Supplemental Document

Optics EXPRESS

Co-designed metaoptoelectronic deep learning: supplement

CARLOS MAURICIO VILLEGAS BURGOS,[1] PEI XIONG,[1] LIANGYU QIU,[1] YUHAO ZHU,[2,3] AND A. NICKOLAS VAMIVAKAS[1,*]

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

Supplement DOI: https://doi.org/10.6084/m9.figshare.21731222

Parent Article DOI: https://doi.org/10.1364/OE.479038

Co-designed metaoptoelectronic deep learning: supplemental document

When working with a 4f system whose PSF we want to engineer by optimizing the transfer function yielded by a phase mask in the Fourier plane, we need to figure out what the size of this phase mask needs to be. We also need to know what is the appropriate size of the "phase pixels" (subregions of the phase mask that impose a specified phase to light that is incident on them) in the phase mask's plane. Both the metasurface size and the phase pixel size depend on the specifications of the optical elements on our system, such as the focal lengths of the lenses and the pixel size of the display used to project images.
In addition to finding the values of these parameters, the values of the phase imparted on incident light by each of the phase pixels needs to be found. This is done through a Phase Optimization process that iteratively updates the phase profile via a gradient descent algorithm and computes its yielded PSF until it closely resembles the target that contains the kernels of the network's convolutional layer. This phase optimization process accounts for certain effects that affect the appearance of the PSF yielded by the phase mask, which are caused by some traits of the metasurface's layout and structure.
Once the required phase values for each phase pixel are found, a design file containing the required geometrical parameters of the nano-tokens in each metasurface unit cell can be produced and then used to fabricate the metasurface.

1. METASURFACE SIZE

The picture that is projected in the display plane is sampled with a pixel size of Δx_{sample} that is not necessarily the same as the size of the display pixels $\Delta x_{\text{display}}$. In order to correctly perform optical convolution with this system, the sampling of the phase mask needs to match with that of the field that is incident on it. For that effect, we need to review the fundamentals of optical convolution. Let $U_I(\xi, \eta)$ denote the scalar field distribution at the front focal plane of a thin lens, and let $F_I(f_\xi, f_\eta) = \mathcal{F}\{U_I(\xi, \eta)\}$ be its Fourier transform, where (f_ξ, f_η) are spatial frequencies (i.e. coordinates in the frequency domain). Then, the scalar field $U_F(x, y)$ at the back focal plane of the lens is given by[1]:

$$U_F(x, y) = \frac{1}{i\lambda f} F_I\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right), \tag{S1}$$

where λ is the wavelength of the propagating light field, f is the focal length of the lens, and i is the imaginary number unit with $i^2 = -1$.

From Eq. (S1) we know that the spatial frequency component $(f_\eta, f_\xi) = \left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right)$ of the input picture on the display plane is mapped to point (x, y) on the Fourier plane. With this, we can determine what the metasurface size needs to be so that it can cover the largest spatial frequency components of the display plane's field. The largest (absolute value of) spatial frequency component of the input picture that we have to consider is given by one half of the sampling rate of the display in the input plane, that is, $|f_\eta|_{\max} = |f_\xi|_{\max} = 1/(2\Delta x_{\text{sample}})$. Therefore, the size of the metasurface needs to be $\frac{\lambda f}{\Delta x_{\text{sample}}} \times \frac{\lambda f}{\Delta x_{\text{sample}}}$.

In our system, we have $f = 125\,\text{mm}$, $\lambda = 633\,\text{nm}$, and $\Delta x_{\text{display}} = 7.56\,\bar{}\text{m}$; so, using $\Delta x_{\text{sample}} = 10\Delta x_{\text{display}}$ (projecting pictures upscaled by a factor of 10 in the display), we can have the metasurface be $1.05\,\text{mm} \times 1.05\,\text{mm}$ in size.

