



OPEN

Meshless optical mode solving using scalable deep deconvolutional neural network

G. Alagappan✉ & C. E. Png

Optical mode solving is of paramount importance in photonic design and discovery. In this paper we propose a deep deconvolutional neural network architecture for a meshless, and resolution scalable optical mode calculations. The solution is arbitrary in wavelengths and applicable for a wide range of photonic materials and dimensions. The deconvolutional model consists of two stages: the first stage projects the photonic geometrical parameters to a vector in a higher dimensional space, and the second stage deconvolves the vector into a mode image with the help of scaling blocks. Scaling block can be added or subtracted as per desired resolution in the final mode image, and it can be effectively trained using a transfer learning approach. Being a deep learning model, it is light, portable, and capable of rapidly disseminating edge computing ready solutions. Without the loss of generality, we illustrate the method for an optical channel waveguide, and readily generalizable for wide range photonic components including photonic crystals, optical cavities and metasurfaces.

Machine learning^{1,2} techniques are widely used in image classification³, natural language processing⁴, automatic speech recognition⁵ and robotics⁶. Apart from these traditional applications, machine learning methods have recently penetrated vast areas of science and engineering fields which are conventionally explored by deterministic hard computing methods. In photonics, machine learning techniques have been used in improving the state of the art in optical fiber sensing, laser characterization, quantum communications, optical imaging, photolithography, inverse designs of photonic devices with targeted performances, and on-demand designs of metamaterials⁷⁻¹⁴. In this article we tackle the optical mode solving problem for the general parameter space of photonics using deep learning methods.

Optical mode solving is of paramount importance in photonic design and discovery. Traditionally, this is handled by solving time independent Maxwell equations¹⁵⁻¹⁷. It is usually the first step in any photonic device design, and engineers usually spent plenty of their design time in this step extracting and optimizing mode distributions, field confinements, coupling coefficients, effective and group refractive indices. Optical modes of a particular photonic geometry also serve as basis functions in the analysis of perturbative effects such as sensing, defect in nano-cavities, waveguide tapers and waveguide gratings. Therefore, accurate and rapid mode-solving are extremely necessary to have an accelerated photonic model. Apart from the speed, the mode solution also must be memory wise light and portable so that they can be easily deployed as edge or mobile computing solutions¹⁸⁻²⁰. This will greatly benefit many photonic systems such as optical measurement devices, lidars, photonic integrated circuits, and fiber-based communications. Edge computing aims to deliver computing paradigm closer to the source of data. To appreciate this, consider the fabrication process of a photonic waveguide, as an example. Once the waveguide is fabricated, the optical metrological devices will measure the geometrical parameters. The edge computing if available at the measurement device, it will enable instantaneous prediction of the mode profiles at the point of geometrical measurements.

Analytical solutions for optical modes exist for simple one-dimensional slab waveguides. For two-dimensional geometries such as channel, and strip waveguides, photonic crystals, and metasurfaces numerical simulations must be sought for accurate solutions. Usually, to tackle the numerical problem, one applies mesh-based methods such as finite difference, and finite element¹⁵⁻¹⁷. These methods discretize the entire computational geometry into finite number of meshes. The number of meshes increases with the size of the geometry. Alternatively, mesh free methods such as spectral^{21,22} and finite cloud²³ methods are available. In spectral method, the computational geometry is divided into a set of structurally invariant domains. The fields inside the domain are expanded using a pre-selected set of basis functions. In the finite cloud method, instead of domains, clouds of nodes are distributed throughout the geometry. The density of the nodes can be controlled as per anticipated field gradients. In

Institute of High-Performance Computing, Agency for Science, Technology, and Research (A-STAR), Fusionopolis, 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Singapore. ✉email: gandhi@ihpc.a-star.edu.sg

all these methods (with or without mesh), one finally encounters matrix diagonalization to solve for unknown coefficients and fields. The size of the matrix directly scales with the number of meshes (in mesh-based method) and number of domains/nodes in the abovementioned mesh-free methods. Although, matrix diagonalization methods are well established, they consume a significant amount of computational resources, and this hinders design acceleration when performing a large number of geometrical sweeps, and optimizations.

The photonic parameter space is finite and well defined. We know the wavelengths, the exact materials, and the range for the fabrication feasible geometrical dimensions of the devices. Researchers often explore the well-defined parameter space repeatedly with brute force numerical methods. In the current perspective of machine learning era, such repeated exploration is thus, an inefficient use of computational resources. Any finite parameter space can be captured, and an effective representation can be formulated. Such representation often assumes deep learning models and have been proven to be a solid replacement for the traditional photonic models^{24–33}.

In this article, we show that an effective deep learning representation for the optical mode solving can be built for the usual parameter space of photonics³⁴. The field profile of an optical channel waveguide is solved using a deep deconvolutional neural network (DCNN). We illustrate a meshless method, with full resolution scalability that is generic in photonic wavelengths, materials, and geometrical dimensions. The proposed method is readily generalizable for finding mode images in photonic crystals, optical cavities, nanostructures and metasurfaces. It shares ubiquitous benefits of deep learning model of being light, portable, fast and edge computing ready solutions. Furthermore, it can provide an excellent set of basis functions for ultraquick evaluations of advance numerical algorithms that tackles perturbative problems. One such example includes the eigenmode expansion method³⁵, which is often used to design waveguide tapers and waveguide Bragg gratings. Here, the waveguide modes are used as basis function to evaluate the effect of perturbations (i.e., grating or taper profiles). The basis modes traditionally are evaluated using the time-consuming numerical methods. This is clearly a bottleneck in the entire eigenmode expansion algorithm and can be easily fixed by replacing the traditional solver with the deep learning-based mode solvers.

