Multi-band Convolutional Neural Network for Efficient Utilization of Model Parameters

Milton Mondal
Department of Electrical Engineering
Indian Institute of Technology Delhi
milton.mondal@ee.iitd.ac.in

Bishshoy Das
Department of Electrical Engineering
Indian Institute of Technology Delhi
bishshoy.das@ee.iitd.ac.in

Prof. Brejesh Lall
Department of Electrical Engineering
Indian Institute of Technology Delhi
brejesh@ee.iitd.ac.in

Dr. Pushpendra Singh
Electronics and Communication Engineering
National Institute of Technology Hamirpur
spushp@nith.ac.in

Prof. Sumantra Dutta Roy
Department of Electrical Engineering
Indian Institute of Technology Delhi
sumantra@ee.iitd.ac.in

Prof. Shiv Dutt Joshi
Department of Electrical Engineering
Indian Institute of Technology Delhi
sdjoshi@ee.iitd.ac.in

***Abstract*—A convolutional layer of a traditional convolutional neural network (CNN) does not ensure the extraction of complementary bands of the input data. Thus a significant amount of redundancy is observed among different convolutional filters. Here, we propose a novel architecture design framework called 'Multi-band CNN' to efficiently utilize model parameters in a CNN. The framework generates four filters from a single filter by varying their frequency responses, extracting four complementary bands of the input data without increasing the parameter count. This leads to higher parameter utilization and results in a compact network with reduction in trainable parameter count but with close to the same accuracy as the base model. We perform experiments using residual networks (ResNet-32, ResNet-56, and ResNet-110) on datasets like CIFAR-10 and CIFAR-100. Our results show improved classification accuracy for CIFAR-10 when introducing a multi-band layer in the first convolutional layer, while there is no significant drop in accuracy for CIFAR-100. The performance is better when replacing the first convolutional layer instead of the last one, indicating that low-level features generated by sub-band filtering are more beneficial to the network than high-level features provided by filter banks at the final layer. The proposed Multi-band CNN framework offers a potential solution for reducing the number of filters required to train and the computational complexity of generating feature maps in CNNs, while maintaining or even improving classification accuracy.**

***Index Terms*—Convolutional Neural Network (CNN), Filter Bank, Model Compression, Filter Pruning, Deep Learning**

I. Introduction

Convolutional neural networks (CNNs) have emerged as the primary tool for solving various computer vision applications, including multi-class classification, semantic segmentation, image captioning, and other challenging tasks [1], [2], [3]. Over the years, the depth and width of deep neural networks have increased to achieve better representations. However, this increase in complexity leads to a significant rise in computations and parameter count, making it challenging to deploy these models on resource-constrained devices such as mobile phones and embedded systems. To address this issue, researchers have proposed several model compression methods and efficient training mechanisms, such as parameter quantization, tensor decomposition, knowledge distillation, compact network synthesis, and pruning network parameters. These approaches aim to produce compact models without significantly degrading performance by reducing the computational cost and redundancy in deep neural networks.

Although there are several methods that compress a CNN, but our objective in this paper is how can we ensure that different convolutional filters capture complimentary bands information from the input image. In traditional convolutional layer, existing architecture design never assures this or the production of diverse features using these filters. Here, we propose a novel architecture design framework called 'Multi-band CNN' (MCNN) for efficiently utilizing model parameters. The framework generates four filters from a single filter by varying their frequency responses to extract four complementary bands of the input data, without increasing the network's parameter count. This leads to higher utilization of the network's parameters and results in a compact network with reduced parameter count but with close to the same accuracy as the base model.

Previously, few attempts [4] have been made to incorporate the classical multi-band filtering scheme with the neural network to represent the input-output mapping of the dataset more accurately. However, the objective of these methods is to provide the neural network with low-level features as input instead of passing the raw data to the network directly. The learning capacity of the network increases by incorporating this manually chosen feature extractor in the first layer. However, our objective is different from the existing approaches. Our approach provides the flexibility to train network filters of all layers instead of manually fixing the filters of the first layer. We know that the training time of the base model increases with the number of filters present in the model. So, while capturing complementary bands information, our objective is here to ensure that the number of filters that need to be trained reduces, and yet we achieve a similar classification performance. Current CNN models [5], [6], [7] do not guarantee that different filters within a layer will extract complementary information from the input. So, we attempt to solve this problem by designing a novel architecture named 'Multi-band CNN (M-CNN).'

