ClassificationCam: How can optics be used in convolutional neural networks? (not final)

Julie Chang[1,*], Author Author[2], Author Author[1,2,+], and Author Author[2,+]

[1]Stanford University, Bioengineering Department, Stanford, 94305, USA
[2]Stanford University, Electrical Engineering Department, Stanford, 94305, USA
[*]corresponding.author@email.example

ABSTRACT

Example Abstract. Abstract must be under 200 words and not include subheadings or citations. Example Abstract. Abstract must be under 200 words and not include subheadings or citations. Example Abstract. Abstract must be under 200 words and not include subheadings or citations. Example Abstract. Abstract must be under 200 words and not include subheadings or citations. Example Abstract. Abstract must be under 200 words and not include subheadings or citations. Example Abstract. Abstract must be under 200 words and not include subheadings or citations. Example Abstract. Abstract must be under 200 words and not include subheadings or citations.

Introduction

Deep neural networks have found success in a wide variety of applications, ranging from computer vision to natural language processing to game playing[1]. Convolutional neural networks (CNNs), capitalizing on the spatial invariance of certain properties of images, have been especially popular in computer vision problems such as image classification, image segmentation, and even image generation[2–4]. As performance on a breadth of tasks has improved to a remarkable level, the number of parameters and connections in these networks has grown dramatically, and the power and memory requirements to train and use these networks have increased correspondingly.

While the training phase of learning parameter weights is often considered the slow stage, large models also demand significant energy during inference due to millions of repeated memory references and matrix multiplications. For example, the final version of Google DeepMind's AlphaGo in[5] used 40 search threads, 48 CPUs, and 8 GPU to play a game of Go. Live imaging and sensing applications face the additional challenge of power-hungry sensors and high bandwidth transfer of data to feed into the downstream computer vision algorithms[6]. For these reasons, it remains difficult for embedded systems such as mobile vision, autonomous vehicles and robots, and wireless smart sensors to deploy CNNs due to stringent constraints on power and bandwidth.

Optical computing has been tantalizing for its high bandwidth and inherently parallel processing, potentially at the speed of light. Furthermore, certain linear transformations can be performed in free-space or on a photonic chip with minimal to no power consumption, e.g. a lens can take a Fourier transform "for free"[7,8]. Nonlinear operations could also be addressed optically, drawing on passive nonlinear materials or devices whose refractive indices or transmission states are dependent on optical input[9,10]. An optimizable and scalable set of optical configurations that preserves these advantages and serves as a framework for building optical CNNs would be of interest to computer vision, robotics, machine learning, and optics communities. Optical implementation could also have the potential to expand beyond traditional operations of CNNs, potentially by harnessing wave optics and quantum optics in new ways.

We take initial steps toward this broader goal from a computational imaging approach, integrating image acquisition with computation via co-design of optics and algorithms. By pushing one or more layers of a CNN into the optics, we can reduce the workload of the electronic processor when performing inference with a CNN. Imaging systems are often characterized by their point spread function (PSF), which describes how a single point source of light propagates through the system. Hence, for a simple linear and space-invariant system, the image recorded at the output is the convolution of the original object with the system PSF[8]. This built-in convolution motivated us to explore how we could use optics to replace one or more of the layers in a CNN.

In this paper, we propose a toolbox of optical building blocks that could be used to implement common neural network layers. To evaluate these components, we build a simulation framework for testing a few variations of optical CNNs with the relevant physical constraints, including learned optical correlators, hybrid optoelectronic CNNs, and fully optical CNNs. We

train these networks to perform image classification on a few different datasets (MNIST, GoogleQuickdraw, or CIFAR-10), and we compare the simulated ONN accuracy against the unconstrained computer implementation of the same network structure. To demonstrate the validity of our simulations, we build a hybrid optoelectronic two-layer network with an optical convolutional layer and electronic fully connected layer for CIFAR-10 classification. We compare performance with the same inference performed on the computer, with and without the simulated physical constraints of an optical setup.

Overview of limitations. While the proposed ONN architectures offer lower power inference on classification tasks, the physical image formation imposes several constraints on the CNN architecture, including nonnegative signal and weights when using incoherent light, no bias, limited set of nonlinearities, etc. We will discuss in more detail in the paper how much each of these constraints limit the performance of our system. Here we demonstrate proof-of-concept with bulk optics and free-space propagation, which is not necessarily practical or scalable to commercial applications. However, photonic integrated circuits could significantly help in both these regards[11–13]. Combination of these next-generation large-scale photonic circuits with compressed deep learning models could provide a potential route for high performance ONNs.

Related work

Efficient convolutional neural networks. Since our work is motivated by the potential of optics to increase the effiiciency of CNN applications, we first review algorithms and electronic hardware also designed to address this challenge. Pruning, trained quantization, huffman encoding, and altered architectural design have been successfully used to compress CNN models, preserving AlexNet-level accuracy on ImageNet even with 510× less memory usage and 50× fewer parameters[14,15]. On the hardware front, there are now specialized processing units for deep learning, such as TrueNorth, Movidius's USB-based neural compute stick (NCS), and Google's tensor processing unit (TPU). All of these are complementary to our approach, which still requires offline training to optimize the optical components. Other inference-focused efforts aimed at embedded vision applications have tried to incorporate a portion of the image processing on the sensor chip, eliminating or reducing the need to shuttle full image data to a processor. Analog circuitry has been used to detect edges and orientations, to perform wavelet or discrete cosine transforms, and even to execute layers of a CNN[16,17]. These approaches still rely on electronic computation on the image sensor chip, whereas our goal is to push more of the computation into optical hardware that requires no power input.

Computational cameras and optical correlators. Optical computation is attractive because it offers inherent parallelism and high interconnectivity, both of which are encountered when passing signals through neural networks. In the computational imaging community, many system designs already exploit the physical propagation of light through custom optics to encode information about a scene that would be lost in a standard 2D image capture. Computational cameras have been created to record depth, light fields, light transport, and more with a toolbox including coded apertures, lenses and lenslets, active illumination, and wavefront shapers[18–22]. In this work, we propose a computational imaging system modeled after a CNN that assists in performing classification of input images. We begin by learning an optical correlator consisting of a single convolutional layer that essentially performs template matching on images, as has been explored for optical target detection and tracking[23–25]. Since the capabilities of a single layer linear classifier are limited, we then expand beyond a single matched filter in hybrid optoelectronic and fully optical designs.

