

Figure 1: Optical setup and experimental demonstration. (a) A laser beam passes through a pinhole. The resulting point spread function (PSF) is measured by the metasurface array and the color camera. (b) The measured PSF is then convolved with displayed images to simulate imaging through the metasurface system, with the results captured by the same metasurface-camera setup. (c) A close-up view of the fabricated metasurface array. (d) A zoomed-in view showing the metasurface array mounted in front of the camera.

1 Optical convolution

4f optical system: The convolution operation can be implemented using the $4f$ optical system by leveraging the **convolution theorem**, which states that convolution in the spatial domain is equivalent to pointwise multiplication in the frequency domain. Let x denote the input (e.g., a signal or image), k denote the convolutional kernel, and $y = x * k$ denote their convolution. In the spatial domain, the convolution is defined as

$$y(i, j) = \sum_m \sum_n x(m, n) \cdot k(i - m, j - n), \quad (1)$$

Using the Fourier Transform (the first lens in the $4f$ system), the same operation can be expressed in the frequency domain as

$$Y(u, v) = X(u, v) \cdot K(u, v) = \mathcal{F}(x(i, j)) \cdot \mathcal{F}(k(i, j)), \quad (2)$$

where \mathcal{F} denotes the Fourier Transform, and \cdot indicates elementwise multiplication in the frequency domain. (u, v) represent the frequency-domain indices corresponding to the spatial-domain indices (i, j) , where u and v denote the horizontal and vertical frequency components, respectively. Then, the result is transformed back to the spatial domain via the inverse Fourier Transform (the second lens in the $4f$ system)

$$y(i, j) = \mathcal{F}^{-1}(Y(u, v)). \quad (3)$$

Meta-optics: The convolution operation can be performed using Meta-optics, where the resultant image corresponds to the input convolved with the system's point spread function (PSF). The PSF characterizes the response of an optical system to a point source, determining how the system processes and blurs light. For a discrete input $O[n, m]$ and a discrete point spread function $PSF[n, m]$,

the intensity in the image plane $I[n, m]$ is given by

$$I[n, m] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} O[i, j] \cdot PSF[n - i, m - j]. \quad (4)$$

The two-dimensional discrete convolution is formally defined as

$$(f * g)[n, m] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} f[i, j] \cdot g[n - i, m - j], \quad (5)$$

where $f[i, j]$ represents the input function (e.g., the pixel intensity values of the input image), and $g[i, j]$ represents the kernel function (e.g., the point spread function, PSF, of the optical system).

2 Meta-Optical Implementation

PSF-engineered meta-optics are inversely designed using back-propagated loss and differentiable parameters. In our experiments, we tune the meta-optics by varying the scatterer width while maintaining other structural parameters, such as height and period, constant. To ensure high transparency across the entire visible spectrum, we utilize silicon nitride nanorod structures. The periodic arrangement of these nanorods introduces an additional phase delay to the incoming wavefront, which depends on the structural parameters. We compute this phase delay as a function of the silicon nitride pillar width at three distinct wavelengths—450, 532, and 635 nm—using the S4 solver.

For any meta-optics comprising an array of pillar structures, we employ width-to-phase conversion to generate a two-dimensional phase profile. We then compute wave propagation to the predetermined focal plane using the band-limited angular spectrum method, followed by calculating the PSF of the meta-optics. To optimize the meta-optics so that the resulting PSFs closely match two-dimensional convolutional kernels, we (1) ensure the width-to-phase conversion remains differentiable and (2) define a loss function that quantifies the discrepancy between simulated PSFs and convolutional kernels.

