Discrete Cosin TransFormer: Image Modeling From Frequency Domain

Xinyu Li[1], Yanyi Zhang[2], Jianbo Yuan[1], Hanlin Lu[1], Yibo Zhu[1]
[1]ByteDance, [2]Rutgers University
[1]{lixinyu.arthur,jianbo.yuan,hanlin.lu,zhuyibo}@bytedance.com, [2]yz533@scarletmail.rutgers.edu

Abstract

In this paper, we propose Discrete Cosin TransFormer (DCFormer) that directly learn semantics from DCT-based frequency domain representation. We first show that transformer-based networks are able to learn semantics directly from frequency domain representation based on discrete cosine transform (DCT) without compromising the performance. To achieve the desired efficiency-effectiveness trade-off, we then leverage an input information compression on its frequency domain representation, which highlights the visually significant signals inspired by JPEG compression. We explore different frequency domain downsampling strategies and show that it is possible to preserve the semantic meaningful information by strategically dropping the high-frequency components. The proposed DCFormer is tested on various downstream tasks including image classification, object detection and instance segmentation, and achieves state-of-the-art comparable performance with less FLOPs, and outperforms the commonly used backbone (e.g. SWIN) at similar FLOPs. Our ablation results also show that the proposed method generalizes well on different transformer backbones.

1. Introduction

Different types of image representations are often used for different types of downstream tasks. The RGB-based representation carries rich semantic information, and thus becomes the mainstream solution for visual content understanding and associated computer vision tasks, e.g. image classification [11], object detection [31], etc. The frequency domain representation better separates the information from different frequency bands, which is commonly used for image compression and image quality assessment [15, 16]. In this paper, we explore image modeling directly on frequency representation unlike the conventional RGB-based image modeling, and furthermore, *efficient image modeling* by dropping the non-visually significant information directly from the frequency representations. Performing efficient image modeling directly on frequency domain is often overlooked, because in the downstream tasks that focus on semantics and content understanding, RGB-based modeling approaches generally yield better performance. There are two major challenges for efficient image modeling directly on frequency domain: (1). how to model the frequency representations as the adjacent pixels lack direct spatial associations; and (2). how to compress the non-visually significant information without compromising the performance.

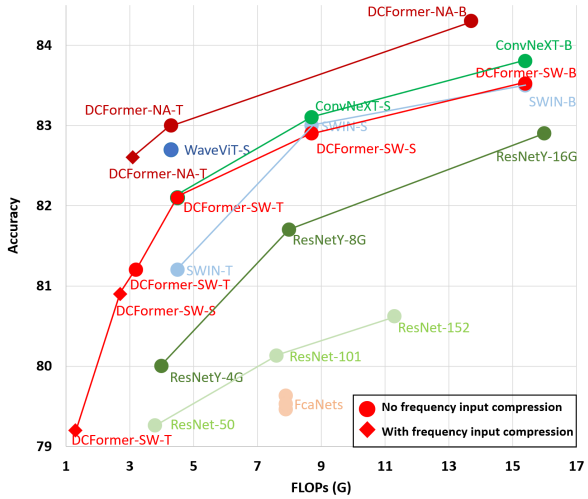

Figure 1: Image classification on ImageNet-1K. DCFormer (red lines) is able to achieve better efficiency/effectiveness balance: DCFormer achieves similar performance at lower FLOPs, and better accuracy at similar FLOPs. DCFormer-SW and -NA denotes DCFormer with SWIN [33] and NAT [19] as backbones, respectively. Details in section 4.

For frequency representations modeling, we find that the inverse-DCT transformation shares a similar mathematical representation to a transformer (self-attention based networks), indicating that it is possible for transformer-based encoders to simulate the inverse-DCT process (details in Section 3). Therefore, we propose *Discrete Cosin TransFormer* (DCFormer) using frequency domain representations for image modeling. To ensure the frequency representation generally works with conventional transformers (e.g. ViT [12], SWIN transformer [33], etc.), we propose

a frequency embedding in addition to the positional embedding to maintain both spatial and frequency band information. We further empirically demonstrate that our DCFormer is able to capture the semantics directly from frequency representation without any performance compromise compared with RGB-based approaches.

As for the second challenge, strategically dropping the information from the input is non-trivial and it is challenging for the RGB-based representation, as previous research show that any types of pooling on the input will harm the performance [9]. Inspired by image compression approaches, we propose to strategically drop more high frequency components and less low-frequency components to better maintain the semantic information. We also introduce an reconstruction aux loss to help the training process.

We tested our model on image classification tasks with Imagenet-1K dataset, and object detection and instance segmentation tasks with MS-COCO [31]. The proposed DCFormer generalizes well to different transformer backbones without performance compromises, including SWIN [33], ViT [12], and NAT [19]. With the help of the proposed frequency down-sampling strategy, the DCFormer is capable of taking input images at different resolutions for better efficiency vs. effectiveness trade-off (Figure 1). We further show that the DCFormer is able to achieve performances comparable to commonly used RGB-based models at lower computational costs, demonstrating the frequency modeling is a promising direction for building efficient model. Our contributions are summarized as follows:

1. DCFormer, the transformer directly performs image modeling for various downstream tasks on DCT-based frequency representation. The DCFormer learns semantics directly from DCT based frequency representation without performance compromise.
2. Study different input down-sampling methods, and propose zigzag based hard-selection for DCT-based input compression. With the proposed input compression strategy, DCFormer is able to achieve even better efficiency-effectiveness trade-off.
3. Detailed experimental results and ablations, which can be used for future reference.

