FcaNet: Frequency Channel Attention Networks

Zequn Qin¹, Pengyi Zhang¹, Fei Wu¹,², Xi Li¹,²*
¹College of Computer Science, Zhejiang University,
²Shanghai 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., channel attention mechanism uses scalar to represent channel, which is difficult due to massive information loss. In this work, we start from a different view and regard the channel representation problem as a compression process using frequency analysis. Based on the frequency analysis, we mathematically prove that the conventional global average pooling is a special case of the feature decomposition in the frequency domain. With the proof, we naturally generalize the compression of the channel attention mechanism in the frequency domain and propose our method with multi-spectral channel attention, termed as FcaNet. FcaNet is simple but effective. We can change a few lines 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 consistently outperform the baseline SENet, with the same number of parameters and computational cost. Our code and models are publicly available at https://github.com/cfzd/FcaNet.

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 [40] and natural language processing [34]. 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 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.

Typically, a core step of channel attention approaches is to use 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. Nevertheless, every rose has its thorn. The simplicity of GAP makes it hard to well capture complex information for various inputs. Some methods like CBAM [39] and SRM [23] further use global max pooling and global standard deviation pooling to enhance the performance of GAP. Different from previous works, we consider the scalar representation of a channel as a compression problem. Namely, the information of a channel should be compactly encoded by a scalar while preserving the representation ability of the whole channel as much as possible. In this way, how to effectively compress a channel with a
scalar due to the constrained computational overhead is a major difficulty, and it is crucial to channel attention.

With the above motivation, we propose to use the discrete cosine transform (DCT) to compress channels in the channel attention mechanism for the following reasons: 1) DCT is a widely used data compression method in signal processing, especially with digital images and videos. Many widely used image and video formats like JPEG, HEIF, MPEG, and H.26x use DCT to realize data compression. DCT has a strong energy compaction property [1, 30], so it could achieve high data compression ratio with high quality [2, 22]. This property meets the demand of the channel attention that representing a channel with a scalar. 2) DCT can be implemented with an element-wise multiplication, and it is differentiable. In this way, it can be easily integrated into CNNs. 3) Surprisingly, DCT can be viewed as a generalization of GAP. Mathematically, GAP (showing the effectiveness in SENet [21]) is only equivalent to the lowest frequency components of DCT, leaving many other potentially useful frequency components unexplored. This strongly motivates us to tailor DCT for the channel attention mechanism.

In this paper, we further propose a simple, novel, but effective multi-spectral channel attention (MSCA) framework based on the above discussions. In order to better compress channels and explore the components left out by GAP, we propose to tailor DCT and use multiple but limited frequency components of DCT for the channel attention mechanism. Note that although we use multi-spectral channel attention, each channel is still represented by only one scalar. Along with the MSCA framework, how to select the frequency component of DCT for each channel is important. In this way, we propose three kinds of frequency component selection criteria to fulfill and validate the MSCA framework, which are LF (Low Frequency based selection), TS (Two-Step selection), and NAS (Neural Architecture Search selection). With these selection criteria, our method 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 regard the channel attention as a compression problem and introduce DCT in the channel attention. We then prove that conventional GAP is a special case of DCT. Based on this proof, we generalize the channel attention in the frequency domain and propose our method with the multi-spectral channel attention framework, termed as FcaNet.
- We propose three kinds of frequency component selection criteria along with the proposed multi-spectral channel attention framework to fulfill FcaNet.
- Extensive experiments demonstrate the proposed method achieves state-of-the-art results on both ImageNet and COCO datasets, with the same computational cost as SENet. The results on ImageNet are shown in Fig. 1.

2. Related Work

Attention Mechanism in CNNs In [40], 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 [35] is proposed with a spatial attention mechanism using downsampling and upsampling. Besides, SENet [21] 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 [20] uses spatial attention to better exploit the feature context, and $A^2$-Net [7] builds a relation function for image or video recognition.