2. PHASE PIXEL SIZE

During the phase optimization process, both the phase mask and the region of the PSF containing the convolution kernels are represented as numerical arrays with $N \times N$ pixels. From the way in which the target PSF is constructed, we deemed $N = 500$ to be a suitable value to encode the phase mask. With this, the size of the phase pixels is $\frac{1.05\,\text{mm}}{500} = 2.1\,\bar{}\text{m}$. Due to fabrication constraints and the conditions in the simulations used to study the amplitude and phase response of nano-tokens, the size of the unit cells in the metasurface is chosen to be $350\,\text{nm} \times 350\,\text{nm}$, with

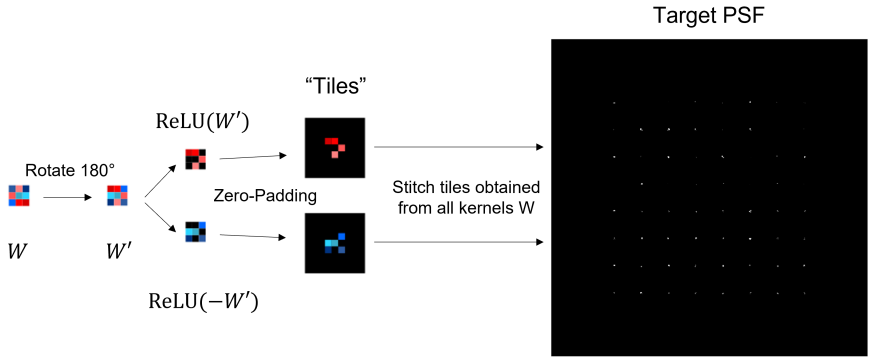

Fig. S1. Construction process to build the numerical array that contains the convolution kernels of the original convolutional layer. From each kernel W, a pair of tiles is obtained. These tiles are built by first rotating W by 180 degrees, yielding W'. Then W' is separated into (the absolute value of) its positive and negative parts, and some zero-padding is added to both. The pairs of tiles obtained this way from all kernels of the convolutional layer are then stitched together to form the numerical array that is referred to as the target PSF. The target PSF of our system is shown on the right portion of this figure.

each unit cell containing one nano-token at its center. Since the phase pixel size is $2.1\bar{}\text{m} \times 2.1\bar{}\text{m}$, this means that each phase pixel is composed by a group of 6×6 identical unit cells.

3. TARGET PSF DESIGN CONSIDERATIONS

The convolution kernels inserted into the system's PSF need to be separated by a distance that is large enough so that when the input picture is convolved with them, there is no overlay between the sub-images that result from each individual convolution. Additionally, it is necessary for the inserted kernels W to be rotated by 180 degrees, because the digital convolution layers are really computing the cross-correlation between the convolution kernels and the input picture, which is equivalent to performing convolution between the kernels rotated by 180 degrees and the input picture. The rotated kernels will be denoted as W'.

As a final step, it is necessary to split each kernel W' into a pair of sub-kernels: one containing the positive values of of W' and the other containing the (absolute value of) negative values of W'. In a more technical description, the first sub-kernel contains $\text{ReLU}(W')$ and the second sub-kernel contains $\text{ReLU}(-W')$, where ReLU is the rectifier linear unit function, which is applied element-wise on a numerical array and maps non-negative numbers to themselves and negative numbers to zero. This step is necessary in order to be able to implement convolution with kernels that contain non-positive values through optical convolution, which would be otherwise not possible because a PSF can only contain positive values. It can be easily proven that for any numerical array A containing real numbers, the equality $A = \text{ReLU}(A) - \text{ReLU}(-A)$ holds true. Thus, if a picture I_{in} is convolved with blur kernels containing $\text{ReLU}(W')$ and $\text{ReLU}(-W')$, and then the resulting sub-images containing the results of these convolutions are subtracted during digital post-processing, the result of that subtraction will be equivalent to the result of convolving I_{in} with W'.

After obtaining $\text{ReLU}(W')$ and $\text{ReLU}(-W')$, some zero-padding is added, forming a pair of "tiles" that should be slightly larger than the pictures projected into the optical system, so that the convolved sub-images do not overlap. Once the pair of tiles associated with each kernel W are obtained, they are stitched together into a numerical array that will serve as the target PSF. They are stitched in a way so that the tiles containing the positive parts of the W' are placed in the top half of the numerical array while those with the negative parts of the W' are placed in the bottom half of the array. Additionally, within each half of the array, the tiles are arranged from left to right and from top to bottom in the order of the kernel indices. The process used to build the target PSF as described in this sub-section is illustrated in Fig. S1.

