Optik - International Journal for Light and Electron Optics 273 (2023) 170494

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.com/locate/ijleo

Efficient lens design enabled by a multilayer perceptron-based machine learning scheme

Menglong Luo [a], Bishal Bhandari [a], Hongliang Li [a], Stuart Aberdeen [b], Sang-Shin Lee [a,b,*]

[a] Department of Electronic Engineering, Kwangwoon University, Seoul 01897, South Korea
[b] Nano Device Application Center, Kwangwoon University, Seoul 01897, South Korea

ARTICLE INFO

Keywords:
Efficient lens design
Machine learning
Artificial neural networks
Multilayer perceptron

ABSTRACT

Machine learning is a major branch of artificial intelligence that has been widely implemented in optical applications. Here we establish and validate two fully connected feed forward artificial neural networks (ANNs) based on the multilayer perceptron incorporating double hidden layers. The constructed ANNs act as regressors to efficiently predict the divergence and deflection angles of beams emerging from the beam-converging and -deflecting lenses, respectively. The target lens specifications can then be inversely queried by the desired beam divergence and deflection angles contained in the forecasted datasets. With the aid of meticulous hyperparameter tuning, the optimized ANNs of the beam-converging and -deflecting lenses yield high coefficient of determination (R^2) scores of 9.9964e–1 and 9.9933e–1, and low mean squared error (*MSE*) losses of 1.0e–5 and 2.2e–5, respectively. Compared with the conventional optical design, the proposed scheme has been confirmed to substantially alleviate the complexity of lens design, provide rich lens specification solutions for different beam divergence and deflection angles, and drastically reduce the computation time by over four orders of magnitude.

1. Introduction

Geometric optical lenses are indispensable elements in a wide range of optical modules and systems. They are particularly exploited in light detection and ranging [1,2], near-infrared imaging [3], microscopy [4], and medical imaging [5]. Currently, in various satellite and wireless communication systems, the data rate and throughput are impeded by the narrow propagation angle of information-carrying beam [6,7]. Additionally, nonmechanical light detection and ranging systems involving optical phased arrays are also prone to exhibit relatively large divergence angles and limited scanning ranges, resulting in unsatisfactory resolution and cramped field of view [8–10]. The finite propagation angle and excessive divergence of beams hinder versatile optical applications. Therefore, geometric lenses, which can widen the beam deflection angle and shrink the beam divergence, act as a viable alternative to overcome this challenge. However, the lens specifications required to diminish the beam divergence and magnify the deflection angle are initially unknown. As a result, devising geometric lenses with the desired characteristics is a complex and iterative process that entails the extensive tuning of multiple parameters. Multitudinous combinations of lens parameters, such as different radii and thicknesses, lead

Abbreviations: Machine learning, ML; Artificial neural network, ANN; Multilayer perceptron, MLP.

* Corresponding author at: Department of Electronic Engineering, Kwangwoon University, Seoul 01897, South Korea.
E-mail address: slee@kw.ac.kr (S.-S. Lee).

https://doi.org/10.1016/j.ijleo.2022.170494
Received 24 September 2022; Received in revised form 18 December 2022; Accepted 28 December 2022
Available online 29 December 2022
0030-4026/© 2022 Elsevier GmbH. All rights reserved.

to different performance in terms of the beam divergence and deflection angle. The process of setting appropriate parameters leading to the desired lens performance can be viewed as an optimization process, which requires repeated iterations and endures massive computation time. Under such circumstances, there is an imperative demand to develop a strategy to speed up the derivation of lens structural parameters while simultaneously streamlining the lens development process.

Over the past decade, machine learning (ML) has been extensively explored for diverse photonics applications, such as optical fiber-optic imaging [11,12], optical lenses [13,14], crystal fibers [15,16], [17], waveguides [18–20], sensors [21,22], photodetectors [23], and denoising of micro-interferograms [24–27]. ML can fit the independent and dependent variables of a dataset and approximate sophisticated functions with various nonlinear activation functions, thus avoiding the potential errors caused by human learning. Numerous ML algorithms are dedicated to solving classification and regression problems, which differ in whether the output of an artificial neural network (ANN) is discrete or continuous [28]. The widespread deployment of ML in optics is partly attributable to its capability to mitigate inevitable limitations such as high demands on the computer hardware configuration, manual adjustment of multiple parameters, and lack of rapid optimization. The ML scheme holds the potential to be integrated into conventional optical design tools to overcome the slow and troublesome lens development process, thereby relieving lens design practitioners of tremendous labor and reliance on proficient skills [29]. In this work, we put forward two ANNs based on a perceptron resorting to double hidden layers, which can efficiently predict the performance of the two lenses devised. The proposed lenses are exploited to shrink the beam divergence in the vertical direction and flexibly enlarge the deflection range of the beam in the lateral direction, respectively. The trained ANNs avoid the laborious process of manually modifying the lens parameters, thereby abating the concomitant complexity and computation time of the lens design. The proposed approach is anticipated to expand the scope of artificial intelligence-empowered optical design.

2. Lenses proposed for adjusting the convergence of beam and amplifying its deflection angle

In this work, we are primarily concerned about the design of two different types of lenses, a beam-converging lens (lens A) and beam-deflecting lens (lens B), which are deemed to minify the divergence of a beam emitted by an optical fiber and enlarge the deflection angle of a collimated beam, respectively. Lenses A and B are quintessential examples of lenses for reducing beam divergence and enhancing beam deflection. The proposed ML-based scheme mainly focuses on these two types of lenses but is not limited to them, which can be readily applied to other optical components. The two lenses are made of polymethyl methacrylate with a refractive index of 1.48 at a wavelength of 1550 nm. They have been rigorously devised and inspected using a ray-optic tool LightTools (Synopsys Inc., USA), which is a widely utilized commercial simulation tool. It is worth mentioning that an additional Synopsys commercial software, with the same reputation as Zemax, is CODE V, which is mainly used for imaging design, while illumination analysis is undertaken by LightTools. Therefore, LightTools from the same company as CODE V is categorically trustworthy in terms of geometric optical design. The equations defining the front and rear aspheric surfaces of lenses A and B can be described as $y(z)_{\text{lens A}} = \frac{cz^2}{1+\sqrt{1-(1+k)c^2z^2}}$ and $y(x)_{\text{lens B}} = \frac{cx^2}{1+\sqrt{1-(1+k)c^2x^2}}$ [30], where c and k represent the curvature and conic constant, respectively. To demonstrate that lens A helps suppress the beam divergence, in LightTools, the laser source is particularly configured to mimic the large divergence angle characteristics of an ultra-high numerical aperture fiber (UHNA4 type, Nufern), exhibiting a divergence angle of ~28° under a Gaussian intensity distribution. As delineated in Fig. 1(a), a beam travels through the anterior surface of the lens A at an incidence angle θ_{in} and exits its

