

Full length article

DenSFA-PU: Learning to unwrap phase in severe noisy conditions

Muhammad Awais^a, Taeil Yoon^b, Chi-Ok Hwang^c, Byeongha Lee^{a,*}^a School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Korea^b Department of Physics, Korea University, Seoul 02855, Korea^c Department of Mathematical Sciences, Natural Science College, Gwangju Institute of Science and Technology, Gwangju 61005, Korea

ARTICLE INFO

Keywords:

Phase unwrapping
Digital holography
Deep learning
Densely connected neural networks
Encoder-decoder architecture
Spatial feature aggregator

ABSTRACT

In optics, phase measurement techniques face challenges because phase values are confined within 2π , leading to the problem of phase unwrapping. Many methods, including deep learning-based approaches, have been proposed to address this issue. However, high noise in a wrapped phase image often causes these techniques to fail, resulting in error accumulation and high computation time. To overcome these challenges, we propose a robust and fast deep learning-based method called DenSFA-PU (Densely Connected Spatial Feature Aggregator for Phase Unwrapping), which treats this problem as a regression task. Our approach uses an encoder-decoder architecture with densely connected neural networks and a spatial feature aggregator module for noise reduction and robust feature representation. Comparative analysis using both synthetic data and real-world data obtained through digital holography demonstrates that our method outperforms existing techniques, achieving greater computational efficiency with an average unwrapping time of 29.31 ms, significantly faster than other methods. It also shows superior accuracy, with consistently good NRMSE, PSNR, and SSIM values across all cases, highlighting its robustness in handling highly noisy wrapped phase images. Additionally, its ability to operate with minimal training data makes it highly suitable for the applications requiring fast and accurate phase unwrapping with a limited data set.

1. Introduction

Extracting phase information is essential in various fields such as synthetic aperture radar interferometry [1], digital holography [2], and magnetic resonance imaging [3]. However, direct phase measurement over a wide range is usually not feasible. Typically, phase is wrapped within a limited range, like $-\pi$ to π or 0 to 2π . Therefore, phase unwrapping is necessary to obtain the true phase distribution across an object, ensuring smooth continuity without abrupt jumps. This process poses significant challenges in real-world applications, requiring the development of highly robust algorithms [4,5].

In optics, phase information is typically presented as a two-dimensional (2D) image or map. A true phase map is typically represented as a continuously varying function along any line on the map. While the true phase can be recovered from the wrapped one under ideal conditions, challenges arise in complex situations involving noise, abrupt surface variations, and imaging artifacts. Conventional noise filtering-based techniques can negatively affect phase jumps and cause phase shift errors in the subsequent unwrapping process [6]. Since phase

unwrapping is a nonlinear process, the true phase is not uniquely determined. Generally, the unwrapping involves detecting phase jumps between nearby pixels and correcting them by adding or subtracting multiples of 2π . In the path-following techniques, for instance, unwrapping decisions are made by minimizing the total phase differences along chosen paths [7]. Although these methods have proven effective in many scenarios, they face challenges in complex situations and often require high computation time. Additionally, small residual errors can propagate and lead to significant accumulation of defects [8–12]. These limitations highlight the need for more robust and efficient methods, such as deep learning-based ones.

Phase unwrapping has recently emerged as a significant application area for deep learning, where Convolutional Neural Networks (CNNs) [13] trained with large datasets have shown promising results in both accuracy and speed. These methods, particularly deep regression [14], directly map a wrapped phase image to its true phase, simplifying the process by eliminating the need for complex pre- and post-processing. In contrast, the wrap count method estimates the wrap count for each pixel using CNNs, treating phase unwrapping as a classification or

* Corresponding author.

E-mail address: leebh@gist.ac.kr (B. Lee).<https://doi.org/10.1016/j.optlastec.2025.112757>

Received 16 October 2024; Received in revised form 7 February 2025; Accepted 3 March 2025

Available online 12 March 2025

0030-3992/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

segmentation task [15–24], but it adds complexity with the need for pre- and post-processing. Wang et al. [25] introduced a regression approach that excelled in noise reduction. Dardikman-Yoffe et al. [26] developed a deep residual CNN for analyzing biological cells in aqueous media. Spoorthi et al. [27] enhanced wrap count prediction accuracy by incorporating prior knowledge of absolute phase values and gradients into a loss function. Qin et al. [28] integrated residual blocks throughout the Res-UNet architecture, reporting significant benefits despite high computational costs. Perera et al. [29] proposed a joint convolutional encoder-decoder architecture for phase unwrapping. Park et al. [30] evaluated the use of Generative Adversarial Networks (GANs), which Zhou et al. [31] employed for InSAR phase unwrapping. Zhou et al. [32] also improved the stability and effectiveness of deep learning-based phase unwrapping. Xu et al. [33] increased the accuracy and robustness by adding skip connections to the Res-UNet and using a composite loss function.

Another study highlighted the importance of accurate cell phase recovery in biological sample analysis, showcasing the effectiveness of GAN-based learning models [34]. Gontarz et al. [35] introduced a method combining noise removal and phase unwrapping, using semantic segmentation to describe regions with phase discontinuities. Zhao et al. [36] proposed VDE-Net, which uses a weighted jump-edge attention mechanism to focus on jump regions in a wrapped phase pattern. He et al. [37] introduced UN-PUNet, an encoder-decoder network designed to handle unwrapping in uneven and noisy patterns, featuring a dual-branch encoder, multi-scale fusion, and a CBAM module. Automated phase unwrapping in digital holography is enhanced using a Pix2Pix GAN for fast and accurate reconstruction of unwrapped phase images with biological samples [30]. Zhang et al. [38] introduced U²-Net, a nested U-shaped network with 11 residual U-blocks, enhancing accuracy, noise resistance, and generalization. Li et al. [39] proposed an attention-based DeepLabV3 + network for interferogram unwrapping. SFNet [40] and C-HRNet [42] utilized self-attention mechanisms and contextual representation for error minimization. Other models include Residual U-Net combinations [41], deformable convolution-based PUDCN [43], and a Hybrid CNN-Vision Transformer network [44].

Although these deep learning-based methods offer significant advantages, they face challenges such as the need for large and diverse training datasets which are often hard to obtain in real-world scenarios. There might be poor generalization and inaccuracy when the model encounters unseen data. Moreover, many models struggle to capture feature relationships effectively during local convolution and pooling, limiting their understanding of the overall image. Additionally, most models are optimized for Gaussian noise, thus often fail in real-world scenarios like digital holography where speckle noise is common.

To address these challenges, we introduce a robust and efficient network named DenSFA-PU, which stands for **D**ensely Connected **S**patial Feature Aggregator for **P**hase **U**nwrapping. The network utilizes an encoder-decoder architecture, where one convolution layer and four Dense blocks with transition layers' act as the encoder and a simple CNN decoder is used for accurate phase reconstruction. In addition, a spatial feature aggregator (SFA) module serves as a bridge between the encoder and decoder. The SFA module consists of bi-directional long short-term memory (bi-LSTM) [45] layers, two convolutional layers, and a lightweight bottleneck attention module (BAM) [46], which together capture long-range dependencies and enhance feature refinement. This approach enables accurate and rapid phase unwrapping without requiring extensive training on large datasets.

Our key contributions are as follows;

1. We propose DenSFA-PU, a simple and reliable method that integrates Densely Connected Convolutional Networks (DenseNets) [47], bi-LSTM, and BAM to enable simultaneous denoising and phase unwrapping.

2. Using DenseNets as the encoder allows robust feature extraction through densely connected layers, improving feature reuse and reducing parameter counts.
3. Combining bi-LSTM and BAM as a spatial feature aggregator module captures both sequential dependencies and global context, allowing the network to focus on key features, reduce noise, and improve accuracy in noisy conditions.
4. We employ the composite loss function that combines mean squared error with variance and total variation loss, minimizing prediction errors while maintaining smoothness.

Both Gaussian and speckle noise are included in the experiments. To evaluate the performance of the proposed method, tests are conducted on synthetic data and real data obtained from a homemade digital holography setup. Comparisons are also made with conventional methods and the latest deep regression methods, which are directly comparable to DenSFA-PU. For quantitative evaluation, three metrics are used: normalized root mean square error (NRMSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM). Additionally, ablation studies are performed to assess the impact of different loss functions and explore the influence of integrating BAM with the bi-LSTM module.

2. Proposed methodology

Phase unwrapping is inherently an ill-posed inverse problem, and conventional methods typically address it by estimating discrete wrap counts [48]. In contrast, the phase unwrapping can be treated as a regression task, predicting a continuous output phase map directly. Our regression-based deep learning model utilizes a neural network to directly learn the function (F) mapping between the input-wrapped phase (X) and the predicted output-unwrapped phase ($\hat{Y}$) at a point (x, y) as:

$$\hat{Y}(x, y) = F(X(x, y)) \quad (1)$$

In general, the wrapped phase can be simply computed from the true phase (Y) as:

$$X(x, y) = \tan^{-1} \left\{ \frac{\text{Im}[Y(x, y)]}{\text{Re}[Y(x, y)]} \right\} \quad (2)$$

2.1. Dataset preparation

The data set for training is generated with a linear ramp and multiple Gaussian functions having different offsets (μ) and variances (σ) as:

$$Y(x, y) = m_1x + m_2y + C + \sum_{n=1}^N A_n \exp \left[- \left(\frac{(x - \mu_x)^2}{2\sigma_x^2} + \frac{(y - \mu_y)^2}{2\sigma_y^2} \right) \right] \quad (3)$$

Where m_1 and m_2 are the linear coefficients representing the phase changes along the x and y -axes, respectively. The values of m_1 and m_2 are randomly selected within the range 0 to 0.5. The ranges of x and y are from -128 to 127 , and C is a constant representing the DC phase, randomly chosen from the range 1 to 10. The number of Gaussian functions (N) varies between 1 and 20 with the offsets randomly selected between 2 and 127. The variances are chosen from

100 to 1000 range, and the amplitude A of each Gaussian is randomly set between 0 and 1. Gaussian noise with varying signal-to-noise ratio (SNR) and speckle noise with different variance are applied to the true phase image. These random variations in Gaussian parameters and noise result in unpredictable and complex arbitrary patterns. Fig. 1 illustrates an example of a true phase image generated using Eq. (3) and the corresponding wrapped image with intentional noise. In this example, the values of m_1 , m_2 and C were 0.146, 0.149, and 2, respectively. Speckle noise (variance = 0.01) and Gaussian noise (SNR = 0) were added to the

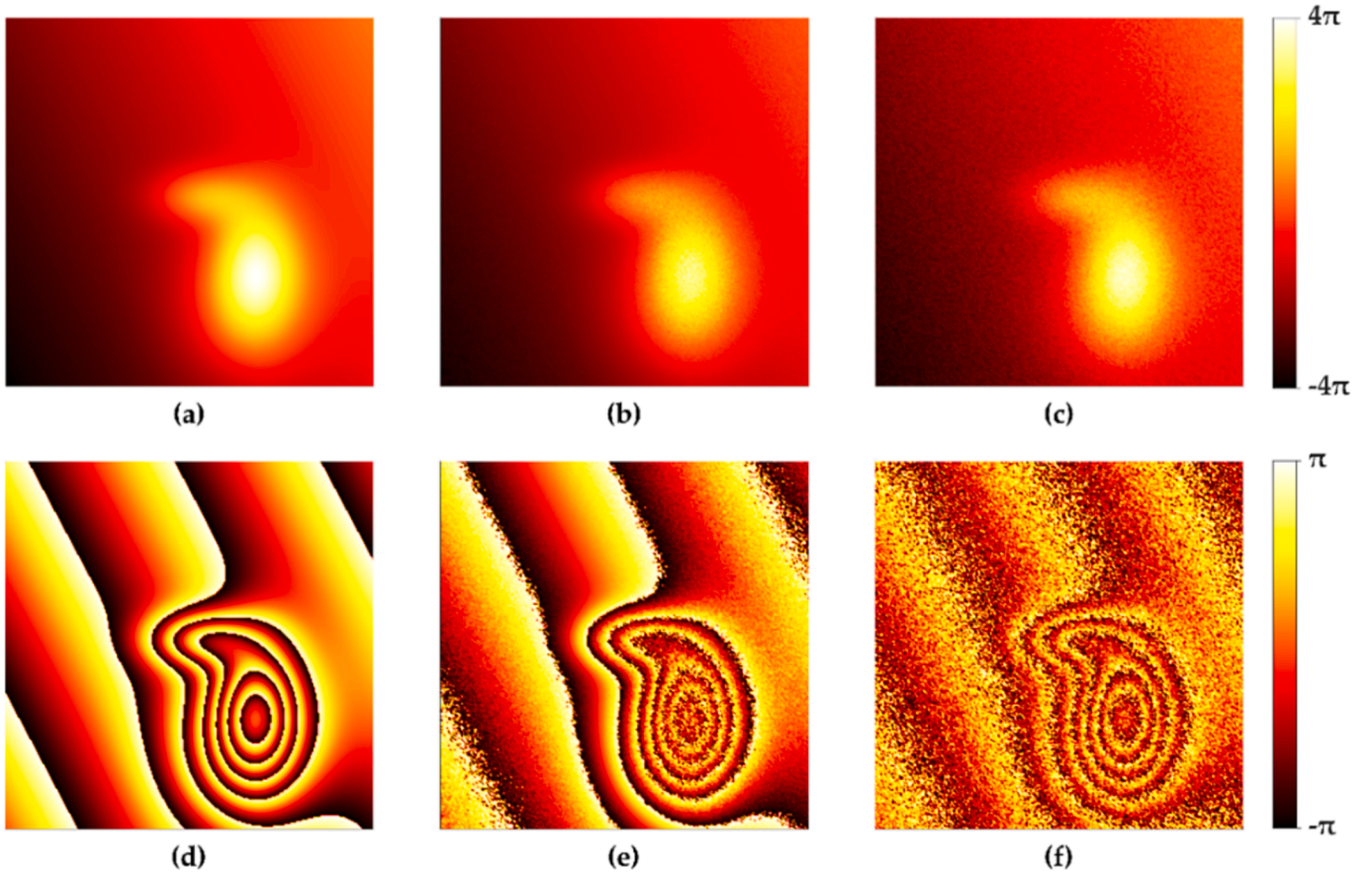


Fig. 1. Example of data generated synthetically. True phase image (a) without noise, (b) with speckle noise, (c) with Gaussian noise. The corresponding wrapped phase image (d) without noise, (e) with speckle noise, (f) with Gaussian noise. Each image has 256×256 pixels.

true phase image, resulting in the corresponding wrapped phase images with noise.

