the absolute improvement of 0.2% against the best competitive Wave-ViT-L* (Top-1 accuracy: 85.5%). Our SpectFormer-L* achieves better efficiency-vs-accuracy trade-off by enabling initial spectral blocks in the transformer encoder and attention blocks are at the top blocks. Overall, these findings demonstrate the efficacy of spectral block along with attention block in enhancing visual representation learning.

4.4. Task Learning: Object Detection

Training setup: In this section, we examine the pre-trained SpectFormer-H-small behavior on COCO dataset for two downstream tasks that localize objects ranging from bounding-box level to pixel level, *i.e.*, object detection and instance segmentation. Two mainstream detectors, *i.e.*, RetinaNet [36] and Mask R-CNN[21] as shown in table-8 of the main paper, and two state-of-the-art detectors *i.e.*, GFL [31], and Cascade Mask R-CNN [3] in mmdetection [6] in this supplementary doc. We are employed for each downstream task, and we replace the CNN backbones in each detector with our SpectFormer-H-small for evaluation. Specifically, each vision backbone is first pre-trained over ImageNet1K, and the newly added layers are initialized with Xavier [16]. Next, we follow the standard setups in [39] to train all models on the COCO train2017 ($\sim$118K images). Here the batch size is set as 16, and AdamW [40] is utilized for optimization (weight decay: 0.05, initial learning rate: 0.0001, betas=(0.9, 0.999)). We used learning rate (lr) configuration with step lr policy, linear warmup at every 500 iterations with warmup ration 0.001. All models are finally evaluated on the COCO val2017 (5K images). For state-of-the-art models like GFL [31], and Cascade Mask R-CNN [3], we utilize 3 × schedule (*i.e.*, 36 epochs) with the multi-scale strategy for training, whereas for RetinaNet [36] and Mask R-CNN[21] we utilize 1 × schedule (*i.e.*, 12 epochs).

Table 9. The performances of various vision models on the COCO val2017 dataset for the downstream tasks of object detection and instance segmentation. RetinaNet is used as the object detector for the object detection task, and the Average Precision (AP) at different IoU thresholds or two different object sizes (*i.e.*, small and base) are reported for evaluation. For instance segmentation task, we adopt Mask R-CNN as the base model, and the bounding box and mask Average Precision (*i.e.*, AP^b and AP^m) are reported for evaluation.

Backbone	Mask R-CNN 1x [21]						RetinaNet 1x [36]					
	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet50 [22]	38.0	58.6	41.4	34.4	55.1	36.7	36.3	55.3	38.6	19.3	40.0	48.8
Swin-T [39]	42.2	64.6	46.2	39.1	61.6	42.0	41.5	62.1	44.2	25.1	44.9	55.5
Twins-SVT-S [10]	43.4	66.0	47.3	40.3	63.2	43.4	43.0	64.2	46.3	28.0	46.4	57.5
LITv2-S [43]	44.9	-	-	40.8	-	-	44.0	-	-	-	-	-
RegionViT-S [4]	44.2	-	-	40.8	-	-	43.9	-	-	-	-	-
PVTv2-B2 [60]	45.3	67.1	49.6	41.2	64.2	44.4	**44.6**	**65.6**	**47.6**	**27.4**	**48.8**	58.6
SpectFormer-S-FN	**46.2**	**68.1**	**50.8**	**42.0**	**65.2**	**45.4**	44.2	64.8	47.3	27.3	48.1	**59.5**
ResNet101 [22]	40.4	61.1	44.2	40.4	61.1	44.2	38.5	57.8	41.2	21.4	42.6	51.1
Swin-S [39]	44.8	66.6	48.9	40.9	63.4	44.2	44.5	65.7	47.5	27.4	48.0	59.9
Twins-SVT-B [10]	45.2	67.6	49.3	41.5	64.5	44.8	45.3	66.7	48.1	28.5	48.9	60.6
RegionViT-B [4]	45.4	-	-	41.6	-	-	44.6	-	-	-	-	-
LITv2-M [43]	46.8	-	-	42.3	-	-	46.0	-	-	-	-	-
PVTv2-B3 [60]	47.0	68.1	51.7	42.5	65.7	45.7	45.9	**66.8**	49.3	28.6	**49.8**	**61.4**
SpectFormer-B-FN	46.9	**68.8**	**51.8**	**42.7**	**65.9**	**45.7**	**46.0**	66.4	**49.7**	**29.5**	49.7	61.1

Table 10. We conducted a comparison of various transformer-style architectures for image classification on ImageNet. This includes **vision transformers [53], MLP-like models [52, 37], spectral transformers [47] and our SpectFormer models**, which have similar numbers of parameters and FLOPs. The top-1 accuracy on ImageNet's validation set, as well as the number of parameters and FLOPs, are reported. All models were trained using 224 × 224 images. We used the notation ”↑384” to indicate models fine-tuned on 384 × 384 images for 30 epochs.