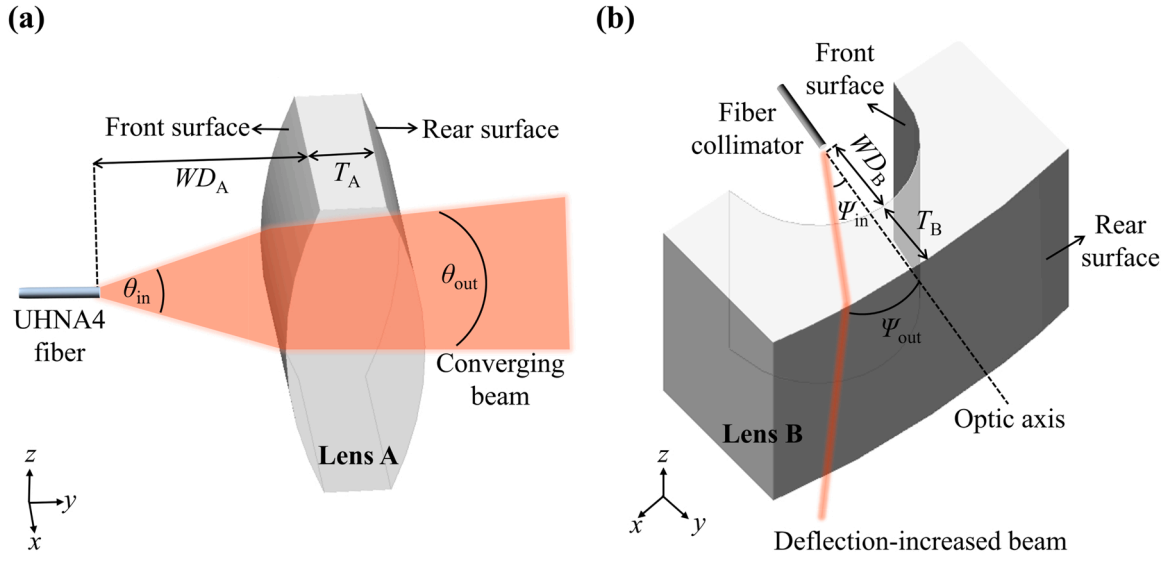

Fig. 1. Schematic of the designed lenses and their ray-optic operation. (a) Converging of a beam from a fiber through lens A. (b) Increased deflection of a collimated beam through lens B.

posterior surface into air, serving a reduced angular divergence θ_{out}. Here, θ_{out} is represented as θ_{out_1/e^2} and θ_{out_FWHM} corresponding to full widths at $1/e^2$ and half maximum levels, respectively. The working distance (WD_A), thickness (T_A), front and rear conic constants (CC_{A_F} and CC_{A_R}, respectively), and front and rear curvatures (C_{A_F} and C_{A_R}, respectively) of the lens are the parameters that mainly affect its ability to reduce θ_{in}. In the cases of different geometric parameters, θ_{out_1/e^2} and θ_{out_FWHM} vary from 3.04° to 21.18° and 1.88–12.54°, respectively. Next, Fig. 1(b) portrays the behavior of lens B, which enlarges the deflection angle of a cluster of light rays emerging from a fiber collimator with respect to the optic axis. The outgoing beam leaves the lens with an enlarged deflection angle ψ_{out} in the lateral direction. The design parameters of lens B may include the working distance (WD_B), thickness (T_B), front and rear conic constants (CC_{B_F} and CC_{B_R}, respectively), and front and rear curvatures (C_{B_F} and C_{B_R}, respectively). In response to the geometric parameters of the lens, the angle of deflection is amplified from $\psi_{in} = 15°$ to ψ_{out}, which varies from 17.64° to 44.83°. The beam divergence angles θ_{out_1/e^2} and θ_{out_FWHM}, and the deflection angle ψ_{out} are adopted as the indicators to assess the performance of the proposed lenses A and B. Field curvature aberration, stigmatism, and spherical error are evaluation indexes that are used to measure the image-formation property of an optical system. In this work, we focus on the illumination (non-imaging) implemented by the optical system rather than imaging. Therefore, the aforementioned imaging characteristics used to measure the performance of the designed optical lenses cannot be obtained.

3. ML-based design of the proposed lenses

The transition from the conventional lens design to an ML-based lens development entails a series of steps to be accomplished. As described in Fig. 2, the relating transformation process includes the accumulation of raw simulated data for lenses A and B, preprocessing of the acquired data, training of ANNs, performance evaluation of the trained ANNs, and selection of lens specifications from the predicted lens performance. A typical approach of ML is regarded as supervised learning, which utilizes algorithms to explore the relation between the input and corresponding output data contained in the training set, resulting in an inferring function. The yielded function is then used to predict outputs for previously unknown input data samples. Hence, the simulated data samples that consist of arrays of independent and dependent variables called features and labels, respectively, play a crucial role in ML. The objective of training ANNs in this work is to build a map that infers the optical performance of lenses from their geometric parameters. The parameters of the lenses and their performance are collected as features and labels of the ANNs, respectively. Each lens itself has various geometric parameters, such as its length, width, height, thickness, radius of the curvature, conic constant, and working distance. Reducing the number of features and dimensions of the training data set can make the generalization of an ANN more robust, reduce the possibility of overfitting, and enhance the correlation between features and labels. Consequently, it is necessary to select the parameters that have the greatest impact on the optical performance of the lens as the features for the training data set. Parameters such as length, width, and height, which have little influence on the optical path of light beams passing through the lenses, exhibit the least correlation with the lens performance and should be discarded. The feature selection eliminates irrelevant features, thereby

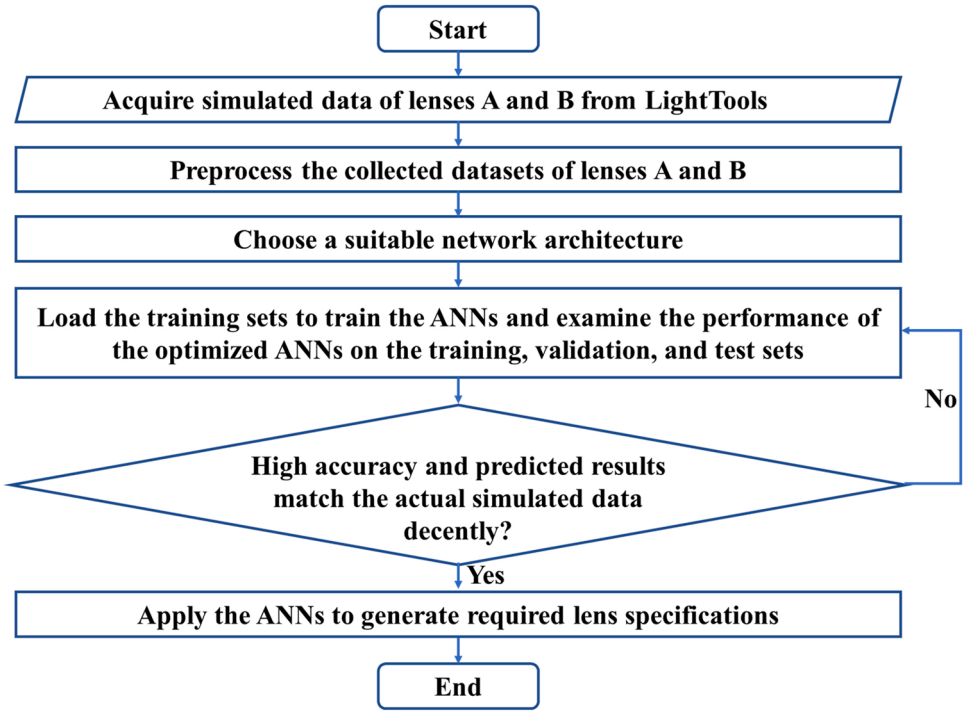

Fig. 2. Flowchart of the proposed ML-based lens design.