Inspired by these works, a series of works like BAM [28], DAN [9], CBAM [39], scSE [32], and CoordAttention [19] are proposed to fuse spatial attention [45] 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. Similarly, SRM [23] also propose to use GAP with global standard deviation pooling. Motivated by CBAM, GSoP [11] introduces a second-order pooling method for downsampling. NonLocal [37] proposes to build a dense spatial feature map. AANet [3] proposes to embed the attention map with position information into the feature. SkNet [24] introduces a selective channel aggregation and attention mechanism, and ResNeSt [42] proposes a similar split attention method. Due to the complicated attention operation, these methods are relatively large. To improve efficiency, GCNet [4] proposes to use a simple spatial attention module and replace the original spatial downsampling process. ECANet [36] 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 [14], saliency detection [44], visual explanation [10], and super-resolution [43].

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 [8, 13], frequency analysis is introduced in the CNNs by JPEG encoding. Then, DCT is incorporated in [41] to reduce communication bandwidth. There are also some applications in the model compression and pruning tasks like [6, 27, 38].
3. Method

In this section, we first revisit the formulation of DCT and channel attention. Then, based on these works, we elaborate on the derivation of our multi-spectral channel attention framework. Meanwhile, along with the multi-spectral channel attention framework, three kinds of frequency components selection methods are proposed.

3.1. Revisiting DCT and Channel Attention

Discrete Cosine Transform (DCT) Typically, the basis function of two-dimensional (2D) DCT [1] is:

\[ B^{i,j}_{h,w} = \cos\left(\frac{\pi h}{H}(i + \frac{1}{2})\right) \cos\left(\frac{\pi w}{W}(j + \frac{1}{2})\right). \]

(1)

Then the 2D DCT can be written as:

\[ f_{2d}^{i,j}_{h,w} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} B_{h,w}^{i,j} \]

s.t. \( h \in \{0, 1, \cdots, H - 1\}, w \in \{0, 1, \cdots, W - 1\}, \)

(2)

in which \( f_{2d}^{i,j}_{h,w} \in \mathbb{R}^{H \times W} \) is the 2D DCT frequency spectrum, \( x_{i,j}^{2d} \in \mathbb{R}^{H \times W} \) 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:

\[ x_{i,j}^{2d} = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} f_{2d}^{i,j}_{h,w} B_{h,w}^{i,j}; \]

(3)

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

Please note that in Eqs. 2 and 3, some constant normalization factors are removed for simplicity, which will not affect the results in this work.

Channel Attention The channel attention mechanism is widely used in CNNs. It uses scalar to represent and evaluate the importance of each channel. 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. As discussed in Sec. 1, we treat the scalar representation in channel attention as a compression problem since it has to represent the whole channel while only one scalar can be used. In this way, the attention mechanism can be written as:

\[ att = \text{sigmoid}(fc(\text{compress}(X))), \]

where \( att \in \mathbb{R}^{C} \) is the attention vector, \( \text{sigmoid} \) is the Sigmoid function, \( fc \) represents the mapping functions like fully connected layer or one-dimensional convolution, and \( \text{compress} : \mathbb{R}^{C \times H \times W} \rightarrow \mathbb{R}^{C} \) is a compression method.

After obtaining the attention vector of all \( C \) channels, each channel of input \( X \) is scaled by the corresponding attention value:

\[ \widetilde{X}_{i, : , :} = att_{i} X_{i, : , :}, \quad \text{s.t.} \quad i \in \{0, 1, \cdots, C - 1\}, \]

(5)

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.

Typically, global average pooling is the de-facto compression method [21, 36] for its simplicity and effectiveness. There are also compression methods like global max pooling [39] and global standard deviation pooling [23].

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 prove 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. 2 are 0, we have:

\[ f_{0,0}^{2d} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} \cos\left(\frac{0}{H}(i + \frac{1}{2})\right) \cos\left(\frac{0}{W}(j + \frac{1}{2})\right) \]

\[ = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} \]

\[ = \text{gap}(x^{2d}) H W. \]

(6)

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.

