Check for updates

ADVANCED INTELLIGENT SYSTEMS Open Access
www.advintellsyst.com

Memory-Reduced Convolutional Neural Network for Fast Phase Hologram Generation

*Chenliang Chang, Chenzhou Zhao, Bo Dai, Qi Wang, Dawei Zhang, Songlin Zhuang, and Chao Ping Chen**

Computer-generated holography (CGH) has advanced the development of human-centric holographic near-eye displays. Recent work has proposed an end-to-end convolutional neural network that converts 2D images into 3D holograms, enabling real-time 3D holographic displays from widespread available images. However, the high computational cost of deep learning-based methods limits their practical application. Deploying such algorithms on resource-constrained mobile platforms requires more efficient models with reduced computational memory and power demands, which plays a crucial role in promoting human-centric virtual reality/augmented reality displays. In this article, a lightweight 3D hologram generation model is proposed using neural network quantization from the input of single 2D image. Specifically, a 2D-to-3D CGH model is quantized from 32-bit floating-point to 8-bit integer precision. The results show that the INT8 model reduces size by 60%, improves processing speed by a factor of three, and achieves comparable hologram quality to the FP32 model. This work enables the practical deployment of 2D-to-3D CGH model on low-power platforms, bridging the gap between high-performance holographic computation and real-world wearable display systems.

1. Introduction

Holographic 3D display technology,[1] which reconstructs 3D scenes encoded in wavefronts, has garnered significant attention for its ability to deliver an immersive visual experience. Its potential applications span various fields, including scientific visualization, medical imaging, and entertainment, particularly in the emerging metaverse, where creating realistic virtual environments is of paramount importance. With the rapid evolution of the metaverse and augmented reality/virtual reality (AR/VR) technologies, there is an increasing demand for immersive and realistic virtual environments.[2,3] In the metaverse, high-quality 3D holographic displays can significantly enhance immersive experiences by enabling realistic and natural depth perception. Additionally, in the field of medical imaging, holographic 3D visualization can assist in the analysis of computed tomography (CT) and magnetic resonance imaging (MRI) scans, allowing doctors to interact with volumetric data more intuitively for diagnostics and surgical planning. These applications highlight the potential of holographic displays in revolutionizing both virtual interaction and medical visualization. At the core of these holographic display systems lies the computation of computer-generated holograms. Computer-generated holography (CGH) is a technique that uses computational methods to render holograms for controlling light waves. One particularly promising application is AR displays.[4] CGH-based AR displays have been widely studied due to their ability to provide natural depth cues and simplify optical setups.[5–9]

One primary challenge is the tremendous computational cost required to perform diffraction simulation for holographic wavefront propagation. CGH requires numerical light propagation from a 3D scene to compute a hologram. This calculation is highly nonlocal, meaning that a single geometric primitive affects many hologram pixels. Therefore, the computational cost typically scales with the number of hologram pixels and the complexity of the 3D scene. Several schemes[10–15] have already been reported to further accelerate these diffraction-based algorithms are proposed to improve the CGH calculation speed from 3D point-cloud model. Nonetheless, it is still challenging to achieve real-time calculation using consumer hardware for high-resolution and dense sampled 3D models.

Recently, machine learning has become a hot topic in the field of computational imaging. Neural networks have gained attention for their potential to overcome the trade-off between computational cost and image quality in CGH. Deep learning approaches can be trained to understand the transformation

C. Chang, C. Zhao, B. Dai, Q. Wang, D. Zhang, S. Zhuang
School of Optical-Electrical and Computer Engineering
Engineering Research Center of Optical Instrument and System
Ministry of Education and Shanghai Key Laboratory of Modern Optics System
University of Shanghai for Science and Technology
Shanghai 200093, P. R. China

C. P. Chen
State Key Laboratory of Avionics Integration and Aviation System-of-Systems Synthesis
School of Integrated Circuits
Shanghai Jiao Tong University
Shanghai 200240, P. R. China
E-mail: ccp@sjtu.edu.cn

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202500354.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202500354

between object space and hologram, and then be used to infer holograms from given objects rapidly. Some studies has demonstrated that deep-learning-based CGH generating strategies[16–29] have the ability to significantly speed up of the runtime for real-time CGH generation as well as high-quality. Among them, one seminal study is called "TensorHolography",[30] which uses a convolutional neural network (CNN) to efficiently approximate point-based light propagation. However, this research required specialized equipment, such as a Tof camera and time-of-flight camera (LiDAR), to acquire depth information. Therefore, there is a problem of difficulty in obtaining RGB-D (RGB&Depth) images. Another approach involves predicting depth maps from 2D images using deep learning and subsequently generating 3D holograms. Liu et al.[31] proposed the DGE-CNN (depth gradient extracting) network for monocular depth estimation (MDE) and generated 3D CGH of a single image by layer-based angular spectral method. Ishii et al.[32] sequentially concatenated three CNNs for MDE, 3D CGH prediction, and quality optimization. However, these mentioned methods still need to acquire 3D depth information as a prior for the following CGH generation step, they just use MDE network to replace hardware devices for depth acquisition, and therefore, still follows a two-step process for hologram generation rather than an end-to-end solution. Therefore, the neural network framework of this kind of methods requires high computational resources and is not a sufficiently efficient solution.

This problem was previously addressed by the work "From Picture to 3D Hologram",[33] which proposed a data-driven CNN framework for real-time, end-to-end generation of photorealistic 3D CGHs from a single 2D image. While the method demonstrates strong robustness across natural scenes, it relies heavily on high-performance graphics processing units (GPUs), resulting in large system sizes and high power consumption. For high-resolution inputs, the fully convolutional architecture incurs substantial computational and memory costs due to the large number of convolution operations and feature maps. These factors make deployment on resource-limited platforms, such as AR headsets or automotive displays, challenging. To enable practical applications, a more lightweight and efficient solution is required. While the recent work "Configurable Learned Holography" by Zhan et al.[34] proposed a 2D-to-3D hologram generation framework and achieves nearly a twofold speedup through knowledge distillation, their method introduces several limitations for real-world deployment. First, the approach depends on a teacher-student training setup, where a large pretrained teacher network must guide a smaller student model. This adds significant training complexity and makes the student model inherently reliant on the teacher, preventing truly independent or plug-and-play use. Second, even after distillation, the student model typically retains 32-bit floating point precision (FP32) precision, which poses challenges for inference on low-power or edge devices. Many such devices do not support floating-point operations efficiently or at all, making INT8 quantization a more practical and scalable solution for deployment in resource-constrained environments.

Neural network quantization is a promising approach for creating lightweight models suitable for computationally constrained platforms.[35–37] Quantization involves converting the weights and activations of a model from a higher precision format, typically FP32, to a lower precision format, such as 8-bit integer precision (INT8). This technique helps reduce model size and memory consumption. Furthermore, quantized models can take advantage of low-precision algorithms, thereby accelerating computational speed. Although quantization has proven to be effective, its application in computational holography has not been thoroughly studied or widely adopted. A recent study proposed incorporating quantization methods into "Tensor Holography" to achieve a lightweight CGH generation scheme. However, this network still requires depth maps as prior information, and optimizing MDE networks for lightweight deployment remains a significant challenge. Therefore, bypassing depth information and applying quantization solely to a single end-to-end network for 2D-to-3D conversion presents a highly significant research perspective for achieving efficient and lightweight deployment.

