Overcoming Hardware Imperfections in Optical Neural Networks Through a Machine Learning-Driven Self-Correction Mechanism

Minjoo Kim, Beomju Kim, Yelim Kim, Lia Saptini Handriani, Suhee Jang, Dae Yeop Jeong, Sung Ik Yang, and Won Il Park

***Abstract*—We developed an optical neural network (ONN) for efficient processing and recognition of 2-dimensional (2D) images, employing a conventional liquid crystal display panel as optical neurons and synapses. This configuration allowed for optical signal outputs proportional to matrix-vector multiplication for 2D image inputs. However, our experimental results revealed a 26.6% decrease in the optical classification accuracy, despite utilizing digitally pre-trained parameters with 100% accuracy for 500 handwritten digits. This decline can be attributed to system imperfections associated with non-ideal functions of optical components and optical alignment. Rather than pursuing an elusive, imperfection-free ONN or attempting to calibrate these defects individually, we addressed these challenges by introducing a self-correction mechanism that utilizes a machine learning algorithm. This approach effectively restored the recognition accuracy and minimized loss of our ONN to levels comparable to the digitally pre-trained model. This study underscores the potential of constructing defect-tolerant hardware in ONNs through the application of machine learning techniques.**

***Index Terms*—Hardware imperfections, matrix-vector multiplication (MVM), optical neural networks (ONNs), recognition accuracy, self-correction approach, training algorithm.**

I. Introduction

ARTIFICIAL neural networks (ANNs) have been developed to emulate the efficient information processing capabilities of biological neural networks (BNNs). However, traditional computer systems based on the Von Neumann architecture [1], [2], with separate central processing units (CPUs) and memories, face limitations in parallel neuromorphic data processing. To overcome these limitations, hardware solutions such as graphics processing units (GPUs) and field programmable gate arrays (FPGAs) have been incorporated into neuromorphic computing [3], [4]. [5], [6]. Additionally, hardware-implemented neural networks using devices like memristor crossbar arrays [7], [8], [9] and transistor-based memory arrays [10], [11], [12] have demonstrated energy-efficient computations in neural algorithms. However, electronic systems have inherent limitations such as nonlinear potentiation/depression characteristics and small differences in memory conduction states [13]. The issues of electronic interconnections and crosstalk interference further restrict the construction of high-density neuromorphic networks comparable to BNNs [14], [15].

In contrast, optical neural networks (ONNs) have emerged as an alternative solution for implementing neuromorphic computations using optical signals. ONNs offer advantages in terms of processing speed, energy efficiency, and low heat generation compared to their electronic counterparts [16], [17], [18], [19]. Leveraging the parallel processing capabilities of optical components, ONNs enable fast and simultaneous computations on large amounts of data [20], [21]. The use of optical rays with multiple wavelengths allows for parallel transmission in free space with minimal interference. This hardware could offer an effective solution for implementing various deep learning models, including transformers [22], convolutional neural networks (CNNs) [23], and spiking neural networks (SNNs) [24], to meet their computational demands across a range of applications.

II. Background and Related Works

Recent advancements in ONNs have demonstrated fast and energy-efficient computations, achieving vector computing speeds of 11 trillion operations per second (TOPS) [25] and energy efficiency up to 2.5×10^{-19} J per weight multiplication [17]. To achieve high computation power and accuracy, many ONN implementations import digitally pre-trained parameters and utilize sophisticated components and systems [17], [20], [25], [26]. However, the integration of optic-to-electronic and electronic-to-optic conversions, modulation, and optical coupling in ONNs introduces potential challenges in terms of non-ideal operations and noise generation [27]. These factors can result in reduced system performance and computation accuracy. Despite efforts to minimize these imperfections, achieving imperfection-free hardware remains a challenge. The individual

Manuscript received 23 December 2023; revised 25 January 2024; accepted 30 January 2024. Date of publication 5 February 2024; date of current version 27 February 2024. This work was supported by the National Research Foundation (NRF) of Korea funded by the Ministry of Science, ICT and Future Planning (MSIP) of Korea under Grant 2021R1A5A1032996 and Grant 2021R1A2B5B02002596. *(Minjoo Kim, Beomju Kim, and Yelim Kim contributed equally to this work.) (Corresponding authors: Won Il Park; Sung Ik Yang.)*

Minjoo Kim, Beomju Kim, Yelim Kim, Lia Saptini Handriani, Suhee Jang, Dae Yeop Jeong, and Won Il Park are with the Division of Materials Science and Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: wipark@hanyang.ac.kr).

Sung Ik Yang is with the Department of Applied Chemistry, Kyung Hee University, Yongin 17104, South Korea (e-mail: siyang@khu.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/JPHOT.2024.3361930, provided by the authors.

Digital Object Identifier 10.1109/JPHOT.2024.3361930

calibration of these imperfections demands additional digital computations, eroding the inherent advantages of analog computations in ONNs. Additionally, the majority of experimental validations of ONN performance have been confined to assessing system accuracy solely during the optical feed forward (i.e., forward propagation), with few notable exceptions [28]. This constraint arises from the incapacity to effectively leverage previously computed information in the current stage, potentially impeding the learning of diverse patterns. In contrast, the human visual system, despite its inherent diversity and irregularity, outperforms artificial machines in the perception and recognition of diverse and irregular images directly collected from the environment.

