

# Convolutional neural network optimisation to enhance ESPI fringe visibility

José Manuel Crespo\*  and Vicente Moreno

QMatterPhotonics Research Group, Optics Area, Department of Applied Physics, Faculty of Physics / Faculty of Optics and Optometry, University of Santiago de Compostela, 15782 Santiago de Compostela, Galicia, Spain

Received 31 January 2023 / Accepted 23 March 2023

**Abstract.** The use of convolutional neuronal networks (CNN) for the treatment of interferometric fringes has been introduced in recent years. In this paper, we optimize and build a CNN model, based U-NET architecture, to maximize its performance processing electronic speckle interferometry fringes (ESPI). The proposed approach is based on quick and light trainings to select the architecture parameters (network depth and kernel sizes) to maximize the performance of the neural network improving the visibility of ESPI images. To measure the performance, the structural similarity index (SSMI) will be the lead indicator, and the need for large datasets to train neural networks, unavailable for ESPI images, forces the use of a simulated ESPI image dataset along the process. This dataset is computed using Zernike polynomials to simulate local surface deformations in the specimen under test and simulated true speckle fields for the reference and object field involved in ESPI techniques.

**Keywords:** ESPI, Convolutional neural networks, Image denoising.

## 1 Introduction

Electronic speckle pattern interferometry (ESPI) has been used extensively since its inception in 1970 [1] in the field of optical metrology with applications to non-destructive testing, but the high frequency noise generated in the speckle interferometry process is a relevant caveat in its applications, as this noise is also the carrier of the information on surface displacement, the variable under study with these techniques. The right balance in the noise retrieval without compromise the encoded information on displacement, is a key success factor for the qualitative or quantitative interpretation of the results and industrial applications.

Over the years, to mitigate the low visibility and high noise level issues in ESPI interferometry, some techniques were developed and applied, including filtering process in the frequency or spatial domain, or Fourier transform based denoising techniques, or more traditional image processing filters like the median or low pass filters and its variations [2–4]. All of them applied to the resulting speckle image or the computed wrapped or unwrapped phase field.

Over the last few years, the continuous advances in artificial intelligence (AI) techniques and specialized hardware to run AI models, have eased their application to different areas of research, and more specifically to improve

image visualization, including interferometric images improvement.

Its applications to denoise interferometric images and clean the interference fringes has been broadly studied, revealing these techniques as a game changer to enhance the qualitative (and quantitative) results of interference data [5], and with an appropriate computational framework, a powerful and easy to use tool to enhance visibility of interference fringes.

However, even if the use of a specialised frameworks, like KERAS, PyTorch or Tensorflow, simplify the use of CNN and removes the implementation complexity, there are some initial decisions on neural network design that must be taken to select the network topology fitting to the specific problem to solve. Those initial decisions are, basically:

- (a) How to design a training dataset and
- (b) Decide the hyperparameters of the neural network: the number of layers and kernels to use.

To denoise ESPI fringe patterns, this paper proposes the use the U-NET architecture for the network topology [6], a powerful and easy to use type of convolutional neural networks broadly used in interferometric denoise, wrapped phase denoise and SAR image denoise [5]; and to address

\* Corresponding author: [josemanuel.crespo.continas@rai.usc.es](mailto:josemanuel.crespo.continas@rai.usc.es)

the decisions to take implementing the CNN, we propose to build a training dataset using simulated electronic speckle pattern interferometry following the method described by Goodman [7], and the use Zernike polynomials to simulate local deformations.

The hyperparameter selection is done using a grid search over the results of several models (trained with the same training dataset) applied to a small dataset containing sample images not used to train the models. This grid search will select the number of layers and number of kernels maximising the structural similarity index (SSMI) [8] between the input and reconstructed images.

### 1.1 Speckle pattern interferometry

Electronic speckle pattern interferometry (ESPI) is a technique used to measure from sub-micron to tenth of microns displacements in optically rough surfaces assessing the overlap of two speckle patterns. Since its initial uses in 1971, noise, and low visibility are the major drawbacks for this technique and the artificial intelligence has been proved as an effective technique to enhance and mitigate those major drawbacks. The optical systems in Figure 1 shows a simple implementation of ESPI interferometer system, where the speckle field coming from a diffuse specimen under test (test object) is interfered with a different speckle field arising from a rough surface acting as reference (diffuse reference).

