Fringe-pattern analysis with ensemble deep learning: Supplementary Material

Shijie Feng[a,b,c], **Yile Xiao**[a,b,c], **Wei Yin**[a,b,c], **Yan Hu**[a,b,c], **Yixuan Li**[a,b,c], **Chao Zuo**[a,b,c,*], **Qian Chen**[a,b,*]

[a]Smart Computational Imaging (SCI) Laboratory, Nanjing University of Science and Technology, 200 Xiaolingwei Street, Nanjing, China, 210094

[b]Jiangsu Key Laboratory of Spectral Imaging Intelligent Sense, Nanjing University of Science and Technology, 200 Xiaolingwei Street, Nanjing, China, 210094

[c]Smart Computational Imaging Research Institute (SCIRI) of Nanjing University of Science and Technology, 8 Jialingjiang Eastern Street, Nanjing, China, 210019

Abstract. This document provides the information about the system set-up, the generation of ground-truth data, architectures of the base models, and other supplementary contents to the primary manuscript "Fringe-pattern analysis with ensemble deep learning".

*Chao Zuo, zuochao@njust.edu.cn *Qian Chen, chenqian@njust.edu.cn

1 Optical system and the generation of ground-truth data

The fringe projection system consists of a projector with a resolution of 1024×768 (DLP 4100, Texas Instruments) and a camera with a resolution of 1280×800 (V611, Vision Research Phantom). The projector illuminates test objects with pre-designed fringe patterns and the camera captures the images simultaneously from a different perspective. The captured fringe patterns are 8-bit grayscale images. In the data pre-processing stage, the input fringe pattern was normalized before being fed into the deep neural networks. The schematic is shown in Fig. S1.

To generate ground-truth labels, the 12-step phase-shifting (PS) algorithm was applied. The projected 12-step PS fringe patterns can be written as

$$I_n^p(x^p, y^p) = a + b\cos\left(2\pi f x^p - \frac{2\pi n}{N_{ps}}\right), \tag{S1}$$

where (x^p, y^p) is the pixel coordinate of the projector, $N_{ps} = 12$, and $n = 0, 1, 2, ..., 11$. Parameters a, b, f are the DC component, amplitude and spatial frequency of the fringe pattern, respectively.

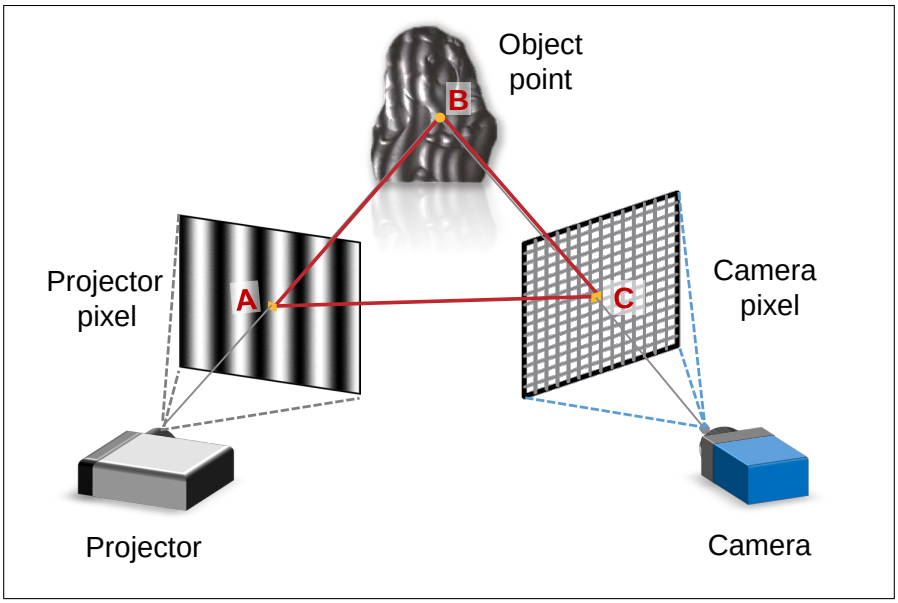

Fig S1 Schematic of the fringe projection system.