Multi-Spectral Channel Attention Module Based on the theoretical analysis and Theorem 1, we can see that using GAP in the channel attention mechanism means only the lowest frequency information is preserved. And all components from other frequencies are discarded, which also encode the useful information patterns in representing the channels and should not be left out.

To better compress channels and introduce more information, we propose to generalize GAP to more frequency

...
components of 2D DCT and compress more information with 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 $[X^0, X^1, \ldots, X^{n-1}]$ as the parts, in which $X^i \in \mathbb{R}^{C' \times H \times W}$, $i \in \{0, 1, \ldots, n-1\}$, $C' = \frac{C}{n}$, and $C$ should 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 compression results of channel attention. In this way, we have:

$$Freq^i = 2DDCT^{u_i,v_i}(X^i),$$

$$= \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} X^i_{h,w} B_{h,w}^{u_i,v_i}$$

$$s.t. \quad i \in \{0, 1, \ldots, n-1\},$$

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

$$Freq = \text{compress}(X) = \text{cat}(\{Freq^0, Freq^1, \ldots, Freq^{n-1}\}),$$

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

$$ms_{att} = \text{sigmoid}(fc(Freq)).$$

From Eqs. 8 and 9, we can see that our method generalizes the original GAP method to a framework with multiple frequency components. By doing so, the channel information after compression is effectively enriched for representation. The overall illustration of our method is shown in Fig. 2.

Criteria for Choosing Frequency Components  There exists an important problem of how to choose frequency
component indices \([u_i, v_i]\) for each part \(X^i\). In order to fulfill the multi-spectral channel attention, we propose three kinds of criteria, which are FcaNet-LF (Low Frequency), FcaNet-TS (Two-Step selection), and FcaNet-NAS (Neural Architecture Search).

FcaNet-LF means FcaNet with low-frequency components. As we all know, many compression methods use low-frequency information of DCT to compress information. Moreover, some methods [21, 41] have shown CNNs prefer low-frequency information. In this way, the first criterion for choosing frequency components is to only select low-frequency components.

FcaNet-TS means FcaNet selects components within a two-step selection scheme. Its main idea is to first determine the importance of each frequency component and then investigate the effects of using different numbers of frequency components. Namely, we evaluate the results of each frequency component in channel attention individually. Finally, we choose the Top-k highest performance frequency components based on the evaluation results.

FcaNet-NAS means FcaNet with searched components. For this criterion, we use neural architecture search to search the best frequency component for channels. For each part \(X^i\), a set of continuous variables \(\alpha = \{\alpha(u,v)\}\) are assigned to search components. The frequency components of this part can be written as:

\[
Freq_{nas}^i = \sum_{(u,v) \in O} \sum_{(u',v') \in O} \exp(\alpha(u,v)) \cdot \exp(\alpha(u',v')) \cdot 2DDCT_{u,v}(X^i),
\]

in which \(O\) is the set containing all 2D DCT frequency component indices. After training, the frequency component for \(X^i\) is derived by \((u_i^*, v_i^*) = \arg \max_{(u,v) \in O} \alpha(u,v))\).

The ablation studies about these criteria can be seen in Sec. 4.2.

4. Experiments

In this section, we first elaborate on the details of our experiments. Second, we show the ablation studies about FcaNet. Third, we give discussions about how the information is compressed in our framework, complexity, and code implementation. At 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 [35], 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 hyperparameter settings in [16] and [17]. Concretely, the input images are cropped randomly to \(224 \times 224\) 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 [12]. All models are trained within 100 epochs with cosine learning rate decay and label smoothing. Notably, for training efficiency, we use the Nvidia APEX mixed precision training toolkit.

To evaluate our method on MS COCO [26] using Faster R-CNN [31] and Mask R-CNN [15], we use the implementation of detectors from the MMDetection [5] 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 [29] framework and with eight Nvidia RTX 2080Ti GPUs.

