# FcaNet: Frequency Channel Attention Networks

Zequn Qin[1], Pengyi Zhang[1], Fei Wu[1,2], Xi Li[1,2*]

[1]College of Computer Science, Zhejiang University,
[2]Shanghai Institute for Advanced Study, Zhejiang University

zequnqin@gmail.com, pyzhang@zju.edu.cn, wufei@cs.zju.edu.cn, xilizju@zju.edu.cn

## Abstract

*Attention mechanism, especially channel attention, has gained great success in the computer vision field. Many works focus on how to design efficient channel attention mechanisms while ignoring a fundamental problem, i.e., using global average pooling (GAP) as the unquestionable pre-processing method. In this work, we start from a different view and rethink channel attention using frequency analysis. Based on the frequency analysis, we mathematically prove that the conventional GAP is a special case of the feature decomposition in the frequency domain. With the proof, we naturally generalize the pre-processing of channel attention mechanism in the frequency domain and propose FcaNet with novel multi-spectral channel attention. The proposed method is simple but effective. We can change only one line of code in the calculation to implement our method within existing channel attention methods. Moreover, the proposed method achieves state-of-the-art results compared with other channel attention methods on image classification, object detection, and instance segmentation tasks. Our method could improve by 1.8% in terms of Top-1 accuracy on ImageNet compared with the baseline SENet-50, with the same number of parameters and the same computational cost. Our code and models will be made publicly available.*

## 1. Introduction

As an important and challenging problem in feature modeling, attention mechanisms for convolutional neural networks (CNNs) have recently attracted considerable attention and are widely used in many fields like computer vision [35] and natural language processing [29]. In principle, they aim at selectively concentrating on some important information and have many types of variants (e.g., spatial attention, channel attention, and self-attention) corresponding to different feature dimensions. Due to the simplicity and
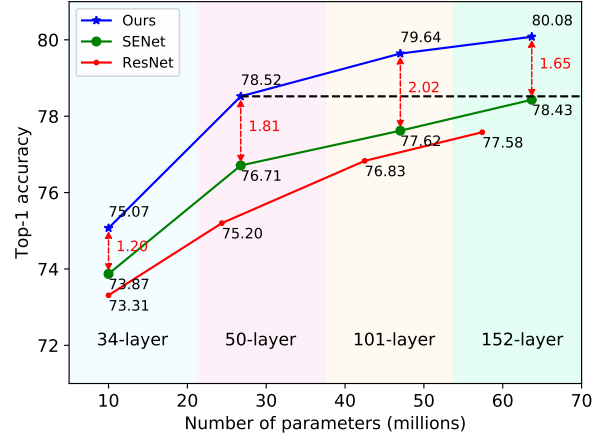
---
*Corresponding author.



Figure 1. Classification accuracy comparison on ImageNet. With the same number of parameters and computational cost, our method consistently outperforms the baseline SENet by a large margin. Our method with the ResNet-50 backbone could even outperform SENet with the ResNet-152 backbone.

effectiveness in feature modeling, channel attention directly learns to attach importance weights with different channels, becoming a popular and powerful tool for the deep learning community.

In the literature, conventional channel attention approaches devote great efforts to constructing various channel importance weight functions (e.g., SENet [19] using fully connected layers and ECANet [31] based on local one-dimensional convolutions). Typically, such weight functions require a scalar for each channel to conduct the calculation due to the constrained computational overhead, and global average pooling (GAP) becomes the de-facto standard choice in the deep learning community because of its simplicity and efficiency. Despite the simplicity and efficiency, there exists a potential problem that GAP is incapable of well capturing the rich input pattern information, and thus lacks feature diversity when processing different inputs. Consequently, there arises a natural question of whether the mean value information only is adequate for representing various channels in channel attention.

With the above motivation, we carry out a theoretical analysis of the GAP for channel attention in the following aspects. First, there exist numerous possibilities that different channels could have the same mean values, while their corresponding semantic content information is distinct. Second, from the perspective of frequency analysis, we prove that the GAP is equivalent to the lowest frequency of the discrete cosine transform (DCT) and only using GAP is equivalent to discarding the other frequency components containing much useful information on feature channels. Third, CBAM [34] also shows that only using GAP is not enough and adopts both GAP and global max pooling to enhance feature diversity.

In this paper, we propose a simple, novel, but effective multi-spectral channel attention framework. First, we mathematically prove that GAP is a special case of DCT frequency analysis, and it is equivalent to the lowest frequency component of DCT. Second, we naturally generalize the existing GAP channel attention mechanism in the frequency domain. We propose to use multiple but limited frequency components instead of one single GAP in the attention mechanism. By incorporating more frequency components into the attention processing, the information from these different frequency components can be exploited, leading to a multi-spectral description. As a result, the problem of insufficient information used in channel attention from the single frequency (i.e., GAP) can be addressed. Third, based on performance evaluations, we propose a two-step feature selection criterion for choosing different frequency components in the attention mechanism. Using the feature selection criterion, the proposed multi-spectral channel attention framework achieves state-of-the-art performance against the other channel attention ones.

In a word, the main contribution of this work can be summarized as follows.

- We prove that GAP is a special case of DCT. Based on this proof, we generalize the channel attention in the frequency domain and propose FcaNet with the multi-spectral channel attention framework.

- We propose a two-step criterion for choosing frequency components by exploring the effects of using different numbers of frequency components as well as their different combinations.

- Extensive experiments demonstrate the proposed method achieves state-of-the-art results on both ImageNet and COCO datasets. Based on the ResNet50 backbone, it could outperform SENet by $1.8\%$ in terms of Top-1 accuracy on ImageNet, with the same number of parameters and computational cost. The results are shown in Fig. 1.

- Our method is simple yet effective and can be implemented with only one line change of code within existing channel attention implementations.