Figure 1a shows the schematic of the optical channel waveguide. The waveguide has a rectangular geometry with width, w and height, h . We consider a fabrication feasible photonic parameter space for w and h . The spans of w and h are $0.15 \mu\text{m} < w < 1.1 \mu\text{m}$, and $0.1 \mu\text{m} < h < 0.5 \mu\text{m}$. Silica is used as a cladding material as it the popular choice for a wide range of applications. The core refractive index is taken as n , and the span $1.6 < n < 4$ covers refractive indices of most photonic materials. The prediction domain has a square shape with sides of $L = 1.5 \mu\text{m}$. The model is designed to predict the magnetic field pattern of the fundamental waveguide mode the with transverse electric (TE) polarization. Fundamental TE modes are used for most of the on-chip applications.

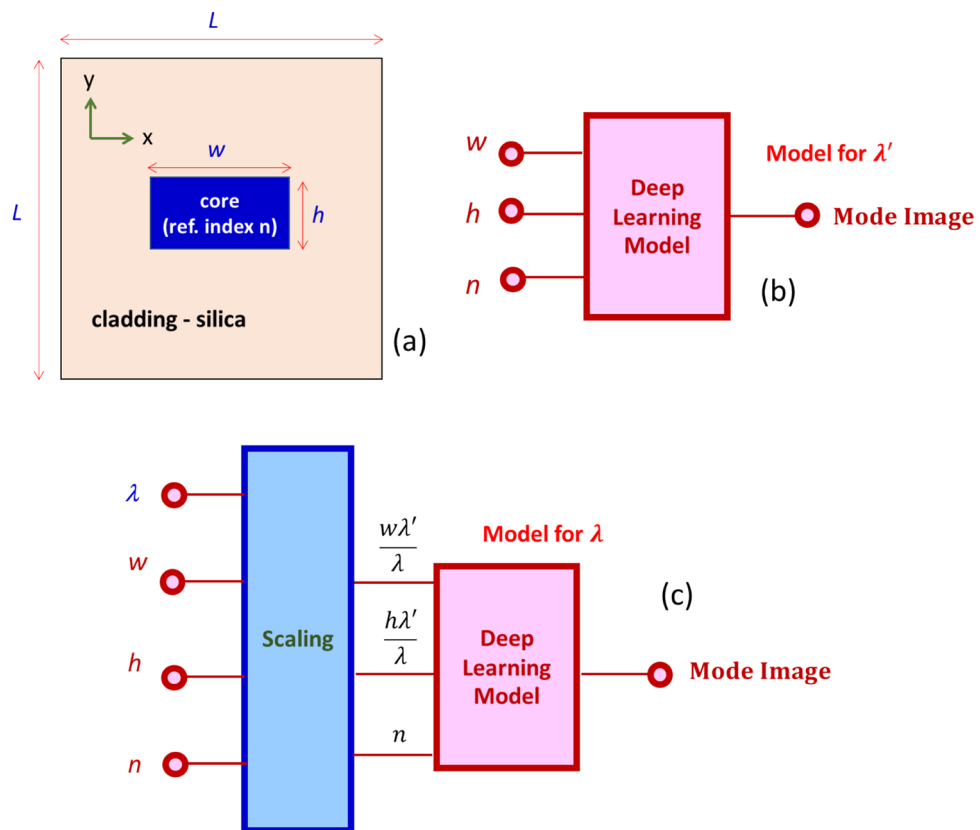


Figure 1. (a) Schematic of the optical channel waveguide. (b) The deep learning model trained at $\lambda' = 1.55 \mu\text{m}$. (c) The wavelength generic deep learning model.

A total of 3000 data points is prepared using commercially available Lumerical MODE software³⁶. To save the computational time in Lumerical MODE software, we applied anti-symmetric and symmetric boundary conditions along x and y directions (see axes definition in Fig. 1a), respectively. For details on the symmetric and anti-symmetric boundary conditions, please see Ref.³⁷.

In Lumerical MODE simulation, the size of the computation domain is not fixed to $L = 1.5 \mu\text{m}$. The computational domain for numerical calculation is kept sufficiently large to ensure proper decay of the evanescent field. The computational window for low refractive index contrast waveguides (i.e., longer evanescent tail) are much larger than computational window for high refractive index contrast waveguides (i.e., shorter evanescent tail). However, during the data preparation stage the mode images are truncated with $L = 1.5 \mu\text{m}$. This will help to standardize the size of the prediction window, and the output mode images. The value of $L = 1.5 \mu\text{m}$ is conveniently picked such that it is larger than the maximums of w and h in the input parameter space. All data points are checked for their convergence and existence of a sensible mode. The associated effective refractive indices of all mode profiles in the data set are also verified independently with our earlier high precision deep learning models that predicts the effective refractive indices³². Therefore, the DCNN is trained with only convergent solutions of Maxwell equations within the specified parameter space. As a result, it will only output convergent solutions for inputs within the boundaries of the input parameter space.