4.2. Ablation Study

As discussed in Sec. 3.2, we propose three kinds of criteria, including FcaNet-LF (Low Frequency), FcaNet-TS (Two-Step selection), and FcaNet-NAS (Neural Architecture Search). In this section, we first show the ablations about these variants. Then we discuss the relation between FcaNet and fully learnable channel attention.

The effects of individual frequency components For FcaNet-TS, the first step is to determine the importance of each frequency component. 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. 3, 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 [41] 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 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.
Figure 3. Top-1 accuracies on ImageNet using different frequency components in channel attention individually.

The effects of different numbers of frequency components For FcaNet-LF, we verify the results of using K lowest-frequency components. For FcaNet-TS, we select Top-K highest performance frequency components in Fig. 3. For simplicity, K could be 1, 2, 4, 8, 16, or 32.

As shown in Fig. 4, we can see two phenomena. 1) All experiments with multi-spectral attention have a significant performance gain compared with the one only using the GAP in channel attention. This verifies our idea of using multiple frequency components in channel attention. 2) For FcaNet-LF and FcaNet-TS, the settings with 2 and 16 frequency components gain the best performance, respectively. In this way, we use these settings in our method and all other experiments.

Comparison with fully learnable channel attention As shown in Eq. 7, we use the 2D DCT basis functions $B^{(k)}_{u,v}$ can be simply regarded as a tensor containing DCT coefficients. In this way, a natural question is that how about directly learning a tensor to compress channels. We compare our method with three different kinds of tensors, which are Fixed tensor with Random initialization (FR), Learned tensor with Random initialization (LR), and Learned tensor with DCT initialization (LD). In this case, our method can be viewed as a Fixed tensor with DCT initialization (FD).

Figure 5. Comparison with fully learnable channel attention. FR means a Fixed tensor with Random initialization, LR means a Learned tensor with Random initialization, LD means a Learned tensor with DCT initialization, and FD means a Fixed tensor with DCT initialization, which is our method. For settings with the random initialization, the error bars are shown.

The comparison is shown in Fig. 5. We can see that all settings with the DCT initialization (LD, FD) outperform the ones without DCT (FR, LR). Moreover, the setting with fixed DCT initialization (ours) even outperforms the fully learnable channel attention methods (LR, LD), which shows the effectiveness of using DCT to compress channels.

4.3. Discussion

How the multi-spectral framework compresses and embeds 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 [18, 46]. 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
Table 1. Comparison of different attention methods on ImageNet. All results are reproduced and trained with the same training setting except AANet, which has no official code.