## 2. Related Work

**Attention Mechanism in CNNs** In [35], a visual attention method is first proposed to model the importance of features in the image caption task. Then many methods start to focus on the attention mechanism. A residual attention network [30] is proposed with a spatial attention mechanism using downsampling and upsampling. Besides, SENet [19] proposes the channel attention mechanism. It performs GAP on the channels and then calculates the weights of each channel using fully connected layers. What's more, GE [18] uses spatial attention to better exploit the feature context, and $A^2$-Net [6] builds a relation function for image or video recognition. Inspired by these works, a series of works like BAM [24], DAN [8], CBAM [34], and scSE [27] are proposed to fuse spatial attention [40] and channel attention. Among them, CBAM claims that GAP could only get a sub-optimal feature because of the loss of information. For addressing this problem, it uses both the GAP and the global max pooling and gains significant performance improvement. Motivated by CBAM, GSoP [10] introduces a second-order pooling method for downsampling. Non-Local [32] proposes to build a dense spatial feature map. AANet [2] proposes to embed the attention map with position information into the feature. SkNet [20] introduces a selective channel aggregation and attention mechanism, and ResNeSt [37] proposes a similar split attention method. Due to the complicated attention operation, these methods are relatively large. To improve efficiency, GCNet [3] proposes to use a simple spatial attention module and replace the original spatial downsampling process. ECANet [31] introduces one-dimensional convolution layers to reduce the redundancy of fully connected layers and obtains more efficient results.

Besides these works, many methods try to extend the attention mechanism to specific tasks, like multi-label classification [13], saliency detection [39], visual explanation [9], and super-resolution [38].

**Frequency Domain Learning** Frequency analysis has always been a powerful tool in the signal processing field. In recent years, some applications of introducing frequency analysis in the deep learning field emerge. In [7, 12], frequency analysis is introduced in the CNNs by JPEG encoding. Then, DCT is incorporated in [36] to reduce communication bandwidth. There are also some applications in the model compression and pruning tasks like [5, 23, 33].

## 3. Method

In this section, we first revisit the formulation of channel attention and DCT frequency analysis. Then, based on these works, we elaborate on the derivation of our ==multi-spectral channel attention framework==. Meanwhile, a two-step criterion for choosing frequency components in the framework is also proposed. At last, we give discussions about effectiveness, complexity, and code implementation.

### 3.1. Revisiting Channel Attention and DCT

We first elaborate on the definitions of channel attention mechanism and discrete cosine transform. Then, we briefly summarize the properties of channel attention and DCT.

**Channel Attention**   The channel attention mechanism is widely used in CNNs. It uses a learnable network to weight the importance of each channel and generates more informative outputs. Suppose $X \in \mathbb{R}^{C \times H \times W}$ is the image feature tensor in networks, $C$ is the number of channels, $H$ is the height of the feature, and $W$ is the width of the feature. Then the attention mechanism can be written as [19, 31]:

$$att = sigmoid(fc(gap(X))), \qquad (1)$$

where $att \in \mathbb{R}^C$ is the attention vector, $sigmoid$ is the Sigmoid function, $fc$ represents the mapping functions like fully connected layer or one-dimensional convolution, and $gap$ is the global average pooling. After obtaining the attention vector of all $C$ channels, each channel of input $X$ is scaled by the corresponding attention value:

$$\widetilde{X}_{:,i,:,:} = att_i X_{:,i,:,:}, \quad s.t. \ i \in \{0, 1, \cdots, C-1\}, \quad (2)$$

in which $\widetilde{X}$ is the output of attention mechanism, $att_i$ is the $i$-th element of attention vector, and $X_{:,i,:,:}$ is the $i$-th channel of input.

**Discrete Cosine Transform (DCT)**   Typically, the definition of DCT can be written as [1]:

$$f_k = \sum_{i=0}^{L-1} x_i \cos(\frac{\pi k}{L}(i + \frac{1}{2})), \ \ s.t. \ k \in \{0, 1, \cdots, L-1\}, \quad (3)$$

in which $f \in \mathbb{R}^L$ is the frequency spectrum of DCT, $x \in \mathbb{R}^L$ is the input, and $L$ is the length of the input $x$. Moreover, two-dimensional (2D) DCT can be written as:

$$f_{h,w}^{2d} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} \underbrace{\cos(\frac{\pi h}{H}(i + \frac{1}{2})) \cos(\frac{\pi w}{W}(j + \frac{1}{2}))}_{\text{DCT weights}},$$
$$s.t. \ h \in \{0, 1, \cdots, H-1\}, w \in \{0, 1, \cdots, W-1\}, \quad (4)$$

in which $f^{2d} \in \mathbb{R}^{H \times W}$ is the 2D DCT frequency spectrum, $x^{2d} \in \mathbb{R}^{H \times W}$ is the input, $H$ is the height of $x^{2d}$, and $W$ is the width of $x^{2d}$. Correspondingly, the inverse 2D DCT can be written as:

$$x_{i,j}^{2d} = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} f_{h,w}^{2d} \underbrace{\cos(\frac{\pi h}{H}(i + \frac{1}{2})) \cos(\frac{\pi w}{W}(j + \frac{1}{2}))}_{\text{DCT weights}},$$
$$s.t. \ i \in \{0, 1, \cdots, H-1\}, j \in \{0, 1, \cdots, W-1\}. \quad (5)$$

Please note that in Eqs. 4 and 5, some constant normalization factors are removed for simplicity, which will not affect the results in this work.

With the definitions of channel attention and DCT, we can summarize two key properties: a) existing methods use GAP as their pre-processing before channel attention. b) DCT can be viewed as a weighted sum of inputs with the cosine parts in Eqs. 3 and 4 being the weights.

GAP is an operation of mean value due to constrained computational overhead, and it can be viewed as the simplest spectrum of input. As described in the introduction section, it is inadequate to use single GAP information in channel attention. Motivated by these properties, we can proceed to introduce our multi-spectral channel attention method.

### 3.2. Multi-Spectral Channel Attention

In this section, we first theoretically discuss the problem of existing channel attention mechanisms. Based on the theoretical analysis, we then elaborate on the network design of the proposed method.

**Theoretical Analysis of Channel Attention**   As discussed in Sec. 3.1, DCT can be viewed as a weighted sum of inputs. We further propose that GAP is actually a special case of 2D DCT.