Firstly, we built our deep learning model at the telecommunication wavelength (λ') of $1.55 \mu\text{m}$. The block diagram is shown in Fig. 1b. The input is a three-dimensional vector comprising w , h and n , and the output is an image (a matrix) representing the distribution of the field value in the square domain of size L , with a resolution of N by N pixels. The model built for wavelength λ' can be used for a generic wavelength λ by having a scaling layer as shown in Fig. 1c. Here we apply the scaling property of Maxwell's equation. There is no fundamental length scale in Maxwell's equations which fortuitously allow devices to be modelled at different wavelengths by simply scaling the geometrical dimensions in proportion to the change in wavelengths. For an example, if you would like to find mode image at $w = w_0$, $h = h_0$ for a wavelength λ_0 using the model developed for λ' , then the geometrical inputs to the λ' model must be scaled by quantity λ'/λ_0 . The model presented in Fig. 1c is able to handle the refractive index dispersion of the core material. Varying refractive index is now can be set for different wavelengths [i.e., $n(\lambda)$].

In Fig. 2, we reveal the architecture of the deep learning model embedded in the block of Fig. 1b. As the input is a vector and the output is an image, the architecture must be able transform the vector to a matrix. We accomplish this by mean of a DCNN. The proposed DCNN has two stages performing projection and resolution scaling. The first stage projects the three-component vector into a higher dimensional space, resulting in a vector of D^2 components. This stage comprises fully connected feedforward layers with rectified linear unit (ReLU) activations^{38,39}. In training deep neural networks, vanishing gradient is a common issue. ReLU activation helps circumventing this problem. The expression for ReLU is $f(x) = \max(0, x)$. It is a piecewise linear function that

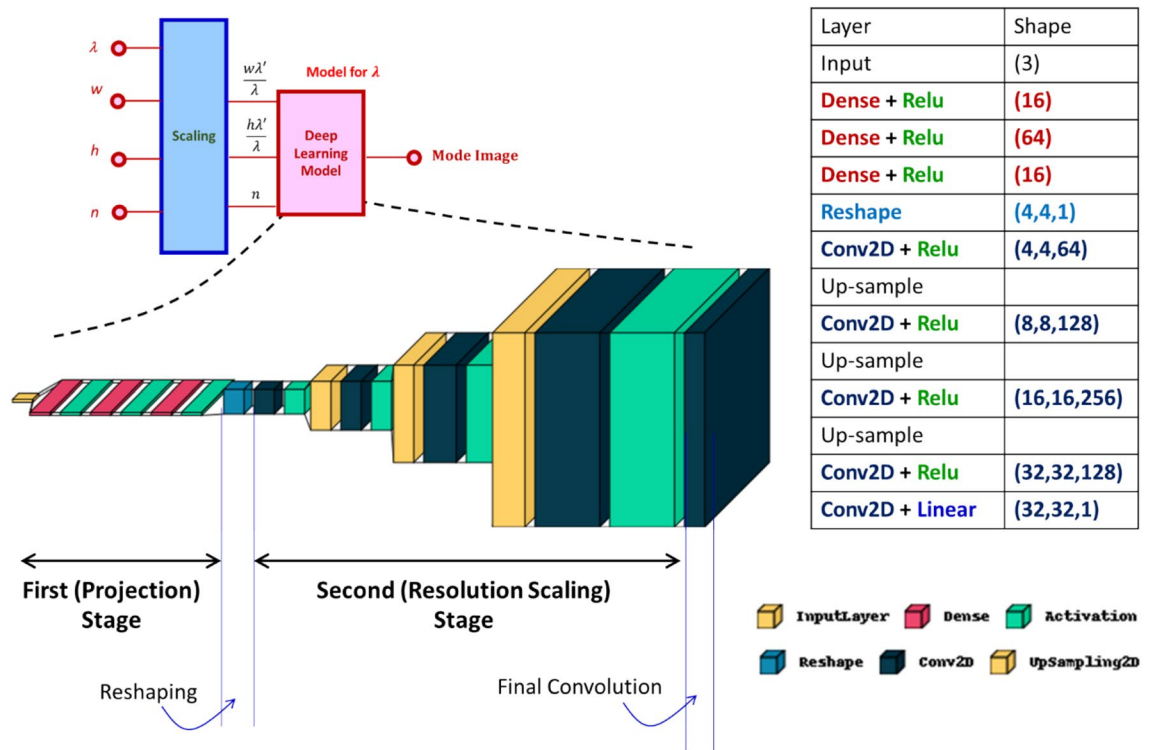


Figure 2. The DCNN architecture. It has two stages, projection, and resolution scaling. The key on the right bottom, labels the layers of different colors. The table in the figure elaborates the details of the layers and their shapes.

outputs the input directly if it is positive, otherwise, outputs zero. ReLU is the most common activation in CNN and DCNN since Alexnet discovered in 2012⁴⁰.

We used $D=4$. This means the 3-component vector is transformed into a 16 (D^2) component vector at the end of stage one. Before entering the second stage, the vector with D^2 components is reshaped to a $D \times D$ matrix. The second stage scales the resolution to the required $N \times N$ pixels, and it has multiple blocks of up-sampling and convolution layers with varying filter (channel) size. The kernel size of the sliding convolution window is 3 by 3. At the end of the second stage, we have a final convolution layer with linear activation. It collapses all the filters in the earlier layer and produces the final image. The trainable parameters consist of neurons (first stage) and filters (second stage). As higher resolution is handled using an additional resolution-scalable block (convolution block) in the second stage, the higher resolution does not constitute to more neurons. However, the extra convolution block increases the total numbers of filters that to be trained. The extra learning is addressed effectively via a transfer learning approach and will be elaborated later in this article.