2. Related Work

2.1. Image Modeling

As the foundation of many computer vision tasks, the image classification has been studied for decades, from heavily rely on manually-crafted features [59, 28, 38] to the deep neural network [30] era, the deep learning dominates the image modeling since 2012 [29]. For the past decade, the networks go deeper, wider and more complex [44, 47, 21, 56] to fit into various tasks including classification [21], detection [40, 32], segmentation [20, 8], pose estimation [45] and more. Besides the network architectures, the convolution layers also evolve from the basic convolution to the depth-wise convolution [57], non-local convolution [53] and deformable convolution [10]. In parallel with convolution networks, the recent researches show that the attention based architecture, previously commonly used for NLP tasks [51], transfers to image modeling well. The pioneer work ViT [12] and the following works DEiT [50], SWIN [33], CoaT [58], and more recent Mixer [49] all achieve comparable or superior performances compared with convolution networks. The majority of works on image modeling focus on the performance while in this paper, we focus on both efficiency and effectiveness trade-off.

2.2. Frequency-domain Learning

Frequency domain learning gains much less attention compared with RGB domain modeling in past decades. Only a few works propose to make use of JPEG encoding for faster image classification [17, 14]. Although efficient, these works are less effective than the SOTA image classification models at their times. Some recent works try to incorporate the frequency components from DCT transformation to the channel for better modeling [2, 1], however, the effectiveness gap still exists. Besides, the frequency domain representation has also been used in compression [55, 35], pruning [35] and convnet compression [54, 13]. Although works, the frequency domain modeling generally suffers from the low accuracy and low efficiency, which make them less favored by many image downstream tasks. Our proposed DCFormer with image compression achieves parallel performance compared with SOTA RGB networks at much low computational costs, which stands out from previous frequency domain modeling works. The recent work WaveViT [60] achieves strong performance with discrete wavelet transformation based representation. We share the similar scope on frequency representation based modeling, but instead leverage DCT-based representation because of its flexibility to support strategical down-sampling that improves efficiency without performance compromise.

2.3. Efficient Image Modeling

There are several attempts for efficient image modeling. The convolution kernel or network compression [18, 22] is a straight-forward way to reduce model FLOPs but often leads to noticeable performance drop. Later the carefully designed compact networks with very small memory footage are proposed to work on edge devices including squeezeNet [24], MobileNets [23, 42], and ShuffleNets [61, 36]. More recently, the neural architecture search is widely used as a tool for searching the efficient and accurate network architectures e.g. Proxylessnas [5] and EfficientNet [48]. Different from these approaches that

try to build a smaller network, we propose to reduce computation based on frequency domain image compression.

3. Methodology

3.1. Frequency Domain Modeling

In this paper, we adopt DCT based frequency domain representation because it is commonly used for image compression [41], image encoding [43], and various computer vision tasks.

3.1.1 Domain Converting Preliminaries

For an RGB image $I_{RGB} \in \mathbf{R}^{3\times W\times H}$, we first convert the image color space from RGB to yCrCb color space (I_{YCrCb}) and patchfy the image as:

$$P = [P_0, P_1, \ldots, P_i] = \text{patchfy}(I_{YCrCb}) \tag{1}$$

where $P \in \mathbf{R}^{(3\times \frac{H}{ps}\times \frac{W}{ps}\times ps\times ps)}$ is a sequence of patches, ps denotes the side length of each patch. DCT [4] is applied on each patch to generate the frequency domain representation:

$$D_i = \text{DCT}(P_i) \tag{2}$$

where the DCT map of each patch $D_i \in \mathrm{R}^{3\times ps\times ps}$ has the same dimension as the original patch P_i.

The patchfy operation preserves the relative spatial information, while each point in a patch D_i carries certain frequency information. Patch size selection involves a trade-off. Smaller patches lead to higher spatial resolution but less fine-grained frequency information and opposite for the larger patch size. We empirically select 8×8 as patch size for the best efficiency-effectiveness trade-off, the same as JPEG's encoding process. Such design potentially allows us to directly obtain the DCT components from raw JPEG images for faster training and inference.

3.1.2 Frequency Domain Encoder

For a compressed frequency map S, the pixels no longer hold spatial relationships inside each patch. Different from previous works that try to shift frequency components to channels for the convolution based modeling [1, 2], we propose to build a network that directly works on the frequency map. The 2D Inverse-DCT (IDCT) transformation for each frequency patch can be mathematically formulated as:

$$\text{IDCT}(S_i) = A^{\intercal}(S_i)A \tag{3}$$

where A denotes the DCT transform matrix. The above equation can be further illustrated as below and is consistent with the transformer layer's formulation:

$$\text{IDCT}(S_i) = A^{\intercal}(S_i)A = (W_q \Lambda W_k^{\intercal})(W_v S_i)(W) \tag{4}$$

where W_k, W_q, W_v denotes the learnable linear projection for key, query and value. W denotes the learnable weights for linear layers after attention. Although it is not guaranteed that W is strictly the transpose of $(W_q \Lambda W_k^{\intercal})$, it is possible for transformers to learn the approximation of the IDCT. The observation in Equation 4 makes the transformer architecture a good fit for our compressed image modeling. Note that, it is possible for convolution networks to simulate IDCT through carefully designed kernel size and strides, but it will be less effective compared with transformer networks (ablations in Table 4d). The commonly used transformers, including sequence transformer (e.g. ViT[12]) and hierarchical vision transformer (e.g. SWIN[33], NAT [19]) work directly in the frequency domain with minor changes to the patch size and embedding.

