Research Article Vol. 34, No. 6 / 23 Mar 2026 / Optics Express 10173
Optics EXPRESS

Towards a non-Fourier event-based opto-electronic convolution accelerator

HANNAH KIRKLAND,[1,*] TRUNG H. LE,[1] PIPER TAYLOR,[1] MEHRAN KEIVANIMEHR,[1] ISAAC J. SLEDGE,[2] AND SANJEEV J. KOPPAL[1]

[1]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA
[2]Advanced Signal Processing and Automated Target Recognition Branch, Naval Surface Warfare Center, Panama City, FL, USA
[*]hkirkland@ufl.edu

Abstract: The current advancement of neural networks has led to a resurgence of research in optical computing. Exploiting the physical properties of light to enable optical neural networks is advantageous due to optical computing's higher parallelism, speed, and power consumption compared to conventional compute methods. However, many state-of-the-art methods rely on coherent light and conventional image sensors or photodetectors. For the first time, to the best of our knowledge, we demonstrate using an event camera for reconfigurable optical compute, implemented via a neuromorphic camera, light source, two digital micromirror devices (DMDs), and a computer. We describe the fundamental advantages and limitations of our device, then report initial results using the device for two simple machine learning tasks.

1. Introduction

The impact of deep networks continues to explode, and with it the demand for fast, energy-efficient compute. Optical computing offers desirable properties that have the potential to fulfill this demand, with its high speed, low-latency, and parallelism. Optical approaches for performing convolutions have a long history [1] and have seen a recent resurgence [2–4]. Convolutional neural networks (CNNs) have also seen their own resurgence, utilizing large kernel convolutions that rival the performance of transformer networks [5–8].

Most of the work surrounding opto-electronic neural networks (ONNs) use conventional image sensors or photodetectors to capture the device response. However, neuromorphic image sensors (i.e., event cameras) offer superior speed, dynamic range, and power efficiency compared to CMOS image sensors [9]. Additionally, while photodetectors also offer high capture speeds, their applications are limited as single-pixel detectors.

Recent research has utilized non-Fourier optical convolution methods [4,10], but the majority of research involving optical convolutions remains in the Fourier domain [11–13]. These methods require precise calibration. Incoherent convolutions, on the other hand, stay clear of the Fourier domain, simplifying the required optics and calibration.

Our proposed opto-electronic convolution accelerator generates incoherent, non-Fourier convolutions by utilizing a neuromorphic sensor, two digital micromirror devices (DMDs), a bright light source, and a computer. The first DMD modulates incoming light to form the convolution kernel, which is then diffused. The second DMD is patterned with the convolution input features and positioned in the optical path. This interaction results in a convolutional response projected onto a screen that is imaged by an event camera. In this way, the computationally burdensome convolutions are performed optically while the digitized output is processed in electronics, enabling various non-linearities and other functions such as batch normalization and pooling. Additionally, our system is reconfigurable as we use DMDs for the kernel and input features. This allows the convolutional response to be looped back into the system, enabling machine learning applications.

#584235 https://doi.org/10.1364/OE.584235
Journal © 2026 Received 12 Nov 2025; revised 12 Feb 2026; accepted 24 Feb 2026; published 13 Mar 2026

Optics EXPRESS

Utilizing an event camera in this way enables a novel method of capturing convolutional responses for optical neural networks and machine learning tasks. This method offers higher speeds and lower power consumption when compared to CMOS image sensors. As it is a non-Fourier method, the hardware required is straightforward to calibrate. **Our contributions are:**

1. ***The novel use of an event camera for optical computation,***
2. Mathematical characterization of non-Fourier, event-based convolutions, and
3. Demonstrating our device on two simple machine learning tasks.

Finally, we note that our system utilizes binary weights, taking advantage of recent work on binary neural networks to demonstrate our event-based opto-electronic accelerator.

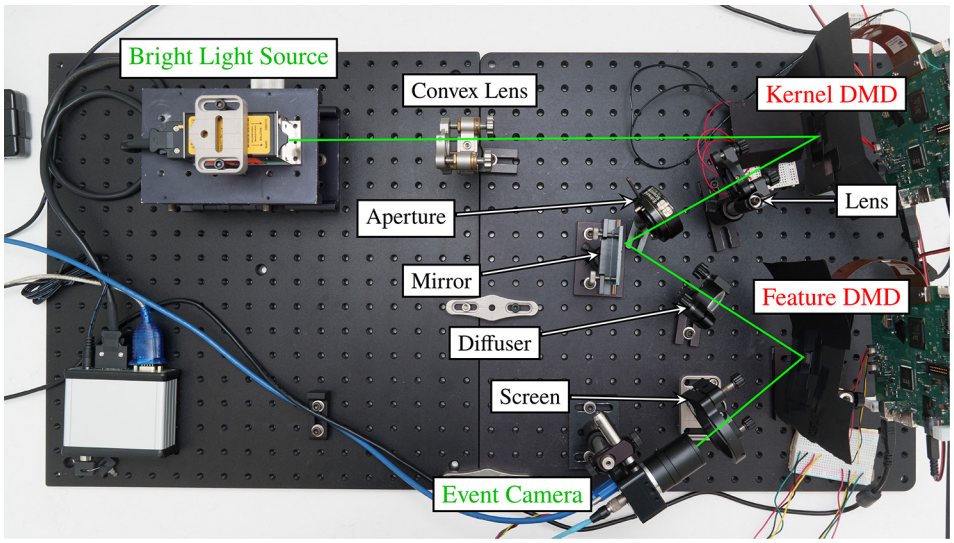


Fig. 1. *Top down view of our proposed device.* Light is patterned with the kernel DMD then passed through a lens to converge diffraction duplicates, which are blocked by the aperture. After this, the light is diffused, ensuring incoherence. The kernel is then convolved with the input features displayed on the second DMD. The result is projected on a screen, with an event camera focused on the screen. The changing response results in an event stream, which is processed to construct the total convolution response.

2. Related work

Optical computing exploits physical properties of light to perform data processing and computation. It is a mature and well studied sub-field [1] with multiple textbooks written on the subject [55,56]. In the late 1980s, neural networks were implemented in optical hardware as the result of cross-disciplinary collaboration [57–66], but enthusiasm died down after this. Recently, the resurgence of neural networks in computer vision has led to new impacts and research in the field, as outlined in Table 1.