Model	Params (M)	FLOPs (G)	Resolution	Top-1 Acc. (%)	Top-5 Acc. (%)
gMLP-Ti [37]	6	1.4	224	72.0	-
DeiT-Ti [53]	5	1.2	224	72.2	91.1
GFNet-Ti [47]	7	1.3	224	74.6	92.2
SpectFormer-T	9	1.8	224	76.8	93.3
ResMLP-12 [52]	15	3.0	224	76.6	-
GFNet-XS [47]	16	2.9	224	78.6	94.2
SpectFormer-XS	20	4.0	224	80.2	94.7
DeiT-S [53]	22	4.6	224	79.8	95.0
gMLP-S [37]	20	4.5	224	79.4	-
GFNet-S [47]	25	4.5	224	80.0	94.9
SpectFormer-S	32	6.6	224	81.7	95.6
ResMLP-36 [52]	45	8.9	224	79.7	-
GFNet-B [47]	43	7.9	224	80.7	95.1
gMLP-B [37]	73	15.8	224	81.6	-
DeiT-B [53]	86	17.5	224	81.8	95.6
SpectFormer-B	57	11.5	224	**82.1**	**95.7**
GFNet-XS↑384 [47]	18	8.4	384	80.6	95.4
GFNet-S↑384 [47]	28	13.2	384	81.7	95.8
GFNet-B↑384 [47]	47	23.3	384	82.1	95.8
SpectFormer-XS↑384	21	13.1	384	82.1	95.7
SpectFormer-S↑384	33	22.0	384	83.0	96.3
SpectFormer-B↑384	57	37.3	384	82.9	96.1

Table 11. This table presents information about datasets used for transfer learning. It includes the size of the training and test sets, as well as the number of categories included in each dataset.

Dataset	CIFAR-10 [28]	CIFAR-100 [28]	Flowers-102 [42]	Stanford Cars [27]
Train Size	50,000	50,000	8,144	2,040
Test Size	10,000	10,000	8,041	6,149
#Categories	10	100	196	102

Table 12. The performances of various vision backbones on COCO val2017 dataset for the downstream task of object detection. Four kinds of object detectors, *i.e.*, GFL [31], and Cascade Mask R-CNN [3] in mmdetection [6], are adopted for evaluation. We report the bounding box Average Precision (AP^b) in different IoU thresholds.

Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}
ResNet50 [22]	GFL [31]	44.5	63.0	48.3
Swin-T [39]		47.6	66.8	51.7
PVTv2-B2 [60]		50.2	69.4	54.7
SpectFormer-H-S-FN		**50.3**	**70.0**	**55.2**
ResNet50 [22]	Cascade Mask [3] R-CNN	46.3	64.3	50.5
Swin-T [39]		50.5	69.3	54.9
PVTv2-B2 [60]		51.1	69.8	55.3
SpectFormer-H-S-FN		**51.5**	**70.2**	**56.3**

We conducted experiments on MS COCO 2017, which is a widely used benchmark for object detection and instance segmentation, comprising around 118K images for the training set and approximately 5K images for the validation set. Our approach involved experimenting with two detection frameworks, namely RetinaNet [36] and Mask R-CNN[21], and we measured model performance using Average Precision (AP). We use the pre-trained model SpectFormer trained on the ImageNet-1K dataset to initialize the backbone architecture and Xavier initialization for additional layers of the network. These results are shown in table- 9. The experimental results, as presented in table- 9, indicate that SpectFormer has comparative results on both the RetinaNet [36] and Mask R-CNN[21] models. We have compared with the latest work including LITv2 [43], RegionViT [4], and PVT [59] transformer models. Further, our SpectFormer model demonstrated significantly better performance than ResNet in terms of AP. More importantly, SpectFormer outperformed all compared vanilla ViT models and hierarchical transformer models, achieving the best AP performance. We compared the performance of SpectFormer with other hierarchical models such as Swin-T [38] and Pyramid Vision Transformer (PVT) [60] for two state-of-the-art object detection tasks. The results are presented in Table 12, and demonstrate that SpectFormer outperforms other SOTA transformer-based models for these object detection models. Additionally, we report the object detection performance of GFL[31] model and Cascade Mask R-CNN [3] on MS COCO val2017 dataset, which demonstrates an improvement in performance, as shown in Table-12.

