EPJ Web of Conferences **266**, 13007 (2022) https://doi.org/10.1051/epjconf/202226613007
EOSAM 2022

Convolutional neural network optimisation to enhance ESPI fringe visibility

J.M. Crespo[1,*], and *V. Moreno*[1]

[1] QMatterPhotonics Research Group. Optics Area. Department of Applied Physics. Faculty of Physics / Faculty of Optics and Optometry. University of Santiago de Compostela. E-15782 Santiago de Compostela. Galicia. Spain

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).

1 Introduction

Over the last few years, continuous advances in artificial intelligence techniques and specialized hardware for their use have eased their application to different areas of research, and more specifically to the application of these techniques to improvements in image visualization.
Its applications to denoise interferometric images to clean the interference fringes has been broadly studied, being these techniques a disruptive tool to use and enhance the qualitative results of interference data.

Specifically, the convolutional neural networks using U-NET architectures with the appropriate computational framework, are powerful and easy to use tools to enhance visibility of interference fringes. However, some initial decisions on the neural network design must be taken and the final network architecture must be designed to fit the problem to solve. Namely the design of the training dataset and the hyperparameters of the neural network: the layers and the kernels to be used.

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 ESPI fringe pattern is obtained from a pair of images, I and I', taken from a specimen under study before and after deformation. The fringe pattern is obtained subtracting those images and the resulting intensity pattern is according to the formula [1]

$$|I - I'| = \left|4\sqrt{I_o I_r}\sin\left(\phi_o - \phi_r + \frac{\varphi}{2}\right)\sin\left(\frac{\varphi}{2}\right)\right| \qquad (1)$$

with I_o, I_r, ϕ_o, and ϕ_r are the amplitude and phase of the object and reference speckle fields, and φ is the optical phase difference induced by the deformation in the surface of the specimen under study. The minimum value for the resulting intensity occurs due to the term $\left|\sin\left(\frac{\varphi}{2}\right)\right|$ where $\varphi = n2\pi$ and the maximum value for this term at $\varphi = (n+1)2\pi$, revealing the ESPI fringe patterns which can be used to measure sub-micron displacements in the structure.

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 [3], wrapped phase denoise [3] and SAR image denoise

The U-NET is a convolutional encoder – decoder with internal connections between the encoding and decoding paths, where the noisy 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 detail (features). Once the image has been fully encoded, there is a decoder path reversing the encoding operations to get a cleaned image.

2 Network optimisation

The optimisation of the network design is reduced to the selection of the number of layers in the network and the size of the kernel to be used in the Conv2D operations accordingly to the problem to solve. The selection of the hyperparameters will be based in a small and quick training using a small training dataset maximising the

* Corresponding author: josemanuel.crespo.continas@rai.usc.es

indicator used to measure the network performance. In our case, the Structural Similarity Index (SSMI) [4], commonly used indicator for the quantitative assessment of the perceived increase of quality across the reconstructed image.

2.1 Training dataset

To build the training dataset to be used for the hyperparameter selection, we use the formula (1) where for the estimation of φ we use random Zernike polynomials to build a random phase, $\varphi = \sum_0^n c_i Z_i$, where c_i and Z_i are a random coefficient and the i-Zernike polynomial respectively and n is the order of the Zernike expansion.

Using this approach, any displacement field can be simulated, setting the proper interval for c_i and moreover, selecting the order of the Zernike expansion will control the complexity of the simulated deformation.

To simulate the interfering true speckle fields, we use the method described by J. Goodman [1]. Following this procedure, we produced 5000 pair of images $[|I - I'|, \varphi]$ to be used in the selection procedure.

2.2 Hyperparameter selection

To select the depth and kernel size, we run several trainings with a 5000 pair of images in the training dataset, each one with different set of hyperparameters, levels in the encoding path varying from 3 to 6 and kernel size from [3x3] to [7x7] checking 9 potential combinations.

After the training procedure we run another test dataset composed of 1000 random pair of images to select the set of hyperparameters where the average SSMI index, comparing the output of the trained network and the ground true (our simulated φ), has its maximum value.

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

		KERNEL SIZE		
		3x3	5x5	7x7
Levels	4	0,896	**0,900**	0,801
	5	0,798	0,760	0,728
	6	0,880	0,859	0,760

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 5x5.

3 Model Results

With the selected hyperparameters (kernel size = 5x5 and layers = 4), we build and train a U-NET using a training dataset composed of 15.000 pair of images $[|I - I'|, \varphi]$. The final training was implemented using ADAM optimisation [5] and MEAN_SQUARED_ERROR as the loss function. The training set was divided into training and validation (80% and 20%). The training process took 23 min using a GOOGLE COLLAB instance with GPU support, and it automatically stopped due to the early stop condition (3 epochs without improvement in the loss) after 115 epochs, reaching a loss value of 0.0084 on the validation set. The average SSMI index between the ground true (expected output) and the reconstructed image using the network was 0.899.

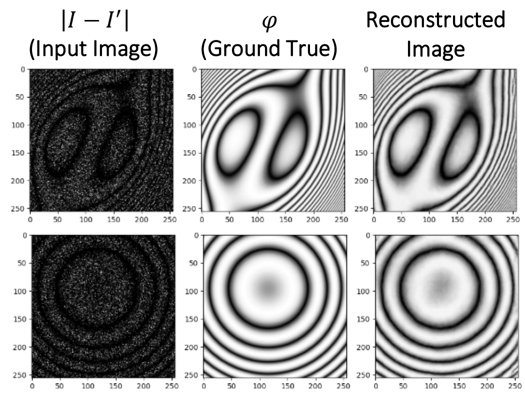

Fig. 1. Results of the Neural Network on a sample of Test Dataset

4 Conclusion

We built a specialised U-NET to process ESPI fringe images, optimising its architecture using an easy method to select the network hyperparameters.

The resulting network improves the perceived quality of the ESPI fringe field and U-NET networks are reliable for their use in speckle interferometry, and the use of synthetic generated training datasets eases the training process removing the need for specialised lab equipment for the design phase.

References

1. Joseph W. Goodman, *Speckle phenomena in optics*, SPIE (2020)
2. Ronneberger, O., Fischer, P., & Brox, T. *U-net: Convolutional networks for biomedical image segmentation*. International Conference on Medical image computing and computer-assisted intervention. 2015.
3. Zuo *et al*. Deep learning in optical metrology: a review. Light: Science & Applications (2022) 11:39
4. Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. *Image quality assessment: from error visibility to structural similarity*. IEEE Transactions on Image Processing. 2004
5. Diederik P. Kigma, Jimmy BA. *Adam: A method for stochastic optimization*. 2014.