Deep Diffractive Networks: Deep diffractive neural networks (D^2NNs) utilize diffraction [14] across a series of specially engineered surfaces to construct task-specific models [15–17]. Feed-forward networks with millions of neurons and hundreds of billions of connections across fully-connected layers have been fabricated with this method [18–21,24,25]. These approaches are fully optical, and light attenuation usually limits the number of layers. Most deep-diffraction

Optics EXPRESS

Table 1. Related work overview[a]

Approach	Property				Sensor Type
Deep Diffractive Networks [14–25]	✗	✗	✓	✓	CCD, Photodetector
Fiber-Optical Convolutions [26–40]	✗	✗	✓	✓	Photodetector
Photonic Neural Networks [41–48]	✗	✓	✓	✓	Photodetector
Computational Imaging [49–54]	✓	✗	✗	✓	CMOS
Ours	✔	✔	✔	✔	**Neuromorphic**
	Incoherent illumination	Feature pooling	Reconfigurable filters	Fully connected layers	

[a]Deep diffractive networks utilize coherent light and specialized masks to create fully connected layers. Typically, these networks do not have reconfigurable filters [14–23], but recent work has demonstrated this capacity [24,25]. D^2NNs typically use a CCD image sensor or array of photodectors to capture the output response. Fiber-optical convolutions have similar properties–i.e., reconfigurable filters and fully connected layers but no feature pooling–and use coherent light sources. Typically, these setups use a single photodetector to capture the output response. Photonic neural networks retain the advantages of the previous approaches while also supporting feature pooling. These systems also use photodetectors. Computational imaging, usually in the form of a computational camera or image sensor, uses incoherent light and can support fully connected layers. However, these systems are not reconfigurable and do not support feature pooling. Our proposed device retains the advantages of the previous work while also utilizing a unique image sensing technology: an event camera.

neural networks do not have non-linear capabilities, limiting them to only realize linear neural activation functions [17]. However, recent works have used special materials to create non-linear effects, enabling more complex models [22,23].

Fiber-Optical Convolutions: Optical processing with optical fibers attempt to mimic pathways found in the brain. Data is encoded into pulse trains that are sent down optical fibers, where Rayleigh scattering is exploited to produce concentric backscattering patterns. Multi-mode fibers [26,27] have seen extensive use for pattern recognition tasks [28–32]. They have been combined with optical reservoir computing systems [33–36] to achieve high throughput rates [37].

Photonic Neural Networks: Photonic-integrated neural chips can perform fast matrix-vector multiplications and additions [41–43]. They typically utilize either micro-loop / micro-ring modulators [44–48] or Mach-Zehnder modulators [38–40] in conjunction with some supporting hardware. These modulators manipulate the phase of the incoming coherent light, which is provided by semiconductor lasers. At the end, the signal is received by a photodetector.

Computational Imaging: Simple tasks like edge detection and depth perception have been implemented via lensless and programmable imaging. These computational cameras use a mask placed in front of a camera sensor to perform optical convolution [49–51], use specialized lenses for a specific task [52], or use custom image sensors [53,54]. In all cases, the input to the system is a real world scene, and most of these efforts focus on a single passive optical convolution step before image capture.

Optics EXPRESS

3. Characterization of non-Fourier, event-based convolutions

Convolutions are the basic building block of CNNs. W.l.o.g, here we consider 2D convolutions. Since these transformations are linear, any ND convolution can be broken up into sets of 2D convolutions that are added together.

We consider the 2D convolution of a kernel image I_K and the input features image I_F, as

$$\begin{aligned} I_{conv}(x,y) &= I_K(x,y) * I_F(x,y) \\ &= \iint_{\tau_1,\tau_2=0}^{\infty} I_K(\tau_1,\tau_2) \cdot I_F(x-\tau_1, y-\tau_2) d\tau_1 d\tau_2. \end{aligned} \tag{1}$$

3.1. Matching effective pixel sizes

Consider a convolution implemented via conventional compute methods that we wish to perform optically. In software, the kernels, input features, and resulting convolutions are discrete, with each element of their respective tensors having the same extent (i.e., one element equals one element). Our device does not inherently have this constraint, as light is continuous, and the scale of the projected images varies with the optical setup. However, the DMDs and image sensor all have a fundamental unit dimension (i.e., pixel pitch). To accurately characterize the optical convolution being performed, we must either ensure parity between the extent of each unit by manipulating the optical setup or account for any disparity in software.

We start by considering the effective spatial extent of one element of the kernel, K_{eff}, and input features, F_{eff}, in our optical setup for a given set of pixel pitches (K_{pt} and F_{pt}):

$$\begin{aligned} K_{eff} &= \varsigma K_{pt} \\ F_{eff} &= \sigma F_{pt}, \end{aligned} \tag{2}$$

where ς is the software scaling factor, representing how much the kernel is upscaled on the DMD; and σ is the optical scaling factor, dictated by the physical distances between the diffuser, the input features DMD, and the screen as discussed by [67]. The optical scaling factor σ is given by:

$$\sigma = \frac{u+z}{u}, \tag{3}$$

where the distance between the diffuser and input features DMD is represented by z, and the distance between the input features DMD and the screen is represented by u. The software scaling factor, ς can be used to ensure parity between the effective pixel sizes of the kernel and feature map:

$$\varsigma = \frac{\sigma F_{pt}}{K_{pt}}. \tag{4}$$

3.2. Effective stride

In software, stride is typically constrained to positive integers. Fractional strides instead work by dilating the input features with zeros. In optics, stride can be fractional without dilation and is determined primarily by the image sensor pitch Φ_{pt} and distances u and z. The image sensor pitch can be artificially increased by skipping pixels, decimating the result. Here the decimation factor is denoted as Δ. From similar triangles:

$$S_{mm} = \frac{z\Phi_{pt}}{u} \tag{5}$$

$$\begin{aligned} S_{eff} &= \frac{S_{mm}\Delta}{K_{eff}} \\ &= \frac{z\Phi_{pt}\Delta}{F_{pt}(u+z)} \end{aligned} \tag{6}$$

Optics EXPRESS

where S_{mm} denotes the stride in millimeters and S_{eff} denotes the effective stride in pixels.

3.3. Handling negative values

The intensity of light cannot be less than 0, so the values of kernel image I_K and input features image I_F are constrained to $[0, \infty)$. The values are further constrained by the lower noise limit of the system and the upper light intensity that the system can produce, (I_{min}, I_{max}). When we map between the optical convolutions and digital convolutions we begin by splitting the positive and negative parts of the desired total kernel K and input features F. Each convolution happens optically, and subtraction occurs in software.