The intensity distribution recorder by the detector in the imaging system follows:

$$I = I_r + I_o 2\sqrt{I_o I_r} \cos(\phi_r - \phi_o), \quad (1)$$

where  $I_o$ ,  $I_r$ ,  $\phi_o$ , and  $\phi_r$  are the amplitude and phase of the object and reference speckle waves at the detector. The ESPI fringe pattern is obtained from a pair of images,  $I$  and  $I'$ , recorded by the imaging system before and after deformation:

$$\begin{aligned} I &= I_r + I_o 2\sqrt{I_o I_r} \cos(\phi_r - \phi_o) \\ I' &= I_r + I_o 2\sqrt{I_o I_r} \cos(\phi_r - \phi_o - \varphi), \end{aligned} \quad (2)$$

where  $\varphi$  is the phase change induced in the object wave by a deformation in the test specimen.

The ESPI fringe pattern is finally obtained subtracting those images and the resulting intensity pattern is according to the formula [1]:

$$|I - I'| = \left| 4\sqrt{I_o I_r} \sin\left(\phi_o - \phi_r + \frac{\varphi}{2}\right) \sin\left(\frac{\varphi}{2}\right) \right|. \quad (3)$$

The resulting intensity field is proportional to the term  $|\sin(\varphi/2)|$ , revealing a fringe field where the minimum value for the resulting intensity (perfect correlation between  $I$  and  $I'$ ) happens where the value for the term  $|\sin(\varphi/2)|$  is  $\varphi = 2n\pi$  and the maximum value for this term occurs at  $\varphi = (2n + 1)\pi$ .

This behaviour reveals a correlation fringe field like interference patterns, the ESPI fringe pattern, used to measure sub-micron displacements in the illuminated structure. This resulting fringe pattern is related to  $\varphi$ , the optical path

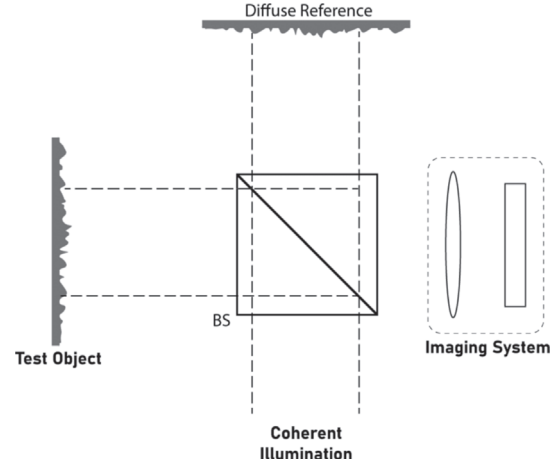


Fig. 1. Example setup for speckle pattern interferometry.

difference produced by the deformation in the specimen surface between frames  $I$  and  $I'$ .