Frequency Embedding: Each frequency point $S_{i,j}$ on S_i carries the relative positional information as well as certain frequency band information. To maintain the frequency information, we propose frequency embedding (FE) in addition to the commonly used positional embedding as:

$$\begin{aligned} FE_{(j,2k)} &= \sin(j/10000^{2k/dm}) \\ FE_{(j,(2k+1))} &= \sin(j/10000^{(2k+1)/dm}) \end{aligned} \tag{5}$$

where $j \in [0, ps^2]$ denotes the position in the down-sampled frequency patch S_i. k denotes the k-th dimension of total dm feature dimensions. We apply the frequency embedding by adding it to the frequency map S.

Classification: We unpatchfy the compressed patches based on their relative locations as:

$$\tilde{S} = \text{unpatchfy}(S) \tag{6}$$

where the unpatchfy operation reorganize a sequence of compressed DCT map S to $\tilde{S} \in \mathrm{R}^{3\times \frac{H}{\tau}\times \frac{W}{\tau}}$ based on their relative spatial location. The DCFormer encoder extras feature embedding from the compressed frequency domain representations as:

$$X^E = \phi(\tilde{S}) \tag{7}$$

where X^E stands for the DCFormer encoder feature map. The classification can be done by adding a [CLS]-token [12] or use a linear layer [33].

3.2. Efficient Frequency Domain Modeling

3.2.1 Frequency Domain Compression

The famous JPEG compression infers that compression can be achieved by discarding the non-visually significant values via quantization [52]. Following similar intuition, we aim maintain only the informative frequency components from each DCT map D_i for efficient image modeling. We explored three types of compression strategies as follows:

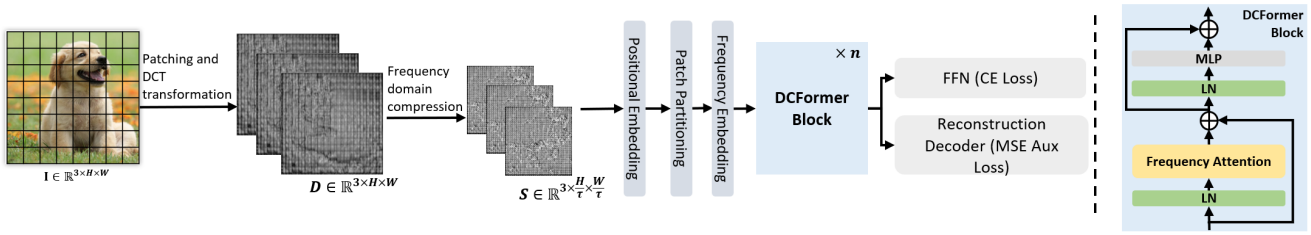

Figure 2: An overview of our model, DCFormer. The model first takes RGB image as input, converting it to DCT-based frequency representation, follows by an optional frequency compression module. The compression module (when τ>1) offers significant efficiency boost, with slight performance trade-off. The frequency based representation is then augmented with positional and frequency embedding and feed into a set of DCFormer blocks. The frequency attention is compatible with various transformer attentions (e.g. SWIN attention, neighbour attention). An linear projection with CE loss is used for classification, and a MSE reconstruction loss can be used as aux loss when frequency compression is applied.

Averaging: An average pooling with $\tau \times \tau$ kernel over the DCT map D_i as:

$$S_i(\frac{j}{\tau}, \frac{k}{\tau}) = \frac{1}{\tau^2}[D_i(j,k) + ... + D_i(j,k+\tau) + ... + D_i(j+\tau, k+\tau)] \quad (8)$$

where j and k are the coordinates of D_i. S_i denotes the compressed DCT map D_i and τ is the compression ratio.

Soft-selection: A cross-attention based soft-selection method on D_i as:

$$S_i = \text{MHCA}(Conv2D(D_i), q_{emb}) \quad (9)$$

where MHCA denotes the multi-head cross-attention module, $q_{emb} \in \mathrm{R}^{\frac{ps^2}{\tau^2} \times c}$ is the query embedding, $Conv2D$ is an 1×1 2D convolution that expands the channel of D^i, e.g. $c = 128$, to support multi-head attentions.

Hard-selection: A hard-selection follows zigzag pattern that focus on low frequency components as:

$$S_i = [\psi(D_i)]_k, k \in [1, \frac{ps^2}{\tau^2}] \quad (10)$$

where ψ is the zigzag encoding used in [37], $\psi(D_i) \in \mathrm{R}^{\frac{ps}{\tau} \times \frac{ps}{\tau}}$ stands for the sequence of frequency components after zigzag encoding, which are then sorted by their frequency band from low to high. To maintain visually significant information, we hard select the first $\frac{1}{\tau^2}$ elements from $\psi(D_i)$, which covers the DC component and most of low-frequency and part of middle frequency information.

