Expert Systems With Applications 260 (2025) 125463

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Frequency Regulated Channel-Spatial Attention module for improved image classification

Chengyuan Zhuang [c,1], Xiaohui Yuan [c,*], Lichuan Gu [a], Zhenchun Wei [b], Yuqi Fan [b], Xuan Guo [c]

[a] Anhui Agriculture University, Hefei, 230036, China
[b] Hefei University of Technology, Hefei, 230009, China
[c] University of North Texas, Denton, TX, 76203, USA

ARTICLE INFO

Keywords:
Classification
Attention
Regularization
Filter

ABSTRACT

Convolutional neural networks play a vital role in image classification, with attention mechanisms enhancing discriminability on large datasets like ImageNet. However, challenges persist in optimizing performance for smaller or moderate real-world datasets due to limited data availability. There is a deficiency in effectively leveraging both channel and spatial attention for enhanced effectiveness in such scenarios. Although predefined filters offer advantages, their integration with attention mechanisms for complementary strength remains under-explored. In this paper, we propose the Frequency Regulated Channel-Spatial Attention (FReCSA) module to address this challenge by leveraging the power of channel attention and spatial attention. Four subsets and the complete ImageNet dataset, along with five additional datasets are used to evaluate FReCSA in our experiments. Integrating the FReCSA module into ResNet50 significantly enhances the top-1 accuracy, which is demonstrated by a 10.13% increase over the second-best on the ImageNet-40 dataset. On the other hand, our FReCSA module introduces minimal computational and parameter overhead to the deep network in terms of model size and computational operations, which makes FReCSA a good choice for learning tasks. The source code of this work is available at https://github.com/CoVIS-UNT/FReCSA.

1. Introduction

Convolutional neural networks (CNNs) have revolutionized computer vision, particularly image classification, as demonstrated by their success in the ImageNet Challenge (Russakovsky et al., 2015). To achieve satisfactory performance, a large amount of training data is commonly required. The development of deep networks for improved performance includes exploration of increasing network depth (He, Zhang, Ren, & Sun, 2016; Simonyan & Zisserman, 2015) or width (Zagoruyko & Komodakis, 2016), which in turn leads to additional layers or channels and therefore an increased number of parameters. Alternatively, attention (Qin, Zhang, Wu, & Li, 2021; Wang et al., 2020) has shown promise by emphasizing informative features while suppressing less useful ones, with the advantage of very few additional computational operations to the existing network. Still, the requirement for massive training datasets persists. In many real-world applications, however, such a large amount of training data is often unavailable. The limited training data poses a substantial challenge for deep neural networks (Alzubaidi et al., 2023; Brigato, Barz, Iocchi, & Denzler, 2022).

Although attention was designed to improve the encoding process of deep networks, weighing features according to their significance makes it a promising idea to address the requirement of large training data (Guo et al., 2022). With the extraction of more relevant, prominent features, less data could be used to achieve competitive performance (Hassanin, Anwar, Radwan, Khan, & Mian, 2024). While channel attention and spatial attention have been developed, they focus on massive datasets (e.g., ImageNet Deng et al., 2009), with less emphasis on optimization for smaller or moderate datasets due to data availability. There is also a deficiency in effectively harnessing both channel attention and spatial attention to enhance overall effectiveness in such a scenario. Additionally, studies (Chen et al., 2019; Ma, Luo, & Yang, 2020; Ulicny, Krylov, & Dahyot, 2019) suggest that incorporating predefined filters not only improves performance but also reduces the number of parameters that need to be learned. This, in turn, enables

* Corresponding author.
E-mail addresses: chengyuan.zhuang1@upr.edu (C. Zhuang), xiaohui.yuan@unt.edu (X. Yuan), glc@ahau.edu.cn (L. Gu), weizc@hfut.edu.cn (Z. Wei), yuqi.fan@hfut.edu.cn (Y. Fan), xuan.guo@unt.edu (X. Guo).

[1] Dr. Chengyuan Zhuang is currently with the Department of Computer Science at the University of Puerto Rico, Río Piedras. The work was done when he was with the University of North Texas.

https://doi.org/10.1016/j.eswa.2024.125463
Received 7 April 2024; Received in revised form 11 September 2024; Accepted 25 September 2024
Available online 28 September 2024

a more data-efficient training process. However, the exploration of integrating predefined filters with attention mechanisms to harness the strengths of both remains under-explored.

This paper proposes a frequency-regulated attention module with enhanced channel and spatial attention designs to address the aforementioned challenge. Our channel attention module incorporates Batch Normalization after channel embedding to accelerate learning and enhance frequency features. The learnable individual channel scaling simplifies connections and facilitates learning the contributions of different channels that capture diverse frequency components. Our spatial attention module leverages predefined filters to alleviate the reliance on large datasets for training. The filter outputs are used to modulate the input for spatial attention, enabling efficient learning of prominent features. The design of our method is highly modulated and can be integrated into popular deep networks with ease. Similar to many attention modules, our module can be appended to the end of convolution operations within each convolutional block.

The main contributions of this work include

- a simplified channel attention design incorporating Batch Normalization and individual channel scaling,
- a frequency-modulated spatial attention design integrating predefined filters via local-local spatial interaction,
- an improved deep network that leverages the complementary power of both attention modules for smaller or moderate datasets.

The rest of this paper is organized as follows: Section 2 provides a review of related work, primarily focusing on attention mechanisms and predefined filters. Section 3 describes our proposed method in detail. Section 4 presents the experimental results with a discussion. Section 5 concludes and summarizes the article.

2. Related work

To increase the volume of data for training deep network models, synthetic data is often created and employed using different techniques. Variational autoencoders (VAEs) (Kingma & Welling, 2013) employ an encoder network to generate a probability distribution over the latent space from the input image and use a decoder network to randomly sample and generate the output. Generative adversarial networks (GANs) (Goodfellow et al., 2014) train a generator to produce synthetic data that is difficult to distinguish and a discriminator to differentiate between the generated and real data. However, there remains a substantial domain gap between synthetic and real images (Man & Chahl, 2022). Training purely on synthetic data would not improve testing performance on real data (Tsirikoglou, Eilertsen, & Unger, 2020). It is inferior and much less data-efficient when trained from scratch, and there is also performance degradation during fine-tuning compared to real data (He et al., 2022).

To enhance recognition performance, researchers have explored increased network depth or width, leading to more layers or channels. Using small filter sizes, VGGNet (Simonyan & Zisserman, 2015) extends its depth beyond 16 convolutional layers, GoogLeNet (Szegedy et al., 2015) surpasses 20 convolutional layers, and ResNet (He et al., 2016) exceeds 50 convolutional layers due to its residual design. All of these contribute significantly to the improvement; however, further increasing depth leads to marginal gains, even with the large dataset ImageNet. Increased width (Zagoruyko & Komodakis, 2016) is alternatively explored, resulting in substantial computational cost and constrained depth. The increase in depth or width also results in a significant growth in parameters, posing a higher risk of overfitting and greater demand for data volume.

Alternatively, attention mechanisms emphasize the most informative features for enhanced discriminability. They are typically designed as an add-on module with a sigmoid control gate for recalibration through multiplication. Residual Attention Network (Wang et al., 2017) directly computes the 3D attention map. It adopts an encoder–decoder style with multiple downsampling (max-pooling), residual units, and upsampling, followed by two 1×1 convolutions. To reduce overhead, the decomposition of attention along the channel or spatial axis is subsequently investigated.

