A Design Framework for Metasurface Optics-based Convolutional Neural Networks

CARLOS MAURICIO VILLEGAS BURGOS,[1] TIANQI YANG,[2] YUHAO ZHU,[2,3] AND NICKOLAS VAMIVAKAS[1,*]

[1]*University of Rochester, Institute of Optics, 275 Hutchison Road, Rochester, NY 14627, USA*
[2]*University of Rochester, Department of Computer Science, 2513 Wegmans Hall, Rochester, NY 14627, USA*
[3]*yzhu@rochester.edu*
[*]*nick.vamivakas@rochester.edu*

Abstract: Deep learning using convolutional neural networks (CNNs) has been shown to significantly out-perform many conventional vision algorithms. Despite efforts to increase the CNN efficiency both algorithmically and with specialized hardware, deep learning remains difficult to deploy in resource-constrained environments. In this paper, we propose an end-to-end framework to explore how to optically compute the CNNs in free-space, much like a computational camera. Compared to existing free-space optics-based approaches which are limited to processing single-channel (i.e., grayscale) inputs, we propose the first general approach, based on nanoscale metasurface optics, that can process RGB input data. Our system achieves up to an order of magnitude energy saving, simplifies the sensor design, all the while sacrificing little network accuracy.

1. Introduction

Deep learning (a.k.a., deep neural networks, or DNNs) have become instrumental in a wide variety of tasks ranging from natural language processing [1–3] and computer vision [4,5] to drug discovery [6]. However, despite enormous efforts both algorithmically [7,8] and with specialized hardware [9, 10], deep learning remains difficult to deploy in resource-constrained systems due to the tight energy budgets and computational resources. The plateau of current digital semiconductor technology scaling further limits the efficiency improvements, and necessitates new computing paradigms that could move beyond current semiconductor technology.

Optical deep learning has recently gained attention because it has the potential to process DNNs efficiently with low energy consumption. A lens executes the Fourier transform "for free" [11]. Leveraging this property, one can custom-design an optical system to execute convolution (or matrix multiplication) [12–15], since convolution can be performed with the help of Fourier transform operations. The effective convolution kernel is given by the optical system's point spread function (PSF).

Motivation and Challenge Existing free-space optical mechanisms can not handle layers with multiple input channels [16–19], which limits the applicability to a single layer with one single input channel, e.g., MNIST hand-written digits classification. Fundamentally, this limitation is because their systems could not provide different and independent PSFs for multiple frequencies of light, as they use elements such as digital micromirror devices (DMDs) [20] or diffractive optical elements (DOEs) [21] to engineer their system's PSF. DMDs modulate the amplitude of light, but impart the same amplitude modulation to all frequencies of light. Similarly, DOEs modulates the phase of incident light by imparting a propagation phase, but the phase profile imparted to one frequency component is linearly correlated with the phase profile imparted to another frequency component. Thus, these elements can not provide different and independent PSFs for multiple frequencies, and are limited to only single-channel convolution kernels.

Our Solution To enable generic convolution with multiple input channels and widen the

(a) Existing DNN-based vision system pipeline.

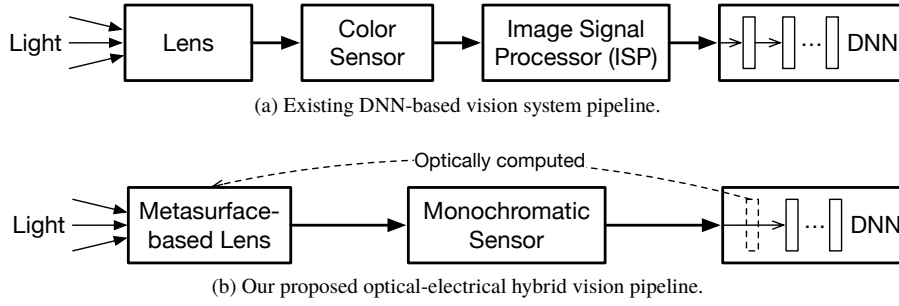(b) Our proposed optical-electrical hybrid vision pipeline.

Figure 1. Comparing our proposed system with conventional systems. Our system moves the first layer of a DNN into the optical domain based on novel metasurface-based designs. This hybrid system allows us to use a simple monochromatic sensor while completely eliminating the image signal processor (ISP), which is power-hungry and on the critical path of the traditional pipeline.

applicability of optical deep learning, we propose to use *metasurface* [22, 23] optics, which are ultra-thin optical elements that can produce abrupt changes on the amplitude, phase, and polarization of incident light. Due to the flexibility to shape light properties along with their ultra-compact form factor, metasurfaces are increasingly used to implement a wide range of optical functionalities. One such functionality, which we use on our work, is imparting a geometrical phase to circularly polarized light that interacts with the nano-structures (referred to as *meta-elements* from here on out) on the metasurface. The portion of incident light that gets embued with this geometrical phase also has its handedness changed [22, 24]. The power ratio between light embued with this geometrical phase and the total incident light is a function of frequency, and this function is referred to as the polarization conversion efficiency (PCE) [25].

Given a target convolution layer to be implemented optically, we purposely design the metasurface such that each of the meta-elements in a given unit cell has a unique PCE spectrum that responds to only one specific, very narrow frequency band corresponding to a particular input channel. In this way, metasurfaces let us apply different phase profiles to different input channels, which enables the system to have different PSFs for each channel, which effectively allows processing multi-channel inputs.

However, using metasurface optics for optical deep learning is not without challenges. Determining the metasurface parameters to produce a given PCE specification, known as the inverse design problem of a metasurface, is known to be extremely time-consuming, because it involves exploring billions of design points, each of which requires simulating the time evolution of the system by computing solutions to Maxwell's equations [26, 27]. Second, an optical layer when directly plugged into a DNN model will likely not match the exact results of the mathematical convolution due to optical design imperfections and sensor artifacts (e.g., noise).

We propose an *end-to-end* optimization framework that avoids the inverse design problem altogether while minimizing the overall network loss. We start from existing meta-element designs whose PCEs are close enough to our desired PCE specifications. We then accurately model the metasurface behaviors to integrate the metasurface into an end-to-end design framework, which co-optimizes the metasurface design with the rest of the network including the analog sensor and the digital layers in a DNN model. Critically, the end-to-end pipeline is differentiable, which allows us to generate a complete system design (metasurface optical parameters and the digital layer parameters) using classic gradient descent methods.

Contributions Leveraging the end-to-end design framework, we explore a system where the first layer in a DNN is moved to the optical domain; the image sensor captures the convolutional layer's output feature map, which is then processed in the digital domain to finish the entire network. Fig. 1 illustrates the idea of our system.

