A Deep Learning-based Model for Phase Unwrapping

G.E. Spoorthi
Indian Institute of Technology
Tirupati, A.P
Spoorthi.ge92@gmail.com

Subrahmanyam Gorthi
Indian Institute of Technology
Tirupati, A.P
s.gorthi@iittp.ac.in

Rama Krishna Sai Gorthi
Indian Institute of Technology
Tirupati, A.P
rkg@iittp.ac.in

ABSTRACT

Phase unwrapping is an important problem in several applications that attempts to restore original phase from wrapped phase. In this paper, we propose a novel phase unwrapping model based on the deep convolutional neural network by formulating the phase unwrapping as a semantic segmentation problem. The proposed architecture consists of a convolutional encoder network and corresponding decoder network followed by a pixel-wise classification layer. One of the critical challenges in DCNN is availability of large set of labeled training data. This issue is effectively circumvented for the proposed framework through a generic simulation procedure that automatically generates large labeled data. Results from the proposed method are compared with widely used quality-guided phase unwrapping algorithm for various SNR values. It is found that the proposed method is performing well both in terms of accuracy and computational time, even in the presence strong noise. To the best of our knowledge, this is the first work that uses convolutional neural network for phase unwrapping, and this will hopefully pave the way to a new class of techniques for unwrapping the phase.

CCS CONCEPTS

• **Hardware** → *Digital signal processing*; • **Computing methodologies** → *Image segmentation*; Reconstruction; Image processing.

KEYWORDS

Phase Unwrapping, Deep Convolutional Neural Network (DCNN), Encoder, Decoder, Semantic Segmentation

ACM Reference Format:

G.E. Spoorthi, Subrahmanyam Gorthi, and Rama Krishna Sai Gorthi. 2018. A Deep Learning-based Model for Phase Unwrapping. In *11th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2018), December 18–22, 2018, Hyderabad, India,* Anoop M. Namboodiri, Vineeth Balasubramanian, Amit Roy-Chowdhury, and Guido Gerig (Eds.). ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3293353.3293372

1 INTRODUCTION

Recovering the original phase value from the principal value is a classic signal processing problem often known as phase unwrapping. Two-dimensional phase unwrapping problem arises in various applications, such as terrain elevation estimation in Synthetic Aperture Radar (SAR) [8], degree of magnetic field inhomogeneity in the water/fat separation problem of Magnetic Resonance Imaging (MRI) [4] and optical measurement techniques like Fringe Projection Techniques (FTP) [14] and digital holographic interferometry [22].

Phase unwrapping aims to retrieve the original phase from the wrapped phase by removing the artificial 2π discontinuities and making it continuous. Although obtaining the true phase from the wrapped phase appears effortless that could be computed by addition/subtraction of 2π at each pixel depending on the phase difference between the neighboring pixels, however in the presence of strong noise and inconsistencies the problem is ill-posed and recovery of true phase becomes challenging.

Many unwrapping algorithms have been proposed over the years. These algorithms can be broadly classified into two categories: Path-following approaches and Minimum norm approaches. Most path-following algorithms perform phase integration along path chosen to recover true phase. There are four kinds of path-following algorithms: 1) quality-guided algorithm [25], [24]; 2) branch cut algorithm [11]; 3) mask cut algorithm [7]; 4) minimum discontinuity algorithm [23], [6], [18]. Generally these algorithms are computationally efficient but are not robust to severe noise as the error present at a point or local region may propagate along the path. Minimum norm methods [10], [21] minimize the difference between the local derivative of the true phase and that of the wrapped phase to carry out phase unwrapping which produces over smooth phase. Minimum norm methods are robust to noise and can produce acceptable results as the noise or inconsistency in local region will not propagate as in path following methods. However, minimum norm methods are computationally intensive and slow thus making them unsuitable for real time measurements.