Channel attention recalibrates channels based on their importance, which is implemented through either fully connected or simplified connections with respect to channel interactions. For the fully connected style, the 'Squeeze-and-Excitation' (SE) module from SENet (Hu, Shen, & Sun, 2018b) is widely adopted in various works, including Cao, Xu, Lin, Wei, and Hu (2019), Li, Wang, Hu, and Yang (2019), Park, Woo, Lee, and Kweon (2018), Qin et al. (2021), Woo, Park, Lee, and Kweon (2018), Zhang, Zu, Lu, Zou, and Meng (2022) and Zhuang et al. (2023). This module incorporates a squeeze operation for channel embedding using global average pooling (GAP) and an excitation operation for channel dependency modeling, through two fully connected layers (FC) with dimension reduction. For modifications made during the adoption, BAM (Park et al., 2018) introduces Batch Normalization (Ioffe & Szegedy, 2015) between the FC layers. CBAM (Woo et al., 2018) includes more information through global max and average pooling. KNets (Li et al., 2019) incorporate more convolution branches with different kernel sizes for a 'Selective Kernel' (SK) strategy. GCNet (Cao et al., 2019) instead explores long-range dependencies for channel embedding, with LayerNorm (Ba, Kiros, & Hinton, 2016) between the FC layers. FcaNet (Qin et al., 2021) incorporates additional frequencies into channel embedding, including the lowest frequencies (-LF) or selected frequencies through a two-step process (-TS). GSoP-Net (Gao, Xie, Wang, & Li, 2019) alternatively captures channel correlations using the covariance matrix, which involves pairwise location dependencies across channels and is employed between channel reduction and row-wise convolution. On the other hand, simplified connections receive limited exploration. SRM (Lee, Kim, & Nam, 2019) recalibrates individual channels for style-related tasks, exploring style information through channel mean and variance. It uses separate learnable weighting parameters for summation, serving as channel encoding before Batch Normalization. ECA-Net (Wang et al., 2020) introduces a local cross-channel interaction through 1D convolution to enhance the efficiency of the SE module. It reveals the unnecessary nature of modeling dependencies across all channels and the adverse effect of channel dimension reduction.

Spatial attention recalibrates features based on their spatial importance, which is implemented through local-only, local-local, or local–global interactions. Local-only interaction typically employs convolution to capture local importance. BAM (Park et al., 2018) employs two dilated convolutions (3×3) on reduced channels, with channel attention in parallel through summation. CBAM (Woo et al., 2018) similarly applies a 7×7 convolution on the concatenated features through max and average pooling along the channel axis, with channel attention first. The GE module (Hu, Shen, Albanie, Sun, & Vedaldi, 2018) integrates a series of strided depthwise convolutions (3×3) to achieve a large spatial extent for subsequent redistribution. All these methods incorporate Batch Normalization. Furthermore, HPA (Zhuang et al., 2023) employs a predefined high-pass filter on individual channels to facilitate learning spatial attention, resulting in additional channels to be concatenated and explored by the SE module for channel attention. For local-local interaction, Non-Local Network (Wang, Girshick, Gupta, & He, 2018) adopts the self-attention mechanism to capture long-range dependencies. This is achieved by computing pairwise relations between all spatial positions within the non-local block. CCNet (Huang, Wang et al., 2019) simplifies pairwise relations through two consecutive Criss-Cross (CC) modules for semantic segmentation, with each module considering the dependencies between each pixel and the pixels along its horizontal and vertical paths. SimAM (Yang, Zhang, Li, & Xie, 2021) introduces a simple attention module design to highlight distinctive values from spatial surroundings. It normalizes squared zero-centered data values based on channel variance, using a predefined

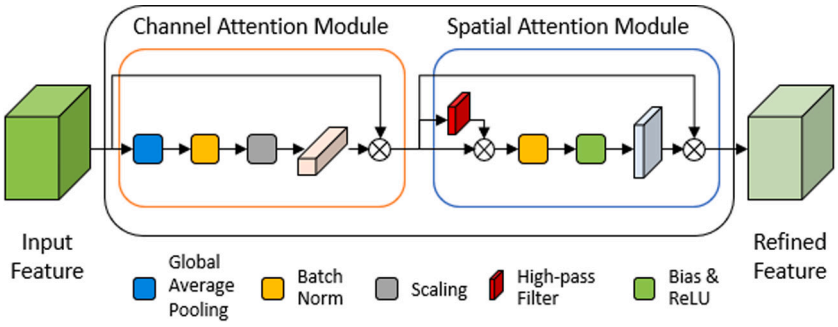

Fig. 1. The structure of our proposed FReCSA module.

energy function with no learnable parameters. For local–global interaction, the SGE module (Li, Li, & Yang, 2022) explores local–global spatial similarity within channel groups. This module is implemented through global average pooling, element-wise multiplication, and a simplified version of Batch Normalization with less learnable parameters.

On the other hand, the incorporation of predefined filters in CNNs has been explored, which can be implemented through direct replacement, integration within, or as an add-on component with respect to existing learnable filters. For direct replacement, Deep Hybrid Networks (Oyallon, Belilovsky, & Zagoruyko, 2017) substitute learnable convolutions in the initial layers with predefined filters derived from wavelet transformations, demonstrating that it can be data-hungry to rely solely on learnable filters. Harmonic Networks (Ulicny et al., 2019) replace learnable convolution filters with Discrete Cosine Transform filters within the harmonic block across more layers. Additionally, Batch Normalization is applied to enhance high-frequency components with small magnitudes. For integration within, OctConv (Chen et al., 2019) incorporates a low-pass filter into the original convolution, which generates additional low-frequency features in reduced resolution, allowing convolution within and between the new and original features. This process helps in learning the latter, which is ultimately retained for the last layer. For the add-on component, the Multi-level Wavelet CNN (Liu, Zhang, Lian, & Zuo, 2019) incorporates discrete wavelet transform during the downsampling stage, generating low- and high-frequency bands to ease the learning burden, with expanded channel numbers followed by channel reduction. High-frequency residual learning (Cheng, Xiao, Wang, Huang, & Zhang, 2020) employs a dual network strategy for small networks, which leverages low-frequency features from an auxiliary network using low-resolution input to facilitate the learning of high-frequency information. Anti-aliasing (AA) (Zhang, 2019) integrates a low-pass filter after dense computation in the downsampling process to mitigate aliasing artifacts from high-frequency signals and therefore enhance feature learning.

3. Frequency regulated channel-spatial attention

Our proposed Frequency Regulated Channel-Spatial Attention (FReCSA) module consists of two key components: a simplified channel attention module and a frequency-regulated spatial attention module. Fig. 1 illustrates the overview of our FReCSA module, where each component recalibrates the input feature map through multiplication in a sequential order.

3.1. Simplified channel attention

Given the input feature map $F \in \mathbb{R}^{C\times H\times W}$, our channel attention module learns to infer the 1D channel attention map $A \in \mathbb{R}^{C\times 1\times 1}$ for channel recalibration. This module adopts simplified channel connections and comprises four steps: (1) channel embedding, (2) normalization, (3) scaling, and (4) recalibration. The structure of our channel attention module is illustrated in Fig. 2.

Channel Embedding. Global Average Pooling (GAP) is applied to generate the channel descriptor $P \in \mathbb{R}^{C\times 1\times 1}$:

$$P = \mathrm{GAP}(F), \tag{1}$$

where F denotes the input feature map. The pooling operation aggregates feature responses across all locations ($H\times W$) within one channel into a single value to capture the global distribution. Therefore, the cth channel of the descriptor P represents an average value of that channel:

$$P_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c(i,j), \tag{2}$$

where $F_c(i,j)$ refers to the feature value at the spatial location (i,j) in the cth channel of the input feature map.

Normalization. Different channels that capture various frequencies (Hinton, Krizhevsky, & Sutskever, 2012; Zeiler & Fergus, 2014) could exhibit diverse characteristics. This diversity may lead to significant variations in the magnitude of the channel descriptor (e.g., high-frequency features are usually sparse), posing challenges in the subsequent steps of learning the relative contributions of channels. Instead of directly applying the channel descriptor, we incorporate Batch Normalization (BN) to obtain the normalized channel descriptor $B \in \mathbb{R}^{C\times 1\times 1}$:

$$B = \mathrm{BN}(P), \tag{3}$$

where P denotes the channel descriptor from the previous step. The normalization value for the cth channel, represented as B_c, is calculated as follows:

$$B_c = \frac{P_c - \bar{P}_c}{\sqrt{\sigma(P_c)+\varepsilon}}\times\gamma+\beta, \tag{4}$$

where P_c refers to the cth channel of the channel descriptor P, $\bar{P}_c$ denotes the mean of P_c, $\sigma(P_c)$ computes the standard deviation of P_c, while γ and β are the scaling parameter and bias for enhanced representation power, which take an initial value of 1 and 0, respectively.

Scaling. The batch-normalized channel descriptor B may not be ideal to directly recalibrate each channel of the input feature map F, particularly in the absence of a substantial amount of data. Therefore, we introduce a learnable scaling parameter $V \in \mathbb{R}^{C\times 1\times 1}$, where each channel is assigned a unique scaling factor. These factors are initialized as 0 but can be easily learned to acquire suitable amplitudes. They can take positive values for amplification and negative values for attenuation before the Sigmoid function. This scaling process leads to the scaled channel descriptor $S \in \mathbb{R}^{C\times 1\times 1}$:

$$S = B\times V, \tag{5}$$

where B is the batch-normalized channel descriptor and V is the learnable scaling parameter.