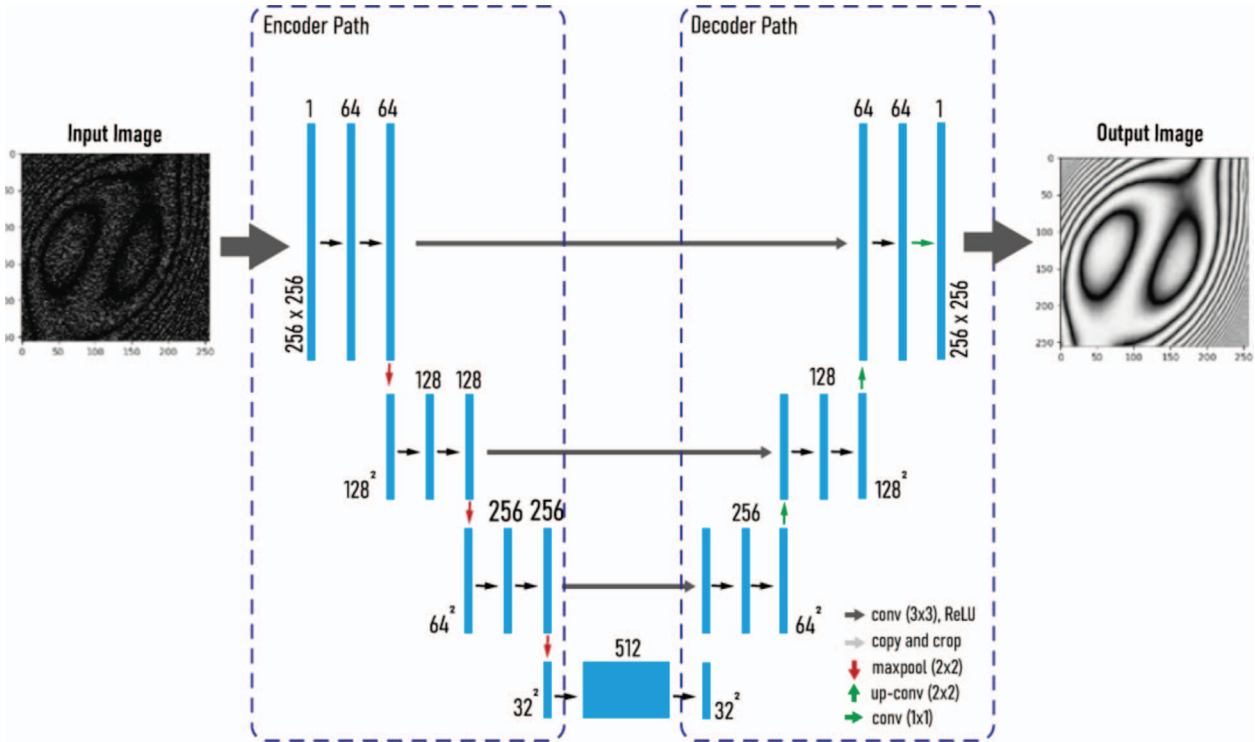
### 1.2 U-NET architecture

Our proposed design to denoise ESPI fringe fields, is the U-NET architecture [2], a powerful and easy to use type of convolutional neural networks, broadly used in interferometric denoise, wrapped phase denoise [5] and SAR image denoise. The U-NET is a convolutional encoder – decoder with internal connections between the encoding and decoding paths, where the noisy input image to be cleaned is introduced in the encoder path where 2D-convolutional (Conv2d) and maxpooling layers are combined to reduce the spatial resolution of the input image, capturing the image details (features).

Once the input image has been fully encoded, the decoder path reverses the operations done along the encoding phase and get a cleaned output image.

This architecture is depicted in Figure 2, where a 1 channel image of  $256 \times 256$  pixels feeds in the neural network and is encoded using convolutions while its spatial resolution is reduced with maxpooling operations. Along the encoding path, the number of channels of the processed images is increased by the convolution operations, resulting with a compressed image of  $32 \times 32$  pixels and 512 channels after all encoding operations. The spatial features of the input image are encoded along those 512 channels.

There are improvements introduced by the U-NET architecture over the classic autoencoder architectures [6]. In the classical autoencoder architecture, the information is compressed in a linear way along the encoder path reducing the dimensions of the input image and missing some features from the input image along the encoding path. The addition of “skip connections” to the decoding path in the U-NET architecture improves the network training and overall performance and overcomes the loss of information produced in the encoding phase using the outputs of the corresponding layer of the encoder path as additional input to the decoder layer.



**Fig. 2.** Example U-NET network with an input image of  $256 \times 256$  and 1 channel (B/N) and  $32 \times 32$  resolution after the full encoder path. The decoder path reverses the encoding operations and uses inputs from the corresponding encoder layer (skip connections) to end with a cleaned output image.

## 2 Network optimisation

To effectively apply artificial networks to solve a problem, there are some decisions on network architecture and hyperparameters to take in advance, at least *the selection of the model*, in our case U-NET, *the depth of the network* (the hidden layers in the decoding and encoding paths) and the *kernel size* to be used in the convolution operations along the network.