In this study, we developed an ONN that emulates the efficient processing and recognition of 2-dimensional (2D) images, inspired by the human visual system. Instead of pursuing a completely imperfection-free ONN, we chose to implement the ONN hardware using readily available and cost-effective optical components. The RGB color pixel array in conventional liquid crystal display (LCD) technology served as our optical neurons and synapses, facilitating optical signal outputs proportionate to matrix-vector multiplication for 2D image inputs. The RGB color pixel array allows independent computation for each color signal. For clarity, this means that computations can be performed separately for each color channel. Notably, we harnessed this capability to represent both positive and negative weights simultaneously. During the evaluation of our ONN, utilizing digitally pre-trained parameters for 500 MNIST (Modified National Institute of Standards and Technology) [29] handwritten digits, we observed a decrease in recognition accuracy by 20–30%. This decrease can be attributed to system imperfections associated with non-ideal functions of optical components and optical alignment. However, by applying our self-correction approach that leverages a machine learning algorithm, we compensated for any hardware deficiencies so that our ONN can restore the recognition accuracy and loss of our ONN to levels comparable to the digitally pre-trained model. This study highlights the potential of constructing defect-tolerant hardware using a machine learning algorithm inspired by biological acclimation and adaptation processes.

III. Methods

A. Principle of ONN

Fig. 1(a) illustrates the operational scheme of a single-layer perceptron in our ONN system designed for the classification of MNIST handwritten digits among 10 categories (0, 1, 2, …, 9) [19], [30]. The input vector (X_i) and synaptic weights (W_{ij}) are represented as 2D pixel arrays, where i and j denote the order of each element among 784 pixels (i.e., 28 × 28) and the order of each class (j = 0, 1, 2, …, 9) of output nodes, respectively.

The input image generated by beam projector (1st plane) is projected onto the LCD plane (2nd plane), where the transmittance of each pixel is set proportionally to its corresponding W_{ij} value. The pixels of X_i and W_{ij} overlap in a one-to-one correspondence, allowing the transmitted light intensities of each pixel (I_{ij}) to be proportional to $X_i \times W_{ij}$ (Fig. S1). This results in optical signals undergoing scalar multiplications between each element of the input image and its corresponding weight. We also employed a white dot and bias B_j in the input and weight planes, respectively, to introduce an offset and enable the model to be more flexible. These were treated as the last element of the pixels (referred to as X_{785} and W_{785j} here after). Finally, optical MVMs are implemented by focusing the image outputs on fine points and detecting the intensities of the focused beams for the 10 classes of W_{ij} in parallel [31], [32].

B. ONN Hardware

We implemented the ONN hardware, as shown in Fig. 1(b), utilizing commercially available affordable components such as a beam projector and an LCD panel (see Supplement for additional details). These components were used to generate MNIST handwritten digit images and electronically updateable weights, corresponding to the 1st and 2nd planes in Fig. 1(a), respectively. The pixel values (X_i and W_{ij}) of both the beam projector and LCD are expressed with an 8-bit color depth, consisting of subpixel color indices (R, G, B) [33]. Each color index (R, G, and B) defines the intensity of red, green, and blue colors, respectively, with values ranging from 0 to 255. For monochromatic image recognition, one color channel is sufficient. However, since our system, based on incoherent light, cannot represent negative values for weights, we used red and blue colors to denote positive and negative signs of weights. Specially, unlike conventional digital neural networks where weight values typically range from -1 to 1, we normalized the weights to integer values within the range of –255 to 255. This allowed us to represent the weights directly using color indices, such that a W_{ij} value of 255 and –255 is expressed as (255, 0, 0) and (0, 0, 255) respectively.

After modulation of the input light by the LCD, the optical signals were focused by a lens onto an optical fiber to measure the intensities of red and blue light, I_j^R and I_j^B, respectively. Their difference I_j is assumed to be proportional to the true MVM outputs O_j:

$$I_j = I_j^R - I_j^B \tag{1}$$

$$O_j = \sum_{i=1}^{785} X_i \cdot W_{ij} = \alpha I_j. \tag{2}$$

where α (0.02) is employed to ensure comparability between the I_j values and the digital MVM outputs. Representative spectra and outputs achieved from a specific handwritten image (the 78th digit from the MNIST training dataset) are shown in Fig. 1(c). Based on these measurements, our system infers the image input as the digit '1' since the signal output I_1 (or O_1) is the highest among the 10 classes.

C. Digital Training and Optical Training

To obtain the weight and bias parameters for our ONN, we developed a single-layer neural network using LabVIEW (Fig. 2). As comparable to the conventional 'backward propagation of errors', the training algorithm for our ONN system involves Data loading, Training Loop, and End processes, as detailed follows:

[Data Loading]

- MNIST data (training data, N) is loaded from .csv files.