Recalibration. The channel attention map $A \in \mathbb{R}^{C\times 1\times 1}$ is finally obtained by applying the Sigmoid function:

$$A = \mathrm{Sigmoid}(S), \tag{6}$$

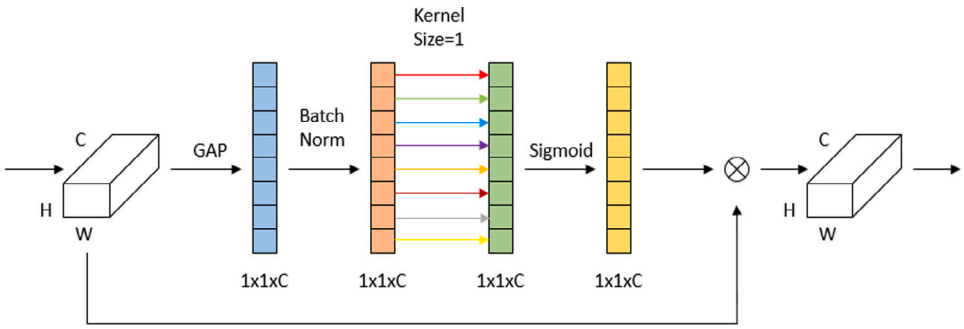

Fig. 2. The structure of our channel attention (CA) module.

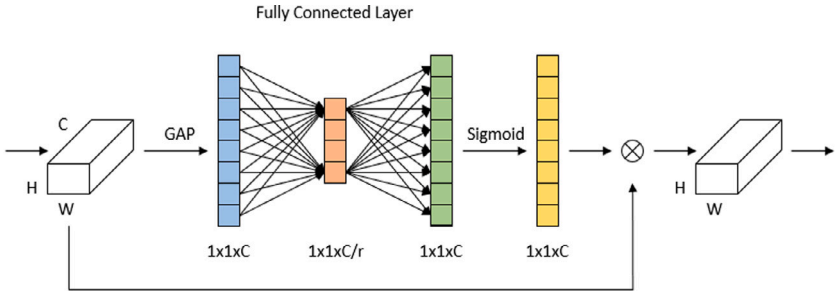

Fig. 3. The structure of the SE module.

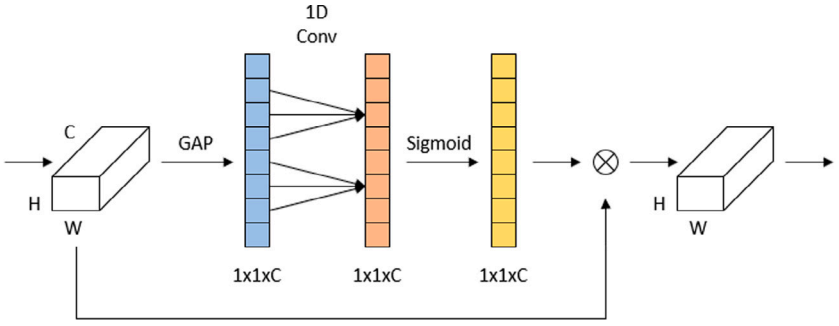

Fig. 4. The structure of the ECA module.

where S refers to the scaled channel descriptor. Subsequently, the channel-refined feature map F' is achieved by recalibration:

$$F' = A \times F, \tag{7}$$

where F represents the input feature map. Therefore, after recalibration, important channels are highlighted.

Design Distinctions. In Figs. 3 and 4, we illustrate two types of channel connections: fully connected (e.g., the SE module Hu et al., 2018b) and simplified connections (e.g., the ECA module Wang et al., 2020). Our channel attention design distinguishes itself by incorporating Batch Normalization and individual channel scaling for accelerated learning and increased flexibility. The independent channel relationship aims to alleviate the learning burden. This design differs from the fully connected connections used by the SE module, which links all channels, and the ECA module, which employs 1D convolution to connect nearby channels.

3.2. Frequency-regulated spatial attention

Given the input feature map $F' \in \mathbb{R}^{C\times H\times W}$, our frequency-regulated spatial attention module learns to infer the 2D spatial attention map $A' \in \mathbb{R}^{C\times H\times W}$ for spatial recalibration. This module is designed to enhance channel attention and benefit less voluminous datasets. Therefore, we adopt a local-local interaction strategy, incorporating a predefined high-pass filter to highlight spatial changes and alleviate the learning burden. This module consists of five steps: (1) predifined filtering, (2) local-local interaction, (3) normalization, (4) activation, and (5) recalibration. The structure of our spatial attention module is depicted in Fig. 5. Additional details on parameter and design selection are provided in the ablation study, Section 4.5.

Predifined Filtering. We explore spatial information for recalibration by incorporating a predefined high-pass filter (HPF). This filter preserves rapid changes while attenuating gradual changes, aiming to enhance the visibility and sharpness of features, such as edges or boundaries, without the need to learn additional parameters. The high-frequency features $High \in \mathbb{R}^{C\times H\times W}$ are generated within individual channels using the high-pass filter (HPF) as:

$$High = \mathrm{HPF}(F'), \tag{8}$$

where F' refers to the channel-refined feature map. This can be equivalently achieved by employing a low-pass filter (LPF) and subtracting the filtered signal from the original input:

$$\begin{aligned} Low &= \mathrm{LPF}(F'), \\ High &= F' - Low, \end{aligned} \tag{9}$$

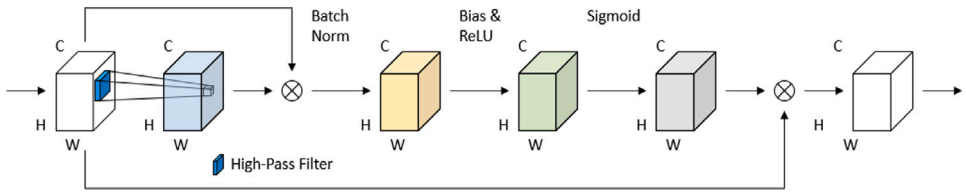

Fig. 5. The structure of our spatial attention (SA) module.

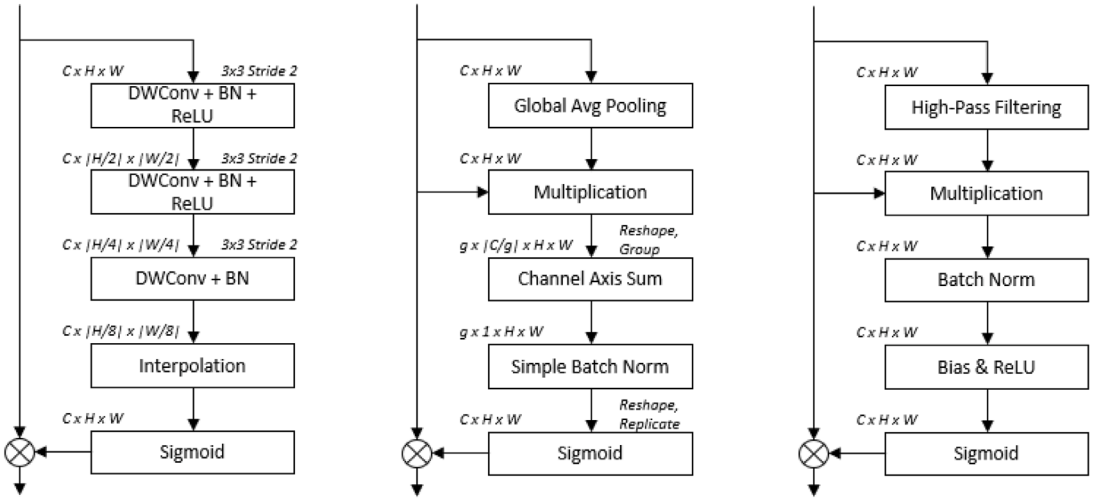

Fig. 6. The flow chart of three spatial interactions for spatial attention: (a) Local-only, (b) Local-global, and (c) Local-local.

where Low denotes the low-frequency features. In terms of implementation, we employ average pooling for the low-pass filter (LPF), and set filter size as 7 following CBAM (Woo et al., 2018).