**Theorem 1.** *GAP is a special case of 2D DCT, and its result is proportional to the lowest frequency component of 2D DCT.*

*Proof.* Suppose $h$ and $w$ in Eq. 4 are 0, we have:

$$f_{0,0}^{2d} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} \cos(\frac{0}{H}(i + \frac{1}{2})) \cos(\frac{0}{W}(j + \frac{1}{2}))$$
$$= \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} \qquad (6)$$
$$= gap(x^{2d})HW.$$

In Eq. 6, $f_{0,0}^{2d}$ represents the lowest frequency component of 2D DCT, and it is proportional to GAP. In this way, theorem 1 is proved. ∎

(a) Original SENet


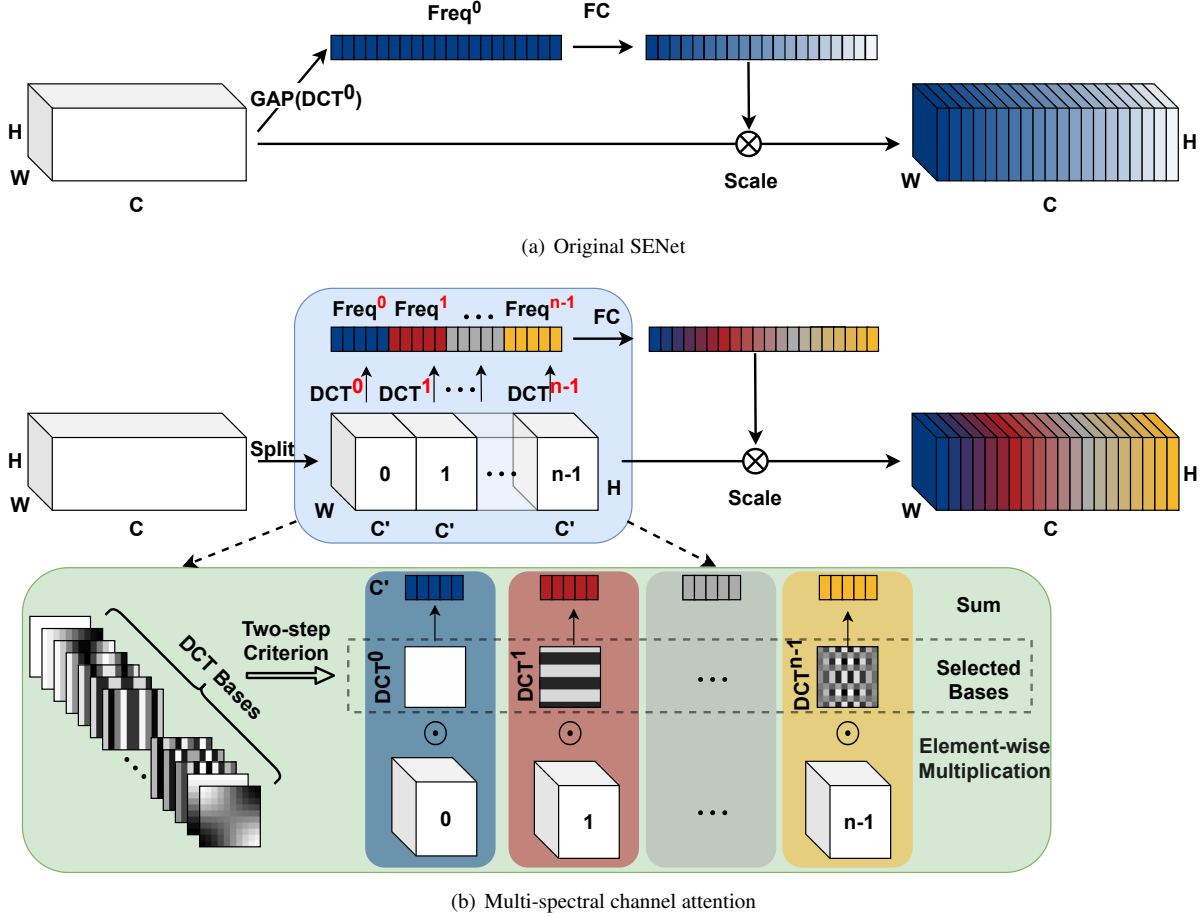
(b) Multi-spectral channel attention

Figure 2. Illustration of existing channel attention and multi-spectral channel attention. For simplicity, the 2D DCT indices are represented in the one-dimensional format. We can see that our method uses multiple frequency components with the selected DCT bases, while SENet only uses GAP in channel attention. Best viewed in color.

Based on theorem 1 that GAP is a special case of 2D DCT, this prompt that we could also incorporate other frequency components in the channel attention mechanism. Moreover, we can discuss the reason why we need to incorporate other frequency components using 2D DCT.

For simplicity, we use $B$ to represent the basis functions of 2D DCT:

$$B_{h,w}^{i,j} = \cos(\frac{\pi h}{H}(i+\frac{1}{2}))\cos(\frac{\pi w}{W}(j+\frac{1}{2})).\quad(7)$$

Then, the 2D DCT in Eq. 5 can be rewritten as:

$$x_{i,j}^{2d} = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1} f_{h,w}^{2d}\cos(\frac{\pi h}{H}(i+\frac{1}{2}))\cos(\frac{\pi w}{W}(j+\frac{1}{2}))$$

$$\stackrel{\text{Eq. }7}{=} f_{0,0}^{2d}B_{0,0}^{i,j} + f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}$$

$$\stackrel{\text{Eq. }6}{=} gap(x^{2d})HWB_{0,0}^{i,j} + f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}$$

$$s.t.\ \ i \in \{0,1,\cdots,H-1\}, j \in \{0,1,\cdots,W-1\}.$$

$$(8)$$

It is natural to see that an image or feature can be represented as a combination of different frequency components. According to Eq. 1, we have:

$$att = sigmoid(fc(gap(X))).\quad(9)$$

Channel attention is only based on the results of the GAP. However, combined with Eq. 8, we can see that the information of input $X$ is not only composed of GAP[1]:

$$X = \underbrace{gap(X)HWB_{0,0}^{i,j}}_{\text{utilized}} + \underbrace{f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}}_{\text{discarded}}.$$

$$(10)$$

The term $HWB_{0,0}^{i,j}$ is a constant scale factor and can be ignored in the attention mechanism. In this way, only a small part of the information is used by the channel attention mechanism. The other frequency components and information are discarded in the existing channel attention methods.

---

[1]More rigorous derivation can be found in the appendix.

4

**Multi-Spectral Attention Module**   Based on the theoretical analysis and theorem 1, we find the information used in existing channel attention is inadequate, and the pre-processing method GAP of channel attention is a special case of 2D DCT. In this way, we could naturally generalize GAP to more frequency components of 2D DCT and introduce more information to solve the problem of inadequate information in channel attention.

To introduce more information, we propose to use multiple frequency components of 2D DCT, including the lowest frequency component, i.e., GAP.