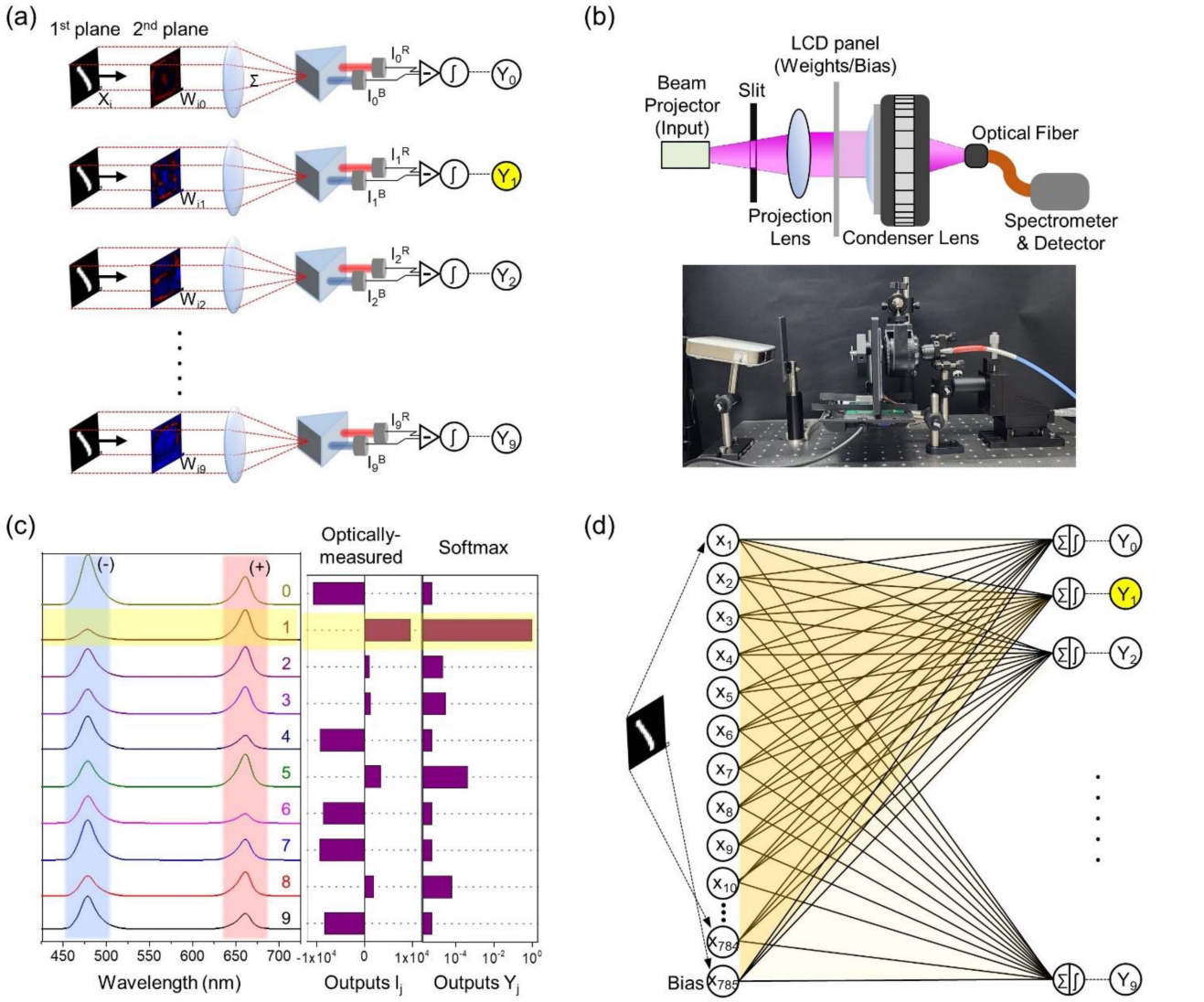

Fig. 1. (a) Operational scheme of a single-layer perceptron in the ONN system for MNIST handwritten digit classification. The input image in the first plane is projected onto a second plane, where the transmittance of each pixel corresponds to its weight value. Optical signals undergo scalar multiplications between the input image pixels and their corresponding weights, generating multiple 2D image outputs. Optical MVMs are implemented by focusing the image outputs and detecting the intensities of the focused beams. (b) Schematic (top) and photo image (bottom) of the hardware implementation of ONN. (c) Representative spectra and optically measured outputs I_j for the 78th MNIST handwritten image from training dataset for 10 classes (left and center panel). The outputs Y_j of the softmax function converted from the I_j (right panel). The optical measurement results and corresponding outputs for class "1" are highlighted in yellow. (d) Operational scheme of a conventional neural network system.

- Features and labels are extracted, converting 1D arrays of each image into 2D arrays.
- Labels undergo one-hot encoding.
- Weights are initialized.

[Training Loop]

- Feed-forward (MVM operation): Performs digital/optical MVM operation to compute the weighted sum (Oj(n)/Ij(n)).
- Calculate Loss: Applies softmax activation to generate output probabilities and computes the loss.
- Calculate Loss Gradient: Computes the loss gradient with respect to the weights of the network.
- Weight Update: Repeats the feed-forward and loss gradient calculation steps for N training data. Updates weights, iterating through epochs until accuracy exceeds target level.
- Return Accuracy and Loss: Returns the computed accuracy and loss for monitoring training progress.

[End]

- Shuts down the system.

1) Digital Training: In the data loading step, the network received input data comprising 500 handwritten digit images from the MNIST training dataset and initial weight values. For the digital training phase, weights were randomly assigned within the range of -255 to 255, after which they were fed into the network.