reducing the noise of the dataset, improving the accuracy of an ANN, and compressing the training time considerably. A built-in tool in LightTools called Parameter Analyzer is adopted to collect datasets that can be fed into the ANNs to train them and examine their prediction power. The optical characteristic parameters of lenses A and B are utilized as input features X_{Ai} and X_{Bi} ($1 \leq i \leq 6$), respectively. Concurrently, the lens performance θ_{out_1/e^2} and θ_{out_FWHM} are considered as the corresponding labels Y_{As} ($1 \leq s \leq 2$) of the ANN for lens A, while ψ_{out} is the output label Y_B of the ANN for lens B. After the acquisition conditions are configured, the tool automatically runs simulations for all combinations of input features to produce the corresponding outcomes. The obtained datasets comprise 13,690 and 13,366 sample combinations for lenses A and B, respectively. The features of the model acquired from lenses A and B are tabulated in Tables 1 and 2 below, respectively.

The prediction power of an ANN depends in part on the quality of the training set, which is critical to ensure precise predictions. However, feeding raw data directly to a network model may lead to an inferior prediction quality. Hence, preprocessing of raw datasets should be fulfilled via operations encompassing denoising, partitioning, and scaling. When the lens parameters vary over a wide range, abnormal parameter combinations occur to engender rays with a bimodal non-Gaussian distribution [31]. By comparing and analyzing the normal Gaussian beam divergence in the acquired data, the ratio of θ_{out_1/e^2} to θ_{out_FWHM} is between 1.5 and 1.8 (mostly around 1.7). A beam with a ratio of θ_{out_1/e^2} to $\theta_{out\_FWHM} \notin [1.5,\ 1.8]$ exhibits a bimodal angular profile. If bimodal non-Gaussian beams are introduced to the dataset adopted by the ANN of the lens A, they are deemed to mislead the training of the ANNs and impair their predictions. Therefore, rays with bimodal profiles are classified as noisy data, which lack learning significance for the network to be trained and therefore are filtered out prior to operation. After denoising, the dataset size of lens A is slightly reduced to a value of 12918. Considering the performance of lens B is determined by ψ_{out}, which does not involve the issue of bimodal non-Gaussian distribution, the dataset of lens B requires no denoising and thus its volume remains at a value of 13366. The denoised dataset of lens A and dataset of lens B are then shuffled separately, where 80% of datasets are randomly split into the training sets for the ANNs, while 25% of the data in the training sets are set aside for validation. The ANNs terminate training early when the validation scores do not improve within 15 consecutive epochs. The remaining 20% of the denoised data are assigned to an unknown test set to evaluate the generalizability of the trained model. Consequently, the ratio of training, validation, and test data volumes is determined to be 6:2:2. The features and labels extracted at different scales make unequal contributions to the fitting and learning of the model. The objective of scaling is to convert the acquired input features and output labels into a specific scale, thereby reducing the number of iterations to accelarate the gradient descent of the ANNs to discovery the optimal solutions and decreasing the value of loss function to enhance the prediction accuracy. Thereupon, input features and output labels of each column in the datasets are scaled to $[-1,\ 1]$ and $[0,\ 1]$, respectively.

The selection of an appropriate ANN architecture ensues when the data preprocessing is completed. During training, the selected framework learns from the provided data to execute predictions. Scikit-learn is a renowned and powerful ML library which can handle classification and regression problems [32]. The relation between the specifications and performance of a lens can be regarded as a regression problem. Therefore, the MLPRegressor belonging to the scikit-learn library, a popular algorithm in the field of ML that is adept in handling regression problems, was taken into consideration [33]. The underlying multilayer perceptron (MLP) framework of MLPRegressor facilitates the ANNs to efficiently execute lens design. However, before finalizing the MLPRegressor, linear regression models were also investigated. However, the accuracy of the linear regression models was slightly inferior to that of MLPRegressor so that the linear models were discarded. The performance comparison between the linear regression models and proposed MLP-based ANNs on the test sets of lenses A and B can be found in Table 6 of Section 4.1. Each of the two ANNs constructed consists of an input layer with six inputs, two hidden layers with forty neurons each, and an output layer, where the output layers of the ANNs for lenses A and B possess two outputs and one output, respectively. The neurons in each hidden layer are seamlessly connected to each neuron in the adjacent layers. The schematic of the ANN of lens A is illustrated in Fig. 3. The ANN of lens B is almost identical to that of lens A, except that the number of output neurons is one less. The performance of the trained ANNs is judged by its capability to fit the loaded training sets and deliver precise predictions. An poorly tuned ANN is deemed to fail to befit the training data. Indispensable hyperparameters in the hidden layers are inclusive of the depth and width of hidden layers, nonlinear activation function used to create sophisticated mappings between the inputs and outputs, weight optimization solver, batch size, and initial learning rate. The selection of hyperparameters entails a complicated process. Extensive experiments with various hyperparameters were fulfilled prior to finalizing the configuration of the two ANNs. Ultimately, a set of hyperparameters that lead to a reasonably high accuracy of the constructed ANNs were selected. It is worth mentioning that the ANNs of lenses A and B share the same set of hyperparameter configurations. The deployment of the ANNs is enumerated in Table 3.

Assessing a trained ANN in terms of such as the coefficient of determination (R^2) score and mean squared error (*MSE*) loss on the training, validation, and test sets is presumed to shed light on whether the model is configured with the appropriate hyperparameters. An R^2 score of ~1 and an *MSE* loss close to 0 signify high correspondence between actuals and forecasts. Once the hyperparameters pertaining to an ANN are tailored, with increasing number of epochs, the R^2 scores corresponding to the training, validation, and test

Table 1
Features collected from the lens A.

Boundand increment	WD_A [mm] (X_{A1})	T_A [mm] (X_{A2})	CC_{A_F} (X_{A3})	CC_{A_R} (X_{A4})	C_{A_F} [mm^{-1}] (X_{A5})	C_{A_R} [mm^{-1}] (X_{A6})
Lower bound	4	4	–1	–1	0.05	0.05
Upper bound	6	6	1	1	0.125	0.125
Increment	1	1	1	1	0.00625	0.00625

Table 2
Features collected from the lens B.

