Inception Transformer

Chenyang Si[1*] **Weihao Yu**[1,2*] **Pan Zhou**[1] **Yichen Zhou**[1,2] **Xinchao Wang**[2] **Shuicheng Yan**[1]

[1]Sea AI Lab [2]National University of Singapore

{sicy,yuweihao,zhoupan,zhouyc,yansc}@sea.com, xinchao@nus.edu.sg

Abstract

Recent studies show that Transformer has strong capability of building long-range dependencies, yet is incompetent in capturing high frequencies that predominantly convey local information. To tackle this issue, we present a novel and general-purpose *Inception Transformer*, or *iFormer* for short, that effectively learns comprehensive features with both high- and low-frequency information in visual data. Specifically, we design an Inception mixer to explicitly graft the advantages of convolution and max-pooling for capturing the high-frequency information to Transformers. Different from recent hybrid frameworks, the Inception mixer brings greater efficiency through a channel splitting mechanism to adopt parallel convolution/max-pooling path and self-attention path as high- and low-frequency mixers, while having the flexibility to model discriminative information scattered within a wide frequency range. Considering that bottom layers play more roles in capturing high-frequency details while top layers more in modeling low-frequency global information, we further introduce a frequency ramp structure, *i.e.*, gradually decreasing the dimensions fed to the high-frequency mixer and increasing those to the low-frequency mixer, which can effectively trade-off high- and low-frequency components across different layers. We benchmark the iFormer on a series of vision tasks, and showcase that it achieves impressive performance on image classification, COCO detection and ADE20K segmentation. For example, our iFormer-S hits the top-1 accuracy of 83.4% on ImageNet-1K, much higher than DeiT-S by 3.6%, and even slightly better than much bigger model Swin-B (83.3%) with only 1/4 parameters and 1/3 FLOPs. https://github.com/sail-sg/iFormer.

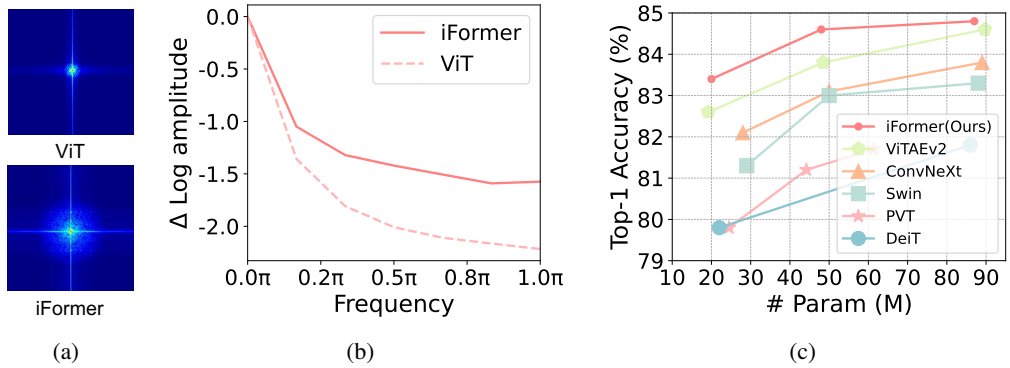
1 Introduction

Transformer [1] has taken the natural language processing (NLP) domain by storm, achieving surprisingly high performance in many NLP tasks, *e.g.*, machine translation [2] and question-answering [3]. This is largely attributed to its strong capability of modeling long-range dependencies in the data with self-attention mechanism. Its success has led researchers to investigate its adaptation to the computer vision field, and Vision Transformer (ViT) [4] is a pioneer. This architecture is directly inherited from NLP [1], but applied to image classification with raw image patches as input. Later, many ViT variants [5–13] have been developed to boost performance or scale to a wider range of vision tasks, *e.g.*, object detection [10, 11] and segmentation [12, 13].

ViT and its variants are highly capable of capturing low-frequencies in the visual data [14], mainly including global shapes and structures of a scene or object, but are not very powerful for learning high-frequencies, mainly including local edges and textures. This can be intuitively explained: self-attention, the main operation used in ViTs to exchange information among non-overlap patch tokens, is a global operation and much more capable of capturing global information (low frequencies) in the data than local information (high frequencies). As shown in Fig. 1(a) and 1(b), the Fourier spectrum

*Equal contribution. Weihao Yu did this work during an internship at Sea AI Lab.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) (b) (c)

Figure 1: **(a) Fourier spectrum of ViT [18] and iFormer. (b) Relative log amplitudes of Fourier transformed feature maps. (c) Performance of models on ImageNet-1K validation set.** (a) and (b) show that iFormer captures more high-frequency signals.

and relative log amplitudes of the Fourier show that ViT tends to well capture low-frequency signals but few high-frequency signals. This observation also accords with the empirical results in [14], which shows ViT presents the characteristics of low-pass filters. This low-frequency preferability impairs the performance of ViTs, as 1) low-frequency information filling in all the layers may deteriorate high-frequency components, *e.g.*, local textures, and weakens modeling capability of ViTs; 2) high-frequency information is also discriminative and can benefit many tasks, *e.g.*, (fine-grained) classification. Actually, human visual system extracts visual elementary features at different frequencies [15–17]: low frequency provides global information about a visual stimulus, and high frequency conveys local spatial changes in the image (*e.g.*, local edges/textures). Hence, it is necessary to develop a new ViT architecture for capturing both high and low frequencies in the visual data.

CNNs are the most fundamental backbone for general vision tasks. Unlike ViTs, they cover more local information through local convolution within the receptive fields, thus effectively extracting high-frequency representations [19, 20]. Recent studies [21–25] have integrated CNNs and ViTs considering their complementary advantages. Some methods [21, 22, 24, 25] stack convolution and attention layers in a serial manner to inject the local information into global context. Unfortunately, this serial manner only models one type of dependency, either global or local, in one layer, and discards the global information during locality modeling, or vice versa. Other works [23, 26] adopt parallel attention and convolution to learn global and local dependencies of the input at the same time. However, it is found in [27] that part of the channels are for processing local information and the other for global modeling, meaning current parallel structures have information redundancy if processing all channels in each branch.

To address this issue, we propose a simple and efficient *Inception Transformer (iFormer)*, as shown in Fig. 2, which grafts the merit of CNNs for capturing high-frequencies to ViTs. The key component in iFormer is an Inception token mixer as shown in Fig. 3. This Inception mixer aims to augment the perception capability of ViTs in the frequency spectrum by capturing both high and low frequencies in the data. To this end, the Inception mixer first splits the input feature along the channel dimension, and then feeds the split components into high-frequency mixer and low-frequency mixer respectively. Here the high-frequency mixer consists of a max-pooling operation and a parallel convolution operation, while the low-frequency mixer is implemented by a vanilla self-attention in ViTs. In this way, our iFormer can effectively capture particular frequency information on the corresponding channel, and thus learn more comprehensive features within a wide frequency range compared with vanilla ViTs, which can be clearly observed in Fig. 1(a) and 1(b).

