IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 September 2023, accepted 25 September 2023, date of publication 28 September 2023,
date of current version 4 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320642

RESEARCH ARTICLE

Frequency Regularization: Reducing Information Redundancy in Convolutional Neural Networks

CHENQIU ZHAO, GUANFANG DONG, SHUPEI ZHANG, ZIJIE TAN, AND ANUP BASU
Department of Computing Science, University of Alberta, Edmonton, AB T6G 2R3, Canada
Corresponding author: Chenqiu Zhao (zhaochenqiu@gmail.com)

ABSTRACT Convolutional neural networks have demonstrated impressive results in many computer vision tasks. However, the increasing size of these networks raises concerns about the information overload resulting from the large number of network parameters. In this paper, we propose Frequency Regularization to restrict the non-zero elements of the network parameters in the frequency domain. The proposed approach operates at the tensor level, and can be applied to almost all network architectures. Specifically, the tensors of parameters are maintained in the frequency domain, where high-frequency components can be eliminated by zigzag setting tensor elements to zero. Then, the inverse discrete cosine transform (IDCT) is used to reconstruct the spatial tensors for matrix operations during network training. Since high-frequency components of images are known to be less critical, a large proportion of these parameters can be set to zero when networks are trained with the proposed frequency regularization. Comprehensive evaluations on various state-of-the-art network architectures, including LeNet, Alexnet, VGG, Resnet, ViT, UNet, GAN, and VAE, demonstrate the effectiveness of the proposed frequency regularization. For a very small accuracy decrease (less than 2%), a LeNet5 with 0.4M parameters can be represented by only 776 float16 numbers (over 1100× reduction), and a UNet with 34M parameters can be represented by only 759 float16 numbers (over 80000× reduction). In particular, the original size of the UNet model is reduced from 366 Mb to 4.5 Kb.

INDEX TERMS Frequency domain, information redundancy, network regularization, convolutional neural network.

I. INTRODUCTION

Convolutional neural networks have become increasingly popular in computer vision applications, such as image classification, image segmentation, and so on. However, as the learning ability of these networks has improved, so has their size, growing from a few megabytes to hundreds of gigabytes [1]. This leads to challenges such as enormous storage space or long transmission time on the Internet. After conducting a thorough literature review, we found that although larger networks tend to perform better, the accuracy improvement is not always directly proportional to the network size. In some cases, doubling the network size may not result in a significant accuracy improvement. From this observation we conclude that there may be information redundancy within various network architectures, leading to the question: "How can we reduce network information redundancy?"

It is commonly accepted that the performance of networks comes from the features learned by their parameters. For example, when a convolutional neural network such as Alexnet [2] is applied to an image classification task, the features learned by its convolutional kernels are closely related to the training images. Under this condition, features learned by a network are expected to have properties similar to the training images in which the high-frequency components are known to be less important. Thus, it is reasonable to apply the frequency domain transforms, such as the discrete cosine transform, to the network parameters. Unfortunately, given the poor interpretability of complex network architectures, even a small change in a few key parameters can significantly affect the network performance. Based on this insight, the potential of frequency domain transforms for compressing or pruning networks may not be well-developed on pre-trained

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

models, which has been the focus of previous approaches [3], [4], [5], [6]. Instead, we focus on using the frequency domain transform for network regularization to restrict information redundancy during the training process and introduce Frequency Regularization (FR) as shown in Fig. 1.

The proposed frequency regularization can be divided into two steps: dynamic tail-truncation and inverse discrete cosine transform (IDCT). During network training, parameter tensors are maintained in the frequency domain, with the tail parts representing high-frequency information zigzag truncated. The truncation process is implemented through a dot product with a zigzag mask matrix to ensure differentiability. After dynamic tail-truncation, tensors are input into the IDCT to reconstruct the spatial tensors that are then used as regular learning kernels in networks. Since the IDCT is a differentiable process, the actual tensors in the frequency domain can be updated through backpropagation algorithms. Moreover, as the reconstructed spatial tensors have the same size as those maintained in the frequency domain, the proposed frequency regularization can be easily applied to almost any type of network architecture. Furthermore, as features related to computer vision tasks are closely correlated to images, many parameters (from 90% to 99.99%) in the frequency domain can be set to 0 without an obvious decrease in network performance. The proposed frequency regularization has a few desirable properties:

- Generality: The proposed frequency regularization can be applied to almost any type of network architecture, as it is designed for tensors. Given this, we are able to evaluate the proposed frequency regularization on various state-of-the-art network architectures including LeNet, AlexNet, VGG, ResNet, ViT, UNet, GAN and VAE.
- Effectiveness: A large number of parameters can be truncated in the networks trained with the proposed frequency regularization, since it is widely recognized that high-frequency information is unimportant for image data and the features learned by networks are closely correlated to images.
- Lossless property: There is no need to worry that the proposed frequency regularization will decrease network performance, since the inverse discrete cosine transform (IDCT) is usually considered as a lossless transformation. When no parameter is truncated, the proposed frequency regularization has almost no effect on the network performance.

II. RELATED WORK

Frequency information has been widely used in convolutional neural networks for improving performance [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] pruning and network compression [4], [5], [6], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], or increasing the detection accuracy [31], [32], [33]. For instance, Wang et al. [18] represented object edges and smooth structures using high and low-frequency information, respectively.

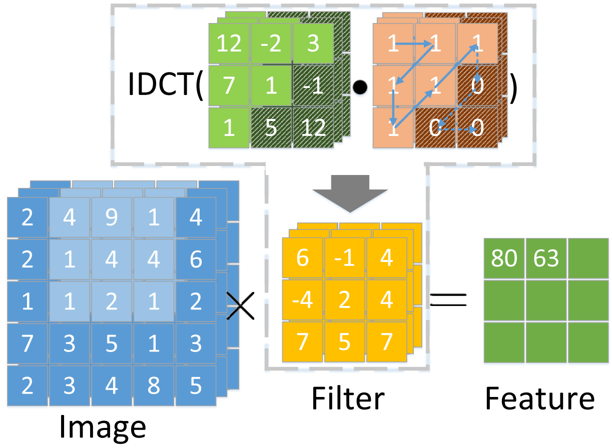

FIGURE 1. Illustration of the proposed frequency regularization. The tail elements of tensors in the frequency domain are truncated first, then input into the inverse of the discrete cosine transform to reconstruct the spatial tensor for learning features.

