Journal of Imaging Science and Technology® 67(6): 1–11, 2023.

Digital Modeling on Large Kernel Metamaterial Neural Network

Quan Liu, Hanyu Zheng, Brandon T. Swartz, Ho hin Lee, and Zuhayr Asad
Vanderbilt University, Nashville, TN 37212

Ivan Kravchenko
Oak Ridge National Laboratory, Oak Ridge, TN 37830

Jason G. Valentine and Yuankai Huo▲
Vanderbilt University, Nashville, TN 37212
E-mail: yuankai.huo@vanderbilt.edu

Abstract. *Deep neural networks (DNNs) utilized recently are physically deployed with computational units (e.g., CPUs and GPUs). Such a design might lead to a heavy computational burden, significant latency, and intensive power consumption, which are critical limitations in applications such as Internet of Things (IoT), edge computing, and usage of drones. Recent advances in optical computational units (e.g., metamaterial) have shed light on energy-free and light-speed neural networks. However, the digital design of the metamaterial neural network (MNN) is fundamentally limited by its physical limitations, such as precision, noise, and bandwidth during fabrication. Moreover, the unique advantages of MNN's (e.g., light-speed computation) are not fully explored via standard 3×3 convolution kernels. In this paper, we propose a novel large kernel metamaterial neural network (LMNN) that maximizes the digital capacity of the state-of-the-art (SOTA) MNN with model re-parametrization and network compression, while also considering the optical limitation explicitly. The new digital learning scheme can maximize the learning capacity of MNN while modeling the physical restrictions of meta-optics. With the proposed LMNN, the computation cost of the convolutional front-end can be offloaded to fabricated optical hardware. The experimental results on two publicly available datasets demonstrate that the optimized hybrid design improved classification accuracy while reducing computational latency. The development of the proposed LMNN is a promising step towards the ultimate goal of energy-free and light-speed AI.*

[DOI: 10.2352/J.ImagingSci.Technol.2023.67.6.060404]

1. INTRODUCTION

Digital neural networks (DNN) are essential in modern computer vision tasks. The convolutional neural network (CNN) is arguably the most widely used AI approach for image classification [1–3], segmentation [4, 5], and detection [6, 7]. Even for more recent vision transformer-based models, convolution is still an essential component for extracting local image features [8–12]. Current CNNs are typically deployed with computational units (e.g., CPUs and GPUs). Such a design might lead to a heavy computational burden, significant latency, and intensive power consumption, which are critical limitations in applications such as Internet of Things (IoT), edge computing, and usage of drones. Therefore, the AI community has started to seek DNN models with less energy consumption and lower latency. However, we may not achieve energy-free and light-speed DNN following the current trends in research.

Fortunately, the recent advances in optical computational units (e.g., metamaterial) have shed light on energy-free and light-speed neural networks (Figure 1). At its current stage, the SOTA metamaterial neural network (MNN) is implemented as a hybrid system, where the optical processors are used as a light-speed and energy-free front-end convolutional operator with a digital feature aggregator. Such design reduces the computational latency since the convolution operations are implemented by optical units, which off-loads more than 90 percent of the floating-point operations (FLOPs) in conventional CNN backbones like VGG [13] and ResNet [14]. However, the digital design of the MNN is fundamentally limited by its physical structures, namely (1) the optic system can only take positive value; (2) non-linear computations are challenging for free-space optic devices at low light intensity; (3) the implementation of the optical convolution is restricted by limited kernel size, channel number, precision, noise, and bandwidth. Furthermore, limitations also exist in the current optic fabrication process: (1) only the first layer of a neural network can be fabricated, and (2) limited layer capacity and weight precision. Therefore, the unique advantages of the MNNs (e.g., light-speed computation) are not fully explored via standard 3×3 convolution kernels. The large convolution kernel (greater than 3×3) provides the larger reception fields which plays essential roles in segmentation and classification tasks [15–17]. Compared with traditional small kernel convolution [18], A larger receptive field (achieved using larger kernels or more convolutional layers) allows the network to see and model larger spatial contexts, which can be crucial in tasks where spatial details like boundaries matter [19].

▲ IS&T Member.

Received July 16, 2023; accepted for publication Nov. 15, 2023; published online Jan. 8, 2024. Associate Editor: Yi Wang.

1062-3701/2023/67(6)/060404/11/$25.00

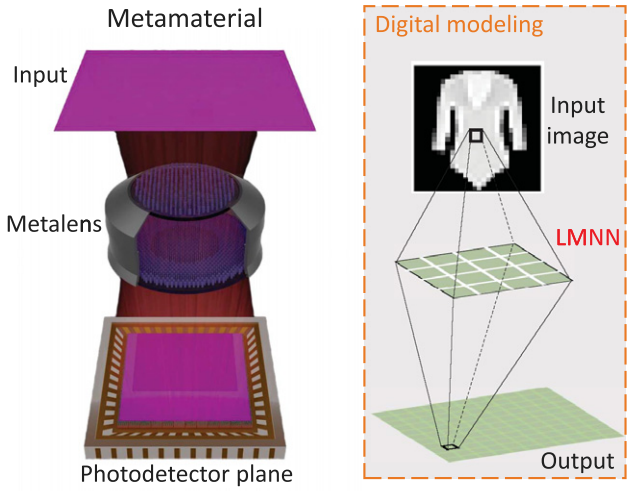

Figure 1. This study provides a digital modeling platform for designing and optimizing a metamaterial neural network (MNN). The proposed large kernel metamaterial neural network (LMNN) is able to maximize the performance of an MNN without introducing extra computational complexity during the inference stage.

In this paper, we propose a novel large kernel metamaterial neural network (LMNN) that maximizes the digital capacity of the SOTA MNN with model re-parametrization and network compression, while also considering the optical limitation explicitly. Our model maximizes the advantage of the light-speed natural of optical computing by implementing larger convolution kernels (e.g., 7×7, 11×11). The proposed LMNN yields larger reception fields, without sacrificing low computational latency and low energy consumption. Furthermore, the aforementioned physical limitations of LMNNs are explicitly addressed via optimized digital modeling. We evaluate our model on image classification tasks using two public datasets: FashionMNIST [20] and STL-10 [21]. The proposed LMNN achieved superior classification accuracy as compared with the SOTA MNN and model re-parametrization methods. Overall, the system's contributions can be summarized as follows:

- We propose the large convolution kernel design for an LMNN to achieve a larger reception field, lower computational latency, and less energy consumption.
- We introduce the model re-parameterization and multi-layer compression mechanism to compress the multi-layer multi-branch design to a single layer for the LMNN implementation. This maximizes the model capacity without introducing any extra burden during the optical inference stage.
- The physical limitations of LMNNs (e.g., limited kernel size, channel number, precision, noise, non-negative restriction, and bandwidth) are explicitly addressed via optimized digital modeling.
- We implemented a single-layer LMNN with real physical metamaterial fabrication to demonstrate the feasibility of our hybrid design.