First, we obtain a lookup table from the S4 solver, which establishes a one-to-one relationship between the scatterer width and the additional phase shift. However, this relationship is neither differentiable nor directly applicable for gradient-based backpropagation. To address this, we define a differentiable width-to-phase mapping using a Gaussian-based fitting function

$$f_{\lambda}(w) = 2\pi n_{eff} L / \lambda + A \exp((w - B)^2 / C) - f_0, \quad (6)$$

where $f_{\lambda}(w)$ is additional phase at certain wavelength (λ) and scatterer width (w), n_{eff} and L are the effective refractive index and height of the pillars, and A , B , and C are fitting parameters. Three fitting functions corresponding to three different wavelengths (450, 532, and 635 nm) are used to represent red, green, and blue colors.

Next, after converting the meta-optics into a phase map, we simulate the PSFs and evaluate the loss relative to the convolutional kernels at each wavelength. The simulated PSFs and ground-truth convolutional kernels are normalized, followed by calculating the root mean square (RMS) error for individual pixels after normalization, and then computing the RMS error across all three colors. To optimize the polychromatic meta-optics for ideal three-color PSFs, we use the Adam optimizer with a learning rate of 0.005.

Nanofabrication enables the transformation of the design into physical meta-optics. First, we deposit silicon nitride onto a quartz substrate using plasma-enhanced chemical vapor deposition. The surface is then cleaned using acetone and isopropyl alcohol, followed by oxygen plasma treatment. Next, an electron beam resist is applied to the chip and patterned via electron beam lithography, creating a periodic structure with a 283 nm pitch and a minimum feature size of 60 nm. Instead of using resist, we deposit alumina as a hard mask using an electron beam evaporator, then transfer the pattern onto the silicon nitride using a dry etching process.

3 Optical characterization

Optical Experiment Setup Figure S1 illustrates the optical setup used to characterize the PSFs and convolved images produced by the meta-optics. We employ a set of lasers (CPS450, CPS532,

and CPS635; Thorlabs) along with a pinhole ($\phi = 25 \mu\text{m}$) to generate a point source for PSF measurements. The meta-optics is positioned approximately 2.4 mm in front of the color camera (GT 1930C; Allied Vision). Each color pixel of the camera has a size of $5.86 \mu\text{m}$, and the designed PSFs feature an enlargement factor of 2, which determines the number of camera pixels corresponding to a single pixel in the convolutional kernel. For instance, a 3×3 convolutional kernel maps to a 6×6 pixel region on the color camera.

For image convolution, we replace the pinhole with an OLED display, projecting an arbitrary image onto it to be convolved with the PSFs of the meta-optics on the color camera. During this replacement, the configuration of the meta-optics and color camera remains unchanged.

For segmentation tasks, we use a total of 56 digital kernels; however, the number doubles for meta-optics, as negative optical intensity cannot be practically achieved on the camera. To manage the large number of meta-optics and image data, we distribute the kernels across two separate chips — one containing the first 8 and 16 kernels of the U-Net, and the other containing the remaining 24 kernels. In our experiments, we first measure the PSFs and convolved image sets for one chip before switching to the second chip to complete the full set of measurements.

Point Spread Functions (PSFs) Each meta-optics chip features metalenses at its edges to ensure precise alignment. To achieve near-normal incidence of light, we position the laser and pinhole at a considerable distance from the meta-optics. The meta-optics is then carefully aligned with the camera sensor in terms of tilt, rotation, and translation. Proper alignment is verified by ensuring that all four corner metalenses focus simultaneously, maintaining parallel orientation with the camera.

PSF measurements are performed five times to balance signal-to-noise ratio while avoiding saturation, and the resulting PSFs are then combined. The central 6×6 pixel regions of each meta-optics are cropped for analysis. Measurements are conducted using three different wavelength lasers (450 nm, 532 nm, and 635 nm) without adjusting any other optical components except the laser holder. This approach guarantees that the meta-optics simultaneously captures polychromatic PSFs on the color camera, enabling the visualization of colorful display images. However, a spectral discrepancy arises between the lasers and the display images, as the display exhibits a broader wavelength range than the lasers.