<table>
<thead>
<tr>
<th>Method</th>
<th>Years</th>
<th>Backbone</th>
<th>Parameters</th>
<th>FLOPS</th>
<th>Train FPS</th>
<th>Test FPS</th>
<th>Top-1 acc</th>
<th>Top-5 acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet [16]</td>
<td>CVPR16</td>
<td>ResNet-34</td>
<td>21.80 M</td>
<td>3.68 G</td>
<td>2898</td>
<td>3840</td>
<td>74.58</td>
<td>92.05</td>
</tr>
<tr>
<td>ECANet [36]</td>
<td>CVPR20</td>
<td>ResNet-101</td>
<td>21.80 M</td>
<td>3.68 G</td>
<td>2703</td>
<td>3682</td>
<td>74.65</td>
<td>92.21</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td></td>
<td></td>
<td>21.95 M</td>
<td>3.68 G</td>
<td>2717</td>
<td>3356</td>
<td>74.95</td>
<td>92.16</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td></td>
<td></td>
<td>21.95 M</td>
<td>3.68 G</td>
<td>2717</td>
<td>3356</td>
<td>75.02</td>
<td>92.07</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td></td>
<td></td>
<td>21.95 M</td>
<td>3.68 G</td>
<td>2717</td>
<td>3356</td>
<td>74.97</td>
<td>92.34</td>
</tr>
<tr>
<td>ResNet [16]</td>
<td>CVPR16</td>
<td>ResNet-50</td>
<td>25.56 M</td>
<td>4.12 G</td>
<td>1644</td>
<td>3622</td>
<td>77.27</td>
<td>93.52</td>
</tr>
<tr>
<td>SENet [21]</td>
<td>CVPR18</td>
<td>ResNet-50</td>
<td>28.07 M</td>
<td>4.13 G</td>
<td>1457</td>
<td>3417</td>
<td>77.86</td>
<td>93.87</td>
</tr>
<tr>
<td>ECANet [36]</td>
<td>CVPR20</td>
<td>ECANet ResNet-50</td>
<td>25.56 M</td>
<td>4.13 G</td>
<td>1430</td>
<td>3331</td>
<td>78.43</td>
<td>94.15</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td></td>
<td></td>
<td>28.07 M</td>
<td>4.13 G</td>
<td>1430</td>
<td>3331</td>
<td>78.57</td>
<td>94.10</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td></td>
<td></td>
<td>28.07 M</td>
<td>4.13 G</td>
<td>1430</td>
<td>3331</td>
<td>78.46</td>
<td>94.09</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td></td>
<td></td>
<td>28.07 M</td>
<td>4.13 G</td>
<td>1430</td>
<td>3331</td>
<td>79.06</td>
<td>94.06</td>
</tr>
<tr>
<td>ResNet [16]</td>
<td>CVPR16</td>
<td>ResNet-50</td>
<td>44.55 M</td>
<td>7.85 G</td>
<td>816</td>
<td>3187</td>
<td>78.72</td>
<td>94.30</td>
</tr>
<tr>
<td>AANet [3]</td>
<td>ICCV19</td>
<td>AANet ResNet-50</td>
<td>45.40 M</td>
<td>8.05 G</td>
<td>-</td>
<td>-</td>
<td>78.70</td>
<td>94.40</td>
</tr>
<tr>
<td>ECANet [36]</td>
<td>CVPR20</td>
<td>ECANet ResNet-50</td>
<td>44.55 M</td>
<td>7.86 G</td>
<td>721</td>
<td>3000</td>
<td>79.09</td>
<td>94.38</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td></td>
<td></td>
<td>49.29 M</td>
<td>7.86 G</td>
<td>705</td>
<td>2936</td>
<td>79.46</td>
<td>94.60</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td></td>
<td></td>
<td>49.29 M</td>
<td>7.86 G</td>
<td>705</td>
<td>2936</td>
<td>79.63</td>
<td>94.63</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td></td>
<td></td>
<td>49.29 M</td>
<td>7.86 G</td>
<td>705</td>
<td>2936</td>
<td>79.53</td>
<td>94.64</td>
</tr>
<tr>
<td>ResNet [16]</td>
<td>CVPR16</td>
<td>ResNet-50</td>
<td>60.19 M</td>
<td>11.58 G</td>
<td>559</td>
<td>2721</td>
<td>79.39</td>
<td>94.74</td>
</tr>
<tr>
<td>SENet [21]</td>
<td>CVPR18</td>
<td>SENet ResNet-50</td>
<td>66.77 M</td>
<td>11.60 G</td>
<td>508</td>
<td>2566</td>
<td>79.84</td>
<td>94.82</td>
</tr>
<tr>
<td>AANet [3]</td>
<td>ICCV19</td>
<td>AANet ResNet-50</td>
<td>61.60 M</td>
<td>11.90 G</td>
<td>-</td>
<td>-</td>
<td>79.10</td>
<td>94.60</td>
</tr>
<tr>
<td>ECANet [36]</td>
<td>CVPR20</td>
<td>ECANet ResNet-50</td>
<td>60.19 M</td>
<td>11.59 G</td>
<td>515</td>
<td>2619</td>
<td>79.86</td>
<td>94.80</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td></td>
<td></td>
<td>66.77 M</td>
<td>11.60 G</td>
<td>502</td>
<td>2387</td>
<td>80.13</td>
<td>94.90</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td></td>
<td></td>
<td>66.77 M</td>
<td>11.60 G</td>
<td>502</td>
<td>2387</td>
<td>80.02</td>
<td>94.89</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td></td>
<td></td>
<td>66.77 M</td>
<td>11.60 G</td>
<td>502</td>
<td>2387</td>
<td>79.96</td>
<td>94.94</td>
</tr>
</tbody>
</table>