Our design has three advantages. First, since the sensor captures the feature map rather than the original RGB image, our system enables us to use a simple monochromatic sensor rather than a RGB sensor used in conventional systems, eliminating engineering efforts of many components such as the Color Filter Array (CFA).

Second, since we use a monochromatic sensor that captures the output feature map, we can avoid the often power-hungry computations in the image signal processor (ISP), which is required by conventional RGB sensors to correct sensor artifacts such as demosaicing and tone mapping. Avoiding ISP leads to significant overall energy reduction.

Finally, by moving the first layer into optics, we reduce the amount of computation left in the digital domain.

In summary, this paper makes the following contributions:

- We are the first to address generic (multi-channel) convolution in free-space optics. This is achieved by a novel use of metasurface optics to independently modulate the phase that is imparted to different frequency components of incident light.

- We present an end-to-end differentiable optimization framework, which allows us to co-train the optical parameters and digital layer parameters to maximize the overall accuracy and efficiency.

- We evaluate our system on multiple vision networks (AlexNet, VGG, and ResNet50). Our simulation results show that our system can reduce the overall system energy by up to an order of magnitude while sacrificing an accuracy loss of 1.9% on average (0.8% minimum).

2. Related Work

Integrated Photonics DNN Existing optical deep learning systems take two major approaches: integrated photonics [28–30] and free-space optics [16–18, 31]. Burgos et al. [32] propose a Multi-Layer Perceptron (MLP) implementation based on microring resonators (MRRs) [33]. Other implementations used different optical mechanisms such as Mach-Zehnder interferometers (MZI) [28, 30], but the general principle holds. Integrated photonics circuits are fundamentally designed to execute matrix multiplications. Convolutions must be first converted to matrix multiplication, and then mapped to the photonics substrate [30].

The advantages of integrated photonics is that it could be directly integrated into CMOS chips, to minimize the form-factor. Integrated photonics could potentially enable optical training [34]. The disadvantage of integrated photonics is two-fold. First, they fundamentally require coherent light (generated from a laser source) and thus could not directly operate on real-world scenes, where light is incoherent. Second, due to the quadratic propagation losses of integrated waveguides, the power consumption of the integrated photonics system could be extremely high when implementing realistic DNN models [32]. The power goes up to 500 W to execute an MLP layer with 10^5 neurons, which in turn limits the scale of the DNN to be implemented.

Free-Space Optical DNNs Motivated by the limitations of integrated photonics, recent research starts exploring free-space optics, which exploit the physical propagation of light in free space through custom lens to perform convolution. Chang et al. [16] demonstrate a 4-f system using DOEs to implement convolution layers, which support multiple output channels but does not support multiple input channels. Lin et al. [17] proposes a DOE-based free-space design, which supports a unique network architecture that is neither an MLP nor a CNN: it consists of the basic structures of an MLP, but has a limited receptive field typically found in CNNs. Miscuglio et al. [19] demonstrate a 4-f system where a DMD is placed in the Fourier plane to implement the convolutional kernels of a CNN layer by modulating only the amplitude of the incident light. However, this system is also limited to only being able to process single-channel inputs.
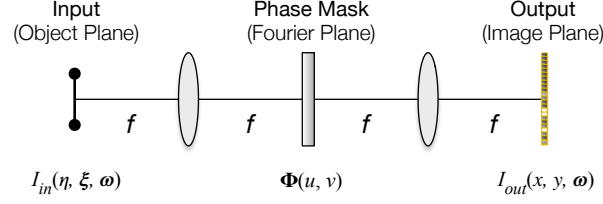
Figure 2. A typical "4-f" system.

Deep (Computational) Optics Implementing deep learning (convolution) in free-space optics is reminiscent of classic computational optics and computational photography, which co-designs optics with image processing and vision algorithms [35]. Recent progresses in deep learning has pushed this trend even further to co-train optical parameters and DNN-based vision algorithms in an end-to-end fashion. Examples include monocular depth estimation [36–39], single-short HDR imaging [40, 41], object detection [36], super-resolution and extended depth of field [42].

Computing in Metasurface Optics Lately, metasurface optics have been used for various computation tasks through co-optimizing optics hardware with backend algorithms. Lin et al. [43] uses a simple ridge regression as an example backend while our work employs a neural network. Lin et al. focuses on exploiting fullwave Maxwell physics in small multi-layer 3D devices for non-paraxial regime, and enhanced spatial dispersion, spectral dispersion and polarization coherence extraction. In contrast, our work focuses on exploiting large-area single-layer metasurfaces for computer vision tasks.

Colburn et al. [44] uses metasurface for convolution, too. To support multiple input channels, however, they use three separate laser sources, each going through a separate 4-f system, whereas we leverage polarization to support multiple input channels with one 4-f system, potentially simplifying the design and the overall form factor of the system.

3. Fundamentals of Free-Space Optics

A typical setup for free-space optical DNN is similar to computational cameras [35, 45], where a custom-designed lens directly interrogates the real-world scene as an optical input signal. The lens executes the first convolution or fully connected layer of a DNN; the layer output is captured by an image sensor and sent to the digital system to execute the rest of the DNN model. This setup is a natural fit for machine vision applications where the inputs are already in the optical domain and have to be captured by a camera sensor anyways.

Free space imaging scenarios can be modeled by assuming spatially incoherent light and a linear, shift-invariant optical system. Under these assumptions the image formed by an optical system (I_{out}) can be modeled as a convolution of the scene, i.e., input signal (I_{in}), with the point spread function (PSF) of the optical system [11]:

$$I_{out}(x, y, \omega) = I_{in}(\eta, \xi, \omega) \star \mathbf{PSF}(\eta, \xi) \tag{1}$$

where $\star$ denotes convolution, $I_{in}(\eta, \xi, \omega)$ denotes the intensity of input pixel (η, ξ) at frequency ω, and $I_{out}(x, y, \omega)$ denotes the intensity at output pixel (x, y) at frequency ω.