Optical neural networks. The concept of an optical neural network (ONN) captured the attention of many in the late 1980s to mid-1990s, primarily due to the capability of optics to perform the expensive matrix multiply of a fully connected layer. In 1985, an optoelectronic implementation of the Hopfield model, a basic model of a recurrent neural network, was created with 1D LED array input signals and a binary transmission mask[26]. This model divided the weight matrix into two parts, positive and negative, and required electronics for subtraction of the two parts and signal thresholding. Psaltis et al. further explored the potential of dynamic photorefractive crystals to store neural network weights, which could allow for optical backpropagation-based learning in ONNs[27]. Meanwhile, the optoelectronic network of a Hopfield model was extended to 2D signals by partitioning the pixels of a liquid crystal television to store an array of smaller 2D patterns[28]. Furthermore, an optical thresholding perceptron was implemented with liquid crystal light valves (LCLV), which disposed of the need to convert between optical and electronic signals between layers[29]. We draw on some of these insights for the design of our optical CNN. A more extensive overview of the varied implementations of ONNs can be found in[30].

Despite the progress in this area, as neural networks fell out of the spotlight, the demand for ONNs also waned. However, with the resurgence of CNNs that are far more powerful and computationally expensive than before, there is renewed interest in optical computing [1]. Recent works that connect efforts of the last century to modern hardware include a two-layer fully connected neural network based on programmable photonic circuits[13] and a recurrent neural network with DMD-based weights[31]. However, none of the ONNs mentioned previously involve convolutional layers, which have become essential in computer vision applications. The ASP Vision system approaches the task of designing a hybrid optoelectronic CNN, using

[1] Fathom Computing (fathomcomputing.com), Lightelligence (lightelligence.ai), Optalysis (optalysys.com)

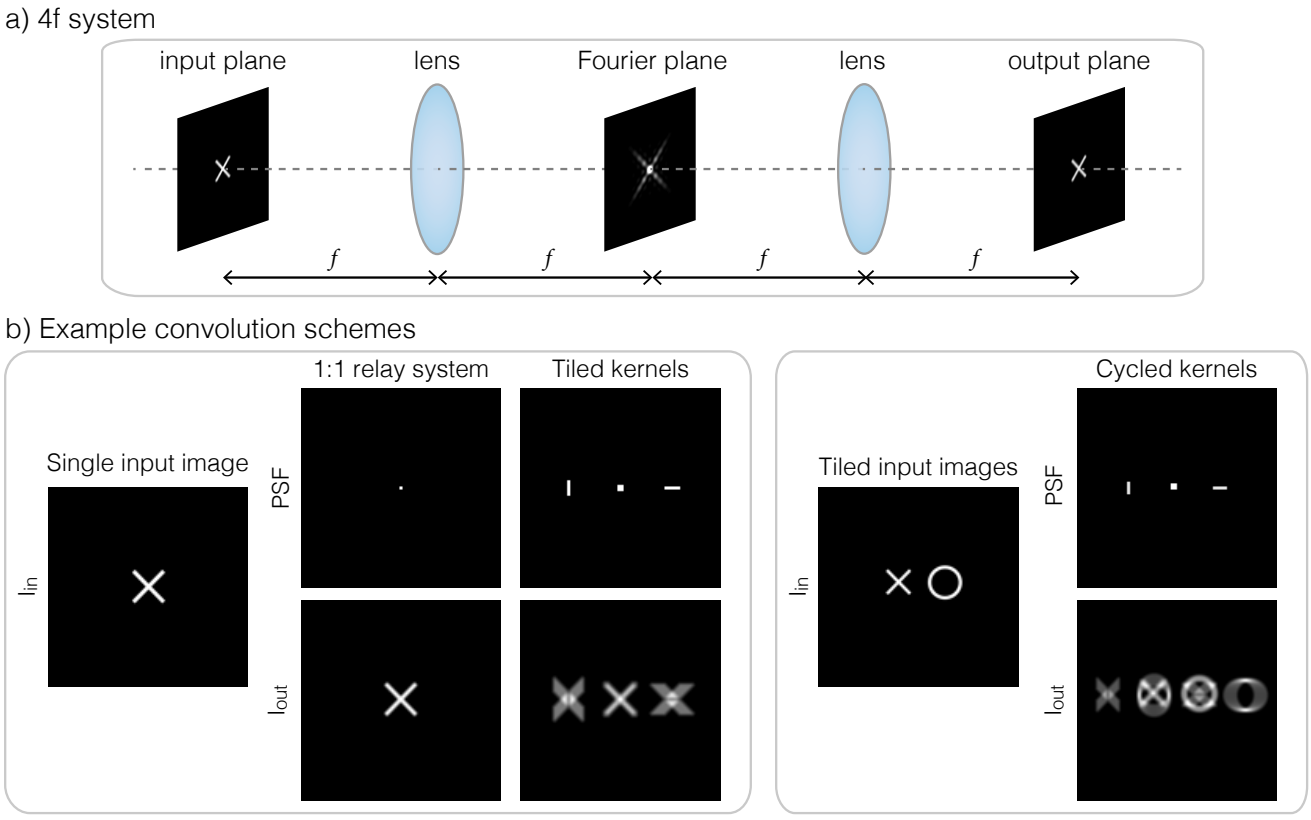

Figure 1. Optical convolutional layer

angle sensitive pixels to approximate the first convolutional layer of a typical CNN, but it is limited to a fixed set of convolution kernels[32]. Our goal is to design a system with optimizable optical elements to demonstrate low-power inference by a custom optical or optoelectronic CNN.

Optical CNN Toolbox

In this section we describe proposed optical building blocks corresponding to common layers in a CNN. We only consider standard feed-forward CNNs, where information is passed in a single direction through a sequence of layers. Cycles, loops, interacting networks, and other more complicated architectures could be interesting to explore in the future. For now, we will focus on the most essential components that define a CNN in the context of an image classification task. Later, these building blocks will be used to simulate classification models consisting of a single optical convolutional layer (i.e. an optical correlator), one optical convolutional layer fed into one digital fully connected layer, and a fully optical convolutional neural network. Note that we assume spatially incoherent light as the input to the proposed systems as this is most relevant to an imaging scenario.