3.4. Temporal capture window

The time throughput of our optical device depends on the refresh rates of the DMD modules and the temporal capture window of the neuromorphic sensor. The minimum temporal capture window that is possible with the system is determined by the slowest device,

$$h_{min} = \min(K_{hz}, F_{hz}, \frac{1}{S_l}), \tag{7}$$

where K_{hz} is the refresh rate of the kernel DMD, F_{hz} is the refresh rate of the input features DMD, and $\frac{1}{S_l}$ is the pixel latency of the event camera. The minimum time needed to capture the convolutional response is $e_{min} = \frac{1}{h_{min}}$. Capture times greater than this slows down computation and collects more events.

3.5. Determining effective bit depth

Convolutions performed on conventional architectures typically utilize 32-bit floating point numbers. This results in a high dynamic range, at the expense of computational complexity. Our device's effective bit depth is determined by a combination of the bit depth of the sensor, the temporal capture window, and the quality of our DMDs. The effective number of bits can be describe by

$$ENOB = \frac{SINAD - 1.76}{6.02},$$

where SINAD is the signal-to-noise and distortion ratio. For simplicity, we will assume distortions are negligible. We can then determine the equation for our device's SINAD:

$$\begin{aligned} SINAD &= 10 \log_{10} \frac{P_{signal} + P_{noise} + P_{distortion}}{P_{noise} + P_{distortion}} \\ &= 10 \log_{10} \frac{RMS(I_{cap})^2}{st.d.(I_{cap})^2}, \end{aligned} \tag{8}$$

where I_{cap} is the resulting image from a test scene.

3.6. Effect of shot noise

When capturing events, multiple forms of noise are present. Pattern noise is dependent on the physical image sensor. Dark shot noise is dependent on temperature and can be controlled by keeping the device adequately cooled. Poisson noise, or shot noise, always occurs when measuring light, but is dominant in low-light imaging. To reduce power requirements, it is desirable to run our system with the lowest amount of light possible. Additionally, a low temporal capture window is desirable to increase speed. These two factors reduce the number of photons converted into measurable current for a given image, and increase the amount of shot noise present. However, reducing noise in our system is also desirable, as the noise from each captured

Optics EXPRESS

convolution may compound in the post-processing needed to address negative values and multiple channels.

Shot noise follows a Poisson distribution, and the probability that k photons hit the sensor is given by:

$$P(\mathbf{X} = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{9}$$

where λ is the expected value of the variable X. Loosely speaking, this is proportional to the intensity of the light-source. We can augment the convolution equation below using the Poisson distribution as:

$$I_{conv}(x, y) = \iint_{\tau_1,\tau_2=0}^{\infty} \mathbf{C} \cdot I_F(x - \tau_1, y - \tau_2) \cdot I_K(\tau_1, \tau_2))d\tau_1 d\tau_2 \tag{10}$$

where the term $\mathbf{C}$ is the cumulative distribution of the Poisson distribution. It is given by $\Sigma_i^{N_e} P(\mathbf{X} = i * \phi)$, where $P(\mathbf{X} = i * \phi)$ is the probability of $i * \phi$ photons with an expected value proportional to the light source brightness B. The photon flux (photons per time unit) is denoted as ϕ and N_e is an integer that depends on the temporal capture window e and the time unit selected.

The cumulative term $\mathbf{C} = \Sigma_i^{N_e} P(\mathbf{X} = i * \phi)$ varies as follows: if the number of trials increases (or the brightness, i.e. power, of the light source increases) this number approaches 1 and the convolution equation resembles a conventional convolution. In a low-light scene, the cumulative term $\mathbf{C}$ can change the value of the convolution. In an extreme case, if the scene is very dark with a short capture window, the probabilities will all be low, and the output of the convolution will be near-zero.

We now characterize a worst case bound due to the effects of Poisson noise in optical convolution. Consider the lowest light intensity across all input features in all layers of a network $I_F^{low}(x - \tau_1, y - \tau_2))$, which we shorten as I^{low}. This value correlates to the lowest number of events captured at a specific pixel location across all input features in all layers of a network in software. We can place the variable cumulative term $\mathbf{C}$ with the term corresponding to the lowest value I^{low}, which induces the Poisson distribution P corresponding to the lowest expected value $\lambda_{low} = \rho_B * I_F^{low}(x - \tau_1, y - \tau_2)$. Given a temporal capture window, this induces the lowest cumulative factor $\mathbf{C}_{low}$ which is a constant, unlike the variable cumulative term $\mathbf{C}$, and can be removed from the convolution equation:

$$I_{conv}(x, y) = \mathbf{C}_{low} \cdot (I_K(x, y) * I_F(x, y)) \tag{11}$$

The impact of this factor $\mathbf{C}_{low}$ depends on the non-linear activations that are applied after each convolutional layer. The activations, such as *ReLU* and *tanh*, are usually monotonic in most regions, except when they cross the zero mark or contain a discontinuity.

We define the robustness of a neural network to be a pair of values (R, r). R is the maximum percentage of such activation "flips" that it can tolerate before classification rate falls below r on a dataset D. We then specify a bound on the noise, given the overall neural network function f_{low} and a desired classification rate r, which specifies the lowest value from the light source I_{min}^{low} such that:

$$\frac{\Sigma_i^{|D|} V(i)}{|D|} \geq r, \; s.t. \; V(i) = \begin{cases} 1, & \text{if } f_{low}(x_i, W_i) = y_i \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where the indicator function V tests the effect of the learned overall network function f, and f_{low} is the implementation of the network where every convolution is reduces by the factor $\mathbf{C}_{low}$ before the application of non-linearity.

Optics EXPRESS

4. Experimental design and results

Our device, shown in Fig. 1 and Fig. 2, consists of a 514 nm Vortran laser, two Texas Instrument 0.65-inch 1080p digital micromirror devices, and an iniVation DVXplorer Lite. Light from the laser is passed through a convex lens before being patterned via the first DMD. From here, we use another convex lens and aperture to block duplicate images caused by diffraction. The light is then passed through a diffuser, ensuring incoherence, before hitting the second DMD. The convolutional response is then projected onto a second diffuser, which serves as the image plane for the event camera. Once captured, the event stream is reconstructed in electronics. To perform multiple convolutions, the system uploads all relevant kernels and input features to their respective DMDs [68] then triggers the DMDs and the event camera with a timing sequence that captures all permutations of input features and kernel pairs.