In this article, we propose a lightweight quantized neural network specifically designed for 2D-to-3D CGH. To enable efficient deployment, we redesign the original 2D-to-3D hologram generation network[33] to fully support INT8 quantization by resolving architectural issues, such as the use of deconvolution layers, which are incompatible with current PyTorch quantization tools. Our improved model maintains comparable holographic quality to the original FP32 version while enabling full INT8 quantization. Notably, our model predicts complex holograms, which are then converted into phase-only holograms using a double-phase encoding technique for efficient reconstruction. This leads to a substantial reduction in model size and computational load, making real-time inference feasible on resource-constrained devices. Unlike previous methods, such as knowledge distillation, which rely on pretrained teacher models and retain FP32 precision, our approach directly produces an INT8 model without external dependencies. To the best of our knowledge, this is the first end-to-end quantized 2D-to-3D CGH framework that achieves high-quality hologram generation while being deployable on low-power hardware, thus offering a practical solution for real-world applications, such as AR headsets and embedded displays.

2. Experimental Section

2.1. 2D-to-3D Framework of Holographic Display through an End-to-End CNN

The conventional approach for achieving 3D holographic display from a single 2D image follows a "two-step" principle. First, an MDE network is employed to predict a depth map for the 2D image. Then, the combination of the 2D image and the depth map forms a 3D model, which is subsequently processed using a layer-based angular spectrum method to compute the final complex hologram. A schematic illustration of this process is shown in **Figure 1**a. However, such methods are computationally intensive and lack flexibility. For instance, MDE networks typically require substantial computational resources to generate depth maps. Additionally, during the computation of complex holograms, the processing time increases proportionally with the number of layers.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

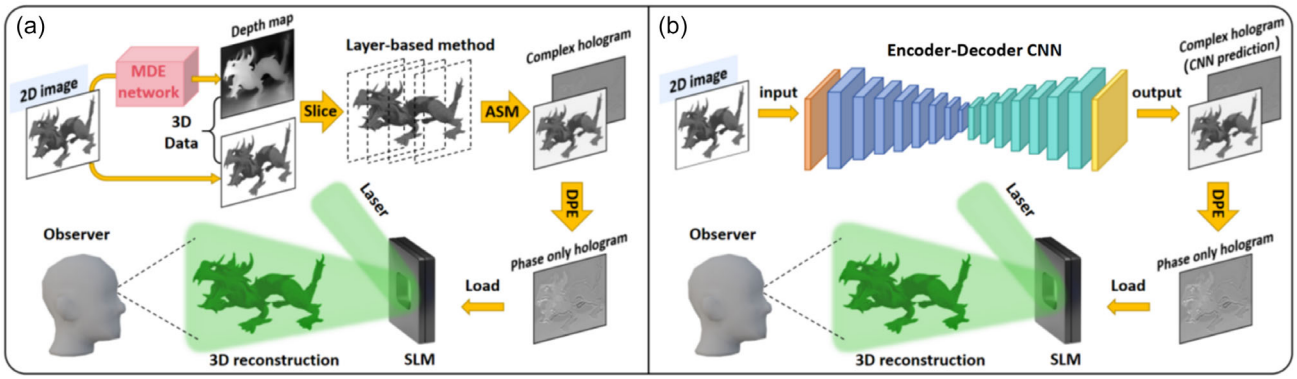

Figure 1. a) Schematic diagram of conventional method for 2D-to-3D holographic system; b) end-to-end CNN method for the 2D-to-3D holographic system. ASM: angular spectrum method; MDE: monocular depth estimator; DPE: double phase encoding. The 3D model "Asian Dragon" used in this figure is obtained from the Stanford 3D Scanning Repository: https://graphics.stanford.edu/data/3Dscanrep/.

The neural network can be trained to learn the process of converting a 2D image into a 3D wavefront, thereby enabling the establishment of an end-to-end CNN-based 2D-to-3D holographic display system. As illustrated in Figure 1b, the proposed method requires only a single 2D image as input to a pretrained CNN, which intelligently and automatically predicts a complex hologram containing 3D wavefront information. This end-to-end CNN completely replaces the functionality of the "MDE + layer-based angular spectrum method." After undergoing double-phase encoding, the complex hologram is transformed into a phase-only hologram, which is then loaded onto a spatial light modulator (SLM). As a result, the observer can perceive a 3D holographic display.

2.2. Model Architecture and Training

Building on the original 2D-to-3D framework,[33] we propose a refined CNN model to achieve neural network quantization and enhance its performance after quantization. As shown in **Figure 2**, our model is a multiscale fully convolutional residual network. The input 2D image is single-channel, and the output consists of two single-channel branches representing the amplitude and phase of the hologram, respectively. Within the network, each intermediate module maintains a fixed channel number of 32. Each residual block uses 3×3 convolutional kernels. The network consists of seven downsampling residual blocks and seven upsampling modules. Unlike original