Mi et al. [17] split channel recognition networks into frequency domains, while Rippel et al. [34] proposed a fully spectral representation of network parameters. Buckler et al. [35] introduced a similar idea with learnable parameters to determine the importance of different channels. Besides this, Han et al. [36] compressed the networks with pruning, quantization, and Huffman coding to achieve excellent results for network compression. Similarly, Wang et al. [6], [5] combined DCT, k-means, quantization, and Huffman coding to compress the network parameters. Chen et al. [37] utilized DCT and hashing to assign high-frequency to less hash buckets to achieve compression. Wang et al. [1] compressed the vision transformer by removing the low-frequency components. Additionally, several works have proposed using the frequency representation of convolution kernels to prune less important channels [3], [38], [39]. Most of these previous works have primarily focused on the frequency domain representation of the input image, feature maps, or network parameters, but ignored the learning ability of the network parameters during the training process. Since the network performance is usually sensitive to changes in a few key parameters, it cannot be guaranteed that these key parameters are located in the low-frequency domain. As a result, these methods have only dropped 30% to 95% of the parameters with minimal accuracy loss. In particular, although a few methods [6], [36] claimed around 50× compression rate, the rate is improved by quantization and entropy coding. The actual proportion of dropped parameters still ranged from 30% to 95%. In contrast, the proposed frequency regularization approach restricts network parameters to the frequency domain, and high-frequency elements are dynamically truncated during network training. This allows us to have much higher drop rates of 90% to 99.99%.

The proposed frequency regularization approach is somewhat related to the DCT-Conv method proposed in [40], as both approaches maintain parameters in the frequency domain and utilize IDCT. However, there are several key differences between the two methods. First, the proposed frequency regularization can be applied to any layer with learnable parameters, including but not limited to convolution layers, fully connected layers, and transposed convolution layers. Therefore, we are able to evaluate the proposed frequency regularization on various network architectures including Alexnet, VGG, Resnet, ViT, UNet, GAN, and VAE. In contrast, DCT-Conv is a convolution layer with DCT, which is only evaluated on VGG and Resnet. Second, DCT-Conv randomly drops parameters in the frequency domain, which is not ideal since the components in low-frequency and high-frequency are considered equally important. The ability to drop parameters of DCT-Conv is thus limited. A similar limitation is also shown in the BA-FDNP method [3] in which the coefficients in the frequency domain are used for pruning. Instead, in our approach, we truncate the tail parts of the parameters, as the high-frequency components have been shown to be non-critical for features related to images. This is the most important difference between the proposed approach and previous methods in [3], [6], [38], [39], and [40], and also enables that the proposed approach to achieve promising compression rates on various of network architectures.

III. FREQUENCY REGULARIZATION AND TAIL-TRUNCATION

A. METHODOLOGY

When convolutional neural networks are applied to computer vision applications, the features learned are closely related to the training images. In fact, as shown by the visualized filters in AlexNet [2], the learned features actually look like real images. Based on this insight, it is reasonable to assume that the low-frequency components of network parameters are more important than the high-frequency components. Unfortunately, because of the poor interpretability of convolutional neural networks, a network's performance may be highly sensitive to changes in the values of a few key parameters, which are not always located in the low-frequency domain. This limits the proportion of dropped parameters in pre-trained models. Thus, we focus on restricting the number of non-zero parameters during the training process, and propose frequency regularization. Note that network regularizations are typically introduced to prevent overfitting in neural networks [41]. In this paper, however, the definition of regularization is extended to include modifications or restrictions applied to networks with a particular purpose, such as restricting the number of non-zero parameters.

The idea behind the proposed frequency regularization is quite straightforward. Instead of maintaining the tensors of network parameters in the spatial domain, they are maintained in the frequency domain. This allows the tail elements of tensors to be truncated by a dot product with a zigzag mask matrix. During network training, the frequency tensors are input into the inverse discrete cosine transform (IDCT) to reconstruct the spatial tensors. Then, these spatial tensors are used as the regular learning kernels of convolutional neural networks for learning features. Since the IDCT process is differentiable, the parameters maintained in the frequency domain can be correctly updated during backpropagation. Moreover, as the IDCT can be used for tensors with any dimension without changing their size, the proposed frequency regularization can be implemented for any layer involving tensor operations, including convolution layers, fully connected layers, and transposed convolution layers. We first introduce the proposed frequency regularization for 1D tensors, and then discuss the implementation of regularization in higher dimensions.

Assume the 1D tensors of learning kernels to be $T(x) \in \mathbb{R}^{1\times D}$ where x is the index. $T(x)$ represents the learnable parameters in the frequency domain which have been updated during training. $T(x)$ is first computed with a zigzag mask $\mathbb{1}_{x<\epsilon}(x)$ for tail-truncation, then input into the IDCT to reconstruct the spatial tensor $W(y) \in \mathbb{R}^{1\times D}$ which can be the regular learning kernel in a 1D convolution layer. Mathematically:

$$\begin{aligned} W(y) = \mathcal{F}_\epsilon(T(x)) &= \mathrm{IDCT}\left(\bigcup_{x\in G_x} T(x)\cdot \mathbb{1}_{x<\epsilon}(x)\right) \\ &= \sum_{x\in G_x} T(x)\cdot \mathbb{1}_{x<\epsilon}(x)\cdot \cos\left[\frac{\pi}{D}\left(y+\frac{1}{2}\right)x\right] \\ &= \sum_{x<\epsilon} T(x)\cdot \cos\left[\frac{\pi}{D}\left(y+\frac{1}{2}\right)x\right], \end{aligned} \tag{1}$$

where $x, y \in [0\ D{-}1] \cap \mathbb{N} = G_x$ are the indices of tensor $T(x)$ and $W(y)$ respectively. $T(X)$ is the actual tensor in the frequency domain and $W(y)$ is the corresponding reconstructed tensor in the spatial domain. IDCT$(\cdot)$ is the inverse discrete cosine transform. During the implementation, DCT-III which is the inverse of the widely used DCT-II is used. ϵ is the threshold value to control the truncation ratio. $\mathbb{1}_{x<\epsilon}(x)$ is a binary mask to make the truncation process differentiable. Note that we removed the $\frac{T(0)}{2}$ in the above DCT-III since the learnable parameters can adaptively adjust this constant component.