Convolutional layer

A CNN typically begins with a convolutional layer, which essentially performs pattern matching with a set of learnable visual filters. A standard convolutional layer takes an input volume of depth C_{in}, performs a series of correlations with a set of C_{out} kernels each with depth C_{in}, and outputs a new volume of depth C_{out}. The correlation of the kernel across the width and height of the input volume produces a 2D "activation map", and stacking the C_{out} activation maps for all kernels forms the output volume of depth C_{out}. Hyperparameters include the spatial extent of the kernel F, the stride with which the kernel is applied, and the padding of the input volume. Here we assume a stride of 1, meaning the kernel is shifted by one pixel at a time, and zero-padding such that the output volume has the same height and width as the input. Each channel of the output image from a single non-strided, zero-padded convolutional layer can be described as:

$$I_{\text{out, j}} = \sum_{i=1}^{C_{\text{in}}} I_{\text{in, i}} \star \text{W}_{i,j} \text{ , for } j \in 1, 2, \ldots C_{\text{out}} \tag{1}$$

In linear optical systems, image formation is often modeled as a spatially invariant convolution of the scene with the point spread function (PSF) of the system:

$$I_{\text{out}} = I_{\text{in}} * \text{PSF} \tag{2}$$

One way to achieve this setup is with a "4f system", a basic telescope consisting of two convex lenses performing a cascade of two Fourier transforms (Fig. 1a). The system is so-named due to the placing of the first lens one focal distance, f, away from the object plane, producing a Fourier plane another distance f in front of the first lens. The second lens is then placed another distance f from the Fourier plane, producing a conjugate image plane a final distance $4f$ from the original object plane 1. The Fourier plane of such a system can be modulated in amplitude and phase, akin to a bandpass filter in signal processing, which alters the PSF of the system[8]. This simple case can be viewed as a convolutional layer with $C_{\text{in}} = C_{\text{out}} = 1$ and the flipped PSF as the single kernel. We will also refer to the flipped PSF as the kernel since the flipping is trivial.

Tiled kernels

Now suppose we want $C_{\text{out}} = n$ where $n > 1$. By spatially tiling the multiple kernels as the PSF of the system in an $A \times B$ grid, the output becomes the convolution of the input image with multiple 2D kernels, but now the n outputs are tiled laterally instead of stacked in depth. Consideration can be taken to ensure these outputs are non-overlapping by adjusting the shifts Δx and Δy, if desired. The PSF can be described as

$$PSF(x,y) = \sum_{a=1}^{A} \sum_{b=1}^{B} W_{aB+b}(x,y) * \delta(x - a\Delta x, y - b\Delta y), \tag{3}$$

and the resulting image formation as

$$I_{\text{out}}(x,y) = [I_{\text{in}} * PSF](x,y) = \sum_{a=1}^{A} \sum_{b=1}^{B} [I_{\text{in}} * W_i](x) * \delta(x - a\Delta x, y - b\Delta y) \tag{4}$$

where W corresponds to a standard multichannel kernel for a single channel input image. Hence we have a way to convolve a single input image with multiple 2D kernels, with the difference here being that the multiple output channels are tiled across the 2D image plane instead of stacked in a third "depth" dimension.

Cycled kernels

The next important extension is to incorporate $C_{\text{in}} = m$ where $m > 1$. If we needed to exactly imitate the digital CNN, we would need m different kernels for each of the m input channels. This could potentially be implemented with many of the single channel modules in parallel, with the addition of a relay that sums m outputs that correspond to the different depth slices of the same kernel, but this type of setup may be prohibitively complicated to build. If we slightly relax our requirements, we could again rely on Fourier optics to perform the summation. Now suppose we have tiled input images in addition to the kernels, simplifying to 1D for now for clarity:

$$I_{\text{in}}(x) = \sum_{j=1}^{m} I_j(x) * \delta(x - j\Delta x), \; PSF(x) = \sum_{i=1}^{m} W_i(x) * \delta(x - i\Delta x) \tag{5}$$

$$I_{\text{out}} = [I_{\text{in}} * PSF](x) = \sum_{i=1}^{n} [I_{\text{in}} * W_i](x) * \delta(x - i\Delta x) \tag{6}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \left([I_{\text{j}} * W_i](x) * \delta(x - j\Delta x) \right) * \delta(x - i\Delta x) \tag{7}$$

This combination of tiled images and tiled kernels results in some cycling of the kernels, but could still potentially offer enough degrees of freedom for certain tasks. Examples of tiled kernels and cycled kernels are shown in Fig. 1b.

Large PSFs

Finally, we were curious whether we even needed to think about tiling many small kernels, or rather if we could optimize for one large PSF, and leave it to the optimization to decide whether tiling was the optimal strategy. These approaches are compared in later simulations.

Nonlinear activation layer

In this paper, we primarily focus on the convolutional layers and apply nonlinear activations in simulations, when they used. However, we review some possible optically addressed approaches for the benefit of further research in this area, which also informs us on what type of activation functions to apply in simulation.

Nonlinear activation layers are crucial components in the neural network toolbox that allow for modeling of nonlinear relationships between input and output variables. Most commonly used is a rectified linear unit (ReLU), that simply sets all negative values to 0: $\mathrm{ReLU}(x) = \max\{0, x\}$. In an optical intensity-based system, there are no non-negative values, so the standard ReLU function does not directly apply. However, if we consider the purpose of the ReLU layer to zero out some fraction of the neurons below a threshold response level, then we hypothesize that we can accomplish a similar effect by shifting this threshold to a positive value.

This nonlinear behavior translates to an ideal optical element that is fully opaque when incident light is low intensity and fully transmissive when incident light is above a threshold. A perfectly binary switch is difficult to physically realize, so instead we sought a material that would be less transmissive to lower incident intensities and become more transmissive at higher incident intensities. In fact, this type of nonlinear response is remniscent of the PReLU (parametrized ReLU)[33] and (Swish)[34], only centered at a positive threshold instead of zero.