In the training loop, an iterative process involving feed-forward, loss calculation, loss gradient calculation, and weight update are employed. During feed-forward in the digital training phase, the network computed the MVM outputs $O_j(n)$ for the n^{th} input image. These outputs were then used to compute $Y_j(n)$,

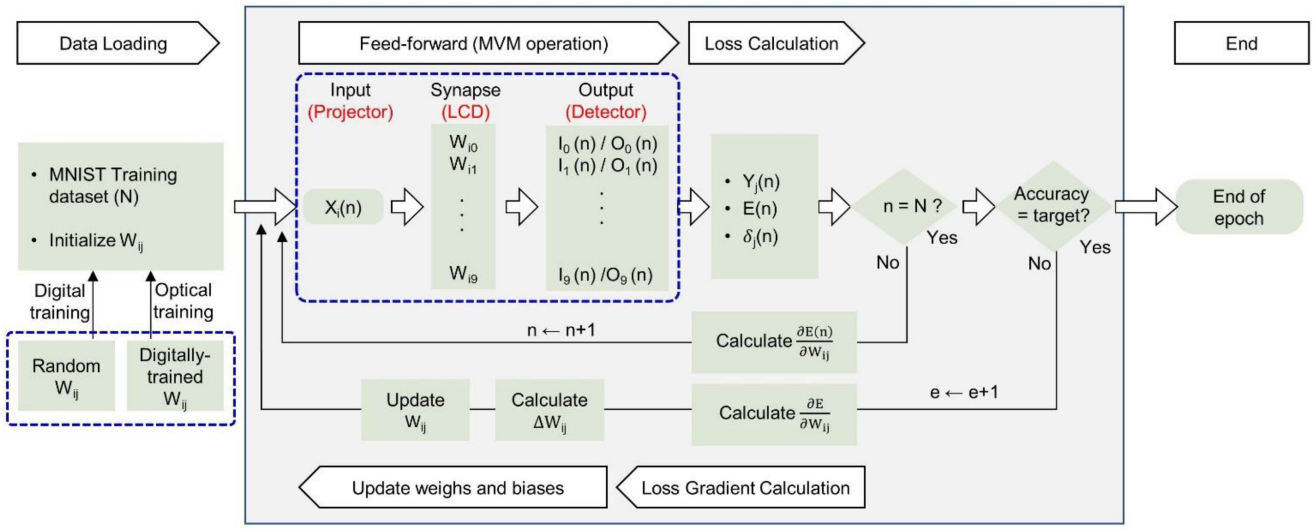

Fig. 2. Schematic illustrating the Digital and Optical Training Procedures. Our ONN system operates through a sequence of steps, including data loading, the Training Loop, and the end process. The weight parameters were initially randomly assigned and then optimized using the digital training process. These parameters were subsequently fed into the ONN system and further updated through optical training (i.e., the machine learning-based self-correction process) designed to address and compensate for hardware imperfections in our ONN. The Training Loop (gray box) is initiated with a feed-forward employing Matrix-Vector Multiplication (MVM) to calculate weighted sums. Subsequently, it calculates losses and loss gradients, traversing backward through the network. The averaged loss gradients for N inputs were employed to update weights, thereby facilitating a training process aimed at minimizing loss and maximizing classification accuracy.

which represents the output of the softmax function and is expressed as:

$$\mathrm{Y_j\,(n)\ =\ \frac{exp\,[O_j\,(n)]}{\sum_{j\,=\,1}^{10} exp\,[O_j\,(n)]}} \tag{3}$$

The loss $\delta_j(n)$ of j^{th} class was then computed using:

$$\mathrm{\delta_j\,(n)\ =Y_j\ \ (n)-T_j\,(n)} \tag{4}$$

where $T_j(n)$ represents the target value for each input digit image.

Our training algorithm employed the cross-entropy error loss function [34], denoted as $E(n)$, to iteratively optimize the weights and minimize the loss, thereby enabling the model to generate accurate outputs for given inputs. The $E(n)$ is defined as:

$$\mathrm{E\,(n)=-\sum_{j\,=\,1}^{10}\,[T_j\,(n)\cdot lnY_j\,(n)]} \tag{5}$$

To calculate the delta-rule weight increment, we leveraged the gradient of the loss with respect to W_{ij}, which can be computed as:

$$\mathrm{\frac{\partial E\,(n)}{\partial W_{ij}}\ =\delta_j\ \ (n)\cdot X_i\,(n)} \tag{6}$$

After repeating the feed-forward and loss gradient calculation steps for N training data, we calculate the average loss gradient $\frac{\partial E}{\partial W_{ij}}$ for N inputs:

$$\mathrm{\frac{\partial E}{\partial W_{ij}}\ =\ \delta_j\ \cdot\ X_i} \tag{7}$$

where δ_j and X_i are the averages of $\delta_j(n)$ and $X_i(n)$ for N inputs. From the averaged loss gradient, the weights are adjusted using the following equation:

$$\mathrm{W_{ij}\leftarrow W_{ij}+\Delta W_{ij}} \tag{8}$$

where

$$\mathrm{\Delta W_{ij}\ =\ -\eta\ \cdot\ \frac{\partial E}{\partial W_{ij}}} \tag{9}$$

Here, η represents the training rate. In our ONN system, η was set to 2, which is approximately two orders of magnitude larger than in a conventional digital training model [4], [35]. This choice was made due to the normalization of weights within the range of +/- 255 in our ONN system, which is also two orders of magnitude larger than the weights in a conventional digital training model. This process was repeated for multiple iterations (epochs) until a satisfactory level of accuracy was attained.

2) Optical Training (Correction of Hardware Imperfections): The ONN operation program employed algorithms similar to the digital neural network, differing only in the import of weight parameters and the implementation of MVM during the feed-forward stage (Fig. 2, indicated by blue broken lines). The weight parameters, pre-trained through the digital training process described above, were exported to our LCD synaptic layer in the ONN hardware.