MCNN reduces the filter count of a network by a factor of four, and generates four filters from a single filter. This indicates reduction in the trainable paramereters by a factor of four in a convolutional layer where multi-band filtering is incorporated. This compact and efficient network provides similar accuracy as the base model. Using M-CNN, we notice improvement in accuracy for standard datasets like CIFAR-10. The results are consistent with several residual networks like ResNet-32, ResNet-56, and ResNet-110 for CIFAR-10 classification. We observe no drop in accuracy using M-CNN for CIFAR-100 classification using ResNet-56.

The main contributions of this paper are as follows:

- We propose a method to generate four filters from a single filter in a convolutional neural network from complementary band information.
- We show that the proposed architecture design framework can be used to generate a compact network with reduced parameter count but with close to the same accuracy as the base model.

II. Related Works

The depth and width of the deep neural networks have increased over the years so that the neural networks can achieve better representations. With more complicated tasks, a greater number of parameters are required to capture the patterns within data. The depth of the neural network is increased over time to achieve better performance but at the cost of a significant increase in computations [8]. The number of parameters and floating point operations (FLOPs) increase as the number of hidden layers grows for a CNN [9]. The number of filters present in a network determines the model size. While dealing with deep architectures, storing millions of parameters and performing billions of floating point operations during inference have become common on workstations and server grade systems [3]. However, the requirement of memory for the storage of the network parameters in handheld devices, and computational resources for performing convolutions thereof cannot be fulfilled due to resource limitations in mobile devices or any other embedded systems when the depth and width of CNN are too high [10].

To deal with this, researchers have proposed several model compression methods and efficient training mechanisms which can produce a compact model without degrading the performance significantly. These solutions include (a) parameter quantization [11], [12], [13], [14], (b) tensor decomposition [15], [16], [17], [18], (c) knowledge distillation [19], [20], [21], [22], (d) compact network synthesis [23], [24], [25], [26], (e) pruning network parameters [27], [28], [29], [30], etc. Out of all these solutions, pruning network parameters reduces the computational cost of deep neural networks (DNN) by first identifying and then deleting the redundant parameters in such a manner that the learning effectiveness is maintained compared to the unpruned (base) model. It is observed that a significant amount of redundancy exists among different convolution kernels and even in a single kernel for deep neural networks [31]. Unlike network synthesis methods, pruning offers the flexibility to the user to choose a standard base model (like VGG [5], ResNet [6] etc.), and thereafter it automatically generates a compact model from the base model by trimming the redundant parameters.

Pruned models can achieve the same classification performance, despite the latter having fewer filters. Filter pruning creates a compact and efficient model by removing unimportant filters from the base model, while network synthesis methods, such as Nest [23], Neurogenesis [24], SCANN [26], design compact architectures from scratch based on gradient profiles and neuron activations. The vanishing gradient problem hinders training for very deep plain networks, but deeper multi-branch networks can overcome this issue with skip connections. The development of neural network synthesis for multi-branch networks like ResNet is still in its early stages. Different approaches have been taken to make the CNN more compact and efficient. However, existing methods does not ensure extraction of complementary bands information while training the network. We solve this problem by novel architecture design in which model filters to capture unique image features. We apply this method to deep residual networks, and redundancy in filters capturing similar features is removed with our proposed MCNN.

III. Mathematical Analysis

First, we discuss our motivation for this work. We then describe the proposed method and thereafter the frequency response of filters of MCNN in detail.

A. Motivation

We introduce a sub-band filtering scheme, ensuring filters in a layer capture complementary bands by design.Unlike previous attempts integrating classical multi-band filtering schemes with neural networks, our approach allows training filters of all layers instead of manually fixing the first layer's filters. This aims to reduce the number of filters required to be trained while maintaining the classification performance. Current CNN models do not guarantee that filters within a layer extract complementary information, which we address using MCNN.

Let's say we have a 3×3 matrix $h_1(r,s)$ which is a filter. We represent the filter coefficients as a 2D matrix. For the sake of this analysis, we assume that the filter has all elements equal to 1, without loss of generality. So, the filter matrix is given by:

$$h_1 = h_{r,s} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Here, $h_1(r,s)$ is a low pass filter. We can generate four filters from this filter by multiplying it with four different multipliers (1, $(-1)^r$, $(-1)^s$, $(-1)^{r+s}$). Let $h_2(r,s)$, $h_3(r,s)$, and $h_4(r,s)$ be three other filters generated from $h_1(r,s)$. We can generate these filters by multiplying $h(r,s)$ with three