An optical convolution system is typically implemented using a so-called "4-f" system [11] as illustrated in Fig. 2, which consists of two lenses, each with focal length f. The first lens is placed one f away from the object plane, producing a Fourier plane another f after the first lens. The second lens is placed another f from the Fourier plane, producing an inverted conjugate image plane, which is in total 4-f from the original object plane. An image sensor is placed at the image plane to capture the output for future processing (e.g., activation and subsequent layers)

in the electronic domain. The reason why the back focal plane of the first lens is referred to as Fourier plane is because the electric field on that plane can be expressed in terms of the Fourier transform of the electric field on the front focal plane. Mathematically speaking, let $U_i(\eta, \xi)$ be the field in the input plane of the 4f system, where (η, ξ) denote the spatial coordinates in that plane. Also, let $U_F(u, v)$ be the field in the Fourier plane of the 4-f system, where (u, v) denote the spatial coordinates in that plane. Then, the field in the Fourier plane is expressed in terms of the field in the input plane as:

$$U_F(u, v) = F_i \left(\frac{u}{\lambda f}, \frac{v}{\lambda f} \right), \tag{2}$$

where λ is the wavelength of the incident light, f is the focal length of the lenses in the 4-f system, and $F_i(f_\eta, f_\xi)$ is the Fourier transform of $U_i(\eta, \xi)$, with (f_η, f_ξ) denoting spatial frequency components.

The PSF of the optical system is dictated by the *phase mask* placed at the Fourier plane. The phase mask alters the PSF by imparting certain phase change to the incident light. It achieves this by introducing a phase term to the field on the Fourier plane, meaning it multiplies $U_F(u, v)$ by $\exp(i\Phi(u, v))$ (here the imaginary unit is denoted as i) .The phase change induced by a phase mask is called the phase mask's *phase profile*, which is often determined by the geometric properties of the meta-elements on the phase mask (e.g., dimension, orientation, thickness). Our goal is to find the phase profile Φ to generate a PSF that matches a target image that contains the intended convolution kernel W. This can be formulated as an optimization problem:

$$\min_{\Phi} || W - \mathbf{PSF}(\Phi)||. \tag{3}$$

Without the phase mask, the image captured by the image sensor simply reproduces the input I_{in}. With the optimized phase mask, the output image is naturally the result of the convolution in Eq. (1).

The output of a generic convolutional layer is given by:

$$I_{out}^j(x, y) = \sum_{i=1}^{C_{in}} I_{in}^i(x, y) \star W_{j,i}(x, y), \quad \forall j \in [1, C_{out}]. \tag{4}$$

The optical convolution in Eq. (1) can be seen as a simplification of Eq. (4) whenever the number of input channels (C_{in}) and the number of output channels (C_{out}) of the convolutional layer are both equal to 1. Supporting multiple output channels (i.e., multiple convolution kernels) is achieved by designing a PSF that spatially tiles the C_{out} convolution kernels on the same plane; each tile corresponds to a convolution kernel $W_{j,\cdot}$ [16]. As the input image is convolved with the system's PSF, it is naturally convolved with the different convolution kernels, effectively generating multiple output channels.

However, supporting multiple input channels is a fundamental challenge unaddressed in prior work. Essentially, the optical system must have multiple PSFs each of which must apply to *only a particular frequency* (i.e., channel) of the incident light. This is inherently unattainable in prior works that use diffractive optical elements (DOE) [21] to implement the phase mask. This is because the phase profile that is implemented for one frequency is a scalar multiple of the one that is implemented for other frequencies. As such, it is not possible to create independent phase profiles for the different frequency components of incident light. To overcome this limitation, we instead propose to implement the phase mask using geometrical phase metasurfaces, which we describe in detail next.
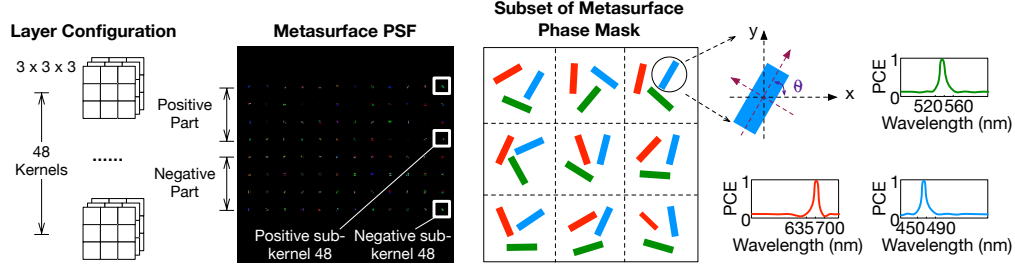
Figure 3. The target PSF for a convolutional layer that has 48 kernels, where each kernel has 3 channels and each channel has 3×3 elements. The PSF is arranged as 12×8 tiles, with 1 positive and 1 negative tile for each kernel. Each tile is 3×3. Each tile has 3 channels, corresponding to different frequencies of incident light. On the metasurface, each meta-element has a unique in-plane rotation θ. Each unit cell in the metasurface contains 3 kinds of meta-elements with distinct geometric parameters, and each kind is associated to a different channel. All the meta-elements associated to the same channel collectively provide a PSF that matches the corresponding channel of the convolution kernels. Meta-elements associated to the same channel have the same PCE spectrum, and in the ideal case, the corresponding PSF is applied to the associated input channel only, without cross-talk.

4. Metasurface-based Optical Convolution

The goal is to design the phase mask in a way that different PSFs are applied to different input channels. Unlike DOEs that modulate only the phase of the incident light by applying a propagation phase, metasurfaces are able to also introduce a geometrical phase. Our key idea is to use different kinds of meta-elements that embue a geometrical phase to incident light of different frequency components. That way, the metasurface introduces different phase profiles to each of the frequency components (channels) of incident light. In the following, we first describe the phase modulation assuming only one input channel; and then discuss the case with multiple input channels, which allows us to attain generalized free-space optical convolution with multiple input and output channels.

4.1. Single-channel case

Our metasurface phase mask consists of a set of meta-elements, each of which induces a particular phase change to the incident light. Physically, we use the Pancharatnam-Berry (P-B) phase [22, 24]. P-B phase comes from the spin rotation effect that arises from the interaction between circularly-polarized light and an anisotropic meta-element [46]. Specifically, a meta-element that is geometrically symmetric with respect to its main axes and that has an in-plane orientation angle θ induces a geometric phase 2θ on the incident light [46, 47]. Therefore, for a phase mask that consists of $M \times N$ meta-elements, each having an in-plane rotation of $\theta_{m,n}$, the phase profile Φ of the phase mask is also a $M \times N$ matrix, where each element is $2 \times \theta_{m,n}$.