In the feed-forward stage, the ONN hardware conducted optical MVM operations by measuring the I_j value for each class, as described in the previous Section B. Similar to digital training, the outputs of the softmax functions Y_j were computed from the optically measured outputs (1)–(3), as typical example is illustrated in the left panel in Fig. 1(c).

IV. Results and Discussion

The basic operation principle of our ONN is analogous to the Stanford-Vector-Multiplier architectures [31], but it is more

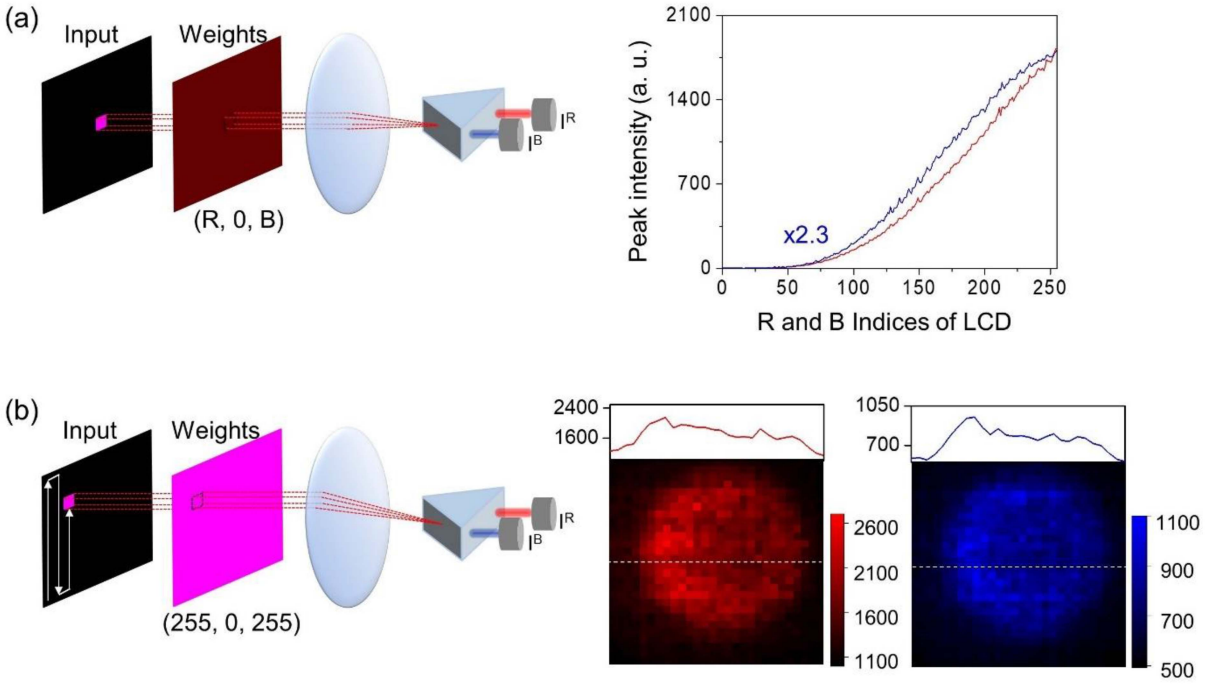

Fig. 3. Non-ideal performance and imperfections in the ONN System. (a) Non-linear behavior of light transmittance through the LCD pixels. (b) Non-uniform beam collection observed during fiber coupling. The plots in the top panels depict line profiles along the white dashed line in the images of the red and blue beams, respectively. The RGB indices of the single-pixel beams from the beam projector in (a) and (b) were set to (255, 0, 255) for both cases.

efficient for image recognition. In conventional neural networks, as shown in Fig. 1(d), 2D image inputs are converted to a 1D array and delivered to input nodes for subsequent signal processing and transfer to the output nodes. However, implementing conventional neural networks in hardware using 1D arrays for input nodes faces challenges as the number of input neurons increases. This results in increased system complexity. In this regard, our ONN architecture has an advantage in that 2D images can be directly imported without flattening the image or establishing individual wirings corresponding to the total number of pixels [19], [30], [36], [37]. Additionally, diverse wavelengths of light rays can be exploited as independent signals. In this case, different colors were employed to represent positive and negative signs of weights, as explained in Section III-B.

To assess the performance of our ONN, we selected an initial set of 500 samples from the MNIST training dataset, which comprises a total of 60000 samples. We created a digital model with a single fully connected layer, analogous to our ONN, and optimized the weights using a standard training algorithm (see Section III-C.1 and III-C.2 for more details). The training process for our digital model was continued until it achieved 100% classification accuracy specifically for the selected 500 samples.

After obtaining the digitally trained parameters, we imported them into our ONN hardware and conducted experimental classification using the same set of MNIST digits. However, our optical classification test yielded results that significantly deviated from the expected outcomes based on simulations. In fact, the classification accuracy in our optical test decreased by approximately 26.6% compared to the performance of the digital model [Fig. 4(a) and (b)], despite our dedicated efforts to minimize errors.