First, the input $X$ is split into many parts along the channel dimension. Denote $[X^0, X^1, \cdots, X^{n-1}]$ as the parts, in which $X^i \in \mathbb{R}^{C' \times H \times W}$, $i \in \{0, 1, \cdots, n-1\}$, $C' = \frac{C}{n}$, and $C$ should be be divisible by $n$. For each part, a corresponding 2D DCT frequency component is assigned, and the 2D DCT results can be used as the pre-processing results of channel attention. In this way, we have:

$$Freq^i = 2\text{DDCT}^{u,v}(X^i),$$
$$= \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} X^i_{:,h,w} B^{u,v}_{h,w} \qquad (11)$$
$$s.t. \ \ i \in \{0, 1, \cdots, n-1\},$$

in which $[u, v]$ are the frequency component 2D indices corresponding to $X^i$, and $Freq^i \in \mathbb{R}^{C'}$ is the $C'$-dimensional vector after the pre-processing. The whole pre-processing vector can be obtained by concatenation:

$$Freq = \text{cat}([Freq^0, Freq^1, \cdots, Freq^{n-1}]), \qquad (12)$$

in which $Freq \in \mathbb{R}^C$ is the obtained multi-spectral vector. The whole multi-spectral channel attention framework can be written as:

$$ms\_att = sigmoid(fc(Freq)). \qquad (13)$$

From Eqs. 12 and 13, we can see that our method generalizes the original method that only uses the GAP, i.e., the lowest frequency component to a framework with multiple frequency sources. By doing so, the inadequate problem of original methods is addressed. The overall illustration of our method is shown in Fig. 2.

**Criterion for choosing frequency components**   There exists an important problem of how to choose frequency component indices $[u, v]$ for each part $X^i$. For each channel with a spatial size of $H \times W$, we can get $HW$ frequency components after 2D DCT. In this case, the total number of combinations of these frequency components is $CHW$. For example, $C$ could equal to 2048 for ResNet-50 backbone. It is expensive to test all combinations. In this way, we propose a heuristic two-step criterion to choose the frequency components in the multi-spectral attention module.

The main idea is to first determine the importance of each frequency component and then determine the effects of using different numbers of frequency components together. First, we examine the results of each frequency component in channel attention individually. Then, we choose the Top-k highest performance frequency components based on the results. In this way, the multi-spectral channel attention can be fulfilled.

The ablation studies about this two-step criterion can be seen in Sec. 4.2.

### 3.3. Discussion

**How the multi-spectral framework embed more information**   In Sec. 3.2, we show that only using GAP in channel attention is actually discarding information of all other frequency components except the lowest one, i.e., GAP. In this way, generalizing channel attention in the frequency domain and using the multi-spectral framework could naturally embed more information in the channel attention mechanism.

Besides the above derivation, we also give a thought experiment to show that more information could be embedded. As we all know, deep networks are redundant [17, 41]. If two channels are redundant for each other, we can only get the same information using GAP. However, in our multi-spectral framework, it is possible to extract more information from redundant channels because different frequency components contain different information. In this way, the proposed multi-spectral framework could embed more information in the channel attention mechanism.

**Complexity analysis**   We analyze the complexity of our method from two aspects: the number of parameters and the computational cost.

For the number of parameters, our method has no extra parameters compared with the baseline SENet because the weights of 2D DCT are pre-computed constant.

For the computational cost, our method has a negligible extra cost and can be viewed as having the same computational cost as SENet. With ResNet-34, ResNet-50, ResNet-101, and ResNet-152 backbone, the relative computational cost increases of our method are 0.04%, 0.13%, 0.11%, and 0.11% compared with SENet, respectively. More results can be found in Table 2.

**One line change of code**   Another important property of the proposed multi-spectral framework is that it can be easily realized with existing channel attention implementations. As described in Sec. 3.1 and Eq. 11, 2D DCT can be viewed as a weighted sum of inputs. In this way, the implementation of our method can be simply achieved by element-wise multiplication and summation. The implementation is illustrated in Fig. 3.

```
1    import torch
2    from torch import nn
3
4    class SELayer(nn.Module):
5        def __init__(self, channel, reduction=16):
6            super(SELayer, self).__init__()
7            self.avg_pool = nn.AdaptiveAvgPool2d(1)
8
9            self.fc = nn.Sequential(
10               nn.Linear(channel, channel // reduction, bias=False),
11               nn.ReLU(inplace=True),
12               nn.Linear(channel // reduction, channel, bias=False),
13               nn.Sigmoid()
14           )
15
16       def forward(self, x):
17           n, c, _, _ = x.size()
18           y = self.avg_pool(x).view(n, c)
19           #shape: n,c,h,w --> n,c,1,1 --> n,c
20           y = self.fc(y).view(n, c, 1, 1)
21           return x * y.expand_as(x)
```

(a) Original SENet

```
1    import torch
2    from torch import nn
3    import get_dct_weights
4    class FcaLayer(nn.Module):
5        def __init__(self, channel, reduction=16):
6            super(FcaLayer, self).__init__()
7            self.register_buffer('pre_computed_dct_weights',
8                                 get_dct_weights(...))
9            self.fc = nn.Sequential(
10               nn.Linear(channel, channel // reduction, bias=False),
11               nn.ReLU(inplace=True),
12               nn.Linear(channel // reduction, channel, bias=False),
13               nn.Sigmoid()
14           )
15
16       def forward(self, x):
17           n, c, _, _ = x.size()
18           y = torch.sum(x*self.pre_computed_dct_weights, dim = [2,3])
19           #shape:n,c,h,w * 1,c,h,w --> n,c,h,w --> n,c
20           y = self.fc(y).view(n, c, 1, 1)
21           return x * y.expand_as(x)
```

(b) Our implementation

Figure 3. Implementation of our method and SENet. In the calculation, we only need to change one line of code to implement our method based on the existing code. The lines in red and green indicate the difference between SENet and our work. The get_dct_weights function is to implement Eq. 7 and the details can be found in the appendix.

As we can see, the only difference between the calculation of SENet and our method is the pre-processing part. For SENet, GAP is used while we use multi-spectral 2D DCT. In this way, our method could be easily integrated into arbitrary channel attention methods.

## 4. Experiments

In this section, we first elaborate on the details of our experiments. Then, we show ablation studies about FcaNet. Last, we investigate the effectiveness of our method on the task of image classification, object detection, and instance segmentation.