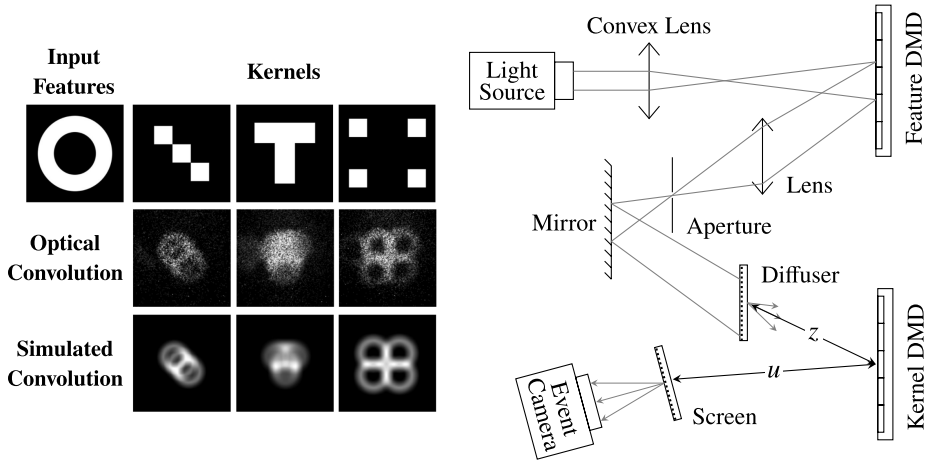


Fig. 2. *Left:* Example convolutional outputs for a given kernel and input features are shown. Optical convolutions were reconstructed from the captured event stream. Simulated convolutions were performed with the same kernel and input features on a conventional computer. *Right:* Ray diagram of our device, as pictured in Fig. 1. The distance z is a summation of the distance between the diffuser, mirror, and kernel DMD.

4.1. Benefits of binary convolutions

While our device is not *inherently* constrained to binary inputs (DMDs are available with high resolution bit depths through pulse width modulation), we opted to use binary DMD modules for speed and ease of use with event cameras. This provides both an advantage and a design constraint: our design gains superior robustness to noise but is constrained to using binarized kernels and input features. Binary weights and input features can tolerate a higher level of Poisson noise before "flipping" to the other value. They also maximize contrast, which reduces the amount of post-processing needed in software.

As we chose to constrain our system to binary kernels and input features, we opted to target binary convolutions as used in binary convolutional neural networks [69] with our device. We consider a quantization approach similar to that of Rastegari et al. [70], Li et al. [71], and others [72–76] for specifying the filters. Essentially, the binary weights are estimated by scaling the weights according to their average absolute value and using the sign of the weight magnitude. The input features are similarly binarized after applying batch normalization.

Binary CNNs have reduced representation capability compared to conventional CNNs. Furthermore, the BCNNs used are small networks, resulting in a lower baseline accuracy compared

Optics EXPRESS

to conventional CNNs. We compare our device's performance to the same model optimized on conventional computing hardware.

4.2. Optimization strategy

Currently, gradient-free methods are necessary as enabling backpropagation on device requires accurate gradients that account for the properties of the optical setup and event camera. One way of theoretically achieving this is to capture the device's light transport matrix and including it in the convolution function. Consider the discretized 2D convolution that accounts for a light transport matrix T:

$$y_{i,j} = \sum_m \sum_n K_{m,n} \cdot F_{i+m,j+n} \cdot T_{i,j,m,n,i+m,j+n},$$

which would result in gradients:

$$\frac{\partial E}{\partial K^l_{m',n'}} = \sum_i \sum_j \frac{\partial E}{\partial y^l_{i,j}} \frac{\partial y^l_{i,j}}{\partial K^l_{m',n'}}$$
$$\frac{\partial E}{\partial F^l_{i',j'}} = \sum_m \sum_n \frac{\partial E}{\partial y^l_{m,n}} \cdot K^l_{-m,-n} \cdot T_{i',j',-m,-n,i',j'}.$$

However, this 8D light transport matrix proves difficult to capture. Iterating through each pixel individually results in an extreme low-light situation, leading to poor, noisy results. Decimating the capture by iterating through patches of pixels can solve the low-light issue but results in low precision. In both cases, the slight inaccuracies incurred during capture is enough to make the gradients unusable.

Another method for enabling backpropagation with our device would be to estimate the true gradients with either a model-based or a model-free approach. Both strategies have been used with diffractive ONNs [77–80] and may transfer to our event-based approach. Further work is needed to explore these different ways of enabling backpropagation with an event-based opto-electronic convolution accelerator.

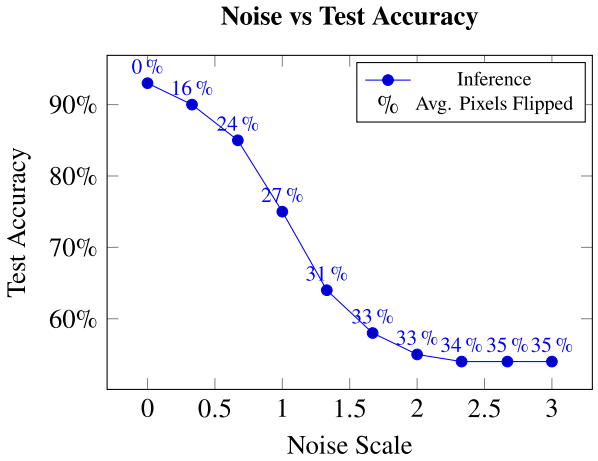
Instead, for the following experiments, we focus on using gradient-free optimization. For the kernel recovery task discussed in Sect. 4.4, we employ a naive brute force strategy, iterating through every possible kernel. This approach is suitable for recovering a 3×3 kernel with 2^9 possible combinations, but becomes exponentially unfeasible as the number of weights increases. In Sect. 4.5 we instead employ two optimization algorithms: simulated annealing and random walk. Both simulated annealing and random walk require a high number of optimization steps, but the maximum number of steps per experiment was limited by developmental drivers for the DMDs that resulted in long loading times between image batches.

4.3. Poisson noise experiments

In Fig. 3 we show the effect of simulated Poisson noise on classification accuracy for a two layer BCNN trained as a binary classifier on a balanced combination of Labeled Faces in the Wild [81] ("Face") and CIFAR-10 [82] ("Not Face") data. We refer to this as the Face-Not-Face (FNF) dataset. In these experiments, Poisson noise was steadily increased for every convolution in the network, and the number of pixels "flipped" was recorded after each subsequent binarization.

Next we explored the impact that light source brightness has on the stability of the captured response. Using the same image sizes as the previous simulation, a chosen kernel and input features was captured ten times with our device. The event stream was reconstructed into an image, down-sampled, and binarized, simulating the impact Poisson noise would make in a future BCNN layer. These processed responses were compared to each other, resulting a percentage of pixels "flipped" between the responses.