Local-Local Interaction. To convey intensity information, we employ a local-local interaction strategy. This is achieved by exploring the similarity score $Sim \in \mathbb{R}^{C\times H\times W}$ between the original values and high-frequency features at corresponding locations within individual channels, which is calculated through pointwise multiplication:

$$Sim = High \times F', \tag{10}$$

where $High$ refers to the high-frequency features, and F' represents the channel-refined feature map.

Normalization. High-frequency information typically exhibits small values. To prevent the multiplication result from becoming excessively small, Batch Normalization (BN) is employed to obtain the normalized similarity score $S' \in \mathbb{R}^{C\times H\times W}$:

$$S' = \mathrm{BN}(Sim), \tag{11}$$

where Sim refers to the similarity score from the previous step. According to Eq. (4), the normalization is performed using the mean and standard deviation, with the scaling parameter and bias initialized as 1 and 0, respectively.

Activation. We further introduce an activation step before the Sigmoid function for recalibration to avoid eliminating signals with small values. We choose ReLU activation (Nair & Hinton, 2010):

$$\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \le 0 \end{cases} \tag{12}$$

to replace negative values with zero and allow positive values to pass through unchanged. In this activation step, a bias term b is included to enhance the contrast in the output for the negative inputs. The activation feature map $Z \in \mathbb{R}^{C\times H\times W}$ is computed as:

$$Z = \mathrm{ReLU}(S' + b), \tag{13}$$

where S' refers to the normalized similarity score. Bias value is investigated in the ablation study.

Recalibration. The spatial attention map $A' \in \mathbb{R}^{C\times H\times W}$ is finally obtained by applying the Sigmoid function:

$$A' = \mathrm{Sigmoid}(Z), \tag{14}$$

where Z is the activation feature map from the previous step. Subsequently, the spatial-refined feature map F'' is achieved through recalibration:

$$F'' = A' \times F', \tag{15}$$

where F' represents the channel-refined feature map as input. This spatial-refined feature map F'' serves as the final output of our entire module.

Design Distinctions. We highlight the key design differences that distinguish our spatial attention module. In Fig. 6, we illustrate three types of spatial interactions: local only (e.g., the GE module Hu et al., 2018), local–global (e.g., the SGE module Li et al., 2022) and local-local (our SA module). Our spatial attention design distinguishes itself by incorporating a predefined high-pass filter to alleviate the learning burden, as shown in the flowchart. Additionally, we establish local-local interaction by multiplying the original and high-frequency information at corresponding spatial locations. Our interaction differs from the local-only interaction employed by the GE module, which is exclusively based on convolution, and the local–global interaction used by the SGE module, which involves the original and the global average pooling information for multiplication.

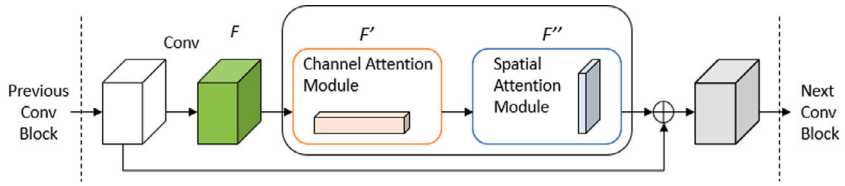

Fig. 7. The FReCSA module within the residual block of ResNet.

Table 1
ResNet50 architecture and with FReCSA module added.

ResNet-50	FReCSA-ResNet-50
conv, 7 × 7, 64, stride 2	
max pool, 3 × 3, stride 2	
[conv, 1 × 1, 64 conv, 3 × 3, 64 conv, 1 × 1, 256] × 3	[conv, 1 × 1, 64 conv, 3 × 3, 64 conv, 1 × 1, 256 FReCSA, 256] × 3
[conv, 1 × 1, 128 conv, 3 × 3, 128 conv, 1 × 1, 512] × 4	[conv, 1 × 1, 128 conv, 3 × 3, 128 conv, 1 × 1, 512 FReCSA, 512] × 4
[conv, 1 × 1, 256 conv, 3 × 3, 256 conv, 1 × 1, 1024] × 6	[conv, 1 × 1, 256 conv, 3 × 3, 256 conv, 1 × 1, 1024 FReCSA, 1024] × 6
[conv, 1 × 1, 512 conv, 3 × 3, 512 conv, 1 × 1, 2048] × 3	[conv, 1 × 1, 512 conv, 3 × 3, 512 conv, 1 × 1, 2048 FReCSA, 2048] × 3
Global average pool, 1000-d fc, softmax	

Table 2
The number of training, Test images, and categories.

Dataset	Training	Test	Category
ImageNet-40	40,000	50,000	1000
ImageNet-80	80,000	50,000	1000
ImageNet-160	160,000	50,000	1000
ImageNet-320	320,000	50,000	1000
ImageNet-full	1,281,167	50,000	1000
Food-101	75,750	25,250	101
Oxford-IIIT Pet	3680	3669	37
Caltech-256	7680	6400	256
SUN397	19,850	19,850	397
MINC	48,875	5750	23

Table 3
The parameters used in our experiments.

Parameter	Value	Parameter	Value
Image size	224 × 224	Random scale	[0.08, 1.0]
Batch size	128	Random ratio	[3/4, 4/3]
Batch acc. (Im/ft)	2/1	Initial lr (Im)	0.1
SGD momentum	0.9	Ini. lr (Food/SUN)	0.0125
Weight decay	0.0001	Ini. lr (Pet/MINC/Caltech)	0.00375

3.3. Module integration

We illustrate the placement of our FReCSA module within the Residual Block of ResNet (He et al., 2016) in Fig. 7, positioned similarly to other attention methods such as SE (Hu et al., 2018b), CBAM (Woo et al., 2018), etc. Specifically, in the Residual Block, our module is situated on the convolutional branch after the existing convolution operation. It sequentially derives the channel-refined feature map F' and the spatial-refined feature map F'' as output, given the input feature map F. For simplicity, Batch Normalization (BN) and ReLU activation are omitted.

We describe the architecture of ResNet-50 and FReCSA-ResNet-50 for the ImageNet dataset (1000 categories) in Table 1, with the latter incorporating our proposed module. After the initial convolution, ResNet50 comprises four convolutional stages, each consisting of 3, 4, 6, and 3 convolutional blocks, respectively. Our module is positioned within each block after the last convolution while keeping the output channel number unchanged.

4. Experimental results and discussion

4.1. Experiment settings and data sets

In our experiments, we utilize six datasets, including ImageNet (Deng et al., 2009), Food-101 (Bossard, Guillaumin, & Van Gool, 2014), Oxford-IIIT Pet (Parkhi, Vedaldi, Zisserman, & Jawahar, 2012), Caltech-256 (Griffin, Holub, & Perona, 2007), SUN397 (Xiao, Hays, Ehinger, Oliva, & Torralba, 2010), and MINC (Bell, Upchurch, Snavely, & Bala, 2015). ImageNet contains 1.28 million training images and 50,000 validation images of 1000 object categories. To explore scenarios with smaller-scale training data, we create four subsets by randomly selecting 40, 80, 160, and 320 training images per category, denoted as ImageNet-40, ImageNet-80, ImageNet-160, and ImageNet-320, respectively, while keeping the complete validation data. The remaining datasets, which are already limited in size, are introduced as follows. Food-101 comprises 750 training images and 250 test images per category across 101 food categories. Oxford-IIIT Pet consists of approximately 100 training images and 100 test images per category within 37 categories of cats and dogs. Caltech-256 includes 30 training and 25 test images per category across 256 object categories. SUN397 features 50 training images and 50 test images per category among 397 scene categories. MINC contains 2125 training images and 250 test images per category within 23 material categories. Dataset information is summarized in Table 2.

For training from scratch on ImageNet (Im) datasets, we follow the practice regarding ResNet (He et al., 2016), with input image size 224 × 224, initial learning rate (lr) 0.1 (which decays by a factor of 10 every 30 epochs), optimizer SGD, momentum 0.9, and weight decay 0.0001. Data augmentation includes random horizontal flipping as in Hinton et al. (2012), along with random scale in the range [0.08, 1.0] and random aspect ratio in the range [3/4, 4/3] as in Szegedy et al. (2015). Experiments are conducted on 4 GPUs for 100 epochs, with a batch size of 128 and batch accumulation of 2, following Zou, Xiao, Yu, and Lee (2020). We use the Pytorch (Paszke et al., 2019) deep learning framework, following its official ImageNet training example. For fine-tuning (ft) on target datasets, we set batch accumulation as 1, and select learning rate from {0.15, 0.115, 0.075, 0.0125, 0.00375} as suggested by Huang, Cheng et al. (2019). We evaluate the performance of ResNet using 50% of the training data for validation, and the selected learning rate is 0.0125 for Food-101 and SUN397 and 0.00375 for the other datasets. The parameters are summarized in Table 3.