The selection of the hyperparameters will be based in a light and quick training using a small training dataset maximising one indicator measuring the network performance.

To assess the performance of the network, we use the SSMI [8]. The advantage of this index over other error measurements, like the MSE (mean squared error) or RMSE (root mean squared error) is that it measures the perceived increase of quality across the reconstructed image, removing the bias that the MSE and RMSE could have with ESPI images where there is a predominant random high frequency component (the speckle field) impacting those error measures.

### 2.1 Training dataset

The training procedure of artificial neural networks involves the use of large dataset of annotated images to train the network. While there is availability large image datasets in other research domains, like SAR or MRI, there are

not available datasets with ESPI images to train the network, making necessary build a dataset with simulated images for our training and optimisation procedures.

To build the training datasets used for our neural network optimisation and training, we use the formula (3) with three main parameters to simulate:  $I_o$ ,  $I_r$  and  $\varphi$ , being  $I_o$  and  $I_r$  corresponding to object and reference speckle fields and  $\varphi$  the optical path difference induced by a small deformation on the specimen surface.

There are common methods to simulate  $I_o$  and  $I_r$  generating pseudo-random numbers following a normal distribution over an interval  $[0, I_m]$  and  $[0, \rho I_m]$  with  $[I_m]$  a random number in the interval  $[0, 255]$  and  $\rho$  a normalised visibility parameter.

For our specific case, we implemented in Python the method described by Goodman [7] to simulate true speckle fields, simulating a random amplitude and phase for the object and reference speckle fields ( $I_o$ ,  $I_r$ ,  $\phi_o$ , and  $\phi_r$ ).

The pending parameter for the simulation is  $\varphi$ , the optical path difference introduced by a displacement in the test specimen. That surface deformation can be easily simulated using Zernike polynomials widely used in optics to represent surfaces or phase variation [5, 9]. Using the Zernike polynomials as the basis to simulate the deformation, the phase variation can be written as follows:

$$\varphi = \sum_0^n c_i Z_i \quad (4)$$

where  $c_i$  and  $Z_i$  are random coefficients and the  $i$ -Zernike polynomial respectively; and  $n$  is the order of the Zernike expansion.

Using this approach for  $\varphi$ , we can simulate any displacement field selecting the coefficients  $c_i$  and the Zernike polynomials used in the Zernike approximation, and moreover, introduces into the simulation the possibility to control the complexity, shape, and size of the resulted displacement just selecting the Zernike components to use in the equation (4) and the random coefficients  $c_i$ .

Following this procedure, we produced the datasets composed of 5000 pair of images  $[|I - I'|, |\sin(\varphi/2)|]$  to be used in the hyperparameter selection.

In Figure 3 some sample images from the training dataset are presented corresponding to different values of  $n$  for the  $\varphi$  simulation. If the  $n$  is increasing, more Zernike components are used in the simulated displacement and more complexity is shown in the  $\sin(\varphi/2)$  component in equation (3).

## 2.2 Hyperparameter selection

To select the depth and kernel size, we run several training procedures with a 5000 pair of  $256 \times 256$ -pixel monochrome images, each one with different set of hyperparameters. For the depth of the network (the levels in the encoding path) we checked values varying from 3 to 6 hidden layers in the encoding path and for kernel size we checked different squared kernels starting from a  $[3 \times 3]$  kernel finishing with the biggest kernel used in the optimisation procedure with a size of  $[7 \times 7]$  (Table 1).

This initial step builds 9 different candidate trained neural network models. In the next step of the optimisation, we select the one with the best performance as the final network architecture to use.