Optics EXPRESS



Laser Power (mW)	Avg. Pixels Flipped
60	11.74%
50	12.48%
40	13.14%
30	15.09%
20	17.81%

Fig. 3. *Left:* The impact of Poisson noise on test accuracy in simulation. A two layer BCNN model was trained as a binary classifier with an original test accuracy of 92.88% on the FNF dataset. During inference, scaled Poisson noise was added to the output of each BCNN layer, and the difference in pixel values was recorded after binarization. *Right:* The average percentage of pixels "flipped" at a given laser power level. A series of eleven different kernels were displayed for a given input features and captured ten times. The captured response was then post-processed and binarized as would be typical in our BCNN model. The resulting 16×16 binary images were compared and the difference in pixel values recorded.

Having an average percentage of pixels "flipped" in simulation on a conventional computer and from our device gives us the ability to compare noise levels. With this, we can gauge whether the Poisson noise present in our system is within an acceptable limit. From these noise experiments, we would expect a <5% test accuracy decrease solely from Poisson noise.

4.4. Kernel recovery

Next we explore the system's functional ability. For our opto-electronic convolution accelerator to be useful, the captured response needs to be distinct for varying inputs. This is impacted by noise (as discussed in Sec. 4.3), capture time (Sec. 3.4), and alignment. To determine the functional ability of our system, we implemented a kernel recovery task. The goal of this task was to recover the kernel used to generate a chosen convolutional response. First, we chose a 3x3 binary pattern for the kernel and a 16x16 binary pattern for the input features. The convolutional response was captured and served as the ground truth data during the recovery step.

During the recovery step, we convolved every 3x3 binary kernel pattern with the input features, resulting in 2^9 iterations. The response was compared with the target using MSE loss, and the iteration with the lowest loss was chosen as the output kernel. The results of three experiments are shown in Fig. 4, where the kernel was successfully recovered in two out of the three trials. In the case where the kernel was not successfully recovered, the output kernel is a shifted version of the target and has a similar MSE loss (≤ 0.1).

Optics EXPRESS

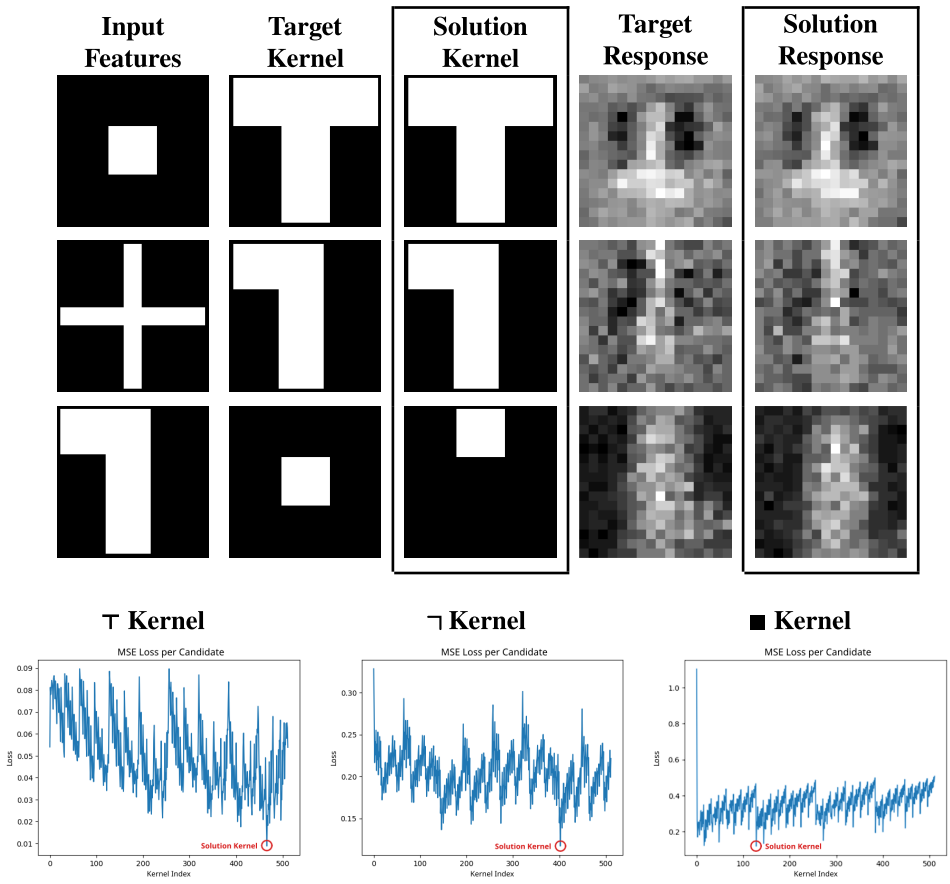


Fig. 4. Results for the kernel recovery task. The "Input Features" column shows the input image for each experiment. The "Target Kernel" column shows the weights used to generate the ground truth convolutional response, shown in "Target Response." Each possible kernel was displayed, captured, and compared to the target response. The kernel with the lowest MSE loss is displayed under "Solution Kernel" and the captured convolutional response under "Solution Response." Below, each graph shows the MSE loss for every candidate kernel for the task.

4.5. *Binary classification*

We constructed a binary classification model targeting two tasks. The first was differentiating between "Face" and "Not Face" using the FNF dataset, and the second was differentiating between letters and digits using a balanced subset of EMNIST [83] data. This model consisted of three BCNN layers followed by a pooling layer. Optimization occurred in an iterative, layerwise approach. During the optimization of the first and hidden layers, the dot product was calculated between the current layer and two complementary binary masks, as shown in Fig. 5. These two values were compared to generate an output classification, which was then used in binary cross entropy with the data label. This method separates the two classes consistently during layerwise optimization. Binary cross entropy is directly used with the model output and the data labels for the final layer.