Since the high dimensional IDCT can be decomposed into several 1D IDCTs, so high dimensional frequency regularization can also be expressed by several 1D frequency regularizations. Assume the two N-dimensional tensors in the frequency domain and spatial domain to be $T(\vec{x}), W(\vec{y}) \in \mathbb{R}^{D_1\times D_2,\cdots,D_N}$ where $\vec{x} = \{x_1, x_2, \cdots, x_N\}$ and $\vec{y} = \{y_1, y_2, \cdots, y_N\}$ are index vectors, $x_i, y_i \in [0, D_i{-}1] \cap \mathbb{N}$. Then, high dimensional frequency regularization $\mathcal{F}_\epsilon^N$ is:

$$\begin{aligned} W(\vec{y}) = \mathcal{F}_\epsilon^N(T(\vec{x})) &= \mathrm{IDCT}^N\left(\bigcup_{\vec{x}\in G_{\vec{x}}} T(\vec{x})\cdot \mathbb{1}_{|\vec{x}|_1<\epsilon}(\vec{x})\right) \\ &= \sum_{\vec{x}\in G_{\vec{x}}} T(\vec{x})\cdot \mathbb{1}_{|\vec{x}|_1<\epsilon}(\vec{x}) \prod_{i=1}^{N} \cos\left[\frac{\pi}{D_i}\left(y_i+\frac{1}{2}\right)x_i\right] \end{aligned}$$

$$= \sum_{|\vec{x}|_1 < \epsilon} T(\vec{x}) \prod_{i=1}^{N} \cos\left[\frac{\pi}{D_i}\left(y_i + \frac{1}{2}\right)x_i\right] \quad (2)$$

where IDCT^N is the N-dimensional inverse discrete cosine transform, which can be easily implemented by N IDCTs in different dimensions. ϵ is the threshold value to control the truncation ratio. $|\vec{x}|_1 = \sum_{x=1}^{N} |x_i|$ is the L_1 norm of the index vector $|\vec{x}|$. The indicator function $\mathbb{1}_{|\vec{x}|_1<\epsilon}(\vec{x})$ is used to approximate the zigzag binary mask for truncating parameters.

Frequency regularization is proposed for tensors. Thus, it can be applied to any layer of a convolutional neural network, such as the convolution layer or fully connected layer. After applying the proposed frequency regularization, the formula of the convolution process can be re-expressed as:

$$Z = WX + B \Rightarrow Z = \mathcal{F}_{\epsilon}^{N}(T_W)X + \mathcal{F}_{\epsilon}(T_B), \quad (3)$$

where T_W and T_B are the tensors representing the weight and bias of the convolution layer. Usually, the size of the tensor in the convolution layer is 4D, representing the number of kernels, input channel, and kernel size. Based on the requirements of different applications, the proposed frequency can be 1D, 2D, 3D, or 4D which can be controlled by users. In particular, 4D frequency regularization gives us the highest compression rate, which has been used in the proposed evaluation experiments.

B. DYNAMIC TAIL-TRUNCATION:

According to the evaluation experiments we propose in Section IV, usually over 99% of parameters in convolutional neural networks can be truncated without an obvious decrease in accuracy. However, there is also a serious issue that the remaining parameters may not have suitable gradients for backpropagation. In this condition, the training loss of the network restricted by the proposed frequency regularization sometimes may not change for hundreds of training epochs. This issue has occurred many times in our evaluation experiments. To address this problem, we propose the dynamic tail-truncation strategy. Instead of directly setting over 99% of the parameters to 0, this strategy continuously sets a few tail elements to 0 in every training epoch. In particular, as the total number of truncated parameters increases, the number of parameters truncated in each training epoch decreases. Mathematically, the ratio of truncated parameters is controlled by the following function:

$$\beta_n = \beta_{n-1} - \gamma(\beta_{n-1} - \epsilon)$$

where n is the index of training epochs. $\beta_{n-1}, \beta_n \in [0, 1]$ is the ratio of non-zeros parameters in epoch $n-1$ and n. γ is the user parameter to control the speed of truncating parameters. During our evaluation experiment, $\gamma = 0.01$ is used. ϵ is the user parameter to control the minimum ratio of non-zero parameters in the network. By changing the value of ϵ, we can control the percentage of parameters that will be truncated in a network. For example, $\epsilon = 0.01$ means that around $1 - \epsilon = 99\%$ of parameters will be truncated. Although the dynamic tail-truncation strategy requires extra training epochs, it results in a more stable training result when the minimum ratio ϵ is very small. This strategy has been used for all the evaluation experiments in this paper.

C. IMPLEMENTATION DETAILS:

The proposed approach is implemented in PyTorch, the source will be available at https://github.com/guanfangdong/pytorch-frequency-regularization.git. Since the proposed approach is devised at the tensor level, it can be easily implemented for different network layers including but not limited to linear layer, convolution, and transposed convolution in 1d, 2d, and 3d. We pack our implementation into a PyTorch Module, so it can be used as regular PyTorch layers. Please check our source code for more details. It is also because of the same reason, we are able to evaluate the proposed approach on diverse network architectures.