To investigate the reasons behind the observed discrepancy between the digital and optical tests, we examined the non-ideal performance of the components in our ONN. The analysis revealed several imperfections within the system. First, the light intensities of the input image from the beam projector were not linearly proportional to the target values of X_i and also varied depending on the color, as illustrated in Fig. S2. While we attempted to digitally compensate for the intensity differences between red and blue lights (by reducing the red input values by a factor of 2.3), the nonlinearities and offsets still persisted. Second, the light transmission through each pixel of the LCD varied non-linearly with the weight values represented by the R and B color indices of the LCD. In Fig. 3(a), we plotted the measurement results of transmitted light intensities for a single-pixel beam with an RGB index of (255, 0, 255) from the beam projector. It was noted that the transmitted intensities barely changed when the R and B indices were set below approximately 50, and gradually increased as the RGB values were further incremented. Third, we discovered non-uniform beam collection related to fiber coupling for both the red and blue beams. Despite our efforts in beam alignment, we observed that the beam at the core region exhibited approximately 1.3 to 1.5 times higher intensity compared to the beam at the edge region for both the red and blue beams [Fig. 3(b)].

Instead of rectifying the inherent hardware imperfections or substituting the existing components with faultless counterparts, we have devised a machine learning-based training algorithm to address and compensate for these imperfections, utilizing

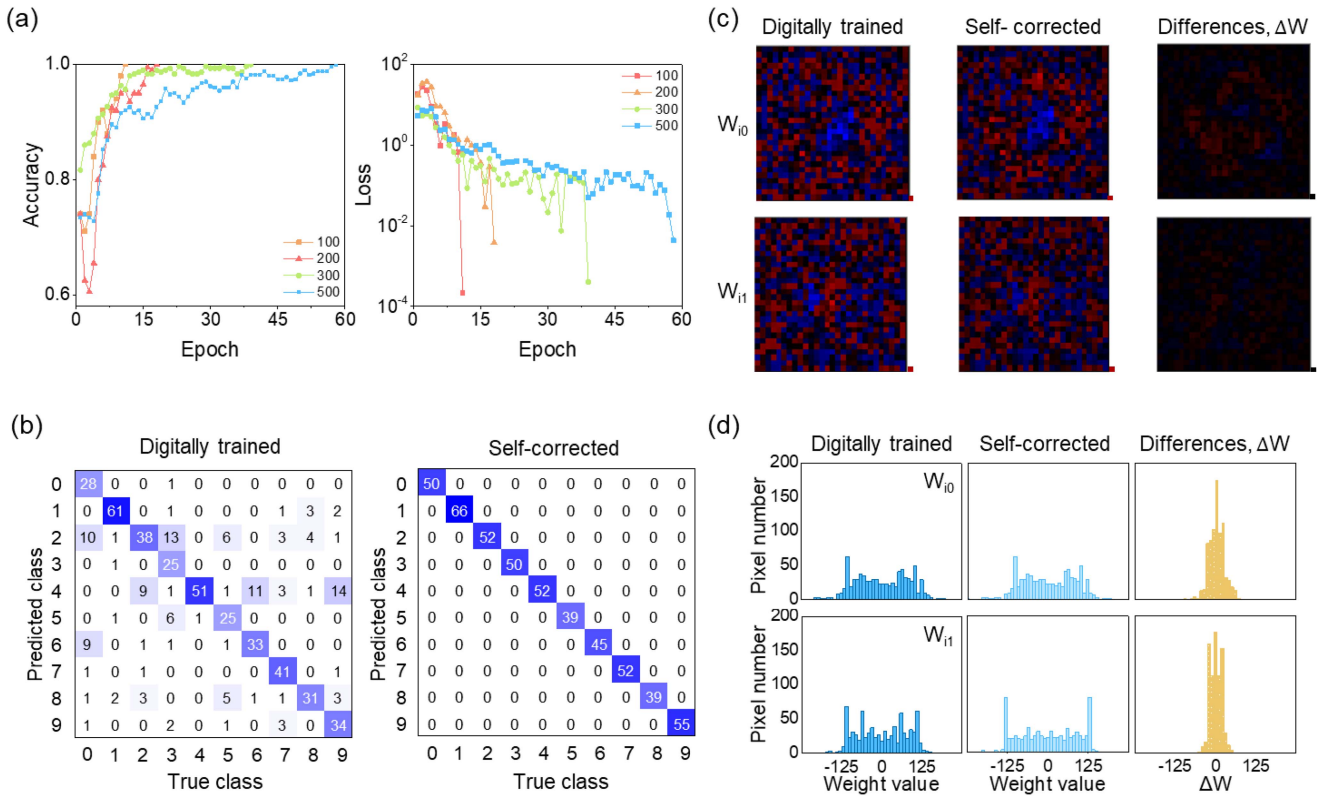

Fig. 4. Evaluation and analysis of the machine learning-based self-correction process. (a) Classification accuracy (left) and loss (right) during training epochs for subsets of 100, 200, and 500 handwritten digit images from the MNIST training dataset. (b) Confusion matrices illustrating the relationship between predicted classes and actual classes for the classification of 500 handwritten digits before and after the self-correction process. (c) 2D plots of the weights before (left) and after (center) self-correction, accompanied by their differences (right), for classes "0" and "1". The dots in the lower right corners correspond to the biases (or the last elements of weights, W_{785j}). (d) Histogram showcasing the distributions of weights before (left) and after (center) the self-correction process, along with their differences (right), for classes "0" and "1". In (b)–(d), the data labeled as "Digitally Trained" are obtained via optical feed-forward (i.e., optical MVM) operations employing the digitally pre-trained weights and biases. Digital pre-training was performed over 500 training datasets until the classification accuracy for the entire dataset reached 1 (after 45 iterative epochs). The data labeled as "Self-Corrected" are acquired after 59 epochs of optical training (self-correction) process, starting from the digitally pre-trained weights.

the algorithm described in the Section III-C.1 and III-C.2 (also refer to Fig. 2). As a result of iterative self-correction based on optical measurements, the classification accuracy of our system gradually improved. Notably, when subjected to testing with 100 and 200 handwritten digits, perfect classification (i.e., accuracy of 1) was achieved after 11 and 18 update epochs, respectively [Fig. 4(a)]. In pursuit of a more challenging benchmark, we extended our evaluation to include the Fashion MNIST dataset. Despite the heightened challenge posed by Fashion MNIST, particularly for our single-layer neural networks, our ONN system overcome hardware imperfections after training via our self-correction approach, restoring accuracy from 40% to 83% after 79 iterative epochs. We anticipate that upgrading our system to DNNs could potentially enhance its performance (Fig. S3).