Bound and increment	WD_B [mm] (X_{B1})	T_B [mm] (X_{B2})	CC_{B_F} (X_{B3})	CC_{B_R} (X_{B4})	C_{B_F} [mm^{-1}] (X_{B5})	C_{B_R} [mm^{-1}] (X_{B6})
Lower bound	15	3	–1.5	–1	–0.2	0.02
Upper bound	17	5	–0.5	1	–0.06	0.05
Increment	1	1	0.5	1	0.01	0.003

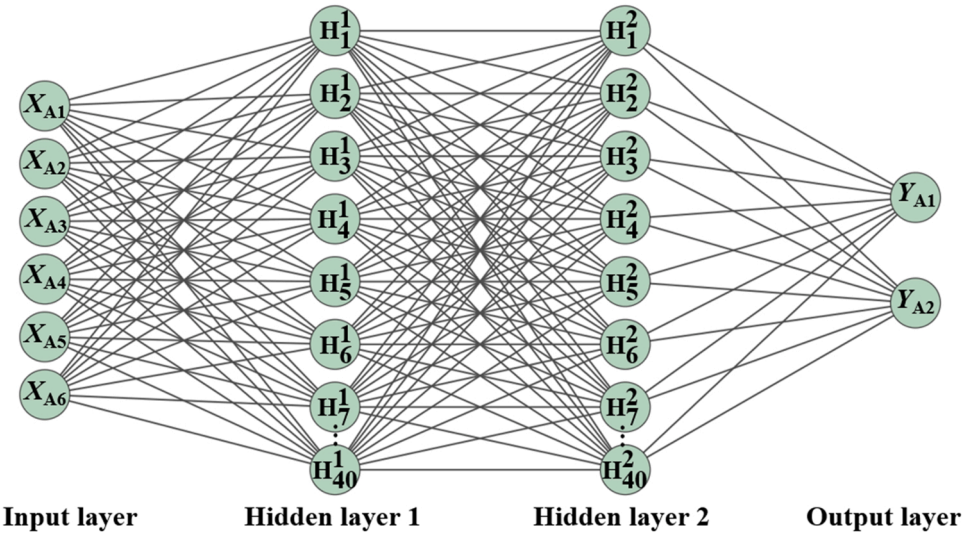

Fig. 3. Schematic of the proposed MLP-based ANN of the lens A incorporating an input layer (six inputs), two hidden layers (forty neurons per layer), and an output layer (two outputs).

Table 3
Configuration involved in the constructed ANNs.

Parameter	Value	Function
Algorithm	MLPRegressor	A supervised learning algorithm that can solve regression problems.
Depth	2	The number of hidden layers. The role of the multiple hidden layers is to further partition different types of data.
Width	40	The number of neurons in each hidden layer. Multiple neurons can perform multiple nonlinear divisions to continuously approach the decision boundary of the algorithm.
Activation function	ReLU	The ReLU function $f(x) = \max(0, x)$, which is exploited to add nonlinear factors to solve problems that cannot be solved by linear models. ReLU handles the common vanishing gradient problem and is the fastest activation function to compute gradients.
Solver	Adam	For weight optimization, the loss function can be minimized by adjusting the weights and biases. The significance of the loss function is to calculate the deviation of the actual value of the target value from the predicted value.
Batch size	128	The number of data samples captured per epoch during training. It makes the direction of gradient descent more accurate and improves the training speed.
Initial learning rate	0.001	It plays a decisive role in the convergence of the ANNs.
Early stopping	Yes	It prevents overfitting of the ANNs. The model reaches most stable state after 174 and 160 epochs for lenses A and B, respectively.

sets are observed to approach 1 whereas the *MSE* losses decline to 0. Further verification was performed by comparing the agreement between the forecast outcomes and simulation data of lenses A and B. The metrics as well as the agreement between predicted and actual values evidence that the ANNs can give rise to accurate predictions. Both the accuracy of the proposed ANNs and the comparison between the ML-based lens design and its conventional counterpart are rigorously expounded in Section 4.

4. Result analysis

The proposed ANNs were implemented in the programming language of Python and executed on a computer equipped with an Intel Core i5–8500 processor with a base clock frequency of 3.00 GHz and memory of 32 GB. The performance of the trained ANNs for lenses A and B has been appraised by examining the accuracy in terms of the R^2 score and *MSE* loss. To intuitively judge the effectiveness of

the ANNs, the predictions were compared with the actual simulation data. Additionally, the process of the constructed ANNs predicting the lens specifications was elaborated, and complexity and computation time of the ANNs and LightTools were comparatively scrutinized.

4.1. R^2 scores and MSE losses of the proposed ANNs

R^2 score is one of the most crucial statistics reflecting how well an ANN fits the dataset, while the *MSE* loss measures difference between the predicted and actual values. High R^2 scores and low *MSE* losses on the training, validation, and test sets imply that an ANN is deployed with appropriate hyperparameters and can conduct precise forecasts. Various values for the depth and width of the hidden layers, and activation functions were experimented before deciding on the final hyperparameter configuration. The deployment of the depth and width of the hidden layers is the primary issue to be addressed. When deploying the ANNs with one hidden layer, the ANN of lens A achieved an R^2 score of 9.9886e–1 on the test set. With the configuration of two hidden layers, the trained ANN could realize a higher prediction accuracy with an R^2 score of 9.9964e–1. The depth of the hidden layers continued to increase to three, at which point the R^2 score dropped to 9.9918e–1 and the training time increased with the number of hidden layers. Furthermore, deeper hidden layers may incur overfitting issues. Thus, two hidden layers were finally assigned to the ANNs. The issue of determining the width of per hidden layer ensues when the depth of the hidden layers has been fixed. As a rule of thumb, the width of the hidden layer may be between the number of neurons in the input and output layers [34]. However, the width required for the hidden layers depends more on the complexity of the training cases encountered and is not necessarily in accordance with the aforementioned rule. Determining the hidden layer width is a process of trial and error [35]. When determining the number of neurons per hidden layer, experiments were first carried out empirically. However, the R^2 scores were relatively low. To improve the R^2 scores, 40 neurons were selected for each hidden layer after extensive attempts. The R^2 scores and *MSE* losses of the ANNs on the test sets of lenses A and B are shown in Table 4.