Convolved Images After replacing the pinhole with the display, we optimize the image size to maximize visibility while avoiding overexposure, resulting in a final resolution of 46×26 pixels on the camera. Image display and capture are automated to efficiently process approximately 4,000 images for training and 1,000 images for testing the neural network. With an exposure time of about 1.5 seconds per image, capturing the entire dataset for two sets of chips requires approximately 10 hours.

Energy Consumption In a hybrid ONN, energy is required for both image capturing and digital backend processing, similar to conventional digital networks that also rely on initial image acquisition. For the color camera used in our experiments, we calculate an energy consumption of approximately 28.5 pJ per pixel. The segmentation dataset has an input image size of 384×216 pixels, corresponding to an energy requirement of about 2.36 mJ.

Since our hybrid ONN requires additional images to match the number of kernels, we employ 112 meta-optics—56 for positive kernels and 56 for negative kernels—which significantly increases the total energy needed for image capture. However, instead of acquiring full-resolution images, we capture reduced (compressed) convolved images at 46×26 pixels per kernel, resulting in a total energy consumption of 3.82 mJ. While our approach requires slightly more pixels and energy for capturing images in hybrid ONN, it remains feasible across diverse tasks, including MNIST and CIFAR-10 classifications as well as segmentation.

On the other hand, in digital computation, hybrid ONNs typically require significantly fewer operations, leading to substantial energy savings. Our GPU consumes approximately 15 pJ per floating-point operation (FLOP) and 30 pJ per multiply-accumulate (MAC) operation (assuming one MAC equals two FLOPs). The U-Net model we use performs approximately 239 million MAC operations, whereas the hybrid ONN requires only about 65 million MAC operations. This reduction in computation lowers the energy consumption to 7.37 mJ for the all-digital network and 2.01 mJ for the hybrid ONN.

Overall, the total system-level energy consumption is estimated to be 9.73 mJ for the all-digital approach and 5.83 mJ for the hybrid ONN—representing a reduction of more than 40%.

Table 1: System level energy consumption (in joules) per a single image segmentation task in each step of the computer vision depends on the network architecture.

Network architecture	Optical frontend	Digital backend	System
Original CNN with optimal camera pixels	2.36×10^{-3}	7.37×10^{-3}	9.73×10^{-3}
Our hybrid optical/digital CNN with optimal camera pixels	3.82×10^{-3}	2.01×10^{-3}	5.83×10^{-3}

4 Noise Robustness in Wide Finite-Width Networks

Let the single-layer neural network be parameterized by θ , and its output for input x be $f(x; \theta)$. The NTK for inputs x and x' is defined as

$$\Theta(x, x') = \nabla_{\theta} f(x; \theta)^{\top} \nabla_{\theta} f(x'; \theta), \quad (7)$$

where $\nabla_{\theta} f(x; \theta)$ represents the gradient of the network output with respect to the parameters.

4.1 Noise Perturbation to Parameters

Suppose a small noise δ is added to the parameters, resulting in $\theta' = \theta + \delta$. The perturbed NTK is expressed as

$$\Theta_{\delta}(x, x') = \nabla_{\theta'} f(x; \theta')^{\top} \nabla_{\theta'} f(x'; \theta'). \quad (8)$$

Expanding $\nabla_{\theta'} f(x; \theta')$ using a first-order Taylor approximation

$$\nabla_{\theta'} f(x; \theta') \approx \nabla_{\theta} f(x; \theta) + H(x; \theta) \delta, \quad (9)$$

where $H(x; \theta)$ is the Hessian matrix of $f(x; \theta)$ with respect to θ .