Moreover, we find that lower layers often need more local information, while higher layers desire more global information, which also accords with the observations in [27]. This is because, like in human visual system, the details in high frequency components help lower layers to capture visual elementary features and also to gradually gather local information for having a global understanding of the input. Inspired by this, we design a frequency ramp structure. In particular, from lower to higher layers, we gradually feed more channel dimensions to low-frequency mixer and fewer channel dimensions to high-frequency mixer. This structure can trade-off high-frequency and low-frequency components across all layers. Its effectiveness has been verified by experimental results in Sec. 4.

Experimental results show that iFormer surpasses state-of-the-art ViTs and CNNs on several vision tasks, including image classification, object detection and segmentation. For example, as shown in Fig. 1(c), with different model sizes, iFormer makes consistent improvements over popular frameworks on ImageNet-1K [28], *e.g.*, DeiT [29], Swin [5] and ConvNeXt [30]. Meanwhile, iFormer outperforms recent frameworks on COCO [31] detection and ADE20K [32] segmentation.

2 Related work

Transformers [1] are firstly proposed for machine translation tasks and then become popular in other tasks like natural language understanding [33–35] and generation [36, 37] in NLP domain, as well as image classification [18, 29, 38], object detection [6, 39, 40] and semantic segmentation [41, 42] in computer vision. The attention module in Transformers has an outstanding ability to capture global dependency, but it makes the models produce similar representations across layers [27]. Moreover, self-attention mainly captures low-frequency information and tends to neglect high-frequency components related to the detailed information [14].

CNNs [43–47] are the de-facto model for vision tasks due to their outstanding ability to model local dependency [47–49] as well as extract high-frequency [19, 50]. With these advantages, CNNs are rapidly introduced into Transformers in a serial or parallel manner [23–26, 51–53]. For serial methods, convolutions are applied at different positions of the Transformer. CvT [25] and PVT-v2 [54] replace the hard patch embedding with a layer of overlapping convolution. LV-ViT [51], LeViT [55] and ViT$_C$ [21] further stack several layers of convolutions as the stem for models, which is found helpful in training and achieving better performance. Besides the stem, ViT-hybrid [18], CoAtNet [24], Hybrid-MS [56] and UniFormer [22] design early stages with convolution layers. However, the combination of convolution and attention in a serial order means each layer can only process either high or low frequency and neglects the other part. To enable each layer to process different frequencies, we adopt the parallel manner to combine convolution and attention in a token mixer.

Compared with serial methods, there are not many works combining attention and convolution in a parallel manner in literature. CoaT [26] and ViTAE [23] introduce convolution as a branch parallel to attention and utilize elementwise sum to merge the output of the two branches. However, Raghu *et al.* find that some channels tend to extract local dependency while others are for modeling global information [27], indicating redundancy for the current parallel mechanism to process all channels in different branches. In contrast, we split channels into branches of high and low frequencies. GLiT [53] also adopt parallel manner but it directly concatenate the features from convolution and attention branches as the mixer output, lacking the fusion of features in different frequencies. Instead, we design a explicit fusion module to merge the outputs from low- and high-frequency branches.

3 Method

3.1 Revisit Vision Transformer

We first revisit the Vision Transformer. For vision tasks, Transformers first split the input image into a sequence of tokens, and each patch token is projected into a hidden representation vector with a leaner layer, denoted as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ or $\boldsymbol{X} \in \mathbb{R}^{N \times C}$, where N is the number of patch tokens and C indicates the dimension of features. Then, all of the tokens are combined with a positional embedding and fed into the Transformer layers that contain multi-head self-attention (MSA) and a feed-forward network (FFN).

In MSA, the attention-based mixer exchanges information between all patch tokens so that it strongly focuses on aggregating the global dependency across all layers. However, excessive propagation of global information would strengthen the low-frequency representation. It can be seen from the visualization of Fourier spectrum in Fig. 1(a) that low-frequency information dominates the representations of ViT [18]. This actually impairs the performance of ViTs, as it may deteriorate the high-frequency components, *e.g.*, local textures, and weakens the modeling capability of ViTs [14]. In the visual data, high-frequency information is also discriminative and can benefit many tasks [19, 20]. Hence, to address the issue, we propose a simple and efficient Inception Transformer, as shown in Fig. 2, with two key novelties, *i.e.*, Inception mixer and frequency ramp structure.

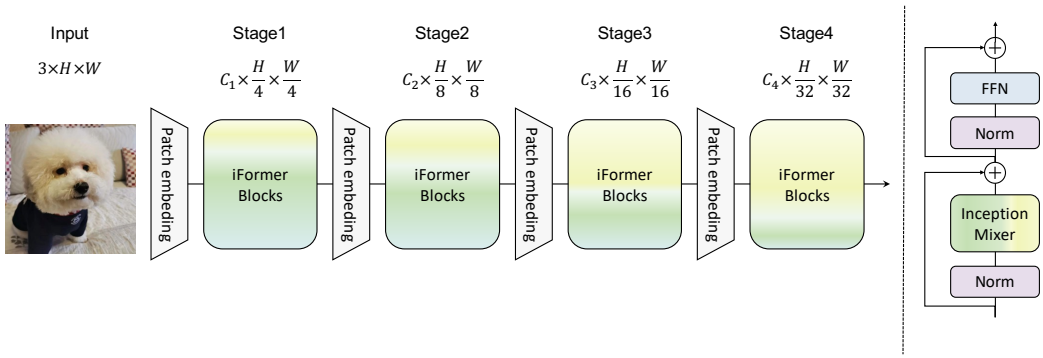


Figure 2: **The overall architecture of iFormer and details of iFormer block** . For each block, yellow and green indicate low- and high-frequency information, respectively. Best viewed in color.

3.2 Inception token mixer

We propose an Inception mixer to graft the powerful capability of CNNs for extracting high-frequency representation to Transformers. Its detailed architecture is depicted in Fig. 3. We use the name of "Inception" since the token mixer is highly inspired by the Inception module [46, 57–59] with multiple branches. Instead of directly feeding image tokens into the MSA mixer, the Inception mixer first splits the input feature along the channel dimension, and then respectively feeds the split components into high-frequency mixer and low-frequency mixer. Here the high-frequency mixer consists of a max-pooling operation and a parallel convolution operation, while the low-frequency mixer is implemented by a self-attention.