2.2. DenSFA-PU architecture

The proposed DenSFA-PU network, illustrated in Fig. 2(a), employs an encoder-decoder architecture. The encoder extracts local features from the input image and forms a feature map, while the decoder reconstructs the output image from the refined features. The SFA module integrates bi-LSTM and BAM, as shown in Fig. 3, to capture spatial dependencies and apply attention, enabling the network to focus on key features, suppress noise, and preserve critical information from the input image. The encoder consists of a convolutional layer, four dense blocks, and three transition layers. The input wrapped image first passes through a convolutional layer with an 8×8 kernel, followed by four dense blocks with three transition layers in between. Each dense block contains dense connectivity through multiple layers, including a batch normalization (BN) layer [49], a rectified linear unit (ReLU) activation layer, a 3×3 convolutional layer, and a dropout layer, as shown in Fig. 2 (b). The transition block located between dense blocks controls the spatial dimensions (width and height) of the feature map. These transition blocks differ slightly from the dense blocks, replacing the 3×3 convolutional layer and incorporating a 2×2 pooling layer to down-sample the feature map while preserving essential information.

Let's consider a layer 's' in the network that performs a composite function ' H_s ', which includes basic operations such as BN, ReLU, and convolution (Conv2D). Then, the feature map " z_s " of the s^{th} layer can be described as a function of the feature maps of all previous layers as:

$$z_s = H_s \left(\left[z_0, z_1, \dots, z_{s-1} \right] \right). \quad (4)$$

Simply, dense layers concatenate each feature map with all others. This effectively reduces the vanishing-gradient issue, which occurs when the gradients become very small and makes it difficult for the network to learn. By establishing direct connections between layers, the network learns more efficiently and quickly. This efficient learning and robust feature propagation, enabled by dense connections, allow the network to better handle the high noise levels. It also promotes feature reuse, reducing overfitting risks and improving generalization, even with limited training data.

Once an accurate feature representation of the input image is obtained, it is crucial to capture the spatial relationship between feature maps, as the phase value of a given pixel is often influenced by its neighboring pixels. To address this, we employ the SFA module, as shown in Fig. 3, which integrates bi-LSTM layers, two convolutional layers, and the attention module BAM. While the bi-LSTM captures long-range dependencies, the BAM focuses on key features. This combination effectively enhances spatial feature extraction and improves context understanding across the network, facilitating more accurate and robust phase unwrapping.

Let $E \in \mathbb{R}^{c \times h \times w}$ represent a feature map from the encoder, where c is the number of channels, h is the height, and w is the width of the map. To fully capture spatial relationships, the feature map is processed in four directions: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Each traversal creates a sequence and is processed by separate bi-LSTM layers to model relationships along each axis. For each direction, the feature map is reshaped into a sequence of feature vectors, where each position p corresponds to a feature vector $e^{(p)} \in \mathbb{R}^c$, representing the channel-wise features at that specific spatial location in the feature map. The input-output relationship of a bi-LSTM layer, when processing the feature map, can be expressed as:

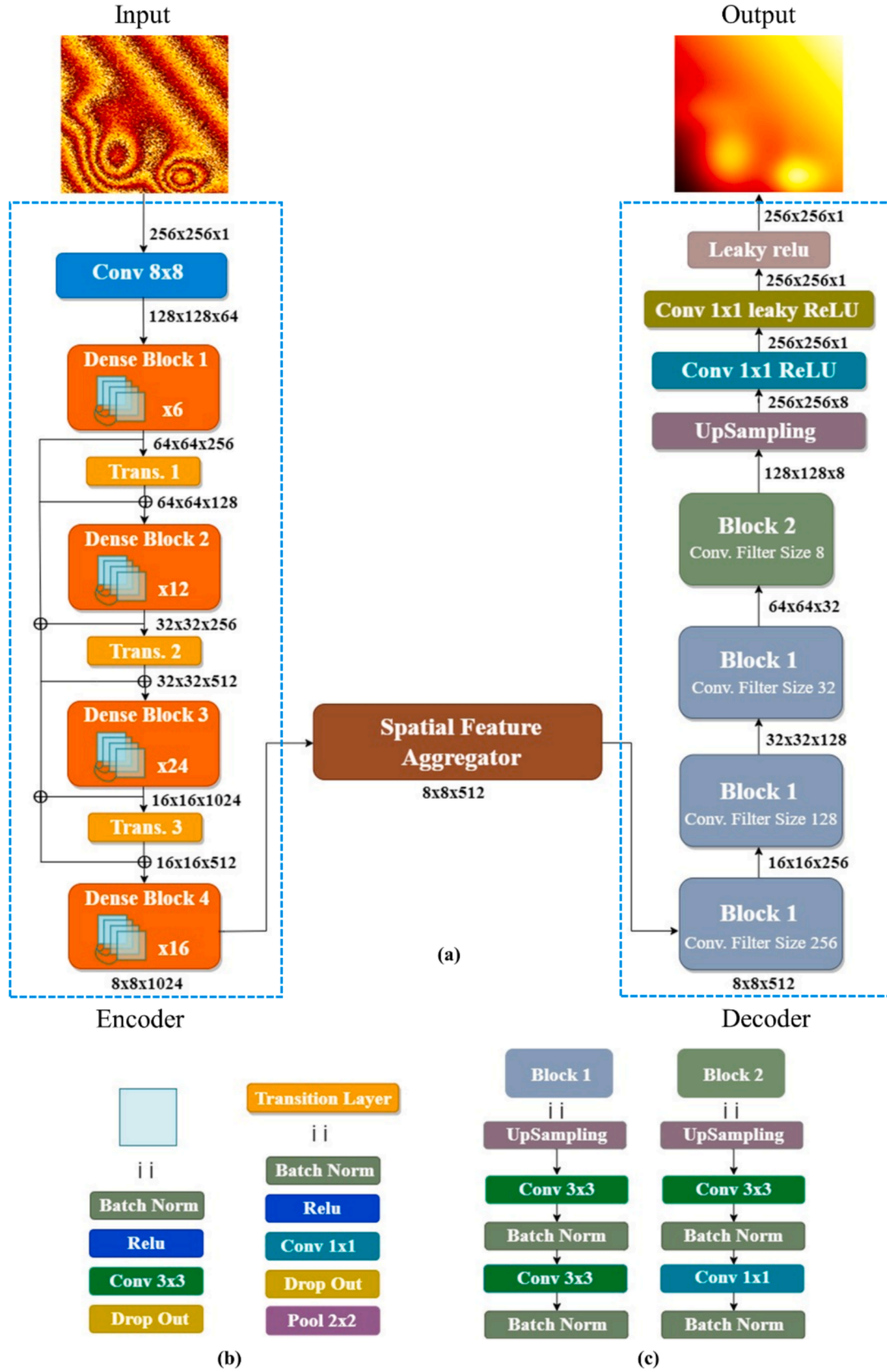


Fig. 2. (a) Architecture of the proposing DenSFA-PU network. (b) Encoder dense-block units and transition layer block. (c) Decoder blocks 1 and 2.

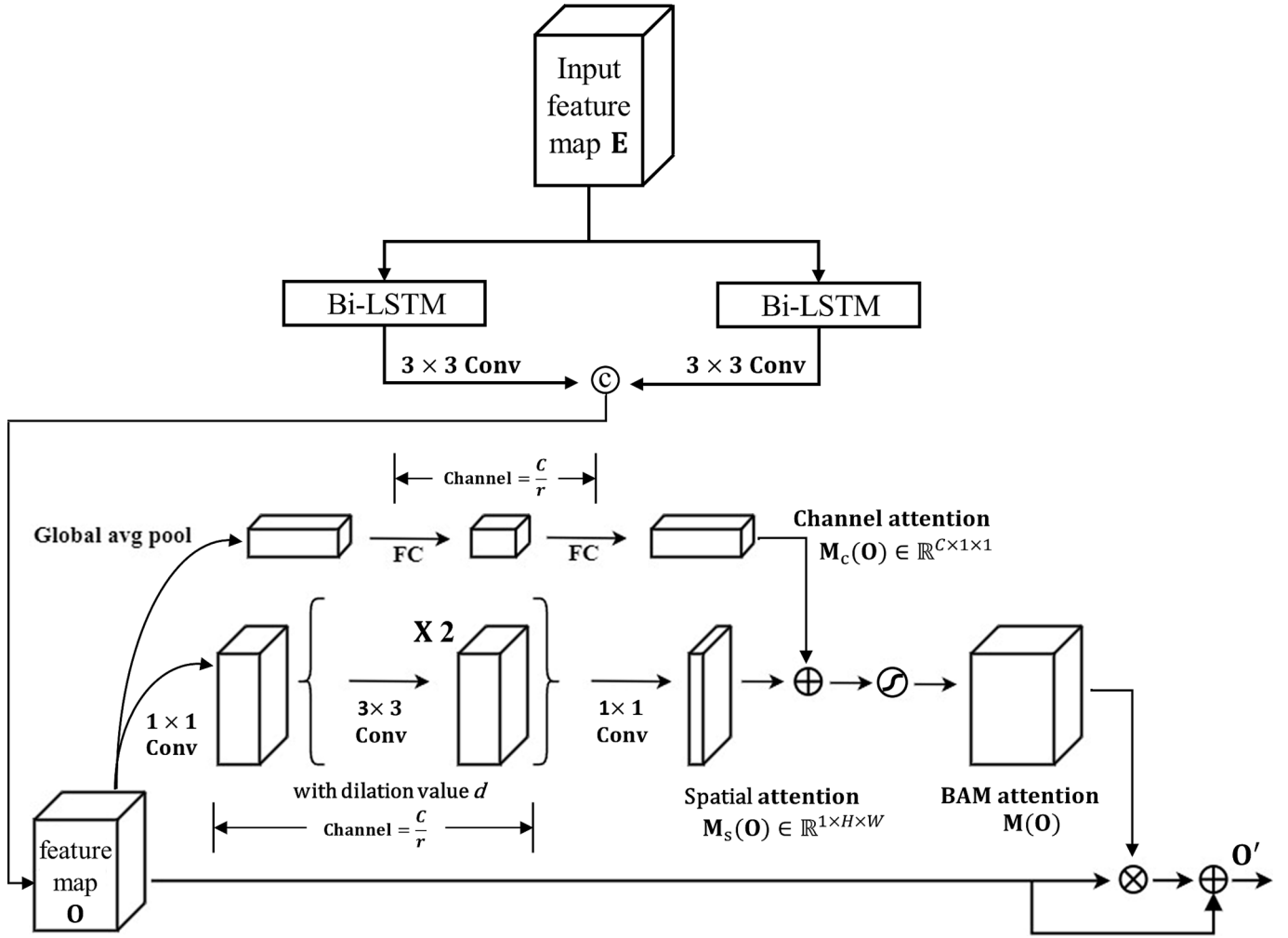


Fig. 3. Detailed structure of the spatial feature aggregator (SFA) module. The feature map E from the encoder is first passed through bi-LSTM layers to capture spatial dependencies, and then followed by convolutional layers to further capture local spatial associations. After concatenation, the refined feature map O is passed to the bottleneck attention module (BAM). It processes this intermediate feature map to generate an attention map $M(O)$ through two distinct attention pathway: channel attention $M_c(O)$ and spatial attention $M_s(O)$. Two key hyperparameters are used to optimize performance: the dilation value $d = 4$, which controls the receptive field size, and reduction ratio $r = 16$, which balances capacity and computational cost across both attention pathways.

$$o^{(p)} = \text{bi-LSTM}(e^{(p)}, o^{(p-1)}; W, u) \quad (5)$$

Here, $o^{(p)} \in \mathbb{R}^u$ is the output of bi-LSTM at a position p ($p \in [1, 2, \dots, w \times h]$), and $e^{(p)}$ is the feature vector, representing the local information at that spatial position. $o^{(p-1)}$ is the output from the previous position $p-1$ in the sequence, ensuring the model captures relationships between sequential feature vectors. W represents the trainable LSTM weights, which are not predefined. They are initialized by using standard techniques depending on activation functions, and subsequently updated or learned during the training process through backpropagation and optimization. While, u denotes the number of units in the LSTM layer. By applying this relationship to all positions in the feature map, the network effectively captures both local and long-range spatial dependencies. After processing the feature map in all four directions using bi-LSTM layers, the output features are passed through convolutional layers to further capture local spatial associations. The outputs from the convolutional layers are then concatenated, resulting in a feature map $O \in \mathbb{R}^{c \times h \times w}$, enriched with both global and local spatial dependencies.