The rest of the paper is organized as follows. In Section 2, we introduce background and related research relevant to large kernel convolution, re-parameterization, and optical neural networks. In Section 3, our proposed LMNN model is presented. It includes the large kernel re-parameterization, meta-optic adaptation, and model compression strategy. Section 4 focuses on presenting the dataset and experiment implementation details. Section 5 provides the experimental results and ablation study. Then, in Sections 6 and 7, we provide the discussion and conclude our work.

2. RELATED WORK

2.1 *Models with Large Kernel Convolution*

For a decade, a common practice in choosing optimal kernel size in convolution is to leverage 3×3 kernels. In recent years, more attention has been put into a larger kernel design. The Inception network proposes an early design of adapting large kernels for vision recognition tasks [22]. After developing several variations [23, 24], large kernel models became less popular. Global Convolution Networks (GCNs) [16] employ the large kernel idea by utilizing $1 \times K$ followed by $K \times 1$ to achieve improvement in model performance for semantic segmentation.

Current limitations in leveraging large kernel convolution kernel can be divided into two aspects: (1) scaling up the kernel sizes lead to the degradation of model performance, and (2) its high computational complexity. According to the Local Relation Networks (LRNet) [25], the spatial aggregation mechanism with dynamic convolution is used to substitute traditional convolution operation. As compared with the traditional 3×3 kernels, the LRNet [25] leverages 7×7 convolution to improve model performance. However, the performance becomes saturated by scaling up the kernel size to 9×9. Similar to RepLKNet [11], scaling up the convolution kernel size to 31×31 without prior structural knowledge demonstrates the decrease of model performances. To leverage the heavyweight computation of large kernel convolution, [26] introduced the Shufflemixer for lightweight design.

2.2 *Model Compression and Re-parameterization*

Though many complicated ConvNets [27, 28] deliver higher accuracy than more simple ones, the drawbacks are significant. (1) The complicated multi-branch designs (e.g., residual addition in ResNet [14] and branch-concatenation in Inception [23]) make the model difficult to implement and customize, and slow down the inference and reduce memory utilization. (2) Some components (e.g., depthwise convolution in Xception [22] and MobileNets [29], and channel shuffle in ShuffleNets [30]) increase memory access costs and lack support for various devices.

Model compression [31] aimed to reduce the model size and computational complexity [32, 33] while maintaining their performance including pruning and quantization. Pruning has been widely used to compress deep learning models by removing the unnecessary or redundant parame-

ters from a neural network without affecting its accuracy [34–36]. Quantization has two categories: Quantization-Aware Training (QAT) [37, 38] and Post-Training Quantization (PTQ). QAT applies quantization operation in the training stage. In contrast, PTQ takes a full precision network for training and quantized it in the post stage [39–41]. Attempting to decrease the redundancy of CNN, SCConv [42] compresses the model by exploiting the spatial and channel redundancy among features.

2.3 *Optical Neural Network*

Optical neural network (ONN) uses light instead of electrical signals to perform matrix multiplications [43–45] which can be much faster and more energy-efficient than traditional digital neural networks. Most ONNs use a hybrid model structure and implement linear computation with optic device and non-linear operation digitally [46–49]. Besides the use of optical devices, ONNs have been implemented on nanophotonic circuits [50, 51] and light-wave linear diffraction [52, 53] to improve model efficiency. For non-linear computation, [54, 55] have proposed implementing non-linear operations with optic device on ONN.

3. METHOD

Problem statement. The goal of this study is to develop a new digital learning scheme to maximize the learning capacity of MNN while modeling the physical restrictions of meta-optics. With the proposed LMNN, the computation cost of the convolutional front-end can be offloaded into fabricated optical hardware, so as to get optimal energy and speed efficiency under current fabrication limitations. We adapt our innovations with three aspects: (1) large kernel re-parameterization, (2) meta-optic adaptation, and (3) model compression.

3.1 *Large kernel Re-parameterization*

To tackle the limitation of fabricating only the first layer in CNNs, we need to maximize the performance of the first layer, while it is feasible to adapt the fabrication processing. With significant progress in Vision Transformers (ViTs), the key contribution for the performance gained is largely credited to the large effective receptive field, which can be generated similarly by the depthwise convolution with large kernel sizes in CNNs. Therefore, we explore the feasibility of adapting large kernel convolution in (1) single-branch and (2) multi-branch settings. The overarching methodology for large kernel design can be delineated into two primary steps: (1) The deployment of stacked depthwise convolution layers to accommodate expansive convolution kernel receptive fields, as detailed in the "Single Branch Design" section; (2) The amalgamation of results from various large convolution layers, each offering distinct scales of view, elaborated in the "Multi-branch Design" section.

Single Branch Design. Inspired by [11], a large depthwise convolution kernel is equivalent to having the same receptive fields to a stack of small kernels. With the intrinsic structure of depthwise convolution, such a stack of kernel weights, can be compressed into a single operator. It is thus essential for the LMNN to maximize the model performance via a relatively simple meta-optic design, with a single compressed convolution layer. The compressed design further introduces fewer model FLOPs in the model inference stage. For conventional convolution operation, the convolution weight matrix $\boldsymbol{W} \in \mathbb{R}^{C_i \times C_o \times K_h \times K_w}$. The C_i and C_o are input and output channels of the convolution layer, respectively. K_h and K_w are height and width of convolution kernel. Note that we have an input patch x of size $H \times W$ and the output is y, we have conventional convolution as Eq. (1).

$$y = W * x, \tag{1}$$

where $y = \sum_{p=0}^{n_i} W_p * x_p$, $*$ represents convolution between matrices. For the input x, the computation time complexity will be $O(H \times W \times C_i \times C_o \times K_h \times K_w)$. For the depthwise convolution model, channels C_i in the convolution layer are separated along with the input data channels of x. The depthwise convolution follows Eq. (2)

$$y_i' = W_i * x_i, \tag{2}$$

where y_i is the ith channel of output y, W_i and x_i are the ith channel from Convolution weight W and input data x, respectively. The time complexity is $O(H \times W \times C_i \times K_h \times K_w)$. Normally, the input channel number equals the output channel number. We can infer the theoretical speed-up ratio r on model FLOPs between convention convolution and depthwise convolution following Eq. (3)

$$r = \frac{O(H \times W \times C_i \times C_o \times K_h \times K_w)}{O(H \times W \times C_i \times K_h \times K_w)} = O(C_i), \tag{3}$$

where C_i is the channel number of the convolution layer. Depthwise convolution has $C_i = C_o = C$. The depthwise convolution operation saves more FLOPs when the channel number is large compared with convention convolution.