Technically, given the input feature map $\boldsymbol{X} \in \mathbb{R}^{N \times C}$, it is factorized $\boldsymbol{X}$ into $\boldsymbol{X}_h \in \mathbb{R}^{N \times C_h}$ and $\boldsymbol{X}_l \in \mathbb{R}^{N \times C_l}$ along the channel dimension, where $C_h + C_l = C$. Then, $\boldsymbol{X}_h$ and $\boldsymbol{X}_l$ are assigned to high-frequency mixer and low-frequency mixer respectively.

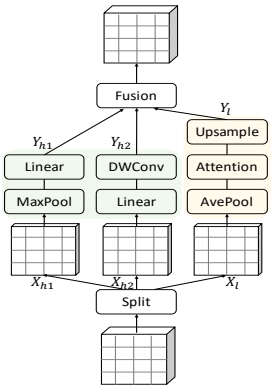


Figure 3: **The details of Inception mixer.**

High-frequency mixer. Considering the sharp sensitiveness of the maximum filter and the detail perception of convolution operation, we propose a parallel structure to learn the high-frequency components. We divide the input $\boldsymbol{X}_h$ into $\boldsymbol{X}_{h1} \in \mathbb{R}^{N \times \frac{C_h}{2}}$ and $\boldsymbol{X}_{h2} \in \mathbb{R}^{N \times \frac{C_h}{2}}$ along the channel. As shown in Fig. 3, $\boldsymbol{X}_{h1}$ is embedded with a max-pooling and a linear layer [46], and $\boldsymbol{X}_{h2}$ is fed into a linear and a depthwise convolution layer [60–62]:

$$\boldsymbol{Y}_{h1} = \mathrm{FC}\left(\mathrm{MaxPool}\left(\boldsymbol{X}_{h1}\right)\right), \tag{1}$$

$$\boldsymbol{Y}_{h2} = \mathrm{DwConv}\left(\mathrm{FC}\left(\boldsymbol{X}_{h2}\right)\right), \tag{2}$$

where $\boldsymbol{Y}_{h1}$ and $\boldsymbol{Y}_{h2}$ denote the outputs of high-frequency mixers.

Finally, the outputs of low- and high-frequency mixers are concatenated along the channel dimension:

$$\boldsymbol{Y}_{\boldsymbol{c}} = \mathrm{Concat}\left(\boldsymbol{Y}_l, \boldsymbol{Y}_{h1}, \boldsymbol{Y}_{h2}\right). \tag{3}$$

The upsample operation in Eq. (7) selects the value of the nearest point for each position to be interpolated regardless of any other points, which results in excessive smoothness between adjacent tokens. We design a fusion module to elegantly overcome this issue, *i.e.*, a depthwise convolution exchanging information between patches, while keeping a cross-channel linear layer that works per location like in previous Transformers. The final output can be expressed as

$$\boldsymbol{Y} = \mathrm{FC}\left(\boldsymbol{Y}_{\boldsymbol{c}} + \mathrm{DwConv}\left(\boldsymbol{Y}_{\boldsymbol{c}}\right)\right). \tag{4}$$

Like the vanilla Transformer, our iFormer is equipped with a feed-forward network (FFN), and differently it also incorporates the above Inception token mixer (ITM); LayerNorm (LN) is applied before ITM and FFN. Hence the Inception Transformer block is formally defined as

$$\boldsymbol{Y} = \boldsymbol{X} + \mathrm{ITM}\left(\mathrm{LN}\left(\boldsymbol{X}\right)\right), \tag{5}$$

$$\boldsymbol{H} = \boldsymbol{Y} + \mathrm{FFN}\left(\mathrm{LN}\left(\boldsymbol{Y}\right)\right). \tag{6}$$

Low-frequency mixer. We use the vanilla multi-head self-attention to communicate information among all tokens for the low-frequency mixer. Despite the strong capability of the attention for learning global representation, the large resolution of feature maps would bring large computation cost in lower layers. We therefore simply utilize an average pooling layer to reduce the spatial scale of $\boldsymbol{X}_l$ before the attention operation and an upsample layer to recover the original spatial dimension after the attention. This design largely reduces the computational overhead and makes the attention operation focus on embedding global information. This branch can be defined as

$$\boldsymbol{Y}_l = \text{Upsample}\left(\text{MSA}\left(\text{AvePooling}\left(\boldsymbol{X}_l\right)\right)\right), \tag{7}$$

where $\boldsymbol{Y}_l$ is the output of low-frequency mixer. Note that the kernel size and stride for the pooling and upsample layers are set to 2 only at the first two stages.

3.3 Frequency ramp structure

In the general visual frameworks, bottom layers play more roles in capturing high-frequency details while top layers more in modeling low-frequency global information, *i.e.*, the hierarchical representations of ResNet [47]. Like humans, by capturing the details in high frequency components, lower layers can capture visual elementary features, and also gradually gather local information to achieve a global understanding of the input. We are inspired to design a frequency ramp structure which gradually splits more channel dimensions from lower to higher layers to low-frequency mixer and thus leave fewer channel dimensions to high-frequency mixer. Specifically, as shown in Fig. 2, our backbone has four stages with different channel and spatial dimensions. For each blocks, we define a channel ratio to better balance the high-frequency and low frequency components, *i.e.*, $\frac{C_h}{C}$ and $\frac{C_l}{C}$, where $\frac{C_h}{C} + \frac{C_l}{C} = 1$. In the proposed frequency ramp structure, $\frac{C_h}{C}$ gradually decreases from shallow to deep layers, while $\frac{C_l}{C}$ gradually increases. Hence, with the flexible frequency ramp structure, iFormer can effectively trade-off high- and low-frequency components across all layers. The configuration of different iFormer models will be described in the appendix.

4 Experiments

We evaluate our iFormer on several vision benchmark tasks, *i.e.*, image classification, object detection and semantic segmentation, by comparing it with representative ViTs, CNNs and their hybrid variants. Ablation analysis is also conducted to show the contribution of each novelty in our method. More results will be reported in the appendix.

4.1 Results on image classification

Setup. For image classification, we evaluate iFormer on the ImageNet dataset [28]. We train the iFormer model with the standard procedure in [6, 22, 29]. Specifically, we use AdamW optimizer with an initial learning rate 1×10^{-3} via cosine decay [70], a momentum of 0.9, and a weight decay of 0.05. We set the training epoch number as 300 and the input size as 224 $\times$ 224. We adopt the same data augmentations and regularization methods in DeiT [29] for fair comparison.