We empirically choose zigzag-baed hard-selection for compression as it works best without introducing additional computation. The cross-attention based soft-selection is deprecated since it is computationally heavy. Averaging based approaches perform the worst since averaging frequency responses over different bands does not make sense.

3.2.2 Reconstruction Aux Loss

Because frequency compression causes information lose, we further introduce a reconstruction decoder adopting auxiliary loss during training which encourages the DCFormer encoder to generate comprehensive and semantic-related feature embeddings. The decoder is not used in the inference and hence does not introduce additional inference computations. It is worth mentioning that the decoder has to be lightweight with limited capacities, so that the decoder utilizes the encoded features as much as possible instead of learning the new semantics features by itself. We propose a simple eight-layer convolution neural network with 3×3 kernels as the decoder. Because convolution is less effective on frequency-to-RGB domain converting, the decoder has to rely on semantic information generated by DCFormer encoder to reconstruct the RGB image. The reconstruction process thus enforces the encoder to generate more comprehensive representation during training. The decoder has four up-sampling stages, each stage has two convolution layers and a spatial up-sampling layer defined as:

$$X_i^D = \begin{cases} X^E & i = 1 \\ conv2D(conv2D(U(X_{i-1}^D))) & i \in [2,4] \end{cases} \quad (11)$$

where U denotes the $4\times$ bilinear interpolation up-sampling. The X_i^D denotes the feature from i-th decoder stage.

3.2.3 Losses

We apply the categorical cross-entropy to the DCFormer encoder output as classification loss ($\mathcal{L}_{cls}$). For the reconstructed images, we calculate the MSE loss as a measure of reconstruction quality ($\mathcal{L}_{MSE}$). We also adopt the perceptual loss ($\mathcal{L}_{perceptual}$), which is commonly used in image super-resolution [25] tasks, to encourage the DCFormer encoder to generate semantic related representations. The final loss is thus defined as:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{MSE} + \beta\mathcal{L}_{perceptual} \quad (12)$$

We empirically determined the $\alpha = 0.1$ and $\beta = 0.01$.

Model	Size	FLOPs(G)	# Param.	Top1 (%)
DCT-64T[2]	256	-	-	77.2
FcaNet-TS-50[1]	256	4.13	28M	78.6
FcaNet-LF-152[1]	256	11.6	61M	80.1
WaveViT-S[60]	256	4.3	20M	82.7
WaveViT-B*[60]	256	7.2	33M	84.8
SWIN-T[33]	256	4.5	28M	81.2
DCFormer-SW-T ($\tau = 1$)	256	4.5	28M	81.2
DCFormer-SW-T ($\tau = 2$)	256/384/512	1.3/3.2/4.5	28M	79.2/81.2/82.1
SWIN-S[33]	256	8.7	50M	83.0
DCFormer-SW-S ($\tau = 1$)	256	8.7	50M	82.8
DCFormer-SW-S ($\tau = 2$)	256/384/512	2.7/6.3/8.7	50M	80.9/82.1/82.9
SWIN-B[33]	256	15.4	88M	83.5
DCFormer-SW-B ($\tau = 1$)	256	15.4	88M	83.1
DCFormer-SW-B ($\tau = 2$)	256/384/512	4.8/10.9/15.4	88M	81.4/82.9/83.5
NAT-T[19]	256	4.3	28M	83.1
DCFormer-NA-T ($\tau = 1$)	256	4.3	28M	83.2
DCFormer-NA-T ($\tau = 2$)	384	3.8	28M	82.6
NAT-B[19]	256	13.7	90M	84.3
DCFormer-NA-B ($\tau = 1$)	256	13.7	90M	84.3

Table 1: Frequency modeling results on ImageNet-1K validation set. $\tau = 1$ and $\tau = 2$ denotes frequency input without and with 4X compression. * denotes WaveViT [60] trained on additional data. DCFormer-SW and DCFormer-NA denotes DCFormer with SWIN [33] and NAT [19] backbones, respectively.

4. Experiments

We conduct image classification on ImageNet-1K dataset [11]; object detection and instance segmentation tasks on COCO object detection dataset [31]. We will first compare the proposed DCFormer with other frequency domain modeling approaches on each task, and then further establish comparison with SOTA RGB-based models. Finally we present the ablation analysis on the design choices and generalizability of DCFormer. We use $\tau = 1$ for DCFormer without frequency compression and $\tau = 2$ for DCFormer with $4\times$ frequency compression.

4.1. Frequency Domain Modeling Results

4.1.1 Classification on ImageNet

Setting: We follow [33] for ImageNet training with minor changes. We adopt the AdamW [27] optimizer and use a cosine learning rate scheduler. The training process starts with 30 epochs of linear warm-up, followed by 270 training epochs. A batch size of 1024, an initial learning rate of 0.001, and a weight decay of 0.05 are used similar to [33], the learning rate scales according to the batch size for different experiments. We follow the augmentation and regularization strategies of [50] in training, including color and size jittering, mixup and label smoothing, etc.

Results: We first demonstrate that the transformer is able to learn the semantics directly from frequency representation. We compare the image classification accuracy with SWIN[33] that takes RGB image as input and our DCFormer that takes frequency-based representation as input. The results (Table 1, $\tau = 1$) show that DCFormer is able to achieve similar performance comparing with different RGB backbones. This proves the validness of our intuition, that transformer can learn semantics from frequency representations directly without performance compromise.