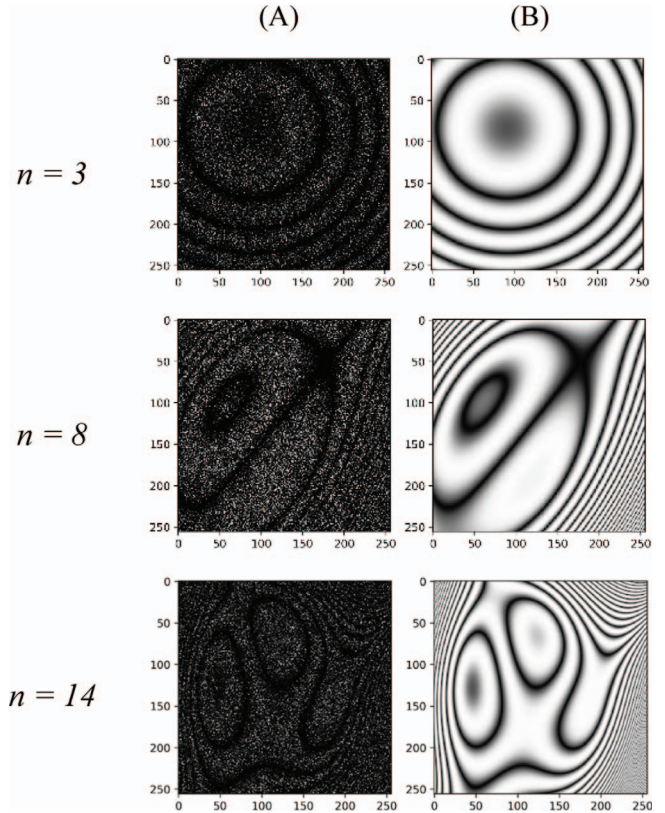
With a test dataset composed of 1000 random pairs of images, we run each one of the candidate models feeding simulated ESPI images, the equivalent to column (A) in Figure 3, comparing the output of the network with the expected output (or ground true image, the equivalent column (B) in Fig. 3).

For the comparison between the network output and expected output we selected the SSMI index, selecting the depth and kernel size combination with the biggest average SSMI along the whole test dataset as the final hyperparameters to use in the proposed network architecture.

For our specific case the selected outputs to define the final U-NET hyperparameters are a depth of 4 levels in the encoder path and a kernel size of  $5 \times 5$ . With this hyperparameter combination, the computed average SSMI index is 0.900 for the whole input dataset.

## 3 Model results

With the selected hyperparameters (kernel size =  $5 \times 5$  and layers = 4), we build and train the final neural network model using a new training dataset and finally check the model. To build this new training dataset, we followed



**Fig. 3.** Generated image samples. (A) Computed ESPI image using only the  $n = 3, 8$  and  $14$  first Zernike polynomials to simulate the complexity of specimen displacement. (B) Clean image or ground true, the  $\sin(\varphi/2)$  component used to generate image (A).

**Table 1.** Average SMMI value for the checked combinations of levels in the encoder path and kernel size.

|       |   | Kernel size  |              |              |
|-------|---|--------------|--------------|--------------|
|       |   | $3 \times 3$ | $5 \times 5$ | $7 \times 7$ |
| Depth | 4 | 0.896        | <b>0.900</b> | 0.801        |
|       | 5 | 0.798        | 0.760        | 0.728        |
|       | 6 | 0.880        | 0.859        | 0.760        |