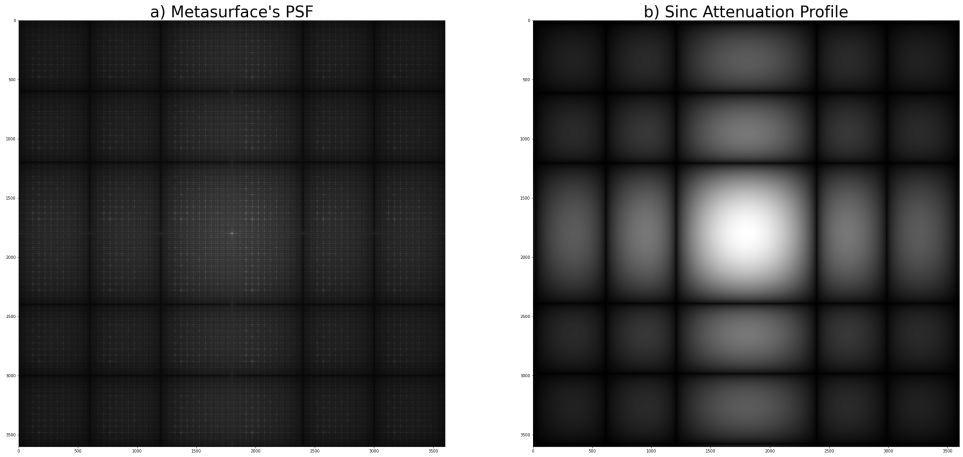

Fig. S2. a) PSF yielded by the metasurface. Its main features are the delta-like function on center, and an array of off-center copies of the target PSF shown in Fig. S1. Everything is affected by an attenuation profile described by a sinc function. b) A plot of the sinc attenuation profile, shown for comparison. Both plots have an increased contrast, to facilitate the observation of their main features.

4. PHASE OPTIMIZATION CONSIDERATIONS

The target PSF constructed with the process described in the previous subsection is encoded on a numerical array with $N \times N$ pixels. A gradient descent process is used to find the $N \times N$ numerical array containing the phase profile Φ that minimizes a loss function that measures the difference between the yielded PSF and the target PSF; this process will be referred to as phase optimization moving forward.

However, when the phase profile contained in Φ is implemented by the physical metasurface, the PSF that is yielded by the latter is different from the one yielded by the digital numerical array. The PSF yielded by the numerical array Φ contains the convolution kernels of the convolutional layer, but it is just a portion of the PSF that is yielded by the physical metasurface. This is because the PSF computed from the numerical array assumes that the complete region covered by the phase pixels on the array imparts that phase value on incident light; meanwhile, in the metasurface's phase pixels, that phase value is imparted on incident light only on the area of the phase pixel that is covered by nano-tokens. The layout of the metasurface makes the PSF it yields to look like the one shown in Fig. S2.

There is an explanation for the structure behind this yielded PSF, as well as a way to quantify the differences it has with the one computed from the numerical array Φ, so that they can be accounted for during the phase optimization process. Let $R_c(x, y)$ denote the complex reflection coefficient that the metasurface would imbue on light that is incident on point (x, y) of the metasurface plane if the phase values contained in array Φ were imparted on the whole area of the phase pixels on which the metasurface is divided (instead of just on the area covered by the nano-tokens). We can express the complex reflection coefficient profile from the real metasurface $R(x, y)$ in terms of $R_c(x, y)$ with the following equation:

$$R(x, y) = \left(1 + s(x, y) * \left(\text{comb}\left(\frac{x}{\Delta x}, \frac{y}{\Delta y}\right)(R_c(x, y) - 1)\right)\right) \text{rect}\left(\frac{x}{L_x}, \frac{y}{L_y}\right), \tag{S2}$$

where Δx and Δy are the transverse dimensions of the phase pixels (both equal to 2.1 ¯m in our case), L_x and L_y are the transverse dimensions of the substrate where the metasurface lies, comb is the comb function, which consists on an infinite sum of delta functions whose centers are equally spaced along both transverse dimensions, rect is a binary function that is equal to 1 in the points where the absolute value of both of its arguments is less than or equal to $\frac{1}{2}$ (and is equal to 0 elsewhere), and $s(x, y)$ is a binary function that is 1 in the area of the phase pixel that is covered by nano-tokens and 0 elsewhere, and it shall be referred to as the pixel shape function.