The spatial frequency of the projected fringes is $f = 160$ in our experiments. These PS patterns are projected onto a test object and the captured images can be expressed as

$$I_n(x,y) = A(x,y) + B(x,y)\left[\varphi(x,y) - \frac{2\pi n}{N_{ps}}\right], \tag{S2}$$

where (x, y) is the pixel coordinate of the camera, A the background signal, B the amplitude, and φ the phase of test objects. The ground-truth numerator and denominator can be calculated as

$$M(x,y) = \sum_{n=0}^{N_{ps}-1} I_n(x,y)\sin\left(\frac{2\pi n}{N_{ps}}\right), \tag{S3}$$

$$D(x,y) = \sum_{n=0}^{N_{ps}-1} I_n(x,y)\cos\left(\frac{2\pi n}{N_{ps}}\right). \tag{S4}$$

2 The structures of the base models

2.1 The structure of the U-Net

The first base model is U-Net and it structure is shown in Fig. S2. The input is a fringe image and we train the model to learn to predict the numerator M and the denominator D. The input fringe image in our experiments has resolution of $W \times H$. It is then processed by the encoder, where C channels of features are extracted from the fringe pattern ($C = 50$ in our experiments) by the convolutional layer which is activated by the rectified linear unit (ReLU) as activation function, i.e., $\mathrm{ReLU}(x) = \max(0, x)$. For deep convolutional layers, more channels (i.e., from 2C to 16C) are utilized to extract the information of output data of the previous layer. The extracted features are then down-sampled by 1/2 along both x and y directions. With the next several similar down-sampling convolutional blocks that extract more image details at different scales, high-level features of $\frac{W}{16} \times \frac{H}{16} \times 16C$ are obtained by the encoder at last. They are then handled by the decoder that performs up-sampling to synthesize and recover the final results of the input image's original size. The up-sampling is carried out by bilinear interpolation, which is followed by some convolutional layers. A skip connection can be found at each step of the decoder, which is used to concatenate the convolutional layers' output with feature maps from the encoder at the same level. Features at different levels can be collected at the same time with this architecture. The last layer is a convolutional layer, which is activated linearly. The numerator and the denominator can be learned automatically by supervised learning.

2.2 The structure of the multi-path deep neural network

The second base model in our work is the multi-path deep neural network (MP DNN) and its structure is shown in Fig. S3. The labeled dimension of each layer or block indicates the size of

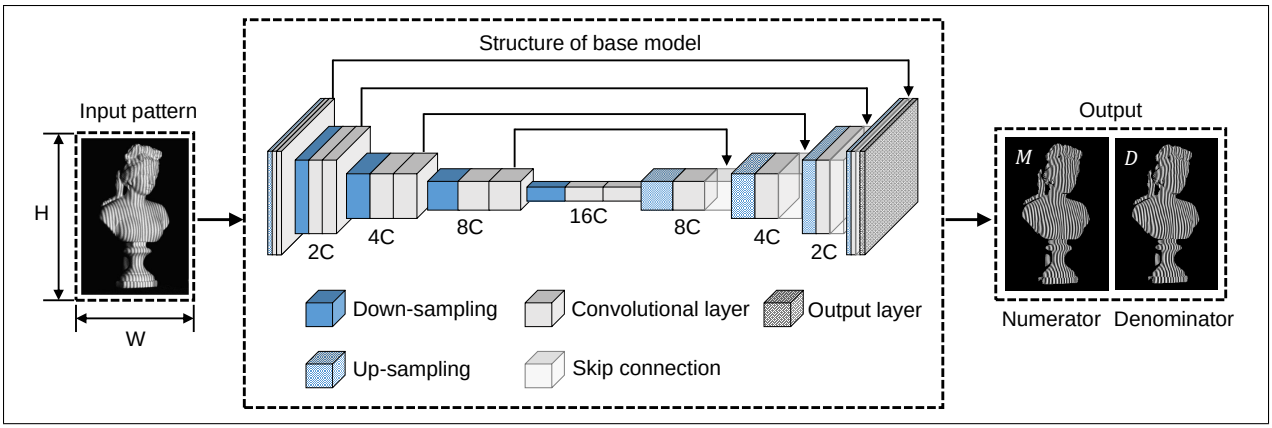

Fig S2 The architecture of U-Net used in our work. The network consists of an encoder and a decoder, which gives it the u-shaped architecture. The input is a fringe pattern and the output is a two-channel tensor that consists of a numerator and a denominator.