Optics EXPRESS

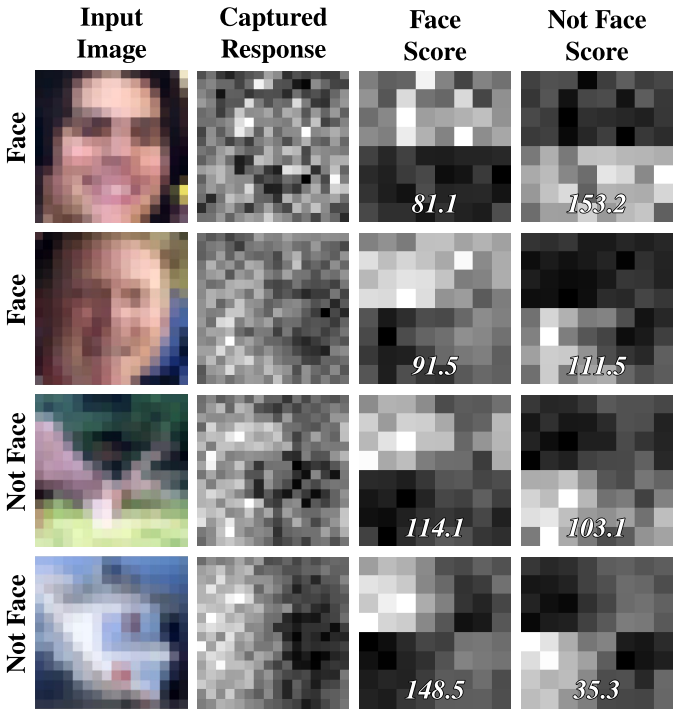


Fig. 5. Example images from the FNF dataset and corresponding device outputs. The far left column denotes the label for each row ("Face" or "Not Face"), and the "Image" column is the dataset image data after downsampling to 16×16. The "Captured Response" column is the captured response after reconstructing the event stream. The images in the "Face Score" and "Not Face Score" columns represent the output after element-wise multiplication but before summation. The values in these columns are the results after summation, i.e., the dot product of the output with each binary classification mask. These values are subsequently passed to the optimization criterion, which in this case is binary cross entropy. The lower of the two values determines the predicted label. Each of these examples correctly predicted the target label. For a full summary of test results, see Table 2.

4.5.1. Face-not-face

Simulated annealing was used to optimize the model for the FNF dataset with a temperature of 10. The model was optimized seven times in this way and the test accuracies recorded. The same optimization was performed fifty times using a conventional computer, with both results reported in Table 2.

Due to the low number of steps used for optimization, the same model was optimized using simulated annealing and random walk with a temperature of 5. This was repeated five times for each optimization strategy and reported in Table 2. Simulated annealing achieved slightly higher results (54.6% vs 51.5%), but also had a slightly higher standard deviation.

Optics EXPRESS

Table 2. Test accuracy results of the three layer BCNN model optimized on the FNF dataset[a]

	Simulated Annealing			*Random Walk*
Statistic	Conventional Computer	Opto-Electronic Device		
Max	72.8%	66.7%	63.0%	56.4%
Min	30.3%	52.4%	46.7%	48.9%
Mean	49.9%	55.1%	54.6%	51.5%
St.D.	9.6%	12.1%	5.9%	3.2%
	Temperature = 10		*Temperature = 5*	

[a]The model was optimized in an iterative, layerwise fashion. The "Opto-Electronic Device" column was optimized on real hardware, while the "Conventional Computer" column reports the test accuracies from optimizing the same model on a conventional computer. The "Simulated Annealing" column was optimized using simulated annealing while the "Random Walk" column was optimized with random walk.

4.5.2. Letters vs. digits

Like the previous section, the model was optimized with simulated annealing on balanced subset of EMNIST data. The temperature was 10, the model was optimized three times, and the test accuracies recorded. The same optimization was performed five times using a conventional computer, with both results reported in Table 3. Our device achieved a similar mean test accuracy to the conventional computer (2.7% difference) but, similar to the previous experiment, had a higher standard deviation.

Table 3. Test accuracy results of the three layer BCNN model optimized on the EMNIST dataset[a]

Statistic	Conventional Computer	Opto-Electronic Device
Mean	58.5±0.83%	55.8±2.24%
Max	58.9%	58.6%
	Temperature = 10	

[a]The model was optimized in an iterative, layerwise fashion using simulated annealing. The "Opto-Electronic Device" column was optimized on real hardware, while the "Conventional Computer" column reports the test accuracies from optimizing the same model on a conventional computer.

5. Conclusion

Here we have proposed an opto-electronic non-Fourier convolution accelerator utilizing a neuromorphic image sensor and digital micromirror devices. By using an event camera, we enable higher speeds, dynamic range, and power efficiency offered by neuromorphic image sensors over conventional CMOS image sensors. By targeting non-Fourier convolutions, we avoid the complexity typical of many other optical convolution methods.

In characterizing this device, we described the way these convolutions differ from those performed on conventional hardware (i.e., GPUs). However, due to these differences, further study is needed to enable backpropagation with an unconstrained convolutional response, rather than attempting parity with those obtained via conventional means. Despite this, our initial noise experiments and kernel recovery task demonstrate the level of information present in the captured response of our system. Additionally, our device performs similarly to simulation for the binary classification task, indicating that the sub-optimal performance is due to limitations in training

Optics EXPRESS

methods. Overall, using event cameras to capture non-Fourier optical convolutions unlocks new ways to implement opto-electronic convolution accelerators.

Funding. Office of Naval Research (N000142312363, N000142312429); U.S. National Science Foundation (1942444, 2330416).

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in CIFAR-10, Ref. [82]; Labeled Faces in the Wild, Ref. [81]; and EMNIST, Ref. [83].