Multi-branch design. Inspired by RepVGG [56] and RepLKNet [11], the multi-branch design demonstrates the feasibility of adapting large kernel convolutions (e.g., 31×31) with optimal convergence using a small kernel convolution in parallel. The addition of the encoder output enhances the large kernel convolution in the locality. According to the properties of convolution operation, the abstracted feature map from the parallel convolution path can be overlapped by learning different features. By using different convolution kernel sizes, the features from different scales of view are abstracted simultaneously.

We denote that output y' and input patch x use a two-branch convolution block W.

$$y_i' = W_1 * x + W_2 * x, \tag{4}$$

where W_1 and W_2 are two different convolution layers with different kernel sizes. For multiple parallel paths, the

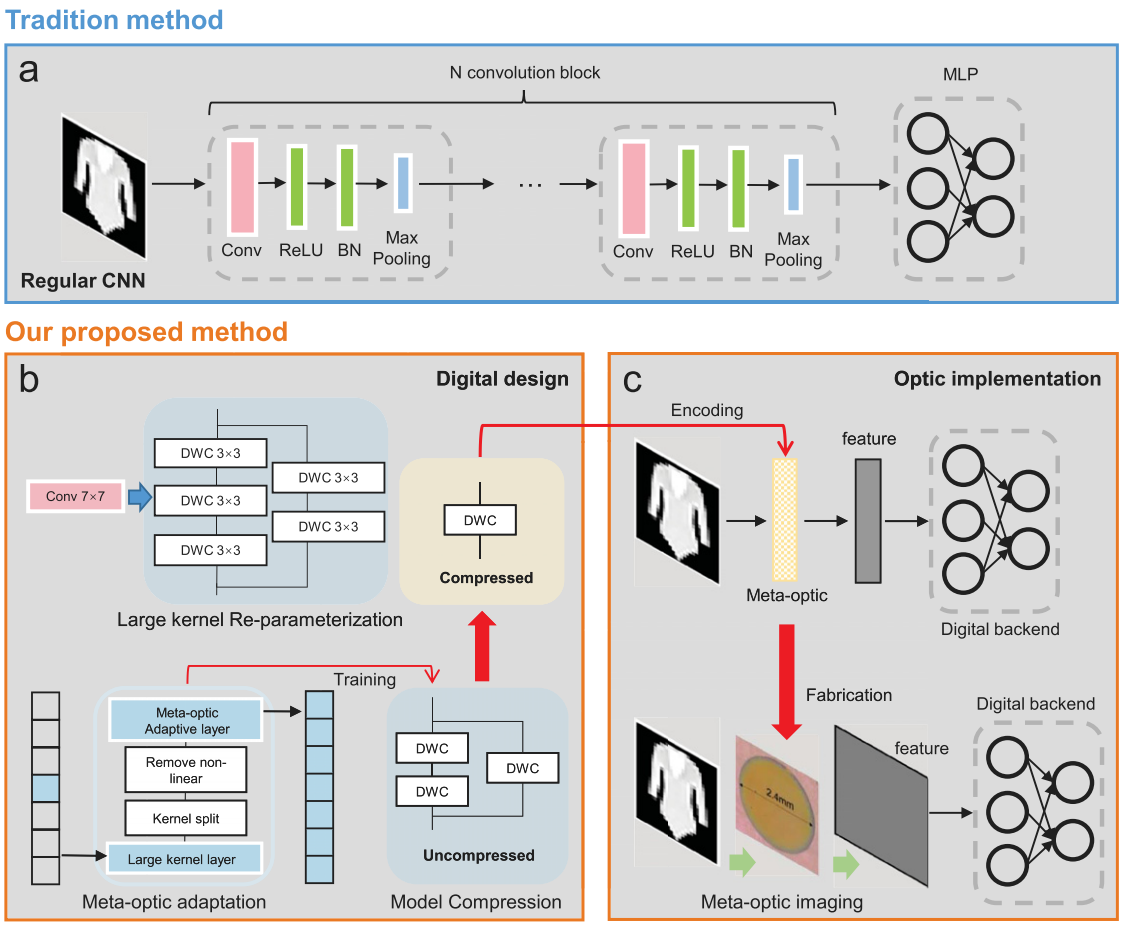

Figure 2. The upper panel (a) shows the conventional CNN model on the image classification task with Batch Normalization (BN) and Multilayer Perceptron (MLP). The lower panels (b) present our proposed LMNN method with digital design and optic implementation with depthwise convolution (DWC) layer. The large kernel re-parameterization efficiently achieves a large receptive field with a multi-branch multi-layer structure. Physical constraints are modeled via the meta-optic adaptation. The multi-branch multi-layer model is further compressed to a single-layer LMNN. (c) The digital design is fabricated as a real meta-optic device for inference. The red arrow shows the main pipeline to build the LMNN. The green arrow shows the image processing path in meta-optic imaging system.

N-branch convolution can be generated as Eq. (5).

$$y = \sum_{q=0}^{N} W_q * x. \tag{5}$$

According to the Eq. (5), output y has the feature map from multiple scales of views. The overlap of convolution output from different scales redistribute the feature map which is proved by [11] to have better performance.

3.2 *Meta-optic Adaptation*

To integrate the large kernel convolution design into meta-optic devices, we need to consider and model the physical restrictions explicitly in our model design, beyond the conventional digital training (Figure 2 and Figure 3). First, the weight in convolution kernel should be positive for fabrication. Second, the convolution layer that substitutes by metalens should be the first layer of the model. Third, in this study, metalens is designed at single wavelength (color). Thus, all RGB images are transferred to grayscale images. Fourth, for optic implementation purposes, the size of the convolution kernel is limited. Last, the channel number of the convolution layer is limited by the size of the optic device capacity.