the output data. The input of the network is a fringe image. The size of the input image is $W \times H$. Four data-flow paths are constructed to process the input images at different scales. In the first path which keeps the original size of input data, the fringe images are successively processed by a convolutional layer, a group of residual blocks and another convolutional layer. C is the number of filters used in the convolutional layer and equals the number of channels of output data. Each filter is used to extract a feature map (channel) for the output tensor. The same input data also undergoes similar but more sophisticated procedures in the second to the fourth paths where the data are first downsampled by $\times 2$, $\times 4$, and $\times 8$ for high-level perceptions and then upsampled to match the original dimensions. Eventually, the results of each data-flow path are concatenated to produce the final output. With the design of multi-scale data-flow paths, geometric details that the input images contain can be perceived precisely, ensuring the estimation of high-quality phase information.

2.3 The structure of the Swin-Unet

Due to the limited visual field of the convolution operation, it is hard for CNNs to learn global information of input data. To further increase the diversity of the base models and to learn global fea-

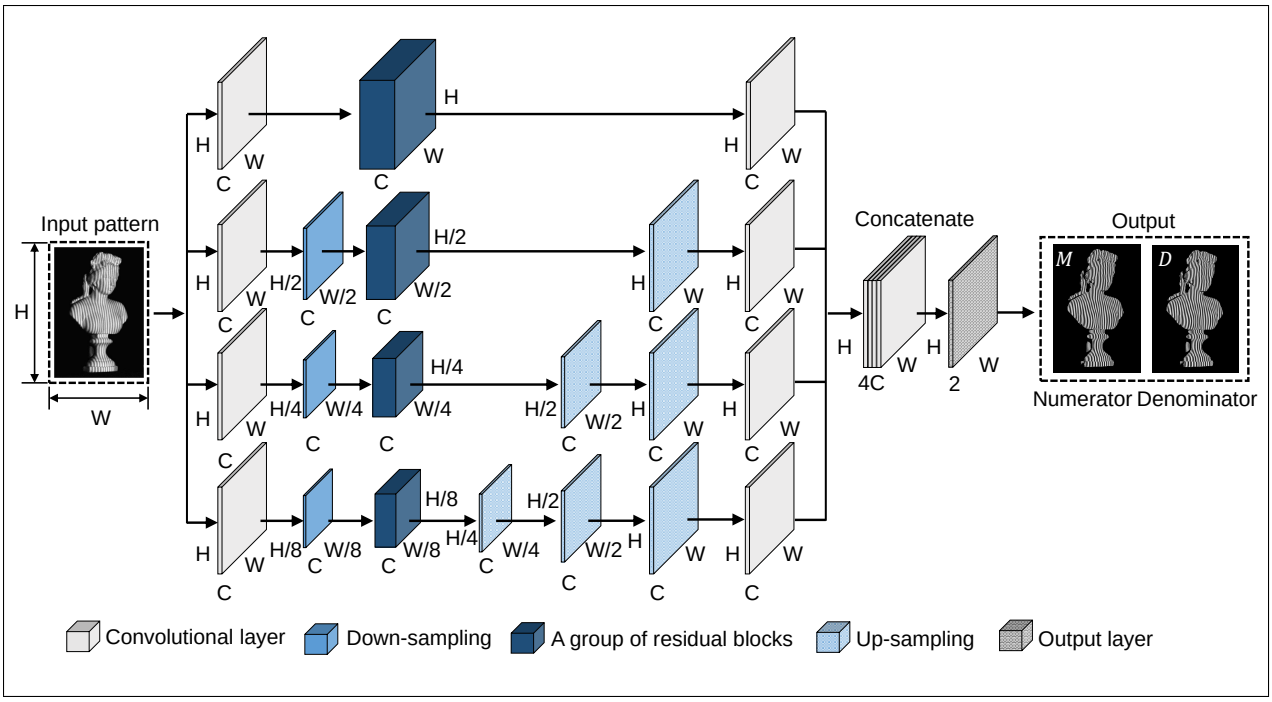

Fig S3 The architecture of the MP DNN in our work. It consists of several data-flow paths that can process the data in different scales in parallel. The input is a fringe pattern and the output is a two-channel tensor that consists of a numerator and a denominator.