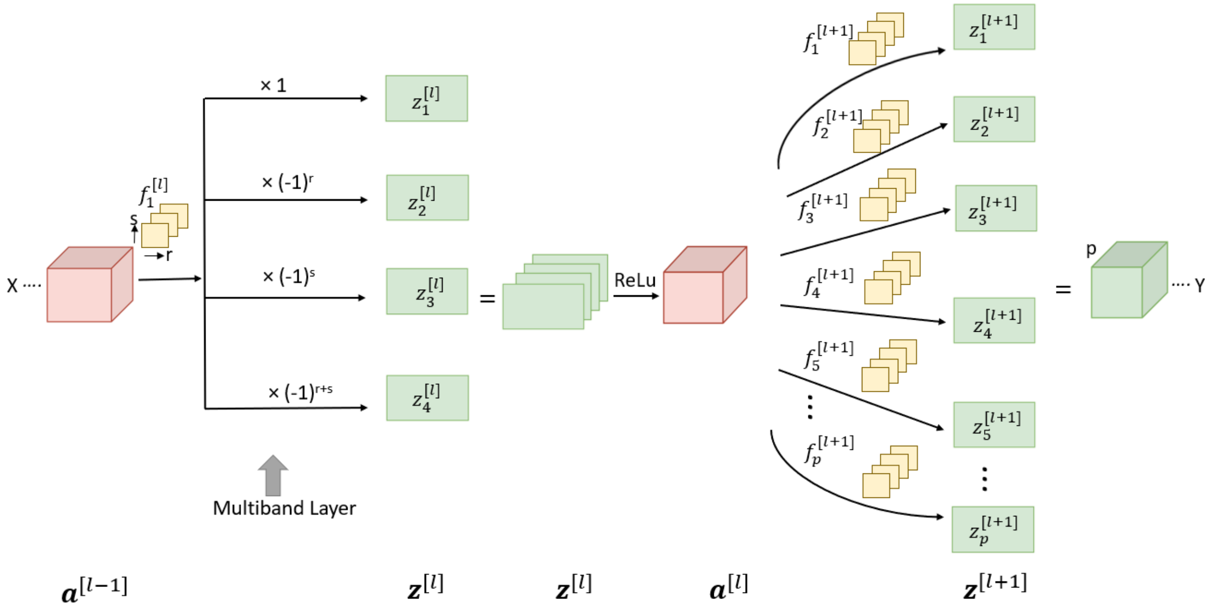

Fig. 1. Block diagram of a Multi-band CNN. Here, conventional convolutional layer is replaced by multi-band convolutional layer for the first layer.

different multipliers (ignoring the trivial multiplication of 1 for $h_1(r, s)$) .

$$h_2 = (-1)^r h_{r,s} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

h_2 is a vertical edge detector.

$$h_3 = (-1)^s h_{r,s} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

h_3 is a horizontal edge detector.

$$h_4 = (-1)^{r+s} h_{r,s} = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

h_4 is a diagonal edge detector.

B. Proposed Method

In M-CNN, We produce four filters from a single filter by altering their frequency responses in order to extract four complimentary bands of the input data. By maintaining the original filter and adding three distinct masks to it, a single filter may produce a total of four filters. Three filters that are generated are referred to as “derived” filters, while we refer to the filter as the “base” filter from which they are developed. We multiply a single filter with four different deterministic multipliers to capture the complementary bands of the input. Fig. 1 depicts the block diagram of M-CNN, in which the first layer is a multi-band convolutional layer. Multi-band convolutional layer is different from the existing convolutional layer. This work explores the impact of substituting a conventional convolutional layer with a multi-band layer. So, Multi-band CNN approach can generate a vertical edge detector (filter h_2), a horizontal edge detector (filter h_3), and a diagonal edge detector (filter h_4) from a single filter h_1. These are three useful masks for diverse feature exatraction from a single image. The impact of multi-band CNN is such that even when the base filter smoothens the original image, then also the derived filters can extract different types of edges (horizontal, vertical and diagonal) from the image using the same base filter. This is because the derived filters are generated from the original filter by altering their frequency responses.

All filters participate in the training for conventional convolutional layers during CNN training. However for a multi-band layer, only the base filters are learned, and the derived filters are obtained directly from the base filters. In a multi-band layer, all derived filters have the same absolute value as the base filter, but their ‘sign’ varies depending on the spatial position of the filter coefficient. The combination of a base filter and derived filters constitutes a filter bank.