Split kernel. To keep the model convolution kernel weight positive for the optic device implementation, we split the convolution kernel into two parts; positive weight and negative weight. As shown in Figure 4, the final convolution kernel results are the subtraction of the two feature maps from the positive and negative convolution kernels respectively. Positively and negatively valued kernels are achieved for incoherent illumination by using polarization multiplexing, combined with a polarization-sensitive camera and optoelectronic subtraction.

Removal of Non-linear Layer In traditional convolution operation, non-linear layer is typically added between the convolution layers. The non-linear layer, including batch normalization and activation layers (eg. ReLU) introduce the non-linear transformation to the model. However, the nonlinear operation is not included in our meta-optic

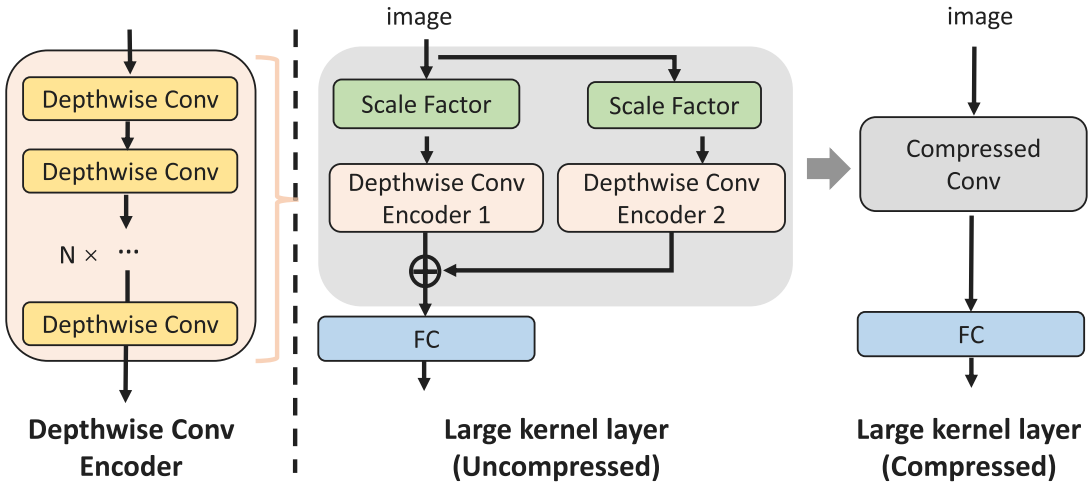

Figure 3. An overview of our proposed large convolution kernel block with re-parameterization is presented. Learnable scaling factors are employed to mimic the scaling function of batch normalization. In the inference stage, the block can be converted to a single convolution layer. 'FC' refers to fully connected layer in the figure.

device due to implementation cost. As shown in Fig. 4, the non-linear layers are removed from the parallel convolution branch and connected behind the large kernel convolution layer.

Non-negative Weight in Optic Kernel In traditional deep learning model, both positive weights and negative weights are stored. The meta-optic model implementation can only take positive kernel weights. Adaptation methods are applied to convolution model training to constrain model weight to a positive value. Four methods are introduced in model training; square of trigonometric functions, masking out the negative value, adding non-negative loss, and our proposed kernel split. The former three methods constrain convolution kernel weight positive in digital model training. The last, kernel split is achieved by meta-optic implementation.

Square of trigonometric function: Instead of directly updating the weight, we define weight as Eq. (6). The weight W_i stays positive and in range $[0, 1]$ whatever the value of θ. To clarify, we utilize the square of the trigonometric function to constrain weights within the [0, 1] range during the model training process. This approach offers distinct advantages over normalizing the weights at the inference stage. Specifically, the parameter θ can be adjusted freely across any range without introducing negative weights, which is especially beneficial for our meta-optic implementation.

$$W_i = \mathrm{Sin}^2(\theta_i) \tag{6}$$

Masking out the negative value: In the training process, the weight smaller than 0 is assigned as 0 manually after each iteration update.

Adding non-negative loss: To maintain the model weight positive, a non-negative weight loss is added to the loss function, which is defined as Eq. (7).

$$\mathrm{loss} = \sum(\mathrm{model.weight} < 0) \tag{7}$$

Bandwidth and precision. Due to the accuracy of the current fabrication of meta-optic, the optical inference might lose precision. As a result, the model bandwidth and weight precision should also be modeled during the training process. For example, PyTorch has a default 32-bit precision, which is not feasible for the LMNN. Thus, the quantize is employed to simulate the model performance when all digital neural networks are implemented with optic devices. Taking the noise in optic implementation into consideration, which will affect the model weights precision, we add the Gaussian noise to the digital convolution weight.

3.3 *Model Compression*

The stacked depthwise convolution and re-parameterization can potentially improve the model performance by learning with variance. The multi-layer structure can be regarded as multiple stacked depthwise convolution layers which make the model deeper. The multi-branch structure will make the model wider. It is obvious that the designed model is a complex structure. To save image processing time in the inference stage, the multi-layer structure can be squeezed into a single layer. In this paper, we only explore the squeezed convolution layer. To get the equivalent squeezed layer, a non-linear component should be eliminated. The non-linear layers such as activation function and BN are moved out of our squeezed block. The stacked convolution kernel follows the Eq. (8).

$$\begin{aligned} y &= (W_N * (W_{N-1} * \cdots (W_2 * W_1))) * x \\ &= W^* * x \end{aligned} \tag{8}$$

$$W^* = (W_N * (W_{N-1} * \cdots (W_2 * W_1))) \tag{9}$$

W^* is the equivalent weight to the stacked setting in Eq. (9). As the number of stacked convolution layers increases, the equivalent convolution kernel is larger. The equivalent kernel size k and the number of stacked 3×3

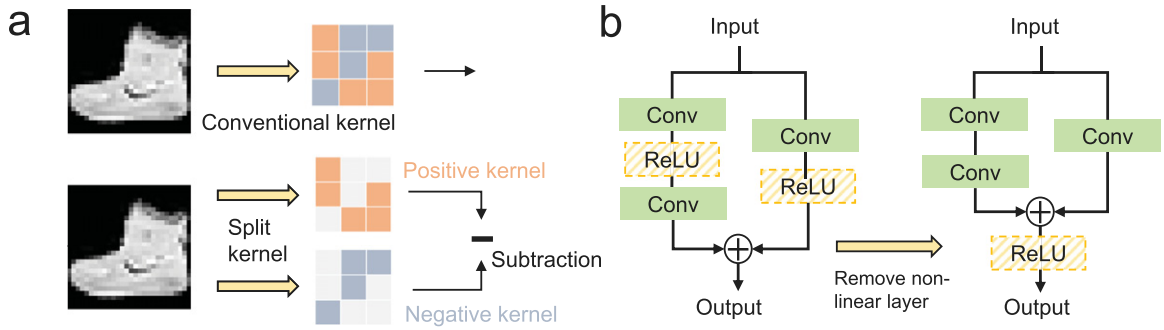

Figure 4. Adaptation for meta-optic implementation. (a) To implement the kernel with negative weight, we split the kernel into the positive kernel and negative kernel and subtract from their feature map. (b) The non-linear layer needs to be removed from the parallel convolution path.

convolution layer n follow Eq. (10).

$$k = 2 \times n + 1 \tag{10}$$

For example, two 3×3 convolution kernels are equivalent to a 5×5 convolution kernel. The multi-branch convolution layer can be compressed as shown in Fig. 3.