Among commonly used activation functions including Sigmoid function, the ReLU function is most prominent, which has been chosen as the default activation function of MLP. The advantages of the ReLU function are as follows. Firstly, it increases the nonlinearity of an ANN, making it more accurate in fitting the data set [36]. Secondly, the calculation speed of ReLU is faster than Sigmoid function. The formulas of Sigmoid and ReLU functions are $f_{\text{Sigmoid}}(x) = \frac{1}{1+e^{-x}}$ and $f_{\text{ReLU}}(x) = \max(0, x)$, respectively. Since the Sigmoid function involves computationally intensive exponential operations, while the operation of ReLU function is not complex, which can be simply conducted by calculating a threshold. ReLU brings sparsity to an ANN to make the ANN more efficient [37]. Thirdly, when the input of the Sigmoid function is an extremely large positive or small negative value, its output tends to be saturated, which destroys the gradient of an ANN. The vanishing of the gradient makes it arduous to update the weights of the model, which drastically hinders the learning ability of an ANN. However, the ReLU function circumvents the gradient vanishing problem [38]. Lastly, the output of Sigmoid function is not zero-centered, resulting in the sluggish convergence of the ANN, whereas ReLU has a huge acceleration effect on the convergence of stochastic gradient descent [39]. Softmax function is hardly used in the hidden layers but mostly used in the output layer of an ANN. Thus, the scope of its application is relatively narrow. It is generally dedicated to multi-classification tasks to predict probability distributions over mutually exclusive class labels. Therefore, in the context of the regression problem encountered in this work, Softmax is not a worthy choice of activation function compared to ReLU. Since the proposed ANNs using ReLU function as an activation function achieve a reasonably high accuracy and precisely predict the lens performance, ReLU activation function is preferred to Sigmoid and Softmax ones. The accuracy on the test sets and training time-consuming comparisons of the ANNs for the cases of Sigmoid and ReLU functions are detailed in Table 5. Furthermore, the accuracy of the ANNs for lenses A and B is compared with linear regression models on their test sets, as shown in Table 6, which manifests that the ANNs based on the MLP algorithm do enhance the prediction accuracy. Consequently, it is necessary to utilize MLP-based ANNs to deal with the forecasting of the lens performance. Ultimately, the ANNs were determined to deploy the hyperparameters in Table 3 above.

The metric curves of the ANNs with the final configuration on the training, validation, and test sets are plotted in terms of epoch in Fig. 4. For lens A, Fig. 4(a) and (b) show that the initial R^2 scores start at a low level in the vicinity of 0 and the maximum starting value of *MSE* loss is 5.4411e–2. It is observed that the R^2 score improves and the *MSE* loss diminishes over the iterative process, assuming slight fluctuations. After 174 epochs, the training is terminated as the validation score shows no further increase in 15 successive epochs, resulting in a final average R^2 score and *MSE* loss of 9.9957e–1 and 1.3e–5 on the dataset, respectively. A similar trend is witnessed for lens B, as shown in Fig. 4(c) and (d). The final average R^2 score reaches 9.9931e–1, and the average *MSE* loss drops to

Table 4
R^2 scores and *MSE* losses on the test sets of lenses A and B for ANNs with different hidden layer widths.

Number of hidden neurons	ANN of the lens A		ANN of the lens B	
	R^2	MSE	R^2	MSE
2	9.8639e–1	3.5e–4	9.8694e–1	3.7e–4
3	9.8678e–1	3.7e–4	8.8700e–1	3.8e–4
4	–3.4428e–3	2.8e–2	9.8736e–1	3.7e–4
5	9.8689e–1	3.7e–4	9.8829e–1	3.4e–4
6	9.9248e–1	2.1e–4	9.8883e–1	3.2e–4
40	9.9964e–1	1.0e–5	9.9933e–1	2.2e–5

Table 5
The R^2 scores and *MSE* losses on the test sets, and training time comparison for ANNs using the Sigmoid and ReLU activation functions.

Activation function	ANN of the lens A			ANN of the lens B		
	R^2	*MSE*	Training time	R2	MSE	Training time
Sigmoid	9.9784e–1	6.0e–5	16.2 s	9.9579e–1	1.4e–4	11.5 s
ReLU	9.9964e–1	1.0e–5	4.7 s	9.9933e–1	2. 2e–5	4.3 s

Table 6
Performance comparison between linear regression models and the proposed MLP-based ANNs on the test sets of lenses A and B.

Model	Lens A		Lens B	
	R^2	*MSE*	R^2	*MSE*
Linear regression models	9.8742e–1	3.4e–4	9.5360e–1	1.4e–3
MLP-based ANNs	9.9964e–1	1.0e–5	9.9933e–1	2.2e–5

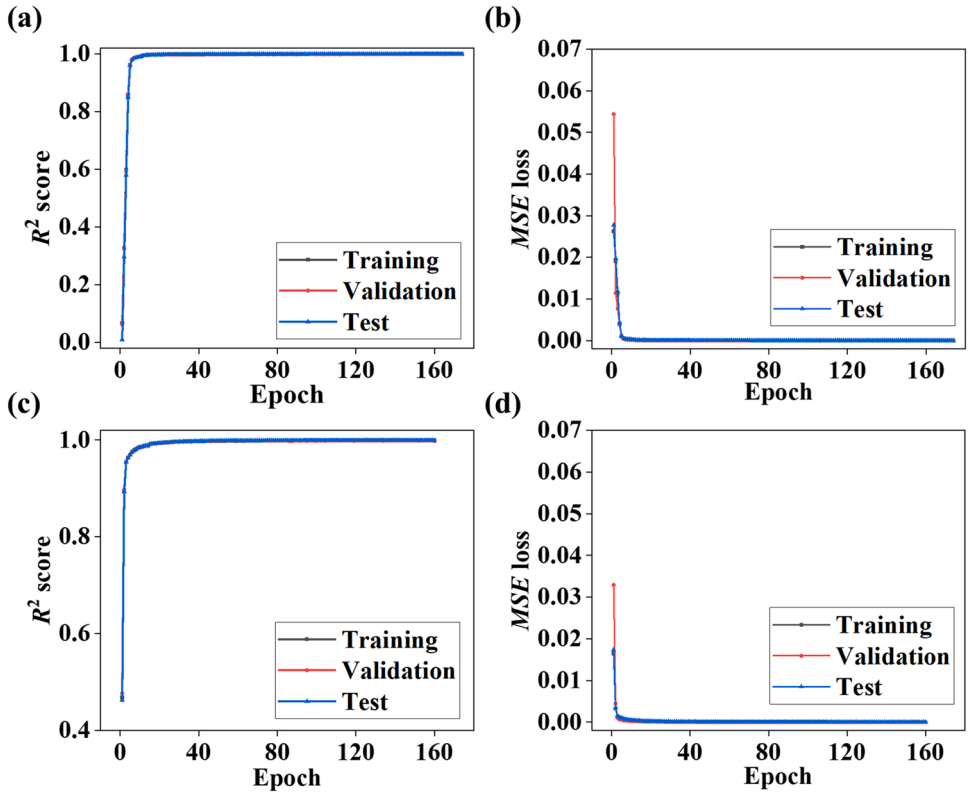

Fig. 4. Evaluation of the performance of the proposed ANNs for lenses A and B on the training, validation, and test sets during an iterative process. (a) R^2 score and (b) *MSE* loss of the ANN for the lens A. (c) R^2 score and (d) *MSE* loss of the ANN for the lens B.

Table 7
R^2 scores and *MSE* losses of the ANNs for lenses A and B.