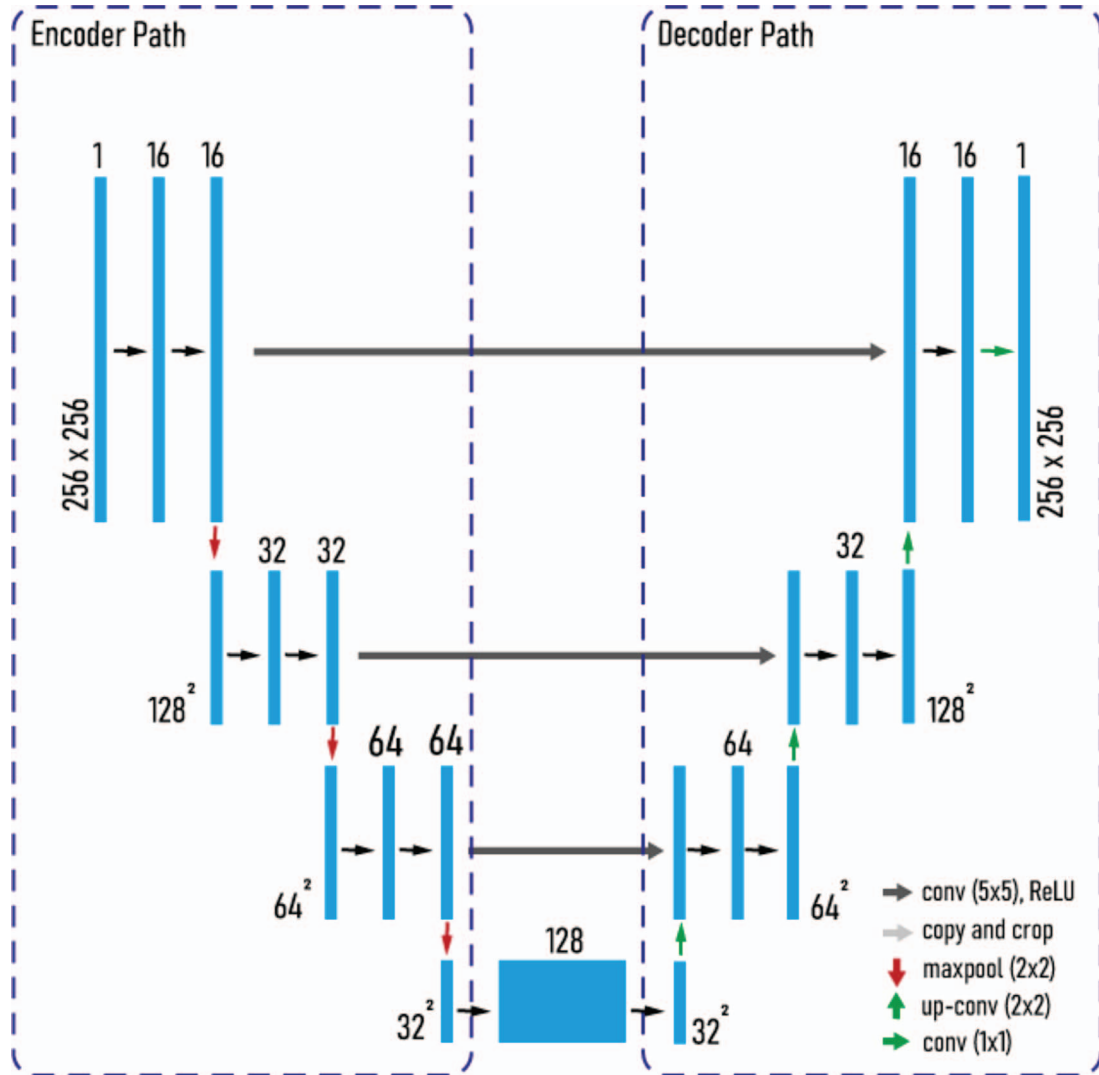
In bold, the combination with best performance.

the method described in Section 2.1 to simulate 15.000 pair of  $256 \times 256$ -pixel monochrome images.

The 15.000 generated images were divided into training and validation dataset, with 80% of image pairs for the training dataset, and 20% for the validation dataset.

The final training was implemented using simple options for the training: ADAM optimisation [10] and MEAN\_SQUARED\_ERROR as the loss function. We included an early stop condition of 3 epochs without improving the loss, to prevent overtraining.

The final network architecture is represented in Figure 4, and it is composed of 4 blocks along the encoding path, performing the convolution, activation and convolution



**Fig. 4.** U-NET network finally used concatenating blocks consisting of two  $5 \times 5$  convolutions (each one followed by a ReLU activation unit) and a  $2 \times 2$  maxpooling operation with stride = 2 for downsampling along the encoding path, reversing the operations along the decoder path using blocks composed of upsampling operations followed by a  $2 \times 2$  convolution (up-conv), concatenated with the corresponding output of the encoding part and followed by two  $5 \times 5$  convolutions (each one followed by a ReLU activation unit).