Bacteriorhodopsin (BR) is a membrane protein found in the bacterium *Halobacterium salinarium* that has been shown to exhibit logarithmic transmittance at one of its absorbance peaks of $\sim$570 nm[35]. The BR protein reversibly cycles between two states with the absorption of light, causing the appearance of a BR film to change from a deep purple to a light transparent purple. Furthermore, the shape of the transmittance function can be tuned by adjusting the concentration and pH of the BR solution before creation of the film[36]. Besides BR, visible light-responsive DASA-polymer conjugates have been synthesized that also demonstrate reversible tunable absorption properties dependent on incident light intensity[37]. This compound shows more of a linear transmittance function, becoming more transparent with higher intensity light[38]. We consider these in simulation to assess if a film or gel of these or similar substances could be used as a nonlinear activation layer in an ONN.

Fully-connected layer

The fully connected layer is so named because every input neuron is connected to every output neuron. The output of the previous layer is flattened into a single vector and multiplied with a matrix of size $D_{\text{out}} \times D_{\text{in}}$, where $D_{\text{in}} = H\text{in} \times W\text{in} \times C_{\text{in}}$. For a limited size of D_{out}, this elementwise product could be implemented by splitting the input image into D_{out} copies and performing an elementwise matrix multiplication using an amplitude mask. Unfortunately, this strategy becomes unreasonable as D_{out} grows.

Fortunately, fully connected layers may not be necessary, as fully convolutional networks have been successful in several prominent cases[15,39]. For example, if the goal is to classify among Z classes, it is possible to end with a convolutional layer that produces an output with Z channels, and then average each of the Z channels to produce a score for each of the classes. This suggests we may be able to implement a series of optical convolutional layers, divide the final output image into Z subregions, and then take the mean of each of those subregions.

Pooling layer

Pooling layers can be inserted, commonly between convolutional layers, to reduce spatial size and consequently computation. Pooling operations, for example "maximum", operate on each depth slice independently. One of the main reasons for pooling is to reduce the computing needs by reducing the dimensions of the next input image, which we do not need to consider in an optical system. Otherwise, pooling may improve training and prevent overfitting.

While it is not obvious how to take the spatial maximum of an optical signal without active sensing, average pooling can be approximated with a reduction in the spatial resolution of the image. This can be accomplished with a low-pass filter, i.e. a small iris placed in the pupil plane of the last $4f$ system. Spectral pooling is another interesting concept used in spectral representations of CNNs that carries over easily to our ONN setup[40]. Spectral pooling can be viewed as a generalization of low-pass filter pooling, in particular allowing for specific bandwidths to be selected for using custom amplitude masks. We only mention pooling layers here but do not explore these in our further experiments, as they were not essential for our analysis of convolutional layers.

Learning an Optical CNN

This may be moved to Methods.

To train our optical CNNs, we build the forward model of computational light transport in a Tensorflow framework and use built-in backpropagation and stochastic gradient-based optimizers to learn the weights, both of the PSFs and the optical elements.

Spatial domain optimization

Our strategy to building understanding of optical CNNs was to begin with a vanilla CNN and incrementally add constraints and features unique to optical models. Hence we begin in the spatial domain, assuming there exist optical elements that can produce the PSFs found by the optimization. From a standard CNN, we remove biases, impose non-negativity on the input and weights, and reduce the number of channels of the kernels while increasing their height and width. We also substitute our own nonlinearities for the standard ReLU function. Since some of the PSFs we try to optimize are much larger than normal, we use an FFT-based convolution to increase speed.

Phase mask optimization

After spatial domain optimization, the task still remains of connecting these desired PSFs to an optical implementation. As mentioned earlier, the Fourier plane of a $4-f$ system can be modulated with an aperture transfer function (ATF) to control the incoherent PSF of the optical relay:

$$PSF(x,y) = |\mathscr{F}\{ATF(k_x,k_y)\}(x,y)|^2, \tag{8}$$

where $k_x = \frac{x}{\lambda f}$ and $k_y = \frac{y}{\lambda f}$ denote spatial frequencies and λ is the wavelength of light. The ATF is a potentially complex function that can be decomposed into amplitude and phase as $ATF = A(k_x,k_y)\cdot\exp(i\Delta\phi(k_x,k_y))$, where local amplitude A can be implemented with a (usually binary) transparency mask, and phase shifts $\Delta\phi$ can be realized with a clear optical element of spatially varying thickness, which controls the optical path length and thereby phase shift induced by the element. To prevent loss of light and reduce the fabrication complexity of the ATF-defining optical element, we restrict our optimization to phase-only control.

Given the optimized PSF(s) from the spatial domain optimization, we now want to optimize phase masks that can generate these desired PSFs:

$$\underset{\phi}{\text{minimize}} \|PSF_{\text{opt}} - |\mathscr{F}\{e^{i\Delta\phi}\}|^2\|_{\text{F}}^2 \tag{9}$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm. Many different approaches have been taken to solve the phase retrieval problem[41], but since we are already using a learning-based approach above, we can use the Tensorflow framework here as well. We initialize a random phase mask and propagate training images through the current iterate of the $4f$ system. An error is calculted against the ground truth images where the training images are convolved with the desired PSF_{opt}, and then the gradients are backpropagated to the phase masks.

End-to-end optimization

Instead of separately optimizing the PSFs and then the corresponding phase masks, we also explored the possibility of an end-to-end optimization. To combine these steps into a single optimization problem, we implement a variant of the classification Tensorflow model where the phase mask heights were the optimizable parameter rather than the PSF weights themselves, such that the gradients of the classification error function are backpropagated all the way to the phase mask heights. Unfortunately, this approach did not always produce desired results, as we will show in the next section.

Evaluation of Optical CNNs

We build three convolution-based classifiers in simulation to better understand the performance of our optical CNN building blocks. We train these models to classify images from either the Google QuickDraw (PNG version) or CIFAR-10 dataset. Below we detail the setups and insights from each model.

Learned Optical Correlator

For our first experiment, we simulated a system with a single optical convolutional layer to confirm that our proposed optical convolution layer would function as expected. With a single convolutional layer, we expect to learn an optical correlator, which essentially performs template matching of the input image with the learned PSF. Here we are also able to apply end-to-end learning successfully (Fig. 3b).

While this was interesting, an optical correlator is not powerful enough for more difficult classification tasks, for example with natural images or with more categories (see Supplement for the same experiment with CIFAR-10 images). Furthermore, at this point with a single layer, we still only have a linear classifier.