We further study the performance of proposed frequency input compression on DCFormer (Table 1, $\tau = 2$) and comparing with previous frequency modeling works [1, 2]. Under the same input resolution, the DCFormer with SWIN transformer backbone outperforms most recent frequency domain modeling approach [1], with less FLOPs. The reduced FLOPs and better performance demonstrate our zigzag compression better maintains salient information than the commonly used channel-wise convolution based frequency domain compression methods [2]. DCFormer-NAT-T also slightly outperforms the mot recent image frequency modeling work using wavelet transformation [60]. It is worth mentioning that WaveVit-B achieves strong performance with additional data, and is not a direct comparison to other results. We notice by increasing the input resolution, while performing the proposed input compression, we see performance increase (Table 1, $\tau = 2$). The DCFormer with $\tau = 2$ and $4\times$ input resolution runs at same FLOPs comparing to DCFormer take input without compression, but with comparable or slightly better performance (e.g., on DCFormer-SW-T vs. SWIN-T).

4.1.2 Object Detection on COCO

Setting: We finetune the DCFormer-SWIN with Mask R-CNN [20] pipeline on the COCO 2017 dataset [31]. During fine-tuning, we use the multi-scale training [46, 7], AdamW optimizer[27], with weight decay of 0.05, and the same learning decay schedule as [6, 34]. The pipeline follows [33]. Because our image compression leads to smaller feature maps at the final stage, we decrease the spatial scale factor of the RoIAlign and FPN layers to match the scale of the feature maps, e.g. scale $= 2$ for $\tau = 2$.

Results: The proposed DCFormer outperforms other frequency domain modeling works [1, 2, 60] by a significant margin (Table 2). Under Mask R-CNN setup, the DCFormer-T outperforms [2] by 3.5% with half of FLOPs and DCFormer-S outperforms [1] by 3.8% with only 60% of its FLOPs. The DCFormer stands out in detection tasks among other frequency domain based methods because: (1) instead of squeezing the frequency band into the channels, the DCFormer is able to maintain the feature map used for ROIalign at a reasonable size, which is critical for detection tasks; (2) the transformer better models the frequency-domain representation compared with stacked convolutions.

4.1.3 Instance Segmentation on COCO

Setting: We also evaluate the instance segmentation performance on the COCO dataset, and our training follows

Backbone	Input size	FLOPs	Tr.(f/s)	Object Detection			Instance Segmentation		
				AP^{box}	$AP^{box}_{0.5}$	$AP^{box}_{0.75}$	AP^{box}	$AP^{box}_{0.5}$	$AP^{box}_{0.75}$
ResNet-50-FPN	800 × 1333	-	-	37.3	59.0	40.2	34.2	54.9	36.2
SWIN-T[33]	800 × 1333	267G	23.1	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T[34]	800 × 1333	262G	25.6	46.2	67.9	50.8	41.7	65.0	44.9
DCT-64S[2]	1600 × 2666	-	-	38.1	59.6	41.1	35.0	56.5	37.4
FcaNet-LF[1]	800 × 1333	262G	-	40.3	61.9	43.9	36.3	58.3	38.6
FcaNet-TS[1]	800 × 1333	262G	-	40.3	62.0	44.1	36.2	58.6	38.1
FcaNet-NAS[1]	800 × 1333	262G	-	40.3	61.9	43.9	36.3	58.3	38.6
WaveViT-S[60]	800 × 1333	-	-	42.4	65.5	45.8	-	-	-
WaveViT-B[60]	800 × 1333	-	-	43.0	66.4	46.0	-	-	-
DCFormer-SW-T	800 × 1333	116G	39.3	41.6	63.2	44.8	37.9	59.3	40.3
DCFormer-SW-S	800 × 1333	139G	34.2	44.1	65.4	48.1	39.2	61.7	42.2
DCFormer-SW-T	1200 × 2000	183G	28.6	44.4	66.2	47.4	40.0	62.8	43.3
DCFormer-SW-S	1200 × 2000	235G	25.9	46.4	67.7	49.8	42.7	64.6	44.1

Table 2: Comparison on COCO Object detection and instance segmentation on 5k validation set, with 800 × 1333 input images. DCFormer-SW and DCFormer-NA denots DCFormer with SWIN [33] and NAT [19] as backbones, respectively.

Model	Size	FLOPs	# Param.	Top1 (%)
ResNet-50[21]	256	3.8G	26M	79.3
ResNet-101[21]	256	7.6G	45M	80.1
RegNetY-4G[39]	256	4.0G	21M	80.0
RegNetY-8G[39]	256	8.0G	39M	81.7
RegNetY-16G[39]	256	16.0G	84M	82.9
EffiNet-B3[48]	300	1.8G	12M	81.6
EffiNet-B4[48]	380	4.2G	19M	82.9
EffiNet-B5[48]	456	9.9G	30M	83.6
SWIN-T[33]	256	4.5G	28M	81.2
SWIN-S[33]	256	8.7G	50M	83.0
SWIN-B[33]	256	15.4G	88M	83.5
ConvNeXt-T[34]	256	4.5G	29M	82.1
ConvNeXt-S[34]	256	8.7G	50M	83.1
ConvNeXt-B[34]	256	15.4G	89M	83.8
NAT-T[34]	256	4.3G	29M	83.2
NAT-S[34]	256	7.8G	50M	83.5
NAT-B[34]	256	13.7G	89M	84.3
DCFormer-SW-S ($\tau = 2$)	256	2.7G	28M	80.9
DCFormer-SW-T ($\tau = 2$)	512	4.5G	28M	82.1
DCFormer-NA-T ($\tau = 2$)	384	3.8G	28M	82.6
DCFormer-NA-B ($\tau = 1$)	256	13.7G	90M	84.1

(a) Comparing with SOTA RGB-based model on ImageNet classification tasks.