* Please note that although GSoPNet1 has higher performance, the computational cost is 1.5 times larger than ours.

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 because the weights of DCT are predefined constants.

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 backbones, 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 1.

A Few lines of code change  Another important property of the proposed multi-spectral framework is that it can be easily carried out with existing channel attention implementations. The only difference between our method and SENet is the channel compression method (GAP vs. multi-spectral 2D DCT). As described in Sec. 3.1 and Eq. 7, 2D DCT can be viewed as a weighted sum of inputs. It can be simply implemented by element-wise multiplication and summation. In this way, our method could be easily integrated into arbitrary channel attention methods.

4.4. 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 [21], CBAM [39], GSoP-Net1 [11], GCNet [4], AANet [3], and ECANet [36]. The evaluation metrics include both efficiency (i.e., network parameters, floating point operations per second (FLOPs), and frame per second (FPS)) and effectiveness (i.e., Top-1/Top-5 accuracy). As shown in Table 1, our method achieves the best performance nearly in all experimental settings.
Table 2. Object detection results of different methods on COCO val 2017.

<table>
<thead>
<tr>
<th>Method</th>
<th>Detector</th>
<th>Parameters</th>
<th>FLOPs</th>
<th>AP</th>
<th>AP_{50}</th>
<th>AP_{75}</th>
<th>AP_{S}</th>
<th>AP_{M}</th>
<th>AP_{L}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td></td>
<td>41.53 M</td>
<td>215.51 G</td>
<td>36.4</td>
<td>58.2</td>
<td>39.2</td>
<td>21.8</td>
<td>40.0</td>
<td>46.2</td>
</tr>
<tr>
<td>SENet</td>
<td>Faster-RCNN</td>
<td>44.02 M</td>
<td>215.63 G</td>
<td>37.7</td>
<td>60.1</td>
<td>40.9</td>
<td>22.9</td>
<td>41.9</td>
<td>48.2</td>
</tr>
<tr>
<td>ECANet</td>
<td>Faster-RCNN</td>
<td>41.53 M</td>
<td>215.63 G</td>
<td>38.0</td>
<td>60.6</td>
<td>40.9</td>
<td>23.4</td>
<td>42.1</td>
<td>48.0</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td>Faster-RCNN</td>
<td>44.02 M</td>
<td>215.63 G</td>
<td>39.0</td>
<td>61.3</td>
<td>41.9</td>
<td>23.4</td>
<td>42.5</td>
<td>49.7</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td>Faster-RCNN</td>
<td>44.02 M</td>
<td>215.63 G</td>
<td>39.0</td>
<td>61.1</td>
<td>42.3</td>
<td>23.7</td>
<td>42.8</td>
<td>49.6</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td>Faster-RCNN</td>
<td>44.02 M</td>
<td>215.63 G</td>
<td>39.0</td>
<td>60.9</td>
<td>42.3</td>
<td>23.0</td>
<td>42.9</td>
<td>49.9</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>Faster-RCNN</td>
<td>60.52 M</td>
<td>295.39 G</td>
<td>38.7</td>
<td>60.6</td>
<td>41.9</td>
<td>22.7</td>
<td>43.2</td>
<td>50.4</td>
</tr>
<tr>
<td>SENet</td>
<td>Faster-RCNN</td>
<td>65.24 M</td>
<td>295.58 G</td>
<td>39.6</td>
<td>62.0</td>
<td>43.1</td>
<td>23.7</td>
<td>44.0</td>
<td>51.4</td>
</tr>
<tr>
<td>ECANet</td>
<td>Faster-RCNN</td>
<td>60.52 M</td>