tures, a state-of-the-art vision transformer is utilized as one of the base models. To be concrete, it is the Swin-Unet that is a pure Transformer-based U-shaped architecture and it is shown in Fig. S4. Different from the traditional CNN network, the Swin-Unet uses Transformer to replace the convolution layer. Also, the minimum processing unit is changed from pixels to patches. The operation between the original image and the convolution kernel is transformed into the calculation of the correlation between each patch in each window. The Swin Transformer block is used to build the encoder, bottleneck and decoder. Like other base models, the input of Swin-Unet is a fringe pattern of $H \times W$. It is then converted into several patches through the partition and embedding module. $P \times P$ pixels are embedded in a patch without overlapping. In the encoder, two consecutive Swin Transformer block modules learn features from tokenized inputs. The Swin Transformer block is composed of LayerNorm (LN) layers, multi-head self-attention module, residual connection and 2-layer MLP with GELU non-linearity. The window-based multi-head attention module (W-MSA)

performs attention calculation within the window.The moving-window-based multi-head attention module (SW-MSA) uses the sliding window method to achieve the information interaction between different windows. The patch merging module divides the input patches into four parts and concatenates them together, and up-samples the feature dimension to $2\times$ the original dimension by the linear layer. In the structure of the encoder, the Swin Transformer block module and the patch merging module are cyclically arranged. High-level features of $\frac{H}{8P} \times \frac{W}{8P} \times 8C$ are obtained by the encoder at last. After the bottleneck consisting of two Swin Transformer block modules, we use the patch expand module in the decoder to up-sample the extracted deep features. The patch expanding layer reshapes the feature maps of adjacent dimensions into large feature maps with $2\times$ up-sampling of resolution. The output of the patch expanding module and the multi-scale features of the encoder of the same level are fused through skip connections. The fusion features of different layers are collected by this structure, and finally the feature map is up-sampled to the original resolution of $H \times W$ through the patch reverse module, generating the numerator and the denominator.

3 Tests of different kinds of objects

Here, we measured more objects to verify the performance of our approach for unseen scenarios which include an intersection of a cone and a cylinder (Scenario 1), a pair of ceramic spheres (Scenario 2), a glasses case (Scenario 3), a scene that contains a face mask and a cylinder (Scenario 4), and a statue (Scenario 5). Figure S5 (a) shows the captured fringe patterns. A 7-fold average ensemble was used to generate seven groups of training data, which were used to train our base models. The adaptive ensemble was then used to combine the outputs of base models and make the final predictions. The predicted wrapped phase maps are shown in Fig. S5 (b). For comparison, a