In a 4-f system (with a DOE or a metasurface), the PSF relates to the phase profile (Φ) of the phase mask according to the following equation:

$$\mathbf{PSF}(\Phi) = \left| \mathcal{F}^{-1} \left\{ e^{i\Phi} \right\} \right|^2, \tag{5}$$

where $\mathcal{F}^{-1}$ denotes the inverse 2D Fourier transform, and $e^{i\Phi}$ acts as the system's Transfer Function [11]. Thus, finding the optimal phase profile is formulated as the following optimization problem:

$$\min_{\Phi_k} || W_k - |\mathcal{F}^{-1} \left\{ e^{i\Phi_k} \right\} |^2 ||_F^2, \tag{6}$$

where $|| \cdot ||_F$ denotes the Frobenius norm, W_k denotes the kth channel of the target PSF that encodes the convolution kernels, and Φ_k denotes the discretized 2D phase profile of channel k.

When performing the phase optimization process, we use a discretized version of the target PSF, encoding it in a numerical array of $N \times N$ pixels. During this phase optimization process, we iteratively update the numerical array Φ_k of $N \times N$ pixels that contains the phase profile. The optimization process has the objective of making the PSF yielded by Φ_k be as close as possible to the target PSF that encodes W_k. Once we obtain the optimized numerical array Φ_k, we can do the mapping between the (row, column) positions in the numerical phase array and (u, v) positions in the plane of the phase mask. For this, we remember that each point (u, v) in the Fourier's plane (where the phase mask lies) is associated with a point (f_η, f_ξ) in the frequency space domain given by $(f_\eta, f_\xi) = \left(\frac{u}{\lambda f}, \frac{v}{\lambda f} \right)$, as can be seen in Eq. (2). Additionally, we know that the largest (absolute value of) spatial frequency component of the input image that makes it into the Fourier plane is given by one half of the sampling rate of the display in the input plane, known as the Nyquist rate [11]. That is:

$$\left| f_\eta \right|_{\max} = \left| f_\xi \right|_{\max} = \frac{1}{2d}, \tag{7}$$

where d is the pixel pitch of the display in the input plane that projects the input image into the optical system. Given this, the size of the spatial frequency domain is $1/d$, and since the phase mask needs to cover it all, the size of the phase mask needs to be $\frac{\lambda f}{d}$. We then divide the region of $\frac{\lambda f}{d} \times \frac{\lambda f}{d}$ into $N \times N$ subregions (or "pixels"). Thus, the size of these "phase pixels" is given by:

$$\Delta u = \frac{\lambda f}{N d}. \tag{8}$$

After we have obtained the numerical values of the elements in the optimized Φ_k array, we simply divide them by 2 to obtain the value of the in-plane orientation of each meta-element on the metasurface. We would like to note that the objective function in Eq. (6) is differentiable. Thus, phases (and therefore, the values of the meta-elements' orientations) could be co-optimized with the rest of DNN in an end-to-end fashion, as we will show in Sec. 5.

Now, we'll discuss how we design the target PSF that encodes the network's convolution kernels. Since PSF physically corresponds to light intensity [11] and there is no negative light intensity, the negative values in a kernel cannot be physically implemented. A mathematical explanation is that the PSF is the square form of an expression, as shown in Eq. (5), and thus must be non-negative. To address this limitation, we use the idea of Chang et. al. [16] by splitting a kernel into two *sub-kernels*; one contains only the positive portion and the other contains only the (absolute value of the) negative portion of the kernel. The final kernel is calculated by subtracting the negative sub-kernel from the positive sub-kernel.

As a result, for a layer that has K kernels, the target PSF contains $2K$ tiles that are used to represent these kernels. Fig. 3 shows the target 3-channel PSF for a layer that has 48 kernels, each of dimension 3×3×3. The PSF has 96 tiles, where the ones in the top half are the positive sub-kernels, and the ones in the bottom half contain the are sub-kernels. Each tile is 3×3, which corresponds to the 3×3 elements in one channel of a 3-channel sub-kernel. In order to yield this target PSF, the in-plane rotation of every meta-element on the metasurface is optimized from Eq. (6). The three meta-elements in a unit cell of the metasurface are used to yield the three channels of the target multi-channel PSF, which we discuss next.

4.2. Multi-channel case

With what was discussed above, we are able to use the in-plane rotations of the meta-elements to engineer the PSF to match any given 2D matrix (i.e., a channel in the target kernel). When

going from the single-channel case to the multi-channel case, we first note that the size of the "phase pixels" will depend on the wavelength λ according to Eq. (8). This means that the pixels (regions in the metasurface that impart a given phase value to incident light) associated with different channels will have different sizes. We address this by noting that for our choice for the values of f, d, and λ range, the size of the metasurface phase pixels for all channels will be larger than the size of the metasurface's unit cells (the former being in the order of microns and the latter being in the order of hundreds of nanometers). This means that the phase pixels of any given channel are filled with multiple unit cells (each of which contains one of each kind of meta-element associated to the different channels). Since the orientation of each meta-element in the unit cell is independent to that of the others, we can essentially multiplex the phase profiles associated to the different channels on the same metasurface without issues.

The remaining challenge is making sure that the PSF that is yielded by the set of meta-elements of one kind should only be applied to light of the corresponding channel (frequency component) in order to correctly implement the convolution layer. That is, we must minimize cross-talk effects that make meta-elements apply the geometrical phase to a portion of the light of the other channels. We make use of the fact that each meta-element acts as a phase retarder, which makes a portion of incident circularly-polarized light to change its handedness and get embued with a geometrical phase. This portion is dependent on the wavelength of incident light, and is the polarization conversion efficiency (PCE). Effectively, when incident light from the input image interacts with the set of meta-elements that has a given PCE spectrum, the corresponding intensity profile in the output plane (given by the convolution between the input intensity profile and the PSF yielded by this set of meta-elements) is weighted by PCE(ω) at frequency ω. With the PCE taken into account, the contribution of intensity in the output (capture) plane that is attributed to an input channel c can be expressed as:

$$I_{con}^{c}(x, \ y, \omega) = \sum_{c'=1}^{C_{in}} \mathbf{PCE}^{c'}(\omega)(\mathbf{PSF}^{c'}(\eta, \ \xi) \star I_{in}^{c}(\eta, \ \xi, \ \omega)), \tag{9}$$

where channel c of the input image is affected by the PSFs c' yielded by the different meta-element sets in the metasurface, which is where the sum over PSF channels c' comes from.

In the ideal case, the PCE is a delta function that is 1 for a very narrow band, e.g., red wavelength, and 0 for all other wavelengths. That way, the PSF carried by the meta-elements will effectively convolve only with one particular input channel. Fig. 3 shows an example that supports 3 input channels (e.g., natural light from the real-world). The key is that each unit cell consists of three meta-elements, each carrying a unique PCE. When white light containing the RGB wavelengths (channels) is incident on the phase mask, the red channel will acquire the phase imparted by the meta-element associated to the red-channel only, mimicking the convolution between the red channel of the input and the corresponding PSF; the same process applies to green and blue wavelengths as well. The output is naturally the superposition (sum of intensity) of the three convolution outputs.