Deep learning methods have been extensively studied in object detection and image classification e.g., [15] [2] and results have outperformed previous state-of-the-art. Deep learning techniques have also been applied in various image processing applications such as image super resolution [3], medical image segmentation [20], depth predication in stereo and monocular images [5] etc. In recent years, the applications of deep learning methods have spanned to vast number of fields and applications and they prove to be the best state-of-the-art algorithms in each case. However, to the best of our knowledge, phase unwrapping based on Deep Convolutional Neural Networks (DCNN) has not been explored in the literature.

Motivated by the advances in DCNNs we propose a new method for phase unwrapping using convolutional encoder-decoder architecture by posing it as semantic segmentation problem. Semantic segmentation achieves fine-grained inference by making predication at every pixel also referred to as dense classification or pixel-wise classification in which each pixel is labeled with a class of its enclosing object or region [19], [1].

ICVGIP 2018, December 18–22, 2018, Hyderabad, India

ACM ISBN 978-1-4503-6615-1/18/12.
https://doi.org/10.1145/3293353.3293372

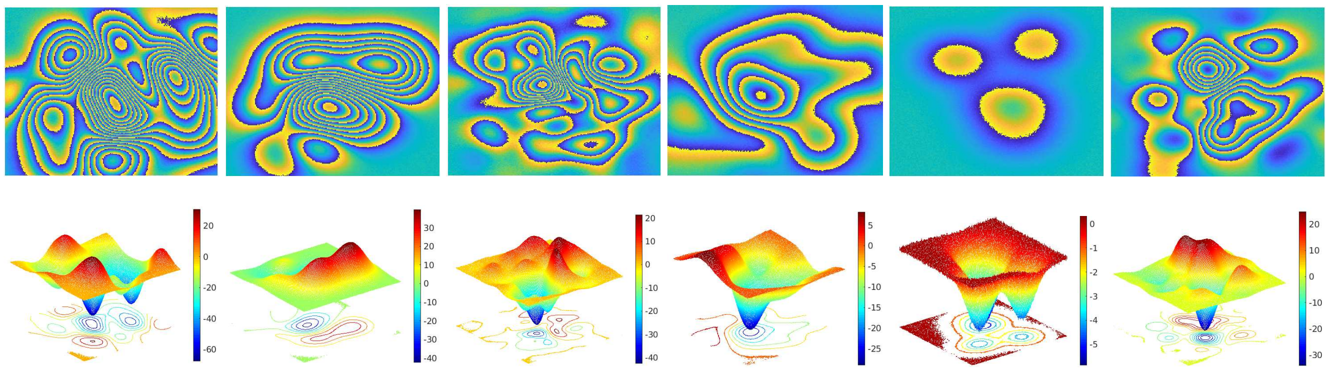


Figure 1: Sample training data that is generated by repeatedly performing random arithmetic operations on the Gaussian function of various mean and variance values.

The proposed model takes wrapped phase as input and outputs the wrap-count (the associated 2π jump with respect to the wrapped phase in unwrapped phase) at each pixel as semantic label. The mathematical relationship between wrapped phase and wrap-count is leveraged and large training dataset of several random shapes is generated in order to contemplate the network on learning generalized phase functions rather than conformity in the training data as shown in Fig. 1. The output of the network is further enhanced by enforcing model smoothness by clustering approach. The proposed method offers high noise susceptibility compared to conventional methods despite of not training DCNN with high noise levels. Furthermore, computational time is less and that makes it adaptable for real time applications.

2 PROPOSED METHOD

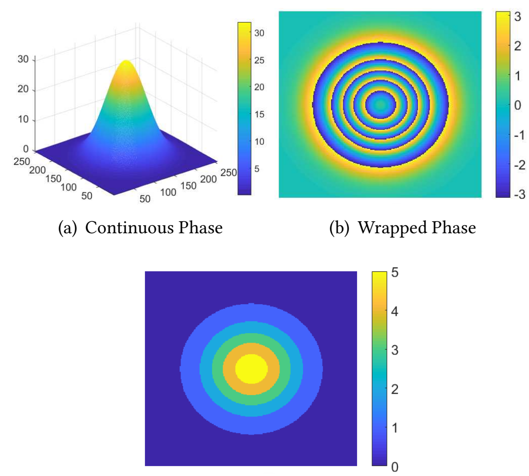
Phase unwrapping can also be described as determining the unknown integral multiple of 2π to be added at each pixel of the wrapped phase map, referred to as wrap-count, to restore underlying true phase. The true phase Φ can be estimated from wrapped phase Ψ as