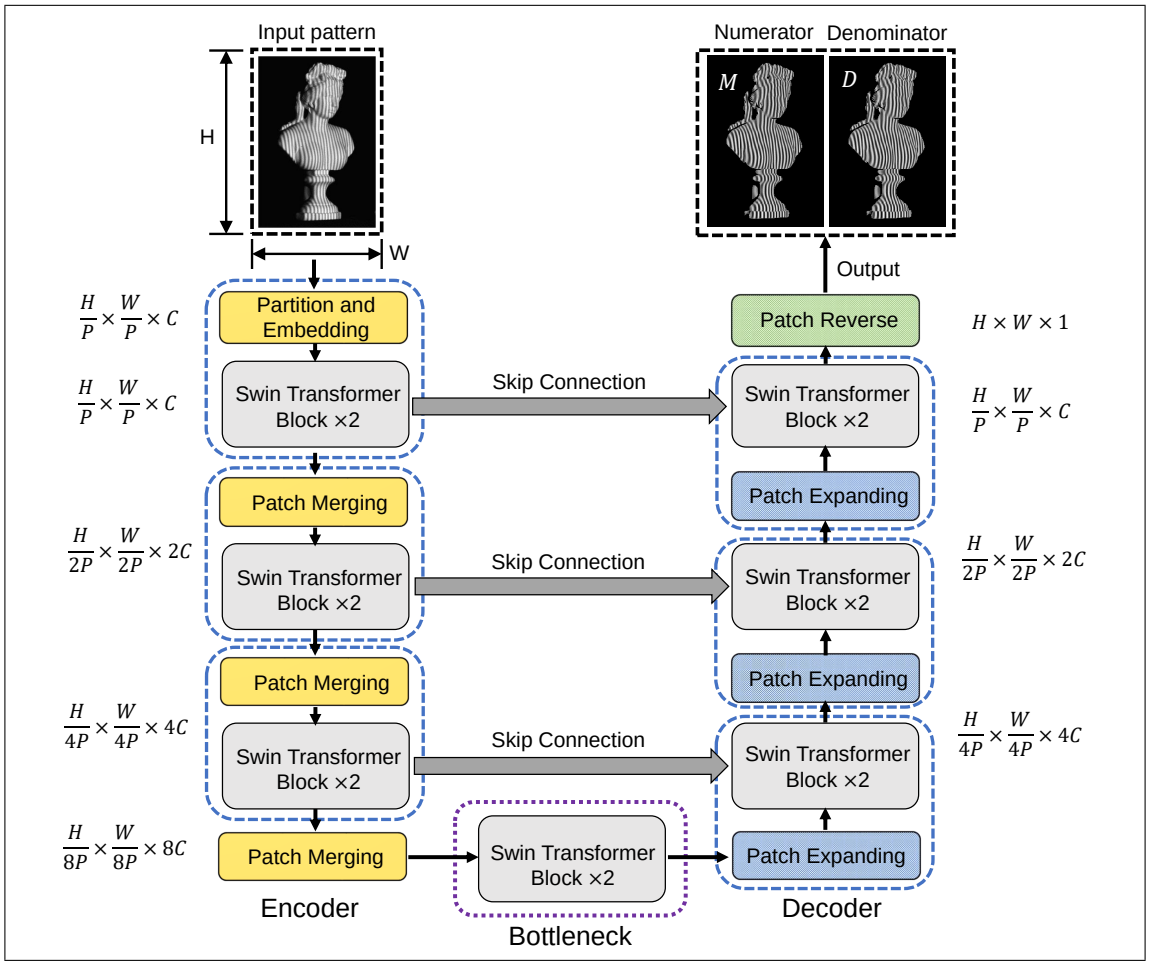

Fig S4 The architecture of the Swin-Unet. Different from the U-Net and the MP DNN which are convolutional neural networks, the Swin-Unet uses the transformer to replace the convolutional layers. The input is a fringe pattern and the output is a two-channel tensor that consists of a numerator and a denominator.

single U-Net was also trained to analyze these fringe patterns. Figures S5 (c) and (d) demonstrate the absolute phase errors of the U-Net and our method respectively. By comparing the overall error distributions, we can see that our method greatly reduces the phase errors for all these scenes. For detailed investigations, a region of interest (ROI) is selected from each scene. For scenarios 1 and 2, the edges of these objects have been measured with higher accuracy. For scenarios 3 and 4, the errors of the texture boundaries and varying depth regions were suppressed successfully by the proposed method. For the last scenario, there are many subtle details on the hair of the statue and they were measured with large errors by the single U-Net. By contrast, when several models were trained by our method to measure this area together, the accuracy has been improved significantly.