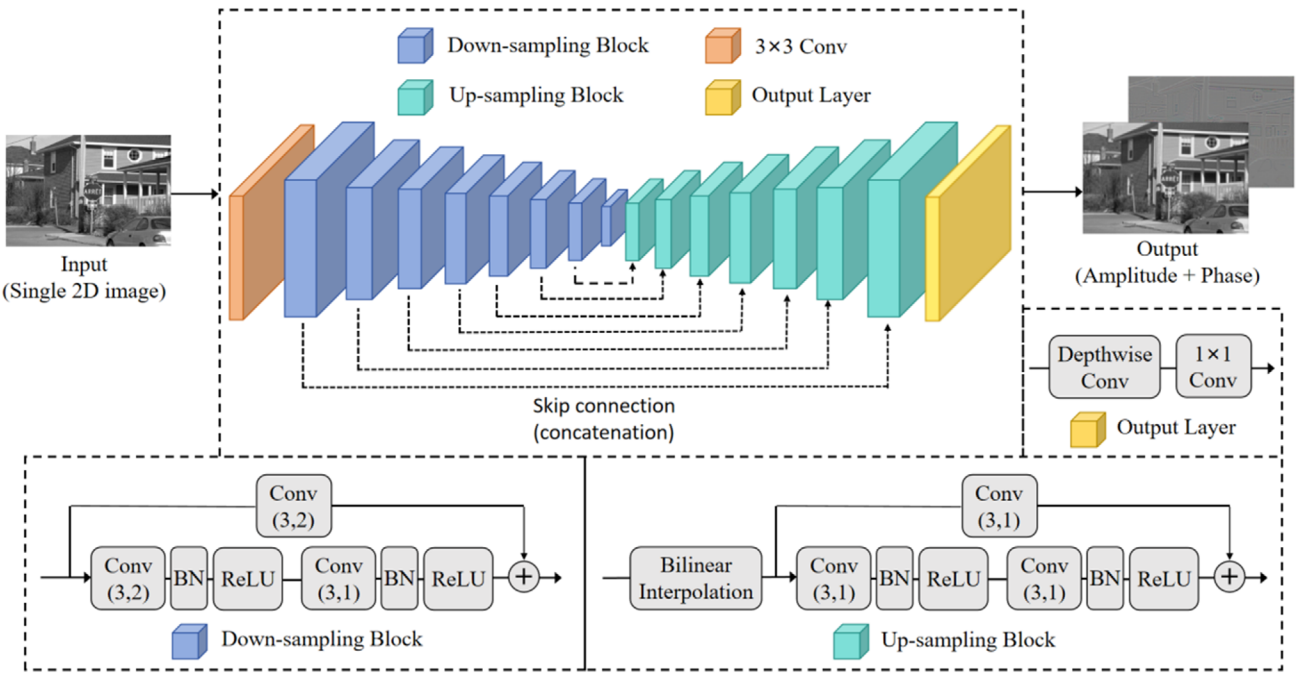

Figure 2. Detailed schematic structures of the refined end-to-end network.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2D-to-3D framework, our upsampling module has been modified in our research. In the original model, upsampling is performed using deconvolution residual modules, but deconvolution modules cannot be quantized in the current PyTorch framework. Therefore, in the refined version, we replace the original upsampling module with a "Bilinear Interpolation + Residual Block" structure to achieve the same upsampling operation without affecting the network's performance. The input image is filtered and passed through the downsampling blocks with reduced resolution. Each upsampling module uses skip connections from the activation outputs of the downsampling modules, enabling it to resolve higher-resolution details. In the output layer of the network, the amplitude and phase of the complex hologram are computed through output branches, which consist of depthwise separable convolution blocks and 1×1 pointwise convolutions. Finally, the output is constrained within the range [0, 1] by a Hardtanh activation function, fine-tuning and normalizing the output results. The depthwise convolutions in the output layer help maintain good performance in low-precision computing environments after quantization. Quantization alters the bit precision of the weights and activation values, and the depthwise convolution structure in the quantized model is better suited for low-precision multiplication-accumulation (MAC) operations. This adaptation contributes to increased computational speed and reduced model size, while minimizing the impact on the quality of the holograms. In the original model, the output uses the Sigmoid function, which involves exponential operations that are computationally expensive in resource-limited environments. In our refined version, we replace the Sigmoid function with the Hardtanh function because it has a lower computational cost and can be pruned by appropriately setting the clipping range, thereby aiding the efficient training of the neural network.

To maximize the learning capability of the CNN and adapt it to different testing domains, we generate our training dataset by randomly selecting 30 000 individual images from the popular COCO2017 dataset[38] of natural images. All images are resized to 1536×768 pixels, and we sequentially apply the monocular depth estimator and layer-based diffraction algorithm from Equation (1) and (2) to compute the corresponding ground truth complex holograms. We combine the 2D images with the generated 3D ground truth complex holograms to form a large-scale training dataset, which is then fed into the CNN for end-to-end supervised training using the mean squared error (MSE) loss function. The loss function is as follows

$$l = \sum_{n=1}^{N} [\mathrm{MSE}(a_n, \hat{a}_n) + \mathrm{MSE}(\varphi_n, \hat{\varphi}_n)] \tag{1}$$

where $a_n$and $\varphi_n$represent the n-th target amplitude and phase, and $\hat{a}_n$and $\hat{\varphi}_n$ represent the n-th predicted amplitude and phase. The MSE() function computes the MSE.In the ground truth hologram computation, we set the depth range of all the image scenes from $z = 0$ m (near plane) to $z = 0.012$ m (far plane). The calculation uses a wavelength of 520 nm, with a pixel spacing of 6.4 μm. Our CNN is trained on an NVIDIA GeForce RTX 4070 GPU platform using the Adam optimizer, with a batch size of 2 and 30 epochs. The learning rate starts at 0.0001 and is halved every 10 epochs.

2.3. Comparison Against Existing Works

Before proceeding with the quantization experiments, we first conduct a comparative study to evaluate whether our refined CNN architecture, in its full-precision (FP32) form, is sufficiently competitive with existing methods. This comparison aims to establish the effectiveness and representational capacity of our model design, serving as a solid foundation for the subsequent quantization work. It involves three models: the original 2D-to-3D framework,[33] our refined version, and Holobeam,[39] a representative prior work that utilizes an 18-channel U-Net to generate 3D holograms from single 2D images for AR near-eye display applications. We performed inference of complex holograms (amplitude and phase) using the FP32 precision parameter models of three CNNs on images from 20 different scenes. The outputs were compared with the ground truth, and the average peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) were calculated. As shown in **Figure 3**, our refined CNN achieves nearly the same quantitative performance as the original CNN, while producing hologram reconstructions of noticeably higher visual quality than those generated by Holobeam. These results demonstrate that the refined network not only preserves the accuracy and robustness of the original model but also delivers improved visual realism compared with existing methods. Therefore, it provides a strong foundation for subsequent quantization while maintaining superior reconstruction performance.

2.4. Neural Network Quantization

Quantization in neural networks is a technique used to reduce the bit precision of weights and activations. Specifically, quantization converts the FP32 format, commonly used for model parameters in most machine learning frameworks, into lower precision formats, such as 16-bit floating point, INT8, 4-bit integer, or even ternary or binary values. Reducing bit width has advantages in terms of model size and memory footprint. Moreover, if the processor supports high-throughput arithmetic pipelines for low-bit formats, the computational speed can be enhanced as well. However, reducing bit precision may lead to a loss in model accuracy, particularly for smaller models. Nevertheless, many neural network models are over-parameterized, meaning that with appropriate quantization, the bit width can be reduced without significantly compromising performance.

We first introduce quantization and dequantization operations for model weights and activations at the beginning and end of the feedforward process of the model. Here, we focus on uniform integer quantization, where a floating-point number x is mapped to a b-bit signed integer $xq \in [-2^{b-1}, 2^{b-1} - 1]$. The quantization function is defined as

$$Q(x) = \mathrm{round}\left(\frac{x}{S}\right) + Z \tag{2}$$

Here, the function round() rounds to the nearest integer, and S and Z are the quantization parameters known as the scaling factor and zero-point, respectively. The reversion from integer to floating-point is defined as