Data set	ANN of the lens A		ANN of the lens B	
	R^2	*MSE*	R^2	*MSE*
Training	9.9966e–1	1.0e–5	9.9937e–1	1.9e–5
Validation	9.9942e–1	2.0e–5	9.9930e–1	2.5e–5
Test	9.9964e–1	1.0e–5	9.9933e–1	2.2e–5

2.2e–5, with both metrics stabilizing after 160 epochs. Initially, the *MSE* loss curves for training, validation, and test sets are slightly different from each other, but eventually these curves converge and maintain almost the same value, which signifies that the models sufficiently fit the datasets without overfitting and underfitting. The R^2 scores and *MSE* losses are listed in Table 7.

4.2. Comparison between the predicted and actual lens performance

The accuracy of the ANNs after training has been investigated in terms of the R^2 score and *MSE* loss. To further visualize the prediction power and generalizability of the ANNs, comparisons between the predicted and simulated outcomes are presented in Fig. 5. Fig. 5(a)–(c) depict the agreement between the predicted and actual values of θ_{out_1/e^2}, θ_{out_FWHM}, and ψ_{out}, respectively. Since the data used for validation are also partitioned from the training set, Fig. 5 compares the actual data samples and predicted results on the training and test sets. Fig. 5(a) show a linear relationship between the actual and predicted θ_{out_1/e^2} on the training and test sets, which demonstrates the accuracy and generalization of the ANNs, respectively. The fitting is linear with a relation $y = kx$, where k is the slope. The slopes of the two fitted lines for the cases of training and test are 0.9994 and 0.9998, respectively. Hence, the regression equation is given by $y \approx x$, indicating a nearly exquisite prediction accuracy. In the same manner, Fig. 5(b) and (c) reveal the actual θ_{out_FWHM} and ψ_{out} alongside their predictions on the training and test sets, as in the case of Fig. 5(a). The datapoints display negligible deviations from the fitted lines, signifying strong correlations between the predicted and actual values. To examine the consistency between the predicted and actual results, a statistical measure of mean absolute error (*MAE*) was adopted in a bid to represent the mean of the absolute error between predicted and actual values. Typically, an *MAE* value close to zero implies that the predictions closely match the actual simulated results. The slopes of fitted lines and *MAE* values for θ_{out_1/e^2}, θ_{out_FWHM}, and ψ_{out} are listed in Table 8. The *MAE* of θ_{out_1/e^2} and θ_{out_FWHM} is slightly lower than that of ψ_{out}, implying that the ANN predicts the performance of lens A more accurately than the case of lens B. Nevertheless, the level of *MAE* corresponding to ψ_{out} is still low and thus acceptable. Considering the slopes of the fitted lines are in the vicinity of 1, it can be claimed that the ANN-based prediction of the performance of lens B is credible. Consequently, the superb prediction accuracy of the proposed ANNs has been verified.

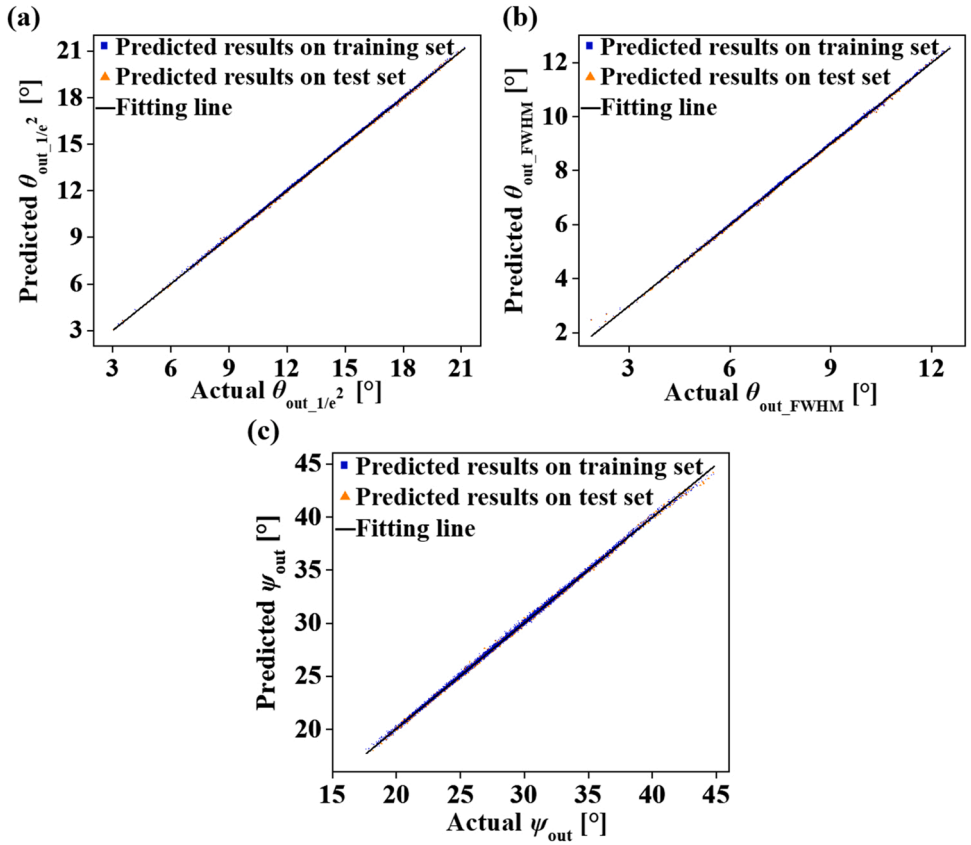

Fig. 5. Comparison of the actual θ_{out_1/e^2}, θ_{out_FWHM}, and Ψ_{out} simulated in LightTools with the predicted results of the proposed ANNs. (a) Actual and predicted θ_{out_1/e^2} for the training and test sets. (b) Actual and predicted θ_{out_FWHM} for the training and test sets. (c) Actual and predicted ψ_{out} for the training and test sets.

Table 8
Slopes of the fitted lines and *MAE* values corresponding to the actual and predicted results.

Lens performance	Training set		Test set	
	Slope	*MAE*	Slope	*MAE*
θ_{out_1/e^2}	0.9994	0.0432	0.9998	0.0430
θ_{out_FWHM}	1	0.0250	0.9996	0.0255
ψ_{out}	0.9951	0.0905	0.9948	0.0941

4.3. Process of the proposed ANNs to forecast lens specifications and comparison of complexity and computation time between the ANNs and LightTools

When it comes to the design of a lens, its geometric parameters are to be tailored to exert optimum performance as per the requirements. Thus, a relatively slow and complicated process of manually setting numerous parameter combinations is unavoidable. Moreover, the command macros of widely used commercial optical design tools are burdensome for beginners, and manual tuning does not necessarily secure favorable lens performance. Even experienced engineers are often required to spend considerable time and effort writing and debugging programs to create lenses leading to the desired performance. The trained ANNs automatically analyze a substantial number of design parameters pertaining to a lens so as to predict its performance, thereby obviating manual parameter tuning and creation of complex programs. The proposed ANNs are intended to relieve the burden of lens designers.