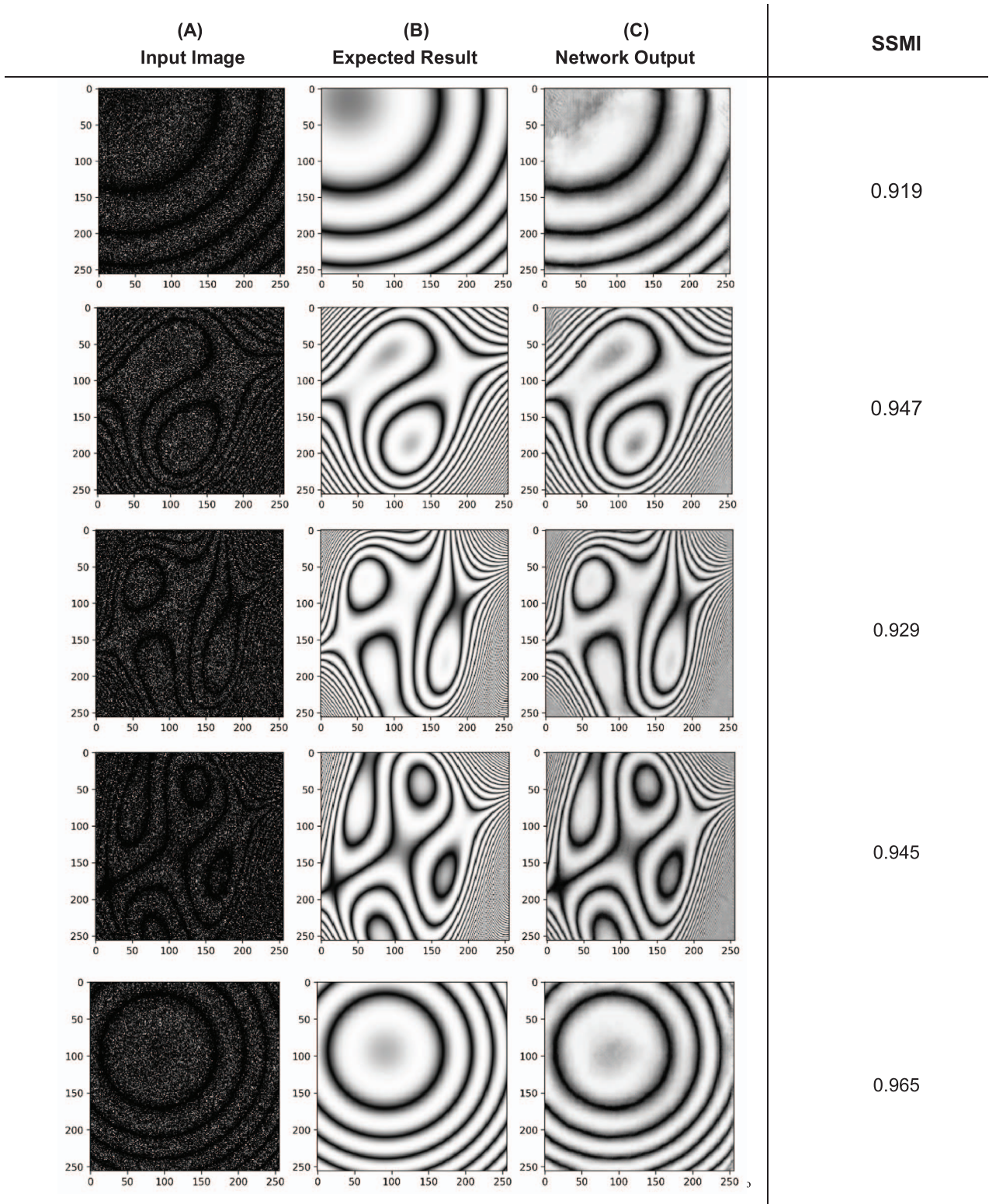
operations, using ReLU as the activation function for the layer and a  $5 \times 5$  kernel for the convolution operations, followed of a maxpooling operation with stride  $2 \times 2$ . Each step in the encoding path is reducing the dimension of the latent image, increasing (doubling) the feature channels.