The resulting feature maps are passed through the attention module BAM, enhancing the network's focus on critical features. While bi-LSTM layers capture long-range dependencies, they lack a mechanism to focus on the most relevant channels or spatial locations. BAM addresses this by applying channel attention to prioritize important feature channels and

spatial attention to highlight key regions, focusing on the areas having important features like phase discontinuities or noise. Now, let's assume we have an intermediate feature map O after passing through the convolutional layers, which serves as the input to BAM. This attention module calculates both channel attention $M_c \in \mathbb{R}^c$ and spatial attention $M_s \in \mathbb{R}^{h \times w}$ through two separate pathways. The channel attention path utilizes global average pooling to generate a channel vector $O_c \in \mathbb{R}^{c \times 1 \times 1}$, which encodes global information for each channel. A multi-layer perceptron (MLP) with reduced channel dimensions is then applied to compute the attention across channels. To ensure alignment with the spatial branch output, a BN layer [49] is applied afterward. The channel attention is computed as:

$$M_c(O) = \text{BN}(\text{MLP}(\text{AvgPool}(O))) \\ = \text{BN}(W_1(W_0 \text{AvgPool}(O) + b_0) + b_1) \quad (6)$$

Where, $W_0 \in \mathbb{R}^{(c/r) \times c}$, $b_0 \in \mathbb{R}^{c/r}$, $W_1 \in \mathbb{R}^{c \times (c/r)}$, $b_1 \in \mathbb{R}^c$, and r is the reduction ratio, which in our case is 16. BN denotes batch normalization.

The spatial branch generates the spatial attention map that enhances or suppresses features at different positions by integrating a bottleneck structure with dilated convolutions [50]. To identify key positions, it is crucial to effectively use contextual information, which requires a large

receptive field [51]. Dilated convolutions expand these receptive fields efficiently, enabling the creation of a more effective spatial map. The bottleneck structure reduces both the number of parameters and computational overhead. Specifically, the feature map $\mathbf{O} \in \mathbb{R}^{c \times h \times w}$ is first projected into a reduced dimension $\mathbb{R}^{(c/r) \times h \times w}$ using a 1×1 convolution layer, which integrates and compresses the feature map across channels. After the dimensional reduction, two 3×3 dilated convolutions are applied to efficiently capture contextual information. Finally, the features are reduced to a $\mathbb{R}^{1 \times h \times w}$ spatial attention map using another 1×1 convolution layer, followed by batch normalization layer for scale adjustment. The spatial attention is computed as:

$$\mathbf{M}_s(\mathbf{O}) = \text{BN}(f_3^{1 \times 1}(f_2^{3 \times 3}(f_1^{3 \times 3}(f_0^{1 \times 1}(\mathbf{O})))))) \quad (7)$$

Here, f represents a convolution operation, and the superscripts refer to the sizes of the convolutional filters. This sequence includes two 1×1 convolutions for channel reduction, followed by 3×3 dilated convolutions to gather contextual information with an expanded receptive field. The final attention map $\mathbf{M}(\mathbf{O})$ is formed by combining the channel attention $\mathbf{M}_c(\mathbf{O})$ and the spatial attention $\mathbf{M}_s(\mathbf{O})$, followed by the sigmoid function (σ) as:

$$\mathbf{M}(\mathbf{O}) = \sigma(\mathbf{M}_c(\mathbf{O}) + \mathbf{M}_s(\mathbf{O})) \quad (8)$$

Both attention branches are resized to the same dimension $\mathbb{R}^{c \times h \times w}$. These attention maps are then combined by summing them into a final attention map $\mathbf{M}(\mathbf{O}) \in \mathbb{R}^{c \times h \times w}$. The refined feature map $\mathbf{O}'$ is then calculated as:

$$\mathbf{O}' = \mathbf{O} + \mathbf{O} \otimes \mathbf{M}(\mathbf{O}) \quad (9)$$

Where, the operator $\otimes$ represents element-wise multiplication. The detailed structure of the SFA module is shown in Fig. 3.

Finally, the features refined by the attention module are passed into the CNN-based decoder. The decoder plays a crucial role in reconstructing the output phase image having the size of input image. It begins with an up-sampling layer that expands the output dimensions while preserving the learned features, enabling the restoration of fine details. Each up-sampling step is followed by convolutional layers, where features are convolved with learnable filters. The BN layers stabilize the training and improve convergence. This process, repeated over two stages, incrementally enhances the spatial resolution. Fig. 2(c) illustrates the details of decoder blocks 1 and 2, where block 1 is repeated three times with varying filter sizes. The final decoder layer consists of convolutional layers with leaky ReLU activation functions, which introduce controlled non-linearity for maintaining the information propagation and refining the final reconstruction.

2.3. Loss function

We use the composite loss function ($\mathcal{L}_c$) implemented by linearly combining three different losses: the variance ($\mathcal{L}_{var}$), the total variation ($\mathcal{L}_{tv}$), and the mean squared ($\mathcal{L}_2$) error:

$$\mathcal{L}_c = \alpha \mathcal{L}_{var} + \beta \mathcal{L}_{tv} + \gamma \mathcal{L}_2 \quad (10)$$

These losses are defined as:

$$\mathcal{L}_{var} = \mathbb{E}[(\hat{Y} - Y)^2] - (\mathbb{E}[(\hat{Y} - Y)])^2 \quad (11)$$

$$\mathcal{L}_{tv} = \mathbb{E}[|\hat{Y}_x - Y_x| + |\hat{Y}_y - Y_y|] \quad (12)$$

$$\mathcal{L}_2(\text{MSE}) = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (13)$$

The weights α , β , and γ in Eq. (10) balance the three losses, and their values are empirical determined. In our case, we got experimentally $\alpha =$

$1, \beta = 0.1$, and $\gamma = 0.2$. The variance of error loss, defined in Eq. (11), accommodates multiple solutions due to the non-unique nature of phase unwrapping. The total variation of error loss, defined in Eq. (12), measures the mean absolute differences between the predicted and true values in both the horizontal and vertical directions across the image. This loss enhances the similarity between the predicted and true phases, enforcing smoothness and discouraging abrupt changes. The $\mathcal{L}_2$ loss in Eq. (13), calculates the average of the squared differences between true and predicted values, where n represents the number of pixels in the image.

3. Experimental results

3.1. Training and evaluation criteria

The training was conducted on a workstation equipped with an Intel i7-8700 K CPU, an NVIDIA GeForce RTX 2080Ti GPU, and 112 GB of RAM. In Fig. 4, the training and testing process of the DenSFA-PU network is illustrated. Three datasets of 5000 synthetic phase images, each consisting of 256×256 pixels with phase values ranging from -10π to 10π , were generated using Eq. (3). Each dataset included 4000 images for training and 1000 for testing. One dataset was created by adding Gaussian noise with a normal distribution, defined in terms of SNR values ranging from 0 dB to 10 dB. As the SNR decreases, indicating higher noise levels, the unwrapping process becomes increasingly challenging. Another dataset included speckle noise with a mean of 0 and a variance ranging from 0.005 to 0.05. The datasets were named as: “Gaussian noise dataset” for the Gaussian noise, “speckle noise dataset” for the speckle noise, and “noise-free dataset” for the one without any added noise. These datasets were used to train and test the network, which was implemented in Keras using Python 3.7.13. The network was trained on the paired set of a wrapped phase image and its ground truth image. The Adam optimizer was utilized with a learning rate of 0.001, and a batch size of 16 was used. The training losses of three datasets are shown in Fig. 5.

We comprehensively evaluate the performance of phase unwrapping on normalized true and predicted phase images using three key metrics: NRMSE, PSNR, and SSIM, as described in the introduction part. NRMSE measures the deviation between predicted and true values, normalized by the data range, and is expressed as a percentage. A lower NRMSE value indicates higher accuracy. PSNR evaluates the signal-to-noise ratio, with a higher value reflecting better image quality. SSIM compares the predicted phase ($\hat{Y}$) with the true phase (Y), evaluating their similarity in terms of brightness, contrast, and structure. It uses the mean (average pixel values), variance (spread of pixel values), and covariance (similarity of pixel value). In practice, for SSIM, sliding windows are used to divide images into smaller sections. Within each window, Gaussian weighting is applied to calculate the mean, variance, and covariance, which are then used to compute structural similarity. This weighting emphasizes the pixels closer to the center of the window, as they are more perceptually significant according to the human visual system. The overall SSIM score is calculated by averaging the scores across all windows, which ranges from 0 to 1. The evaluation metrics mentioned above are computed as follows:

$$\text{NRMSE} = \frac{\sqrt{\mathbb{E}[(Y - \hat{Y})^2]}}{Y_{\max} - Y_{\min}} \quad (14)$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \quad (15)$$

$$\text{SSIM}(Y, \hat{Y}) = \frac{(2\mu_Y \mu_{\hat{Y}} + c_1)(2\sigma_{Y\hat{Y}} + c_2)}{(\mu_Y^2 + \mu_{\hat{Y}}^2 + c_1)(\sigma_Y^2 + \sigma_{\hat{Y}}^2 + c_2)} \quad (16)$$

In Eq. (14), $Y_{\max}$ and $Y_{\min}$ represent the maximum and minimum values

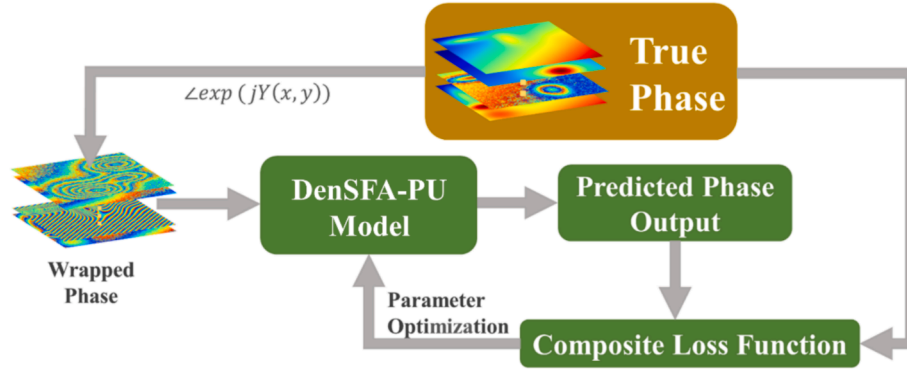


Fig. 4. Training and testing process of DenSFA-PU.

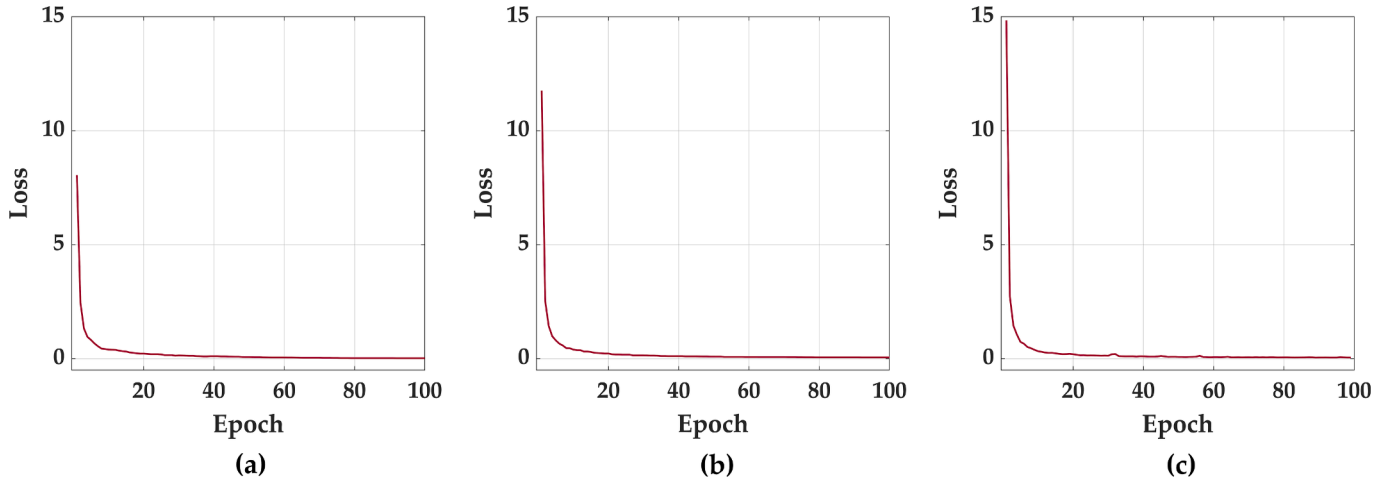


Fig. 5. Training losses of datasets for (a) noise-free dataset, (b) speckle noise dataset, and (c) Gaussian noise dataset.

in the normalized true phase image, respectively. In this study, the data was generated with phase values ranging from -10π to 10π , which were subsequently normalized to the range of $[0, 255]$. Therefore, the maximum possible value for Y_{max} is 255, corresponding to 10π , and the minimum possible value for Y_{min} is 0, corresponding to -10π . In Eq. (15), MAX is the maximum pixel value which is set as 255, and MSE represents the mean square error as defined in Eq. (13). In Eq. (16), μ_Y and $\mu_{\hat{Y}}$ represent the Gaussian-weighted mean pixel values of the true phase image Y and the predicted phase image $\hat{Y}$, respectively. σ_Y^2 and $\sigma_{\hat{Y}}^2$ represent the Gaussian-weighted variances, and $\sigma_{Y\hat{Y}}$ is the Gaussian-weighted covariance between Y and $\hat{Y}$. These can be calculated as:

$$\mu_Y = \sum_i w_i Y_i, \mu_{\hat{Y}} = \sum_i w_i \hat{Y}_i \quad (17)$$

$$\sigma_Y^2 = \sum_i w_i (Y_i - \mu_Y)^2, \sigma_{\hat{Y}}^2 = \sum_i w_i (\hat{Y}_i - \mu_{\hat{Y}})^2 \quad (18)$$

$$\sigma_{Y\hat{Y}} = \sum_i w_i (Y_i - \mu_Y)(\hat{Y}_i - \mu_{\hat{Y}}) \quad (19)$$

Where, w_i is normalized Gaussian weight for pixel i and can be calculated as:

$$w_i = \frac{\exp\left(-\frac{d_i^2}{2\sigma^2}\right)}{\sum_j \exp\left(-\frac{d_j^2}{2\sigma^2}\right)} \quad (20)$$

The distance of pixel i from the center of the Gaussian kernel is denoted by d_i , and σ is the standard deviation of the Gaussian kernel. The constants c_1 and c_2 in Eq. (16) are added to ensure numerical stability and avoid division by zero. They are calculated as:

$$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2 \quad (21)$$

Here, L is the dynamic range of the phase values (the difference between the maximum and minimum pixel values), and k_1, k_2 are typically set to 0.01 and 0.03, respectively. The SSIM measures the overall similarity between images but does not capture pixel-level differences, which are measured using a difference map to calculate the error between the true and predicted images.