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

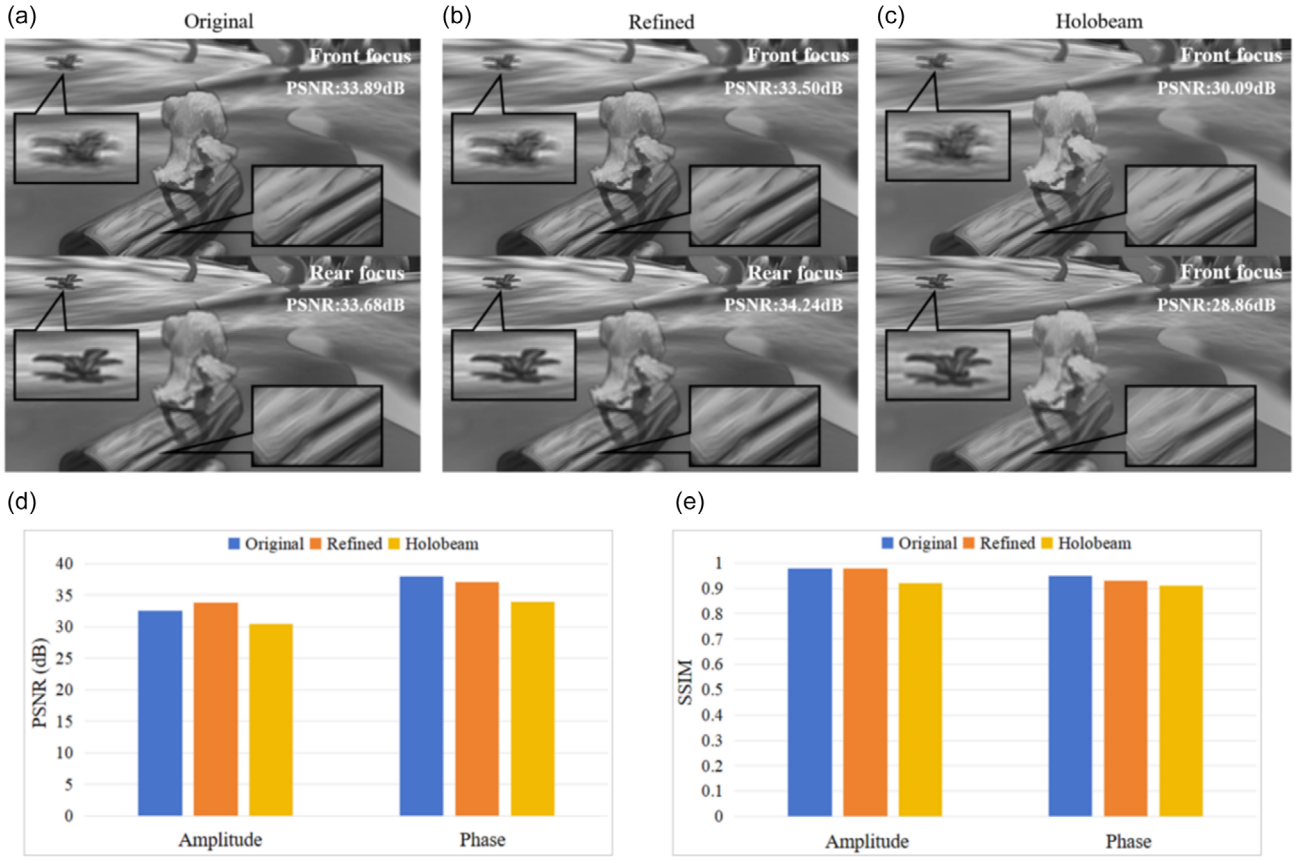

Figure 3. Performance comparison against existing works. a–c) Comparison of hologram reconstruction quality produced by three models: the original CNN, the refined CNN, and Holobeam.[39] The reconstructed images are located at 1 mm for front focus and 12 mm for rear focus, allowing for consistent visual assessment of depth fidelity and clarity. d,e) Quantitative evaluation of the three models in terms of average PSNR and SSIM, computed on complex holograms (amplitude and phase) across 20 test scenes. The refined CNN achieves nearly identical performance to the original model, while performing a bit better to Holobeam.

$$\tilde{Q}(xq) = (xq - Z) \times S \tag{3}$$

where S is determined by the ratio of the input and output ranges as

$$S = \frac{\beta - \alpha}{2^{b} - 1} \tag{4}$$

The range $[\alpha,\beta]$ represents the input clipping boundaries, a process known as calibration. In symmetric quantization, the zero-point Z is set to zero, while in asymmetric quantization, Z is nonzero. Asymmetric quantization uses

$$Z = -\text{round}\left(\frac{\alpha}{S}\right) - 2^{b-1} \tag{5}$$

Quantized weights and activations reduces the precision of MAC operations, which are computationally intensive in most neural networks. While the clipping range for weights can be calculated statically, the clipping range for activations varies for each input and can be determined either dynamically or statically. Dynamic quantization computes the range for each activation during inference and applies quantization accordingly. In contrast, static quantization relies on a representative dataset (i.e., a calibration dataset) to predetermine the activation ranges. While dynamic quantization can achieve higher accuracy than static quantization, it introduces additional computational overhead for calculating the clipping ranges. Moreover, many model deployment tools for embedded platforms only support static quantization models, not dynamic ones.

Post-training quantization (PTQ) and quantization-aware training (QAT) are two methods for model quantization. PTQ quantizes a pretrained FP32 model without additional fine-tuning, making it simple and fast, but it may reduce accuracy, especially with complex parameter distributions. In contrast, QAT retrains the FP32 model while considering quantization, leading to higher accuracy after quantization, though it requires more training time and resources. PTQ is ideal for quick deployment with minimal accuracy loss, while QAT is better for applications that demand higher accuracy despite the additional training effort.

We used multiple different approaches to quantify our FP32 model. **Figure 4** shows a schematic diagram of proposed end-to-end CNN using neural network quantization. The scheme includes post training dynamic quantization (PTDQ), post training static quantization (PTSQ), and static QAT, which are

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

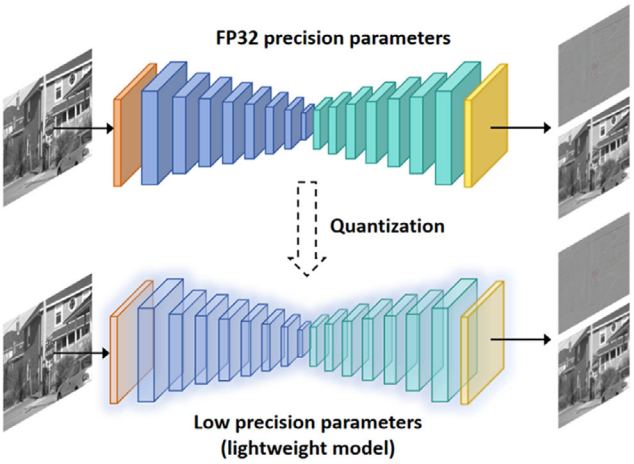

Figure 4. Schematic diagram of the quantization for the end-to-end CNN.

supported by the Pytorch quantization framework.[40] PTDQ and PTSQ are PTQ methods that use dynamic and static quantization, respectively. The target backend used in quantization is QNNPACK, which is set by default for calibration. In the PTSQ scheme, the calibration dataset we use consists of 100 samples from the training dataset. For QAT, we further trained the pre trained FP32 model as a preparatory model for 30 epochs, and statically quantified the weights and activations after training was completed.

3. Results

3.1. Simulation Verification of the CNN

In this section, a numerical simulation is conducted to evaluated the performance of our FP32 and quantized INT8 models in terms of output quality, model size, and speed. **Figure 5** shows the simulation results of different models with different quantization methods. For this purpose, we first selected a synthetic image from the SceneFlow dataset[41] as the test image, with a resolution of 1536×768. The preprocessed test image and the ground truth of complex hologram and reconstruction results are shown in Figure 5a. The simulation conditions are fixed, and the output holograms and reconstructed images of the algorithms are compared, as shown in Figure 5b–e. Figure 5b displays the hologram and reconstructed scene (both front and rear planes) quality for the nonquantized network (FP32 precision). Figure 5c–e correspond to the hologram and reconstructed scene (front and rear planes) quality for the quantized network (INT8 precision) using the PTDQ, PTSQ, and QAT quantization algorithms, respectively. Here, by comparing the PSNR and focal details amplification, it can be demonstrated that the INT8 models obtained through each quantization method produce high-quality results and maintain the accuracy of 3D reconstruction. The output quality after quantization remains comparable to that of the nonquantized version.

In the second set of tests, we selected 20 2D images from the test dataset to evaluate the impact of quantization. PSNR and SSIM were used to assess the quality of the amplitude and phase