C. Frequency Response of Filters

Modification in the impulse response of derived filters with reference to base filter leads to the modification in the frequency response of derived filters. If $h(r, s)$ is the impulse response and $H(u, v)$ is the frequency response of the base filter, then the impulse response and frequency response of the filters belong to the filter bank are shown below,

$$h_1(r, s) = h(r, s) \iff H_1(u, v) = H(u, v) \tag{1}$$

$$h_2(r, s) = (-1)^r \times h(r, s) \iff H_2(u, v) = H(u - \pi, v) \tag{2}$$

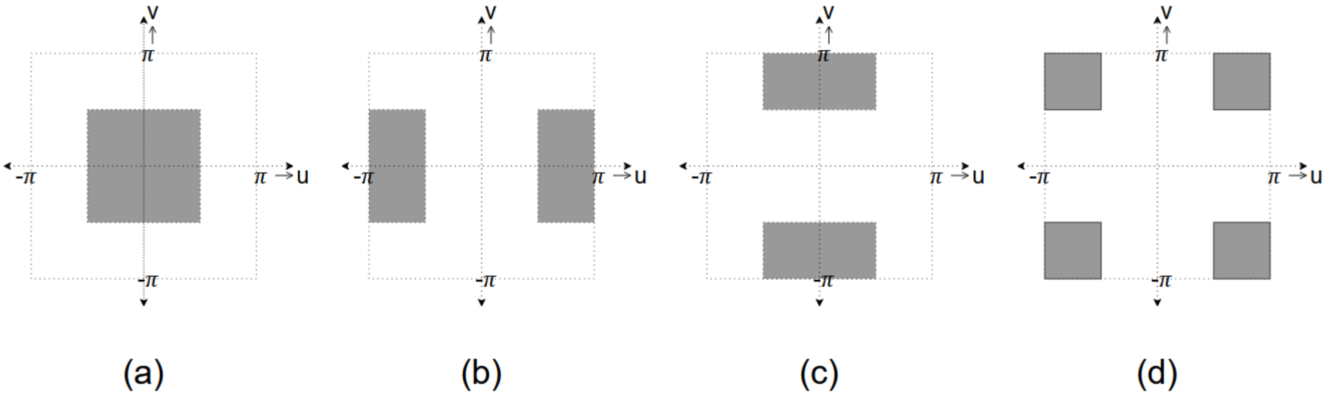

Fig. 2. Frequency response of a filter bank. The frequencies for which a filter has non-zero response, is highlighted in gray. If the frequency response of base filter is (a) $H_1(u,v)$ then the response of derived filters will be (b) $H_2(u,v)$, (c) $H_3(u,v)$, (d) $H_4(u,v)$.

$$\begin{aligned} h_3(r,s) = (-1)^s \times h(r,s) \iff \\ H_3(u,v) = H(u, v-\pi) \end{aligned} \quad (3)$$

$$\begin{aligned} h_4(r,s) = (-1)^{r+s} \times h(r,s) \iff \\ H_4(u,v) = H(u-\pi, v-\pi) \end{aligned} \quad (4)$$

Fig. 2 shows the frequency bands where the filters have a non-zero response. Interestingly, all four filters of a filter bank have been generated from a single filter. Yet, they are capable of capturing different frequency bands of the input signal as shown in Fig. 2.

IV. Experimental Results

In this section, we observe the impact on the performance of the model while replacing a conventional convolutional layer with a multi-band convolutional layer. The effect of introducing a multi-band layer is observed here for the first and last convolutional layer of a network. Over the years, residual networks have been found to be more effective, and they also perform better than plain networks. So we utilize only residual networks for these experiments. We perform experiments with different datasets (CIFAR-10, CIFAR-100) and with different architectures (ResNet-32, ResNet-110).

We find that for the CIFAR-10 classification task [Table I], incorporating a ‘multi-band’ layer in the first convolutional layer improves the classification accuracy by 0.31% for ResNet-32, 0.33% for ResNet-56 and 0.02% for ResNet-110 as shown in Fig. 3. When we replace the last convolutional layer with a multi-band layer, we observe 0.26% and 0.18% improvement in the test accuracy over baseline for ResNet-32 and ResNet-56, respectively. However, a drop of 0.28% in classification accuracy is observed compared to the baseline for ResNet-110. For experiments with CIFAR-100, we observe a minute drop of 0.09% and 0.45% compared to baseline while replacing the first or last convolutional layer respectively when ResNet-56 is used as the base model.

TABLE I
Performance comparison between M-CNN and CNN. MCNN-F and MCNN-L are used when the first and final convolutional layers are replaced with a multi-band layer, respectively.