Hybrid optoelectronic CNN

Next we keep one optical convolutional layer but connect the output image to further electronic computations. This allows the first layer to be performed at zero input power. We do this with grayscale CIFAR-10 images.

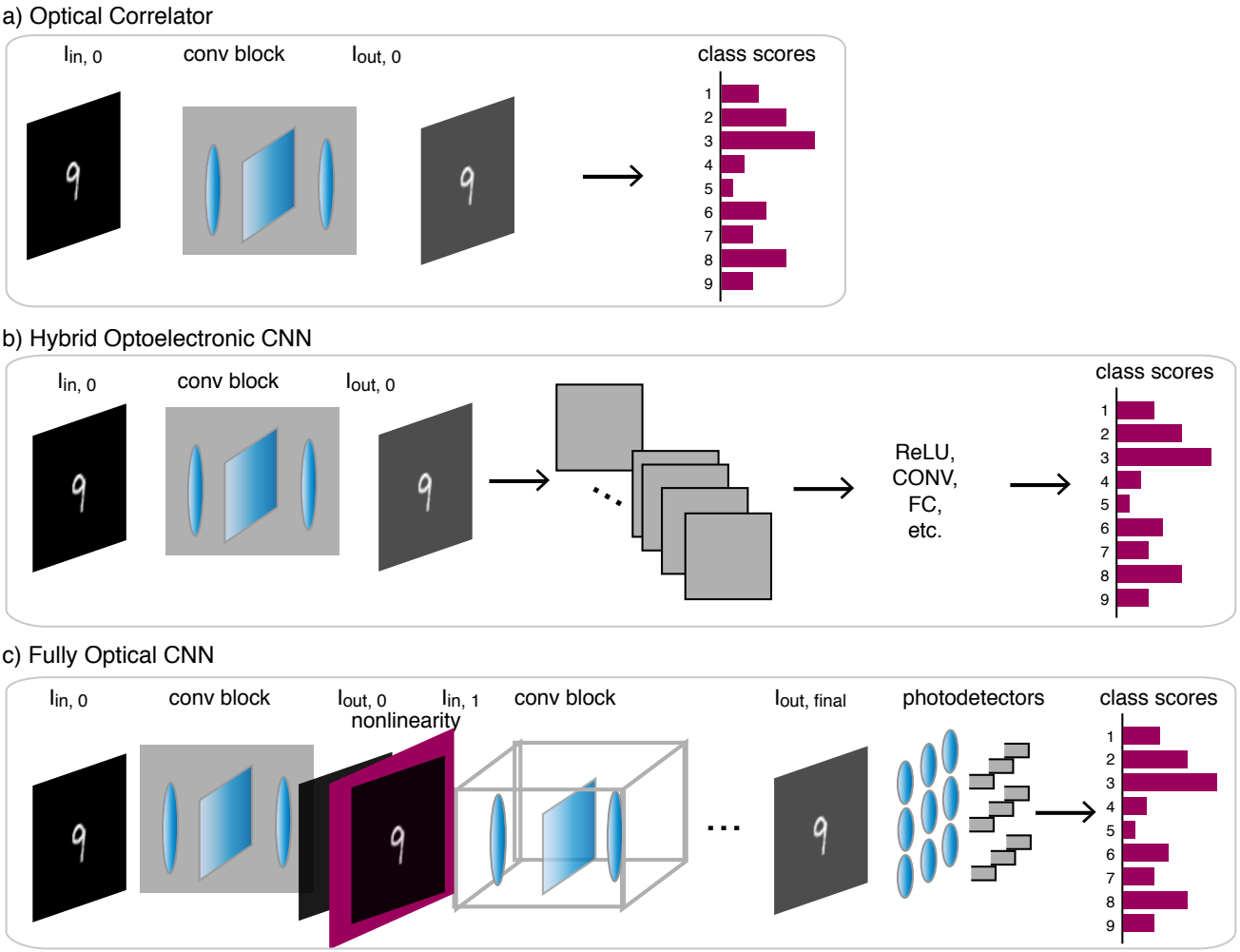

Figure 2. Optical classifier configurations. a) The optical correlator consists of a single optical convolutional layer. b) The hybrid optoelectronic CNN consists of one optical convolutional layer followed by one or more electronic CNN layers. c) The fully optical CNN is envisioned as a cascade of optical convolutional layers with nonlinear activation layers sandwiched between.

Pseudo-negative weights

Talk about the dual channel positive and negative weights.

When we attempted end-to-end optimization, we found that the optical element learned to simply replicate the input image (i.e. the PSF was a single point), so the fully connected layer was responsible for all the computation.

Color filters - Vincent

Alternatively, chromatic aberrations can be harnessed to encode color, so it would be interesting to explore multicolor masks as demonstrated in[42].

Fully optical CNN

The next goal was to create a fully optical CNN by incorporating additional optical convolutional layers with nonlinearity layers sandwiched in between. This doesn't fully work, but can discuss some results.

Optical Prototype

Implement the hybrid optoelectronic two-layer neural network. Goal is to show that the hybrid ONN can perform on par with the electronic ONN, with the same number of layers, and better than the electronic ONN with one fewer layer.