4.5. Transfer Learning Comparison

Training setup: To test the effectiveness of our architecture and learned representation, we evaluated vanilla SpectFormer on commonly used transfer learning benchmark datasets, including CIFAR-10 [28], CIFAR100 [28], Oxford-IIIT-Flower [42] and Standford Cars [27]. Our approach followed the methodology of previous studies [50, 14, 53, 52, 47], where we initialized the model with ImageNet pre-trained weights and fine-tuned it on the new datasets. In table-7 of the main paper, we have presented a comparison of the transfer learning performance of our basic and best models with state-of-the-art CNNs and vision transformers. The transfer learning setup employs a batch size of 64, a learning rate (lr) of 0.0001, a weight-decay of 1e-4, a clip-grad of 1, and warmup epochs of 5. We have utilized a pre-trained model trained on the Imagenet-1K dataset, which we have fine-tuned on the transfer learning dataset specified in table-8 for 1000 epochs. Table-11 displays the dataset information used for transformer learning.

In order to assess the effectiveness of SpectFormer's architecture and learned representation, we conducted evaluations on multiple transfer learning benchmark datasets, which included CIFAR-10 [28], CIFAR-100 [28], Stanford Cars [27], and Flowers-102 [42]. Here we compare the performance of SpectFormer pre-trained on ImageNet-1K and fine-tuned on the new datasets for the image classification task. Both the basic and best models were evaluated for their transfer learning performance, and the comparison is captured in table- 8. The results show that the proposed models performed well on downstream datasets, surpassing ResMLP models by a significant margin and achieving highly competitive performance comparable to state-of-the-art spectral network, GFNet[47]. Our models also exhibited competitive performance when compared to state-of-the-art CNNs and vision transformers.

4.6. Model Fine-tuning for High Resolution input

Our main experiments are conducted on ImageNet [11], a popular benchmark for large-scale image classification. To ensure a fair comparison with previous research [53, 52, 47], we adopt the same training details for our SpectFormer models. For the vanilla transformer architecture (SpectFormer), we use the hyper-parameters recommended by the GFNet implementation [47]. For the hierarchical architecture (SpectFormer-H), we use the hyper-parameters