			MCNN-F		**MCNN-L**	
Dataset	**Arch**	**Baseline acc (%)**	**Final acc (%)**	**Acc drop (%)**	**Final acc (%)**	**Acc drop (%)**
CIFAR10	ResNet32	92.85	93.16	-0.31	93.11	-0.26
	ResNet56	93.52	93.85	-0.33	93.7	-0.18
	ResNet110	93.94	93.96	-0.02	93.66	0.28
CIFAR100	ResNet56	71.79	71.7	0.09	71.34	0.45

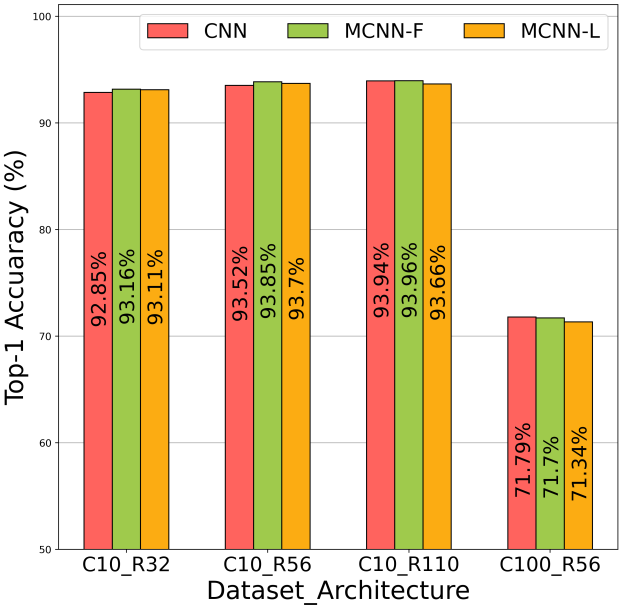

Fig. 3. Performance comparison between M-CNN and CNN. MCNN-F and MCNN-L are used when the first and final convolutional layers are replaced with a multi-band layer, respectively.

TABLE II
NUMBER OF TRAINABLE FILTERS IN THE FIRST AND FINAL CONVOLUTIONAL LAYERS OF RESNET-32 AND RESNET-56, AND RESNET-110.

Arch	**Base Model**		**MCNN-F**		**MCNN-L**	
	First layer	**Last Layer**	**First layer**	**Last Layer**	**First layer**	**Last Layer**
ResNet32	16	64	4	64	16	16
ResNet56	16	64	4	64	16	16
ResNet110	16	64	4	64	16	16

All these experiments show that the performance of the models is better when replacing the first convolutional layer instead of replacing the last convolutional layer with a multi-band layer [Table II]. It indicates that low-level features generated by sub-band filtering are more beneficial to the network than high-level features provided by filter banks at the final layer. In summary, our experiments reveal that replacing the first convolutional layer with a multi-band layer yields better performance compared to replacing the last convolutional layer. This suggests that low-level features generated by sub-band filtering play a more crucial role in the network's performance than the high-level features provided by filter banks at the final layer. The observed improvements in classification accuracy for CIFAR-10 and minimal drops in CIFAR-100 demonstrate the potential of the multi-band layer in enhancing the network's effectiveness. These findings highlight the importance of efficiently capturing low-level features in CNNs and pave the way for further exploration of multi-band layers in different network architectures and applications.

V. CONCLUSION

In conclusion, the proposed Multi-band CNN framework presents an innovative and efficient approach to address the challenges of increased parameter count and computational complexity in convolutional neural networks, particularly for resource-constrained devices. By generating four filters from a single filter and extracting four complementary bands of input data without increasing the parameter count, M-CNN leads to higher utilization of the network's parameters and results in a compact network with reduced parameter count while maintaining or even improving classification accuracy.

Our experiments on benchmark datasets like CIFAR-10 and CIFAR-100 using residual networks such as ResNet-32, ResNet-56, and ResNet-110 demonstrate the effectiveness of the M-CNN framework. For CIFAR-10, M-CNN reduces the number of filters required to train by a factor of four for the first layer, improving classification accuracy consistently across different residual network architectures. Furthermore, we observe no significant drop in accuracy using M-CNN for the CIFAR-100 classification task using ResNet-56. The introduction of the filter bank in the M-CNN framework reduces the number of filters needed to be trained by a factor of four. Moreover, an efficient implementation of convolution for all filters within a filter bank can potentially reduce the number of required multiplications and additions, accelerating the generation of feature maps. This is possible because all the filters of a filter bank have the same coefficients and differ only in their 'sign', unlike conventional convolutional layers where the coefficients of every filter can be entirely different from each other.