Substituting this into $\Theta_{\delta}(x, x')$

$$\Theta_{\delta}(x, x') \approx \Theta(x, x') + \delta^{\top} H(x; \theta)^{\top} \nabla_{\theta} f(x'; \theta) + \delta^{\top} H(x'; \theta)^{\top} \nabla_{\theta} f(x; \theta) \quad (10)$$

$$+ \delta^{\top} H(x; \theta)^{\top} H(x'; \theta) \delta. \quad (11)$$

The perturbation $\Delta\Theta(x, x')$ is given by

$$\Delta\Theta(x, x') = \Theta_{\delta}(x, x') - \Theta(x, x'). \quad (12)$$

4.2 Bounding the Perturbation

The magnitude of $\Delta\Theta(x, x')$ satisfies

$$\|\Delta\Theta(x, x')\| \approx \|\delta\| \cdot (\|H(x; \theta)\| \cdot \|\nabla_{\theta} f(x'; \theta)\| + \|H(x'; \theta)\| \cdot \|\nabla_{\theta} f(x; \theta)\|) \quad (13)$$

$$+ \|\delta\|^2 \cdot \|H(x; \theta)\| \cdot \|H(x'; \theta)\|. \quad (14)$$

For small perturbations $\|\delta\| \ll 1$, the second-order term is negligible, and we obtain

$$\|\Delta\Theta(x, x')\| \sim O\left(\frac{\|\delta\|}{m}\right), \quad (15)$$

which vanishes as $m \rightarrow \infty$.

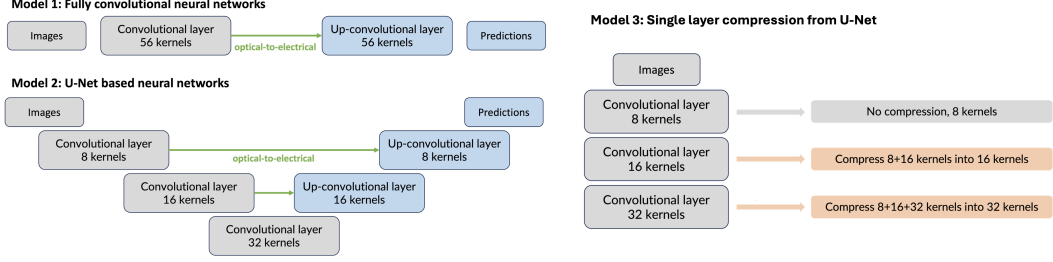


Figure 2: Model Architectures: (1) a fully convolutional network (FCN) with 56 kernels in the convolutional layer, and (2) a U-Net with a progressively increasing number of kernels (8, 16, 32). (3) The parallelized U-Net model compression. The initial convolutional layer starts with 8 kernels, with no compression applied at this stage. Subsequent layers progressively compress kernel counts.

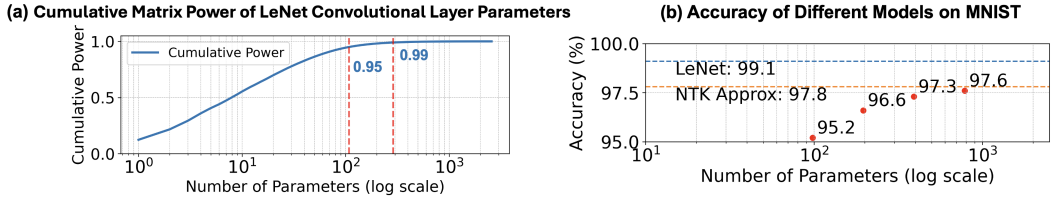


Figure 3: **NTK compressibility analysis.** (a) Cumulative eigenvalue power of the Parameter matrix $g = J^T J$ for LeNet’s convolutional layers, showing that 95% and 99% of power are concentrated in just 108 and 288 parameters. (b) Accuracy of models with varying parameter counts.