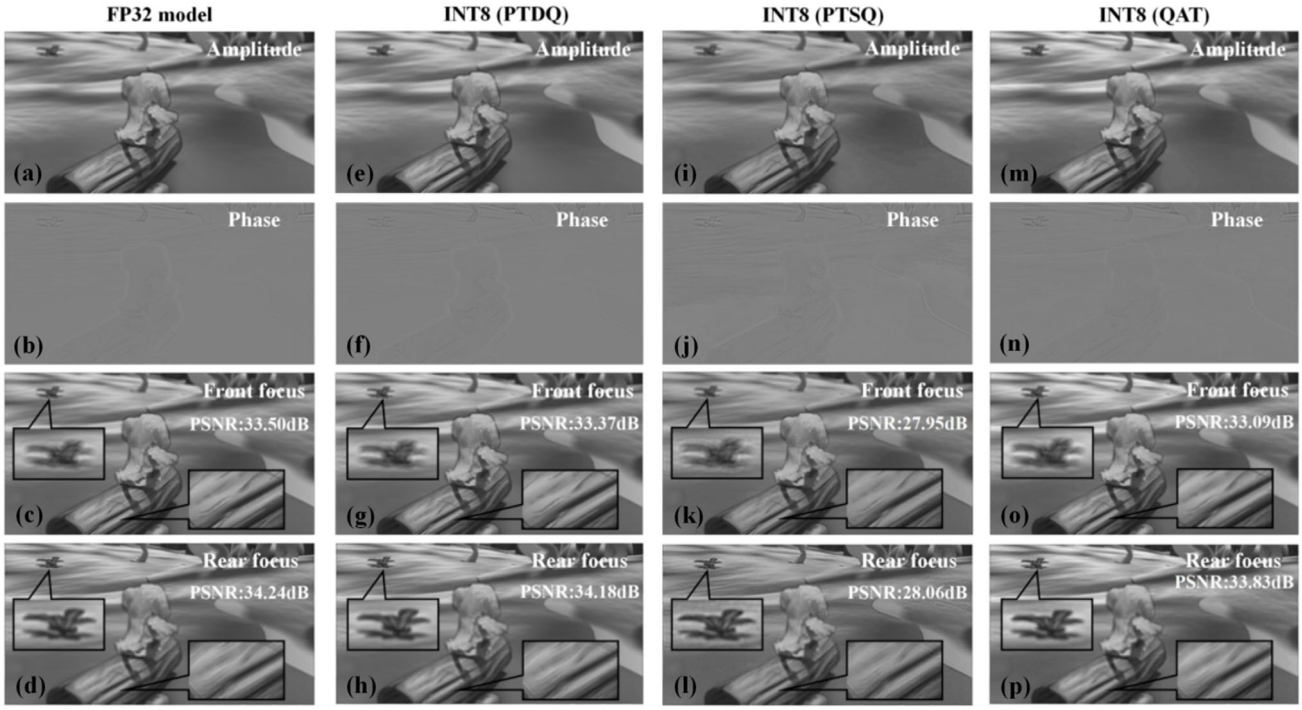

Figure 5. Simulation results of the FP32 model and INT8 models with different quantization methods. a–d) The FP32 model (non quantization), complex hologram, and reconstruction results; e–h) show the INT8 models, complex holograms, and reconstruction results for PTDQ. i–l) show the same for PTSQ, and m–p) are for QAT. The reconstructed images are located at 1 mm for the front focus and 12 mm for the rear focus.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

images of the complex holograms. **Figure 6** shows the average hologram quality for each model on the test dataset. The results indicate that our CNN can adapt well to different quantization methods, with PTDQ achieving output quality closest to that of the FP32 model. Although PTSQ shows a slight decline in accuracy, it does not exhibit a significant drop, and QAT significantly improves the performance of the INT8 model. In comparison, all INT8 models achieved hologram quality comparable to that of the FP32 model. We then compared the memory size and runtime on the CPU for the FP32 model and all INT8 models. The CPU used for testing was an Intel(R) Core(TM) i7-14700KF. The test results are shown in **Table 1**. Although PTDQ provides the accuracy closest to that of the FP32 model, it does not reduce the model size and has a long runtime. PTSQ achieves the smallest model size, reducing it to 33% of the FP32 model, and the runtime is three times faster. However, its accuracy is the lowest among the three INT8 models. On the contrary, QAT offers the best overall performance, reducing the model size by 60%, speeding up the runtime by more than twice, and maintaining performance comparable to that of the FP32 model.

Table 1. Comparison of the parameter precision, memory size, and runtime on the CPU for networks with different quantization methods.

Methods	Precision	Size [KB]	Runtime (CPU)	Runtime (CPU)
			Resolution: 1536 × 768 [ms]	Resolution: 2048 × 1024 [ms]
Nonquantization	FP32	1654	1283.6	1861.2
PTDQ	INT8	1654	964.5	1388.8
PTSQ	INT8	559	464.5	664.2
QAT	INT8	681	506.5	719.2

3.2. Optical Setup

Experimental verification is necessary to further concretely validate the proposed method of realizing from 2D image to 3D display through quantization network. **Figure 7** shows the experimental setup for optical 3D display using CNN-predicted holograms. To encode the complex hologram into a phase-only hologram (POH), an advanced double-phase encoding method was used.[42] A grating phase encoding is introduced to separate the zero-order information of the POH from the zero-order noise generated by the SLM, improving the image quality and contrast. The experiment employs a HOLOEYE PLUTO-2-VIS-014 reflective SLM with a resolution of 1920 × 1080 pixels and a pixel pitch of 6.4 μm to display phase-only CGH at a wavelength of 532 nm. The hologram is relayed through the 4f system. An aperture stop is placed about one focal length behind the first doublet (the Fourier plane) to pick up the desired diffracted signal as well as blocking the zero-order (DC) and higher-order diffractions. The reconstructed 3D images are further magnified by an additional eyepiece lens and finally recorded by a digital camera (DAHENG IMAGING MER2-2000-19U3C, 5496 × 3672, pixel pitch 2.4 μm) at the rear and far focus depths.

3.3. Optical Reconstruction

We first chose the quantized CNN obtained through QAT method for experimental verification. We selected various natural indoor and outdoor 2D images from the COCO Dataset,[38] Make3D Dataset,[43] and NYU Dataset.[44] Additionally, synthetic scene images were chosen from the animated movie "Big Buck Bunny".[45] These 2D images were input into the CNN that we trained, and corresponding 3D CGHs were generated for each image. **Figure 8** shows the experimental results of the reconstructed 3D images recorded at both the rear and far focal planes. The results confirm that our quantized-CNN-based 3D holography method provides accurate focusing cues for these 3D objects. Although the CNN was quantized to an INT8 low-precision format using the QAT method, it still demonstrated high performance in practical optical experiments.

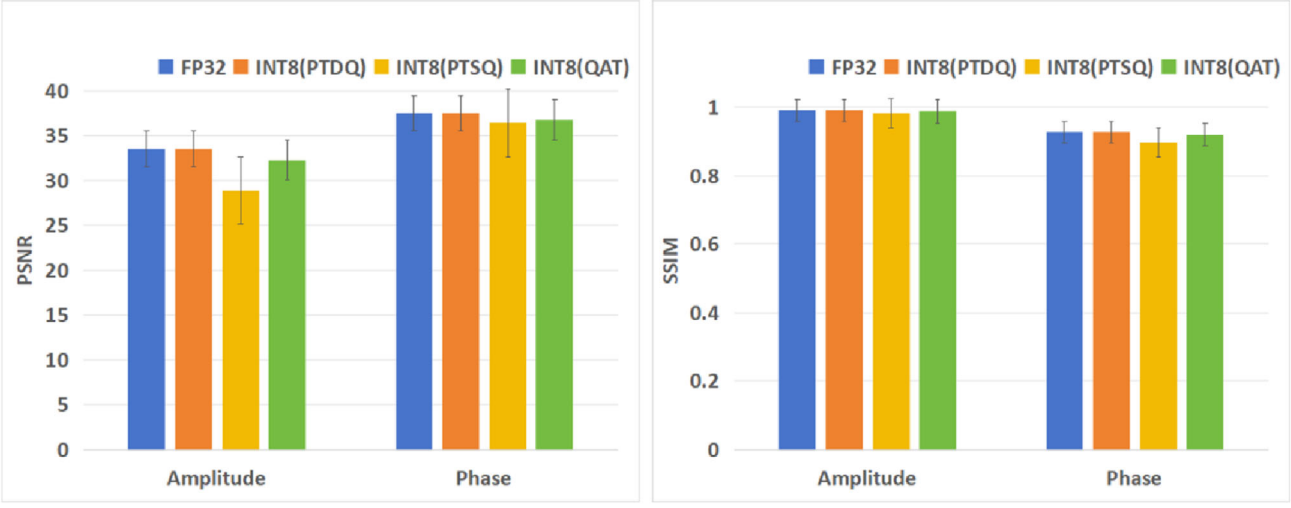

Figure 6. Hologram quality of our refined FP32 model and the INT8 models with PTDQ, PTSQ, and QAT. PSNR and SSIM are averaged over the test dataset.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