In Figs. 3 and 4, we present the glimpses of DCNN ability of producing high quality mode images and cutlines for waveguides constructed with various core materials. The first row in Fig. 3 demonstrates exact and predicted mode images of gallium arsenide (GaAs) waveguide with an aspect ratio of more than 1 [$w/h > 1$]. Here the mode is tightly confined to the core of the waveguide. The second row in Fig. 3 showcases a square diamond waveguide, and the third row displays similar mode patterns for a silicon-oxynitride waveguide with a weak mode confinement. As can be visually seen, the predicted and exact mode patterns are in very good agreement. The third column in Fig. 3 displays the absolute error, $|H_{\text{exact}} - H_{\text{predict}}|$ with H_{exact} and H_{predict} being the exact and predicted values of the magnetic field at each pixel.

Note the depicted mode profiles in Fig. 3 will also apply for a lot of other photonic materials with similar refractive indices such as silicon, titanium oxide and polymers. The exact mode image is obtained via numerical solution to Maxwell's equation using the method finite differences. We used the commercially available Lumerical MODE solver. The deep learning predicted image comes from the DCNN model illustrated in Fig. 2. The fine detail of the architecture is elaborated in the insert of the Fig. 2. Both H_{exact} and H_{predict} are normalized such that the maximum value is 1. Figure 4 shows the field values for $y=0$ (i.e., along the horizontal cutlines in Fig. 3) for various waveguides. The example comprises various cases such as different aspect ratio, weak and tight confinements. Again, we can readily verify from Fig. 4, both predicted and exact mode profiles are in very good agreement for various waveguide geometries and material systems.

Figure 5 portrays the resolution scaling mechanism in DCNN. This is done with a series of scaling blocks. Each scaling block consist of up-sampling, two-dimensional convolution and ReLU activation layers. For a final image with $N \times N$ pixels, a total of $1 + \log_2(N/D)$ scaling blocks is required. For the first scaling block immediately after the first stage, the up-sampling layer is substituted by a reshaping layer. The up-sampling layer increases the dimensionality of the matrix, while the convolution-activation combo learns the fine features of the mode distribution. In the left panel of Fig. 5 we showcase the field patterns of the fundamental mode silicon waveguide ($w = 0.55 \mu\text{m}$, $h = 0.25 \mu\text{m}$, $n = 3.5$), generated by DCNN architectures of varying N . The number of filters (n_{filters})

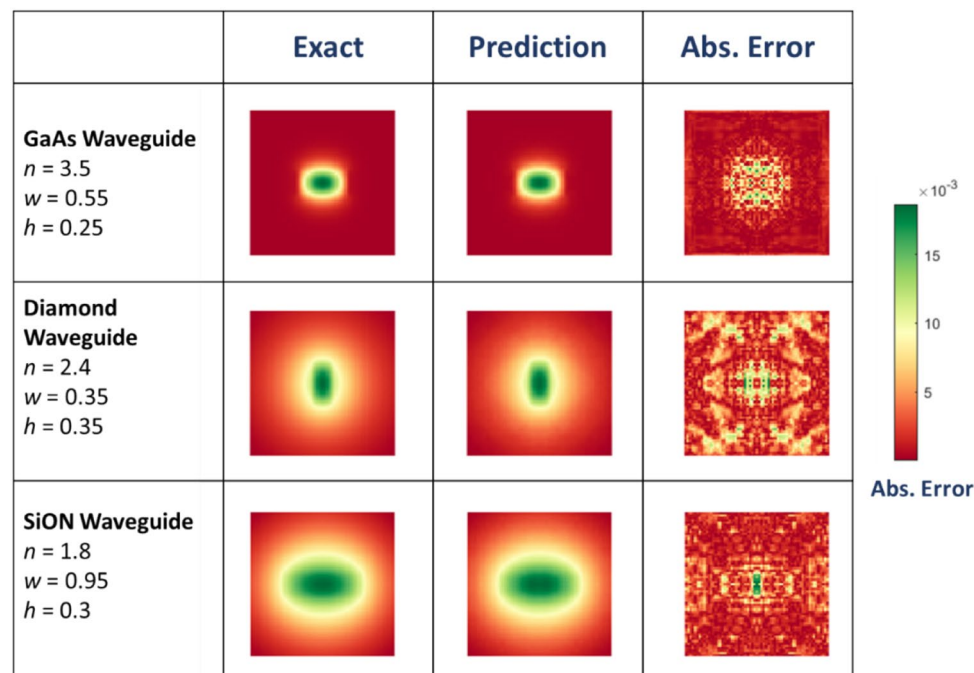


Figure 3. The glimpses of DCNN ability in predicting the mode images of various photonic waveguides. The second column and the third column showcase the exact and predicted (with DCNN architecture in Fig. 2) mode patterns respectively. The third column plots the absolute error at every pixel.