### 4.1. Implementation Details

To evaluate the results of the proposed FcaNet on ImageNet [28], we employ four widely used CNNs as backbone models, including ResNet-34, ResNet-50, ResNet-101, and ResNet-152. We follow the data augmentation and hyper-parameter settings in [15] and [16]. Concretely, the input images are cropped randomly to 224×224 with random horizontal flipping. We use an SGD optimizer with a momentum of 0.9, a weight decay of 1e-4, and a batch size of 128 per GPU at training time. For large models like ResNet-101 and ResNet-152, the batch size is set to 64. The learning rate is set to 0.1 for a batch size of 256 with the linear scaling rule [11]. All models are trained within 100 epochs with cosine learning rate decay. Notably, for training efficiency, we use the Nvidia APEX mixed precision training toolkit.

To evaluate our method on MS COCO [22] using Faster R-CNN [26] and Mask R-CNN [14]. We use the implementation of detectors from the MMDetection [4] toolkit and employ its default settings. During training, the shorter side of the input image is resized to 800. All models are optimized using SGD with a weight decay of 1e-4, a momentum of 0.9, and a batch size of 2 per GPU within 12 epochs.

The learning rate is initialized to 0.01 and is decreased by the factor of 10 at the 8th and 11th epochs, respectively.

All models are implemented in PyTorch [25] framework and with eight Nvidia RTX 2080Ti GPUs.

### 4.2. Ablation Study

As discussed in Sec. 3.2, it is expensive to verify all combinations of frequency components in our method. In this way, we propose the two-step criterion to select frequency components. In this section, we first show the results of using different components in channel attention individually. Then, we show the results of combinations with different numbers of Top-k settings.

**The effects of individual frequency components** To investigate the effects of different frequency components individually in channel attention, we only use one frequency component at a time. We divide the whole 2D DCT frequency space into $7 \times 7$ parts since the smallest feature map size is $7 \times 7$ on ImageNet. In this way, there are in total of 49 experiments. To speed up the experiments, we first train a standard ResNet-50 network for 100 epochs as the base model. Then we add channel attention to the base model with different frequency components to verify the effects. All added models are trained within 20 epochs with a similar optimization setting in Sec. 4.1, while the learning rate is set to 0.02.

As shown in Fig. 4, we can see that using lower frequency could have better performance, which is intuitive and verifies the success of SENet. This also verifies the conclusion [36] that deep networks prefer low-frequency information. Nevertheless, interestingly, we can see that nearly all frequency components (except the highest component) have very small gaps ($<= 0.5\%$ Top-1 accuracy) between the lowest one, i.e., vanilla channel attention with GAP. This

Low Frequency ⟶ High Frequency

| | | | | | | |
|---|---|---|---|---|---|---|
| 76.69 | 76.55 | 76.49 | 76.37 | 76.39 | 76.51 | 76.38 |
| 76.48 | 76.26 | 76.47 | 76.30 | 76.19 | 76.28 | 76.40 |
| 76.30 | 76.32 | 76.36 | 76.30 | 76.26 | 76.28 | 76.21 |
| 76.39 | 76.31 | 76.31 | 76.34 | 76.19 | 76.36 | 76.21 |
| 76.44 | 76.31 | 76.28 | 76.22 | 76.27 | 76.27 | 76.34 |
| 76.44 | 76.28 | 76.31 | 76.33 | 76.31 | 76.33 | 76.27 |
| 76.53 | 76.32 | 76.28 | 76.34 | 76.28 | 76.30 | 75.72 |

(Left axis: Low Frequency ↓ High Frequency)

Figure 4. Top-1 accuracies on ImageNet using different frequency components in channel attention individually.

shows that other frequency components can also cope well with the channel attention mechanism, and it is effective to generalize the channel attention in the frequency domain.

**The effects of different numbers of frequency components** After obtaining the performance of each frequency component, the second step is to determine the number of components that should be used in multi-spectral channel attention. For simplicity, we select Top-k highest performance frequency components, where k could be 1, 2, 4, 8, 16, or 32.

Table 1. The results of using different numbers of frequency components on ImageNet with ResNet-50 backbone.

| Number | Top-1 acc | Top-5 acc |
|---|---|---|
| 1 | 77.26 | 93.55 |
| 2 | 78.40 | 94.07 |
| 4 | 78.38 | 94.03 |
| 8 | 78.33 | 94.06 |
| 16 | **78.52** | **94.14** |
| 32 | 78.42 | 94.10 |

As shown in Table 1, we can see two phenomena. 1) All experiments with multi-spectral attention have a significant performance gap compared with the one only using the GAP in channel attention. This verifies our idea of using multiple frequency components in channel attention. 2) The setting with 16 frequency components gains the best performance. In this way, we use the Top-16 highest performance frequency components in our method and all other experiments[2].

---

[2] Some other kinds of combinations can be found in the appendix.

### 4.3. Image Classification on ImageNet

We compare our FcaNet with the state-of-the-art methods using ResNet-34, ResNet-50, ResNet-101, and ResNet-152 backbones on ImageNet, including SENet [19], CBAM [34], GSoP-Net1 [10], GCNet [3], AANet [2], and ECANet [31]. The evaluation metrics include both efficiency (i.e., network parameters, floating point operations per second (FLOPs)) and effectiveness (i.e., Top-1/Top-5 accuracy).

As shown in Table 2, our method achieves the best performance in all experimental settings. Specifically, with the same number of parameters and computational cost, our method outperforms SENet by a large margin. FcaNet outperforms SENet by 1.20%, 1.81%, 2.02%, and 1.65% in terms of Top-1 accuracy under different backbones. Note that FcaNet could also outperform GSoPNet, which has a significantly higher computational cost than our method. This shows the effectiveness of our method.

### 4.4. Object Detection on MS COCO

Besides the classification task on ImageNet, we also evaluate our method on object detection task to verify its effectiveness and generalization ability. We use our FcaNet with FPN [21] as the backbone (ResNet-50 and ResNet-101) of Faster R-CNN and Mask R-CNN and test their performance on the MS COCO dataset. SENet, CBAM, GCNet, and ECANet are used for comparison.

As shown in Table 3, our method could also achieve the best performance with both Faster-RCNN and Mask-RCNN framework. Identical to the classification task on ImageNet, FcaNet could also outperform SENet by a large margin with the same number of parameters and computational cost. Compared with the SOTA method ECANet, FcaNet could outperform it by 0.9-1.3% in terms of AP.

### 4.5. Instance Segmentation on MS COCO

Besides the object detection, we then test our method on the instance segmentation task. As shown in Table 4, our method outperforms other methods by a more considerable margin. Specifically, FcaNet outperforms GCNet by 0.5% AP, while the gaps between other methods are roughly 0.1-0.2%. These results verify the effectiveness of our method.