While larger training sets required more iterations, optimal classification was primarily accomplished within 60 epochs for training sets containing fewer than 500 samples. For instance, a recognition accuracy of 100% was attained for 500 handwritten digits after 58 epochs. Throughout the training process, our ONN system effectively adjusted the weights to minimize the loss value, resulting in improved prediction accuracy. Consequently, the loss attributed to optical measurement errors exhibited a decrease of approximately three orders of magnitude as the accuracy approached 1. It should be noted that evaluation of our hardware performance focused solely on the training set consisting of 500 handwritten digits, rather than utilizing the MNIST test set. While achieving a classification accuracy above ~0.93 for the MNIST test set is feasible even with a single-layer perceptron model, the experimental implementation of our ONN is limited by suboptimal operating speed, thereby constraining its training capabilities to a level comparable to that of a digital neural network.

The confusion matrices in Fig. 4(b) depict the relationship between predicted classes and actual classes for the classification of 500 handwritten digits. Initially, when the weights were trained digitally, frequent misclassifications occurred, such as '9' being mistaken for '4' and '3' being mistaken for '2' [Fig. 4(b), left panel]. These incorrect predictions were likely due to the similarities in shape between the digits. However, after updating the weights to compensate for hardware imperfections, our ONN was able to accurately classify all 500 test digits from '0' to '9' [Fig. 4(b), right panel]. Fig. 4(c) illustrates 2D plots of the weights for classes '0' and '1' before (left panel) and

after (center panel) the self-correction process. Additionally, it presents the overall change in the weights. Similar plots are shown for other classes in Fig. S4. The corresponding histograms of these weight distributions are depicted in Fig. 4(d) and Fig. S5, which represent the fine adjustment process for system errors. It is evident that the magnitude of weight changes follows a Gaussian distribution with a standard deviation of approximately 23. A similar tendency was observed for most other classes. These findings suggest that slight changes in the weights can effectively eliminate hardware imperfections, leading to a nearly 1000-fold decrease in loss. In our implementation of the ONN, we initially utilized only a small portion of the display. However, the potential for achieving higher throughput is substantial and can be realized by (1) increasing the number of input neurons (X_i) and (2) performing parallel operations for multiple classes (W_j). Recent studies [30] exploring the use of megapixels of optical components suggest the potential for enhanced throughput, reaching up to 1 million multiply–accumulate (MAC) per electronic clock cycle (∼petaMAC/s). The incorporation of state-of-the-art high-speed photodetector arrays [37] could potentially push this throughput even higher. In addition, while we used only red and blue color pixels to represent sign in weights in this study, employing the entire RGB color pixel array allows for more independent computation for each color signal, thereby further improving computation power. In terms of energy efficiency, the use of LCD has some limitations due to the energy cost associated with refreshing the LCD pixels and high optical loss during light transmission (∼95% optical loss). Strategies such as employing fixed or static weights, including optical phase change materials [38] or photochromic materials, and/or high-speed optical sources [39], present promising approaches to potentially approach the theoretical quantum limit, set by shot noise[16], [19].

V. Conclusion

We have presented the development and evaluation of an ONN using inexpensive components that emulates the efficient processing and recognition capabilities of the human visual system. Our ONN system incorporates a self-correction mechanism that leverages a machine learning algorithm to address the inherent imperfections of the hardware. By incorporating a self-correction mechanism that leverages machine learning, we address hardware imperfections. Despite a 20–30% decrease in recognition accuracy caused by optical alignment and non-ideal functions of components, our self-correction approach restores accuracy to levels comparable to pre-trained models. This study highlights the potential of using machine learning for defect-tolerant hardware in optical neural networks, enhancing performance and reliability for image processing and recognition systems. Furthermore, we anticipate a substantial increase in throughput by harnessing most RGB pixels for computation, taking advantage of the high resolution of the display. This suggests a significant improvement in throughput at a larger scale, highlighting the scalability and potential performance enhancements of our ONN architecture in real-world applications.

References

[1] J. Von Neumann, "First draft of a report on the EDVAC," *IEEE Ann. Hist. Comput.*, vol. 15, no. 4, pp. 27–75, 1993.

[2] Z. Wang et al., "Resistive switching materials for information processing," *Nature Rev. Mater.*, vol. 5, no. 3, pp. 173–195, 2020.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 84–90.

[4] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2015, pp. 161–170.

[5] X. Liu, H. A. Ounifi, A. Gherbi, Y. Lemieux, and W. Li, "A hybrid GPU-FPGA-based computing platform for machine learning," *Procedia Comput. Sci.*, vol. 141, pp. 104–111, 2018.

[6] S. Mittal, "A survey of FPGA-based accelerators for convolutional neural networks," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 1109–1139, 2020.