Since the convolution kernel value from the different parallel branches is equivalent to a single kernel by overlapping kernel, a multi-parallel convolution branch can be compressed into a single path.

4. DATA AND EXPERIMENTAL DESIGN

4.1 *Data Description*

Two public datasets, FashionMNIST [20] and STL10 [21], were employed to evaluate the performance of the proposed method on image classification tasks. For the FashionMNIST dataset, we employed 60,000 images for training and 10,000 images for testing. The images were grayscale images in the size of 28×28. FashionMNIST was inspired by the MNIST dataset, which classified clothing images rather than digits. We employed STL-10 as another cohort with a larger input image size (96×96). In our experiments, the RGB images in STL-10 were transferred to grayscale images due to the physical limitation in the LMNN.

4.2 *Large Kernel Re-parameterization*

We proposed the large re-parameterized convolution kernel design in our LMNN network to maximize the computational performance of the precious single metamaterial layer by (1) taking advantage of high-speed light computation, and (2) overcoming the physical limitations in an MNN implementation.

To evaluate the large re-parameterized convolution kernel on FashionMNIST, we constructed a naive model that consisted of a large re-parameterized convolution kernel block, a single fully connected layer, as well as non-linear components (ReLU activation, BN, and the softmax function). Different re-parameterization model structures were evaluated. To demonstrate the impacts of the size, the kernel was tested from 3×3 to 31×31. Besides the kernel size, we evaluated multiple numbers of parallel branches, from a single path to four paths.

4.3 *Meta-optic Model Adaptation*

The performance of the LMNN is fundamentally limited by physical restrictions. We provide the model simulation by modeling optic system limitations. Regarding model limitations, the convolution kernel is implemented with optical devices that can only have limited channels. To include the meta-optic devices in our network, the layer that is to be substituted should be the first layer of our model. The following model structure can be designed digitally. To validate model design on different sizes, deep neural networks with multiple convolution layers are implemented.

To simulate the noise in real meta-optic fabrication, we add random noise following Gaussian distribution. To test the impact of noise level, we simulate the noise amplitude range from 0.05 to 0.2. Considering the meta-optic implementation on the whole model for further research, we quantize the model weight.

In order to evaluate the non-negative weight effect, three methods are evaluated to constrain the model weight positive. “Sin” means weights are defined by square of sin function. “Mask out” is to eliminate the negative weight by screening out. Loss function is also used to define the model with positive weights.

The large kernel convolution design is validated on fabricated meta-optic devices. Based on the well-trained digital convolution kernel weight, meta-optic lenses are implemented and tested in real optic systems shown in Figure 5. The imaging system using a liquid-crystal-based spatial light modulator (SLM) was built. An incoherent tungsten lamp with a bandpass filter was used for SLM illumination. The feature maps extracted by the meta-optic were recorded by a polarization-sensitive camera (DZK 33UX250, Imaging Source) where orthogonally polarized channels are simultaneously recorded using polarization filters on each camera pixel. The algorithm was programmed based on Pytorch 1.10.1 and CUDA 11.0 with a Quadro RTX 5000/PCIe/SSE2 as the graphics cards.

4.4 *Model Compression Efficiency*

Through model compression, the model in inference stage alleviates the computation load with lighter weights. The fabricated convolution kernel by a meta-optic lens with the

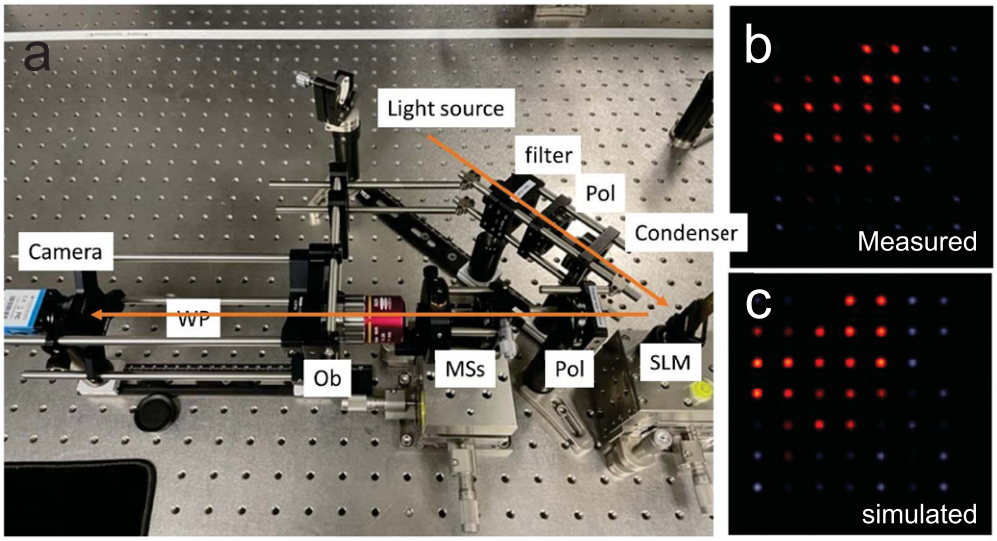

Figure 5. The meta-optic devices simulation and implementation platform. (a) Optic system for meta-optic lens test. The components in the figure are: Light source: Tungsten Lamp; Filter: Wavelength filter; Pol: Polarizer; SLM: Spatial light modulator; Condenser: Lens to focus light on the SLM; MSs: Metasurfaces; Ob: Objective lens. (b) Measured meta-optic kernel weight point spread function, used for optical convolution with the imaged object. (c) Theoretical meta-optic kernel weight point spread function by simulation.

Table I. Large re-parameterized convolution experiment results.