The explanation for the expression in Eq. (S2) is as follows: The product between $R_c(x, y)$ and the comb function results in a collection of equally-spaced delta functions that peak at the centers

of each phase pixel and whose area under the curve corresponds to the values that $R_c(x, y)$ had at those points (which are the values on each phase pixel). Then, convolving that collection of delta functions with the pixel shape function results in a collection of equally-spaced regions that have a reflection coefficient value of $R_c(x, y)$ on the area covered by nano-tokens and a value of 0 elsewhere. However, the substrate of the metasurface is reflective, not opaque, meaning that the area of the metasurface that is not covered by nano-tokens should have a reflection coefficient equal to that of the substrate (assumed to be 1, for simplicity). For that effect, 1 has to be subtracted from $R_c(x, y)$ before the product with the comb function and then 1 has to be added after the convolution with the phase pixel function. With that, now $R(x, y)$ has the values of $R_c(x, y)$ in the areas covered by the nano-tokens and a value of 1 elsewhere. The final product with the rect denotes that the reflection coefficient is equal to 0 at the points that lie beyond the substrate's area.

The PSF yielded by the metasurface is equal to the magnitude squared of the Fourier transform of its reflection coefficient profile $R(x, y)$. The Fourier transform of $R(x, y)$ is given by:

$$\begin{aligned}\mathcal{F}\{R(x, y)\} = &\left[\delta\left(f_x, f_y\right) + \mathcal{F}\{s(x, y)\}\left(\mathcal{F}\{R_c(x, y) - 1\} * \text{comb}\left(\Delta x f_x, \Delta y f_y\right)\right)\right] \\ &* \left[L_x L_y \text{sinc}\left(L_x f_x, L_y f_y\right)\right],\end{aligned} \tag{S3}$$

where $\text{sinc}(A, B)$ is an abbreviation for the product $\text{sinc}(A)\text{sinc}(B)$, and the sinc function is defined as $\text{sinc}(z) = \frac{\sin(\pi z)}{\pi z}$. The expression in Eq. (S3) provides us with an explanation for the traits that the PSF in Fig. S2 has. The Fourier transform of $(R_c(x, y) - 1)$ is convolved with a comb function, which results on an array of copies of $\mathcal{F}\{R_c(x, y) - 1\}$ that are separated from one another by a distance $\frac{1}{\Delta x}$ on the frequency domain, which translates as a distance of $\frac{\lambda f}{\Delta x}$ in the output plane of the 4f system. This array of copies of $\mathcal{F}\{R_c(x, y) - 1\}$ is then multiplied by $\mathcal{F}\{s(x, y)\}$, which translates into an attenuation profile. In addition to these attenuated copies of $\mathcal{F}\{R_c(x, y) - 1\}$, there is a delta function at the center of the plane (caused by the specular reflection from the substrate area), and everything is blurred by the Fourier transform of the rect function that represented the clear aperture of the metasurface's substrate.

When the phase profile is optimized to contain the convolution kernels of the digital convolutional layer, those kernels are contained in the copies of $\mathcal{F}\{R_c(x, y) - 1\}$. However, they are affected by the attenuation profile described by the product with $\mathcal{F}\{s(x, y)\}$. As such, this attenuation needs to be taken into account during the phase optimization process; otherwise we would just be making $|\mathcal{F}\{R_c(x, y)\}|^2$ approach the target PSF, without accounting for the attenuation present in the region of interest in $|\mathcal{F}\{R(x, y)\}|^2$.