IV. EXPERIMENTS

In this section, we evaluate the proposed frequency regularization on several classical state-of-the-art network architectures including LeNet, Alexnet, VGG, ResNet, ViT, UNet, GAN and VAE on several standard datasets. During the evaluation, we first capture the accuracy of the original networks which are used as the reference accuracy to compare with the ones of networks restricted by the proposed frequency regularization. In particular, the implementation provided by authors or popular Github repositories are used. Then, these networks are re-trained with the proposed frequency regularization to compare the with original networks. Since the parameters of networks are zigzag truncated in the proposed frequency regularization, we only need to save the location of the boundary between non-zero elements and zero elements as well as the size of tensors. Thus, the compression rate can be easily computed by dividing the number of non-zero parameters by the total number of parameters. Note that we did not consider the bias in the convolution layers, fully connected layers and transposed convolution layers during our evaluation, since the bias only represents a small portion of network parameters and most of the previous researchers have also ignored them. Similarly, the parameters in batchnorm layers are also ignored for the same reason. Given the page limitation, we only demonstrate the total number of non-zero parameters in networks. More details are given in the supplementary material. All the experiments are performed on GTX A4000 with 16 GB of video memory.

A. COMPARISON WITH STATE-OF-THE-ART METHODS

To the best of our knowledge, there is no work that is directly related to the proposed approach on restricting the information redundancy during network training. The closest work we could find was proposed for network compression by pruning, such as DeepCompress [36], DynSurgery [42] or BA-FDNP [3]. Although there are a few newer methods

TABLE 1. Comparison between the proposed approach and state-of-the-art methods including DeepCompress [36], DynSurgery [42], BA-FDNP [3] on the MNIST dataset [43].

	DeepCompress [36]			DynSurgery [42] and BA-FDNP [3]			The Proposed Approach		
	Top-1 Accuracy	Compression Rate	Number of Parameters	Top-1 Accuracy	Compression Rate	Number of Parameters	Top-1 Accuracy	Compression Rate	Number of Parameters
LeNet300-ref	98.36%	100%(1×)	266K	97.72%	100%(1×)	266K	98.15%	100%(1×)	266K
LeNet300-v1	98.42%	2.5%(40×)	21k(float6-8)	98.01 %	1.8%(56×)	4.8k	97.11%	0.88%(114×)	4238(float16)
LeNet300-v2							95.69%	0.53%(188×)	2816(float16)
LeNet5-ref	99.20%	100%(1×)	431K	99.09%	100%(1×)	431K	98.78%	100%(1×)	431K
LeNet5-v1	99.26%	2.56%(39×)	34K(float6-8)	99.09%	0.92%(108×)	4.0k	98.35%	0.89%(112×)	7668(float16)
				BA-FDNP [3]			The proposed approach		
LeNet5-v2				99.08%	0.67%(150×)	2.8K	97.52%	0.09%(1110×)	776(float16)

TABLE 2. Evaluation of the proposed frequency regularization on Alexnet [2], VGG [44], ResNet [45] and ViT [1] using CIFAR10 dataset [46].

	Top-1 Accuracy	Compression Rate	Number of Parameters
AlexNet-ref	76.45 %	100%(1×)	57,035,456
AlexNet-v1	77.44 %	1%(100×)	570,365
AlexNet-v2	70.46 %	0.1%(1000×)	57,364
AlexNet-v3	59.22 %	0.0025%(40509×)	1,408
AlexNet-v4	58.55 %	0.00123%(81018×)	1,408(float16)
VGG16-ref	85.84 %	100%(1×)	20,024,000
VGG16-v1	84.77 %	1%(100×)	200,578
VGG16-v2	73.01 %	0.1%(987×)	20,302
VGG16-v3	65.99 %	0.0102%(9775×)	2,048
ResNet18-ref	85.43 %	100%(1×)	11,162,624
ResNet18-v1	85.34 %	1%(100×)	111,901
ResNet18-v2	83.23 %	0.1%(1000×)	12,058
ResNet18-v3	77.64 %	0.024%(4156×)	2,688
ViT-ref	78.94 %	100%(1×)	9,500,672
ViT-v1	78.98 %	9.77%(10×)	928,478
ViT-v2	75.05 %	0.83%(120×)	157,848(float16)

proposed in [3], [6], [26], [27], [30], [40], [47], and [48], their advantage on compression rate is not very obvious and almost all of them work on networks pre-trained on large datasets. which makes the comparisons become very expensive considering computational resources. Besides, among all these previous approaches, BA-FDNP [3] claimed the highest compression rate which is 150×. Thus, we compare the proposed approach with DeepCompress [36], DynSurgery [42], and BA-FDNP [3]. The comparison results are shown in Table 1. In particular, the top-1 accuracy [2] is used as the evaluation metric. MNIST dataset is used for evaluation. In the MNIST dataset, 60000 images are used for training, and another 6000 images are used for testing. As shown in Table 1, since previous approaches are based on pre-trained models, they have a 1-2% advantage in top-1 accuracy. Actually, due to the limitation of our computational resources, we are unable to achieve over 99% of top-1 accuracy even without dropping any parameters. However, the top-1 accuracy achieved by our model is very close to the pre-trained model. There is less than a 1% difference in accuracy. Therefore, we trained our own LeNet300-ref and LeNet5-ref for reader references.