$$\Phi_{(x,y)} = \Psi_{(x,y)} + 2\pi k_{(x,y)}, \tag{1}$$

where (x,y) denotes the spatial coordinates of a pixel and k denotes an integer. In the proposed framework, our objective is to build a network that takes wrapped phase($\Psi_{(x,y)}$) as input and outputs ($k_{(x,y)}$, wrap-count). An illustration of phase unwrapping is show in the fig. 2. Fig. 2(a) shows the continuous phase map. Fig. 2(b) is the wrapped phase computed from arctan function. Ground truth of the proposed framework is calculated by Eq. 2 which is shown in the Fig. 2(c).

$$k_{(x,y)} = \text{round}\left(\frac{\Phi_{(x,y)} - \Psi_{(x,y)}}{2\pi}\right) \tag{2}$$

In phase unwrapping, understanding relationship between different classes is the foremost requirement contrary to most of the deep learning methods that are trained to learn conformity through intra class similarity and inter class discrimination or learn features of a specific pattern at different translations and scaling. Two regions having same shape, size and position can still belong to different classes depending on the class of neighborhood region or pixel. Fully connected neural network that have demonstrated superior results in object recognition and image classification are known to lose spatial information. On the other hand fully convolutional encoder-decoder neural network [19], [1] recognizes the spatial relationships between different classes and can take arbitrary sized input and produce corresponding sized output with efficient interface and learning.



(a) Continuous Phase (b) Wrapped Phase

(c) Wrap-count

Figure 2: An illustration of wrapping on continuous phase. Wrapped phase will be given as input to proposed convolutional encoder-decoder architecture and wrap-count is given as ground truth.

Therefore, we considered convolutional encoder-decoder architecture that would harmonize with our problem definition and trained the network to detect phase continuity on synthetic data. The network consists of decoder layer corresponding to each encoder layer and the upsampling layers of decoder uses max-pooling indices obtained from corresponding max-pooling layers of encoder similar to SegNet [1]. The primary advantage of this architecture is preciseness of boundaries which would improve the accuracy of the unwrapping and reduce the number of parameters to be learned end to end during the training.

2.1 Generation of Data

One of the important requirements of deep learning techniques is the need for large set of training data. However, in this formulation we leverage the relationship between wrapped phase and wrap-count and circumvent the training data acquisition through principle based relation between the absolute phase and the wrap-count in generating large training dataset. Specifically, training data was synthesized by generating Gaussian functions with various mean and variance and performing random arithmetic operations between generated Gaussians recurrently which would result in irregular shapes. The purpose of generating irregular shapes was to contemplate network identify phase jumps for any shape rather than to learn regular pattern in the input. Furthermore Gaussian noise was added to generated data to make the approach more practical. Fig. 1 gives a glimpse of training data that is used in the proposed method.

2.2 Formulating the Architecture

In order to ascertain that the deep convolutional networks could indeed enforce phase continuities and detect jumps in the wrapped phase map, we initially tested a small, easy-to train architecture and the model was trained to predict wrap-count from 0 to 2, i.e., unwrap the phase maps in the range 0 to 4π. The model consisted of six stacked convolutional layers each of kernel size $3\times3\times64$ with batch normalization and ReLU activation function. The performance of the model was satisfactory and we thus advanced and experimented with different networks to choose the best performing architecture. We tested the aforementioned model to detect higher wrap-count but the accuracy for the higher value of wrap-count was low.