5 Parameter-Space Compressibility

To examine redundancy of the convolutional layer, we compute the Jacobian $J \in \mathbb{R}^{n \times p}$ of LeNet outputs with respect to its 2,572 convolutional parameters. We then form the parameter Gram matrix $J^T J \in \mathbb{R}^{p \times p}$ and analyze its cumulative eigenvalue spectrum to quantify the parameter effective dimensions. Figure 3.a plots the cumulative eigenvalue power of $J^T J$. 108 parameters account for 95% of the total power, and 288 parameters account for 99%, indicating strong compressibility. To validate this, we trained compressed models with varying parameter counts (Figure 3.b), and analyzed the trade-off between model size and accuracy. For example, a model with just 98 parameters in the convolutional layer achieved 95.2% accuracy, closely matching the 95% cumulative NTK power threshold. Notably, when the cumulative matrix power exceeded 99%, further increases in parameter count without changes to the model structure yielded diminishing returns. For example, increasing parameters from 98 to 784 (twice that of Compressed Meta), only improved accuracy by 0.5%. This marginal gain did not surpass the accuracy predicted by NTK analysis, highlighting the inefficiency of overparameterization and suggesting the use of dimensionality reduction. The Gram matrix can help estimate the efficient number of parameters for a given task.

6 Segmentation Implementation Details

6.1 Teacher: U-Net

The U-Net begins with an initial convolutional layer that applies 8, 16, and 32 kernels in sequence to capture increasingly complex feature representations at each level. This is followed by an up-convolutional process, which restores spatial dimensions in reverse order (32, 16, 8, and finally, 1 channel for output). Figure.S 2 demonstrates U-Net neural networks. Gray blocks represent convolutional layers implemented in the optical domain, while blue blocks indicate up-convolutional layers in the electronic backend, which reconstruct the spatial dimensions to produce the final segmentation map. The optical-to-electrical conversions enable smooth transitions between the optical and electronic domains, employing a hybrid processing approach to achieve efficient segmentation.

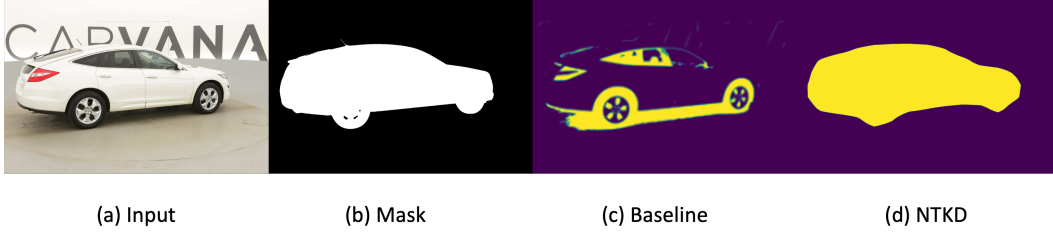


Figure 4: Semantic Segmentation Results for Car Segmentation. Top row: (left to right) Original Input image, Label Mask, Training with FCN (mIoU: 69%), and Training with NTKD (mIoU: 92%).

6.2 Student ONN: Parallelized U-Net

Based on our previous experiments, ONNs could not effectively implement multiple layers directly. As the number of layers increased, misalignments propagated through each layer, resulting in noisy outputs. To address this, we initially leveraged a single convolutional layer (FCN model) to approximate the entire deep neural network (also demonstrated in Figure.S 2). The FCN configuration started with a downsampling step, where the input image (with 3 channels) was processed by a convolutional layer with 56 kernels. The network then performed upsampling using bilinear interpolation, followed by another convolution to generate the final segmentation map. Next, we compressed the parallelized U-Net model. Initially, the first layer operated with 8 kernels without compression. In subsequent layers, we compressed 8 and 16 kernels into a single 16-kernel representation and compressed 8, 16, and 32 kernels into a 32-kernel representation. Then, we used the U-Net backend as the parallelized U-Net.

6.3 Demos of Segmentation Model

Figure.S 4 illustrates the segmentation performance on car images using a compressed model with 56 kernels, demonstrating that computational efficiency can be achieved without compromising predictive accuracy. This strategy builds upon a teacher-student framework to facilitate effective model compression suitable for real-world deployment scenarios. Performance is evaluated in terms of mIoU under various training conditions: direct training without transfer learning yielded a score of 69%, and our proposed NTKD optimization further enhanced performance to 92%.