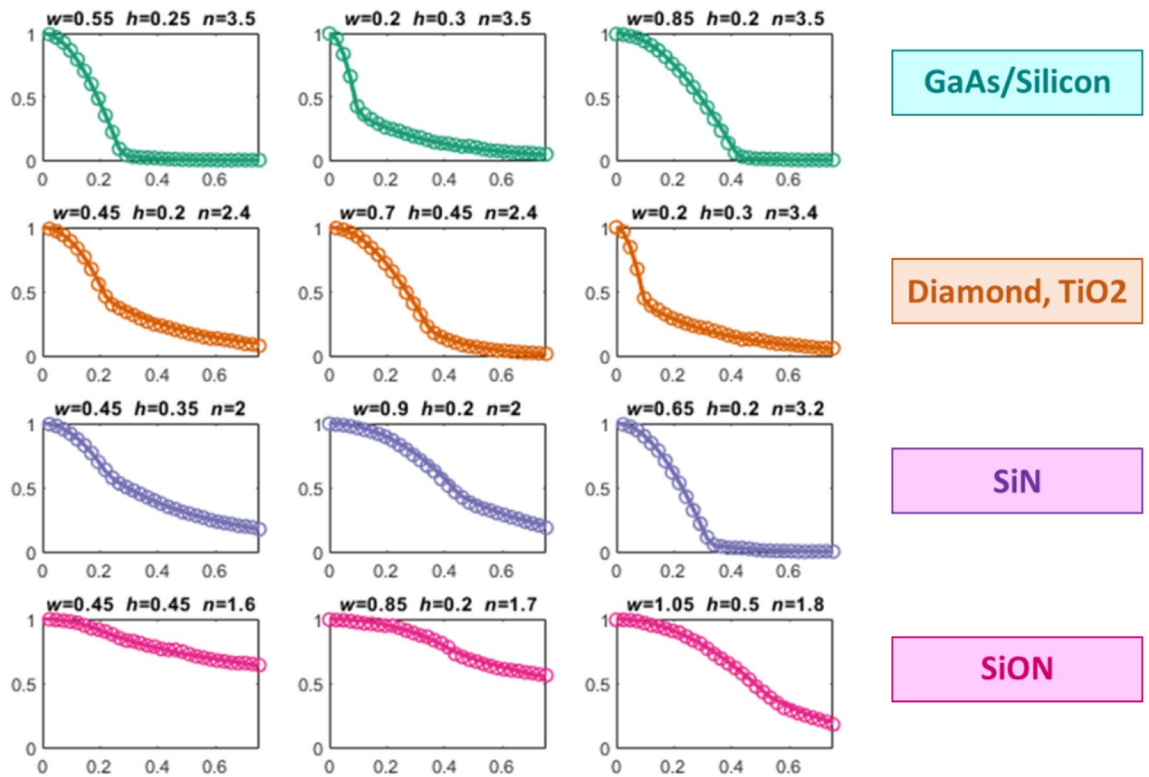


Figure 4. The horizontal cutlines of various photonic channel waveguides. Solid lines—exact numerical computation. Circles—prediction with DCNN architecture in Fig. 2.

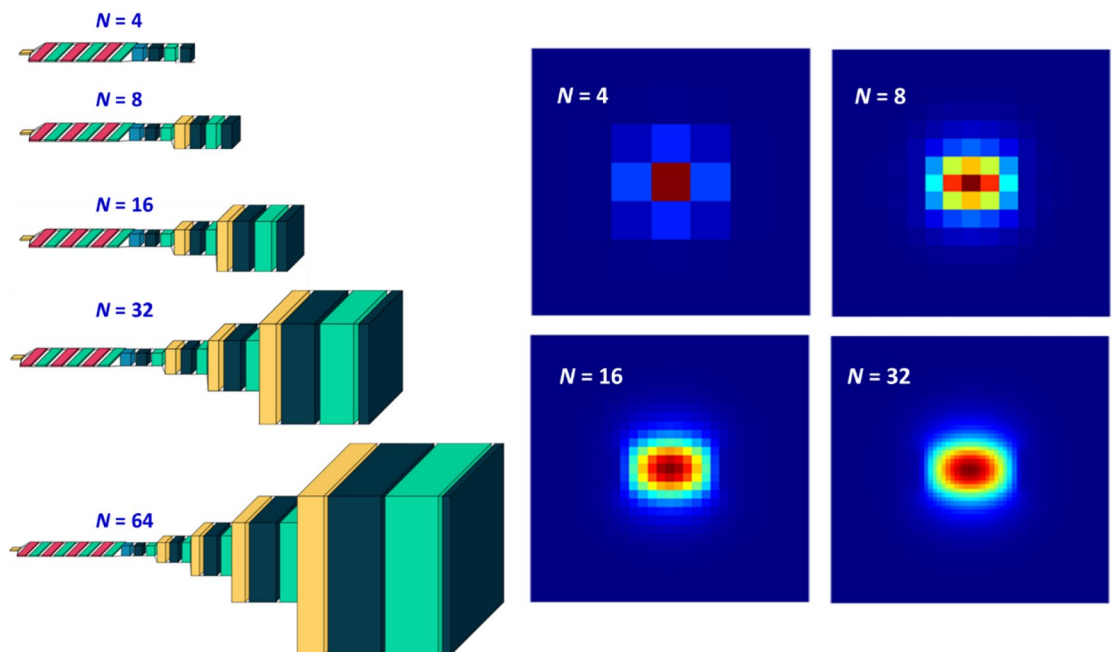


Figure 5. Scalable resolutions via DCNN. (a) DCNN for various resolutions. (b) The electric field pattern of the fundamental waveguide mode in a silicon waveguide with $w=0.55 \mu\text{m}$, $h=0.25 \mu\text{m}$, $n=3.5$.

in each scaling block is an hyperparameter, and the optimization is carried out by means of a transfer learning approach. We start with training architectures of smaller N . The trained weights are transferred [i.e., copied and kept frozen] when training the architecture with a larger N . In the architecture with larger N , only the newly inserted scaling block is trained, and in this manner both precision and training time can be optimized.