The second test was to train the network by introducing two max-pooling and upsampling layers interleaved between six convolutional encoder and six convolutional decoder layers. For upsampling the indices of max location obtained during max-pooling layer of the corresponding encoder is used to produce sparse feature maps and the preceding convolutional layer produces dense feature maps for the next convolutional layer which would result in improved boundary delineation and the method also reduces the number of parameters to be learned from end to end during training. The model could detect up to -30 to 30 radians and the kernel size was extended to $5\times5\times128$ to detect higher wrap-count. The model was further improved to detect higher wrap-counts by introducing the third max-pooling and upsampling layers constituting the final architecture shown in the Fig. 3. Further adding the max-pooling layer to encoder resulted in adverse results and the accuracy of the model plunged below 20%.

2.3 Proposed Model Architecture

The proposed architecture, after extensive experimentation, is illustrated in the Fig. 3. The network consists of encoder network which has three max-pooling layers interleaved between seven convolutional layers. Corresponding to each encoder layer there is a decoder layer that semantically projects the low resolution features learnt by the encoder onto the pixel space. Upsampling layer in the decoder network upsamples its input feature maps from the indices received from the corresponding max-pooling layer of encoder. Constant kernel size of $5\times5\times128$ is maintained throughout network. Drop out layers are introduced after second and third max-pooling layers and before first and second upsampling layers to avoid overfitting. Convolutional layer of encoder is followed by element wise rectified linear non linearity operation (ReLU) and batch normalization [16]. No nonlinearity is present after decoder layer. Max-Pooling layers have window size of 2×2 and stride of two. Dimension of feature map at the final decoder layer is reduced by convolving with $1\times1\times128\times N$ (N=number of classes) trainable filters and the resultant is fed to softmax classifier that could classify each pixel independently. The network configuration of the proposed method is given table 1.

Table 1: Proposed network configuration.

Layer	#Filters	Size	Output Size
$Conv_1$+ReLU	128	5×5	$256\times256\times128$
$Conv_{2,3}$+ReLU	128	5×5	$256\times256\times128$
Max-pooling$_1$		2×2	$128\times128\times128$
$Conv_{4,5}$+ReLU	128	5×5	$128\times128\times128$
Max-pooling$_2$		2×2	$64\times64\times128$
$Conv_{6,7}$+ReLU	128	5×5	$64\times64\times128$
Max-pooling$_3$		2×2	$32\times32\times128$
Upsampling$_1$		2×2	$64\times64\times128$
$Conv_{8,9}$	128	5×5	$64\times64\times128$
Upsampling$_2$		2×2	$128\times128\times128$
$Conv_{10,11}$	128	5×5	$128\times128\times128$
Upsampling$_3$		2×2	$256\times256\times128$
$Conv_{12,13}$	128	5×5	$256\times256\times128$
$Conv_{14}$	N	1×1	$256\times256\times N$

2.4 Enforcing Model Smoothness

Some of the pixels obtained from convolutional neural network were misclassified and we could recognize that there was monotony in the regions that had misclassified labels. The two foremost regions that were misclassified were as follows:

(1) In the presence of closely disconnected regions, the void that was present was considered as wrapped phase typically as shown in Fig. 4(b) for the wrapped phase in Fig. 4(a), before enforcing the model smoothness.
(2) As the network was trained to detect rapid changes the presence of unblunted region such as tip of the Gaussian was incorrectly classified as can be seen in Fig. 4(b).

For the wrapped phase in Fig. 4(a), the learnt DCNN architecture (discussed in Section 2.3), with the training data as generated in Section 2.1, results in wrap-count in Fig. 4(b) as output. Fig. 4(c) represents ground truth. Hence, there was a necessity of explicitly enforcing model smoothness.

The procedure followed to fine-tune the output is as follows;

(1) Wrapped phase is convoluted with isotropic Laplacian filter [12] that measures the second derivative of the image to obtain residual pixels.