	FashionMNIST		STL-10	
	Model Conv	Test	Model Conv	Test
Naive model	3 × 3	0.8495	3 × 3	0.4500
RepLKNet [11]	7 × 7	0.9015	7 × 7	0.4993
			11 × 11	0.5241
RepVGG [56]	7 + 5 + 3	0.9081	7 + 5 + 3	0.5341
			11 + 9 + 7	0.5650
Depthwise conv [22]	3 dwc	0.9084	3 dwc	0.5509
			5 dwc	0.5935
Shufflemixer [26]	7 × 7	0.9047	7 × 7	0.5754
	11 × 11	0.9021	11 × 11	0.5878
SCConv [42]	7 × 7	0.8975	7 × 7	0.5230
	11 × 11	0.8969	11 × 11	0.5117
LMNN (Ours)	3 dwc + 2 dwc + 1 dwc	**0.9115**	5 dwc + 3 dwc + 1 dwc	**0.6120**

'dwc' refer to the depthwise convolution layer, convolution kernel size is 3 × 3

digital backend is assembled as the hybrid model. We test the model's inference time by feeding the same image and recording the model's processing time.

To test the optimal LMNN structure under the meta-optic fabrication limitation, the combination of layer numbers from one to five and channel numbers from nine to twenty. The model digital computation load (FLOPs) and the ratio of meta-optic is computed to find the model structure achieves optimal efficiency.

5. RESULT

In this section, we first evaluate our proposed large kernel network with a simple model structure, using FashionMNIST dataset and STL-10 dataset. We then evaluate the large kernel capability on complex CNNs with the same dataset.

5.1 *Large Re-parameterized Convolution Performance*

We evaluated the large re-parameterized convolution model on FashionMNIST and STL-10 datasets. As shown in Table I, the naive model with 7 × 7 convolution kernels has

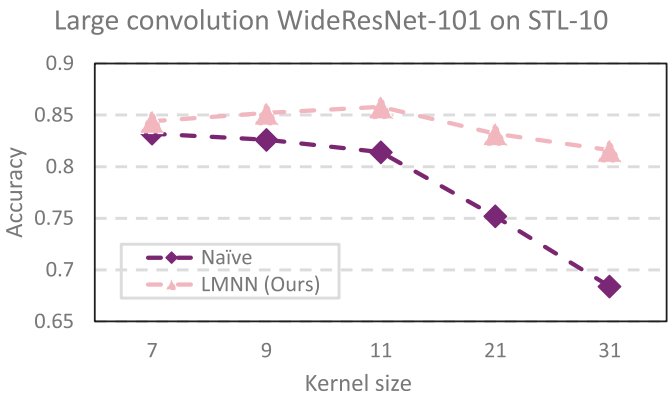

Figure 6. Performance of the Large Convolution WideResNet-101 model on STL-10 dataset. The model was assessed using varying convolution kernel sizes, ranging from 7×7 to 31×31. The LMNN consistently outperforms, highlighting the benefits of utilizing larger convolution kernels.

demonstrated better performance than that of 3×3. With structural re-parameterization, the model prediction accuracy further improves. Meanwhile, the model implemented with a depthwise convolution (DWC) layer outperformed the baselines with both small and large convolution kernels. Other SOTA model performances are included: Shufflemixer reaches 0.5878 with 7×7 kernel while SCConv performs better on 11×11 kernel (0.5230).

Our large kernel model was evaluated on STL10 dataset with a larger image size (96×96). As compared with performance on FashionMNIST (image size 30×30), the large kernel convolution model reveals greater improvements, as shown in Table I. The model with 11×11 kernel size has better accuracy (0.5341) compared with that of using 3×3 and 7×7. By integrating the DWC design, the model performance boosts from 0.5241 to 0.5935. Shufflemixer and SCConv were evaluated on STL10 with kernel size 7×7 and 11×11 and shows comparable model accuracy. Shufflemixer attained 0.9047 on 7×7 and SCConv attained 0.8975 on 11×11 kernel size. Our proposed large kernel block outperformed all SOTA approaches and achieved the best accuracy of 0.6015 with teacher model supervised training.

To further validate our large kernel with DWC design, we conducted experiments on more sophisticated models by replacing all convolution layers with the large re-parameterized convolution layers. Briefly, WideResNet-101 was used as complex model backbone [57]. Model performance is shown in Figure 6. By substituting the first convolution layer with a larger kernel size, the model performance improves from 0.94 to 0.96 when utilizing larger images (256×256 RGB).

5.2 *Performance of Model Adaptation*

To validate our large kernel design on the real metasurface fabrication model shown in Fig. 5, we implement a model trained on FashionMNIST with a large kernel design, utilizing a digital design for comparison. The digital convolution layer has 12 channels 7×7 convolution kernel which is the optimal kernel design under the current meta-optic implementation limit. As shown in Table II, the MNN demonstrates excellent consistency with the theoretical performance of a DNN.

Table II. Metasurface fabrication.

Method	Test
Digital Neural Network (DNN)	0.9015
Large Kernel MNN (LMNN)	0.8760

Due to meta-optic implementation limits, four adaptation methods are applied to constrain kernel weights to positive. According to the model performance, our proposed kernel split method shows superior performance over the common training strategies.

5.3 *Ablation Studies*

To evaluate the upper bound performance on FashionMNIST, a deep model structure is implemented and tested on FashionMNIST. The number of convolution layers in our model ranges from 1 to 5, and the channel number ranges from 9 to 30. The model performance is shown in Fig. 7(a). The model with more parameters shows a higher accuracy. Regardless of the meta-optic fabrication limitation, the meta-optic hybrid model achieves better performance.

To validate our model bandwidth and weight precision limit simulation, the results of the experiment are shown in Figure 8.