<td>295.58 G</td>
<td>40.3</td>
<td>62.9</td>
<td>44.0</td>
<td>24.5</td>
<td>44.7</td>
<td>51.3</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td>Faster-RCNN</td>
<td>65.24 M</td>
<td>295.58 G</td>
<td>41.3</td>
<td>63.4</td>
<td>44.9</td>
<td>24.6</td>
<td>45.6</td>
<td>53.6</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td>Faster-RCNN</td>
<td>65.24 M</td>
<td>295.58 G</td>
<td>41.2</td>
<td>63.3</td>
<td>44.6</td>
<td>23.8</td>
<td>45.2</td>
<td>53.1</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td>Faster-RCNN</td>
<td>65.24 M</td>
<td>295.58 G</td>
<td>41.2</td>
<td>63.3</td>
<td>44.6</td>
<td>23.8</td>
<td>45.2</td>
<td>53.0</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>Mask-RCNN</td>
<td>44.17 M</td>
<td>261.81 G</td>
<td>37.2</td>
<td>58.9</td>
<td>40.3</td>
<td>22.2</td>
<td>40.7</td>
<td>48.0</td>
</tr>
<tr>
<td>SENet</td>
<td>Mask-RCNN</td>
<td>46.66 M</td>
<td>261.93 G</td>
<td>38.7</td>
<td>60.9</td>
<td>42.1</td>
<td>23.4</td>
<td>42.7</td>
<td>50.0</td>
</tr>
<tr>
<td>GCNet</td>
<td>Mask-RCNN</td>
<td>46.69 M</td>
<td>261.94 G</td>
<td>39.4</td>
<td>61.6</td>
<td>42.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ECANet</td>
<td>Mask-RCNN</td>
<td>44.17 M</td>
<td>261.93 G</td>
<td>39.0</td>
<td>61.3</td>
<td>42.1</td>
<td>24.2</td>
<td>42.8</td>
<td>49.9</td>
</tr>
<tr>
<td>FcaNet-LF</td>
<td>Mask-RCNN</td>
<td>46.66 M</td>
<td>261.93 G</td>
<td>40.3</td>
<td>61.9</td>
<td>43.9</td>
<td>24.9</td>
<td>43.6</td>
<td>52.2</td>
</tr>
<tr>
<td>FcaNet-TS</td>
<td>Mask-RCNN</td>
<td>46.66 M</td>
<td>261.93 G</td>
<td>40.3</td>
<td>62.0</td>
<td>44.1</td>
<td>25.2</td>
<td>43.9</td>
<td>52.0</td>
</tr>
<tr>
<td>FcaNet-NAS</td>
<td>Mask-RCNN</td>
<td>46.66 M</td>
<td>261.93 G</td>
<td>40.3</td>
<td>61.9</td>
<td>43.9</td>
<td>24.9</td>
<td>43.6</td>
<td>52.2</td>
</tr>
</tbody>
</table>

4.5. 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 [25] 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 2, 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.

4.6. Instance Segmentation on MS COCO

Besides the object detection, we then test our method on the instance segmentation task. As shown in Table 3, 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 study a fundamental problem of channel attention, that is, how to represent channels and regard this problem as a compression process. We have proved 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 three criteria for frequency components selection. With the same number of parameters and computational cost, our method could consistently outperform SENet. 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 a few lines of code change.

Acknowledgements

This work is supported in part by National Key Research and Development Program of China under Grant 2020AAA0107400, Zhejiang Provincial Natural Science Foundation of China under Grant LR19F020004, key scientific technological innovation research project by Ministry of Education, and National Natural Science Foundation of China under Grant U20A20222.