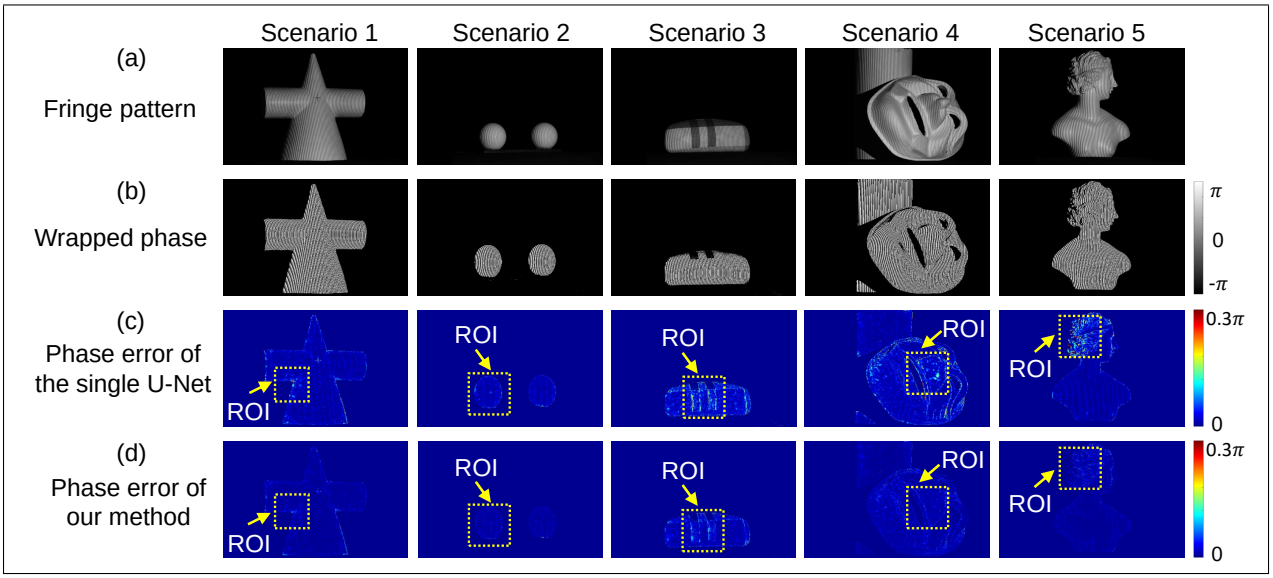

Fig S5 Tests of different unseen objects. (a) Input fringe patterns of an intersection of a cone and a cylinder, a pairs of ceramic spheres, a glasses case, a scene that contains a mask and a cylinder, and a statue. (b) The wrapped phase maps obtained by our method. (c) The absolute phase errors of the single U-Net. (d) The absolute phase errors of our method.

For quantitative investigations, we further compared the MAEs of the U-Net, the K-fold average ensemble, and the adaptive ensemble. The results are shown in Fig. S6. We can see that both the proposed K-fold average ensemble and the adaptive ensemble show smaller MAEs than the U-Net. As the adaptive ensemble trains a MultiResUNet to further combine the outputs ob-

tained by the average ensemble, it demonstrates the smallest phase error. Among these scenarios, the adaptive ensemble can decrease the MAE by at least 17% (i.e., for the second scenario). The reason is that the spheres consist of smooth areas which are not difficult to measure for the U-Net. For the fifth scenario which is the statue with complex shapes on the hair, the improvement is quite obvious, as indicated by decreasing the MAE by 28%. This experiment demonstrates that the generalization error of unseen objects can be reduced effectively by the proposed method using ensemble learning.

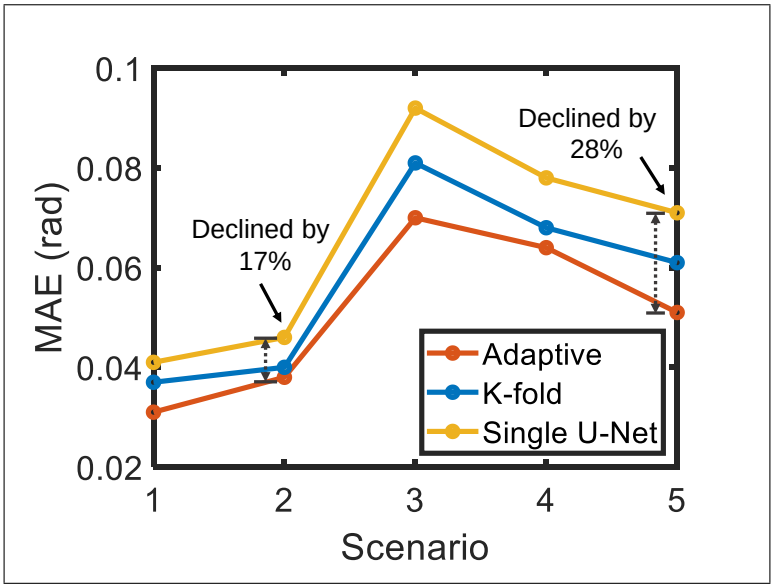

Fig S6 The mean absolute phase errors of the adaptive ensemble, the K-fold average ensemble, and the single U-Net for the tested scenarios.