3.2. Experiments with synthetic data

Simulated experiments were conducted in both ideal (noise-free) and non-ideal (noisy) scenarios using synthetic data. The effectiveness and validity of the proposed DenSFA-PU were initially evaluated using two noise-free phase images, each with 256×256 pixels, as shown in Fig. 6. These images were selected randomly from the test set and were not used for training. The first wrapped phase image, Fig. 6(a), is dominated by ramp phase, providing a clear and continuous gradient. In contrast, the second wrapped image in Fig. 6(e) is dominated by Gaussian phase, showing more complexity and variability, which reflects a more challenging scenario. The corresponding true phase images are shown in Fig. 6(b) and (f). The phase images unwrapped with the proposed DenSFA-PU are shown in Fig. 6(c) and (g). The predictions of the proposed network and the actual true phase images were so similar that it

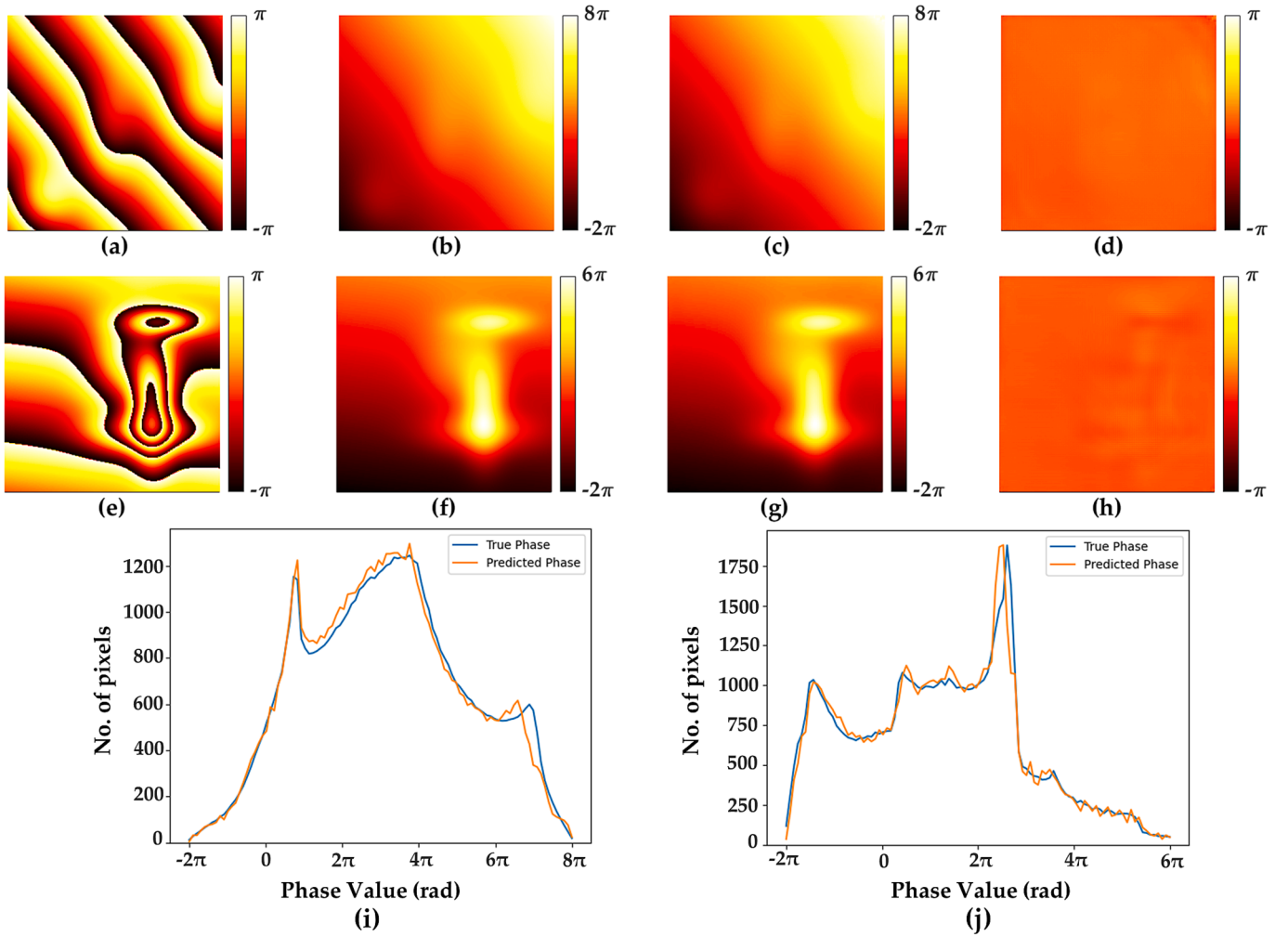


Fig. 6. Simulated experiment conducted with two noise-free phase images, dominated by ramp phase (1st row) and Gaussian phase (2nd row). (a), (e) wrapped phase images; (b), (f) true phase images; (c), (g) images unwrapped with DenSFA-PU; (d), (h) difference maps between the true and predicted phase images. Each image has 256×256 pixels. (i), (j) Histogram analysis of the true and predicted phase images.

was difficult to distinguish them visually. Fig. 6(d) and (h) show the difference between the true and the predicted phase images, where we could observe that most of the phase errors were small and sparsely located. Additionally, histogram analysis in Fig. 6(i) and (j) shows that most predicted values closely followed the true values, despite some minor misalignments.

Table 1 presents the NRMSE (in percentage), PSNR, and SSIM metrics obtained by the proposed DenSFA-PU network with the noise-free data of Fig. 6.

Secondly, the performance of DenSFA-PU was evaluated using two randomly selected phase images taken from the speckle noise dataset to simulate scenarios with varying noise characteristics and examine their impact on the unwrapping performance. The noise levels were chosen to represent low variance (0.001) and high variance (0.03), corresponding to set A and set B, respectively. Low variance corresponds to minimal intensity fluctuations in the speckle noise, resulting in cleaner phase data, whereas high variance indicates significant fluctuations, creating noisier and more challenging data.

Table 1

NRMSE, PSNR, and SSIM for noise-free phase images in Fig. 6.

Wrapped Phase Image	NRMSE(%)	PSNR	SSIM
Dominated by ramp phase	0.52	69.785	1.000
Dominated by Gaussian phase	0.55	67.426	0.999

Comparisons were made with one conventional method, Quality-guided Path Following (QPF) [12], and four deep learning-based methods: DLPU [25], VUR-Net [28], U²-Net [38], and Hybrid Net [44]. Fig. 7 shows an input wrapped phase image (a), referred to as set A, containing speckle noise with a variance of 0.001, and its corresponding true phase image (b). The output unwrapped phase image predicted by DenSFA-PU is shown in (c) and compared with those predicted by DLPU (d), VUR-Net (e), U²-Net (f), QPF (g), and the Hybrid Net (h). While most methods produced visually similar outputs, DLPU and VUR-Net exhibited boundary errors in some regions, U²-Net and Hybrid Net showed residual errors, and QPF had difficulties with corner pixels, as illustrated in Fig. 7(i-n), which highlights the differences between the true and predicted phase images.

Another wrapped phase image (set B) with higher speckle noise (variance = 0.03) is presented in Fig. 8(a), while Fig. 8(b) shows the corresponding true phase image. Fig. 8(c)–8(h) display the unwrapped phase images using DenSFA-PU (c), DLPU (d), VUR-Net (e), U²-Net (f), QPF (g), and Hybrid Net (h). Due to the higher speckle noise, DLPU, VUR-Net, U²-Net, QPF, and the Hybrid Net exhibited significant phase errors, such as large deviations, residual noise artifacts, and discontinuities in certain regions, as shown in difference maps of Fig. 8(j-n). These errors indicate a loss of accuracy and structural inconsistency in the unwrapped phase images. In contrast, the proposed DenSFA-PU demonstrated lower phase errors, maintaining better structural stability and preserving finer details despite the challenging noise conditions,

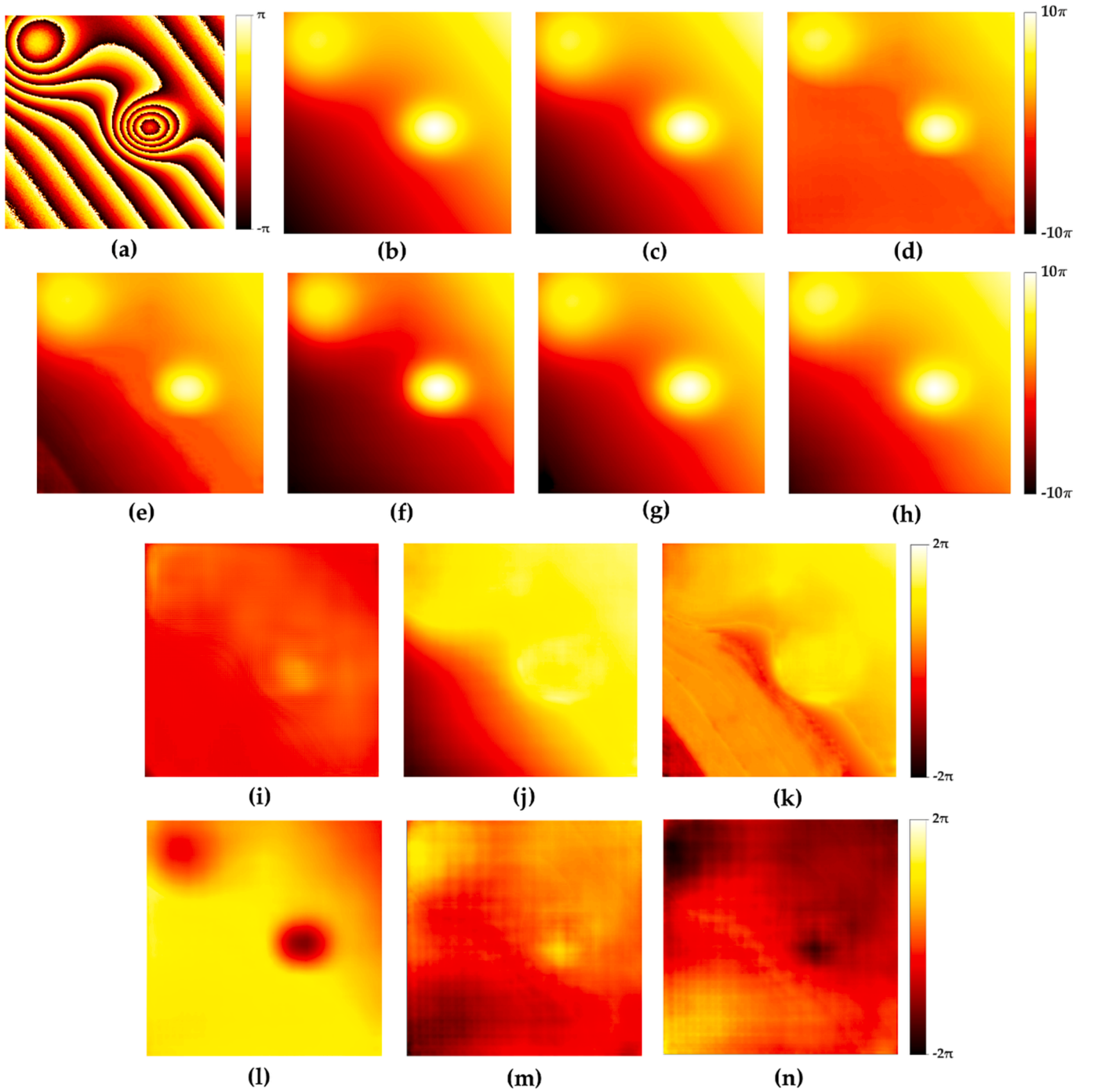


Fig. 7. Phase unwrapping performed on a synthetic phase image having low speckle noise (Variance = 0.001). (a) Network input wrapped phase image and (b) its corresponding true phase image. The images unwrapped with (c) DenSFA-PU, (d) DLPU, (e) VUR-net, (f) U^2 -Net, (g) QPF, and (h) Hybrid Net. Difference maps for (i) DenSFA-PU, (j) DLPU, (k) VUR-Net, (l) U^2 -Net, (m) QPF, and (n) Hybrid Net. Each image has a size of 256×256 pixels.

as shown in the difference map of Fig. 8(i).

Table 2 summarizes the NRMSE (in percentage), PSNR, and SSIM metrics of both sets A and B across the six unwrapping methods. DLPU and VUR-Net exhibited larger phase errors, lower PSNR, and similar SSIM values. QPF performed well under low speckle noise but struggled with high noise. U^2 -Net and the Hybrid Net showed improvements in all three metrics compared to other methods. However, the proposed DenSFA-PU outperformed all others, achieving NRMSE values of 0.61 % for set A and 0.69 % for set B, along with higher PSNR and SSIM of 1.