4.2. ImageNet classification

We evaluate our proposed FReCSA module and the representative approaches introduced in related work from both the perspectives of

Table 4
Top-1 accuracy (%) on ImageNet datasets, including ImageNet-40, ImageNet-80, ImageNet-160, ImageNet-320, and ImageNet-full.

Method	Training dataset size per class				
	40	80	160	320	Full
ResNet-50	22.56	38.58	52.33	63.52	76.00
GCNet (Cao et al., 2019)	22.03	38.04	52.78	64.88	76.93
AA (Zhang, 2019)	22.72	39.27	53.16	65.29	76.92
OctConv (Chen et al., 2019)	22.80	39.54	54.01	65.27	76.94
SKNet (Li et al., 2019)	22.87	39.96	54.34	65.74	77.09
FCA-TS (Qin et al., 2021)	23.71	40.10	54.44	66.12	77.49
Wavelet (Liu et al., 2019)	24.45	39.70	53.96	64.41	76.56
FCA-LF (Qin et al., 2021)	25.11	40.13	54.66	65.65	77.33
SRM (Lee et al., 2019)	25.94	42.01	55.56	65.65	76.97
SENet (Hu et al., 2018b)	26.06	41.41	55.58	65.88	77.20
Harmonic (Ulicny et al., 2019)	26.07	42.75	56.13	66.02	76.95
ECANet (Wang et al., 2020)	26.08	40.68	54.78	65.94	77.24
GSoPNet (Gao et al., 2019)	26.74	42.39	56.54	67.08	**77.78**
Simam (Yang et al., 2021)	27.01	42.37	56.22	66.27	76.96
CBAM (Woo et al., 2018)	27.09	42.35	55.95	66.15	77.48
HPA (Zhuang et al., 2023)	27.35	43.01	56.14	66.34	77.60
SGE (Li et al., 2022)	27.38	42.54	55.77	65.67	77.22
GENet (Hu et al., 2018)	27.54	42.29	55.66	66.12	77.24
FReCSA	**30.33**	**45.00**	**58.10**	**67.27**	77.51

attention and predefined filters, using ResNet-50 (He et al., 2016) as the baseline network. Table 4 presents the top-1 accuracy on the subsets and the complete ImageNet dataset, with methods sorted in ascending order based on their performance on the smallest ImageNet-40 (Im-40) dataset.

A performance enhancement by integrating the FReCSA module into ResNet50 is evident. The top-1 accuracy on the ImageNet-40 dataset witnesses a 34.44% increase, rising from 22.56% to 30.33%, surpassing the second-best performance by 10.13%. The improvements remain consistent at 16.64%, 11.03%, and 5.90% on the ImageNet-80, ImageNet-160, and ImageNet-320 datasets, exceeding the second-best by 4.63%, 2.76%, and 0.28%, respectively, with training data size doubled in each case. Such improvements demonstrate the advantages of integrating the FReCSA module into a deep network. When all examples are used for training a model, our method achieves a highly competitive performance at 77.51%.

Channel attention methods such as SRM (Lee et al., 2019), SENet (Hu et al., 2018b), ECANet (Wang et al., 2020), and GSoPNet (Gao et al., 2019) consistently demonstrate strong performance when a large amount of data is available. On the ImageNet-40 dataset, the accuracy is around or exceeding 26%. SRM performs slightly lower with a large amount of data, while ECANet eventually demonstrates its advantage of simplified channel connections over SENet. Notably, GSoPNet achieves excellent performance as data increases, indicating that learning pairwise location dependencies across channels is powerful but data-hungry. Incorporating spatial attention, such as CBAM (Woo et al., 2018) and HPA (Zhuang et al., 2023), further enhances SENet. The accuracy is above 27% on the ImageNet-40 dataset, and consistent improvement is observed across the remaining datasets. Spatial attention methods, such as Simam (Yang et al., 2021), GENet (Hu et al., 2018), and SGE (Li et al., 2022), demonstrate comparable or even better performance on the ImageNet-40 dataset, only slightly lag behind on the complete ImageNet dataset. Such results indicate the significance of spatial recalibration in facilitating feature learning, especially in the scenario with a less massive dataset. However, incorporating additional frequency components (e.g., FCA-LF Qin et al., 2021, FCA-TS Qin et al., 2021) into channel embedding requires substantial amounts of data to effectively enhance SENet. Learning long-range dependencies (e.g., GCNet Cao et al., 2019) or more branches (e.g., SKNet Li et al., 2019) appears to be challenging, with an accuracy below 23% on the ImageNet-40 dataset. Similarly, the same data requirement holds for GCNet and SKNet to achieve comparable performance with SENet. For non-attention methods that incorporate predefined filters, Harmonic (Ulicny et al., 2019) consistently exhibits strong performance, with Batch Normalization to enhance high-frequency information. It achieves an accuracy of 26.07% on the ImageNet-40 dataset, despite a slight lag as data increases. The wavelet method (Liu et al., 2019), which directly generates high- and low-frequency bands, outperforms methods indirectly learning high-frequency information, such as anti-aliasing (Zhang, 2019) and OctConv (Chen et al., 2019) on the ImageNet-40 dataset, while this advantage diminishes with increasing data size.

4.3. Fine-tuning on the target datasets

Table 5 presents the top-1 accuracy achieved through fine-tuning on the target datasets, with methods sorted in ascending order based on their performance on the Food-101 dataset. In this case, each model is initialized with pretrained weights from the complete ImageNet dataset. The output layer (fully connected layer) is replaced to match the category number of the target dataset, and all parameters are updated during the adaptation process.

While the pretrained weights ease the learning process on the target datasets by providing prior knowledge, we can still observe significant performance improvements to the baseline network ResNet50 by integrating our FReCSA module. The top-1 accuracy on the Food-101 dataset increases by 2.11%, rising from 87.25% to 89.09%, surpassing the second-best performance by 0.11%. The top-1 accuracy improvements remain consistent at 0.93%, 2.04%, 3.18%, and 2.56% for the Oxford-IIIT Pet, Caltech-256, SUN397, and MINC datasets, exceeding the second-best by 0.09%, 0.20%, 0.22%, and 0.20%, respectively. These results demonstrate that FReCSA extracts features according to the characteristics of the target datasets.

It is evident that channel attention methods exhibit impressive performance on the Food-101 dataset. SRM achieves an accuracy of 88.54%, while ECANet and GSoPNet attain accuracy levels of 88.78% and 88.99% (ranking as the second-best), respectively. In contrast, SENet only obtains an accuracy of 88.27% on this dataset. For other remaining datasets with a reduced amount of data, only ECANet maintains competitive performance, most likely due to its simplified channel connections for quick adaptation. Incorporating spatial attention, such as HPA and CBAM, consistently improves SENet, with an accuracy of 88.55% and 88.78% on the Food-101 dataset, respectively. The overall better performance of CBAM is likely attributed to the benefits from Batch Normalization in this scenario with generally less amount of training data. While incorporating additional frequency components (e.g., FCA-LF, FCA-TS) into channel embedding improves the performance of SENet to around 88.50% on the Food-101 dataset, its effectiveness is not consistent, particularly on the Pet dataset with the least training data. Spatial attention methods, such as Simam, GENet, and SGE, demonstrate strong overall performance, with an accuracy of

Table 5
Top-1 accuracy (%) on Food-101 (Food), Oxford-IIIT Pet (Pet), Caltech-256 (Caltech), SUN397 (SUN), and MINC datasets.