We also use LayerScale [71] to train deep models. Like previous studies [5, 67], we further fine tune iFormer on the input size of 384×384, with the weight decay of 1×10^{-8}, learning rate of 1×10^{-5}, batch size of 512. For fairness, we adopt Timm [72] to implement and train iFormer.

Results. Table 1 summarizes the image classification accuracy of all compared methods on ImageNet. For the small model size ($\sim$20M), our iFormer surpasses both the SoTA ViTs and hybrid ViTs, although some ViTs, *e.g.*, Swin [5], Focal [64] and CSwin [65], actually already introduce convolution-like inductive bias into their architectures, and hybrid ViTs directly integrate convolution into ViTs. Specifically, our iFormer-S respectively gains 0.7% and 0.5% top-1 accuracy advantage over SoTA ViTs (*i.e.*, CSwin-T) and hybrid ViTs (*i.e.*, UniFormer-S), while enjoying the same or smaller model size.

For the medium model size ($\sim$50M), iFormer-B achieves 84.6% top-1 accuracy, and improves over the SoTA ViTs and hybrid ViTs with similar model sizes by significant margins 1.0% and 0.7% respectively. For CNNs, similar to comparison results on medium model size, our iFormer-B outperforms ConvNeXt-S by 1.5%. As for the large mode ($\sim$100M), one can observe similar results on small and medium model sizes.

Table 1: Comparison of different types of models on ImageNet-1K [28].

Model Size	Arch.	Method	#Param. (M)	FLOPs (G)	Input Size Train	Input Size Test	ImageNet Top-1	ImageNet Top-5
small model size (∼20M)	CNN	RSB-ResNet-50 [47, 63]	26	4.1	224	224	80.4	-
		ConvNeXt-T [30]	28	4.5	224	224	82.1	-
	ViT	Deit-S [29]	22	4.6	224	224	79.8	95.0
		PVT-S [6]	25	3.8	224	224	79.8	-
		T2T-14 [38]	22	5.2	224	224	80.7	-
		Swin-T [5]	29	4.5	224	224	81.3	95.5
		Focal-T [64]	29	4.9	224	224	82.2	95.9
		CSwin-T [65]	23	4.3	224	224	82.7	-
	Hybrid	CvT-13 [25]	20	4.5	224	224	81.6	-
		CoAtNet-0 [24]	25	4.2	224	224	81.6	-
		Container [66]	22	8.1	224	224	82.7	-
		ViTAE-S [23]	24	5.6	224	224	82.0	95.9
		ViTAEv2-S [67]	19	5.7	224	224	82.6	96.2
		UniFormer-S [22]	22	3.6	224	224	82.9	-
		iFormer-S	**20**	**4.8**	**224**	**224**	**83.4**	**96.6**
medium model size (∼50M)	CNN	RSB-ResNet-101 [47, 63]	45	7.9	224	224	81.5	-
		RSB-ResNet-152 [47, 63]	60	11.6	224	224	82.0	-
		ConvNeXt-S [30]	50	8.7	224	224	83.1	-
	ViT	PVT-L [6]	61	9.8	224	224	81.7	-
		T2T-24 [38]	64	13.2	224	224	82.2	-
		Swin-S [5]	50	8.7	224	224	83.0	96.2
		Focal-S [64]	51	9.1	224	224	83.5	96.2
		CSwin-S [65]	35	6.9	224	224	83.6	-
	Hybrid	CvT-21 [25]	32	7.1	224	224	82.5	-
		CoAtNet-1 [24]	42	8.4	224	224	83.3	-
		ViTAEv2-48M [67]	49	13.3	224	224	83.8	96.6
		UniFormer-B [22]	50	8.3	224	224	83.9	-
		iFormer-B	**48**	**9.4**	**224**	**224**	**84.6**	**97.0**
large model size (∼100M)	CNN	RegNetY-16GF [29, 68]	84	16.0	224	224	82.9	-
		ConvNeXt-B [30]	89	15.4	224	224	83.8	-
	ViT	DeiT-B [29]	86	17.5	224	224	81.8	95.6
		Swin-B [5]	88	15.4	224	224	83.3	96.5
		Focal-B [64]	90	16.0	224	224	83.8	96.5
		CSwin-B [65]	78	15.0	224	224	84.2	-
	Hybrid	BoTNet-T7 [69]	79	19.3	256	256	84.2	-
		CoAtNet-3 [24]	168	34.7	224	224	84.5	-
		ViTAEv2-B [67]	90	24.3	224	224	84.6	96.9
		iFormer-L	**87**	**14.0**	**224**	**224**	**84.8**	**97.0**

Table 2 reports the fine-tuning accuracy on the larger resolution, *i.e.*, 384×384. One can observe that iFormer consistently outperforms the counterparts by a significant margin across different computation settings. These results clearly demonstrate the advantages of iFormer on image classifications.

4.2 Results on object detection and instance segmentation

Setup. We evaluate iFormer on the COCO object detection and instance segmentation tasks [31], where the models are trained on 118K images and evaluated on validation set with 5K images. Here, we use iFormer as the backbone in Mask R-CNN [75]. In the training phase, we use iFormer pretrained on ImageNet to initialize the detector, and adopt AdamW to train with an initial learning rate of 1×10^{-4}, a batch size of 16, and $1\times$ training schedule with 12 epochs. For training, the input images are resized to be 800 pixels on the shorter side an no more than 1,333 pixels on the longer

Table 2: Fine-tuning Results with larger resolution (384 × 384) on ImageNet-1K [28]. The models in gray color are trained with larger input size.

Method	#Param. (M)	FLOPs (G)	Input Size Train	Input Size Test	ImageNet Top-1
EfficientNet-B5 [73]	30	9.9	456	456	83.6
EfficientNetV2-S [74]	22	8.5	384	384	83.9
CSwin-T↑384 [65]	23	14.0	224	384	84.3
CvT-13↑384 [25]	20	16.3	224	384	83.0
CoAtNet-0↑384 [24]	20	13.4	224	384	83.9
ViTAEv2-S↑384 [67]	19	17.8	224	384	83.8
iFormer-S↑384	**20**	**16.1**	**224**	**384**	**84.6**
EfficientNet-B7 [73]	66	39.2	600	600	84.3
EfficientNetV2-M [74]	54	25.0	480	480	85.1
ViTAEv2-48M ↑384 [67]	49	41.1	224	384	84.7
CSwin-S↑384 [65]	35	22.0	224	384	85.0
CoAtNet-1↑384 [24]	42	27.4	224	384	85.1
iFormer-B↑384	**48**	**30.5**	**224**	**384**	**85.7**
EfficientNetV2-L [74]	121	53	480	480	85.7
Swin-B↑384 [5]	88	47.0	224	384	84.2
CSwin-B↑384 [65]	78	47.0	224	384	85.4
ViTAEv2-B↑384 [67]	90	74.4	224	384	85.3
CoAtNet-2↑384 [24]	75	49.8	224	384	85.7
iFormer-L↑384	**87**	**45.3**	**224**	**384**	**85.8**

Table 3: Performance of object detection and instance segmentation on COCO val2017 [31]. AP^b and AP^m represent bounding box AP and mask AP, respectively. All models are based on Mask R-CNN [75] and trained by 1× training schedule. The FLOPs are measured at resolution 800×1280.