The proposed approach achieves a much higher compression rate without an obvious decrease in top-1 accuracy. For example, in LeNet300-v1, the proposed approach achieves 112× the compression rate with less than 1% of cost in top1-accuracy. Furthermore, it achieves 1110 × compression rate in LeNet5-v2 with only 2% of top1-accuracy decrease, which is much higher than DeepCompress [36], DynSurgery [42], BA-FDNP [3] as well as strategies in [3], [6], [26], [27], [30], [40], [47], and [48]. Note that the BA-FDNP [3] applied quantization, entropy coding, and pruning in the frequency domain which are related to the proposed approach. However, as we mentioned in Section II, instead of truncating the tail parts of the tensor, BA-FDNP [3] utilized the coefficient matrix for pruning which limits its compression rate. As a result, BA-FDNP only has 150× compression rate on LeNet5-v2, but the proposed approach can achieve around 1110 × reduction with less than 2% of top-1 accuracy decrease. Note that BA-FDNP [3] utilized data argumentation to improve top-1 accuracy. It also applied the pre-trained model for initialization and retrained their model for 20k iterations for searching the highest top-1 accuracy model. In contrast, none of these techniques for accuracy improvement are applied in the proposed approach, considering limitations on computational resources. Consequently, the proposed approach has no advantage in top1-accuracy. However, compared with these previous approaches, the compression rate of the proposed approach is quite promising. Besides, since the proposed approach is devised at the tensor level, it can be applied to almost any network architecture. In contrast, since previous approaches are devised based on specific pre-trained models, they were not evaluated on diverse network architectures. In order to demonstrate the generality of the proposed approach, we also apply the proposed frequency regularization on various state-of-the-art network architectures in the remaining sections.

B. IMAGE CLASSIFICATION

Image classification has been studied for decades in computer vision. There are many excellent architectures that have been proposed in this field. Thus, we evaluate the performance of the proposed frequency regularization on Alexnet [2], VGG16 [44], ResNet18 [45] and ViT [1]. During the evaluation, the cifar10 dataset [46] is used for training and testing and top1-accuracy is used as the evaluation metric. In the cifar10 dataset, 50000 images are used for training and another 10000 are used for testing. Note that we did not apply tuning techniques or pre-trained models during the evaluation considering training time. The accuracy scores can be higher once some training techniques such as dynamic learning rate or data argumentation are utilized, which has been widely used in previous approaches [3], [6], [26], [27], [30], [40], [47], [48].

The evaluation results are shown in Table 2. AlexNet-ref which is the original Alexnet [2] achieves 76.45% top-1 accuracy. After 99% of the parameters have been truncated, AlexNet-v1 achieves 77.44% top-1 accuracy which is a little bit higher than the original one. This improvement comes from the fact that the tail-truncation can be considered as a regularization to prevent a network from overfitting just like the dropout layer. When only 0.1% of parameters are kept in the frequency domain, AlexNet-v2 has 70.46% in top-1 accuracy, and the compression rate becomes 1000×. Furthermore, in the extreme case where only 1408 non-zero parameters are kept in AlexNet, AlexNet-v3 still achieves 59.22%. We also applied the network on the half float condition, and AlexNet-v4 consisting of 1408 float16 numbers achieves 58.55% with the compression rate increasing to 81018×, which is a very interesting discovery. Similar results are also observed in VGG16 and ResNet18. In particular, ResNet18-v3 with only 2688 parameters achieves 77.64%, which is even higher than the original AlexNet-ref with 57M parameters. We also evaluate the proposed frequency regularization on ViT consisting of self-attention layers which are actually not very suitable for frequency domain transformation. Previous work [14], [30] related to transformer only achieves around 50% of pruning ratio. For the proposed approach, there is around 75% top-1 accuracy in ViT-v2 in which over 98% of parameters are truncated. But when only 90% of parameters are truncated, the ViT-v1 has no accuracy decrease, which demonstrates the generality of the proposed approach. The evaluation of the proposed approach on these networks clearly demonstrates that the information redundancy inside networks can be restricted well by the proposed frequency regularization. However, the resolution of images in CIFAR10 is not very high, which may raise a concern that the proposed frequency regularization only works well on small images. Thus, we evaluated the proposed technique on UNet for high-resolution image segmentation.

C. IMAGE SEGMENTATION

After a comprehensive literature review, we found a limited number of methods focussing on pruning or compressing segmentation networks, even DepGraph [30], which is the latest method proposed for any structural pruning, was not evaluated on segmentation networks. One reason may be the sense that the information redundancy for segmentation networks should be lower than that for image classification networks. However, the proposed frequency regularization is supposed to work well on any features related to images and segmentation networks should be one of them. We thus evaluate the proposed frequency regularization on UNet using the Carvana Image Masking Challenge Dataset [49], which is a popular dataset for segmentation challenges in Kaggle competitions. In particular, around 5000 high-resolution images are used for training and another 508 images are used for testing. The Dice Score is used for evaluation. Actually, compared to image classification networks, the UNet architecture usually consists of pure convolution layers without bias which are more suitable for the proposed frequency regularization. As shown in Table 3, the original UNet containing 31M parameters achieves a 99.31% Dice score. Once over 99% parameters have been truncated in the frequency domain, UNet-v1 achieves a 99.51 % Dice score. Moreover, UNet-v3 achieves promising results with 98.86% in the Dice score, in which only 2936 parameters are kept. Finally, we tested the proposed approach for the most extreme condition in which only 759 float16 parameters

TABLE 3. Evaluation of the proposed frequency regularization on UNet for image segmentation tasks using Carvana Image Masking Challenge Dataset [49].

	Dice Score	Compression Rate	Number of Parameters
UNet-ref	99.13 %	100%(1×)	31,043,586
UNet-v1	99.51 %	1%(100×)	310,964
UNet-v2	99.37 %	0.1%(1000×)	31,096
UNet-v3	98.86 %	0.0094%(10573×)	2,936
UNet-v4	97.19 %	0.0012%(81801×)	759(float16)

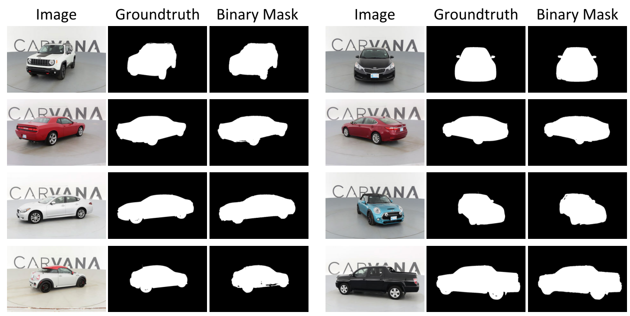

FIGURE 2. The segmentation results of UNet parameters restricted by the proposed frequency domain, with over 99.99% of parameters truncated, for the Carvana Image Masking Challenge Dataset [49].