In reality, the metasurface design is regulated by fabrication constraints and, thus, the actual PCE will never be an ideal delta function, which leads to cross-talk, i.e., the PSF of a channel i will be convolved with input channel j. Therefore, it is critical to optimize the end-to-end system to minimize the impact of cross-talk, which we discuss next.

5. End-to-End Training Framework

Fig. 4 shows the end-to-end system setup, from light in the real-world scene (or equivalently a RGB microdisplay) to the network output. It consists primarily of three components: a metasurface-based optical system that implements the first convolution layer of a DNN, an image sensor that captures the output of the optical convolution, and a digital processor that finishes the
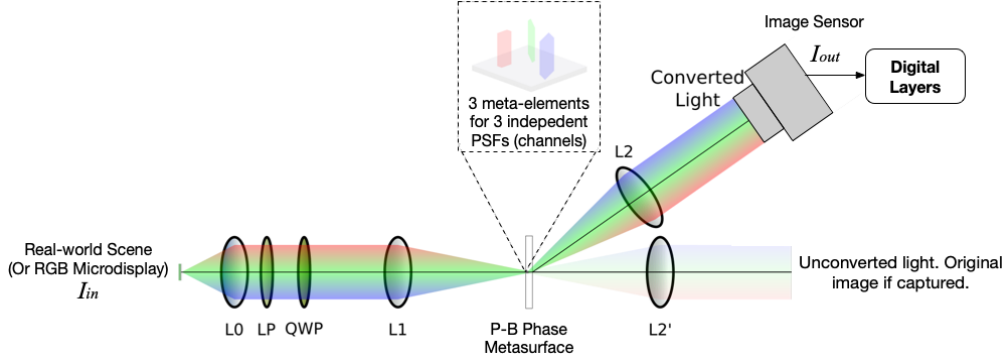
Figure 4. Hardware system. Light from the real-world (or a microdisplay) is collimated by lens L0, and then goes through a linear polarizer (LP) and a broadband quarter-wave plate (QWP) so that light becomes circularly-polarized. The plane of the LP is the input plane of the $4f$ system. A metasurface phase mask is placed at the Fourier plane. A portion of the light gets the engineered phase shift and has its polarization handedness switched. This "converted" portion of the light gets sent to lens L2, via a phase "grating" [48] added to the phase profile, while the unconverted light is sent to lens L2' on the optical axis. L2 is at a distance f from the metasurface. Finally, light is detected by the image sensor placed at a distance f after L2.
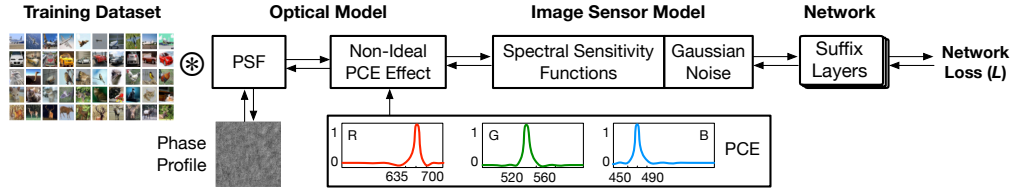

Figure 5. End-to-end training framework. The polarization conversion efficiency profile is network-independent, and is obtained from real metasurface optics fabrications. The phase profile can be optimized using the training data and fine-tuned with the DNN model suffix layers (i.e., layers implemented in the digital domain) together in an end-to-end fashion.

rest of the network, which we call the "suffix layers."

In order to maximize the overall network accuracy, we co-optimize the optical convolution layer with digital layers in an end-to-end fashion. Fig. 5 shows the end-to-end differentiable pipeline, which incorporates three components: an optical model, a sensor model, and the digital network. The phase profile (Φ) in the optical model and the suffix layer weights (W) are optimization variables. The loss function L that is used to measure this system's performance is the same as what can be used with a conventional all-digital neural network, like the log loss function.

The input to the system is light, which in practical applications are from the real-world scene. In our work, we envision a RGB microdisplay displaying images from datasets to emulate the real-world scene as is done in prior work [16]. Thus, channel c in the input is expressed as:

$$I_{in}^c(x, y, \omega) = \mathbf{SPD}^c(\omega)I_{in}^c(x, y), \qquad (10)$$

where $\mathrm{SPD}^c(\omega)$ is the spectral power distribution of color channel c in the input light at frequency ω, and $I_{in}^c(x, y)$ is the light intensity of pixel (x, y) in channel c and is directly read from the RGB image in the training data.

In the forward pass during training (and also inference), the PSF generated from the phase

profile (Φ) is first convolved with the input image from a training dataset under the impact of the PCE as described by Eq. (9). The convolution output is captured by the image sensor, which we model by its spectral sensitivity function (SSF). In addition, image sensors introduce different forms of noise, which we model as a Gaussian noise $N(\mu, \sigma^2)$ based on the central limit theorem. As a result, the output signal yielded by the sensor is given by:

$$I_{out}(x, \ y) = \sum_{c=1}^{C_{in}} \int \mathbf{SSF}(\omega)I_{con}^c(x, \ y, \ \omega)d\omega + N(\mu, \sigma^2). \tag{11}$$

Note that the sum across different input channels, required by correctly implementing a convolution layer, is realized in our design "for free" by superimposing the intensity profile of the different frequency components that are incident on the detector. The total light intensity of at the output plane is naturally the (weighted) sum of the intensities of individual channel components.

Plugging Eq. (10) into Eq. (9) and plugging Eq. (9) into Eq. (11) yields the sensor output, which becomes the input to the digital suffix layers to then produce the network's output and loss. The sensor output can then be expressed as:

$$I_{out}(x, \ y) = \sum_{c}^{C_{in}} \left[\sum_{c'}^{C_{in}} \left(\int \mathbf{SSF}(\omega)\mathbf{PCE}^{c'}(\omega)\mathbf{SPD}^c(\omega)d\omega \right) \mathbf{PSF}^{c'}(\eta, \xi) \right] \star I_{in}^c(\eta, \ \xi) + N(\mu, \sigma^2). \tag{12}$$

The above expression can be simplified into the form:

$$I_{out}(x, \ y) = \sum_{c}^{C_{in}} \left[\sum_{c'}^{C_{in}} A_{c'c}\mathbf{PSF}^{c'}(\eta, \xi) \right] \star I_{in}^c(\eta, \ \xi) + N(\mu, \sigma^2), \tag{13}$$

where $A_{c'c} = \int \mathbf{SSF}(\omega)\mathbf{PCE}^{c'}(\omega)\mathbf{SPD}^c(\omega)d\omega$, and we can define a *cross-talk matrix* whose elements are $A_{c'c}$ for the different values of c' and c. Note that the convolution layer mapped to metasurface, as described by Eq. (11), is differentiable, and thus enables end-to-end training using classic gradient descent-based optimization methods. In doing so, the suffix layers can be optimized to compensate for the effects introduced by cross-talk. During back-propagation, the phase profile (Φ) and the digital layer weights (W) are optimized. Note that the PCEs of the meta-elements are network-independent, and are directly obtained from fabrication. They are kept fixed throughout training.