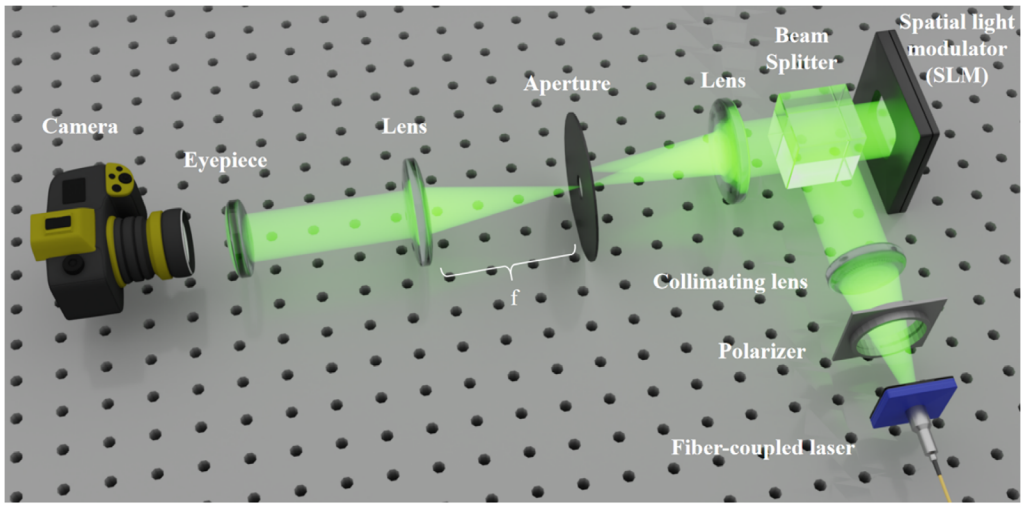

Figure 7. Schematic of the experimental holographic display prototype.

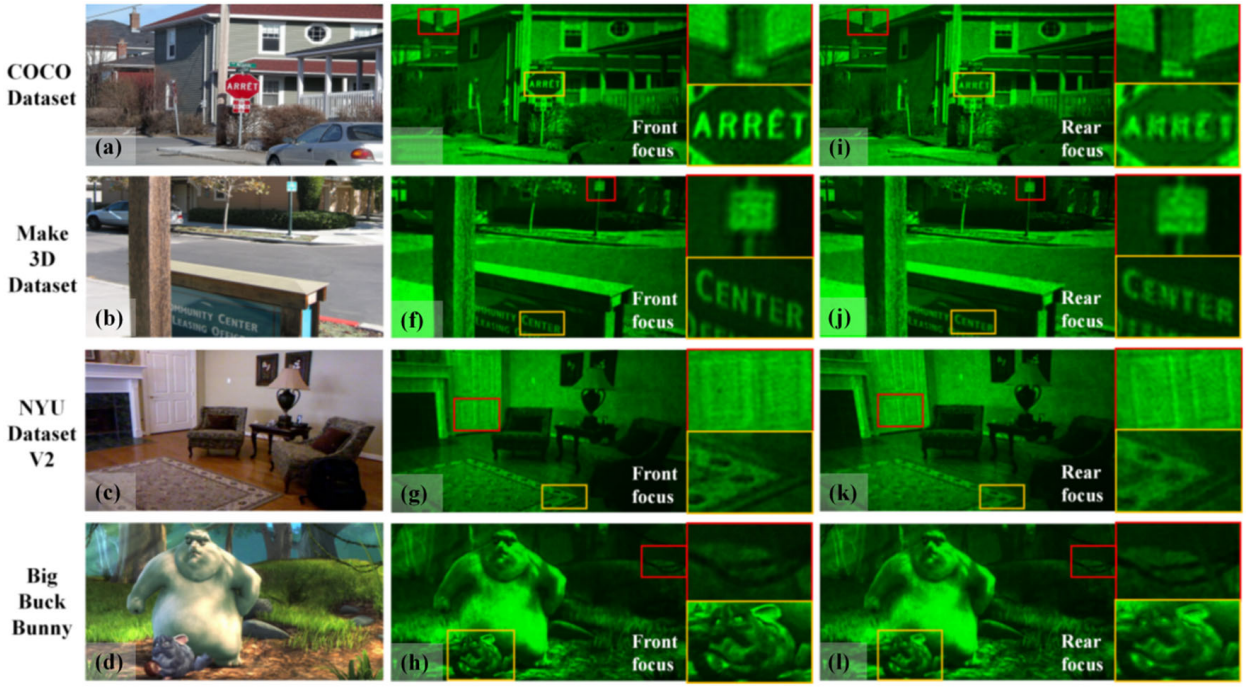

Figure 8. Holographic 3D display from the unquantized FP32 model and the INT8 model using different quantization methods (PTDQ, PTSQ, QAT). a–d) show each input 2D image from various datasets. e–h) show the experimental recordings of 3D reconstructions at the front focal distance ($z = 0.001$ m from the hologram). i–l) show the experimental recordings of 3D reconstructions at the rear focal distance ($z = 0.010$ m from the hologram).

The quality of the generated holograms remained high, and the optical reconstruction produced experimental results with nearly no speckle noise.

Additionally, we selected several 2D images of synthetic scenes from the SceneFlow dataset[41] for further experimental validation. The images were input into the INT8 models obtained by each quantization method (PTDQ, PTSQ, QAT), and the output complex holograms were encoded into POHs. We compared the experimental results with those obtained from the unquantized FP32 model. The 3D optical reconstructed results are shown in **Figure 9**. Each INT8 model generated holograms that, upon optical reconstruction, were able to focus at the corresponding positions, consistent with the simulation results. The quality of the reconstructions was comparable to that of the FP32 model. Among them, the experimental results from the PTDQ method were closest to the FP32 model, while PTSQ and QAT also produced absolutely high-quality results. This demonstrates that our CNN can achieve lightweight models using any quantization method, tailored to specific needs, and exhibits strong generalization performance.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