TABLE 4. Evaluation of the proposed frequency regularization on generative adversarial network (GAN) and variational autoencoder (VAE).

	Fid value	Compression Rate	Number of Parameters
GAN-ref	244	100%(1×)	6,334,464
GAN-v1	248	1.02%(98 ×)	64,383
VAE-ref	81	100%(1×)	971,200
VAE-v1	94	10%(10×)	97,134

are kept in UNet-v4. In particular, we also disabled all the bias in the networks and learning parameters in batchnorm, which guaranteed UNet-v4 only has 759 non-zero float16 parameters. Surprisingly, the UNet-v4 still achieves 97.19% in the Dice score. This is an unbelievable result, a UNet with only 759 parameters can achieve around 97% in Dice score for the Carvana Image Masking Challenge Dataset. Note that there are over 5000 images with 959 × 460 resolution which is much higher than the resolutions of images in CIFAR10 [46]. We double-checked the non-zero parameters in every layer in UNet to make sure that this conclusion is correct, and the visualized segmentation mask is shown in Fig. 2. We also include this pre-trained UNet in the supplementary materials as well as a few testing images. *The original size of the UNet model exceeds 366MB, but our frequency regularization technique reduces it to 40kb. Additionally, using an entropy compression tool on Ubuntu, we further reduce the size to 4.5kb.*

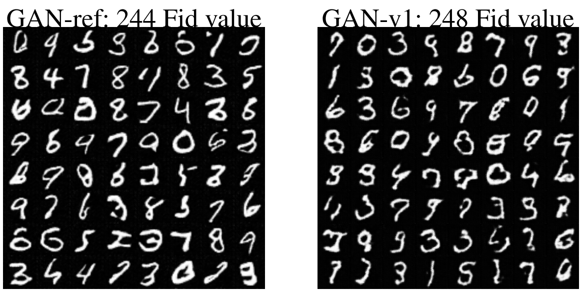

FIGURE 3. Comparisons between images generated by the orignal GAN network and GAN with our frequency regularization in which over 99% of parameters are truncated.

D. IMAGE GENERATION

Compared to classification or segmentation networks, image generation networks such as the generative adversarial network (GAN) or variational autoencoder (VAE) are not very suitable for the proposed frequency regularization, since what has been learned by these networks is claimed as distribution information. However, these distributions are still related to images. Thus, we also evaluate the proposed approach for generative adversarial networks and variational autoencoders. Since both networks usually require a long time for training, we use the MNIST dataset for evaluation and the Fid value [5] is used as the evaluation metric. As shown in Table 4, the proposed approach achieves similar results when 99% of the parameters are truncated compared to the original GAN network. The generated images are also demonstrated in Fig. 3, where the images generated by GAN with the proposed frequency are similar to the original GAN. We also evaluate the proposed approach regularization on VAE. Since the VAE we used is the simplest version in which there are only two fully connected layers in their encoder and three fully connected layers in the decoder, we only truncate 90% of the parameters for VAE. The visual results are shown in Fig. 4. During the training of VAE and GAN, with or without the proposed frequency regularization, we stop the training once the visual results look good. Actually, with more training epochs, the quality of the generated images can be better.

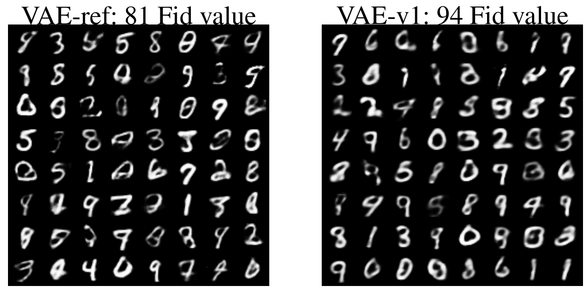

FIGURE 4. Comparisons between images generated by the orignal VAE network and VAE with our frequency regularization in which over 90% of the parameters are truncated.

E. LIMITATION AND FUTURE WORK

1) LIMITATION

The proposed frequency regularization is based on the assumption that the high-frequency component is unimportant. This may not work well on the tensors of parameters that are not related to images. This disadvantage can be seen in the comparison between evaluations on image segmentation in Section IV-C and image generation in Section IV-D, where the parameters in the generation networks cannot be truncated too much. In addition, although the parameter of the proposed approach is in the frequency domain, it still needs to be converted into the spatial domain for convolution operations. This leads to an extra memory cost during network training. However, once the networks have been well-trained, a high compression rate can be achieved, which is very useful for network transmission on the Internet. The memory cost does not exist during testing, since the networks need to be unpacked only once.

2) COMPUTATIONAL COMPLEXITY

The computational complexity of the proposed approach is a worthy concern since IDCT will cost extra computation resources. During the training process, the IDCT will be done in every kernel of every training epoch. However, such cost is not obvious since the most computationally intensive computation in convolutional neural networks happens between the data and network kernels. Consider the UNet as an example. It takes around 1s to process 1 image. In particular, the total time spent on IDCT is around 0.07s. This cost is very small because of two reasons: a) First, IDCT can be implemented by matrix operations, which is very suitable for GPU computation. This can be easily run on machines that can handle a convolutional neural network. b) In addition, since many parameters in the frequency domain are 0, it is easy to optimize. In addition, PyTorch also provides cuFFT library for GPU optimization which is utilized in our implementation.

Moreover, since the proposed approach has already been encapsulated into a PyTorch module, it can also be run in parallel with multiple GPUs. We evaluated the UNet on different GPUs, the evaluation results are shown in Table 5. The UNet takes 79s to process 508 images of dimension with 640×969 when only 1 GPU is used. Once we increase the number of GPUs to 2, the total processing time is reduced to 52s. When 3 GPUs are used, it only takes 39s for processing. However, since the processing time includes computer I/O, the UNet still takes 38s when 4 GPUs are utilized. We included this evaluation in our supplementary materials for readers to check.