5.4 *LMNN Efficiency and Speed Evaluation*

To evaluate the model on both speed and computation load, we computed the model FLOPs except the large convolution layer and the FLOPs ratio of the layer implemented by meta-optic material. The model performance with different structures is shown in Fig. 7(b). The optimal model structure is at the top left corner in the shadow area. As shown, the model with 1 large re-parameterized convolution layer and 12 channels is the optimal structure. To show the speed advantage of our LMNN, the model inference time is

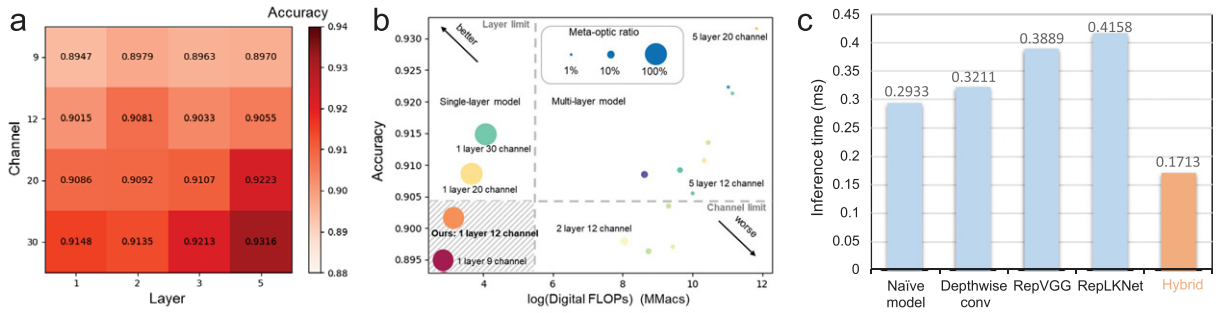

Figure 7. (a) The large re-parameterized convolution model performance with different layer numbers and channel numbers. (b) Large convolution kernel efficiency evaluation. The circle in different colors shows different convolution layer structures. The shadow area is the model structure that can be fabricated. The circle area shows the FLOPs ratio of the layer implemented by meta-optic material. x-axis is the model FLOPs except the layer to be fabricated. (c) Model inference time between the baseline digital model and hybrid model. The orange bar in the figure shows the time used in our model.

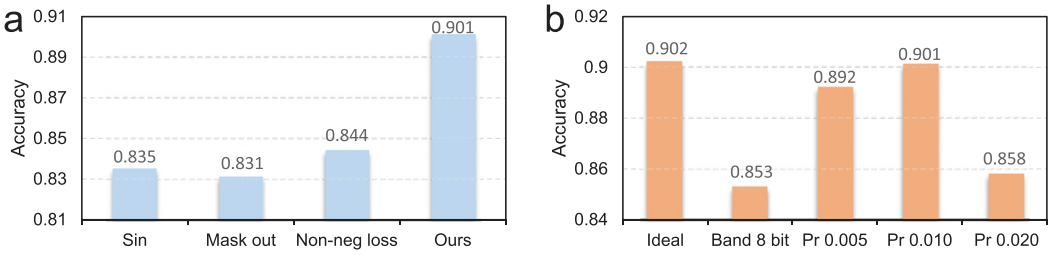

Figure 8. Plot of ablation study on LMNN. (a) Evaluating non-negative weight effect on model performance. (b) Measuring the effect of model bandwidth and weight precision effect on model prediction accuracy. 'Pr' in Figure (b) means precision.

recorded. From Fig. 7(c), the hybrid model shows a speed twice as fast as compared to the digital convolution model.

6. DISCUSSION

In this study, we present a convolution block with a large kernel design that generates larger receptive fields to maximize the digital capacity of LMNN. To validate the large kernel convolution design, we further applied the block to a complex model such as WideResNet-101. From the experiments, two important components contribute to the improvement of large kernel design from traditional 3×3 kernel size. First, the larger convolution kernel can get larger receptive fields. According to the target image size, the larger convolution kernel size is not the better. For FashionMNIST in size of 30×30, 7×7 is the best kernel size. For images from STL-10 dataset in size of 96×96, 11×11 kernel performed the best. Another interesting point is that the stacked depthwise convolution layers have equivalent computing operations to the single convolution layer with a larger kernel size. The multi-layer depthwise convolution and multi-branch structure expand the model capacity without parameter increase.

The proposed LMNN model bridges the disparity between natural objects and digital neural network analysis. Challenges in hybrid neural network design arise from the optical front-end, stemming from noise sources in the analog signals. These include stray light, detector interference, image misalignment due to optical inconsistencies, off-axis imaging aberrations, and fabrication flaws in the metalens and kernel layers. The system's bandwidth is constrained by the multi-channel lens, given the kernel layer's broadband nature. Optimizing the balance between bandwidth and aperture size is crucial for meta-optic systems. While the current optical approach mainly supports linear operations, future layers based on nonlinear media might facilitate activation functions. Even without these functions, refining the neural architecture can shift more linear tasks to the front-end. End-to-end model optimization ensures the meta-optic system effectively balances bandwidth and aperture considerations.

Since the large convolution kernel achieved superior performance on image classification tasks, more computer vision tasks have scope for improvement. For image segmentation task, it can be regarded as a pixel-level classification problem. The large convolution design can be applied to segmentation tasks. Object detection can be another choice for large convolution kernel applications. Different sizes of convolution kernel provide multiple fields of view. The views from multiple scales can abstract representation with more spatial information.

7. CONCLUSION

In this study, we introduced a large-kernel convolution block tailored for implementation on a meta-optic lens. Through model re-parameterization and multi-layer compression, we were able to efficiently condense intricate digital layers, making them compatible with the constraints posed by optical fabrication techniques. By explicitly incorporating the physical restrictions, we re-evaluated and refined the design of a metamaterial neural network. The proposed LMNN

demonstrated superior performance on FashionMNIST and STL-10 datasets, attributable to its expanded receptive fields. Notably, the incorporation of light-speed optical convolution led to reductions in computational latency and energy consumption. Our research underscores the efficacy of optimized digital modeling, presenting a strategic pathway for adapting to physical limits in future optic-digital hybrid designs.

ACKNOWLEDGMENT

H.Z., B.T.S. and J.G.V. acknowledge support from DARPA under contract HR001118C0015, NAVAIR under contract N6893622C0030 and ONR under contract N000142112468. Y.H. and Q.L. acknowledge support from NIH under contract R01DK135597. Y.H. is the corresponding author.

REFERENCES

[1] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Comput. **1**, 541–551 (1989).

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Commun. ACM **60**, 84–90 (2017).

[3] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," *2014 13th Int'l. Conf. on Control Automation Robotics & Vision (ICARCV)* (IEEE, Piscataway, NJ, 2014), pp. 844–848.

[4] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "Doubleu-net: A deep convolutional neural network for medical image segmentation," *2020 IEEE 33rd Int'l. Symposium on Computer-based Medical Systems (CBMS)* (IEEE, Piscataway, NJ, 2020), pp. 558–564.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th Int'l. Conf. Proc. Part III 18* (Springer, Cham, 2015), pp. 234–241.