To further evaluate the robustness of the proposed DenSFA-PU under severe noise conditions, an additional experiment was performed using

speckle noise with a mean of 0 and an increased variance of 0.1. This variance level exceeds the range used in the test dataset. In Fig. 9, (a) shows the wrapped phase image with severe speckle noise, (b) depicts the true phase image, and (c-h) present the unwrapped results. DenSFA-PU reliably produced accurate result, preserving structural consistency. This robust performance is due to its ability to effectively capture both local and global spatial dependencies through its spatial feature aggregator module, which models phase relationships even under high noise. U^2 -Net and the Hybrid Net also demonstrated competitive results, while DLPU and VUR-Net exhibited residual errors. QPF, on the other hand, failed to unwrap. This is also evident from the difference maps in Fig. 9

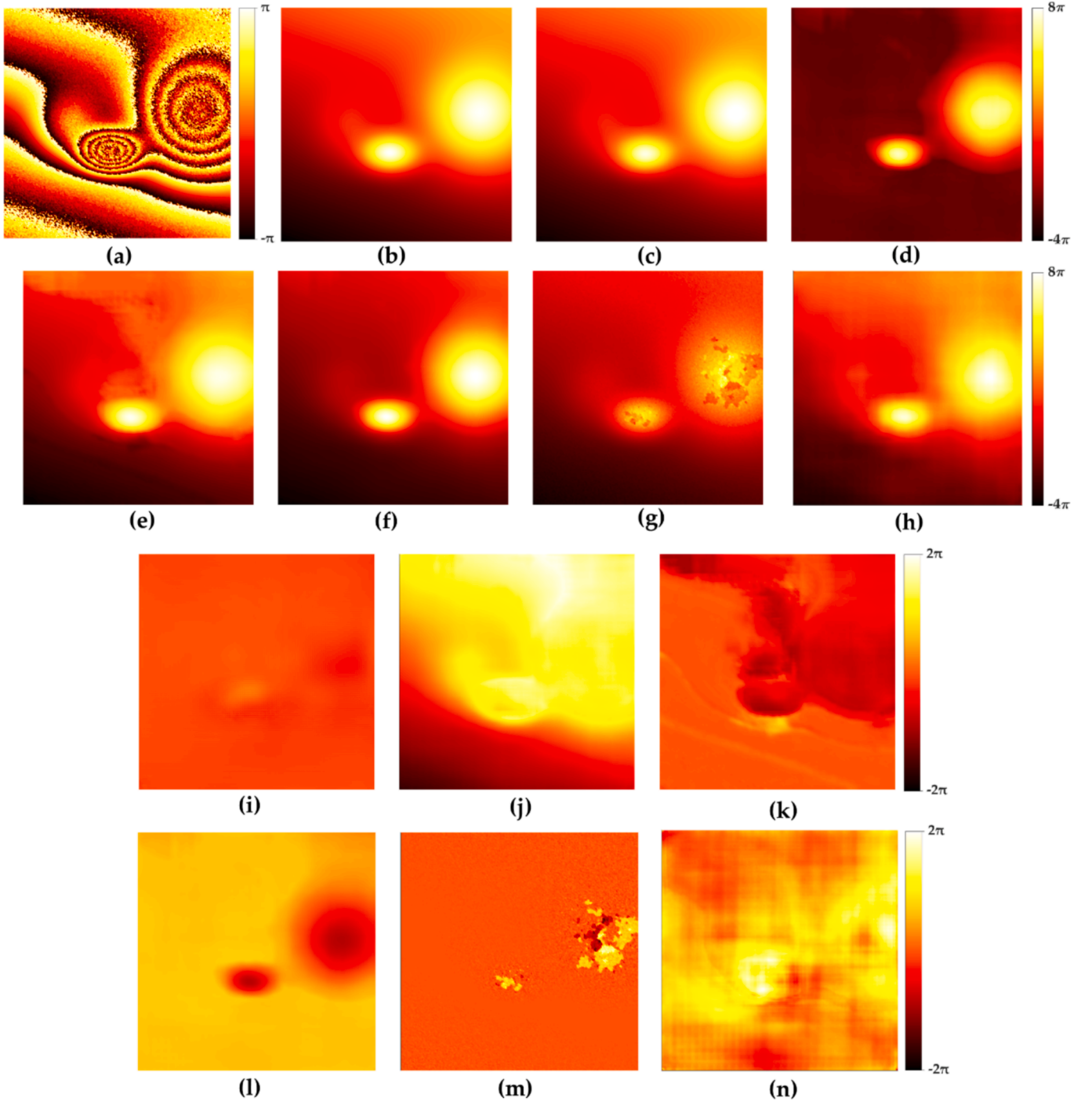


Fig. 8. Phase unwrapping performed on a synthetic phase image having high speckle noise (Variance = 0.03). (a) Network input wrapped phase image and (b) its corresponding true phase image. The images unwrapped with (c) DenSFA-PU, (d) DLPU, (e) VUR-net, (f) U²-Net, (g) QPF, and (h) Hybrid Net. Difference maps for (i) DenSFA-PU, (j) DLPU, (k) VUR-Net, (l) U²-Net, (m) QPF, and (n) Hybrid Net. Each image has a size of 256 × 256 pixels.

Table 2

Comparison of evaluation metrics for speckle noise phase images.

Methods	Set A (Variance = 0.001)			Set B (Variance = 0.03)		
	NRMSE(%)	PSNR	SSIM	NRMSE(%)	PSNR	SSIM
DLPU [25]	2.86	51.565	0.970	3.39	51.078	0.960
VUR-Net [28]	2.73	53.686	0.976	2.96	53.325	0.978
U ² -Net [38]	0.82	67.406	0.994	0.94	65.893	0.988
QPF [12]	0.97	65.151	0.991	1.73	57.886	0.983
Hybrid Net [44]	0.85	66.227	0.987	0.98	62.438	0.986
DenSFA-PU	0.61	70.670	1.000	0.69	69.897	0.999

(i-n), which show high errors in all methods except the proposed method. Table 3 presents the evaluation metrics for the severe noise condition. These findings highlight the superior robustness of DenSFA-PU in handling severe speckle noise.

To investigate the impact of different Gaussian noise levels, three phase images having different shapes and noise levels (10, 5, and 0 dB) were randomly selected from the Gaussian noise dataset. Fig. 10(a) shows the wrapped input phase image with a 10 dB Gaussian noise level, Fig. 11(a) with a 5 dB noise level, and Fig. 12(a) with severe 0 dB noise. Their corresponding true phase images are presented in Fig. 10(b), 11

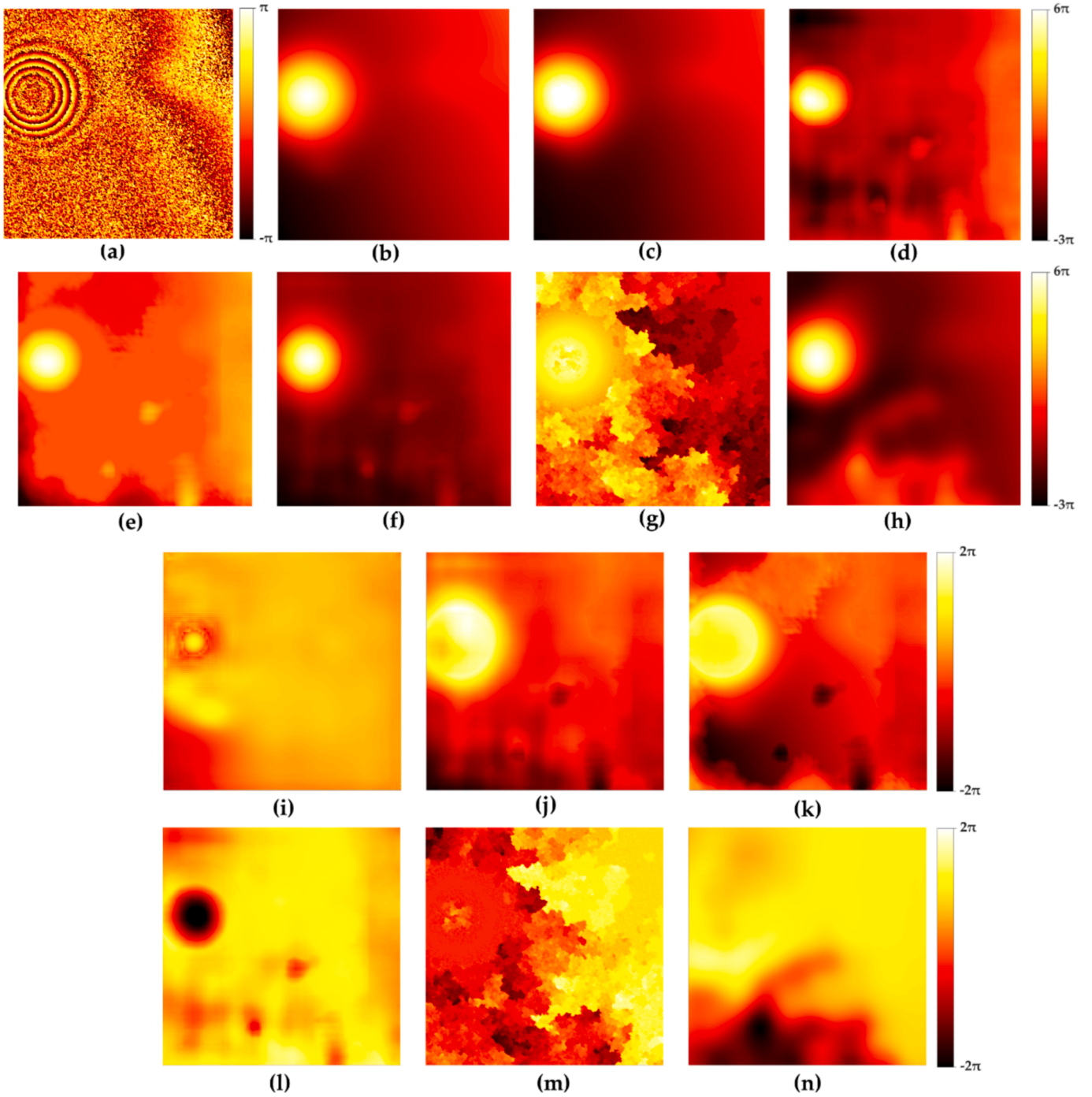


Fig. 9. Phase unwrapping performed on a synthetic phase image having severe speckle noise (Variance = 0.1). (a) Network input wrapped phase image and (b) its corresponding true phase image. The image unwrapped with (c) DenSFA-PU, (d) DLPU, (e) VUR-net, (f) U²-Net, (g) QPF, and (h) Hybrid Net. Difference maps for (i) DenSFA-PU, (j) DLPU, (k) VUR-Net, (l) U²-Net, (m) QPF, and (n) Hybrid Net. Each image has a size of 256×256 pixels.

Table 3

Evaluation of phase images having severe speckle noise (Variance = 0.1) of Fig. 9.

Methods	NRMSE (%)	PSNR	SSIM
DLPU [25]	3.85	50.396	0.957
VUR-Net [28]	3.49	51.278	0.962
U ² -Net [38]	0.96	65.306	0.988
QPF [12]	8.54	37.417	0.792
Hybrid Net [44]	1.02	61.548	0.981
DenSFA-PU	0.77	68.579	0.992

(b), and 12(b). Figs. 10–12(c–h) display the predicted unwrapped phase images obtained using DenSFA-PU, DLPU, VUR-Net, U²-Net, and the Hybrid Net, respectively. The differences between the true phase images and the predicted ones were computed and are shown in Figs. 10–12(i–n). These results clearly show that as the noise level increased, the differences became prominent, but DenSFA-PU maintained consistent performance even under high noisy conditions.