References

1. J. W. Goodman, *Introduction to Fourier Optics* (Roberts and Company Publishers, 2005).
2. G. Wetzstein, A. Ozcan, S. Gigan, *et al.*, "Inference in artificial intelligence with deep optics and photonics," Nature **588**(7836), 39–47 (2020).
3. X. Meng, N. Shi, G. Li, *et al.*, "Optical convolutional neural networks: Methodology and advances," Appl. Sci. **13**(13), 7523 (2023).
4. W. Shi, X. Jiang, Z. Huang, *et al.*, "Lensless opto-electronic neural network with quantum dot nonlinear activation," Photonics Res. **12**(4), 682–690 (2024).
5. X. Ding, X. Zhang, J. Han, *et al.*, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11953–11965.
6. W. Wang, S. Li, J. Shao, *et al.*, "Lkc-net: large kernel convolution object detection network," Sci. Rep. **13**(1), 9535 (2023).
7. C. Li, P. Shi, Q. Chen, *et al.*, "LKCA: large kernel convolutional attention," in *International Conference on Optics and Machine Vision* J. Liu and K. Subramaniyam, eds. (SPIE, 2024).
8. X. Ding, Y. Zhang, Y. Ge, *et al.*, "Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 5513–5524.
9. G. Gallego, T. Delbrück, G. Orchard, *et al.*, "Event-based vision: A survey," IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 154–180 (2022).
10. Z. Huang, W. Shi, S. Wu, *et al.*, "Pre-sensor computing with compact multilayer optical neural network," Sci. Adv. **10**(30), eado8516 (2024).
11. L. Fan, X. Long, J. Dai, *et al.*, "Optical–electronic hybrid fourier convolutional neural network based on super-pixel complex-valued modulation," Appl. Opt. **62**(5), 1337–1344 (2023).
12. Q. Wu, X. Sui, Y. Fei, *et al.*, "Multi-layer optical Fourier neural network based on the convolution theorem," AIP Adv. **11**(5), 055012 (2021).
13. M. Miscuglio, Z. Hu, S. Li, *et al.*, "Massively parallel amplitude-only fourier neural network," Optica **7**(12), 1812–1819 (2020).
14. O. Kulce, D. Mengu, Y. Rivenson, *et al.*, "All-optical information-processing capacity of diffractive surfaces," Light:Sci. Appl. **10**(1), 25 (2021).
15. X. Lin, Y. Rivenson, N. T. Yardimci, *et al.*, "All-optical machine learning using diffractive deep neural networks," Science **361**(6406), 1004–1008 (2018).
16. T. Fu, Y. Zhang, H. Huang, *et al.*, "On-chip photonic diffractive optical neural network based on a spatial domain electromagnetic propagation model," Opt. Express **29**(20), 31924–31940 (2021).
17. D. Mengu, Y. Luo, Y. Rivenson, *et al.*, "Analysis of diffractive optical neural networks and their integration with electronic neural networks," IEEE J. Sel. Top. Quantum Electron. **26**(1), 1–14 (2020).
18. J. Chang, V. Sitzmann, X. Dun, *et al.*, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," Sci. Rep. **8**(1), 12324 (2018).
19. J. Li, D. Mengu, Y. Luo, *et al.*, "Class-specific differential detection in diffractive optical neural networks improves inference accuracy," Adv. Photonics **1**(04), 1 (2019).
20. T. Yan, J. Wu, T. Zhou, *et al.*, "Fourier-space diffractive deep neural network," Phys. Rev. Lett. **123**(2), 023901 (2019).
21. M. S. S. Rahman, J. Li, D. Mengu, *et al.*, "Ensemble learning of diffractive optical networks," Light:Sci. Appl. **10**(1), 14 (2021).
22. Y. Li, J. Li, and A. Ozcan, "Nonlinear encoding in diffractive information processing using linear optical materials," Light:Sci. Appl. **13**(1), 173 (2024).
23. Y. Sun, M. Dong, M. Yu, *et al.*, "Nonlinear All-Optical Diffractive Deep Neural Network with 10.6 *M*m Wavelength for Image Classification," Int. J. Opt. **2021**, 1–16 (2021).
24. Z. Fu, T. Fu, H. Wu, *et al.*, "Reconfigurable binary diffractive optical neural network based on chalcogenide phase change material ge2sb2se4te1," Opt. Express **32**(23), 41433–41444 (2024).
25. P. Feng, F. Liu, Y. Liu, *et al.*, "Diffractive magic cube network with super-high capacity enabled by mechanical reconfiguration," Nat. Commun. **17**(1), 1605 (2026).
26. B. Fischer and S. Sternklar, "Image transmission and interferometry with multimode fibers using self?pumped phase conjugation," Appl. Phys. Lett. **46**(2), 113–114 (1985).

Optics EXPRESS

27. S. G. Leon-Saval, T. A. Birks, J. Bland-Hawthorn, *et al.*, "Multimode fiber devices with single-mode performance," Opt. Lett. **30**(19), 2545–2547 (2005).
28. S. Aisawa, K. Noguchi, and T. Matsumoto, "Remote image classification through multimode optical fiber using a neural network," Opt. Lett. **16**(9), 645–647 (1991).
29. R. Takagi, R. Horisaki, and J. Tanida, "Object recognition through a multi-mode fiber," Opt. Rev. **24**(2), 117–120 (2017).
30. N. Borhani, E. Kakkava, C. Moser, *et al.*, "Learning to see through multimode fibers," Optica **5**(8), 960–966 (2018).
31. U. Teğin, M. Yildirim, I. Oğuz, *et al.*, "Scalable optical learning operator," Nat. Comput. Sci. **1**(8), 542–549 (2021).
32. Z. Liu, L. Wang, Y. Meng, *et al.*, "All-fiber high-speed image detection enabled by deep learning," Nat. Commun. **13**(1), 1433 (2022).
33. K. Vandoorne, W. Dierckx, B. Schrauwen, *et al.*, "Toward optical signal processing using photonic reservoir computing," Opt. Express **16**(15), 11182–11192 (2008).
34. Q. Vinckier, F. Duport, A. Smerieri, *et al.*, "High-performance photonic reservoir computer based on a coherently driven passive cavity," Optica **2**(5), 438–446 (2015).
35. C. Mesaritakis and D. Syvridis, "Reservoir computing based on transverse modes in a single optical waveguide," Opt. Lett. **44**(5), 1218–1221 (2019).
36. F. Duport, B. Schneider, A. Smerieri, *et al.*, "All-optical reservoir computing," Opt. Express **20**(20), 22783–22795 (2012).
37. S. Sunada, K. Kanno, and A. Uchida, "Using multidimensional speckle dynamics for high-speed, large-scale, parallel photonic computing," Opt. Express **28**(21), 30349–30361 (2020).
38. R. Hamerly, L. Bernstein, A. Sludds, *et al.*, "Large-scale optical neural networks based on photoelectric multiplication," Phys. Rev. X **9**(2), 021032 (2019).
39. S. Pai, B. Bartlett, O. Solgaard, *et al.*, "Matrix optimization on universal unitary photonic devices," Phys. Rev. Appl. **11**(6), 064044 (2019).
40. A. N. Tait, T. Ferreira de Lima, M. A. Nahmias, *et al.*, "Silicon photonic modulator neuron," Phys. Rev. Appl. **11**(6), 064043 (2019).
41. Y. Shen, N. C. Harris, S. Skirlo, *et al.*, "Deep learning with coherent nanophotonic circuits," Nat. Photonics **11**(7), 441–446 (2017).
42. M. Y. S. Fang, S. Manipatruni, C. Wierzynski, *et al.*, "Design of optical neural networks with component imprecisions," Opt. Express **27**(10), 14009–14029 (2019).
43. H. Zhang, M. Gu, X. D. Jiang, *et al.*, "An optical neural chip for implementing complex-valued neural network," Nat. Commun. **12**(1), 457 (2021).
44. C. Mesaritakis, V. Papataxiarhis, and D. Syvridis, "Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system," J. Opt. Soc. Am. **30**(11), 3048–3055 (2013).
45. A. N. Tait, T. Ferreira de Lima, E. Zhou, *et al.*, "Neuromorphic photonic networks using silicon photonic weight banks," Sci. Rep. **7**(1), 7430 (2017).
46. A. N. Tait, A. X. Wu, T. Ferreira de Lima, *et al.*, "Two-pole microring weight banks," Opt. Lett. **43**(10), 2276–2279 (2018).
47. C. Huang, S. Bilodeau, T. Ferreira de Lima, *et al.*, "Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits," APL Photonics **5**(4), 040803 (2020).
48. S. Ohno, R. Tang, K. Toprasertpong, *et al.*, "Si microring resonator crossbar array for on-chip inference and training of the optical neural network," ACS Photonics **9**(8), 2614–2622 (2022).
49. S. J. Koppal, I. Gkioulekas, T. Young, *et al.*, "Toward Wide-Angle Microvision Sensors," IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2982–2996 (2013).
50. A. Zomet and S. K. Nayar, "Lensless imaging with a controllable aperture," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2006).
51. S. K. Nayar, V. Branzoi, and T. E. Boult, "Programmable imaging using a digital micromirror array," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2004), pp. 1–8.
52. Q. Guo, Z. Shi, Y.-W. Huang, *et al.*, "Compact single-shot metalens depth sensors inspired by eyes of jumping spiders," Proc. Natl. Acad. Sci. **116**(46), 22959–22965 (2019).
53. H. G. Chen, S. Jayasuriya, J. Yang, *et al.*, "ASP vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 903–912.
54. Y. Fu, Y. Yonggan, Y. Zhang, *et al.*, "SACoD: Sensor algorithm co-design towards efficient CNN-powered intelligent PhlatCam," in *Proceedings of the IEEE International Conference on Computer Vision* (2021), pp. 5168–5177.
55. J. Horner, *Optical Signal Processing* (Elsevier, 2012).
56. C. Denz, *Optical Neural Networks* (Springer Science Business Media, 2013).
57. K. Wagner and D. Psaltis, "Multilayer optical learning networks," Appl. Opt. **26**(23), 5061–5076 (1987).
58. D. Psaltis, D. Brady, and K. Wagner, "Adaptive optical networks using photorefractive crystals," Appl. Opt. **27**(9), 1752–1759 (1988).
59. P. E. Keller and A. F. Gmitro, "Design and analysis of fixed planar holographic interconnects for optical neural networks," Appl. Opt. **31**(26), 5517–5526 (1992).