Object Detection with Cascade Mask R-CNN setup

Backbone	FLOPs	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
SWIN-T[33]	745G	50.3	69.1	54.3
ConvNeXt-T[34]	741G	50.4	69.2	54.7
SWIN-S[33]	838G	51.9	70.7	56.3
ConvNeXt-S[34]	827G	51.9	70.8	56.5
DCFormer-T	595G	43.5	62.6	47.4
DCFormer-S	618G	46.6	64.9	50.4
DCFormer-T (1.5×)	661G	48.4	67.6	52.6
DCFormer-S (1.5×)	714G	50.1	68.8	54.1

Instance Segmentation with Cascade Mask R-CNN setup

Backbone	FLOPs	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
SWIN-T[33]	745G	43.7	66.6	47.3
ConvNeXt-T[34]	741G	43.7	66.5	47.3
SWIN-S[33]	838G	45.0	68.2	48.8
ConvNeXt-S[34]	827G	45.0	68.4	49.1
DCFormer-T	595G	38.0	59.3	40.7
DCFormer-S	618G	40.5	62.2	43.7
DCFormer-T (1.5×)	661G	42.1	65.7	46.5

(b) Comparison on COCO 5K validation set.DCFormer with SWIN backbone and $\tau = 2$ is used.

Table 3: Comparing with RGB-based works on image classification, object detection and instance segmentation. Detection and instance segmentation tasks run on 800 × 1333 input images, 1.5× denotes 1.5 times larger input images. DCFormer-SW/NA denotes DCFormer with SWIN [33] and NAT [19] as backbones, respectively.

the same protocol used in the detection experiments.

Results: DCFormer achieves consistent performance improvements on instance segmentation (Table 2) compared with other frequency domain modeling approaches [1, 2].

4.2. Comparing with RGB-based SOTA

Classification Table 3 (a) shows our results on ImageNet-1K validation set compared with the previous works based on convolution [21, 34], transformer [33, 50]. All the models listed are only trained on ImageNet-1K from scratch. We find that the proposed DCFormer is able to operate at lower computational budget to maintain similar performance comparing with commonly used RGB models. For example, DCFormer-SWIN-S ($= 2$) achieves 80.9% top1 accuracy with only 2.7G FLOPs, significantly more efficient than SWIN-T [33]. The DCFormer-NA-T also achieves slightly better performance at lower FLOPs comparing with recent works ConvNeXT-T [34] and SWIN-T [33]. Furthermore, the DCFormer is able to achieve performance comparable to RGB SOTA at same computational

Module	*FLOPs*	*Top1*
SWIN-T(112^2 RGB)	1.31G	78.54
+Frequency selection	1.31G	78.88
+Frequency embedding	1.31G	79.19
+reconstruction decoder	1.31G	79.35

(a) **Building components.** Each proposed components helps with performance, the frequency selector and reconstruction decoder helps most.

τ	*# selected*	*FLOPs*	*Top1*
4	4	0.39G	73.3
2	16	1.31G	79.2
1.3	36	3.05G	80.4
1	64	4.50G	81.2

(b) **Compression ratio.** Lower compression ratio gives higher accuracy but also higher FLOPs.

Input size	*DCT Patch*	*Top1*
256×256	4^2	70.70
256×256	8^2	78.88
256×256	16^2	78.93
512×512	8^2	81.15
512×512	16^2	82.03

(c) **DCT patch size.** Larger DCT patch size on larger input images lead to better performance.

Backbone	*FLOPs*	*Top1*	*RGB Top1*
ResNet-50[21]	1.0G	75.7	79.3
ResNet-101[21]	2.1G	77.5	80.1
ViT-B[12]	14.9G	75.2	77.9
SWIN-T[33]	1.3G	79.2	81.2

(d) **Generalization.** Transformer based backbone generally works better as encoder.

Selector	*FLOPs*	*Top1*
no sampling	4.1G	81.1
average pooling	1.3G	57.9
cross-attention	3.8G	77.8
zigzag	1.3G	79.2

(e) **Compression methods.** The zigzag works better than others.

Input	*Size*	*FLOPs*	*Top1*
RGB	256	4.1G	81.2
RGB	112	1.3G	78.5
Reconstructed. RGB	112	1.3G	79.0
DCT	112	1.3G	79.2

(f) **Effectiveness.** The propose compression is more effective than spatial down-sampling.