Table 4 presents the NRMSE, PSNR, and SSIM values for the five networks tested with varying levels of Gaussian noise. The data indicates that DenSFA-PU outperforms the other models, particularly in terms of accuracy, noise robustness, and structural similarity. DenSFA-PU

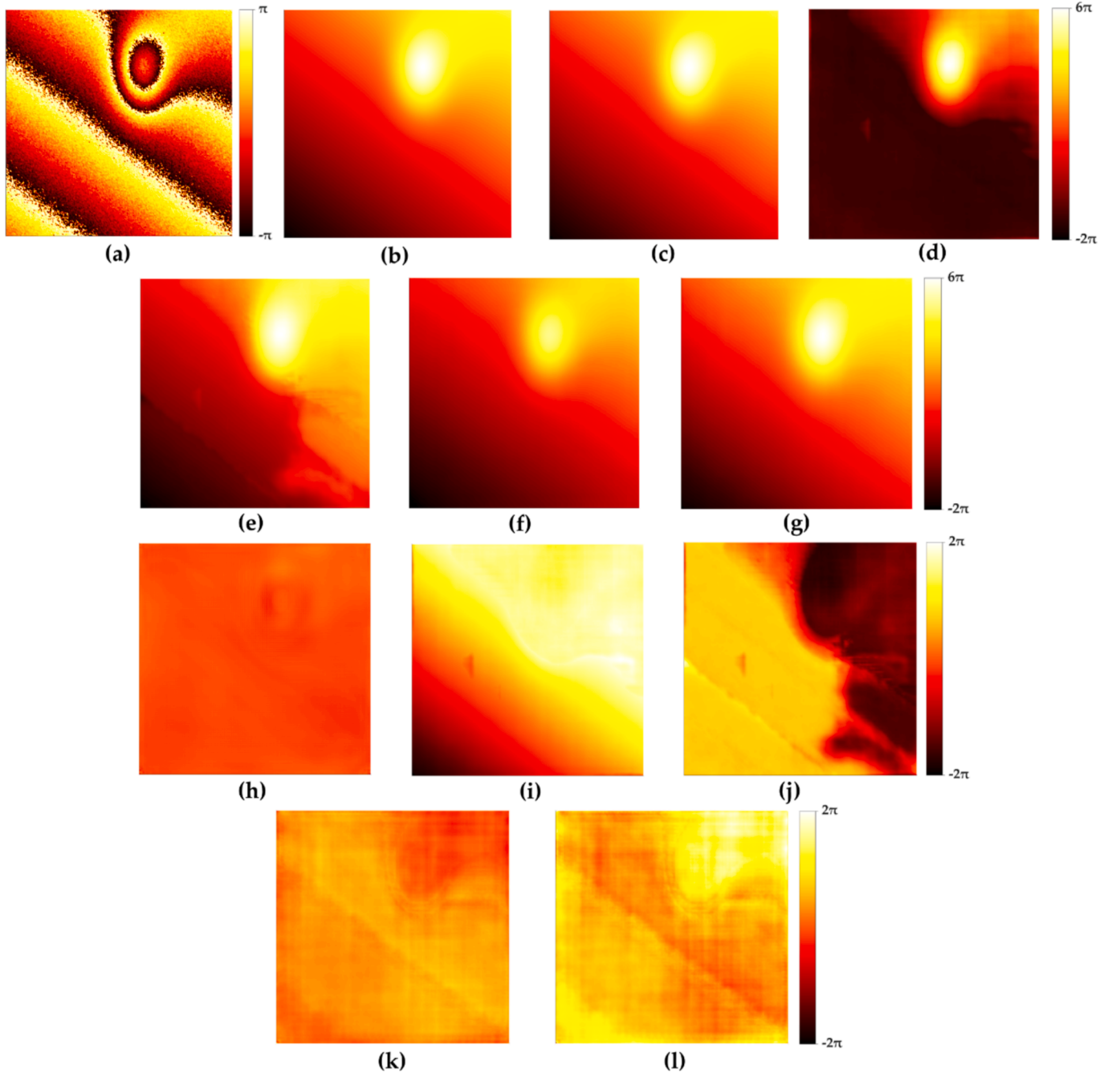


Fig. 10. Phase unwrapping of a synthetic phase image having 10 dB (SNR) Gaussian noise level. (a) Input wrapped phase image and (b) corresponding true phase image. The phase images unwrapped with (c) DenSFA-PU, (d) DLPU, (e) VUR-Net, (f) U²-Net, and (g) Hybrid Net. Difference maps for (h) DenSFA-PU, (i) DLPU, (j) VUR-Net, (k) U²-Net, and (l) Hybrid Net. Each image has a size of 256×256 pixels.

consistently achieved high PSNR, ranging from 68.97 to 70.11, significantly outperforming the other models. While the SSIM values for other networks fluctuated, especially as the noise level increased, DenSFA-PU demonstrated superior structural stability, converging consistently near to 1. This highlights DenSFA-PU's effectiveness in handling noise while preserving structural similarity. Among the tested models, only U²-Net and the Hybrid Net showed competitive performance, whereas DLPU, and VUR-Net failed to deliver reliable results under high noise levels.

The unwrapping process times were measured using 100 phase images from test dataset: 40 with speckle noise, 40 with Gaussian noise, and 20 without any noise. The results for the five networks are presented in Table 5. The average unwrapping time for DenSFA-PU was calculated as 29.31 ms for a phase map of 256×256 pixels. Despite the large

number of parameters in DenSFA-PU, it did not significantly increase the processing time compared to DLPU, VUR-Net, U²-Net, and Hybrid Net, making it suitable for real-time applications.

3.3. Experiments with real data

To further evaluate the robustness of the proposed method, two experiments were conducted with real-world data. The experimental setup of Fig. 13 was used to generate wrapped phase images having various phase patterns and noise levels. Fig. 13(a) is the schematic of the used off-axis holographic imaging system, and (b) a photograph of the setup. The beam from a light source is divided by a beam splitter (BS): one is directed to the sample and the other to a slightly tilted reference mirror.

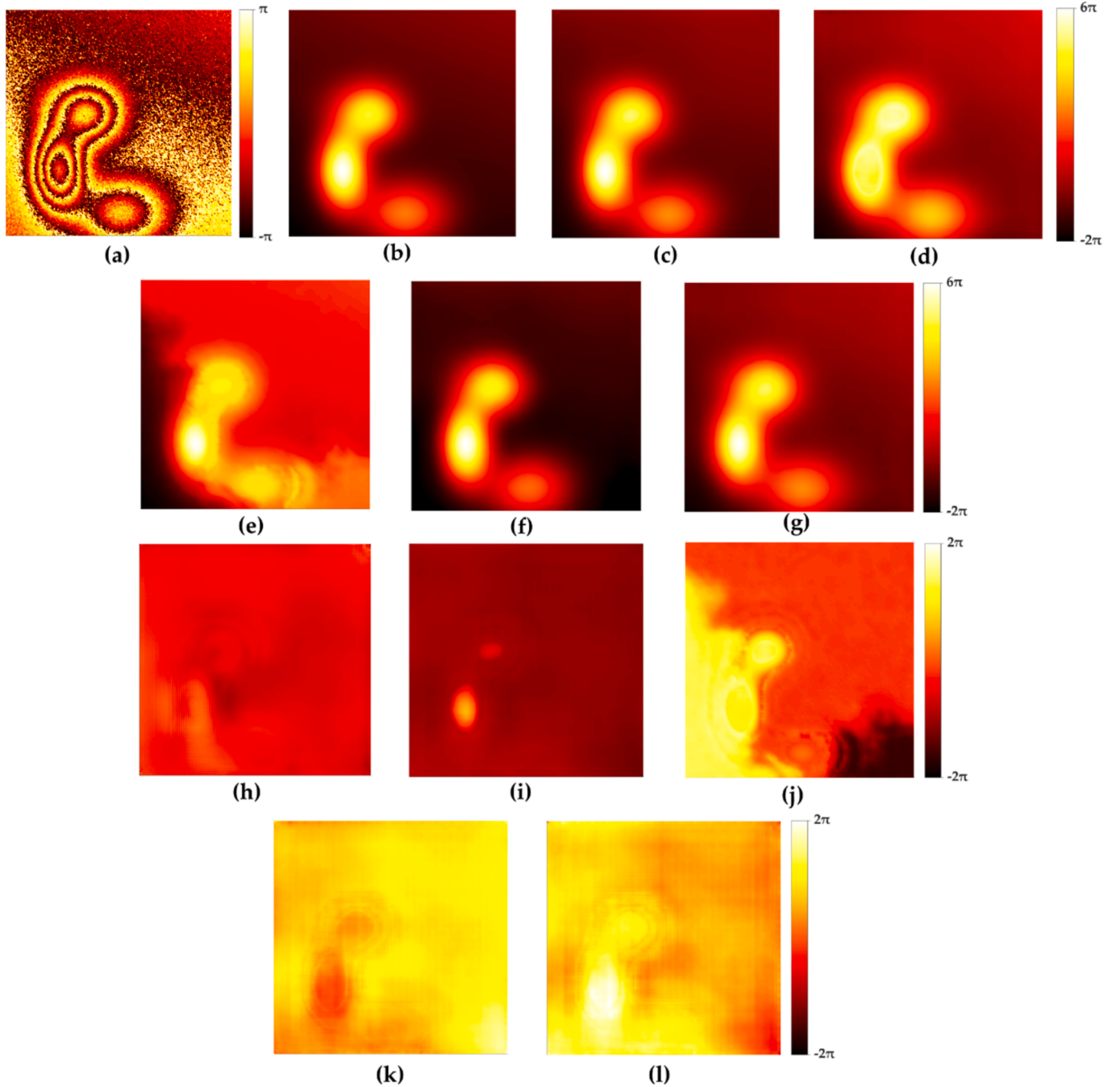


Fig. 11. Phase unwrapping of a synthetic phase image having 0 dB (SNR) Gaussian noise level. (a) Input wrapped phase image and (b) corresponding true phase image. The phase images unwrapped by using (c) DenSFA-PU, (d) DLP, (e) VUR-Net, (f) U²-Net, and (g) the Hybrid Net. Difference maps for (h) DenSFA-PU, (i) DLP, (j) VUR-Net, (k) U²-Net, and (l) the Hybrid Net. Each image has a size of 256×256 pixels.

The reflected beams are combined at the same BS, forming a high spatial frequency fringe pattern, and captured by a scientific complementary metal-oxide- semiconductor (sCMOS) camera. Fourier transform and spatial filtering of its one cross- correlation spectral component, followed by inverse Fourier transform, yield the wrapped phase image of the sample surface. Noise is unintentionally introduced by the imaging modality during the phase acquisition process through laser noise, detector noise, and environmental factors. Two objects were used as samples; a piece of acrylic plate and a polydimethylsiloxane510 (PDMS) resin block made with a 10:1 silicone-to-curing agent ratio (The sample photograph is attached in the [Supplementary Material](#)).

Firstly, the acrylic sample was used to validate the proposed phase

unwrapping method. The surface of the acrylic plate was hard, smooth, and reflective, resulting in distinct and sharp wrapped phase patterns. [Fig. 14\(a\)](#) shows the experimentally obtained phase image of the acrylic sample, wrapped and with a noise level of 6.57 dB SNR. The 3D surface plot of the wrapped image did not provide meaningful information, as shown in [Fig. 14\(b\)](#). By applying the proposed method, the image was unwrapped effectively, as shown in [Fig. 14\(c\)](#), with an unwrapping time of 29.88 ms. In contrast, other deep learning methods failed to unwrap accurately, as illustrated in [Fig. 14\(d\)-\(g\)](#). The unwrapping time was 37.24 ms for DLP, 39.18 ms for VUR-Net, 45.15 ms for U²-Net, and 43.48 ms for Hybrid Net. The 3D surface profiles in [Fig. 14\(h\)-\(l\)](#) clearly demonstrate that our method achieved a smoother and more accurate

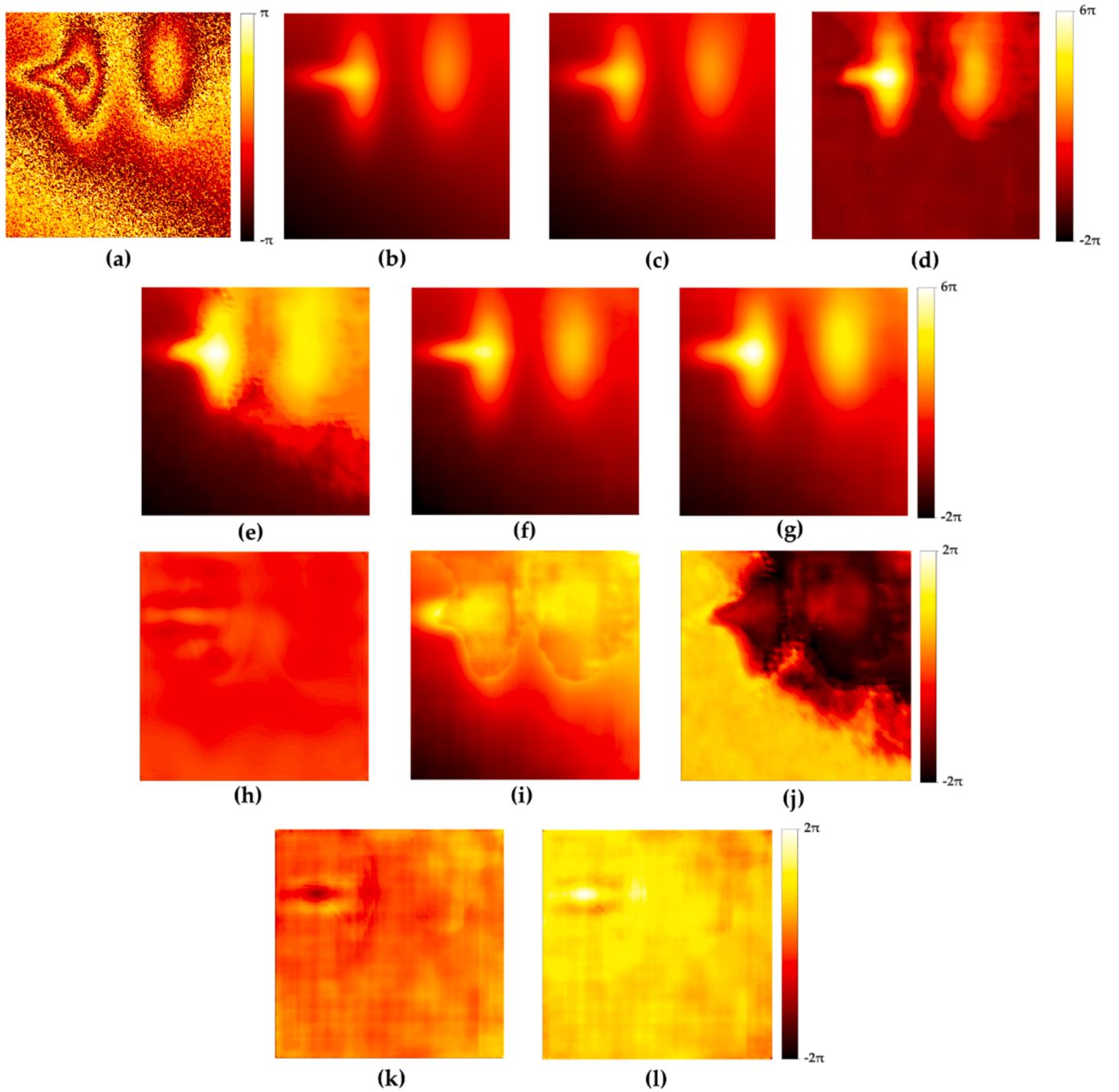


Fig. 12. Phase unwrapping of a synthetic phase image having 5 dB (SNR) Gaussian noise level. (a) Input wrapped phase image and (b) corresponding true phase image. The phase images unwrapped by using (c) DenSFA-PU, (d) DLPU, (e) VUR-Net, (f) U²-Net, and (g) the Hybrid Net. Difference maps for (h) DenSFA-PU, (i) DLPU, (j) VUR-Net, (k) U²-Net, and (l) the Hybrid Net. Each image has a size of 256×256 pixels.