Optics EXPRESS

60. D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," Opt. Lett. **10**(2), 98–100 (1985).
61. E. G. Paek and D. Psaltis, "Optical associative memory using Fourier transform holograms," Opt. Eng. **26**(5), 265428 (1987).
62. M. Ishikawa, N. Mukohzaka, H. Toyoda, *et al.*, "Optical associatron: A simple model for optical associative memory," Appl. Opt. **28**(2), 291–301 (1989).
63. L.-S. Lee, H. M. Stoll, and M. C. Tackitt, "Continuous-time optical neural network associative memory," Opt. Lett. **14**(3), 162–164 (1989).
64. N. H. Farhat, D. Psaltis, A. Prata, *et al.*, "Optical implementation of the Hopfield model," Appl. Opt. **24**(10), 1469–1475 (1985).
65. J.-S. Jang, S.-W. Jung, S.-Y. Lee, *et al.*, "Optical implementation of the Hopfield model for two-dimensional associative memory," Opt. Lett. **13**(3), 248–250 (1988).
66. H. Shouval, I. Shariv, T. Grossman, *et al.*, "An all-optical Hopfield netowrk: Theory and experiment," Int. J. Neur. Syst. **01**(04), 355–360 (1991).
67. A. Zomet and S. K. Nayar, "Lensless imaging with a controllable aperture," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1 (IEEE, 2006), pp. 339–346.
68. P. Pozzi, D. Wilding, O. Soloviev, *et al.*, "High speed wavefront sensorless aberration correction in digital micromirror based confocal microscopy," Opt. Express **25**(2), 949–959 (2017).
69. I. Hubara, M. Courbariaux, D. Soudry, *et al.*, "Binarized neural networks," in *Advances in Neural Information Processing Systems*, vol. 29 D. Lee, M. Sugiyama, and U. Luxburg, eds. (Curran Associates, Inc., 2016).
70. M. Restegari, V. Ordonez, J. Redmon, *et al.*, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proceedings of the European Conference on Computer Vision* (2016).
71. Z. Li, B. Ni, W. Zhang, *et al.*, "Performance guaranteed network acceleration via high-order residual quantization," in *Proceedings of the IEEE International Conference on Computer Vision* (2017).
72. J. Faraone, N. Fraser, M. Blott, *et al.*, "SYQ: Learning symmetric quantization for efficient deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2018).
73. P. Wang, Q. Hu, Y. Zhang, *et al.*, "Two-step quantization for low-bit neural networks," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (2018).
74. D. Zhang, J. Yang, D. Ye, *et al.*, "LQ-Nets: Learned quantization for highly accurate and compact deep neural networks," in *Proceedings of the European Conference on Computer Vision* (2018).
75. A. Mishra, E. Nurvitadhi, J. J. Cook, *et al.*, "WRPN: Wide reduced-precision networks," in *Proceedings of the International Conference on Learning Representations* (2018).
76. B. Martinez, J. Yang, A. Bulat, *et al.*, "Training binary neural networks with real-to-binary convolutions," in *Proceedings of the International Conference on Learning Representations* (2020).
77. G. Zhao, X. Shu, and R. Zhou, "High-performance real-world optical computing trained by in situ gradient-based model-free optimization," IEEE Trans. Pattern Anal. Mach. Intell. **47**(9), 7194–7205 (2025).
78. T. Zhou, X. Lin, J. Wu, *et al.*, "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," Nat. Photonics **15**(5), 367–373 (2021).
79. T. Zhou, L. Fang, T. Yan, *et al.*, "In situ optical backpropagation training of diffractive optical neural networks," Photonics Res. **8**(6), 940–953 (2020).
80. J. Spall, X. Guo, and A. I. Lvovsky, "Hybrid training of optical neural networks," Optica **9**(7), 803–811 (2022).
81. G. B. Huang, V. Jain, and E. Learned-Miller, "Unsupervised joint alignment of complex images," in *Proceedings of the IEEE Conference on Computer Vision* (2007), pp. 1–8.
82. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep. 0, University of Toronto, Toronto, Ontario (2009).
83. G. Cohen, S. Afshar, J. Tapson, *et al.*, "EMNIST: an extension of MNIST to handwritten letters," in *CoRR* (2017).