The pixel shape function can be described with the following expression:

$$s(x, y) = \left[\text{rect}\left(\frac{x}{l_x}, \frac{y}{l_y}\right) * \text{comb}\left(\frac{x}{P_x}, \frac{y}{P_y}\right)\right] \text{rect}\left(\frac{x}{\Delta x}, \frac{y}{\Delta y}\right), \tag{S4}$$

which means that there are nano-tokens with rectangular cross-sections with dimensions of l_x and l_y, which are separated by a distance P_x along the x direction and a distance P_y along the y direction, contained within a region that has the transverse dimensions of the phase pixels, Δx and Δy. P_x and P_y are the transverse dimensions of the metasurface's unit cell, and are both equal to 350 nm in our case. The Fourier transform of this pixel shape function is given by:

$$\begin{aligned}\mathcal{F}\{s(x, y)\} = &\left[\text{sinc}\left(l_x f_x, l_f f_y\right) \text{comb}\left(P_x f_x, P_y f_y\right)\right] \\ &* \left[\Delta x \Delta y \,\text{sinc}\left(\Delta x f_x, \Delta y f_y\right)\right],\end{aligned} \tag{S5}$$

which consists of the two-dimensional sinc function $\left[\Delta x \Delta y \,\text{sinc}\left(\Delta x f_x, \Delta y f_y\right)\right]$ being convolved with an array of attenuated delta functions that are separated by distances $\frac{1}{P_x}$ and $\frac{1}{P_y}$ along the cardinal directions of the frequency domain. However, since these distances are larger than the bandwidth of the imaging system (as the metasurface's unit cell size is smaller than the diffraction limit), the only relevant portion of this convolution result is the convolution between the sinc function and the central delta of the comb array; thus, we can write:

$$\mathcal{F}\{s(x, y)\} = \Delta x \Delta y \,\text{sinc}\left(\Delta x f_x, \Delta y f_y\right). \tag{S6}$$

If we take a look at Fig. S2 a), we can observe that the attenuation profile present on the metasurface's PSF can indeed be represented by the sinc function from Eq. (S6). Notice that the center of this sinc pattern doesn't coincide with the centers of the copies of $\mathcal{F}\{R_c(x, y) - 1\}$.

As stated above, the phase optimization process needs to account for this attenuation profile, whose effect is causing the convolution kernels that are placed further away from the center of metasurface's PSF to become dimmer. As such, computing $\left|\mathcal{F}^{-1}\left\{e^{i\Phi}\right\}\right|^2$ for the numerical array containing the phase profile Φ, should yield a result that contains blur kernels that, after being affected by the attenuation profile, can resemble the convolution kernels of the original digital convolutional layer. In other words, if ATT is a numerical array that encodes this attenuation profile, then the goal of the phase optimization process is to find the numerical array Φ such that $\text{ATT}\left|\mathcal{F}^{-1}\left\{e^{i\Phi}\right\}\right|^2$ approaches the numerical array that contains the convolution kernels of the digital layer (which we have referred to as the target PSF). Thus, the phase optimization process consists on a gradient descent algorithm that finds the numerical array Φ that minimizes the following loss function:

$$L(\Phi;\, \text{PSF}_{\text{target}}) = ||\,(\text{PSF}_{\text{target}}/\text{ATT}) - |\mathcal{F}^{-1}\left\{e^{i\Phi}\right\}|^2\,||_F^2, \tag{S7}$$

where $||\cdot||_F$ denotes the Frobenius norm. The reason why only the phase is optimized for in Eq. (S7) (and not the magnitude) is because we want a phase profile that yields the target PSF while the phase mask has unity (or at least constant) magnitude modulation; this is a choice that is done both to have the phase values be the only degrees of freedom and to attain a system with as few energy losses as possible.

5. NANO-TOKEN GEOMETRICAL PARAMETERS

We used a library containing the magnitude and phase modulation of the field reflected by nano-tokens with different transverse dimensions that was built through simulations. The transverse dimensions of the silver nano-tokens in this library varied from 30 nm to 300 nm in intervals of 10 nm. The nano-tokens have a fixed height of 30 nm, and rest on top of a magnesium fluoride film that is 75 nm thick, which is itself on top of a silver film that is 130 nm thick, and the unit cell used in the simulations was 350 nm in both transversal dimensions.

In order to get the metasurface layout that implements the optimized phase profile (obtained following the process described at the end of last subsection), we first need to take m regularly-spaced discrete values within the 0 to 2π interval, so that we can have a set of m distinct nano-token geometries that can impart each of them. With the library we used, it was possible to use $m = 8$, so we had to take all the phase values in both the optimized phase profile and the phase modulation mapping of the library to their closest integer multiple of $2\pi/8$.