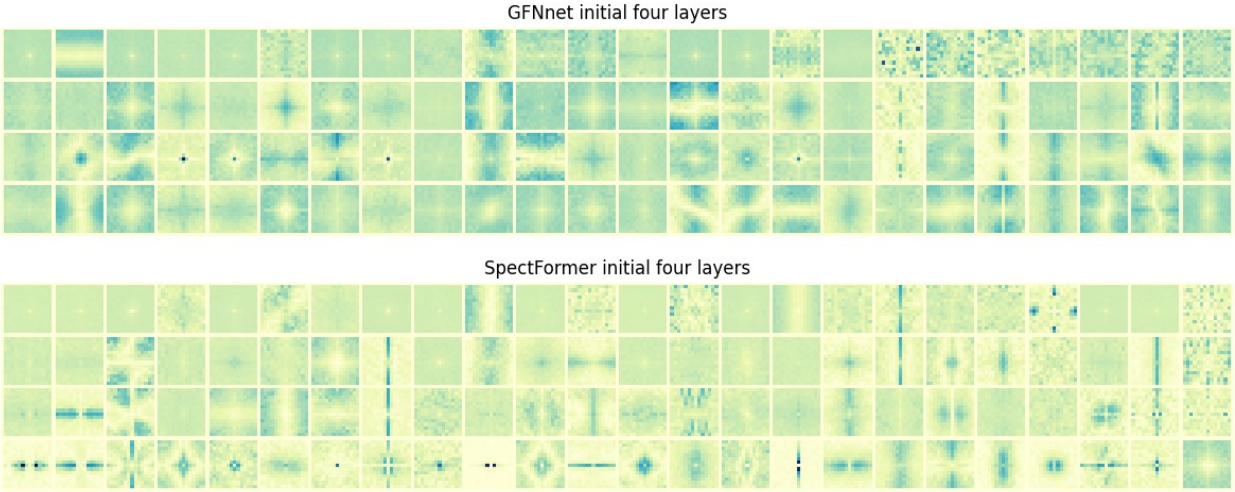
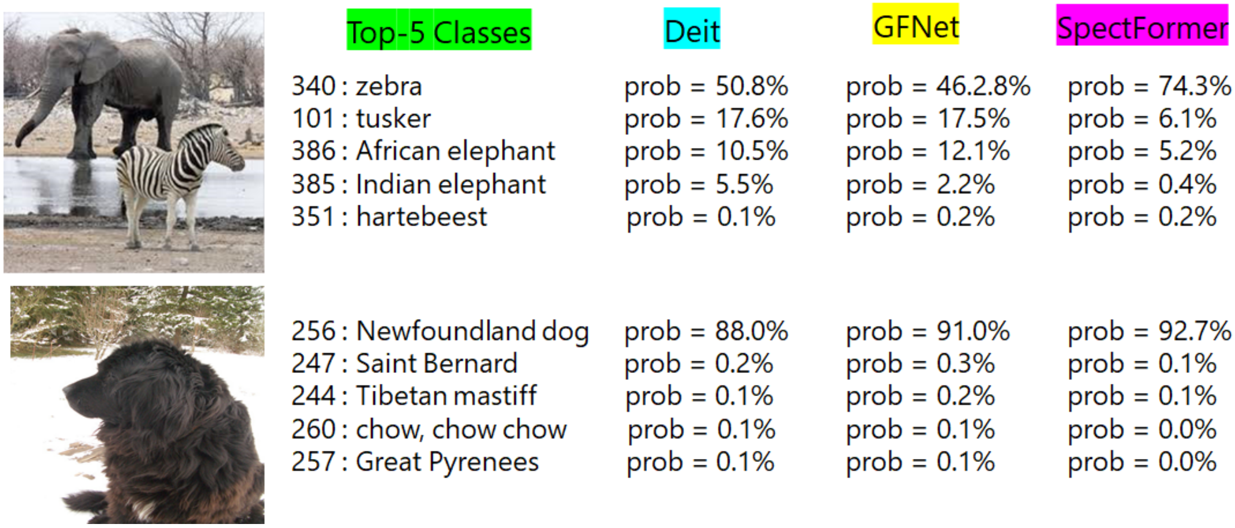


Figure 6. This figure shows the Filter characterization of the initial four layers of the GFNet [47] and SpectFormer model. It clearly shows that spectFormer captures local filter information such as lines and edges of an Image.



Top-5 Classes	Deit	GFNet	SpectFormer
340 : zebra	prob = 50.8%	prob = 46.2.8%	prob = 74.3%
101 : tusker	prob = 17.6%	prob = 17.5%	prob = 6.1%
386 : African elephant	prob = 10.5%	prob = 12.1%	prob = 5.2%
385 : Indian elephant	prob = 5.5%	prob = 2.2%	prob = 0.4%
351 : hartebeest	prob = 0.1%	prob = 0.2%	prob = 0.2%
256 : Newfoundland dog	prob = 88.0%	prob = 91.0%	prob = 92.7%
247 : Saint Bernard	prob = 0.2%	prob = 0.3%	prob = 0.1%
244 : Tibetan mastiff	prob = 0.1%	prob = 0.2%	prob = 0.1%
260 : chow, chow chow	prob = 0.1%	prob = 0.1%	prob = 0.0%
257 : Great Pyrenees	prob = 0.1%	prob = 0.1%	prob = 0.0%

Figure 7. The figure shows the top-5 class prediction probability scores for Deit [53], GFNet [47], and our SpectFormer model, indicating that SpectFormer predicts the ’Zebra’ class (Top-1 Class) with greater confidence than GFNet and Deit.

recommended by the WaveVit implementation [67]. We use the hyper-parameters recommended by the GFNet implementation [47] and train our models for 30 epochs during fine-tuning at higher resolutions. All models are trained on a single machine equipped with 8 A100 GPUs. In our experiments, we compared the fine-tuning performance of our models with GFNet [47]. Our observations indicate that our SpectFormer model outperforms GFNet’s base spectral network. Specifically, SpectFormer-S(384) achieves a performance of 83.0%, which is 1.2% higher than GFNet-S(384), as shown in Table 10. Similarly, SpectFormer-XS and SpectFormer-B perform better than GFNet-XS and GFNet-B, respectively. In the results section, we present the fine-tuned results for models trained on 224 x 224 and fine-tuned on 384 x 384, as depicted in Table-10.