Table 4: Ablation studies on ImageNet-1K. All the experiments use DCFormer-SW-T and images of 256×256 as backbone without the reconstruction decoder, unless specified.

budget. For example, the DCFormer-SWIN-T ($\tau = 2$) with 512×512 input resolution slightly outperforms SWIN-T [33] at same FLOPs. The DCFormer-NA-B also achieves performance comparable to SOTA NAT-B [19] with same input resolution and same FLOPs. The results demonstrate the outstanding efficiency and effectiveness of the proposed approach.

Object Detection To compare with SOTA RGB models, we trained DCFormer with SWIN backbone on cascade mask RCNN pipeline. The proposed DCFormer is able to reduce the FLOPs and latency on object detection tasks (Table 3 (b)). Giving the same input image resolution, the DCFormer-SWIN-S achieves slightly worse performance compared with the SOTA SWIN-T and ConvNeXt-T models but with 15% less FLOPs. Similar to image classification, the performance gap can be compensated by higher input resolution without significant FLOPS increase. By scaling up the input image by $1.5\times$ (1200×1666), the DCFormer-SWIN-S is able to achieve comparable performance with 11% less FLOPs.

Instance Segmentation The DCFormer with large input resolution achieves similar performance compared with SOTA SWIN [33] and ConvNeXT [34] based approaches (Table 3 (c)). We notice that the DCFormer maintains good efficiency due to the proposed frequency compression. However, the DCFormer performs slightly worse on instance segmentation tasks. This is probably due to the reduced feature map size and lack of high-frequency (texture) information. There are researches showing that texture information helps with instance segmentation [26]. To better preserve these textures during the compression as well as maintaining the low computation will be our future work.

4.3. Ablation Study

We justify the important design choices, effectiveness and generalization of proposed model on ImageNet-1K image classification task. All the experiments are performed on DCFormer-SWIN-T. The images are 256×256 resolution and have 8×8 frequency patch size, with $\tau = 2$ are used, unless specified.

Building components breakdown. Table 4a analyzes the contribution of each proposed components, by adding them one at a time, to a standard SWIN transformer. We use a SWIN transformer on an RGB image down-sampled to 112×112 as the baseline (same FLOPs). By performing hard-selection on the frequency components, we boost the performance by 0.34% without introducing additional computation. This also verifies our intuition that the frequency domain down-sampling better preserves information that is visually significant. The proposed frequency embedding slightly enhance the performance by 0.2%. Additionally, the reconstruction decoder achieves a slight improvement by 0.16% which shows that our decoder works as expected. Note that the decoder only introduces additional FLOPs in training but it is not used during inference.

Compression ratio. Table 4b compares the classification performance under different frequency compression ratios. The larger compression ratio, e.g. $\tau = 4$, leads to lower FLOPs but lower accuracy since more information was dropped; same for the opposite. Based on the ablation, we choose $\tau = 0.5$ for the best efficiency and effectiveness trade-off. It is also interesting to see that the DCFormer with no frequency down-sampling ($\tau = 1$) achieves the same accuracy and FLOPs as SWIN-T with RGB image input. This indicates the frequency domain representation is as effective as the RGB representation, as we argue that transformer is a good fit for frequency domain modeling.

DCT Patch Size. Table 4c studies the impact of using different DCT patch sizes on images of different resolutions. In general, smaller DCT patch size, e.g. 8^2 works better on smaller images (e.g. 256^2, 384^2). Further increasing the input resolution with same DCT patch size does not consistently enhance the performance, because small DCT patches with limited DCT bases only contain limited information. Larger DCT patches convey more frequency information and yield better performance on inputs with high-resolution. Dynamically adjusting DCT patch size will be our future research.

Generalization. Table 4d explores the generalization of proposed image compression with different backbones. Our approach generalizes to different backbones. The transformer-based encoders generally yield less performance drop by using frequency domain inputs, which proves our illustration that the attention can simulate inverse-DCT operation more effectively. It is also worth mentioning that the ViT-B has high FLOPs due to its lack of multi-scale feature hierarchy.

Compression method. Table 4e compares different frequency compression methods. We first notice that average pooling which is commonly used in spatial down-sampling causes significant performance drop when applied to the frequency domain. This is because averaging data points that belong to different frequency bands does not make sense as they do not have direct spatial associations. We then try to use the cross-attention to learn the weighted average based compression. However, cross-attention requires applying extra convolution layers on input DCT map, which introduces additional computation and makes the compression less efficient and contradict to our motivations. The zigzag selection works best at no additional computations in our case. Similar approach was used in JPEG compression and similar patterns were observed [2].

Effectiveness. Table 4f compares and analyzes several alternatives to the proposed image modeling at similar FLOPs, including: directly down-sample the input image by $2\times$; and after proposed frequency domain compression, reconstruct the RGB image with IDCT and feed it into the standard SWIN transformer. For better comparison, the SWIN-T with RGB input of 256×256 is used as the baseline. The results indicate that the proposed method is more effective than the alternative, as the reconstruction may suffer from noises introduced during padding and converting.

5. Visualization

To qualititively show the proposed frequency domain modeling learns the semantics, we visualize the activation of DCFormer-SWIN wtih attention rollout [3].

Figure 3 visualizes and compares the features learned by SWIN transformer and our DCFormer-SWIN-T ($\tau = 2$). The activation maps are extracted from the last stage of backbone and overlayed to the input image. For most cases, the attentions from SWIN transformer and our DCFormer fall onto the same regions, which indicates the DCFormer learns the same semantic representations as RGB domain modeling (Figure 3, top). We notice that in a few cases, the activation map from SWIN transformer covers broader regions (Figure 3, bottom), this is probably because the SWIN transformer generates $4\times$ larger feature maps than DCFormer giving the same input image.