Practical Considerations Since different sub-kernels are tiled together for the ease of training, the outputs of different sub-convolutions (i.e., convolution between a sub-kernel and the corresponding channel in the input) will be tiled too. We must make sure that the different sub-convolution outputs do not overlap so that we can assemble them (digitally) to form the actual convolution output. To that end, we pad each sub-kernel with space between cells in the phase mask. For instance in the illustrative example shown in Fig. 3, we pad each 3×3 sub-kernel by 8 pixels each side to form a 19×19 matrix, effectively separating neighboring sub-kernels in the phase mask by 16 pixels. In our work, we pad the sub-kernels by 20 pixels each side to form a 43×43 matrix, effectively neighboring sub-kernels by 40 pixels. In summary, given the kernels in a target convolution layer, we first separate each kernel into two sub-kernels, tile all the sub-kernels on the same plane, and then add padding to form the actual target PSF matrix, for which we then apply the phase optimization described in Eq. (6) to derive the required phase profile, and thus the in-plane rotation of each meta-element on the metasurface.

Monochromatic Sensor It is important to note that our system design lets us use a simple monochromatic sensor rather than a conventional RGB sensor. This is because of two reasons. First, the results of all output channels lie completely within a 2D plane, because the tiles of different convolution kernels and of different channels within a kernel are tiled in a 2D plane

(a) PCEs of the three meta-element designs.

(b) The spectral power distributions of the microdisplay and the spectral sensitivity function of the image sensor modeled in our computational experiments.
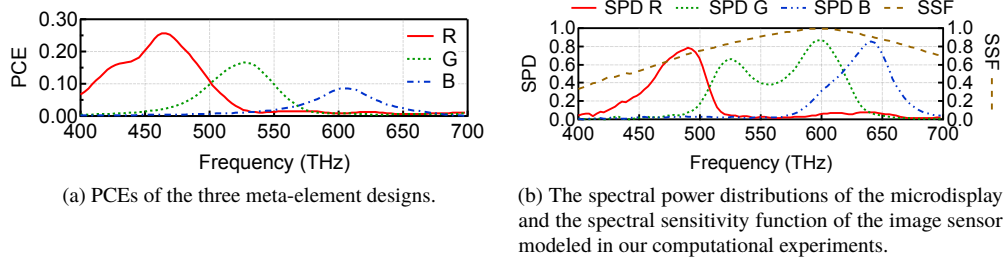
Figure 6. Parameters of hardware components. We use these to compute the cross-talk matrix, which is necessary to account for the effects of the cross-talk between channels.

(Fig. 3). Second, since the monochromatic sensor integrates the intensity of all incident frequency components, this naturally emulates the sum over channels that needs to be performed when calculating the output of a digital convolutional layer, as described by Eq. (4).

6. Evaluation

6.1. Methodology

Networks and Datasets We implement the end-to-end training framework in TensorFlow. We differentiate the metasurface optics layer (from phase profile to the target kernel) and integrate the differentiation into the back-propagation process. We evaluate on three networks: AlexNet [49], VGG-16 [50], and ResNet50 [51], all on the CIFAR-100 dataset [52].

Training Details To accelerate convergence, we first train the entire network in three separate steps: 1) train the entire network digitally, 2) use the first layer's weights to optimize the phase mask, and 3) finally the optimized phase mask and the pre-trained suffix layers are then fine-tuned end-to-end until convergence. The phase optimization process consists on running a gradient descent algorithm to minimize the loss function shown in Eq. (6). In the Supplementary Information, we provide further details about the derivation of the expression of the gradient of this function, as well as a way for it to be computed efficiently. In all networks, the results converge after 2,000 iterations of the gradient descent cycle. The phase masks and suffix layers' parameters are then end-to-end fine-tuned with the suffix layers for 50, 150, and 50 epochs for AlexNet, VGG, and ResNet, respectively.

Hardware Modeling Details The metasurface designs we use are from Wang et al. [25], which provide the detailed design specifications of three kinds of meta-elements, each of which is designed to have their highest response at red, green, and blue bands of the visible spectrum. From the design specifications, we run the well-known MEEP package [53, 54] for electromagnetics simulations to generate the corresponding PCEs of the three kinds of meta-elements, which we use in our design and simulation. Fig. 6a shows the PCE of the three kinds of meta-elements. While the PCEs are not ideal delta functions, they do provide the highest responses in distinct bands.

We use the Sony ICX274AL image sensor, whose SSF we adapt from the design specification [55]. We use the eMagin OLED-XL RGB microdisplay from eMagin, whose SPDs we obtained by contacting the display manufacturer. Fig. 6b shows the normalized SPDs of the microdisplay pixels (left *y*-axis) and the SSF of the sensor (right *y*-axis). Comparing the SPDs and the PCEs in Fig. 6a, the PCE and SPD peaks are close to each other (e.g., both the red PCE and the red light from the microdisplay peak at around 450 THz to 500 THz). We use these PCE and SPD functions to calculate the cross-talk matrix, which is a necessary part in the system's optimization process so that it can account for cross-talk effects.
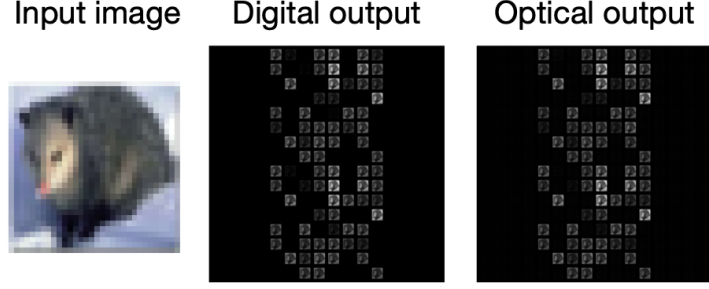
Figure 7. Comparison between the output of a convolution layer when implemented digitally, and when implemented optically. The optical output is simulated, taking the effects of cross-talk between channels into account

(a) Accuracy comparison using actual metasurface designs.