Table. 2: results

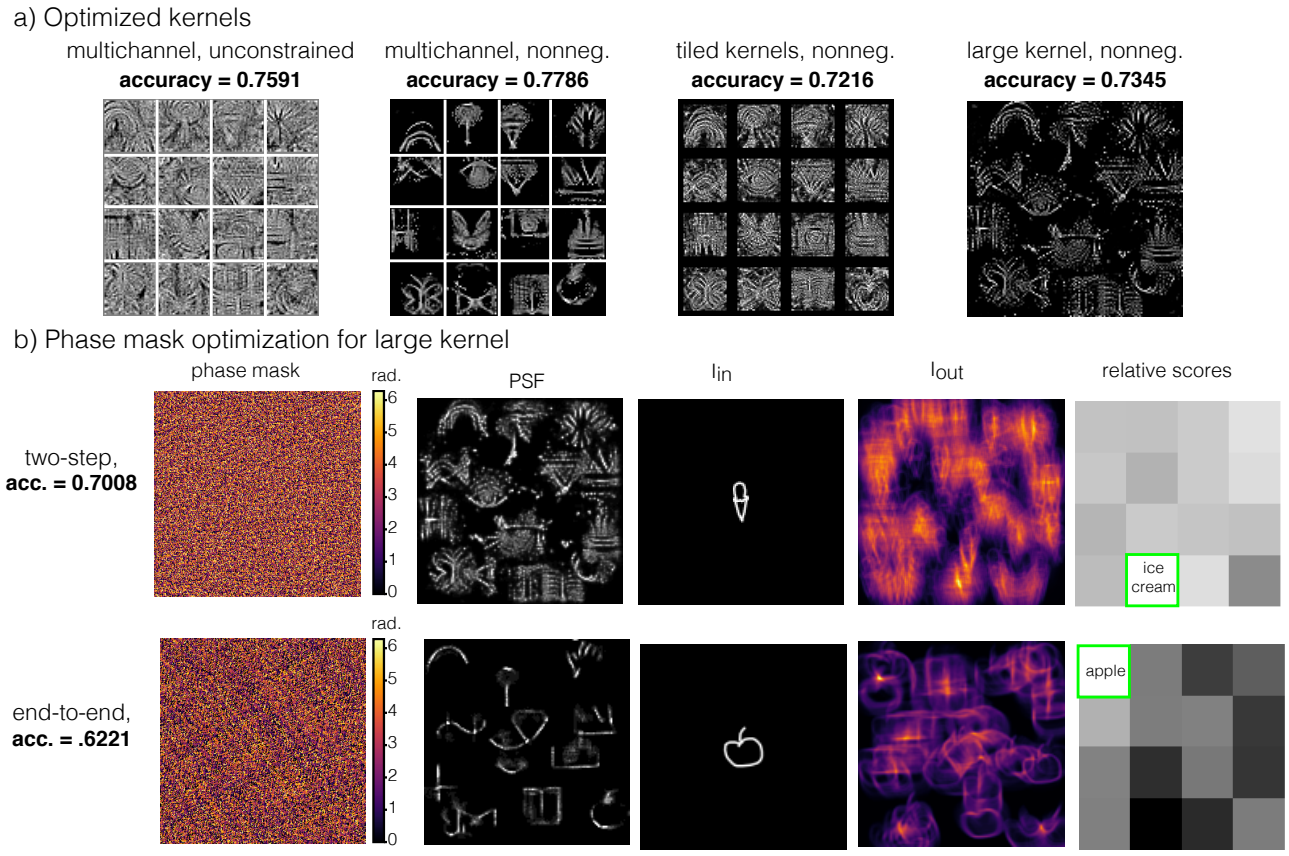

Figure 3. Optical correlator. a) Characteristic optimized kernels of a multichannel unconstrained Tensorflow convolutional layer, a multichannel nonnegative Tensorflow convolutional layer, a single channel optical convolutional layer with tiled kernels, and a single channel optical convolutional layer with a freeform large kernel. b) Optimized phase masks corresponding to the large kernel.

Discussion

- Not straightforward to generalize first optical conv. layer to multiple optical layers
- Discuss importance of negative weights – Vincent?
- Instead of trying to replicate a CNN exactly, could take advantage of optical transformations that aren't as practical in computations. For example, we use a 4f system for convolution, but this requires two extra lenses. Perhaps a single custom learned optical element can be used instead.
- In the future, exploit other properties of light (polarization, phase)
- Specifically, coherent light and holography, photonics

Conclusion, p. 14

Important step towards optical CNNs. We hope this will inspire more research in the area.

Methods

Training

Models were trained in Tensorflow.

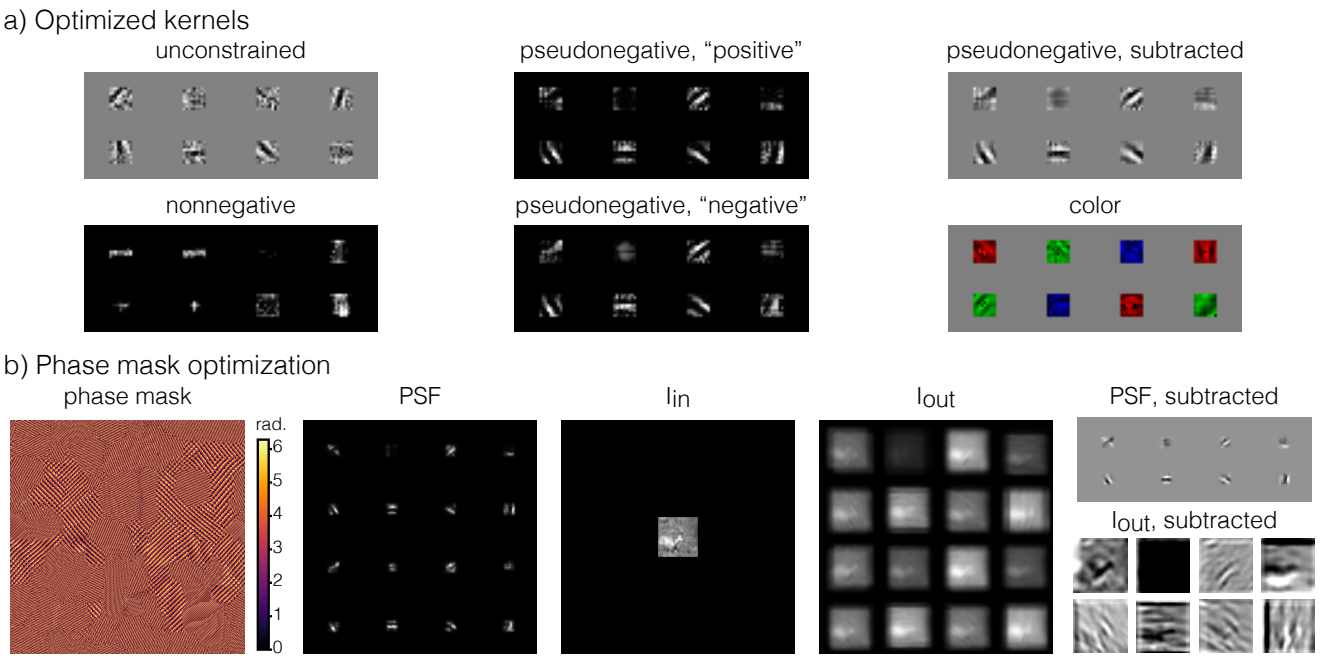

Figure 4. Hybrid optoelectronic CNN

Table 1. Classification accuracies on the grayscale CIFAR-10 dataset. Accuracies are the average of five trials.