[6] R. Chauhan, K. K. Ghanshala, and R. Joshi, "Convolutional neural network (CNN) for image detection and recognition," *2018 First Int'l. Conf. Secure Cyber Computing and Communication (ICSCCC)* (IEEE, Piscataway, NJ, 2018), pp. 278–282.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 779–788.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *Proc. IEEE/CVF Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2021), pp. 10012–10022.

[9] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *Proc. IEEE/CVF Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2021), pp. 568–578.

[10] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2022), pp. 11976–11986.

[11] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2022), pp. 11963–11975.

[12] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, M. Pechenizkiy, D. Mocanu, and Z. Wang, "More convnets in the 2020s: Scaling up kernels beyond 51×51 using sparsity," Preprint, arXiv:2207.03620 (2022).

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Preprint, arXiv:1409.1556 (2014).

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 770–778.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2015), pp. 3431–3440.

[16] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2017), pp. 4353–4361.

[17] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, and W. Liu, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence* (IEEE, Piscataway, NJ, 2020), Vol. 43, pp. 3349–3364.

[18] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," Preprint, arXiv:1811.12231 (2018).

[19] X. Cheng, P. Wang, C. Guan, and R. Yang, "Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion," *Proc. AAAI Conf. on Artificial Intelligence* (AAAI, Palo Alto, CA, 2020), Vol. 34, pp. 10615–10622.

[20] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," Preprint, arXiv:1708.07747 (2017).

[21] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," *Proc. Fourteenth Int'l. Conf. on Artificial Intelligence and Statistics. JMLR Workshop and Conf. Proc.* (JMLR, Cambridge, MA, 2011), pp. 215–223.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2017), pp. 1800–1807.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2015), pp. 1–9.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proc. IEEE Cconf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 2818–2826.

[25] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," *Proc. IEEE/CVF Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2019), pp. 3464–3473.

[26] L. Sun, J. Pan, and J. Tang, "Shufflemixer: An efficient convnet for image super-resolution," *Advances in Neural Information Processing Systems (NeurIPS 2022)* (2022), Vol. 35, pp. 17314–17326.

[27] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "Densenet: Implementing efficient convnet descriptor pyramids," Preprint, arXiv:1404.1869 (2014).

[28] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger, "CondenseNet: An efficient densenet using learned group convolutions," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2018), pp. 2752–2761.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," Preprint, arXiv:1704.04861 (2017).

[30] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2018), pp. 6848–6856.

[31] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Processing Magazine* (IEEE, Piscataway, NJ, 2018), Vol. 35, pp. 126–136.

[32] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," *Deep Learning and Unsupervised Feature Learning Workshop* (NIPS, San Diego, CA, 2011).

[33] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," *Int'l. Conf. on Machine Learning* (PMLR, Cambridge, MA, 2015), pp. 2285–2294.

[34] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," Preprint, arXiv:1507.06149 (2015).

[35] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in Neural Information Processing Systems* (NeurIPS, San Diego, CA, 2015), Vol. 28.
[36] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," *Proc. IEEE Int'l. Conf. Computer Vision* (IEEE, Piscataway, NJ, 2017), pp. 1398–1406.
[37] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," Preprint, arXiv:1412.6115 (2014).
[38] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 4820–4828.
[39] Z. Liu, Y. Wang, K. Han, W. Zhang, S. Ma, and W. Gao, "Post-training quantization for vision transformer," *Advances in Neural Information Processing Systems (NeurIPS 2021)* (2021), Vol. 34, pp. 28092–28103.
[40] J. Fang, A. Shafiee, H. Abdel-Aziz, D. Thorsley, G. Georgiadis, and J. H. Hassoun, "Post-training piecewise linear quantization for deep neural networks," *Computer Vision–ECCV 2020: 16th European Conf. Proc., Part II 16* (Springer, Cham, 2020), pp. 69–86.
[41] Y. Li, R. Gong, X. Tan, Y. Yang, P. Hu, Q. Zhang, F. Yu, W. Wang, and S. Gu, "BRECQ: Pushing the limit of post-training quantization by block reconstruction," Preprint, arXiv:2102.05426 (2021).
[42] J. Li, Y. Wen, and L. He, "SCConv: Spatial and channel reconstruction convolution for feature redundancy," *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2023), pp. 6153–6162.
[43] G. Zhou and D. Z. Anderson, "Acoustic signal recognition with a photorefractive time-delay neural network," Opt. Lett. **19**, 655–657 (1994).
[44] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing," Opt. Express **20**, 3241–3249 (2012).
[45] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," Opt. Express **20**, 22783–22795 (2012).
[46] S. Jutamulia and F. Yu, "Overview of hybrid optical neural networks," Opt. Laser Technol. **28**, 59–72 (1996).
[47] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," Sci. Rep. **2**, 287 (2012).
[48] D. Woods and T. J. Naughton, "Photonic neural networks," Nature Phys. **8**, 257–259 (2012).
[49] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," Optica **5**, 864–871 (2018).
[50] Y. Fang and M. Sun, "Nanoplasmonic waveguides: Towards applications in integrated nanophotonic circuits," Light Sci. Appl. **4**, e294–e294 (2015).
[51] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, "Deep learning with coherent nanophotonic circuits," Nature photonics **11**, 441–446 (2017).
[52] Y. B. Ovchinnikov, J. Müller, M. Doery, E. Vredenbregt, K. Helmerson, S. Rolston, and W. Phillips, "Diffraction of a released bose-einstein condensate by a pulsed standing light wave," Phys. Rev. Lett. **83**, 284 (1999).
[53] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," Science **361**, 1004–1008 (2018).
[54] J. George, R. Amin, A. Mehrabian, J. Khurgin, T. El-Ghazawi, P. R. Prucnal, and V. J. Sorger, "Electrooptic nonlinear activation functions for vector matrix multiplications in optical neural networks," *Signal Processing in Photonic Communications* (Optica Publishing Group, Washington, DC, 2018), p. SpW4G–3.
[55] M. Miscuglio, A. Mehrabian, Z. Hu, S. I. Azzam, J. George, A. V. Kildishev, M. Pelton, and V. J. Sorger, "All-optical nonlinear activation function for photonic neural networks," Opt. Mater. Express **8**, 3851–3863 (2018).
[56] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style convnets great again," *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2021), pp. 13733–13742.
[57] H. D. Kabir, M. Abdar, A. Khosravi, S. M. J. Jalali, A. F. Atiya, S. Nahavandi, and D. Srinivasan, "SpinalNet: Deep neural network with gradual input," *IEEE Transactions on Artificial Intelligence* (IEEE, Piscataway, NJ, 2022), pp. 1165–1177.