(b) Accuracy loss of Step 3 (our design) from Step 1 (fully digital) when using ideal and non-ideal PCEs.
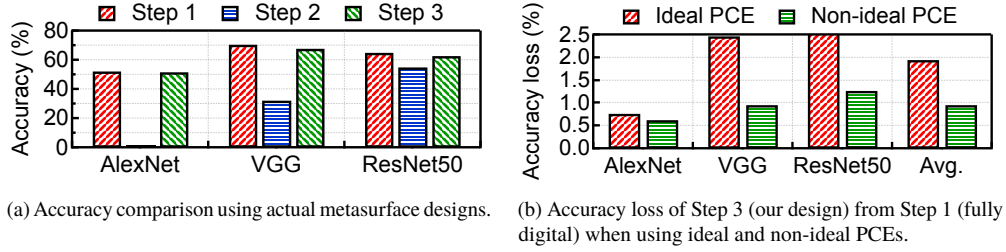
Figure 8. Accuracy results obtained after running the pipeline on different datasets.

6.2. Network Accuracy and Efficiency

Phase Optimization We find that our phase optimization allows us to generate a PSF that resembles the target kernel. The loss decreases by two to three orders of magnitude from its initial value after 2,000 iterations, reaching convergence. Fig. 7 compares the output feature map of the first layer between the digital implementation and the optical implementation, which match well, indicating that the optical output provides a desirable input to the suffix digital layers. Note that the different output channels are tiled in one 2D plane.

Accuracy We find that our end-to-end design leads to little accuracy loss. Fig. 8a shows the network accuracies of the three training steps. As we can see, step 1 is the original network executed fully in the digital domain, and generally has the highest accuracy. Step 2 optimizes the PSF from the target convolution kernel and directly integrates the optimized PSF into the network; this leads to significant lower accuracy. However, after end-to-end fine-tuning with the suffix layers, we could largely gain the accuracy loss. Overall across the three networks, our optical-electrical DNN system leads to on average only an 1.9% (0.8% minimum) accuracy loss compared to fully-digital networks. The accuracy loss could be further improved if we could improve the PCE of the metasurface designs. Fig. 8b compares the accuracy loss on the three networks between using ideal PCEs (i.e., delta functions) and non-ideal PCEs (from actual designs, used in Fig. 8a). Using an ideal PCE leads to an average accuracy loss of only 0.9%. This suggests that our system design can readily benefit from better metasurface designs in the future.

Efficiency Our system reduces the overall computational cost in two ways. First, we completely avoid the need to process raw sensor data using the power-hungry ISP. Traditional vision systems use ISP to process raw sensor data (e.g., demosaic, tone mapping) to generate RGB images for the DNN to process, but since in our system the image sensor captures the output feature map

rather than the RGB image, the ISP processing could be avoided.

We measure the power of the Nvidia's Jetson TX2 board, which is representative of today's high-end systems for mobile vision [56]. The ISP consumes about 0.5 mJ of energy to generate one RGB frame. In contrast, the energy consumed to execute AlexNet, VGG, and ResNet50 on a Google TPU-like specialized DNN processor [9] is 0.31 mJ, 0.36 mJ, 0.05 mJ, respectively, which we estimate using a recent DNN energy modeling framework [57, 58]. Avoiding the ISP yields 2.4× to one order of magnitude energy saving.

Secondarily, by pushing the first CNN layer into the optical domain, we also reduce the total amount of computations (measured in the number of multiply-accumulate, or MAC, operations) to compute the DNN. In AlexNet, the MAC saving is 4.4%, and the MAC saving is 0.8% and 0.5% for ResNet50 and VGG, respectively.

7. Discussion and Conclusion

This paper demonstrates the first free-space optics approach that enables general convolutional layers. The significance our system is three-fold. First, it eliminates the need for the power-hungry ISP. Second, it simplifies the sensor design and enables the use of a simple monochromatic sensor rather than a RGB sensor. Finally, it reduces the computation to execute a DNN. Overall, we show an order of magnitude of energy saving with 1.9% on average (0.8% minimum) accuracy.

We are in the process of fabricating an optical prototype. While our current design pushes only the first DNN layer into optics, pushing more layers into optics is possible but not without challenges. The key challenge is that it is difficult to optically implement non-linear functions, which usually have to be implemented in the electrical domain, introducing optical-electrical interface transduction overhead. Therefore, a future direction is to comprehensively explore the hybrid optical-electrical design space to evaluate the optimal partition between optics and electronics.

8. Funding and Acknowledgments

Supplementary Materials

See Supplement 1 for supporting content.

References

1. T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association,* (2010).
2. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems,* (2013), pp. 3111–3119.
3. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems,* (2014), pp. 3104–3112.
4. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* (2016), pp. 779–788.
5. L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision,* (Springer, 2016), pp. 850–865.
6. H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," Drug discovery today **23**, 1241–1250 (2018).
7. Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems,* (1990), pp. 598–605.
8. S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149 (2015).

9. N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA),* (IEEE, 2017), pp. 1–12.

10. Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *ACM SIGARCH Computer Architecture News,* vol. 44 (IEEE Press, 2016), pp. 367–379.

11. J. W. Goodman, *Introduction to Fourier optics* (Roberts and Company Publishers, 2005).

12. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the hopfield model," Appl. optics **24**, 1469–1475 (1985).

13. D. Psaltis, D. Brady, and K. Wagner, "Adaptive optical networks using photorefractive crystals," Appl. Opt. **27**, 1752–1759 (1988).

14. T. Lu, S. Wu, X. Xu, and T. Francis, "Two-dimensional programmable optical neural network," Appl. optics **28**, 4908–4913 (1989).

15. I. Saxena and E. Fiesler, "Adaptive multilayer optical neural network with optical thresholding," Opt. Eng. **34**, 2435–2440 (1995).

16. J. Chang, V. Sitzmann, X. Dun, W. Heidrich, and G. Wetzstein, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," Sci. reports **8**, 12324 (2018).

17. 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).

18. J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, "Reinforcement learning in a large-scale photonic recurrent neural network," Optica **5**, 756–760 (2018).

19. M. Miscuglio, Z. Hu, S. Li, J. K. George, R. Capanna, H. Dalir, P. M. Bardet, P. Gupta, and V. J. Sorger, "Massively parallel amplitude-only fourier neural network," Optica **7**, 1812–1819 (2020).