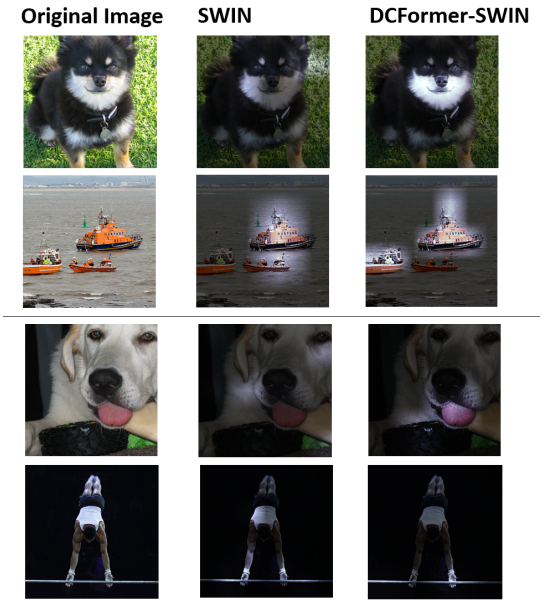

Figure 3: Activation from SWIN-T [33] and DCFormer-SWIN. Most cases share similar activation for both models (top); for some cases SWIN covers larger regions (bottom).

6. Conclusion

In this paper, we introduce DCFormer that enables transformer to learn semantics directly from DCT-based frequency domain representation. Based on DCFormer, we further introduce an frequency input down-sampling method. DCFormer achieves the performance comparable to commonly used transformer backbones with no performance compromise. With proposed frequency input compression, the DCFormer is able to achieve better efficiency-effectiveness trade-off comparing with previous frequency modeling approaches. We hope that these promising results reported will encourage the research on efficient modeling from another perspective and the implementation of proposed approach to many downstream tasks. Exploring the transformer based frequency domain modeling approach with other frequency representation, e.g. discrete wavelet transformation, and refining the frequency compression for better performance will be our future work.

References

[1] Fcanet: Frequency channel attention networks. In *CVPR 2021*.

[2] Learning in the frequency domain. In *CVPR 2020*.

[3] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[4] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[5] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[6] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021.

[7] 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.

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

[9] Tianshui Chen, Liang Lin, Wangmeng Zuo, Xiaonan Luo, and Lei Zhang. Learning a wavelet-like auto-encoder to accelerate deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

[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] 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.

[13] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. Band-limited training and inference for convolutional neural networks. In *International Conference on Machine Learning*, pages 1745–1754. PMLR, 2019.

[14] Max Ehrlich and Larry S Davis. Deep residual learning in the jpeg transform domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3484–3493, 2019.

[15] Ahmet M Eskicioglu and Paul S Fisher. Image quality measures and their performance. *IEEE Transactions on communications*, 43(12):2959–2965, 1995.

[16] Ge Gao, Pei You, Rong Pan, Shunyuan Han, Yuanyuan Zhang, Yuchao Dai, and Hojae Lee. Neural image compression via attentional multi-scale back projection and frequency decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14677–14686, 2021.

[17] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. *Advances in Neural Information Processing Systems*, 31, 2018.

[18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[19] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022.

[20] 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.

[21] 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.

[22] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–800, 2018.

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[24] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[25] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[26] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12975–12984, 2020.

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

[28] Takumi Kobayashi. Bfo meets hog: feature extraction based on histograms of oriented pdf gradients for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 747–754, 2013.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural net-

works. *Advances in neural information processing systems*, 25, 2012.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[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] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[33] 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.

[34] 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.

[35] Zhenhua Liu, Jizheng Xu, Xiulian Peng, and Ruiqin Xiong. Frequency-domain dynamic pruning for convolutional neural networks. *Advances in neural information processing systems*, 31, 2018.

[36] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[37] Pablo Montero, Javier Taibo, Victor Gulías, and Samuel Rivas. Parallel zigzag scanning and huffman coding for a gpu-based mpeg-2 encoder. In *2010 IEEE International Symposium on Multimedia*, pages 97–104. IEEE, 2010.

[38] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010.

[39] 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.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[41] Subhasis Saha. Image compression—from dct to wavelets: a review. *XRDS: Crossroads, The ACM Magazine for Students*, 6(3):12–21, 2000.

[42] 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.

[43] Bo Shen and Ishwar K Sethi. Inner-block operations on compressed images. In *Proceedings of the third ACM international conference on Multimedia*, pages 489–498, 1995.

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

[45] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[46] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021.

[47] 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.

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

[49] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34, 2021.

[50] 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.

[51] 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.

[52] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[54] Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Packing convolutional neural networks in the frequency domain. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2495–2510, 2018.

[55] Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. Cnnpack: Packing convolutional neural networks in the frequency domain. *Advances in neural information processing systems*, 29, 2016.

[56] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.

[57] 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.

[58] 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.

[59] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Conference on computer vision and pattern recognition*, pages 1794–1801. IEEE, 2009.

[60] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. *arXiv preprint arXiv:2207.04978*, 2022.

[61] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.