4.7. Visualization of filter weights of spectral layers

Filter characterization is a technique to analyze the learned filters in transformer networks and gain insight into what kind of features the models is learning at different layers. By visualizing the learned filters, we can get a better understanding of how the transformer is processing images.

In our experiments, we performed filter characterization visualization for the initial four layers of GFNet [47] and SpectFormer as shown in figure-6. In this figure, we display the initial 24 filters of each layer and compare them between these two models. We observed that SpectFormer captures local filter information, such as lines and edges of an image, more clearly than GFNet. This suggests that SpectFormer is better able to capture local image details and, therefore, may be better suited for tasks that require high-resolution image processing.

5. Conclusion

Through this work, we analysed the core architecture of transformers by using a mixed approach that includes spectral and multi-headed attention. Previously, transformers used either all-attention layers or more recently spectral layers have been used. Spectformer combines both these aspects and shows consistently better performance than either all-attention or all-spectral layers. We use a parameterized approach that suggests further scope for adaptation of this work for specific tasks. For instance, work in remote-sensing or medical-imaging may choose a different combination of spectral and attention layers to obtain best performance. The work achieves state-of-the-art (85.7%) top-1 accuracy on ImageNet-1K dataset.

References

[1] https://openai.com/blog/chatgpt/, 2022.

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2021.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[4] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*, 2022.

[5] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

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

[8] Yoni Choukroun and Lior Wolf. Error correction code transformer. In *Advances in Neural Information Processing Systems*, 2022.

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[10] 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. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

[11] 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.

[12] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 74–92. Springer, 2022.

[13] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[15] 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.

[16] 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.

[17] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International Conference on Learning Representations*, 2022.

[18] 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.

[19] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 826–836, June 2022.

[20] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

[21] 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.

[22] 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.

[23] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2022.

[24] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[25] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574, 1959.

[26] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in Neural Information Processing Systems*, 34:18590–18602, 2021.

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[28] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[29] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

[30] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*, 2022.

[31] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020.

[32] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.

[33] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual transformer networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[34] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2020.

[35] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. In *International Conference on Learning Representations*, 2022.

[36] 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.

[37] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021.

[38] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[41] Tan Minh Nguyen, Minh Pham, Tam Minh Nguyen, Khai Nguyen, Stanley Osher, and Nhat Ho. Fourierformer: Transformer meets generalized fourier integral theorem. In *Advances in Neural Information Processing Systems*, 2022.

[42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[43] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast vision transformers with hilo attention. In *Advances in Neural Information Processing Systems*, 2022.

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

[45] Badri N Patro and Vijay Agneeswaran. Efficiency 360: Efficient vision transformers. *arXiv preprint arXiv:2302.08374*, 2023.

[46] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 35:10353–10366, 2022.

[47] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34:980–993, 2021.

[48] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng YAN. Inception transformer. In *Advances in Neural Information Processing Systems*, 2022.

[49] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021.

[50] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10935–10944, 2022.

[52] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[53] 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.

[54] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022.

[55] 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.

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

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

[58] Pichao Wang, Xue Wang, Hao Luo, Jingkai Zhou, Zhipeng Zhou, Fan Wang, Hao Li, and Rong Jin. Scaled relu matters for training vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2495–2503, 2022.

[59] 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.

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

[61] Ziyu Wang, Wenhao Jiang, Yiming M Zhu, Li Yuan, Yibing Song, and Wei Liu. Dynamixer: a vision mlp architecture with dynamic mixing. In *International Conference on Machine Learning*, pages 22691–22701. PMLR, 2022.

[62] 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.

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

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

[65] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021.

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

[67] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 328–345. Springer, 2022.

[68] 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.

[69] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[70] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[71] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022.

[72] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3008, 2021.

[73] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12083–12093, 2022.

[74] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.