Compared with forward prediction which infers the lens performance from its parameters (*X* to *Y*), backward prediction (*Y* to *X*) is more in conformity with actual needs of designers. However, backward prediction is not feasible for the work reported here. The feature-to-label ratios of the ANNs for lenses A and B are 6:2 and 6:1, respectively, where the dimension of lens parameters is much higher than that of the lens performance. The large variances in dimensions of the lens parameters and performance make the ANNs particularly imprecise when using lens performance to forecast parameters of the lenses. Besides, another critical reason is that one-to-many mapping occurs when predicting the specification of a lens based on its performance, i.e., multiple combinations of lens parameters may lead to an identical performance, which makes the forecasts ill-posed and tricky. The inverse prediction accuracy of an MLP model that infers *X* from *Y* has been tested to be as low as ~30 %. Given the particularity of the situation presented here, a strategy that can achieve the same purpose as the back prediction has been proposed. Although the constructed ANNs cannot be utilized to directly deduce the lens specifications from the lens performance, the ANNs can generate massive fresh datasets. Then, the only remaining work in the entire process is to select the required lens specifications. It is worth emphasizing that generating data is the core of the whole work and most time consuming. It takes full advantage of the extremely fast forward prediction speed of the ANNs to implement massive predictions, and then retrieves the lens specifications in reverse via the lens performance. The input features of lenses A and B can be further divided in finer increments within the original intervals, then ANNs enable predictions on the subdivided features owing to its swift inference speed and high accuracy, facilitating the generation of larger datasets in response to additional lens specifications and corresponding beam divergence and deflection angles. Under the condition where the amount of data generated is sufficiently large, lens technicians can back-search lens specifications with respect to the beam divergence and deflection angles predicted in the datasets, thus nullifying the need to finely tune the geometric parameters of the lenses, and readily realizing the desired performance. To expedite the querying of lens specifications, the produced datasets can be written into a database server MySQL (Oracle, US), which is an open-source relational database management system for data warehousing based on the Structured Query Language (SQL). Since MySQL is capable of rapidly organizing large datasets and executing complex conditional queries with the aid of flexible Structured Query Language (SQL), MySQL is utilized to select the desired lens specifications from the new datasets produced by the ANNs. When the generated data sets are written into MySQL, MySQL can reversely query the corresponding lens specifications according to the lens performance in a time period of microseconds. This method provides a new train of thought, which is different from the direct derivation of the lens specifications based on the lens performance. Considering writing data to the database and reverse lookup is neither time-consuming nor strenuous to implement, it is worth considering combining ANNs with MySQL.

The fast inference speed of the proposed ANNs has been practically corroborated by comparing the computation time required for predictions with the simulation time of LightTools, which takes about 14 h to run 10,000 times of simulations. Meanwhile, for the same number of predictions, the proposed ANNs spend only a few seconds, proving that the developed model saves a substantial amount of inference time. The time consumption involved is summarized in Table 9.

5. Discussion and outlook

The two ANNs constructed in this work may not be directly applicable to the design of other optical lens scenarios, because the input features and output labels involved in other lens scenarios may be dissimilar to those required by the two established ANNs. However, with some modifications to the hyperparameters of the ANNs, the proposed MLP-based ML approach can be adopted for other lens design tasks. Due to the negligible burden of retraining a model, the architecture of the ANNs used in this work can be applied to forecast other lens solutions after a brief training period of around 5 s when a fresh training set is available. As a widely used reliable ML algorithm, the high efficiency and accuracy of MLP has been verified for the optical design by other researchers as well.

Recently, multi-lens and free-form lens systems have been widely utilized. Compared with the design of a single geometric lens, devising combined lenses and free-form curvilinear surfaces can be more complicated. Since the complex optical components contain

Table 9
Computation time for the cases of LightTools and proposed ANNs for 10,000 times of simulations and predictions.

Design scheme	Time consumption	Time-consuming ratio of LightTools and ANNs
LightTools	13.9 h	29,435
Proposed ANNs	~1.7 s	

more relevant parameters that affect the performance of the optical system, their design is more time-consuming and labor intensive. These optical systems have more pressing demands for ANN-based optical design. The MLP-based ANN scheme can be retrained for combined lens systems and free-form lenses by simply reacquiring new training sets. Due to the substantially low time consumption and little difficulty of retraining the model, the proposed ANN framework also holds enormous possibilities in the design of complex optical systems for imaging and illumination purposes. However, the architecture of the ANNs proposed in this work is still subject to limitations in inverse design. Further work to integrate the current ANN architecture with advanced algorithms such as tandem neural networks, flow-based generative models, and variational autoencoders, may be conducted to thoroughly implement precise inverse predictions. It is believed that the upgraded ANN should be able to simplify and accelerate the design of more diverse optical structures.

6. Conclusion

In summary, an efficient ML proposal based on an MLP with two hidden layers was proposed and developed to expedite and streamline the lens design process. The proposed two ANNs can rapidly predict the performance of the two devised lenses with disparate parameters, and the target lens specifications can be reversely retrieved from the established database MySQL based on desired lens performance. The most noticeable feature of the scheme is that the ANN models avoid the cumbersome procedure of manually adjusting lens parameters and reduce the calculation time by a factor of 29,435 compared to LightTools. The demonstrated ML approach is highly anticipated to act as a prominent alternative or supporting means to conventional optical design tools.

Declaration of competing interest

The authors declare no conflicts of interest.

Data Availability

Data will be made available on request.

Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (No. 2018R1A6A1A03025242), Ministry of Science and ICT (2020R1A2C3007007), and Kwangwoon University in 2022.