TABLE 5. Running time evaluation of the proposed approach on multiple GPUs using 508 images with 640 × 959.

Number of GPUs	1 GPU	2 GPUs	3 GPUs	4 GPUs
Time in Seconds	79s	52s	39s	38s

3) FUTURE WORK

Currently, the proposed approach only allows for reducing the memory required to store models. In order to run the model during inference, it is necessary to unpack the compressed network. However, since around 99% of the parameters in networks restricted by the proposed frequency regularization can be truncated, it is very much possible to speed up the network inference. One potential solution is doing the convolution on the frequency domain directly. For example, the images can be converted into the frequency domain to compute with the kernel, and the results will be transformed back into the spatial domain before the output of the networks.

V. CONCLUSION

In this paper we proposed frequency regularization to reduce the information redundancy of convolutional neural networks devised for computer vision tasks. The proposed regularization maintains the tensors of parameters in the frequency domain where the high-frequency component can be truncated. During training, the tail part of tensors is truncated first before being input into the inverse discrete cosine transform to reconstruct the spatial tensors that are used for tensor operations. In particular, the dynamic tail-truncation strategy was proposed to improve the stability of network training. We applied the proposed frequency regularization in various state-of-the-art network architectures for evaluation. Comprehensive experiments demonstrated that between 90% to 99.99% of the parameters in the frequency domain can be truncated. This demonstrates the promising ability of the proposed frequency regularization to restrict information redundancy in convolutional neural networks for computer vision tasks.

ACKNOWLEDGMENT

(Chenqiu Zhao, Guanfang Dong, and Shupei Zhang contributed equally to this work.)

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, ''An image is worth 16 × 16 words: Transformers for image recognition at scale,'' in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–21. [Online]. Available: https://openreview.net/pdf?id=YicbFdNTTy

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[3] Z. Liu, J. Xu, X. Peng, and R. Xiong, ''Frequency-domain dynamic pruning for convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[4] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, ''Learning in the frequency domain,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2020, pp. 1737–1746, doi: 10.1109/CVPR42600.2020.00181.

[5] Y. Wang, C. Xu, S. You, D. Tao, and C. Xu, ''CNNpack: Packing convolutional neural networks in the frequency domain,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 1–6. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/3636638817772e42b59d74cff571fbb3-Paper.pdf

[6] Y. Wang, C. Xu, C. Xu, and D. Tao, ''Packing convolutional neural networks in the frequency domain,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2495–2510, Oct. 2019.

[7] H. Chen, Y. Wang, H. Shu, Y. Tang, C. Xu, B. Shi, C. Xu, Q. Tian, and C. Xu, ''Frequency domain compact 3D convolutional neural networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1641–1650.

[8] A. Ciurana, A. Mosella-Montoro, and J. Ruiz-Hidalgo, ''Hybrid cosine based convolutional neural networks,'' 2019, *arXiv:1904.01987*.

[9] C. Luo, Q. Lin, W. Xie, B. Wu, J. Xie, and L. Shen, ''Frequency-driven imperceptible adversarial attack on semantic similarity,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15294–15303.

[10] H. Rhee, Y. I. Jang, S. Kim, and N. I. Cho, ''LC-FDNet: Learned lossless image compression with frequency decomposition network,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6023–6032.

[11] N. Saldanha, S. L. Pintea, J. C. van Gemert, and N. Tomen, ''Frequency learning for structured CNN filters with Gaussian fractional derivatives,'' 2021, *arXiv:2111.06660*.

[12] Y. Shi, M. Wang, S. Chen, J. Wei, and Z. Wang, ''Transform-based feature map compression for CNN inference,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[13] M. Ulicny, V. A. Krylov, and R. Dahyot, ''Harmonic networks for image classification,'' in *Proc. Brit. Mach. Vis. Conf. (BMVC)*. Durham, U.K.: British Machine Vision Association, 2019, pp. 1–13.

[14] Z. Wang, H. Luo, P. Wang, F. Ding, F. Wang, and H. Li, ''VTC-LFC: Vision transformer compression with low-frequency components,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Dec. 2022, pp. 13974–13988.

[15] C. Zhang, Y. Xu, and Y. Shen, ''CompConv: A compact convolution module for efficient feature learning,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3006–3015.

[16] T. Zhao, W. Feng, H. Zeng, Y. Xu, Y. Niu, and J. Liu, ''Learning-based video coding with joint deep compression and enhancement,'' in *Proc. 30th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 3045–3054, doi: 10.1145/3503161.3548314.

[17] Y. Mi, Y. Huang, J. Ji, H. Liu, X. Xu, S. Ding, and S. Zhou, ''DuetFace: Collaborative privacy-preserving face recognition via channel splitting in the frequency domain,'' in *Proc. 30th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 6755–6764, doi: 10.1145/3503161.3548303.

[18] J. Wang, R. Du, D. Chang, K. Liang, and Z. Ma, ''Domain generalization via frequency-domain-based feature disentanglement and interaction,'' in *Proc. 30th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 4821–4829, doi: 10.1145/3503161.3548267.

[19] M. Zhou, J. Huang, C. Li, H. Yu, K. Yan, N. Zheng, and F. Zhao, ''Adaptively learning low-high frequency information integration for pansharpening,'' in *Proc. 30th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 3375–3384, doi: 10.1145/3503161.3547924.

[20] Y. Chen, R. Zhou, B. Guo, Y. Shen, W. Wang, X. Wen, and X. Suo, ''Discrete cosine transform for filter pruning,'' *Int. J. Speech Technol.*, vol. 53, no. 3, pp. 3398–3414, Feb. 2023.

[21] K. Goldberg, S. Shapiro, E. Richardson, and S. Avidan, ''Rethinking FUN: Frequency-domain utilization networks,'' 2020, *arXiv:2012.03357*.

[22] A. Dziedzic, J. Paparrizos, S. Krishnan, A. Elmore, and M. Franklin, ''Band-limited training and inference for convolutional neural networks,'' in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 1745–1754.