Method	Accuracy
FC	0.2982
conv > FC, unconstrained	0.5186
conv > FC, nonnegative	0.3632
conv > FC, pseudonegative	0.5176
optical conv > FC, pseudonegative	0.4142
optical conv > FC, pseudonegative, refined	0.5096

Phase mask fabrication

Phase masks were fabricated via multilayer lithography.

Optical prototype

Laser light was scrambled through a rotating diffuser and illuminated a DMD. Images were relayed through a $4f$ system consisting of two convex $f = 200$ mm lenses. Images were captured by a sCMOS camera (Hamamatsu ORCA-Flash 4.0).

References

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nat.* **521**, 436–444 (2015).
2. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105 (2012).
3. Goodfellow, I. *et al.* Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680 (2014).
4. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440 (2015).
5. Silver, D. *et al.* Mastering the game of go with deep neural networks and tree search. *nature* **529**, 484–489 (2016).
6. LiKamWa, R., Priyantha, B., Philipose, M., Zhong, L. & Bahl, P. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 69–82 (ACM, 2013).

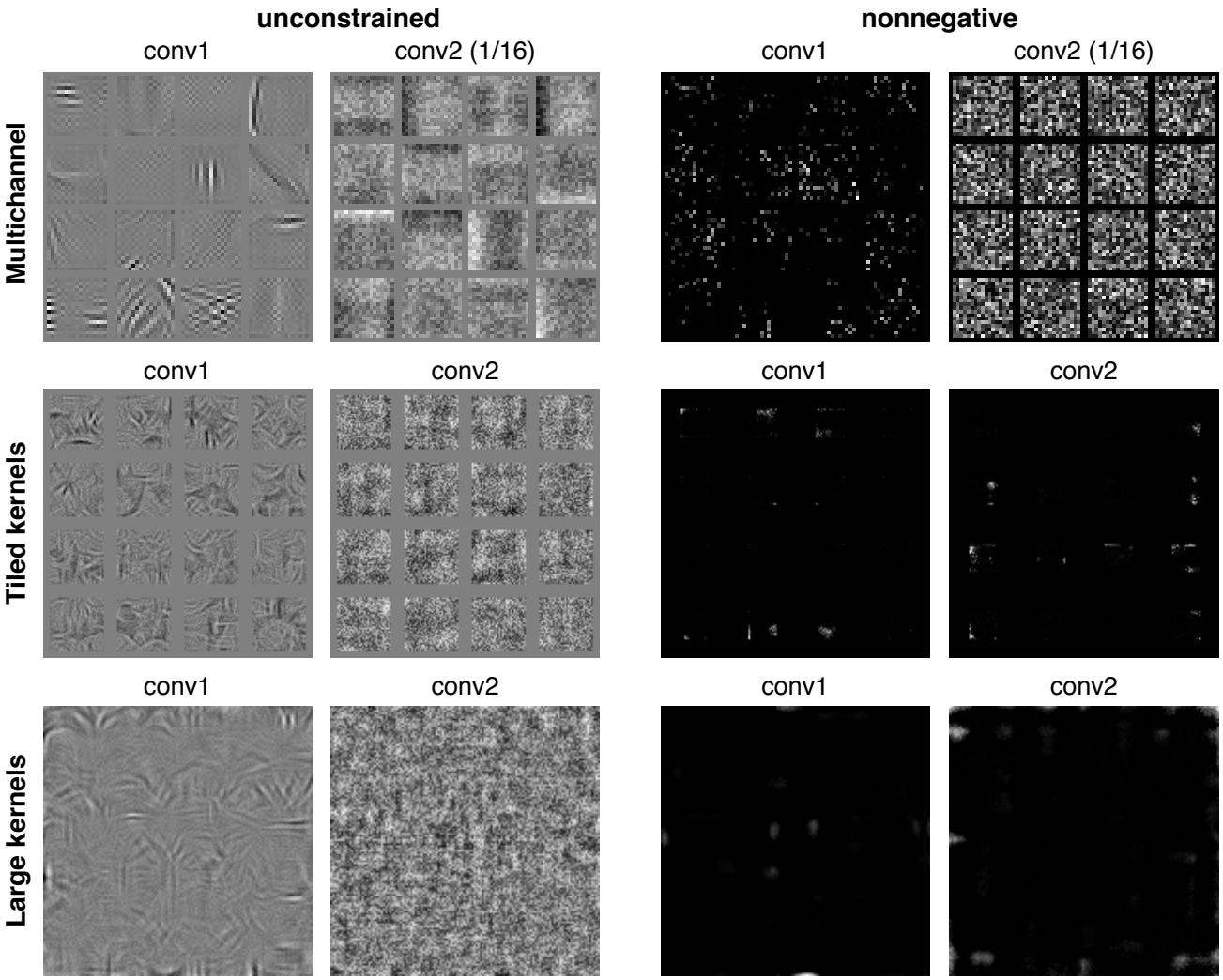

Figure 5. Variations leading to a fully optical CNN.

Table 2. Classification accuracies with various two-layer networks on 16 classes of the QuickDraw dataset. Accuracies are the average of five trials.