Method	#Param. (M)	FLOPs (G)	Mask R-CNN 1 × AP^b	AP^b_{50}	AP^b_{70}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 [47]	44	260	38.0	58.6	41.4	34.4	55.1	36.7
PVT-S [6]	44	245	40.4	62.9	43.8	37.8	60.1	40.3
TwinsP-S [76]	44	245	42.9	65.8	47.1	40.0	62.7	42.9
Twins-S [76]	44	228	43.4	66.0	47.3	40.3	63.2	43.4
Swin-T [5]	48	264	42.2	64.6	46.2	39.1	61.6	42.0
ViL-S [77]	45	218	44.9	67.1	49.3	41.0	64.2	44.1
Focal-T [64]	49	291	44.8	67.7	49.2	41.0	64.7	44.2
UniFormer-S_{h14} [22]	41	269	45.6	68.1	49.7	41.6	64.8	45.0
iFormer-S	**40**	**263**	**46.2**	**68.5**	**50.6**	**41.9**	**65.3**	**45.0**
ResNet101 [47]	63	336	40.4	61.1	44.2	36.4	57.7	38.8
X101-32	63	340	41.9	62.5	45.9	37.5	59.4	40.2
PVT-M [6]	64	302	42.0	64.4	45.6	39.0	61.6	42.1
TwinsP-B [76]	64	302	44.6	66.7	48.9	40.9	63.8	44.2
Twins-B [76]	76	340	45.2	67.6	49.3	41.5	64.5	44.8
Swin-S [5]	69	354	44.8	66.6	48.9	40.9	63.4	44.2
Focal-S [64]	71	401	47.4	69.8	51.9	42.8	66.6	46.1
CSWin-S [65]	54	342	47.9	70.1	52.6	43.2	67.1	46.2
UniFormer-B [22]	69	399	47.4	69.7	52.1	43.1	66.0	46.5
iFormer-B	**67**	**351**	**48.3**	**70.3**	**53.2**	**43.4**	**67.2**	**46.7**

side. For the test image, its shorter side is fixed to 800 pixels. All experiments are implemented on mmdetection [78] codebase.

Results. Table 3 reports the box mAP (AP^b) and mask mAP (AP^m) of the compared models. Under similar computation configurations, iFormers outperforms all previous backbones. Specifically, compared with popular ResNet [47] backbones, our iFormer-S brings 8.2 points of AP^b and 7.5 points AP^m improvements over ResNet50. Compared with various Transformer backbones, our

iFormers still maintain the performance superiority over their results. For example, our iFormer-B surpasses UniFormer-B [22], Swin-S [5] by 0.9 points of AP^b and 3.5 points of AP^b respectively.

4.3 Results on semantic segmentation

Setup. We further evaluate the generality of iFormer through a challenging scene parsing benchmark on semantic segmentation, *i.e.*, ADE20K [32]. The dataset contains 20K training images and 2K validation images. We adopt iFormer pretrained on ImageNet as the backbone of the Semantic FPN [79] framework. Following PVT [6] and UniFormer [22], we use AdamW with an initial learning rate of 2×10^{-4} with cosine learning rate schedule to train 80k iterations. All experiments are implemented on mmsegmentation [80] codebase.

Table 4: Semantic segmentation with semantic FPN [79] on ADE20K [32]. The FLOPs are measured at resolution 512×2048.

Method	#Param. (M)	FLOPs (G)	mIoU (%)
ResNet50 [47]	29	183	36.7
PVT-S [6]	28	161	39.8
TwinsP-S [76]	28	162	44.3
Twins-S [76]	28	144	43.2
Swin-T [5]	32	182	41.5
UniFormer-S_{h32} [22]	25	199	46.2
UniFormer-S [22]	25	247	46.6
UniFormer-B [22]	54	471	48.0
iFormer-S	**24**	**181**	**48.6**

Results. In Table 4, we report the mIoU results of different backbones. On the Semantic FPN [79] framework, our iFormer consistently outperforms previous backbones on this task, including CNNs and (hybrid) ViTs. For instance, iFormer-S achieves 48.6 mIoU, surpassing UniFormer-S [22] by 2.0 mIoU, while using less computation complexity. Moreover, compared with UniFormer-B [22], our iFormer-S still achieves 0.6 mIoU improvement with only 1/2 parameters and nearly 1/3 FLOPs.

4.4 Ablation study and visualization

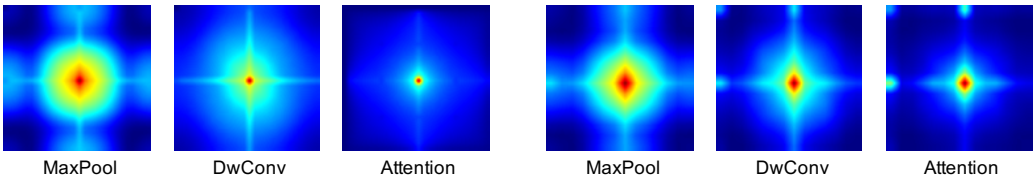
In this section, we conduct experiments to better understand iFormer. All the models are trained for 100 epochs on ImageNet, with the same training setting as described in Sec. 4.1.

Inception token mixer. The Inception mixer is proposed to augment the perception capability of ViTs in the frequency spectrum. To evaluate the effects of the components in the Inception mixer, we remove the max-pooling or convolution from the full model and then report the results in Table 5, where ✓ and ✗ denote whether or not the corresponding branch is enabled. Observably, combining attention with convolution and max-pooling can the highest classification accuracy. To further explore this scheme, Fig. 4 visualizes the Fourier spectrum of the Attention, MaxPool and DwConv branches in Inception mixer. We can see the attention mixer has higher concentrations on low frequencies; with the high-frequency mixer, *i.e.*, convolution and max-pooling, the model is encouraged to learn high frequency information. Overall, these results prove the effectiveness of the Inception mixer for expanding the perception capability of the Transformer in the frequency spectrum.