[23] Z. Shao, X. Chen, L. Du, L. Chen, Y. Du, W. Zhuang, H. Wei, C. Xie, and Z. Wang, ''Memory-efficient CNN accelerator based on interlayer feature map compression,'' *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 2, pp. 668–681, Feb. 2022.

[24] Y. Wang, C. Xu, C. Xu, and D. Tao, ''Beyond filters: Compact feature map for portable deep model,'' in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3703–3711.

[25] M. Heusel H. Ramsauer, T. Unterthiner, and B. Nessler, ''GANs trained by a two time-scale update rule converge to a local Nash equilibrium,'' in *Proc. 31st Conf. Neural Inf. Process. Syst.*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017, pp. 1–12. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[26] Z. Hou, M. Qin, F. Sun, X. Ma, K. Yuan, Y. Xu, Y.-K. Chen, R. Jin, Y. Xie, and S.-Y. Kung, ''CHEX: CHannel EXploration for CNN model compression,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12277–12288.

[27] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, ''HRank: Filter pruning using high-rank feature map,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1526–1535.

[28] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, ''Pruning filters for efficient ConvNets,'' in *Proc. ICLR*, 2017, pp. 1–13.

[29] S. Gao, F. Huang, Y. Zhang, and H. Huang, ''Disentangled differentiable network pruning,'' in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*. Tel Aviv, Israel: Springer, Oct. 2022, pp. 328–345.

[30] G. Fang, X. Ma, M. Song, M. Bi Mi, and X. Wang, ''DepGraph: Towards any structural pruning,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 16091–16101.

[31] S. Jia, C. Ma, T. Yao, B. Yin, S. Ding, and X. Yang, ''Exploring frequency adversarial attacks for face forgery detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4093–4102.

[32] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, ''Detecting camouflaged object in frequency domain,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4494–4503.

[33] W. Shu, J. Wan, K. C. Tan, S. Kwong, and A. B. Chan, ''Crowd counting in the frequency domain,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19586–19595.

[34] O. Rippel, J. Snoek, and R. P. Adams, ''Spectral representations for convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[35] M. Buckler, N. Adit, Y. Hu, Z. Zhang, and A. Sampson, ''Dense pruning of pointwise convolutions in the frequency domain,'' 2021, *arXiv:2109.07707*.

[36] S. Han, H. Mao, and W. J. Dally, ''Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding,'' 2016, *arXiv:1510.00149*.

[37] W. Chen, J. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, ''Compressing convolutional neural networks in the frequency domain,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1475–1484.

[38] R. Lin, J. Ran, D. Wang, K. H. Chiu, and N. Wong, ''EZCrop: Energy-zoned channels for robust output pruning,'' in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3595–3604.

[39] S. Zhang, M. Gao, Q. Ni, and J. Han, ''Filter pruning with uniqueness mechanism in the frequency domain for efficient neural networks,'' *Neurocomputing*, vol. 530, pp. 116–124, Apr. 2023.

[40] K. Cheinski and P. Wawrzynski, ''DCT-Conv: Coding filters in convolutional networks with discrete cosine transform,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–6.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[42] Y. Guo, A. Yao, and Y. Chen, ''Dynamic network surgery for efficient DNNs,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[44] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[45] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[46] A. Krizhevsky et al., ''Learning multiple layers of features from tiny images,'' Toronto, ON, Canada, 2009.

[47] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, ''ThiNet: Pruning CNN filters for a thinner net,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2525–2538, Oct. 2019.

[48] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, ''Object detection networks on convolutional feature maps,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1476–1481, Jul. 2017.

[49] V. Iglovikov and A. Shvets, ''TernausNet: U-Net with VGG11 encoder pre-trained on ImageNet for image segmentation,'' 2018, *arXiv:1801.05746*.

CHENQIU ZHAO received the B.S. and M.S. degrees in software engineering from Chongqing University, Chongqing, China, in 2014 and 2017, respectively, and the Ph.D. degree from the Department of Computing Science, University of Alberta, Edmonton, Canada, in 2022. From 2017 to 2018, he was a Research Associate with the Institute for Media Innovation, Nanyang Technological University, Singapore. He is currently a Postdoctoral Researcher with the Multimedia Research Center, Department of Computing Science, University of Alberta. His current research interests include distribution learning, video segmentation, pattern recognition, and deep learning.

GUANFANG DONG received the M.Sc. degree from the Department of Engineering, The Hong Kong University of Science and Technology, in 2020. He is currently pursuing the Ph.D. degree in computing science with the University of Alberta, under the supervision of Prof. A. Basu. His research interests include explainable AI, noise removal, and background subtraction. His current research interests include distribution learning, video segmentation, pattern recognition, and deep learning.

SHUPEI ZHANG received the bachelor's and master's degrees in engineering from Beihang University, Beijing, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in computing science with the University of Alberta, Edmonton, AB, Canada.

From 2016 to 2019, he was a Research Assistant with Beihang University. Currently, he is a Research Assistant with the University of Alberta. His research interests include visual saliency, video compression, and HDR reconstruction using single exposure. He has published one journal article and one conference paper.

ZIJIE TAN received the B.Sc. degree in computing science from the University of Alberta, in 2022, where he is currently pursuing the M.Sc. degree with the Multimedia Research Center, Department of Computing Science. His current research interests include computer vision, pattern recognition, and deep learning.

ANUP BASU received the Ph.D. degree in CS from the University of Maryland, College Park, USA. He originated the use of foveation for image, video, stereo, and graphics communication in the early 1990s; an approach that is now widely used in industrial standards. He pioneered the active camera calibration method emulating the way the human eyes work and showed that this method is far superior to any other camera calibration method. He pioneered a single camera panoramic stereo, and several new approaches merging foveation and stereo with application to 3D TV visualization and better depth estimation. His current research interests include multi-dimensional image processing and visualization for medical, consumer and remote sensing applications, multimedia in education and games, and robust wireless 3D multimedia transmission.