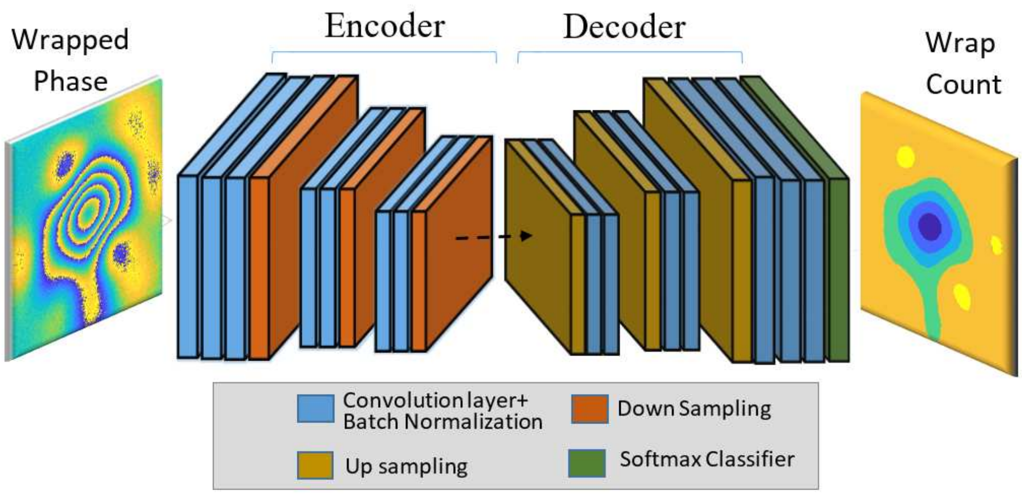


Figure 3: An illustration of the convolutional encoder-decoder architecture. Wrapped phase is given as input and the wrap-count at each pixel is given as ground truth.