Frequency ramp structure. Previous investigations [27] show requirement of more local information at lower layers of the Transformer and more global information at higher layers. We accordingly assume that a frequency ramp structure, *i.e.*, decreasing dimensions at high-frequency components

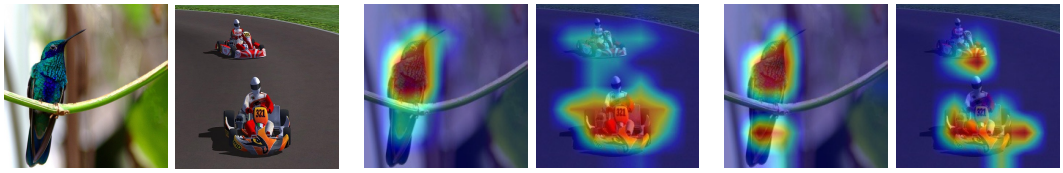
Table 5: Ablation study of Inception mixer and frequency ramp structure on ImageNet-1K. All the models are trained for 100 epochs.

	Attention	MaxPool	DwConv	#Param. (M)	FLOPs (G)	Top-1(%)
Mixer	✓	✓	✗	20	4.9	81.2
	✓	✗	✓	20	4.9	81.4
	✓	✓	✓	20	4.8	81.5
Structure	$C_l/C \downarrow, C_h/C \uparrow$			19	4.7	80.5
	$C_l/C = C_h/C$			19	4.7	80.7
	$C_l/C \uparrow, C_h/C \downarrow$			20	4.8	81.2



(a) 4-th layer (b) 8-th layer

Figure 4: **(a) (b) Fourier spectrum of iFormer-S for the MaxPool, DwConv and Attention branches in the Inception mixer.** We can observe that attention mixer tends to reduce high-frequencies, while MaxPool and DwConv enhance them.



(a) Input (b) Swin-T (c) iFormer-S

Figure 5: **Grad-CAM [81] activation maps of Swin-T [5] and iFormer-S trained on ImageNet.**

and increasing dimensions at low-frequency components from lower to higher layers, has a better trade-off between high-frequency and low-frequency components across all layers. In order to justify this hypothesis, we investigate the effects of the channel ratio ($\frac{C_h}{C}$ and $\frac{C_l}{C}$) in Table 5. It can be clearly seen that the model with $C_l/C \uparrow, C_h/C \downarrow$ outperforms the other two models, which is consistent with the previous investigations. Hence, this indicates the rationality of the frequency ramp structure and its potential for leaning discriminating vision representations.

Visualization. We visualize the Grad-CAM [81] activation maps of iFormer-S as well as Swin-T [5] models trained on ImageNet-1K in Fig. 5. It can be seen that compared with Swin, iFormer can more accurately and completely locate the objects. For example, in the hummingbird image, iFormer skips the branch and accurately attends to the whole bird including the tail.

5 Conclusion

In this paper, we present an Inception Transformer (iFormer), a novel and general Transformer backbone. iFormer adopts a channel splitting mechanism to simply and efficiently couple convolution/max-pooling and self-attention, giving more concentrations on high frequencies and expanding the perception capability of the Transformer in the frequency spectrum. Based on the flexible Inception token mixer, we further design a frequency ramp structure, enabling effective trade-off between high-frequency and low-frequency components across all layers. Extensive experiments show that iFormer outperforms representative vision Transformers on image classification, object detection and semantic segmentation, demonstrating the great potential of our iFormer to serve as a general-purpose backbone for computer vision. We hope this study will provide valuable insights for the community to design efficient and effective Transformer architectures.

Limitation. One obvious limitation of the proposed iFormer is that it requires manually defined channel ratio in the frequency ramp structure *i.e.*, $\frac{C_h}{C}$ and $\frac{C_l}{C}$ for each iFormer block, which needs rich experience to define better on different tasks. it is not trained on large scale datasets, *e.g.*, ImageNet-21K [48], due to computational constraint, which will be explored in further. Also, iFormer requires manually defined channel ratio in the frequency ramp structure *i.e.*, $\frac{C_h}{C}$ and $\frac{C_l}{C}$ for each iFormer block, which needs rich experience to define better on different tasks. A straightforward solution would be to use neural architecture search.

Acknowledgement

Weihao Yu would like to thank TRC program and GCP research credits for the support of partial computational resources. This project is in part supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG2-RP-2021-023).

References

[1] 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.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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*, 2022.

[8] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International Conference on Machine Learning*, pages 4487–4499. PMLR, 2021.

[9] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[10] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020.

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

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

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

[14] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2021.

[15] Jean Bullier. Integrated model of visual processing. *Brain research reviews*, 36(2-3):96–107, 2001.

[16] Moshe Bar. A cortical mechanism for triggering top-down facilitation in visual object recognition. *Journal of cognitive neuroscience*, 15(4):600–609, 2003.

[17] Louise Kauffmann, Stephen Ramanoël, and Carole Peyrin. The neural bases of spatial frequency processing during scene perception. *Frontiers in integrative neuroscience*, 8:37, 2014.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[19] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020.

[20] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021.

[22] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022.

[23] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[25] 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.

[26] 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.

[27] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

[28] 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.

[29] 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.

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[31] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[32] 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.

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

[34] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[36] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[37] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[38] 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.

[39] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

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

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

[42] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[43] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[47] 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.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[50] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3435–3444, 2019.

[51] 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, 2021.

[52] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *arXiv preprint arXiv:2204.01697*, 2022.

[53] Boyu Chen, Peixia Li, Chuming Li, Baopu Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Glit: Neural architecture search for global and local image transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2021.

[54] 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*, pages 1–10, 2022.

[55] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021.

[56] Yucheng Zhao, Guangting Wang, Chuanxin Tang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. A battle of network structures: An empirical study of cnn, transformer, and mlp. *arXiv preprint arXiv:2108.13002*, 2021.

[57] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

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

[59] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[60] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[61] Franck Mamalet and Christophe Garcia. Simplifying convnets for fast learning. In *International Conference on Artificial Neural Networks*, pages 58–65. Springer, 2012.

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

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

[64] 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.

[65] 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. *arXiv preprint arXiv:2107.00652*, 2021.

[66] Jiasen Lu, Roozbeh Mottaghi, Aniruddha Kembhavi, et al. Container: Context aggregation networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[67] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *arXiv preprint arXiv:2202.10108*, 2022.