Future work can explore efficient implementations of convolution for all filters within a filter bank and investigate the impact of the multi-band layer on other network architectures and datasets. Additionally, the extension of the current work to incorporate MCNN framework in advanced multi-branch networks, such as MobileNet, can further enhance the applicability and performance in various computer vision tasks and especially for resource-constrained environments.

REFERENCES

[1] J.-H. Luo and J. Wu, "An entropy-based pruning method for cnn compression," *arXiv preprint arXiv:1706.05791*, 2017.

[2] Q. Huang, K. Zhou, S. You, and U. Neumann, "Learning to prune filters in convolutional neural networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 709–718.

[3] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4340–4349.

[4] S. K. Ghosh, R. K. Tripathy, M. R. Paternina, J. J. Arrieta, A. Zamora-Mendez, and G. R. Naik, "Detection of atrial fibrillation from single lead ecg signal using multirate cosine filter bank and deep neural network," *Journal of medical systems*, vol. 44, no. 6, pp. 1–15, 2020.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[8] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang, "Soft filter pruning for accelerating deep convolutional neural networks," *arXiv preprint arXiv:1808.06866*, 2018.

[9] P. Singh, V. K. Verma, P. Rai, and V. P. Namboodiri, "Acceleration of deep convolutional neural networks using adaptive filter pruning," *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[10] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.

[11] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "Haq: Hardware-aware automated quantization with mixed precision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8612–8620.

[12] Z. Liu, W. Luo, B. Wu, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real net: Binarizing deep network towards real-network performance," *International Journal of Computer Vision*, vol. 128, no. 1, pp. 202–219, 2020.

[13] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[14] P. Nayak, D. Zhang, and S. Chai, "Bit efficient quantization for deep neural networks," in *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*. IEEE, 2019, pp. 52–56.

[15] K. Hayashi, T. Yamaguchi, Y. Sugawara, and S.-i. Maeda, "Exploring unexplored tensor network decompositions for convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] H. Kim, M. U. K. Khan, and C.-M. Kyung, "Efficient neural network compression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 569–12 577.

[17] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7370–7379.

[18] J. Xue, Y. Zhao, S. Huang, W. Liao, J. C.-W. Chan, and S. G. Kong, "Multilayer sparsity-based tensor decomposition for low-rank tensor completion," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[19] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[20] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, "Knowledge distillation: A good teacher is patient and consistent," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 925–10 934.

[21] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1365–1374.

[22] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 953–11 962.

[23] X. Dai, H. Yin, and N. K. Jha, "Nest: A neural network synthesis tool based on a grow-and-prune paradigm," *IEEE Transactions on Computers*, vol. 68, no. 10, pp. 1487–1497, 2019.

[24] K. Maile, E. Rachelson, H. Luga, and D. G. Wilson, "When, where, and how to add new neurons to anns," in *International Conference on Automated Machine Learning*. PMLR, 2022, pp. 18–1.

[25] W. Xia, H. Yin, and N. K. Jha, "Efficient synthesis of compact deep neural networks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[26] S. Hassantabar, Z. Wang, and N. K. Jha, "Scann: Synthesis of compact and accurate neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[27] Z. Liu, X. Zhang, Z. Shen, Y. Wei, K.-T. Cheng, and J. Sun, "Joint multi-dimension pruning via numerical gradient update," *IEEE Transactions on Image Processing*, vol. 30, pp. 8034–8045, 2021.

[28] G. Yuan, X. Ma, C. Ding, S. Lin, T. Zhang, Z. S. Jalali, Y. Zhao, L. Jiang, S. Soundarajan, and Y. Wang, "An ultra-efficient memristor-based dnn framework with structured weight pruning and quantization using admm," in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2019, pp. 1–6.

[29] L. Zeng and X. Tian, "Accelerating convolutional neural networks by removing interspatial and interkernel redundancies," *IEEE transactions on cybernetics*, vol. 50, no. 2, pp. 452–464, 2018.

[30] X. Xiao, Z. Wang, and S. Rajasekaran, "Autoprune: Automatic network pruning by regularizing auxiliary parameters," *Advances in neural information processing systems*, vol. 32, 2019.

[31] A. Polyak and L. Wolf, "Channel-level acceleration of deep face representations," *IEEE Access*, no. 3, pp. 2163–2175, 2015.