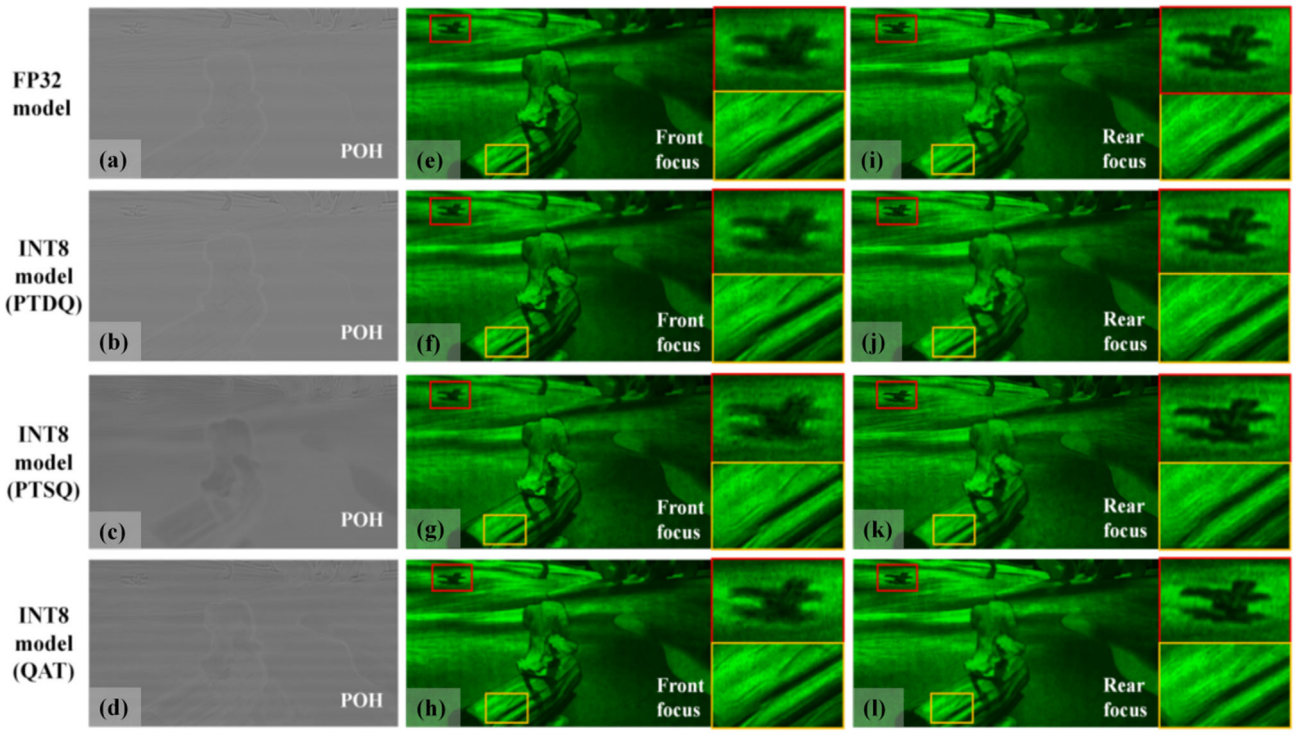

Figure 9. Holographic 3D display from the unquantized FP32 model and the INT8 model using different quantization methods (PTDQ, PTSQ, QAT). a–d) show each POH from different models. e–h) show the experimental recordings of 3D reconstructions at the front focal distance ($z = 0.001$ m from the hologram). i–l) show the experimental recordings of 3D reconstructions at the rear focal distance ($z = 0.010$ m from the hologram).

4. Discussion

4.1. Synthesize of Dataset for End-to-End CNN Training

Our end-to-end CNN is trained under a supervised manner. To effectively train the neural network, we pregenerate a large dataset, where each training sample consists of a 2D image paired with a corresponding complex hologram containing 3D wavefront information as the ground truth. To generate 3D holograms from 2D data, we employ the state-of-the-art pretrained monocular depth estimator network, Depthanything,[46] to first predict the depth map of each training image. Then, in the synthesis of the label data from the training dataset, the depth maps predicted by Depthanything are aligned pixel-by-pixel with the grayscale version of the input 2D images to generate dense 3D volumetric light wave data for monochromatic CGH calculation. Based on the point-cloud diffraction model, the complex wavefront of a specific hologram $H(xm, yn)$, contributed by all points of the volumetric object, is given by the following equation

$$H(x_m, y_n) = \sum_i A(x_i, y_i) \times \exp\left[\frac{i2\pi}{\lambda}\sqrt{(x_i - x_m)^2 + (y_i - y_n)^2 + d(x_i, y_i)^2} + i\varphi_0\right] \tag{6}$$

In this equation, $A(x_i, y_i)$ represents the grayscale amplitude associated with the image pixel at (x_i, y_i) corresponding to the predicted depth $d(x_i, y_i)$ from Depthanything, $A(xi, yi)$ is the wavelength, and φ_0 is the initial phase used to achieve smooth phase contours based on position. To further accelerate the computation of Equation (1), we divide the image content into 256 depth layers based on the 8-bit depth map $d(x, y)$, and then compute the ground truth CGH from each layered image using angular spectrum propagation as

$$H(x_m, y_n, \lambda) = \sum_{d_i} F^{-1}(F(A(x_i, y_i, \lambda) \times e^{i\varphi_0(x_i, y_i)}) \times H_{d_i}(f_x, f_y, \lambda)),$$

$$H_{d_i}(f_x, f_y, \lambda) = \begin{cases} e^{i\frac{2\pi}{\lambda}\sqrt{1-(\lambda f_x^2)-(\lambda f_x^2)}d_i}, & \text{if } \sqrt{f_x^2 + f_y^2} < \frac{1}{\lambda}, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where f_x and f_y represent the spatial frequencies, with F and F^{-1} denoting the 2D Fourier transform and its inverse Fourier transform operators, respectively. Additionally, $d_i (i = 1,2,...,256)$ represents the distance from each layer to the CGH. The generation of ground truth for 3D CGH is a crucial step in CNN training. The MDE + layer-based ASM method we employ enables the generation of a large number of 3D complex hologram labels from 2D data, providing strong data support for the training of our CNN.

4.2. Training and Quantization Details

Our CNN was trained using Pytorch on a platform of NVIDIA GeForce RTX 4070Ti GPU using Adam optimizer with a batch

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

size of 2 and 30 epochs. The learning rate starts at 0.0001 and is decreased by half every 10 epochs. The whole training process took ≈25 h with 30 000 datasets. To quantify the complexity of our model, we calculated the total parameter count of the FP32 version, which amounts to ≈386 598 parameters. The tensor of input image and output amplitude are normalized to the interval of [0, 1], while the phase tensor is normalized to the interval of $[-\pi, \pi]$. To simulate the performance of the quantized CNN on resource-constrained platforms, we conducted performance testing exclusively on a CPU instead of a GPU. The CPU used for testing was an Intel(R) Core(TM) i7-14700KF. The target backend used in quantization is QNNPACK. In the PTSQ method, the validation dataset consists of 100 images that are not included in the training dataset. The calibration process used 300 samples and took ≈1 h. The training details of QAT are identical to those of the FP32 model, with training conducted for 30 epochs. The quantized model (INT8) training required ≈8 h. The model size is measured by the memory size of the exported model file after quantization, which can be viewed through file properties on a computer. It generally reflects both the number of parameters and their precision, such as INT8 or FP32.

4.3. Potential Limitations of This Work

The proposed model adopts a unified 2D-to-3D framework that directly maps 2D images to 3D CGHs, offering a simple and efficient solution. However, this design imposes certain input constraints: the network is tailored specifically for 2D image inputs and does not support cases where depth information is already available. As a result, it lacks flexibility in scenarios involving preprocessed or structured depth data, which could otherwise enhance hologram quality or enable more diverse applications. In addition, while our model achieves high-quality results in simulations, the optical reconstruction quality in real-world experiments still has room for improvement. Future work could incorporate camera-in-the-loop[17] optimization to better bridge the gap between simulated and physical reconstructions, further enhancing practical performance.

5. Conclusion

In this study, we proposed and developed an INT8 quantization method for an end-to-end CNN-based 2D-to-3D hologram generation model to reduce both model size and computation time. Due to compatibility issues with existing quantization frameworks, the original model encountered unsupported modules, prompting us to introduce architectural improvements. The optimized INT8 model achieved hologram quality comparable to that of the FP32 model. Performance evaluations demonstrated that quantization effectively reduced both model size and execution time. Additionally, high-quality 3D reconstructions based on the quantized CNN INT8 model were successfully achieved in optical experiments. We believe this work paves the way for implementing neural network-based 2D-to-3D computational holography on resource-constrained platforms and provides valuable insights for deploying such models on a wider range of devices, including mobile platforms, in the future.

Acknowledgements

This work is supported by Science and Technology Commission of Shanghai Municipality (24511106500); Youth Innovation Promotion Association, Chinese Academy of Sciences (2022232); National Natural Science Foundation of China (62075040); National Key Research and Development Program of China (2021YFF0701100); Ministry of Industry and Information Technology of China (GO0300164/001); and Natural Science Foundation of Chongqing Municipality (cstb2023nscq-msx0465).