20. J. B. Sampsell, "Digital micromirror device and its application to projection displays," J. Vac. Sci. & Technol. B: Microelectron. Nanometer Struct. Process. Meas. Phenom. **12**, 3242–3246 (1994).

21. D. C. O'Shea, T. J. Suleski, A. D. Kathman, and D. W. Prather, *Diffractive optics: design, fabrication, and test*, vol. 62 (Spie Press Bellingham, WA, 2004).

22. N. Yu and F. Capasso, "Flat optics with designer metasurfaces," Nat. Mater. **13**, 139–150 (2014).

23. C. L. Holloway, E. F. Kuester, J. A. Gordon, J. O'Hara, J. Booth, and D. R. Smith, "An overview of the theory and applications of metasurfaces: The two-dimensional equivalents of metamaterials," IEEE Antennas Propag. Mag. **54**, 10–35 (2012).

24. J. P. Balthasar Mueller, N. A. Rubin, R. C. Devlin, B. Groever, and F. Capasso, "Metasurface polarization optics: Independent phase control of arbitrary orthogonal states of polarization," Phys. Rev. Lett. **118**, 113901 (2017).

25. B. Wang, F. Dong, Q.-T. Li, D. Yang, C. Sun, J. Chen, Z. Song, L. Xu, W. Chu, Y.-F. Xiao *et al.*, "Visible-frequency dielectric metasurfaces for multiwavelength achromatic and highly dispersive holograms," Nano letters **16**, 5235–5240 (2016).

26. R. Pestourie, C. Pérez-Arancibia, Z. Lin, W. Shin, F. Capasso, and S. G. Johnson, "Inverse design of large-area metasurfaces," Opt. express **26**, 33732–33747 (2018).

27. S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," Nat. Photonics **12**, 659–670 (2018).

28. Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, "Deep learning with coherent nanophotonic circuits," Nat. Photonics **11**, 441 (2017).

29. F. Xia, M. Rooks, L. Sekaric, and Y. Vlasov, "Ultra-compact high order ring resonator filters using submicron silicon photonic wires for on-chip optical interconnects," Opt. express **15**, 11934–11941 (2007).

30. H. Bagherian, S. Skirlo, Y. Shen, H. Meng, V. Ceperic, and M. Soljacic, "On-chip optical convolutional neural networks," arXiv preprint arXiv:1808.03303 (2018).

31. R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, "Large-scale optical neural networks based on photoelectric multiplication," Phys. Rev. X **9**, 021032 (2019).

32. C. M. V. Burgos and N. Vamivakas, "Challenges in the path toward a scalable silicon photonics implementation of deep neural networks," IEEE J. Quantum Electron. **55**, 1–10 (2019).

33. W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, "Silicon microring resonators," Laser & Photonics Rev. **6**, 47–73 (2012).

34. 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).

35. C. Zhou and S. K. Nayar, "Computational cameras: convergence of optics and processing," IEEE Transactions on Image Process. **20**, 3322–3340 (2011).

36. J. Chang and G. Wetzstein, "Deep optics for monocular depth estimation and 3d object detection," in *Proceedings of the IEEE International Conference on Computer Vision,* (2019), pp. 10193–10202.

37. H. Haim, S. Elmalem, R. Giryes, A. M. Bronstein, and E. Marom, "Depth estimation from a single image using deep learned phase coded mask," IEEE Transactions on Comput. Imaging **4**, 298–310 (2018).

38. L. He, G. Wang, and Z. Hu, "Learning depth from single images with deep neural network embedding focal length," IEEE Transactions on Image Process. **27**, 4676–4689 (2018).

39. Y. Wu, V. Boominathan, H. Chen, A. Sankaranarayanan, and A. Veeraraghavan, "Phasecam3d—learning phase masks for passive single view depth estimation," in *2019 IEEE International Conference on Computational Photography*

(ICCP), (IEEE, 2019), pp. 1–12.

40. Q. Sun, E. Tseng, Q. Fu, W. Heidrich, and F. Heide, "Learning rank-1 diffractive optics for single-shot high dynamic range imaging," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (2020).

41. C. A. Metzler, H. Ikoma, Y. Peng, and G. Wetzstein, "Deep optics for single-shot high-dynamic-range imaging," arXiv preprint arXiv:1908.00620 (2019).

42. V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein, "End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging," ACM Transactions on Graph. (TOG) **37**, 1–13 (2018).

43. Z. Lin, C. Roques-Carmes, R. Pestourie, M. Soljačić, A. Majumdar, and S. G. Johnson, "End-to-end nanophotonic inverse design for imaging and polarimetry," Nanophotonics **1** (2020).

44. S. Colburn, Y. Chu, E. Shilzerman, and A. Majumdar, "Optical frontend for a convolutional neural network," Appl. optics **58**, 3179–3186 (2019).

45. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, P. Hanrahan *et al.*, "Light field photography with a hand-held plenoptic camera," Comput. Sci. Tech. Rep. CSTR **2**, 1–11 (2005).

46. Z. L. Deng and G. Li, "Metasurface optical holography," Mater. Today Phys. **3**, 16 – 32 (2017).

47. W. T. Chen, A. Y. Zhu, V. Sanjeev, M. Khorasaninejad, Z. Shi, E. Lee, and F. Capasso, "A broadband achromatic metalens for focusing and imaging in the visible," Nat. Nanotechnol. **13**, 220–226 (2018).

48. C. A. Palmer and E. G. Loewen, *Diffraction grating handbook*, vol. 5 (Thermo RGL New York, 2002).

49. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems,* (2012), pp. 1097–1105.

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

51. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence,* (2017).

52. A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Tech. rep., Citeseer (2009).

53. A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, "Meep: A flexible free-software package for electromagnetic simulations by the fdtd method," Comput. Phys. Commun. **181**, 687–702 (2010).

54. A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, "Meep: free finite-difference time-domain (FDTD) software for electromagnetic simulations," https://github.com/NanoComp/meep (2020).

55. Sony, "ICX274AL sensor specification," https://www.1stvision.com/cameras/sensor_specs/ICX274.pdf (2020).

56. Nvidia, "NVIDIA Jetson TX2 Delivers Twice the Intelligence to the Edge," (2018).

57. Y. Feng, P. Whatmough, and Y. Zhu, "Asv: Accelerated stereo vision system," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture,* (ACM, 2019), pp. 643–656.

58. "Systolic-array Data-flow Optimizer," https://github.com/horizon-research/systolic-array-dataflow-optimizer.