The training of DCNN is done using a root mean squared backpropagation algorithm with a learning rate of 0.001³⁸. The data set is split into training, validation and testing sets in a ratio of 0.7, 0.2 and 0.1, respectively³⁹. In deep learning framework bulk portion of the data will be used for model building (i.e., training—determination of weights). The major fraction of the remaining data will be used for validation. The validation data points are not used in training, but they are used to monitor the validation loss which signals overfitting of the deep learning model. Finally, a small set of data of will be reserved for testing. The testing data set is kept totally aside from training and validation, and it is used to independently measure and compare the performance of various hyperparameters. As a rectangular symmetry prevails in all mode images, in the training, we used only the quarter of the actual image where there is no symmetry present. We begin training with an architecture of $D=N=4$. For the feedforward connections, we used three hidden layers and considered four sets of neuron distributions. As $D=4$, the number of neurons in the last hidden layer is kept at $D^2=16$. The first set has a neuron distribution of $16 \times 16 \times 16$ [i.e., 16 neurons in each layer]. The second and third sets have distributions of $16 \times 32 \times 16$ and $16 \times 64 \times 16$. The last set has a progressive distribution of $4 \times 8 \times 16$. Figure 6a shows the mean square error in the test data set as a function number of filters in the convolutional layers. From Fig. 6a, we can see that the optimized architecture has a feedforward connection with $16 \times 64 \times 16$ neurons and $n_{filters}=64$. The progressive distribution $[4 \times 8 \times 16]$ performs poorer than the other distributions.

For training architecture with a higher resolution $N=8$, we transfer the learning from the optimized architecture of $N=4$. The architecture for $N=8$ replicates the optimized architecture of $D=N=4$ with copied pre-trained weights and, an extra scaling block at the end for resolution increment [see Fig. 5]. The training and optimization of $N=8$ architecture is done only for this extra scaling block. Figure 6b shows the mean square error as a function $n_{filters}$ of the last scaling block. As can be readily seen, the optimal value for number of filters in the extra scaling block is 128. In the similar manner the optimizations for $N=16$, and 32 architectures are carried out by mean of the described transfer learning and the results are shown in blue curves of Fig. 6c,d. The red curves represent the fine-tuning case when the training of inner scaling blocks are allowed. As we can see Fig. 6d, the final fine-tuned model reaches a mean squared error of 3.7×10^{-5} . This, error translates to precision 2 to 3 decimal places in absolute value. Such error is small²⁷, and visually leaves no discrepancy between predicted and exact mode patterns (see Fig. 3).

In order to assess the resource efficiency of the presented model, we quantified the computational efforts taken by the DCNN model and compared it with the traditional finite difference method. The DCNN model is written

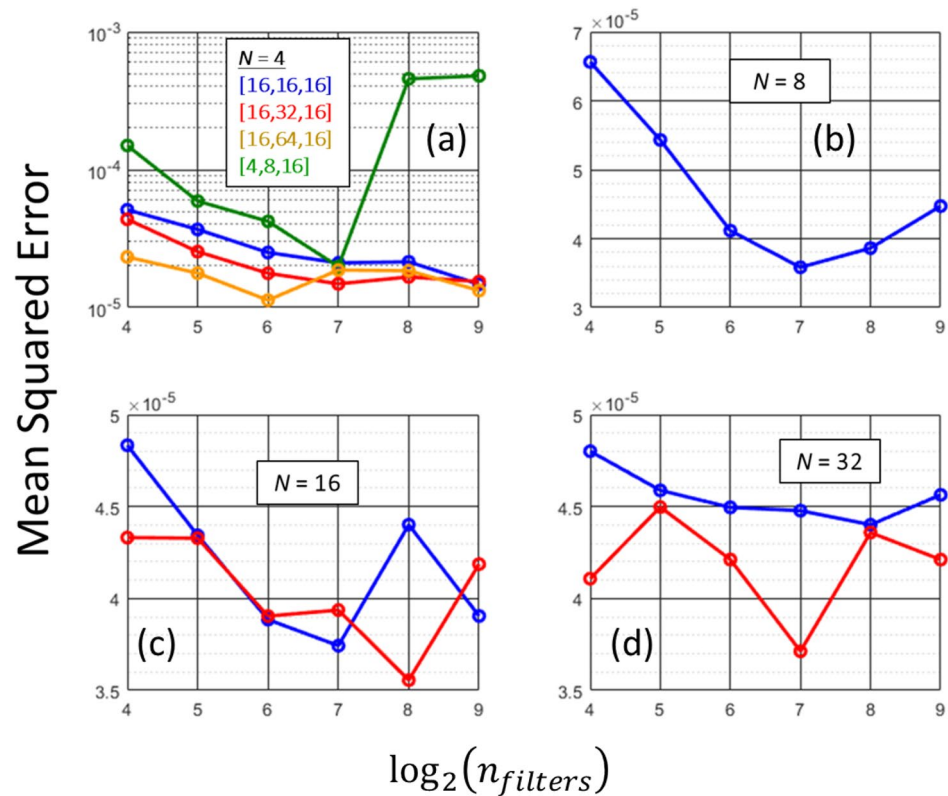


Figure 6. Optimization of number of filters ($n_{filters}$) in DCNN architectures of varying resolutions, N . **(a)** Case $N=4$. Mean square error as a function $n_{filters}$ for various configurations of feedforward connection. **(b)** Case $N=8$. Mean square error as a function $n_{filters}$. **(c)** Case $N=16$. Mean square error as a function $n_{filters}$. **(d)** Case $N=32$. Mean square error as a function $n_{filters}$. In **(b–d)**, the blue curves represent mean square errors for the optimization of final scaling block with frozen inner block weights. The red curves represent learning with fine tuning, where all the scaling blocks participates in the training.