[68] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[69] 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.

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

[71] 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.

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

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

[74] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

[75] 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.

[76] 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, 2021.

[77] 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.

[78] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[79] 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.

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

[81] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[82] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[83] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code will be released in future.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]
(b) Did you mention the license of the assets? [Yes]
(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Potential Impacts. The study introduces a general vision Transformer, *i.e.*, iFormer, which can be used on different vision tasks, *e.g.*, image classification, object detection and semantic segmentation. iFormer has no direct negative societal impact. However, we will realize that iFormer as a general-purpose backbone can be used for harmful applications such as illegal face recognition.

A.1 Results on semantic segmentation

Setup. We further evaluate the generality of iFormer on semantic segmentation with the Uper-net [82] framework. Following the training settings in Swin [5], the model is trained for 160K iterations with a batch size of 16. For training, we use AdamW optimizer with an initial learning rate 6×10^{-5}. All experiments are implemented on mmsegmentation [80] codebase.

Results. Table 6 shows the mIoU and MS mIoU results of different backbones based on the UperNet [82] framework. From these results, it can be seen that our iFormer achieves 48.4 mIoU and 48.8 MS mIoU, consistently surpassing previous backbones on this task. For instance, our iFormer-S outperforms the Swin-T [5] by 3.9 mIoU while using fewer parameters. Compered with UniFormer-S [22], iFormer-S still achieves 0.8 mIoU improvement.

Table 6: Semantic segmentation with semantic UperNet [82] on ADE20K [32]. The FLOPs are measured at resolution 512×2048.

Method	#Param. (M)	FLOPs (G)	mIoU (%)	MS mIoU (%)
TwinsP-S [76]	55	919	46.2	47.5
Twins-S [76]	54	901	46.2	47.1
Swin-T [5]	60	945	44.5	45.8
Focal-T [64]	62	998	45.8	47.0
Shuffle-T [83]	60	949	46.6	47.8
UniFormer-S$_{h32}$ [22]	52	955	47.0	48.5
UniFormer-S [22]	52	1008	47.6	48.5
iFormer-S	**49**	**938**	**48.4**	**48.8**

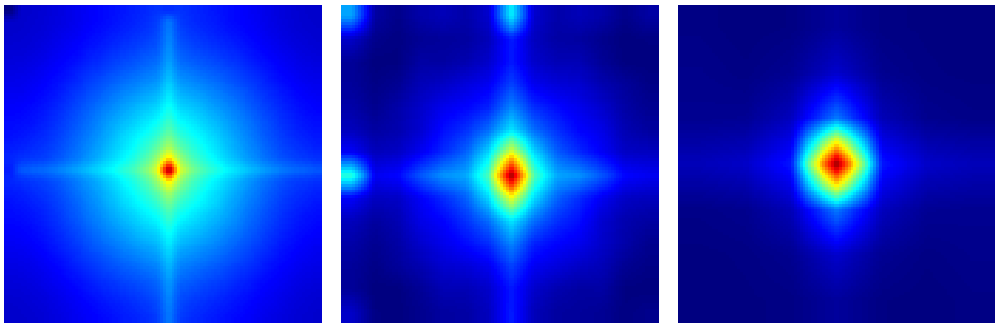
A.2 Ablation study

A.2.1 Down-sample and up-sample for self-attention

In this work, we propose the down- and up-sample structure to reduce the computational cost. Table 7 shows that, when removing this structure, iFormer-S has 7.0G FLOPs, and achieves 83.6% top-1 accuracy. However, by adopting the down- and up-sample structure, iFormer-S gets a similar accuracy (83.4%) but has much less FLOPs (4.8 G).

Table 7: Ablation study of down-up structure for Self-attention and kernel size of convolution.

	Down- and Up	#Param. (M)	FLOPs (G)	Top-1(%)
Former-S	✗	20	7.0	83.6
	✓	20	4.8	83.4
	Kernel Size			Top-1(%)
Former-S	5×5			83.2
	3×3			83.4



(a) 6-th layer (b) 12-th layer (c) 18-th layer

Figure 6: **(a) Fourier spectrum of 6-th, 12-th and 18-th layers for the iFormer-S.**

A.2.2 The kernel size of convolution

We use large kernel (5×5) to replace 3 × 3 kernel of DWConv. The results are summarized in Tables 7. We can see that the model with 3 × 3 kernel achieves better results. We believe that this is because the self-attention has learned the information that is learnt by large-kernel DWConv. Besides, the smaller kernel is more conducive and effective for capturing high-frequency information.

A.3 Visualization

A.3.1 Fourier spectrum of different layers

Frequency ramp structure plays an important role in iFormer, which is designed to learn hierarchical representations, *i.e.*, more high-frequency signals at lower layers of the Transformer and more low-frequency signals at higher layers. In order to justify this hypothesis, we visualize the Fourier spectrum of feature maps for different iFormer layers in Fig 6. We can see that iFormer captures more high-frequency components at 6-th layer and more low-frequency information at 18-th layer. Moreover, the high-frequency information gradually decreases from 6-th layer to 18-th layer, and low-frequency information does the opposite. Hence, these results indicate our iFormer can effectively trade-off high- and low-frequency components across all layers.

A.3.2 CAM

We futher show more examples of the Grad-CAM [81] activation maps of iFormer-S as well as Swin-T [5] models trained on ImageNet-1K in Fig. 7. These examples indicate that compared with Swin, iFormer can more accurately and completely attend to the key objects. Taking the hog picture as example, iFormer locates the hog accurately but Swin also locates irrelevant part.

A.4 Configurations of iFormers

In this work, three variants of iFormer are used for a fair comparison under computation configurations, *i.e.*, iFormer-S, iFormer-B and iFormer-L. Table 8 shows their detailed configurations. Following Swin [5], iFormer adopts 4-stage architecture with $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, $\frac{H}{32} \times \frac{W}{32}$ input sizes, where H and W are the width and height of the input image. In each iFormer block, C_h/C and C_l/C are used to balance the high-frequency and low frequency components. As shown in Table 8, C_h/C gradually decreases from shallow to deep layers, while C_l/C gradually increases. iFormer block uses depthwise convolution and max-pooling as high-frequency mixers. We set the kernel sizes of depthwise convolution and max-pooling to 3×3.