The steps along the decoder path consist in an upsampling operation followed by a  $2 \times 2$  convolution to increase the size of the latent image concatenated with the corresponding output of the encoder path and two  $5 \times 5$  convolutions (each one followed by a ReLU activation unit).

All the computational steps were executed using a Google Collab instance with GPU support and after 23 min and 115 training epochs, the training automatically stopped reaching a loss value of 0.0084 on the validation dataset.

The average SSMI index between the ground true (expected output) and the reconstructed image using the network was 0.899, aligned with the expected value estimated in the hyperparameter selection step.

In Figure 5, some examples of the performance of the network on simulated images shown the performance of the artificial networks to enhance the visibility of ESPI images, improving the qualitative assessment of these fringe fields. The trained neural network processes the ESPI simulated image (A), improving the visibility of the fringe field (one of the biggest caveats in ESPI techniques) providing a clean image (C) facilitating qualitative assessment and further image processing techniques. The computed SSMI index between (B) and (C) shows quantitatively high similarity between expected image and processed image.



**Fig. 5.** Samples of cleaned images using the selected hyperparameters. (A) Input image. (B) Expected output (Ground true). (C) Processed image by the U-NET. The SSMI index is computed using expected and processed image (columns B and C).

## 4 Conclusion

We presented a simple approach to select the depth and kernel size to apply U-NET neural networks to denoise ESPI images to maximise its performance, and the resulting trained network improves the perceived quality of the ESPI fringe field.

The use of synthetic generated datasets removes the caveat of having large image collections to train the network models, and the generated datasets using Zernike polynomials to simulate surface displacements, can be adjusted to the specific case under study, improving the network performance. With the described method, the generated dataset can be customised in terms of image size, speckle size and interferometric setup, to match the experimental conditions and secure better performance for ESPI applications.

The use of specialised hardware, like GPUs or TPUs, and software frameworks like KERAS or PyTorch in cloud environments, ease the application of artificial neural network models to ESPI interferometry and make possible to work remotely, without continuous access to lab to develop the models.

Moreover, the software packaging of those software frameworks, running in standard IT equipment and low-cost GPUs contributes to decrease the associated costs.

## Conflict of interest

The authors declare no conflict of interest.

## References

- 1 Leendertz J.A. (1970) Interferometric displacement measurement on scattering surfaces utilizing speckle effect, *J. Phys. E: Sci. Instrum.* **3**, 3, 214.
- 2 Tounsi Y., Kumar M., Nassim A., Mendoza-Santoyo F. (2018) Speckle noise reduction in digital speckle pattern interferometric fringes by nonlocal means and its related adaptive kernel-based methods, *Appl. Opt.* **57**, 27, 7681–7690.
- 3 Aebischer H.A., Waldner S. (1999) A simple and effective method for filtering speckle-interferometric phase fringe patterns, *Opt. Commun.* **162**, 4–6, 205–210.
- 4 Kemao Q., Soon S.H. (2005) Two-dimensional windowed Fourier frames for noise reduction in fringe pattern analysis, *Opt. Eng.* **44**, 7, 075601–075601.
- 5 Zuo C., Qian J., Feng S., Yin W., Li Y., Fan P., Han J., Qian K., Chen Q. (2022) Deep learning in optical metrology: A review, *Light: Sci. Appl.* **11**, 1, 39.
- 6 Ronneberger O., Fischer P., Brox T. (2015) U-NET: Convolutional networks for biomedical image segmentation, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, October 5–9, Munich, Germany.
- 7 Goodman J.W. (2020) *Speckle phenomena in optics*, SPIE Press, Bellingham, WA, USA.
- 8 Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. (2004) Image quality assessment: From error visibility to structural similarity, in: *IEEE Transactions on Image Processing*, IEEE, pp. 600–612.
- 9 Born M., Wolf E. (1999) *Principles of optics*, Cambridge University Press, London.
- 10 Diederik P., Kigma J. (2014) *Adam: A method for stochastic optimization*. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).