Table 4

Comparison of evaluation metrics for Gaussian noise phase images.

Methods	10 dB			5 dB			0 dB		
	NRMSE(%)	PSNR	SSIM	NRMSE(%)	PSNR	SSIM	NRMSE(%)	PSNR	SSIM
DLPU [25]	2.58	53.035	0.975	3.68	53.785	0.984	4.2	51.925	0.976
VUR-Net [28]	2.87	52.821	0.971	3.95	52.293	0.977	4.13	50.368	0.971
U ² -Net [38]	0.88	67.590	0.999	0.92	65.318	0.997	0.97	64.583	0.997
Hybrid Net [44]	0.92	66.218	0.993	0.94	64.975	0.992	0.98	64.238	0.991
DenSFA-PU	0.62	70.107	1.000	0.64	69.457	1.000	0.65	68.969	0.999

Table 5

Unwrapping process time of different methods.

Method	Unwrapping Time (ms)
DLPU [25]	35.82
VUR-Net [28]	39.62
U ² -Net [38]	44.25
Hybrid Net [44]	41.73
DenSFA-PU	29.31

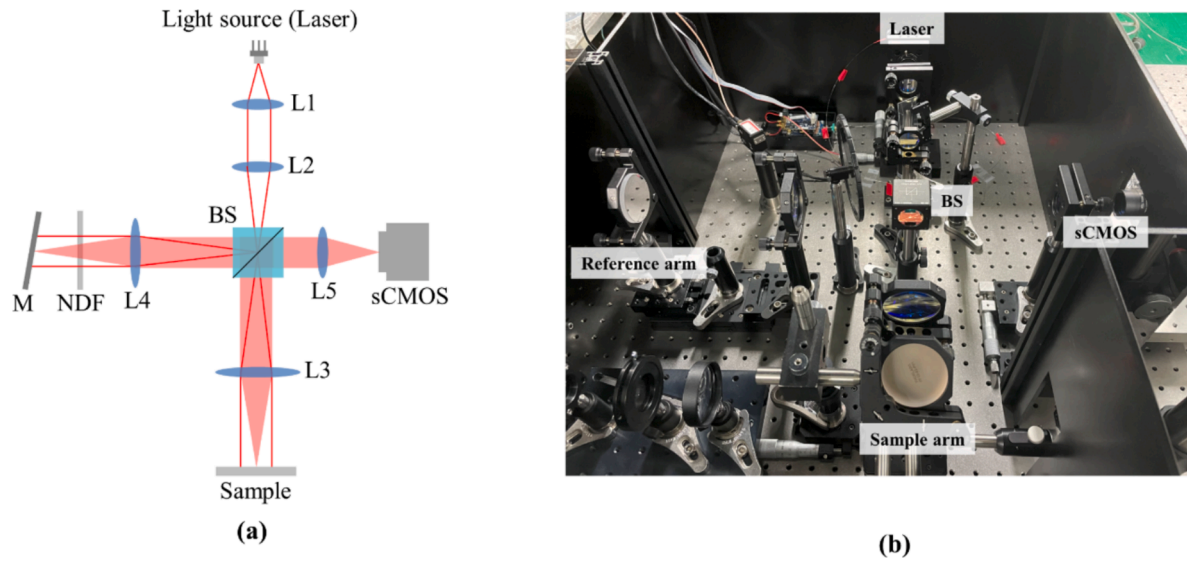


Fig. 13. The experimental setup for off-axis digital holography, (a) schematic and (b) photograph of the setup. L1-L5; lenses, BS; beam splitter, M; mirror, NDF; neutral-density filter, scientific complementary metal-oxide-semiconductor (sCMOS).

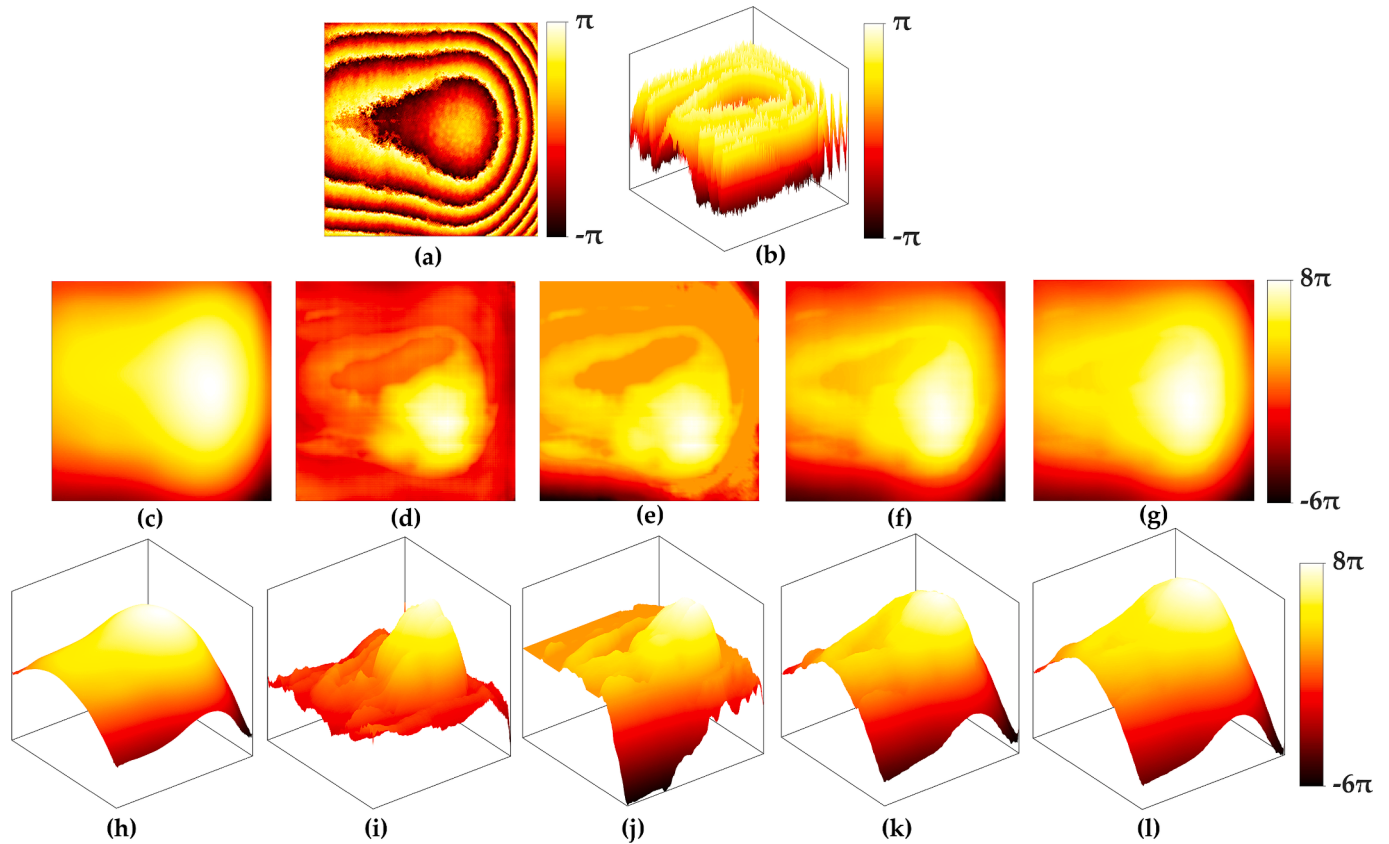


Fig. 14. Experimental results of the phase unwrapping made with an acrylic sample, (a) experimentally obtained phase image. It is wrapped and highly noisy (6.57 dB SNR). (b) 3D profile of the wrapped phase image. The phase images unwrapped with (c) DenSFA-PU, (d) DLP, (e) VUR-Net (f) U^2 -Net, and (g) Hybrid Net. (h)-(l) The 3D plots of corresponding 3D surface profiles. Each image has 256×256 pixels.

result than the others. Further, it also maintained a lower computational cost.

In addition to the acrylic sample, the PDMS sample with a rather flexible surface was used to further validate the proposed method. It exhibited more congested wrapped phase patterns and a widely variable surface profile compared to the acrylic sample, making the unwrapping

challenge more complex, at an SNR of 6.52 dB, as shown in Fig. 15(a). The 3D surface profile of the wrapped image was cluttered and provided no useful information, as seen in Fig. 15(b). Our proposed method in Fig. 15(c) outperformed others in terms of accuracy and robustness, with an unwrapping time of 30.12 ms. Comparative evaluations with other deep learning methods are presented in Fig. 15(d)-(g), where only U^2 -

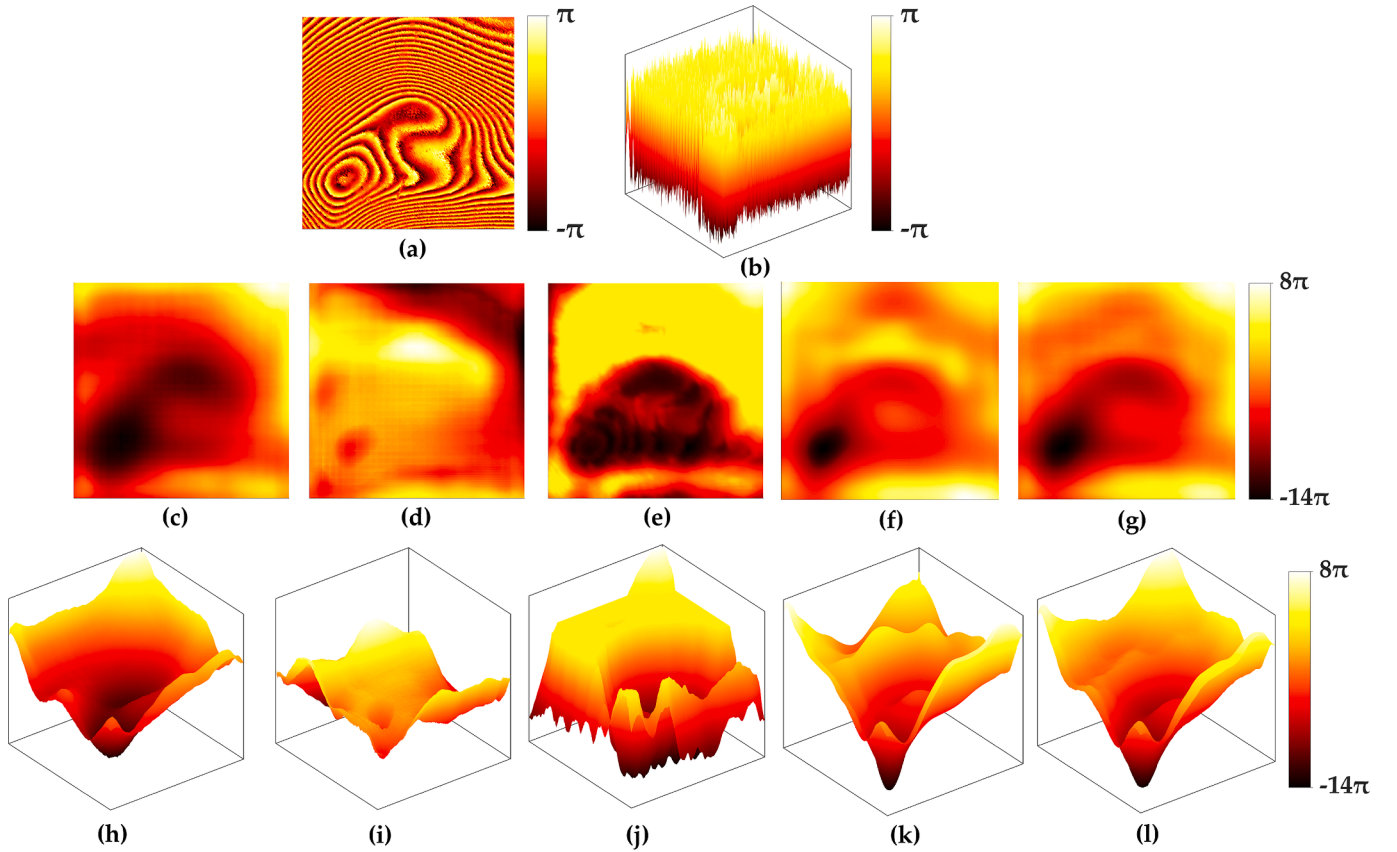


Fig. 15. Experimental results of the phase unwrapping made with a PDMS sample. (a) Experimentally obtained phase image. It is wrapped and highly noisy (6.52 dB SNR). (b) 3D profile of the wrapped phase image. The phase images unwrapped with (c) DenSFA-PU, (d) DLPU, (e) VUR-Net, (f) U²-Net, and (g) Hybrid Net. (h)-(l) The 3D plots of corresponding 3D surface profiles. Each image has 256×256 pixels.