in python, and thus for a fair assessment we made a comparison with a finite difference optical mode solving code written in the same programming language⁴¹. We performed DCNN and finite difference simulations for all geometries in the testing data set. Testing data set consists of 332 geometries (i.e., combinations of w , h and n) randomly sampled in the input parameter space. The resolution in finite difference simulation is kept the same as in the DCNN model [$N=32$, $n_{\text{filters}}=128$]. The computational domain in finite difference is taken as S_w and S_h in x and y directions, respectively. Here, S is a scaling factor and S should be sufficiently large for proper convergence. In DCNN model, as we elucidated earlier, during the data preparation stage the computational domain was kept sufficiently large, and every data point is ensured its convergence with a sensible mode. Using ASUS laptop with intel i7 CPU [1.8 GHz, 4 cores 8 GB RAM], the DCNN model took ~ 14 s to predict the mode profiles of all the 332 test geometries. In the finite difference model, the model consumed ~ 76 , ~ 140 , and 230 s for the cases of $S=3$, 4 and 5 , respectively. DCNN calculations are therefore 5-, 10-, and 16-fold faster than the finite difference simulations with $S=3$, 4 and 5 , respectively. These results portray clear advantages of DCNN in terms of the computational speed.

Throughout this article we used rectangular channel waveguide with isotropic core material as an example. However, the presented meshless, resolution scalable mode solving method is universal, and can be adopted for many photonic devices by modifying the input vector. In the case of directional couplers (i.e., two closely spaced parallel waveguides), apart from the waveguide parameters (w , h and n) the spacing between the two waveguides must be supplied in the input vector. For the case of anisotropic materials, the orientation of the principal axis must be supplied, and single refractive index used for isotropic materials must be substituted with principle refractive indices. For photonic crystals, the input vector must consist of the parameters that describes the geometry of the unit cell and the period of the lattice. In all these devices, architectures of projection stage and resolution scaling stage will remain the same. The number of neurons, feedforward layers, and convolutional filters can be tuned with the help of the presented transfer learning approach.

Simple shapes like circles, squares, and rectangles with perfect geometries are often the best starting point in most photonic device designs. However fabricated structures will have some irregularities. These irregularities can be captured, parameterized, and included in the input vector for optical mode solving. For devices with totally irregular shapes, a different approach is required. One can use autoencoder³⁹ kind of architecture, in which the neural network is supplied with the input image of the waveguide and train it to produce to mode profiles. Though promising, the implementation of such structure requires a huge data set and effort in training.

In summary we have presented a novel deep learning architecture that is capable of predicting mode images of an optical channel waveguide with high precisions. The input to the model is a vector containing the geometrical and material parameters of the waveguide. The internal architecture of the model make uses a deep deconvolutional neural network, and functions in two stages. The first stage transforms the input to a vector in higher dimensional space, and the second stage reshapes the vector to a matrix and performs resolution scaling. The presented deep learning architecture is generic for a prediction of a specific mode with a pre-defined polarization and symmetry. In this article, we illustrate training and predictions of fundamental TE waveguide modes. To predict fundamental transverse magnetic (TM) waveguide modes, one has to simply retrain the model with TM data sets, and optimize the hyperparameters (i.e., number of neurons and filters). For multimode waveguides, the presented architecture has to be adopted for each different modes and the boundaries of parameter space have to be re-defined to include the mode cut-offs.

The mode effective refractive index can be obtained from feedforward deep learning models³² with high accuracies. These architectures are extensively described in our earlier papers. As mode profiles, and effective refractive indices are related to each other, in future, predictions of these two quantities can be combined. This can be potentially done using a dual-output deep learning architecture and employing Keras functional APIs^{38,39}.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 9 July 2022; Accepted: 1 December 2022

Published online: 19 January 2023

References

1. Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, 2016).
2. Angra, S. & Ahuja, S. Machine learning and its applications: A review. *International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, Chirala, 57–60 (2017).
3. Sharma, N., Jain, V. & Mishra, A. An analysis of convolutional neural networks for image classification. *Procedia Comput. Sci.* **132**, 377–384 (2018).
4. Klosowski, P. “Deep Learning for Natural Language Processing and Language Modelling”, *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, Poznan, 223–228 (2018).
5. Nassif, A. B., Shahin, I., Attili, I., Azzeh, M. & Shaalan, K. Speech recognition using deep neural networks: A systematic review. *IEEE Access* **7**, 19143–19165 (2019).
6. Pierson, H. A. & Gashler, M. S. Deep learning in robotics: A review of recent research. *Adv. Robot.* **31**, 821–835 (2017).
7. Zibar, D., Wymeersch, H. & Lyubomirsky, I. Machine learning under the spotlight. *Nat. Photonics* **11**, 749–751 (2017).
8. Rivenson, Y. *et al.* Deep learning microscopy. *Optica* **4**, 1437–1443 (2017).
9. Sinha, A., Lee, J., Li, S. & Barbastathis, G. Lensless computational imaging through deep learning. *Optica* **4**, 1117–1125 (2017).
10. Turdnev, M. *et al.* Ultracompact photonic structure design for strong light confinement and coupling into nanowaveguide. *J. Lightw. Technol.* **36**, 2812–2819 (2018).
11. Malkiel, I. *et al.* Deep learning for design and retrieval of nanophotonic structures. [arXiv:1702.07949](https://arxiv.org/abs/1702.07949) (2017).
12. Ma, W., Cheng, F. & Liu, Y. Deep-learning-enabled on-demand design of chiral metamaterials. *ACS Nano* **12**, 6326–6334 (2018).