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

8-bit quantization, computer-generated holography, convolutional neural network, deep learning

Received: March 28, 2025
Revised: August 28, 2025
Published online: September 13, 2025

[1] C. P. Chen, Y. Su, C. G. Jhun, *J. Opt. Photonics* **2014**, *1*, 1.
[2] J. Orlosky, M. Sra, K. Bektas, H. Peng, J. Kim, N. Kos'myna, T. Hollerer, A. Steed, K. Kiyokawa, K. Aksit, *Front. Virtual Real.* **2021**, *2*, 763340.
[3] H. Hu, C. P. Chen, G. Li, Z. Jin, Q. Chu, B. Han, S. P. Zou, *ACS Appl. Opt. Mater.* **2024**, *2*, 1269.
[4] C. P. Chen, X. Ma, S. P. Zou, T. Liu, Q. Chu, H. Hu, Y. Cui, *Displays* **2023**, *81*, 102582.
[5] K. Wakunami, P.-Y. Hsieh, R. Oi, T. Senoh, H. Sasaki, Y. Ichihashi, M. Okui, Y.-P. Huang, K. Yamamoto, *Nat. Commun.* **2016**, *7*, 12954.
[6] A. Maimone, A. Georgiou, J. S. Kollin, *ACM Trans. Graph.* **2017**, *36*, 85.
[7] C. Jang, K. Bang, G. Li, B. Lee, *ACM Trans. Graph.* **2018**, *37*, 195.
[8] G. A. Koulieris, K. Aksit, M. Stengel, R. K. Mantiuk, K. Mania, C. Richardt, *Comput. Graph. Forum* **2019**, *38*, 493.
[9] C. Chang, K. Bang, G. Wetzstein, B. Lee, L. Gao, *Optica* **2020**, *7*, 1563.
[10] T. Shimobaba, T. Ito, *Opt. Express* **2017**, *25*, 77.
[11] M. Lucente, *J. Electron. Imaging* **1993**, *2*, 28.
[12] T. Shimobaba, N. Masuda, T. Ito, *Opt. Lett.* **2009**, *34*, 3133.
[13] T. Shimobaba, H. Nakayama, N. Masuda, T. Ito, *Opt. Express* **2010**, *18*, 19504.
[14] A. Symeonidou, D. Blinder, A. Munteanu, P. Schelkens, *Opt. Express* **2015**, *23*, 22149.
[15] D. Arai, T. Shimobaba, K. Murano, Y. Endo, R. Hirayama, D. Hiyama, T. Kakue, T. Ito, *Opt. Express* **2015**, *23*, 1740.
[16] R. Horisaki, R. Takagi, J. Tanida, *Appl. Opt.* **2018**, *57*, 3859.
[17] Y. Peng, S. Choi, N. Padmanaban, G. Wetzstein, *ACM Trans. Graph.* **2020**, *39*, 185.
[18] S. Choi, M. Gopakumar, Y. Peng, J. Kim, G. Wetzstein, *ACM Trans. Graph.* **2021**, *40*, 240.
[19] J. Lee, J. Jeong, J. Cho, D. Yoo, B. Lee, B. Lee, *Opt. Express* **2020**, *28*, 27137.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[20] R. Horisaki, Y. Nishizaki, K. Kitaguchi, M. Saito, J. Tanida, *Appl. Opt.* **2021**, *60*, A323.
[21] H. Zheng, J. Hu, C. Zhou, X. Wang, *Photonics* **2021**, *8*, 280.
[22] K. Liu, J. Wu, Z. He, L. Cao, *Opto-Electron. Adv.* **2023**, *6*, 220135.
[23] X. Shui, H. Zheng, X. Xia, F. Yang, W. Wang, Y. Yu, *Opt. Express* **2022**, *30*, 44814.
[24] L. Shi, B. Li, W. Matusik, *Light Sci. Appl.* **2022**, *11*, 247.
[25] Z. Jin, Q. Ren, T. Chen, Z. Dai, F. Shu, B. Fang, Z. Hong, C. Shen, S. Mei, *Opt. Express* **2024**, *32*, 14394.
[26] C. Chang, D. Wang, D. Zhu, J. Li, J. Xia, X. Zhang, *Opt. Lett.* **2022**, *47*, 1482.
[27] Z.-S. Li, C. Liu, X.-W. Li, Y. Zheng, Q. Huang, Y.-W. Zheng, Y.-H. Hou, C.-L. Chang, D.-W. Zhang, S.-L. Zhuang, D. Wang, Q.-H. Wang, *Light Sci. Appl.* **2025**, *14*, 74.
[28] S. M. Imtiaz, T. Amgalan, F. M. F. Hossain, R. K. Shinde, H. Kang, G. Baasantseren, K.-C. Kwon, N. Kim, *Neurocomputing* **2025**, *631*, 129672.
[29] L. Yang, S. Xu, C. Yang, C. Chang, Q. Hou, Q. Song, *Opt. Lasers Eng.* **2024**, *184*, 108704.
[30] L. Shi, B. Li, C. Kim, P. Kellnhofer, W. Matusik, *Nature* **2021**, *591*, 234.
[31] N. Liu, Z. Huang, Z. He, L. Cao, *Opt. Express* **2023**, *31*, 23867.
[32] Y. Ishii, F. Wang, H. Shiomi, T. Kakue, T. Ito, T. Shimobaba, *Opt. Lasers Eng.* **2023**, *170*, 107758.
[33] C. Chang, B. Dai, D. Zhu, J. Li, J. Xia, D. Zhang, L. Hou, S. Zhuang, *Opt. Lett.* **2023**, *48*, 851.
[34] Y. Zhan, L. Shi, W. Matusik, Q. Sun, K. Aksit, *arXiv preprint* **2024**, arXiv:2405.01558.
[35] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, in *31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA **2018**, pp. 2704–2713.
[36] H. Wu, P. Judd, X. Zhang, M. Isaev, P. Micikevicius, *arXiv preprint* **2020**, arXiv:2004.09602.
[37] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, K. Keutzer, in *A Survey of Quantization Methods for Efficient Neural Network Inference* (Eds: G. K. Thiruvathukal, Y.-H. Lu, J. Kim, Y. Chen, B. Chen), Computer Science Chapman and Hall/CRC, New York **2022**), eBook ISBN 9781003162810, https://doi.org/10.1201/9781003162810-13.
[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C. L. Zitnick, in *13th European Conference on Computer Vision (ECCV)*, Zurich, Switzerland **2014**, pp. 740–755.
[39] K. Aksit, Y. Itoh, in *30th IEEE Conference Virtual Reality and 3D User Interfaces (IEEE VR)*, Shanghai, China **2023**, pp. 581–591.
[40] Python API, Quantization, https://pytorch.org/docs/stable/quantization (accessed: August 2025).
[41] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA **2016**, pp. 4040–4048.
[42] Y. Qi, C. Chang, J. Xia, *Opt. Express* **2016**, *24*, 30368.
[43] A. Saxena, M. Sun, A. Y. Ng, *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 824.
[44] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, in *12th European Conference on Computer Vision (ECCV), Florence, Italy* **2012**, pp. 746–760.
[45] Blender, Big Buck Bunny, https://peach.blender.org (accessed: August 2025).
[46] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, H. Zhao, *arXiv preprint* **2024**, arXiv: 2406.09414v1.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2026, 3, Downloaded from https://advanced.onlinelibrary.wiley.com/doi/10.1002/aisy.202500354 by Shanghai Jiaotong University, Wiley Online Library on [18/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License