After digitizing the phase modulation values in the nano-token library, we chose the nano-token geometries that yielded each of the m different phase modulation values while having the largest magnitude modulation and transverse dimensions values available. The nano-token geometries we used for each phase value are shown in Table S1, along with the resulting magnitude modulation associated with each geometry. Finally, this phase-to-geometry mapping was used to produce the design file containing the metasurface layout that would be used for fabrication.

6. DESIGN PIPELINE

The design pipeline consists on a sequence of Steps in which the parameters of the optical system and the digital layers are adjusted in order to optimize the system's performance on an image classification task. This task consists on classifying images from the CIFAR-10 dataset, created by the Canadian Institute For Advanced Research (from which it gets its name) [2]. The dataset is comprised by 60000 32×32 images that can belong to one of ten possible classes along with a label that indicates what class each image belongs to; the dataset is split into a training set with 50000 images and a test set with 10000 images (and their respective labels).

The Steps of the training pipeline are as follows, and they are also illustrated in a diagram found in Fig. S3:

Step 1: Creating and Characterizing the Original Digital Neural Network First, a fully-digital neural network is built and its parameters are adjusted using the dataset's training set in order to perform the image classification task. After the training process is completed, an inference process is ran using the dataset's test set in order to measure the network's classification accuracy when presented with inputs it hadn't seen during the training process.

Phase modulation (rad)	x-width (nm)	y-width (nm)	Magnitude modulation
$\frac{2\pi}{8} \times 0$	100	70	0.94
$\frac{2\pi}{8} \times 1$	300	80	0.87
$\frac{2\pi}{8} \times 2$	290	90	0.82
$\frac{2\pi}{8} \times 3$	280	100	0.81
$\frac{2\pi}{8} \times 4$	270	120	0.87
$\frac{2\pi}{8} \times 5$	240	290	0.93
$\frac{2\pi}{8} \times 6$	90	280	0.94
$\frac{2\pi}{8} \times 7$	80	50	0.97

Table S1. Geometrical parameters of the nano-tokens used to impart each of the 8 possible phase modulation values, along with the magnitude modulation value associated with each nano-token geometry.

Step 2: Creating the Target PSF and Simulating the Optical Convolution Block The convolution kernels of the trained digital network's first layer are saved, and used by a Phase Optimization algorithm whose task is to find the phase modulation profile that needs to be imparted by the metasurface so that the saved convolution kernels can be present in the optical system's PSF. More technical details about this Phase Optimization algorithm can be found in the Supplementary Information document. After the phase modulation profile has been obtained, it is used to simulate the optical system's PSF, and from there, new convolution kernels for the first layer are obtained, which are slightly different from the ones obtained on Step 1. The effect that this difference has on the network's classification performance is quantified by first replacing the old first layer kernels with the new ones and then running an inference process using the test set and measuring the new classification accuracy.

Step 3: Fine-tuning the Suffix Layers The values of the network's first layer's kernels are frozen to be the ones obtained on Step 2, and the parameters of the suffix layers (i.e. the rest of the network) are fine-tuned in order to compensate for the differences that were introduced during Step 2.

Step 4: Co-Training the Phase Profile and the Network A training process is ran where all of the network's layers' parameters are fine-tuned along with the values in the phase modulation profile.

Step 5: Running Inference with the Optical Convolution Block The final values of the phase modulation profile obtained at the end of Step 4 are used to fabricate the metasurface, which is then mounted on the optical convolution block. Pictures from the test set are projected into the optical system using a display, and the yielded output images are captured by a camera and saved into a computer. These camera captures are then post-processed to obtain a set of numerical arrays that resemble the outputs that would have been yielded by the original digital convolutional layer. This set of output numerical arrays are then fed into the suffix layers, whose parameters have the final values from the end of Step 4, via an inference process. More details on this experiment can be found in the Methods section.

Step 6: Performing a Final Fine-tuning of the Suffix Layers Since the outputs of the original digital convolution layer and the optical convolution block are not exactly equal, a final fine-tuning step must be done so that the suffix layers can now account for those differences and have a good performance when they receive inputs that come from the latter instead of from the former. This is done by running the training set through the optical convolution block and the post-processing on the captures, and using the resulting outputs as a training dataset for the digital suffix layers. After the fine-tuning of the suffix layers is done, a final inference process is performed to measure the classification accuracy performance of the joint hybrid system, running the test set through the optical convolution block and the suffix layers.