## 5. Conclusion

In this paper, we have proven that GAP is a special case of DCT and proposed the FcaNet with the multi-spectral attention module, which generalizes the existing channel attention mechanism in the frequency domain. Meanwhile, we have explored different combinations of frequency components in our multi-spectral framework and proposed a two-step criterion for frequency components selection. With the same number of parameters and com-

Table 2. Comparison of different attention methods on ImageNet. All other results are quoted from their original papers if available.

| Method | Years | Backbone | Parameters | FLOPS | Top-1 acc | Top-5 acc |
|---|---|---|---|---|---|---|
| ResNet [15] | CVPR16 | | 21.80 M | 3.68 G | 73.31 | 91.40 |
| SENet [19] | CVPR18 | ResNet-34 | 21.95 M | 3.68 G | 73.87 | 91.65 |
| ECANet [31] | CVPR20 | | 21.80 M | 3.68 G | 74.21 | 91.83 |
| FcaNet (ours) | | | 21.95 M | 3.68 G | **75.07** | **92.16** |
| ResNet [15] | CVPR16 | | 25.56 M | 4.12 G | 75.20 | 92.52 |
| SENet [19] | CVPR18 | | 28.07 M | 4.13 G | 76.71 | 93.38 |
| CBAM [34] | ECCV18 | | 28.07 M | 4.14 G | 77.34 | 93.69 |
| GSoPNet1 [10] | CVPR19 | ResNet-50 | 28.29 M | 6.41 G | 77.98 | 94.12 |
| GCNet [3] | ICCVW19 | | 28.11 M | 4.13 G | 77.70 | 93.66 |
| AANet [2] | ICCV19 | | 25.80 M | 4.15 G | 77.70 | 93.80 |
| ECANet [31] | CVPR20 | | 25.56 M | 4.13 G | 77.48 | 93.68 |
| FcaNet (ours) | | | 28.07 M | 4.13 G | **78.52** | **94.14** |
| ResNet [15] | CVPR16 | | 44.55 M | 7.85 G | 76.83 | 93.48 |
| SENet [19] | CVPR18 | | 49.29 M | 7.86 G | 77.62 | 93.93 |
| CBAM [34] | ECCV18 | | 49.30 M | 7.88 G | 78.49 | 94.31 |
| AANet [2] | ICCV19 | ResNet-101 | 45.40 M | 8.05 G | 78.70 | 94.40 |
| ECANet [31] | CVPR20 | | 44.55 M | 7.86 G | 78.65 | 94.34 |
| FcaNet (ours) | | | 49.29 M | 7.86 G | **79.64** | **94.63** |
| ResNet [15] | CVPR16 | | 60.19 M | 11.58 G | 77.58 | 93.66 |
| SENet [19] | CVPR18 | | 66.77 M | 11.60 G | 78.43 | 94.27 |
| AANet [2] | ICCV19 | ResNet-152 | 61.60 M | 11.90 G | 79.10 | 94.60 |
| ECANet [31] | CVPR20 | | 60.19 M | 11.59 G | 78.92 | 94.55 |
| FcaNet (ours) | | | 66.77 M | 11.60 G | **80.08** | **94.88** |

Table 3. Object detection results of different methods on COCO val 2017.

| Method | Detector | Parameters | FLOPs | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|
| ResNet-50 | | 41.53 M | 215.51 G | 36.4 | 58.2 | 39.2 | 21.8 | 40.0 | 46.2 |
| SENet | | 44.02 M | 215.63 G | 37.7 | 60.1 | 40.9 | 22.9 | 41.9 | 48.2 |
| ECANet | | 41.53 M | 215.63 G | 38.0 | 60.6 | 40.9 | 23.4 | 42.1 | 48.0 |
| FcaNet (Ours) | Faster-RCNN | 44.02 M | 215.63 G | **39.0** | **61.1** | **42.3** | **23.7** | **42.8** | **49.6** |
| ResNet-101 | | 60.52 M | 295.39 G | 38.7 | 60.6 | 41.9 | 22.7 | 43.2 | 50.4 |
| SENet | | 65.24 M | 295.58 G | 39.6 | 62.0 | 43.1 | 23.7 | 44.0 | 51.4 |
| ECANet | | 60.52 M | 295.58 G | 40.3 | 62.9 | 44.0 | **24.5** | 44.7 | 51.3 |
| FcaNet (Ours) | | 65.24 M | 295.58 G | **41.2** | **63.3** | **44.6** | 23.8 | **45.2** | **53.1** |
| ResNet-50 | | 44.17 M | 261.81 G | 37.2 | 58.9 | 40.3 | 22.2 | 40.7 | 48.0 |
| SENet | | 46.66 M | 261.93 G | 38.7 | 60.9 | 42.1 | 23.4 | 42.7 | 50.0 |
| GCNet | Mask-RCNN | 46.69 M | 261.94 G | 39.4 | 61.6 | 42.4 | N/A | N/A | N/A |
| ECANet | | 44.17 M | 261.93 G | 39.0 | 61.3 | 42.1 | 24.2 | 42.8 | 49.9 |
| FcaNet (Ours) | | 46.66 M | 261.93 G | **40.3** | **62.0** | **44.1** | **25.2** | **43.9** | **52.0** |

Table 4. Instance segmentation results of different methods using Mask R-CNN on COCO val 2017.

| Method | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|
| ResNet-50 | 34.1 | 55.5 | 36.2 |
| SENet | 35.4 | 57.4 | 37.8 |
| GCNet | 35.7 | 58.4 | 37.6 |
| ECANet | 35.6 | 58.1 | 37.7 |
| FcaNet (Ours) | **36.2** | **58.6** | **38.1** |

putational cost, our method could consistently outperform SENet by a large margin. We also have achieved state-of-the-art performance on image classification, object detection, and instance segmentation compared with other channel attention methods. Moreover, FcaNet is simple yet effective. Our method could be implemented with only one line change of code based on existing channel attention methods.

# 6. Appendix

## 6.1. Investigation of More Frequency Combinations

This section shows more results about using different frequency combinations in the proposed multi-spectral channel attention module.

In Sec. 4.2, we present a two-step method to select the best frequency component combinations in the proposed multi-spectral channel attention mechanism. Besides the proposed combinations, we also try some other possibilities of combinations, as shown in Fig. 5.

The first one is an intuitive method, termed as Low-k (Lowest-k), as shown in Fig. 5(b). Low-k selects the lowest k frequency components (left upper triangle part of the 2D frequency spectrum) as the combinations. It only considers the frequency of the components and has no relation to the performance of the individual component in Fig. 4.