Net (f) and Hybrid Net (g) delivered appreciable results, with the unwrapping time of 46.31 ms and 45.94 ms, respectively. Other methods, such as DLPU and VUR-Net, failed to effectively unwrap the phase, leaving residual errors, with unwrapping times of 37.79 ms and 40.12 ms, respectively. The 3D profiles in Fig. 15(h)-(l) demonstrate that our method produced a smooth and useful profile, while the others generated irregular shapes and lacked meaningful information. These results highlight the proposed method's ability to denoise while preserving details, making it highly effective for unwrapping even real-world data. This work primarily focuses on the phase unwrapping methods rather than analyzing the physical surfaces with the reconstructed images. However, the phase maps in Figs. 14 and 15 unwrapped using the proposed DenSFA-PU, exhibit smooth and consistent profiles, reflecting the topological characteristics of typical acrylic and PDMS samples. A comparison with other advanced surface profiling modalities is expected to be conducted in the near future.

3.4. Ablation study

To evaluate the effectiveness of the proposed approach, an ablation study was conducted using synthetic data to validate the contribution of each component within the network. The study employed 500 images from synthetic test dataset, comprising 200 speckle noise images, 200 Gaussian noise images, and 100 noise-free images. Firstly, we compared the performance of the proposed method trained with various loss functions ($\mathcal{L}_2$, $\mathcal{L}_{tv}$, and $\mathcal{L}_{var}$). Then, we examined the impact of incorporating BAM attention mechanism in the SFA module. The findings show that DenSFA-PU trained with the composite loss function delivered the best performance across all evaluation metrics. Additionally, the inclusion of the BAM module in the SFA significantly enhanced

Table 6
Results of ablation study.

	NRMSE(%)	PSNR	SSIM
L_2 loss	1.34	56.624	0.989
$L_{var} + L_{tv}$ loss	0.84	67.262	0.998
SFA Without BAM	0.89	65.283	0.991
Composite loss and SFA with bi-LSTM and BAM	0.65	70.676	1.000

the performance. The average NRMSE, PSNR, and SSIM results for these experiments are shown in Table 6. Overall, these ablation experiments demonstrate that both the composite loss function and the BAM module play crucial roles, particularly in challenging noisy conditions.

4. Discussion and conclusion

We propose DenSFA-PU (Densely Connected Spatial Feature Aggregator for Phase Unwrapping) as a robust and efficient network for phase unwrapping, particularly effective in handling uneven and noisy conditions. Our method takes advantage of dense connectivity for enhanced feature propagation and integrates the spatial feature aggregator (SFA) module to capture spatial dependencies, ensuring that the network focuses on critical areas. In the SFA module, by combining bi-LSTM layers with a lightweight bottleneck attention module (BAM), the network efficiently captures long-range dependencies and focuses on the most relevant features. BAM's bottleneck structure reduces both the number of parameters and computational overhead, which is crucial when processing large feature maps generated by the bi-LSTM layers, while maintaining strong performance. A composite loss function, combining the mean squared error, variance of error, and total

variation, is applied to optimize the performance. The superior performance of our method is due to two main factors: (i) accurately capturing fine geometric details by treating phase unwrapping as a regression problem, and (ii) employing dense blocks with the SFA module for adaptive estimation in unreliable regions. This approach allows precise unwrapping without requiring pre- or post-processing, even under severe noisy conditions.

To validate our approach, we have conducted experiments using both synthetic and real-world phase data sets, under different Gaussian and speckle noise conditions. Our network demonstrated superior unwrapping performance compared to other five representative phase unwrapping networks: the conventional QPF method, and four deep learning-based approaches DLPU, VUR-Net, U²-Net, and Hybrid Net. The results showed that DenSFA-PU achieved high accuracy, structural stability, and robustness, particularly in noisy environments. It consistently outperformed other models in terms of NRMSE, PSNR, and SSIM metrics, confirming the network's generalization ability. The experimental results clearly showed the effectiveness of DenSFA-PU on real-world samples, enhancing both SNR and stability. This highlighted its utility and adaptability for effective phase unwrapping tasks frequently met in numerous real-world applications. It is expected that it can find good applications even in the field of gas-insulated switchgear (GIS). The pattern of vibration caused by partial discharges in a GIS chamber can be monitored on its surface with an optical interferometer.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

CRediT authorship contribution statement

Muhammad Awais: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Taeil Yoon:** Writing – review & editing, Visualization, Methodology, Data curation, Conceptualization. **Chi-Ok Hwang:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Byeongha Lee:** Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Informed consent

There are no individual participants included in the study.

Funding

This work was supported by the Technology Innovation Program (20021979) funded by the Ministry of Trade, Industry and Energy (MOTIE), the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2024-00461205), and the GIST Research Project grant funded by GIST in 2024.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.optlastec.2025.112757>

Data availability

Data will be made available on request.

References

- [1] R.M. Goldstein, H.A. Zebker, C.L. Werner, Satellite radar interferometry: Two-dimensional phase unwrapping, *Radio Sci.* 23 (4) (1988) 713–720.
- [2] P.R. Sukumar, R.G. Waghmare, R.K. Singh, G.R.K.S. Subrahmanyam, D. Mishra, Phase unwrapping with kalman filter based denoising in digital holographic interferometry, *ICACCI* 2256–2260 (2015).
- [3] J. Dong, T. Liu, F. Chen, D. Zhou, A. Dimov, A. Raj, Q. Cheng, P. Spincemaille, Y. Wang, Simultaneous phase unwrapping and removal of chemical shift (spurs) using graph cuts: Application in quantitative susceptibility mapping, *IEEE Trans. Med. Imaging* 34 (2) (2015) 531–540.
- [4] S. Zhang, Absolute phase retrieval methods for digital fringe projection profilometry: A review, *Opt. Lasers Eng.*, 107 (Aug. 2018) 28–37.
- [5] R. G. Waghmare, P. R. Sukumar, G. R. K. S. Subrahmanyam, R. K. Singh, and D. Mishra, "Particle-filter-based phase estimation in digital holographic interferometry," *J. Opt. Soc. Amer. A, Opt. Image Sci.*
- [6] V. Katkovnik, J. Astola, K. Egiazarian, Phase local approximation (PhaseLa) technique for phase unwrap from noisy data, *IEEE Trans. Image Process.*, 17 (6) (Jun. 2008) 833–846.
- [7] M. Zhao, L. Huang, Q. Zhang, Su. Xianyu, A. Asundi, Q. Kemao, Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies, *Appl. Opt.*, 50 (33) (Nov 2011) 6214–6224.
- [8] A. Oppenheim, R. Schaffer, *Digital Signal Processing* (prentice-Hall (1975) 507–511.
- [9] A. Baldi, Two-dimensional phase unwrapping by quad-tree decomposition, *Appl. Opt.* 40 (8) (2001) 1187–1194.
- [10] K. Hung, T. Yamada, Phase unwrapping by regions using least-squares approach, *Opt. Eng.* 37 (11) (1998) 2965–2970.
- [11] D.C. Ghiglia, L.A. Romero, Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods, *J. Opt. Soc. Am. A* 11 (1994) 107–117.
- [12] M. Arevalillo-Herráez, D.R. Burton, M.J. Lalor, M.A. Gdeisat, Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a non-continuous path, *Appl. Opt.* 41 (35) (2002) 7437–7444.
- [13] K. O'Shea, R. Nash, An introduction to convolutional neural networks. *arXiv Preprint arXiv:1511.08458*, 2015.
- [14] Kaiqiang Wang, Qian Kemao, Jianglei Di, Jianlin Zhao, "Deep learning spatial phase unwrapping: a comparative review," *Adv. Photon. Nexus* 1(1) 014001 (3 August 2022).
- [15] R. Liang et al., "Phase unwrapping using segmentation," *Provisional Patent Application patent 62/768,624* (2018).
- [16] G.E. Spoorthi, S. Gorthi, R.K.S.S. Gorthi, PhaseNet: a deep convolutional neural network for two-dimensional phase unwrapping, *IEEE Signal Process. Lett.* 26 (1) (2018) 54–58.
- [17] J. Zhang, et al., Phase unwrapping in optical metrology via denoised and convolutional segmentation networks, *Opt. Express* 27 (10) (2019) 14903.
- [18] T. Zhang, et al., Rapid and robust two-dimensional phase unwrapping via deep learning, *Opt. Express* 27 (16) (2019) 23173.
- [19] C. Wu, et al., Phase unwrapping based on a residual en-decoder network for phase images in Fourier domain Doppler optical coherence tomography, *Biomed. Opt. Express* 11 (4) (2020) 1760.
- [20] Z. Zhao, et al., Phase unwrapping method for point diffraction interferometer based on residual auto encoder neural network, *Opt. Lasers Eng.* 138 (2020) 106405.
- [21] S. Zhu, et al., Phase unwrapping in ICF target interferometric measurement via deep learning, *Appl. Opt.* 60 (1) (2021) 10.
- [22] K.S. Vengala, N. Paluru, R.K.S.S. Gorthi, 3D deformation measurement in digital holographic interferometry using a multitask deep learning architecture, *J. Opt. Soc. Am. A* 39 (1) (2022) 167.
- [23] K.S. Vengala, V. Ravi, G.R.K.S. Subrahmanyam, A multitask learning for 2D phase unwrapping in fringe projection, *IEEE Signal Process. Lett.* 29 (2022) 797–801.
- [24] J. Zhang, Q. Li, EESANet: edge-enhanced self-attention network for two-dimensional phase unwrapping, *Opt. Express* 30 (7) (2022) 10470.
- [25] K. Wang, Y. Li, Q. Kemao, J. Di, J. Zhao, One-step robust deep learning phase unwrapping, *Opt. Express* 27 (10) (May 2019) 15100–15115.
- [26] G. Dardikman-Yoffe, et al., PhUn-Net: ready-to-use neural net- work for unwrapping quantitative phase images of biological cells, *Biomed. Opt. Express* 11 (2) (2020) 1107.
- [27] Spoorthi, G. E., Rama Krishna Sai Subrahmanyam Gorthi, and Subrahmanyam Gorthi. "PhaseNet 2.0: Phase unwrapping of noisy data based on deep learning approach." *IEEE transactions on image processing* 29 (2020): 4862-4872.
- [28] Y. Qin, et al., Direct and accurate phase unwrapping with deep neural network, *Appl. Opt.* 59 (24) (2020) 7258.
- [29] M. V. Perera and A. De Silva, "A joint convolutional and spatial quad-directional LSTM network for phase unwrapping," in *ICASSP*, pp. 4055–4059 (2021).
- [30] S. Park, Y. Kim, I. Moon, Automated phase unwrapping in digital holography with deep learning, *Biomed. Opt. Express* 12 (11) (2021) 7064.
- [31] L. Zhou, et al., PU-GAN: a one-step 2-D InSAR phase unwrapping based on conditional generative adversarial network, *IEEE Trans. Geosci. Remote Sens.* 60 (2022) 1–10.

- [32] H. Zhou, et al., The PHU-NET: a robust phase unwrapping method for MRI based on deep learning, *Magn. Reson. Med.* 86 (6) (2021) 3321–3333.
- [33] M. Xu, et al., PU-M-Net for phase unwrapping with speckle reduction and structure protection in ESPI, *Opt. Lasers Eng.* 151 (2022) 106824.
- [34] S. Park, Y. Kim, I. Moon, Automated phase unwrapping in digital holography with deep learning, *Biomed. Opt. Express* 12 (2021) 7064–7081.
- [35] M. Gontarz, V. Dutta, M. Kujawińska, W. Krauze, Phase unwrapping using deep learning in holographic tomography, *Opt. Express* 31 (2023) 18964–18992.
- [36] J. Zhao, L. Liu, T. Wang, X. Wang, X. Du, R. Hao, J. Liu, Y. Liu, J. Zhang, VDE-Net: a two-stage deep learning method for phase unwrapping, *Opt. Express* 30 (2022) 39794–39815.
- [37] H. He, et al., UN-PUNet for phase unwrapping from a single uneven and noisy ESPI phase pattern, *JOSA A* 40 (10) (2023) 1969–1978.
- [38] J. Chen, et al., Two-dimensional phase unwrapping based on U²-Net in complex noise environment, *Opt. Express* 31 (18) (2023) 29792–29812.
- [39] D. Li, X. Xie, Deep Learning-Based Phase Unwrapping Method, *IEEE Access* 11 (2023) 85836–85851.
- [40] Z. Zhang, et al., Efficient and robust phase unwrapping method based on SFNet, *Opt. Express* 32 (9) (2024) 15410–15432.
- [41] B. Wang, et al., An Anti-Noise-Designed Residual Phase Unwrapping Neural Network for Digital Speckle Pattern Interferometry, *Optics* 5 (1) (2024) 44–55.
- [42] W. Zhao, et al., C-HRNet: High Resolution Network Based on Contexts for Single-Frame Phase Unwrapping, *IEEE Photonics J.* (2024).
- [43] Y. Li, et al., PUDCN: two-dimensional phase unwrapping with a deformable convolutional network, *Opt. Express* 32 (16) (2024) 27206–27220.
- [44] K.G. Bujagouni, S. Pradhan, Transformer based deep learning hybrid architecture for phase unwrapping, *Phys. Scr.* 99 (7) (2024) 076017.
- [45] A. Graves, A., Mohamed, A. R., & Hinton, G., May). Speech recognition with deep recurrent neural networks, in: *In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.
- [46] J. Park, S. Woo, J.Y. Lee, et al., A Simple and Light-Weight Attention Module for Convolutional Neural Networks, *Int J Comput vis* 128 (2020) 783–798.
- [47] G. Huang, Z. Liu, K.Q. Weinberger, Densely Connected Convolutional Networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2017 (2016) 2261–2269.
- [48] H. Hongxing, W. Lingda, PUMA-SPA: A Phase Unwrapping Method Based on PUMA and Second-Order Polynomial Approximation, *IEEE Geosci. Remote Sens. Lett.* 11 (11) (Nov. 2014) 1906–1910.
- [49] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, arXiv:1502.03167.
- [50] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. 2015.
- [51] C. Sean Bell, L. Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016.