References

[1] M. Zohrabi, W.Y. Lim, R.H. Cormack, O.D. Supekar, V.M. Bright, J.T. Gopinath, Lidar system with nonmechanical electrowetting-based wide-angle beam steering, Opt. Express 27 (2019) 4404–4415.
[2] C.-S. Im, S.-M. Kim, K.-P. Lee, S.-H. Ju, J.-H. Hong, S.-W. Yoon, T. Kim, E.-S. Lee, B. Bhandari, C. Zhou, S.-Y. Ko, Y.-H. Kim, M.-C. Oh, S.-S. Lee, Hybrid integrated silicon nitride–polymer optical phased array for efficient light detection and ranging, J. Light. Technol. 39 (2021) 4402–4409.
[3] S. Banerji, M. Meem, A. Majumder, F.G. Vasquez, B.S. Rodriguez, R. Menon, Ultra-thin near infrared camera enabled by a flat multi-level diffractive lens, Opt. Lett. 44 (2019) 5450–5452.
[4] P. Pozzi, M. Quintavalla, A.B. Wong, J.G.G. Borst, S. Bonora, M. Verhaegen, Plug-and-play adaptive optics for commercial laser scanning fluorescence microscopes based on an adaptive lens, Opt. Lett. 45 (2020) 3585–3588.
[5] J. Wu, B.N. Li, Development of a magnetoinductive lens for magnetic resonance imaging, IEEE Instrum. Meas. Mag. 20 (2017) 56–60.
[6] S. Lu, M. Gao, Y. Yang, R. Zhu, X. Hou, J. Sun, W. Chen, X. Zhu, Inter-satellite laser communication system based on double Risley prisms beam steering, Appl. Opt. 58 (2019) 7517–7522.
[7] J.G. Marin, J. Hesselbarth, Lens antenna with planar focal surface for wide-angle beam-steering application, IEEE Trans. Antennas Propag. 67 (2019) 2757–2762.
[8] C.V. Poulton, A. Yaacobi, D.B. Cole, M.J. Byrd, M. Raval, D. Vermeulen, M.R. Watts, Coherent solid-state LIDAR with silicon photonic optical phased arrays, Opt. Lett. 42 (2017) 4091–4094.
[9] H. Li, C. Zhou, W.-B. Lee, D.-Y. Choi, S.-S. Lee, Flat telescope based on an all-dielectric metasurface doublet enabling polarization controllable enhanced beam steering, Nanophotonics 11 (2022) 405–413.
[10] H. Ito, Y. Kusunoki, J. Maeda, D. Akiyama, N. Kodama, H. Abe, R. Tetsuya, T. Baba, Wide beam steering by slow-light waveguide gratings and a prism lens, Optica 7 (2020) 47–52.
[11] Z. Liu, L. Wang, Y. Meng, T. He, S. He, Y. Yang, L. Wang, J. Tian, D. Li, P. Yan, M. Gong, Q. Liu, Q. Xiao, All-fiber high-speed image detection enabled by deep learning, Nat. Commun. 13 (2022) 1–8.
[12] L. Wang, Y. Yang, Z. Liu, J. Tian, Y. Meng, T. Qi, T. He, D. Li, P. Yan, M. Gong, Q. Liu, Q. Xiao, High-speed all-fiber micro-imaging with large depth of field, Laser Photonics Rev. 16 (2022) 2100724.
[13] G. Côté, Y. Zhang, C. Menke, J. Lalonde, S. Thibault, Inferring the solution space of microscope objective lenses using deep learning, Opt. Express 30 (2022) 6531–6545.

[14] T. Yang, D. Cheng, Y. Wang, Designing freeform imaging systems based on reinforcement learning, Opt. Express 28 (2020) 30309–30323.
[15] M.A. Jabin, M.P. Fok, Prediction of 12 photonic crystal fiber optical properties using MLP in deep learning, IEEE Photon. Technol. Lett. 34 (2022) 391–3941.
[16] S. Chugh, S. Ghosh, A. Gulistan, S. Ghosh, B.M.A. Rahman, Machine learning approach for computing optical properties of a photonic crystal fiber, Opt. Express 27 (2019) 27523–27535.
[17] S. Sridevi, T. Kanimozhi, N. Ayyanar, Sunny Chugh, M. Valliammai, J. Mohanraj,, Deep learning based data augmentation and behavior prediction of photonic crystal fiber temperature sensor, IEEE Sens. J. 22 (2022) 6832–6839.
[18] G. Alagappan, C.E. Png, Prediction of electromagnetic field patterns of optical waveguide using neural network, Neural Comput. Appl. 33 (2021) 2195–2206.
[19] G. Alagappan, C.E. Png, Deep learning models for effective refractive indices in silicon nitride waveguides, J. Opt. 21 (2019), 035801.
[20] Y. Meng, Y. Chen, L. Lu, Y. Ding, A. Cusano, J.A. Fan, Q. Hu, K. Wang, Z. Xie, Z. Liu, Y. Yang, Q. Liu, M. Gong, Q. Xiao, S. Sun, M. Zhang, X. Yuan, X. Ni, Optical meta-waveguides for integrated photonics and beyond, Light Sci. Appl. 10 (2021) 1–44.
[21] A. Ghosh, P. Satvaya, P.K. Kundu, G. Sarkar, Calibration of RGB sensor for estimation of real-time correlated color temperature using machine learning regression techniques, Optik 258 (2022), 168954.
[22] A. Ghosh, A. Pal, N.R. Das, An approach to design photonic crystal gas sensor using machine learning, Optik 208 (2020), 163997.
[23] W. Cao, Y. Huang, K. Fan, J. Zhang, A novel machine learning algorithm for large measurement range of quadrant photodetector, Optik 227 (2021), 165971.
[24] E.Z. Omar, T.Z.N. Sokkar, A.A. Hamza, In situ investigation and detection of opto-mechanical properties of polymeric fibres from their digital distorted microinterferograms using machine learning algorithms, Opt. Laser Technol. 129 (2020), 106295.
[25] G.A. Lila, T. Sokkar, E. Seisa, E. Omar, Adaptive investigation of the optical properties of polymer fibers from mixing noisy phase shifting microinterferograms using deep learning algorithms, Microsc. Res. Tech. 85 (2022) 667–684.
[26] E.Z. Omar, A refined denoising method for noisy phase-shifting interference fringe patterns, Opt. Quant. Electron. 53 (2021) 1–24.
[27] T.Z.N. Sokkar, E.A. Seisa, G.M. Abo-Lila, E.Z. Omar, Interferometric accurate investigation of opto-thermo-mechanical features with help of artificial intelligence for antimicrobial polyamide-6 fibres grafted by quaternary ammonium salt with nano zinc oxide, Optik 262 (2022), 169312.
[28] H.Q. Tran, C. Ha, Improved visible light-based indoor positioning system using machine learning classification and regression, Appl. Sci. 9 (2019) 1048.
[29] W. Chen, T. Yang, D. Cheng, Y. Wang, Generating starting points for designing freeform imaging optical systems based on deep learning, Opt. Express 29 (2021) 27845–27870.
[30] C. Pruss, E. Garbusi, W. Osten, Testing aspheres, Opt. Photonics N. 19 (2008) 24–29.
[31] J. Henningsen, Non-Gaussian modes in a HeNe laser, Opt. Express 17 (2009) 21427–21432.
[32] M. Tran, S. Panchal, V. Chauhan, N. Brahmbhatt, A. Mevawalla, R. Fraser, M. Fowler, Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery, Int. J. Energy Res. 46 (2022) 786–794.
[33] S. Chugh, S. Ghosh, A. Gulistan, B.M.A. Rahman, Machine learning regression approach to the nanophotonic waveguide analyses, J. Light. Technol. 37 (2019) 6080–6089.
[34] A. Blum. Neural networks in C++, Wiley, New York, 1992.
[35] P. Asadia, M. Gindy, M. Alvarezb, A. Asadic, A computer vision based rebar detection chain for automatic processing of concrete bridge deck GPR data, Autom. Constr. 112 (2020), 103106.
[36] A.D. Rasamoelina, F. Adjailia, P. Sinčák, A review of activation function for artificial neural network, 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (2020) 281–286.
[37] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR (2011) 315–323.
[38] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90.
[39] D. Arpit Y. Bengio The benefits of over-parameterization at initialization in deep ReLU networks arXiv Prepr. arXiv 1901.03611.