Method	Food	Pet	Caltech	SUN	MINC
ResNet-50	87.25	93.13	82.73	61.09	78.92
AA (Zhang, 2019)	87.87	93.92	84.11	62.28	79.08
SKNet (Li et al., 2019)	88.03	92.59	80.30	61.11	78.03
OctConv (Chen et al., 2019)	88.04	93.81	84.02	62.34	80.57
Wavelet (Liu et al., 2019)	88.17	93.60	83.73	62.89	80.38
Harmonic (Ulicny et al., 2019)	88.26	93.81	84.03	62.87	80.64
SENet (Hu et al., 2018b)	88.27	91.77	75.58	61.12	78.00
GCNet (Cao et al., 2019)	88.30	93.19	83.13	61.40	79.51
FCA-LF (Qin et al., 2021)	88.46	90.73	76.34	61.56	78.35
Simam (Yang et al., 2021)	88.48	92.07	83.69	61.64	80.26
GENet (Hu et al., 2018)	88.51	93.81	84.03	62.89	80.78
SRM (Lee et al., 2019)	88.54	92.61	78.53	61.65	79.84
HPA (Zhuang et al., 2023)	88.55	92.10	80.08	61.79	79.25
FCA-TS (Qin et al., 2021)	88.59	90.73	76.17	61.23	79.06
CBAM (Woo et al., 2018)	88.78	93.24	84.25	62.47	79.10
ECANet (Wang et al., 2020)	88.78	93.51	84.11	62.36	79.70
SGE (Li et al., 2022)	88.79	93.46	84.09	62.88	79.44
GSoPNet (Gao et al., 2019)	88.99	92.10	77.20	61.30	78.10
FReCSA	**89.09**	**94.00**	**84.42**	**63.03**	**80.94**

Table 6
Top-1 accuracy (%) on Food-101, Oxford-IIIT Pet, and MINC datasets, minimal 40 images per class. FReCSA† and FReCSA‡ represent our spatial and channel attention, respectively.

Method	Food			Pet		MINC	
	40	80	All	40	All	40	All
ResNet-50	66.40	72.91	87.25	92.23	93.13	61.84	78.92
GC (Cao et al., 2019)	66.19	73.46	88.30	92.04	93.19	61.04	79.51
AA (Zhang, 2019)	67.59	73.43	87.87	92.45	93.92	63.15	79.08
OctConv (Chen et al., 2019)	67.46	73.36	88.04	92.59	93.81	63.44	80.57
SK (Li et al., 2019)	65.61	72.78	88.03	90.00	92.59	58.80	78.03
FCA-TS (Qin et al., 2021)	64.51	72.78	88.59	86.07	90.73	55.98	79.06
Wavelet (Liu et al., 2019)	68.32	74.57	88.17	92.59	93.60	63.60	80.38
FCA-LF (Qin et al., 2021)	64.13	72.95	88.46	86.51	90.73	55.83	78.35
SE (Hu et al., 2018b)	64.17	72.57	88.27	87.52	91.77	57.43	78.00
Harmonic (Ulicny et al., 2019)	68.28	74.52	88.26	92.37	93.81	63.32	80.64
ECA (Wang et al., 2020)	68.23	75.01	88.78	92.21	93.51	62.23	79.70
GSoP (Gao et al., 2019)	65.40	74.18	88.99	88.39	92.10	56.40	78.10
CBAM (Woo et al., 2018)	67.63	74.63	88.78	92.56	93.24	60.66	79.10
HPA (Zhuang et al., 2023)	65.33	72.83	88.55	89.97	92.10	59.06	79.25
SGE (Li et al., 2022)	68.48	75.09	88.79	92.56	93.46	63.25	79.44
GE (Hu et al., 2018)	68.04	74.27	88.51	92.45	93.81	63.65	80.78
FReCSA†	67.85	73.73	88.05	92.50	93.68	**64.57**	80.68
FReCSA‡	68.23	75.28	88.87	92.56	93.57	62.38	80.28
FReCSA	**68.89**	**75.55**	**89.09**	**92.64**	**94.00**	62.97	**80.94**

88.48%, 88.51%, and 88.79% on the Food-101 dataset, respectively. For the remaining datasets, GENet exhibits impressive performance, comparable or close to the second-best, while Simam and SGE lag behind. For non-attention methods incorporating predefined filters, despite exhibiting lower performance on the Food-101 dataset, the overall results on the remaining datasets are impressive, probably due to the reduced data amount. Harmonic consistently achieves strong performance, closely approaching the second-best accuracy on the remaining datasets despite a lower accuracy of 88.26% on the Food-101 dataset. The wavelet method, with direct generation of high- and low-frequency bands, lags slightly behind overall, attaining 88.17% accuracy on the Food-101 dataset. In contrast, methods indirectly learning high-frequency information, such as anti-aliasing and OctConv, primarily demonstrate their advantage on small datasets, with the lowest accuracy of 87.87% and 88.04% on the Food-101 dataset, respectively.

Table 6 presents the accuracy of our proposed FReCSA module and related methods using the Food-101, Oxford-IIIT Pet, and MINC datasets as well as subsets of smaller sizes to train the models. Due to the size of these three datasets, subsets of images are randomly selected to create training sets of sizes at 40 or 80. The images for testing the trained models remain unchanged. We observe a performance drop for ResNet-50 on the Food-40 dataset (from 87.25% to 66.40%), MINC-40 dataset (from 78.92% to 61.84%), and Pet dataset (from 93.13% to 92.23%) when a smaller set of training images are used. Using a small set of training images, our FReCSA achieves a better accuracy for all cases, as highlighted with boldface fonts, whereas the second-best is highlighted with an underscore. Note that the improvement rate of FReCSA with respect to the second-best is much greater when a model is trained with a smaller dataset. For example, the accuracy of the FReCSA model trained with the MINC-40 dataset improved by 1.45% whereas the accuracy of the FReCSA model trained with the entire MINC dataset improved by 0.2%. This suggests that FReCSA learns from smaller training datasets more effectively.

4.4. Complexity analysis

Table 7 illustrates the number of parameters (in millions) as an indicator of model size across different datasets, sorted in ascending order. We observe a slight rise in the number of parameters as our module is integrated into the ResNet50 baseline network on the ImageNet dataset, increasing from 25.56 million to 25.63 million. The tiny increase introduced by our module is noteworthy, especially when contrasted with SENet-related methods, which typically exceed 28 million parameters, and the wavelet method, involving more than 35 million parameters. Additionally, only models such as Harmonic, OctConv, anti-aliasing, ECANet, and SGE contain fewer parameters. This pattern remains consistent in other datasets, with adjustments limited to the

Table 7
Model parameters (in Million) for ImageNet, Food-101, Oxford-IIIT Pet, Caltech-256, SUN397, and MINC datasets.

Method	Im	Food	Pet	Caltech	SUN	MINC
ResNet-50	25.56	23.71	23.58	24.03	24.32	23.56
Harmonic (Ulicny et al., 2019)	25.56	23.71	23.58	24.03	24.32	23.56
OctConv (Chen et al., 2019)	25.56	23.71	23.58	24.03	24.32	23.56
AA (Zhang, 2019)	25.56	23.71	23.58	24.03	24.32	23.56
Simam (Yang et al., 2021)	25.56	23.71	23.58	24.03	24.32	23.56
ECANet (Wang et al., 2020)	25.56	23.72	23.58	24.03	24.32	23.56
SGE (Li et al., 2022)	25.56	23.72	23.59	24.03	24.32	23.56
FReCSA	25.63	23.79	23.66	24.11	24.40	23.63
GENet (Hu et al., 2018)	26.06	24.21	24.08	24.53	24.82	24.05
SKNet (Li et al., 2019)	26.15	24.31	24.18	24.63	24.92	24.15
FCA-LF (Qin et al., 2021)	28.07	26.23	26.10	26.55	26.84	26.07
FCA-TS (Qin et al., 2021)	28.07	26.23	26.10	26.55	26.84	26.07
GCNet (Cao et al., 2019)	28.08	26.24	26.11	26.55	26.84	26.08
SENet (Hu et al., 2018b)	28.09	26.25	26.11	26.56	26.85	26.09
CBAM (Woo et al., 2018)	28.09	26.25	26.12	26.57	26.85	26.09
GSoPNet (Gao et al., 2019)	28.29	26.44	26.31	26.76	27.05	26.28
HPA (Zhuang et al., 2023)	29.72	27.88	27.75	28.20	28.48	27.72
Wavelet (Liu et al., 2019)	35.21	33.37	33.23	33.68	33.97	33.21

Table 8
Model computation per image by GFLOPs (Floating Point Operations in Billion) for ImageNet, Food-101, Oxford-IIIT Pet, Caltech-256, SUN397, and MINC datasets.