13. Liu, D., Tan, Y., Khoram, E. & Yu, Z. Training deep neural networks for the inverse design of nanophotonic structures. *ACS Photonics* **5**, 1365–1369 (2018).
14. Kojima, K., Wang, B., Kamilov, U., Koike-Akino, T. & Parsons, K. Acceleration of FDTD-based inverse design using a neural network approach. In *Advanced Photonics 2017 (IPR, NOMA, Sensors, Networks, SPPCom, PS)*, OSA Technical Digest (online), paper ITu1A.4 (Optical Society of America, 2017).
15. Rahman, B. M. A., Fernandez, F. A. & Davies, J. B. Review of finite element methods for microwave and optical waveguides. *Proc. IEEE* **79**, 1442–1448 (1991).
16. Mabaya, N., Lagasse, P. E. & Vandenbulcke, P. Finite element analysis waveguides of optical. *IEEE Trans. Microw. Theory Tech.* **29**, 600–605 (1981).
17. Yu, C. P. & Chang, H. C. Yee-mesh-based finite difference eigenmode solver with PML absorbing boundary conditions for optical waveguides and photonic crystal fibers. *Opt. Express* **12**, 6165–6177 (2004).
18. Shi, W., Cao, J., Zhang, Q., Li, Y. & Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **3**(5), 637–646. <https://doi.org/10.1109/JIOT.2016.2579198> (2016).
19. Cao, K., Liu, Y., Meng, G. & Sun, Q. An overview on edge computing research. *IEEE Access* **8**, 85714–85728. <https://doi.org/10.1109/ACCESS.2020.2991734> (2020).
20. Abbas, N., Zhang, Y., Taherkordi, A. & Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **5**(1), 450–465. <https://doi.org/10.1109/JIOT.2017.2750180> (2018).
21. Alharbi, F. Full-vectorial meshfree spectral method for optical-waveguide analysis. *IEEE Photonics J.* **5**, 6600315 (2013).
22. Song, D. & Lu, Y. Y. Pseudospectral modal method for computing optical waveguide modes. *IEEE J. Lightw. Technol.* **32**, 1624 (2014).
23. Burke, D. R. & Smy, T. J. Optical mode solving for complex waveguides using a finite cloud method. *Opt. Express* **20**, 17783 (2012).
24. Jiang, J., Chen, M. & Fan, J. A. Deep neural networks for the evaluation and design of photonic devices. *Nat. Rev. Mater.* **6**, 679–700 (2021).
25. Alagappan, G. *et al.* Leveraging AI in photonics and beyond. *Photonics* **9**, 75 (2022).
26. Ma, W. *et al.* Deep learning for the design of photonic structures. *Nat. Photonics* **15**, 77–90 (2021).
27. Wiecha, P. R. & Muskens, O. L. Deep learning meets nanophotonics: A generalized accurate predictor for near fields and far fields of arbitrary 3D nanostructures. *Nano Lett.* **20**, 329–338 (2020).
28. An, S. *et al.* A deep learning approach for objective-driven all-dielectric metasurface design. *ACS Photonics* **6**, 3196–3207 (2019).
29. Alagappan, G. & Png, C. E. Modal classification in optical waveguides using deep learning. *J. Mod. Opt.* **66**, 557–561 (2018).
30. Peurifoy, J. *et al.* Nanophotonic particle simulation and inverse design using artificial neural networks. *Sci. Adv.* **4**, eaar4206 (2018).
31. Gostimirovic, D. & Ye, W. N. An open-source artificial neural network model for polarization-insensitive silicon-on-insulator subwavelength grating couplers. *IEEE J. Sel. Top. Quantum Electron.* **25**, 1–5 (2019).
32. Alagappan, G. & Png, C. E. Universal deep learning representation of effective refractive index for photonics channel waveguides. *J. Opt. Soc. Am. B* **36**, 2636–2642 (2019).
33. Alagappan, G. & Png, C. E. Deep learning models for effective refractive indices in silicon nitride waveguides. *J. Opt.* **21**, 035801 (2019).
34. Saleh, B. E. A. & Teich, M. C. *Fundamentals of Photonics* (Wiley, 1991).
35. Gallagher, D. F. G. & Felici, T. P. Eigenmode expansion methods for simulation of optical propagation in photonics: Pros and cons. In *Proc. SPIE 4987, Integrated Optics: Devices, Materials, and Technologies VII* (2003).
36. Lumerical Inc. <https://www.ansys.com/products/photonics/mode>
37. <https://optics.ansys.com/hc/en-us/articles/360034382694-Symmetric-and-anti-symmetric-BCs-in-FDTD-and-MODE>
38. Moolayil, J. An introduction to deep learning and Keras. In *Learn Keras for Deep Neural Networks* (Apress, 2019). https://doi.org/10.1007/978-1-4842-4240-7_1.
39. Chollet, F. *Deep Learning with Python* (Manning Publications, 2018).
40. <https://en.wikipedia.org/wiki/AlexNet>
41. <https://github.com/jtambasco/modesolverpy>

Author contributions

G.A. formulated the idea, performed calculations and wrote the manuscript. C.E.P. participate in discussion and revised the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to G.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023