The second one is the counterpart of our two-step Top-k method, termed as Bot-k (Bottom-k), as shown in Fig. 5(c). Bot-k selects the k frequency components with the lowest performance, which is exactly the opposite of the Top-k method.
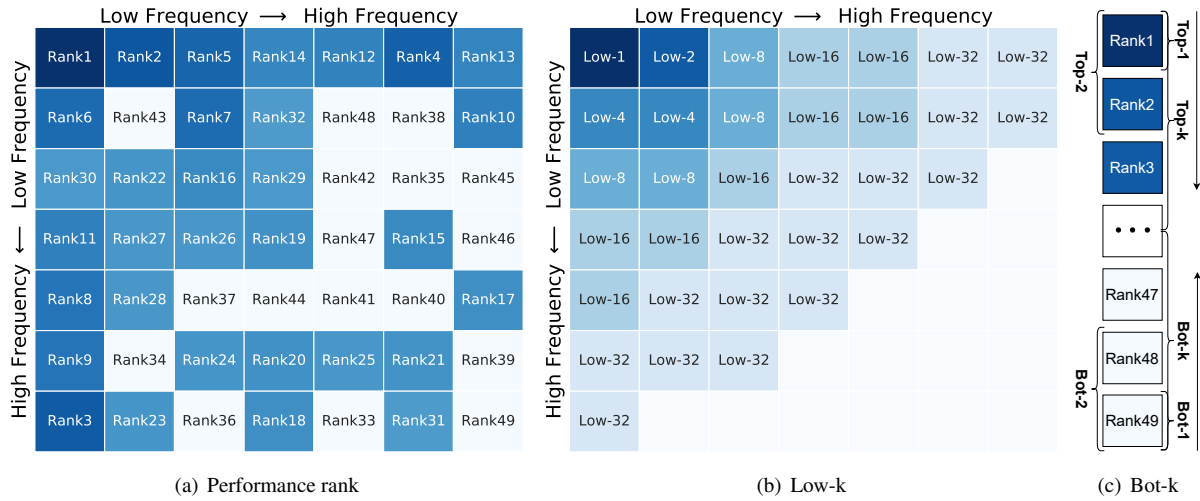


Figure 5. Illustration of different frequency combinations. (a) shows the performance rank as Fig. 4. (b) shows the selection method of Low-k, which only considers the frequency itself. (c) shows the Top-k and Bot-k, which consider the performance rank of each frequency component.

| Table 5. The results of Low-k combinations. | | |
|---|---|---|
| Number | Top-1 acc | Top-5 acc |
| 1 | 77.26 | 93.55 |
| 2 | 78.40 | 94.07 |
| 4 | 78.27 | 94.06 |
| 8 | 78.25 | 94.01 |
| 16 | 78.26 | 94.12 |
| 32 | 78.37 | 94.08 |

| Table 6. The results of Bot-k combinations. | | |
|---|---|---|
| Number | Top-1 acc | Top-5 acc |
| 1 | 77.09 | 93.50 |
| 2 | 77.30 | 93.51 |
| 4 | 77.27 | 93.57 |
| 8 | 77.51 | 93.63 |
| 16 | 77.17 | 93.53 |
| 32 | 77.60 | 93.78 |

The highest performance of our Top-k method is 78.52%. Compared with Bot-k in Table 6, the results show that low-frequency components are important. Compared with Low-k in Table 5, the Top-k method also performs better. This shows that we should take the performance of individual frequency component into consideration and demonstrates the effectiveness of our two-step criterion.

## 6.2. Visualization of Discrete Cosine Transform

In this section, we show some visualization results related to the discrete cosine transform (DCT).

In Fig. 6(a), we show the image of the basis functions of 2D DCT. We can see that 2D DCT basis functions are composed of regular horizontal and vertical cosine waves. These basis functions are orthogonal and data-independent.

9

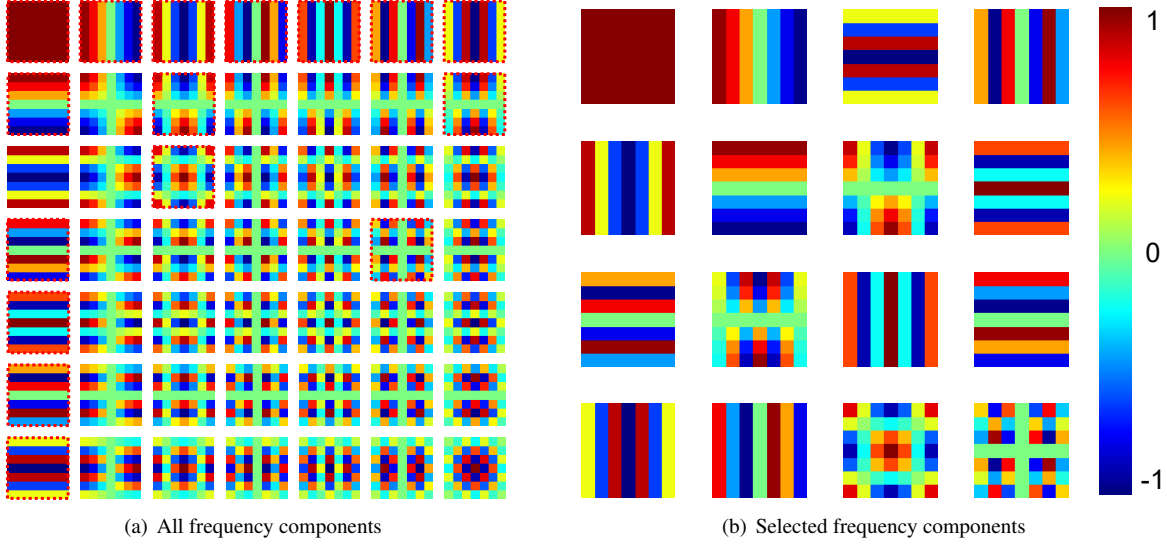(a) All frequency components        (b) Selected frequency components

Figure 6. The visualization about DCT basis functions. (a) shows the visualizations of all frequency components. (b) shows the selected frequency components using our two-step Top-k criterion. The selected components are also highlighted in (a) with red dashed box.

In Fig. 6(b), we show the selected frequency components using our two-step criterion. We can see that the selected frequency components are usually low-frequency.