Method	ImNet	Food	Pet	Caltech	SUN	MINC
ResNet-50	4.122	4.120	4.120	4.120	4.121	4.120
Harmonic (Ulicny et al., 2019)	2.152	2.150	2.150	2.151	2.151	2.150
OctConv (Chen et al., 2019)	3.579	3.577	3.577	3.578	3.578	3.577
Simam (Yang et al., 2021)	4.122	4.120	4.120	4.120	4.121	4.120
FCA-LF (Qin et al., 2021)	4.124	4.123	4.122	4.123	4.123	4.122
FCA-TS (Qin et al., 2021)	4.124	4.123	4.122	4.123	4.123	4.122
ECA (Wang et al., 2020)	4.127	4.126	4.125	4.126	4.126	4.125
SGE (Li et al., 2022)	4.127	4.126	4.125	4.126	4.126	4.125
GC (Cao et al., 2019)	4.128	4.126	4.126	4.126	4.126	4.126
SE (Hu et al., 2018b)	4.130	4.128	4.128	4.128	4.129	4.128
GE (Hu et al., 2018)	4.143	4.141	4.141	4.141	4.141	4.141
FReCSA	4.150	4.148	4.148	4.148	4.148	4.148
CBAM (Woo et al., 2018)	4.151	4.149	4.149	4.150	4.150	4.149
SK (Li et al., 2019)	4.187	4.185	4.185	4.185	4.186	4.185
HPA (Zhuang et al., 2023)	4.940	4.938	4.938	4.939	4.939	4.938
AA (Zhang, 2019)	5.164	5.162	5.162	5.162	5.163	5.162
Wavelet (Liu et al., 2019)	6.292	6.290	6.290	6.290	6.291	6.290
GSoPNet (Gao et al., 2019)	6.405	6.404	6.403	6.404	6.404	6.403

last fully connected layer in accordance with the number of categories in the target dataset.

Table 8 illustrates the GFLOPs (Floating Point Operations in billions) as a measure of the computational complexity for each model. The calculation is based on an input image size of 224 × 224, and the values are sorted in ascending order except for the baseline network ResNet-50. We observe a slight increase in GFLOPs when integrating our module to the ResNet50 baseline on ImageNet, rising from 4.122 to 4.150 billion. This slight increase is particularly noteworthy when compared to HPA, anti-aliasing, wavelet, and GSoPNet, all of which have GFLOP values exceeding 5 or 6 billion. The same trend persists across other datasets, where only the last fully connected layer is adjusted according to the number of categories in the target dataset.

4.5. Ablation study

Submodule performance. To gain deeper insights into the proposed FReCS module, we evaluate the performance of our channel and spatial attention components. Table 9 illustrates the top-1 accuracy of individual components as well as the entire FReCSA module on four subsets and the complete ImageNet dataset. Although our spatial attention component is designed for enhancement rather than stand-alone performance, we could still observe the highly competitive performance of 27.98%, 43.40%, 56.40%, and 66.03% on four subsets, respectively, while the performance of 76.63% on the complete ImageNet dataset is moderate. As a comparison, our channel attention component achieves even higher performance of 28.94%, 44.40%, 57.08%, and 66.74%, respectively, with strong performance of 77.49% on the complete ImageNet dataset. The results of our entire FReCS module clearly demonstrate its ability to leverage the complementary strengths of both channel attention and spatial attention, enhancing the overall effectiveness across each dataset. Furthermore, the superior performance of our channel attention component can be attributed to the substantial amount of data, which facilitates the learning of spatial filters for each channel. In contrast, our spatial attention component exclusively focuses on learning spatial recalibration with a small amount of data. This performance advantage diminishes to some extent as the dataset size substantially increases, which affects the learning of spatial filters.

Table 9
Top-1 Accuracy (%) on ImageNet Datasets. FReCSA† and FReCSA‡ represent our spatial and channel attention, respectively.

Method	Training dataset size per class				
	40	80	160	320	Full
FReCSA†	27.98	43.40	56.40	66.03	76.63
FReCSA‡	28.94	44.40	57.08	66.74	77.49
FReCSA	**30.33**	**45.00**	**58.10**	**67.27**	**77.51**

Table 10 shows the top-1 accuracy of individual components as well as the entire FReCSA module on five additional fine-tuning datasets. Despite achieving a lower accuracy of 88.05% on the Food dataset,

Table 10
Top-1 Accuracy (%) on Food-101, Oxford-IIIT Pet, Caltech-256, SUN397, MINC Datasets. FReCSA† and FReCSA‡ represent our spatial and channel attention, respectively.

Method	Food	Pet	Caltech	SUN	MINC
FReCSA†	88.05	93.68	84.34	62.44	80.68
FReCSA‡	88.87	93.57	83.70	62.58	80.28
FReCSA	**89.09**	**94.00**	**84.42**	**63.03**	**80.94**

Table 11
Top-1 accuracy (%) of FReCSA using different variables (bias, filter size, filter interaction, and spatial interaction. ImageNet-40 is used in this experiment.

Bias	Acc.	Filter size	Acc.	Interaction	Acc.	Filter type	Acc.
0	30.10	3 × 3	30.09	Local–global	27.16	Low-pass	27.89
0.5	30.33	5 × 5	30.20	Local-only	30.12	Identity	28.85
1	29.13	7 × 7	30.33	Local-local	30.33	Depthwise	29.19
						High-pass	30.33

which contains 75,750 training images, and a slightly lower accuracy of 62.44% on the Sun dataset, our spatial attention component demonstrates impressive performance, with accuracy scores of 93.68%, 84.34%, and 80.68% on the Pet, Caltech, and MINC datasets, respectively. For comparison, while achieving a high accuracy of 88.87% on the Food dataset and slightly better accuracy of 62.58% on the SUN dataset, our channel attention component attains lower accuracy scores of 93.57%, 83.70%, and 80.28% on the remaining datasets, respectively, compared to our spatial attention component. The results of our FReCS module demonstrate the advantage of combining channel attention and spatial attention, leveraging their complementary strengths to consistently outperform each component across these datasets. Furthermore, results highlight the challenges of relying solely on one type of attention (e.g., channel attention or spatial attention) for performance improvement across different datasets.

Table 11 reports the performance of our FReCSA module on the ImageNet-40 dataset with different bias values, filter sizes, filter types, and spatial interactions in the spatial attention component. The spatial interactions include local–global (e.g., the SGE module), local-only (e.g., the GE module), and local-local (e.g., the SimAM module).

Bias. Different bias values of 0, 0.5, and 1 are studied. The accuracy at a bias setting of 0, with a value of 30.10%, serves as the baseline performance. The results indicate that an excessively high bias value of 1 may result in a degraded accuracy of 29.13%. This decline is likely attributed to reduced contrast, as all values shift significantly toward the upper bound imposed by the Sigmoid function. Conversely, a median bias value of 0.5 provides contrast benefits, resulting in an improved accuracy of 30.33%. Consequently, we set the default bias value to 0.5. Specifically, we apply a filter size of 7 × 7 to achieve a large receptive field in this exploration, following CBAM.

Filter Size. In addition to the large filter size of 7 × 7, which serves as the baseline performance, we also explore the effect of other common filter sizes. Results demonstrate that decreasing the filter size to 5 × 5 leads to no additional benefits but a slight degradation of performance, with an accuracy of 30.20%. Further reducing to a filter size of 3 × 3 results in an even lower accuracy of 30.09%. Our observations indicate that a large filter corresponding to a large receptive field is favorable. Therefore, we maintain the default value of filter size as 7 × 7.

Filter Type. Besides the predefined high-pass filter, which retains rapid changes within the neighborhood, we explore three additional filter types: the low-pass filter counterpart, which preserves slow-varying components; the identity filter, which outputs the same input; and the learnable depthwise filter. We observe that the predefined high-pass filter yields the highest accuracy while switching to its low-pass filter counterpart results in an 8.05% performance degradation due to the blurring operation. The identity filter mitigates the performance degradation to 4.89%, while using the depthwise learnable filters further alleviates the performance degradation to only 3.74%. Still, it is not as effective as the predefined high-pass filter, which highlights the difficulty in learning such a filter.

Spatial Interaction. Correspondingly, the local–global interaction in our FReCSA module is implemented by multiplying the high-pass filter result with the global average pooling value. The local-only interaction is simply the high-pass filter result, and local-local interaction is achieved by multiplying the high-pass filter result with the original value. Results indicate that local-local interaction attains the highest accuracy, while local–global interaction results in a significant 10.46% performance degradation due to global smoothing. Meanwhile, local-only interaction lags slightly behind by 0.69% due to the lack of the original intensity. Therefore, we opt for local-local interaction in our FReCSA module.