The performance of the system is measured at the end of each Step of the pipeline by performing an inference process, where the system receives pictures from the test set as inputs and then

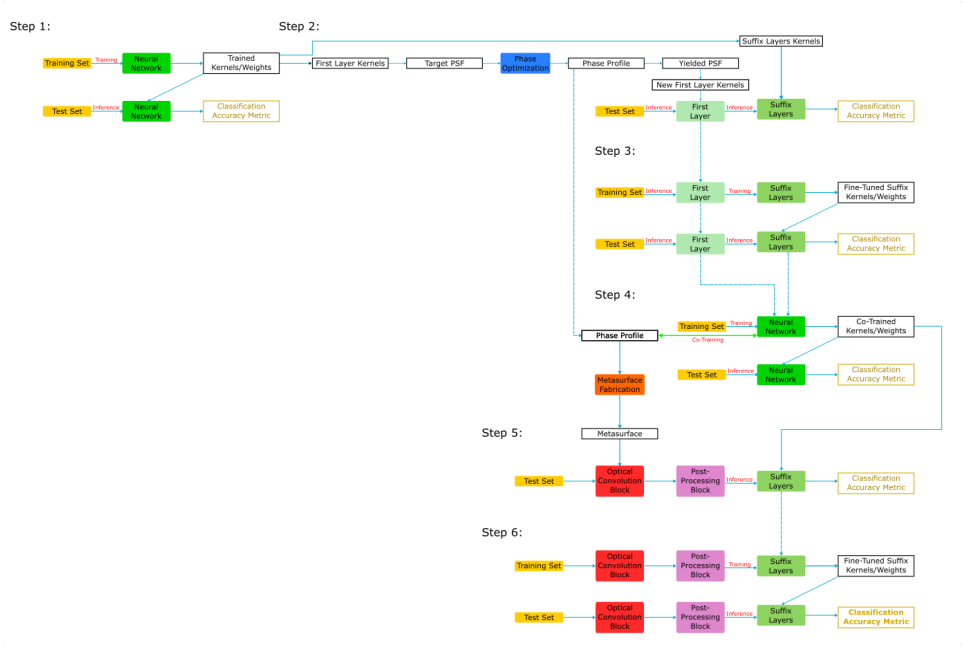

Fig. S3. Design pipeline. Training processes consist on adjusting the system's parameters as it is presented with various inputs. Inferences processes consist on measuring the system's classification accuracy using the current values its parameters have. The pipeline consists on six Steps, where the parameters of the digital network layers and the optical convolution block are adjusted to optimize the classification accuracy performance of the hybrid system where the digital network's first layer is replaced by the optical convolution block.

Step	Accuracy	Log-Loss
Step 1	80%	0.70
Step 2	78%	0.79
Step 3	83%	0.64
Step 4	86%	0.50
Step 5	11%	4.48
Step 6	65%	1.09

Table S2. Results of the system's classification performance on the test set at the end of every Step in the pipeline, measured with the Accuracy and Log-Loss metrics.

outputs a prediction about what classes the pictures belong to. Two metrics are used to measure the classification performance: accuracy and the cross-entropy loss function (also known as log-loss). The former measures how frequently the system's predictions are correct (matching the ground truth), while the latter measures the "confidence" the system has in assigning the corresponding ground truth label to each picture, as the value of the log-loss function is lower if the predicted probabilities assigned to the ground truth label of the pictures are higher [3]. The results for the values of these metrics at the end of each Step of the pipeline are shown in Table S2.

In addition to this, a fully digital network was constructed to have the convolution kernels of the first layer set to the values obtained at the end of Step 4, and the parameters of the suffix layers set to the values from the end of Step 6. The resulting network is the benchmark fully-digital network reported in the Main Document, which has a performance with an accuracy of 66% and a log-loss of 1.11. The performance of the hybrid optical system is shown in the last row of Table S2, and it is very similar to that of the benchmark network.

REFERENCES

1. J. W. Goodman, *Introduction to Fourier optics* (Roberts and Company Publishers, 2005), 3rd ed.
2. A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. rep., Department of Computer Science, University of Toronto (2009). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.
3. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016). http://www.deeplearningbook.org.