## 6.3. Analysis of Channel Attention

In this section, we give a more detailed mathematically analysis of channel attention based on Sec. 3.2. According to Sec. 3.2, we have:

$$x_{i,j}^{2d} = gap(x^{2d})HWB_{0,0}^{i,j} + f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}$$

$$s.t. \ \ i \in \{0, 1, \cdots, H-1\}, j \in \{0, 1, \cdots, W-1\}, \tag{14}$$

and

$$X = \underbrace{gap(X)HWB_{0,0}^{i,j}}_{\text{utilized}} + \underbrace{f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}}_{\text{discarded}}. \tag{15}$$

subsequently, we can give a more detailed derivation.

$$
X_{i,:,:} = \begin{bmatrix} x_{0,0}^{2d} & \cdots & x_{0,W-1}^{2d} \\ \vdots & \ddots & \\ x_{H-1,0}^{2d} & & x_{H-1,W-1}^{2d} \end{bmatrix}
$$

$$
= \begin{bmatrix} \underbrace{\mathbb{G}B_{0,0}^{0,0}}_{\text{utilized}} + \underbrace{\mathbb{D}^{0,0}}_{\text{discarded}} & \cdots & \underbrace{\mathbb{G}B_{0,0}^{0,v}}_{\text{utilized}} + \underbrace{\mathbb{D}^{0,v}}_{\text{discarded}} \\ \vdots & \ddots & \\ \underbrace{\mathbb{G}B_{0,0}^{u,0}}_{\text{utilized}} + \underbrace{\mathbb{D}^{u,0}}_{\text{discarded}} & & \underbrace{\mathbb{G}B_{0,0}^{u,v}}_{\text{utilized}} + \underbrace{\mathbb{D}^{u,v}}_{\text{discarded}} \end{bmatrix}, \tag{16}
$$

in which $X_{i,:,:}$ is the i-th channel of feature, $\mathbb{G} = gap(X)HW$, $u = H-1$, $v = W-1$, and $\mathbb{D}^{i,j} = f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}$.

We can see that the conventional channel attention is actually discarding information from all other frequency components except the lowest one. Note that this derivation is in the matrix form.

## 6.4. Initialization of DCT weights

In this section, we give the details of the initialization, i.e., the get_dct_weights function in Fig. 3. The get_dct_weights function is shown in Fig. 7. It should be noted that the get_dct_weights function is only for the initialization, so the one-line change holds for training and inference.

```python
import torch,math

def get_1d_dct(i, freq, L):
    result = math.cos(math.pi * freq * (i + 0.5) / L) / math.sqrt(L)
    if freq == 0:
        return result
    else:
        return result * math.sqrt(2)

def get_dct_weights( width, height, channel, fidx_u, fidx_v):
    # width    : width of input
    # height   : height of input
    # channel : channel of input
    # fidx_u   : horizontal indices of selected fequency
    # fidx_v   : vertical indices of selected fequency

    dct_weights = torch.zeros(1, channel, width, height)

    c_part = channel // len(fidx_u)
    # split channel for multi-spectal attention

    for i, (u_x, v_y) in enumerate(zip(fidx_u, fidx_v)):
        for t_x in range(width):
            for t_y in range(height):
                dct_weights[:, i * c_part: (i+1)*c_part, t_x, t_y] =
                get_1d_dct(t_x, u_x, width) * get_1d_dct(t_y, v_y, height)
    # Eq. 7 in our paper

    return dct_weights
```

Figure 7. The details of the get_dct_weights function. This code is only for the initialization of the dct weights. It will run only at the very beginning, and will not participate in the training and testing.

# References

[1] Nasir Ahmed, T‿ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE Trans. Comput.*, pages 90–93, 1974. 3

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Int. Conf. Comput. Vis.*, pages 3286–3295, 2019. 2, 7, 8

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *IEEE Conf. Comput. Vis. Worksh.*, pages 0–0, 2019. 2, 7, 8

[4] 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. 6

[5] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Q Weinberger, and Yixin Chen. Compressing convolutional neural networks in the frequency domain. In *SIGKDD*, pages 1475–1484, 2016. 2

[6] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. In *Adv. Neural Inform. Process. Syst.*, pages 352–361, 2018. 2

[7] Max Ehrlich and Larry S Davis. Deep residual learning in the jpeg transform domain. In *Int. Conf. Comput. Vis.*, pages 3484–3493, 2019. 2

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154, 2019. 2

[9] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10705–10714, 2019. 2

[10] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3024–3033, 2019. 2, 7, 8

[11] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[12] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. In *Adv. Neural Inform. Process. Syst.*, pages 3933–3944, 2018. 2

[13] Hao Guo, Kang Zheng, Xiaochuan Fan, Hongkai Yu, and Song Wang. Visual attention consistency under image transforms for multi-label image classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 729–739, 2019. 2

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Int. Conf. Comput. Vis.*, pages 2961–2969, 2017. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 6, 8

[16] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 558–567, 2019. 6

[17] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Int. Conf. Comput. Vis.*, pages 1389–1397, 2017. 5

[18] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, pages 9401–9411, 2018. 2

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7132–7141, 2018. 1, 2, 3, 7, 8

[20] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 510–519, 2019. 2

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2117–2125, 2017. 7

[22] 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 *Eur. Conf. Comput. Vis.*, pages 740–755, 2014. 6

[23] Zhenhua Liu, Jizheng Xu, Xiulian Peng, and Ruiqin Xiong. Frequency-domain dynamic pruning for convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, pages 1043–1053, 2018. 2

[24] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018. 2

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*, pages 8026–8037, 2019. 6

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*, pages 91–99, 2015. 6

[27] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *IEEE Medical Imaging*, pages 540–549, 2018. 2

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, pages 211–252, 2015. 6

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 1

[30] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3156–3164, 2017. 2

[31] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11534–11542, 2020. 1, 2, 3, 7, 8

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7794–7803, 2018. 2

[33] Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. Cnnpack: Packing convolutional neural networks in the frequency domain. In *Adv. Neural Inform. Process. Syst.*, pages 253–261, 2016. 2

[34] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018. 2, 7, 8

[35] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Int. Conf. Mach. Learn.*, pages 2048–2057, 2015. 1, 2

[36] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1740–1749, 2020. 2, 6

[37] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 2

[38] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Eur. Conf. Comput. Vis.*, pages 286–301, 2018. 2

[39] Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3085–3094, 2019. 2

[40] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Int. Conf. Comput. Vis.*, pages 6688–6697, 2019. 2

[41] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In *Adv. Neural Inform. Process. Syst.*, pages 875–886, 2018. 5