Components Contribution. Table 12 investigates the contribution of components within our proposed FReCSA module. Our observation reveals that the best accuracy is achieved when all components are selected. For our channel attention, the initial step of Global Average Pooling (GAP) results in an accuracy of only 21.10%. Further combining Batch Normalization and channel scaling achieves the optimal performance of 28.94%, demonstrating their complementary power, whereas incorporating them individually falls behind. When introducing our spatial attention as enhancement, the initial step of spatial interaction only improves performance to 29.59%. Subsequently incorporating ReLU activation boosts performance to 29.74%, while the integration of Batch Normalization or a bias term alone shows no additional benefits. For the remaining component combinations, ReLU activation consistently exhibits superior performance when compared to combinations without it. Using Batch Normalization instead of a bias term produces better results, and combining both maximizes the benefits. Therefore, we opt to incorporate all components into our FReCSA module.

5. Conclusion

In this paper, we present a frequency regulated channel-spatial attention module to address the learning challenges associated with limited data availability for image classification. In many real-world applications, annotated training datasets are usually small or moderate in size, which may not match the scale of the ImageNet dataset for training large deep networks. Our FReCSA module combines simplified channel attention with frequency-modulated spatial attention to harness their complementary power for efficiently learning from small or moderate datasets.

Table 12
Top-1 accuracy on the ImageNet-40 Dataset for components contribution within Our FReCSA module.

	Global Ave. Pool	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Channel	Batch Norm.		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Attention	Channel scaling			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Spatial Inter.					✓	✓	✓	✓	✓	✓	✓	✓
Spatial	Batch Norm.						✓		✓			✓	✓
Attention	Fixed bias							✓	✓		✓		✓
	ReLU									✓	✓	✓	✓
Accuracy (%)		21.10	19.34	28.34	28.94	29.59	29.41	29.13	29.86	29.74	29.97	30.10	**30.33**

Our experimental results demonstrate that incorporating the FReCSA module into a deep network enhances the network performance with simplified channel connection design and prior knowledge via predefined filtering. The improvement is more significant when the training dataset is small. The improvement rate of top-1 accuracy on the Imagenet-40 dataset is 10.13% over the second-best. Despite an extra module being added to the existing networks, the number of parameters and computational complexity induced by the FReCSA module increase very little in terms of model size and computational operations. Our investigation highlights the individual contributions of the predefined high-pass filter, featuring a large filter size and a median bias value, as well as the local-local interaction, to performance improvement. Combining all components within our FReCSA module yields the best results.

Our findings underscore the challenges of learning long-range dependencies or integrating additional branches, particularly with limited data. While spatial attention complements channel attention, the introduction of additional frequencies into channel attention necessitates a substantial amount of data to fully realize its advantages.

CRediT authorship contribution statement

Chengyuan Zhuang: Concept, Software development, Experiment, Writing manuscript. **Xiaohui Yuan:** Concept, Experiment, Validation, Writing manuscript, Resource. **Lichuan Gu:** Validation, Writing manuscript. **Zhenchun Wei:** Validation, Writing manuscript. **Yuqi Fan:** Validation, Writing manuscript. **Xuan Guo:** Resource.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A. S., Al-dabbagh, B. S. N., et al. (2023). A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, *10*, 46.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Bell, S., Upchurch, P., Snavely, N., & Bala, K. (2015). Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3479–3487).

Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101–mining discriminative components with random forests. In *European conference on computer vision* (pp. 446–461). Springer.

Brigato, L., Barz, B., Iocchi, L., & Denzler, J. (2022). Image classification with small datasets: Overview and benchmark. *IEEE Access*, *10*, 49233–49250.

Cao, Y., Xu, J., Lin, S., Wei, F., & Hu, H. (2019). Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*.

Chen, Y., Fan, H., Xu, B., Yan, Z., Kalantidis, Y., Rohrbach, M., et al. (2019). Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE international conference on computer vision* (pp. 3435–3444).

Cheng, B., Xiao, R., Wang, J., Huang, T., & Zhang, L. (2020). High frequency residual learning for multi-scale image classification. In *30th British machine vision conference*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Gao, Z., Xie, J., Wang, Q., & Li, P. (2019). Global second-order pooling convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3024–3033).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset*. California Institute of Technology.

Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., et al. (2022). Attention mechanisms in computer vision: A survey. *Computational Visual Media*, *8*, 331–368.

Hassanin, M., Anwar, S., Radwan, I., Khan, F. S., & Mian, A. (2024). Visual attention methods in deep learning: An in-depth survey. *Information Fusion*, *108*, Article 102417.

He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., et al. (2022). Is synthetic data from generative models ready for image recognition? In *The eleventh international conference on learning representations*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hinton, G. E., Krizhevsky, A., & Sutskever, I. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1106–1114.

Hu, J., Shen, L., Albanie, S., Sun, G., & Vedaldi, A. (2018). Gather-excite: Exploiting feature context in convolutional neural networks. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 9423–9433).

Hu, J., Shen, L., & Sun, G. (2018b). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., et al. (2019). Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in Neural Information Processing Systems*, *32*.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019). Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision* (pp. 603–612).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Vol. 37*, In *Proceedings of machine learning research* (pp. 448–456).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Lee, H., Kim, H.-E., & Nam, H. (2019). Srm: A style-based recalibration module for convolutional neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1854–1862).

Li, Y., Li, X., & Yang, J. (2022). Spatial group-wise enhance: Enhancing semantic feature learning in cnn. In *Proceedings of the Asian conference on computer vision* (pp. 687–702).

Li, X., Wang, W., Hu, X., & Yang, J. (2019). Selective kernel networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 510–519).

Liu, P., Zhang, H., Lian, W., & Zuo, W. (2019). Multi-level wavelet convolutional neural networks. *IEEE Access*, *7*, 74973–74985.

Ma, Y., Luo, Y., & Yang, Z. (2020). Pcfnet: Deep neural network with predefined convolutional filters. *Neurocomputing*, *382*, 32–39.

Man, K., & Chahl, J. (2022). A review of synthetic image data and its use in computer vision. *Journal of Imaging*, *8*, 310.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning* (pp. 807–814).

Oyallon, E., Belilovsky, E., & Zagoruyko, S. (2017). Scaling the scattering transform: Deep hybrid networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 5618–5627).

Park, J., Woo, S., Lee, J.-Y., & Kweon, I.-S. (2018). Bam: Bottleneck attention module. In *British machine vision conference*. British Machine Vision Association (BMVA).

Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3498–3505). IEEE.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8026–8037.

Qin, Z., Zhang, P., Wu, F., & Li, X. (2021). Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 783–792).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd international conference on learning representations*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tsirikoglou, A., Eilertsen, G., & Unger, J. (2020). A survey of image synthesis methods for visual machine learning. *Vol. 39*, In *Computer graphics forum* (pp. 426–451). Wiley Online Library.

Ulicny, M., Krylov, V. A., & Dahyot, R. (2019). Harmonic networks with limited training samples. In *2019 27th European signal processing conference* (pp. 1–5). IEEE.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803).

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11534–11542).

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision* (pp. 3–19).

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 3485–3492). IEEE.

Yang, L., Zhang, R.-Y., Li, L., & Xie, X. (2021). Simam: A simple, parameter-free attention module for convolutional neural networks. In *International conference on machine learning* (pp. 11863–11874). PMLR.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British machine vision conference*.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.

Zhang, R. (2019). Making convolutional networks shift-invariant again. In *International conference on machine learning* (pp. 7324–7334).

Zhang, H., Zu, K., Lu, J., Zou, Y., & Meng, D. (2022). Epsanet: An efficient pyramid squeeze attention block on convolutional neural network. In *Proceedings of the Asian conference on computer vision* (pp. 1161–1177).

Zhuang, C., Yuan, X., Guo, X., Wei, Z., Xu, J., & Fan, Y. (2023). Improved convolutional neural networks by integrating high-frequency information for image classification. In *Proceedings of the 2023 2nd Asia conference on algorithms, computing and machine learning* (pp. 429–434).

Zou, X., Xiao, F., Yu, Z., & Lee, Y. J. (2020). Delving deeper into anti-aliasing in ConvNets. In *BMVC*.