Method	Unconstrained	Nonnegative
multichannel (16)	0.810	0.810
tiled kernels (16)	0.8575	
large kernel	0.3632	

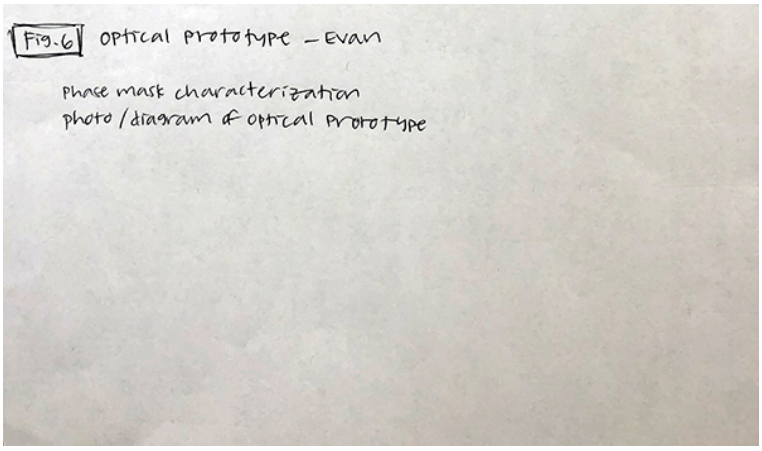

Figure 6. Optical prototype

7. Yang, L., Zhang, L. & Ji, R. On-chip optical matrix-vector multiplier for parallel computation. In *SPIE*, vol. 10, 004932 (2013).

8. Goodman, J. *Introduction to Fourier optics* (McGraw-hill, 2008).

9. Gibbs, H. *Optical bistability: controlling light with light* (Elsevier, 2012).

10. Christodoulides, D. N., Khoo, I. C., Salamo, G. J., Stegeman, G. I. & Van Stryland, E. W. Nonlinear refraction and absorption: mechanisms and magnitudes. *Adv. Opt. Photonics* **2**, 60–200 (2010).

11. Sun, J., Timurdogan, E., Yaacobi, A., Hosseini, E. S. & Watts, M. R. Large-scale nanophotonic phased array. *Nat.* **493**, 195–199 (2013).

12. Rechtsman, M. C. *et al.* Photonic floquet topological insulators. In *Lasers and Electro-Optics (CLEO), 2013 Conference on*, 1–2 (IEEE, 2013).

13. Shen, Y., Harris, N. C., Skirlo, S., Englund, D. & Soljačić, M. Deep learning with coherent nanophotonic circuits. In *Photonics Society Summer Topical Meeting Series (SUM), 2017 IEEE*, 189–190 (IEEE, 2017).

14. Han, S., Mao, H. & Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

15. Iandola, F. N. *et al.* Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360* (2016).

16. Gruev, V. & Etienne-Cummings, R. Implementation of steerable spatiotemporal image filters on the focal plane. *IEEE Transactions on Circuits Syst. II: Analog. Digit. Signal Process.* **49**, 233–244 (2002).

17. LiKamWa, R., Hou, Y., Gao, Y., Polansky, M. & Zhong, L. Redeye: Analog convnet image sensor architecture for continuous mobile vision. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 255–266 (2016). DOI 10.1109/ISCA.2016.31.

18. Ng, R. *et al.* Light field photography with a hand-held plenoptic camera. *Comput. Sci. Tech. Rep. CSTR* **2**, 1–11 (2005).

19. Levin, A., Fergus, R., Durand, F. & Freeman, W. T. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)* **26**, 70 (2007).

20. McGuire, M. *et al.* Optical splitting trees for high-precision monocular imaging. *IEEE Comput. Graph. Appl.* **27** (2007).

21. O'Toole, M. & Kutulakos, K. N. Optical computing for fast light transport analysis. *ACM Trans. Graph.* **29**, 164–1 (2010).

22. Chang, J., Kauvar, I., Hu, X. & Wetzstein, G. Variable aperture light field photography: overcoming the diffraction-limited spatio-angular resolution tradeoff. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 3737–3745 (IEEE, 2016).

23. Gregory, D. A. Real-time pattern recognition using a modified liquid crystal television in a coherent optical correlator. *Appl. optics* **25**, 467–469 (1986).

24. Manzur, T., Zeller, J. & Serati, S. Optical correlator based target detection, recognition, classification, and tracking. *Appl. optics* **51**, 4976–4983 (2012).

25. Javidi, B., Li, J. & Tang, Q. Optical implementation of neural networks for face recognition by the use of nonlinear joint transform correlators. *Appl. optics* **34**, 3950–3962 (1995).

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

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

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

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

30. Denz, C. *Optical neural networks* (Springer Science & Business Media, 2013).

31. Bueno, J. *et al.* Reinforcement learning in a large scale photonic recurrent neural network. *arXiv preprint arXiv:1711.05133* (2017).

32. Chen, H. G. *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*, 903–912 (2016).

33. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034 (2015).

34. Ramachandran, P., Zoph, B. & Le, Q. Searching for activation functions. *arXiv* (2017).

35. Downie, J. D. Nonlinear coherent optical image processing using logarithmic transmittance of bacteriorhodopsin films. *Appl. optics* **34**, 5210–5217 (1995).

36. Thoma, R., Oesterhelt, D., Hampp, N. & Bräuchle, C. Bacteriorhodopsin films as spatial light modulators for nonlinear-optical filtering. *Opt. letters* **16**, 651–653 (1991).

37. Ulrich, S. *et al.* Visible light-responsive dasa-polymer conjugates. *ACS Macro Lett.* **6**, 738–742 (2017).

38. Dolinski, N. D. *et al.* A versatile approach for in situ monitoring of photoswitches and photopolymerizations. *ChemPhotoChem* **1**, 125–131 (2017).

39. Lin, M., Chen, Q. & Yan, S. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

40. Rippel, O., Snoek, J. & Adams, R. P. Spectral representations for convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2449–2457 (2015).

41. Shechtman, Y. *et al.* Phase retrieval with application to optical imaging: a contemporary overview. *IEEE signal processing magazine* **32**, 87–109 (2015).

42. Shechtman, Y., Weiss, L. E., Backer, A. S., Lee, M. Y. & Moerner, W. Multicolour localization microscopy by point-spread-function engineering. *Nat. photonics* **10**, 590 (2016).

Acknowledgements (not compulsory)

Acknowledgements should be brief, and should not include thanks to anonymous referees and editors, or effusive comments. Grant or contribution numbers may be acknowledged.

Author contributions statement

Must include all authors, identified by initials, for example: A.A. conceived the experiment(s), A.A. and B.A. conducted the experiment(s), C.A. and D.A. analysed the results. All authors reviewed the manuscript.

Additional information

To include, in this order: **Accession codes** (where applicable); **Competing financial interests** (mandatory statement).

The corresponding author is responsible for submitting a competing financial interests statement on behalf of all authors of the paper. This statement must be included in the submitted article file.