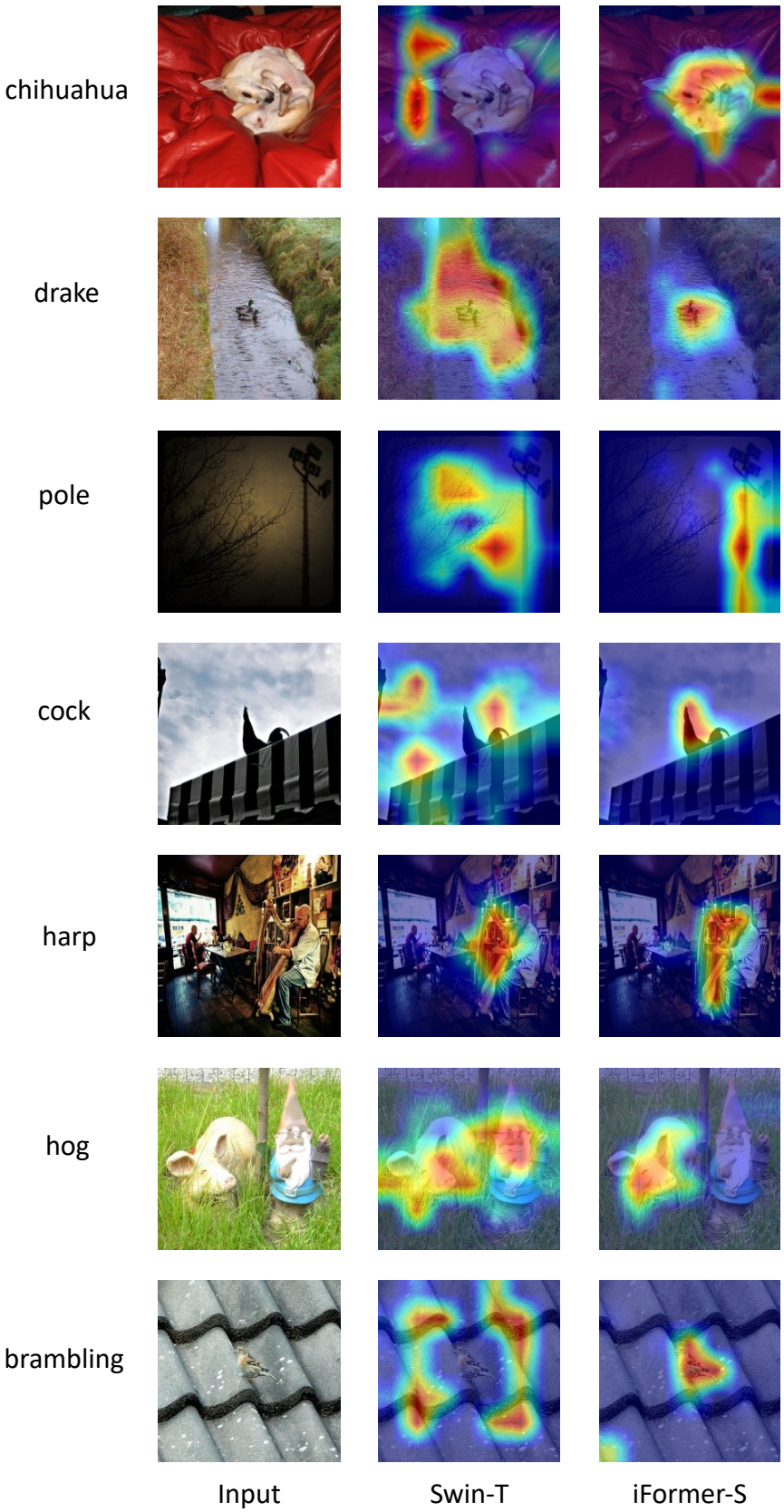


Figure 7: **Grad-CAM [81] activation maps of Swin-T [5] and iFormer-S trained on ImageNet.**

Table 8: Configurations of the variants of iFormer. Pool stride denotes the stride of the pooling and upsample layers in attention branch. The FLOPs are measured at resolution 224×224.

Stage	Layer	iFormer-S	iFormer-B	iFormer-L
1	Patch Embedding	3×3, stride 2, 48 3×3, stride 2, 96	3×3, stride 2, 48 3×3, stride 2, 96	3×3, stride 2, 48 3×3, stride 2, 96
	iFormer Block	$\begin{bmatrix} C_h/h = 2/3 \\ C_l/h = 1/3 \\ \text{pool stride } 2 \end{bmatrix} \times 3$	$\begin{bmatrix} C_h/h = 2/3 \\ C_l/h = 1/3 \\ \text{pool stride } 2 \end{bmatrix} \times 4$	$\begin{bmatrix} C_h/h = 2/3 \\ C_l/h = 1/3 \\ \text{pool stride } 2 \end{bmatrix} \times 4$
2	Patch Embedding	2×2, stride 2, 192	2×2, stride 2, 192	2×2, stride 2, 192
	iFormer Block	$\begin{bmatrix} C_h/h = 1/2 \\ C_l/h = 1/2 \\ \text{pool stride } 2 \end{bmatrix} \times 3$	$\begin{bmatrix} C_h/h = 1/2 \\ C_l/h = 1/2 \\ \text{pool stride } 2 \end{bmatrix} \times 6$	$\begin{bmatrix} C_h/h = 1/2 \\ C_l/h = 1/2 \\ \text{pool stride } 2 \end{bmatrix} \times 6$
3	Patch Embedding	2×2, stride 2, 320	2×2, stride 2, 384	2×2, stride 2, 448
	iFormer Block	$\begin{bmatrix} C_h/h = 3/10 \to 1/10 \\ C_l/h = 7/10 \to 9/10 \\ \text{pool stride } 1 \end{bmatrix} \times 9$	$\begin{bmatrix} C_h/h = 4/12 \to 2/12 \\ C_l/h = 8/12 \to 10/12 \\ \text{pool stride } 1 \end{bmatrix} \times 14$	$\begin{bmatrix} C_h/h = 4/14 \to 2/14 \\ C_l/h = 10/14 \to 12/14 \\ \text{pool stride } 1 \end{bmatrix} \times 18$
4	Patch Embedding	2×2, stride 2, 384	2×2, stride 2, 512	2×2, stride 2, 640
	iFormer Block	$\begin{bmatrix} C_h/h = 1/12 \\ C_l/h = 11/12 \\ \text{pool stride } 1 \end{bmatrix} \times 3$	$\begin{bmatrix} C_h/h = 1/16 \\ C_l/h = 15/16 \\ \text{pool stride } 1 \end{bmatrix} \times 6$	$\begin{bmatrix} C_h/h = 1/20 \\ C_l/h = 19/20 \\ \text{pool stride } 1 \end{bmatrix} \times 8$
#Param. (M)		20	48	87
FLOPs (G)		4.8	9.4	14.0