[7] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[8] Y. Li and K.-W. Ang, "Hardware implementation of neuromorphic computing using large-scale memristor crossbar arrays," *Adv. Intell. Syst.*, vol. 3, no. 1, 2021, Art. no. 2000137.

[9] S. Li et al., "Wafer-scale 2D hafnium diselenide based memristor crossbar array for energy-efficient neural network hardware," *Adv. Mater.*, vol. 34, no. 25, 2022, Art. no. 2103376.

[10] S. Dai et al., "Recent advances in transistor-based artificial synapses," *Adv. Funct. Mater.*, vol. 29, no. 42, 2019, Art. no. 1903700.

[11] Y. Li et al., "One transistor one electrolyte-gated transistor based spiking neural network for power-efficient neuromorphic computing system," *Adv. Funct. Mater.*, vol. 31, no. 26, 2021, Art. no. 2100042.

[12] J. Shi et al., "A fully solution-printed photosynaptic transistor array with ultralow energy consumption for artificial-vision neural networks," *Adv. Mater.*, vol. 34, no. 18, 2022, Art. no. 2200380.

[13] M. M. Waldrop, "The chips are down for Moore's law," *Nature News*, vol. 530, no. 7589, 2016, Art. no. 144.

[14] F. Lacroix, M. Châteauneuf, X. Xue, and A. G. Kirk, "Experimental and numerical analyses of misalignment tolerances in free-space optical interconnects," *Appl. Opt.*, vol. 39, no. 5, pp. 704–713, 2000.

[15] M. E. Fouda, F. Kurdahi, A. Eltawil, and E. Neftci, "Spiking neural networks for inference and learning: A memristor-based design perspective," *Memristive Devices Brain-Inspired Comput.*, vol. 31, pp. 499–530, 2020.

[16] R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, "Large-scale optical neural networks based on photoelectric multiplication," *Phys. Rev. X*, vol. 9, no. 2, 2019, Art. no. 021032.

[17] G. Wetzstein et al., "Inference in artificial intelligence with deep optics and photonics," *Nature*, vol. 588, no. 7836, pp. 39–47, 2020.

[18] T. Zhou et al., "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nature Photon.*, vol. 15, no. 5, pp. 367–373, 2021.

[19] T. Wang, S. - Y. Ma, L. G. Wright, T. Onodera, B. C. Richard, and P. L. McMahon, "An optical neural network using less than 1 photon per multiplication," *Nature Commun.*, vol. 13, no. 1, 2022, Art. no. 123.

[20] Y. Shen et al., "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, no. 7, pp. 441–446, 2017.

[21] X. Lin et al., "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.

[22] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[23] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, pp. 611–629, 2018.

[24] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, 2019.

[25] X. Xu et al., "11 TOPS photonic convolutional accelerator for optical neural networks," *Nature*, vol. 589, no. 7840, pp. 44–51, 2021.

[26] J. Chang, V. Sitzmann, X. Dun, W. Heidrich, and G. Wetzstein, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 12324.

[27] M. Y.-S. Fang, S. Manipatruni, C. Wierzynski, A. Khosrowshahi, and M. R. DeWeese, "Design of optical neural networks with component imprecisions," *Opt. Exp.*, vol. 27, no. 10, pp. 14009–14029, 2019.

[28] L. G. Wright et al., "Deep physical neural networks trained with backpropagation," *Nature*, vol. 601, no. 7894, pp. 549–555, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8791835/pdf/41586_2021_Article_4223.pdf

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[30] L. Bernstein, A. Sludds, C. Panuski, S. Trajtenberg-Mills, R. Hamerly, and D. Englund, "Single-shot optical neural network," *Sci. Adv.*, vol. 9, no. 25, 2023, Art. no. eadg7904.

[31] R. A. Athale and W. C. Collins, "Optical matrix–matrix multiplier based on outer product decomposition," *Appl. Opt.*, vol. 21, no. 12, pp. 2089–2090, 1982.

[32] W. Zhu, L. Zhang, Y. Lu, P. Zhou, and L. Yang, "Design and experimental verification for optical module of optical vector–matrix multiplier," *Appl. Opt.*, vol. 52, no. 18, pp. 4412–4418, 2013.

[33] T. Kumar and K. Verma, "A theory based on conversion of RGB image to gray image," *Int. J. Comput. Appl.*, vol. 7, no. 2, pp. 7–10, 2010.

[34] S. Bruch, X. Wang, M. Bendersky, and M. Najork, "An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retrieval*, 2019, pp. 75–78.

[35] W. Schiffmann, M. Joost, and R. Werner, *Optimization of the Backpropagation Algorithm For Training Multilayer Perceptrons*. Koblenz, Germany: Univ. Koblenz: Institute of Physics, 1994.

[36] Y. Zuo et al., "All-optical neural network with nonlinear activation functions," *Optica*, vol. 6, no. 9, pp. 1132–1137, 2019.

[37] T. Wang et al., "Image sensing with multilayer nonlinear optical neural networks," *Nature Photon.*, vol. 17, no. 5, pp. 408–415, 2023.

[38] M. Wuttig, H. Bhaskaran, and T. Taubner, "Phase-change materials for non-volatile photonic applications," *Nature Photon.*, vol. 11, no. 8, pp. 465–476, 2017.

[39] J. Liu et al., "Research progress in optical neural networks: Theory, applications and developments," *PhotoniX*, vol. 2, no. 1, pp. 1–39, 2021.