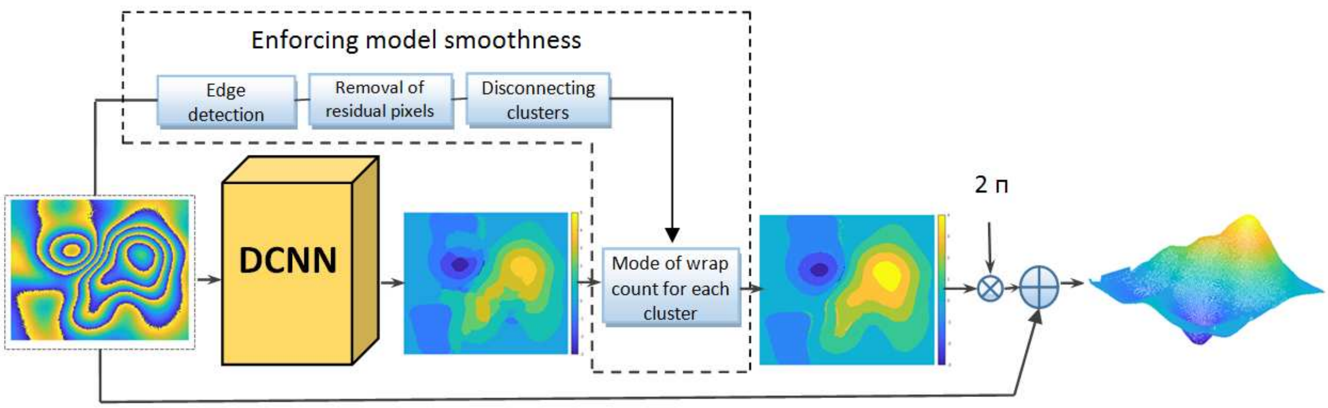


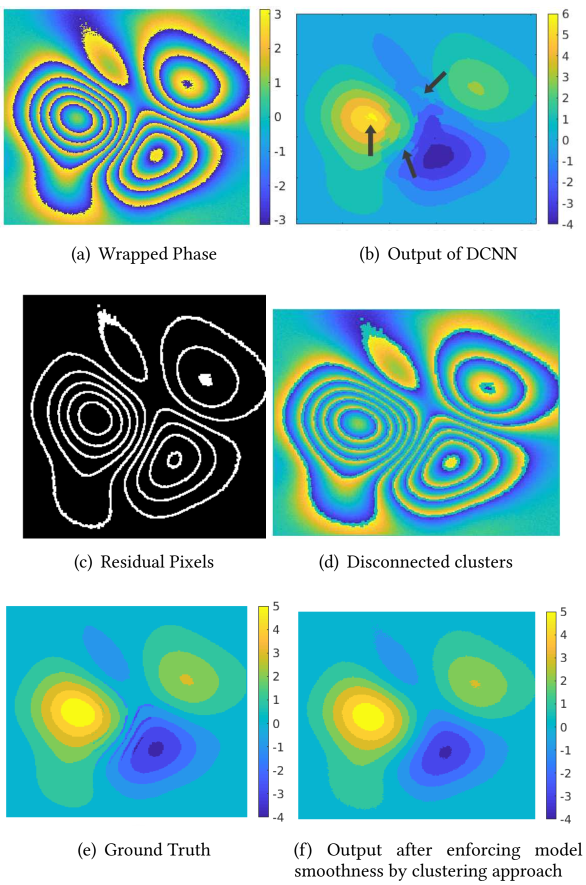
Figure 5: Illustration of the proposed framework. Wrapped phase is given as input to the DCNN. Output of the DCNN with enforced model smoothness gives the wrap-count. It is multiplied with 2π and resultant is added with wrapped phase to retrieve the true phase.

(2) These residual pixels are removed from the wrapped phase by thresolding wrapped phase as shown in Fig. 4(c).
(3) Clusters can now be disconnected by 8-connected neighborhood [12] as shown in Fig. 4(d). Disjoined clusters are binarized and assigned a unique wrap-count by obtaining the probable wrap-count from the output of DCNN at that particular region.
(4) Wrap-count at residual pixels are retained and obtained from output of the DCNN.

Fig. 4(d) shows the wrap-count obtained after enforcing model smoothness. Since the wrap-count at residual pixels location is retained, there is still undesirable classification along the contours of clusters that can be eliminated by passing it through median filter after multiplying wrap-count with 2π and adding that with wrapped phase. Fig. 5 demonstrates the entire framework of the proposed method, integrating all the phases, through a block diagram.

2.5 Training

The data set consists of 10000 training samples and 1000 validation samples each of size $256 \times 256 \times 1$. Normalized wrapped phase is given as input. The weights of all the layers were initialized from scratch by initialization described in [13]. Adam optimizer [17] with moving average decay of 0.99 and initial learning rate of 0.0001 was used. We found that small learning rate was necessary to ensure that the model converges smoothly. Dropout probability was set to 0.25. The network converges after approximately 100K iterations. Training takes about 10 hours on NIVIDA GTX 1080-Ti GPU with 11GB memory.



(a) Wrapped Phase (b) Output of DCNN

(c) Residual Pixels (d) Disconnected clusters

(e) Ground Truth (f) Output after enforcing model smoothness by clustering approach

Figure 4: Illustration of various steps in enforcing model smoothness by clustering approach. Output of DCNN have some incorrectly classified pixels that necessitates enforcing model smoothness explicitly.

2.6 Analysis

The DCNN was trained to detect wrap-count from -15 to 15 (-90 rad to 90 rad) constituting 30 class problem and the proposed framework will detect wrap-count from -6 to 6 (-36 rad to 36 rad) accurately. By increasing the number of channels the framework could be made to detect higher wrap-count. However, in the presence of rapidly varying phase changes the DCNN will lose phase continuity information.

3 EVALUATION ON SIMULATION DATA

To evaluate the robustness of the proposed method for varying noise, *peaks* function of size 256×256 with various noise levels were simulated in MATLAB 2018a. The performance of the method was compared with the well-known quality-guided phase unwrapping (QGPU) [9] method based on the flood-fill algorithm and variance, and MATLAB's unwrap function which corrects the wrapped phase angles based on the concept of adding multiples of $\pm 2\pi$ when absolute jumps between consecutive pixels is greater than or equal to the default jump tolerance of π radians using explicit continuity.

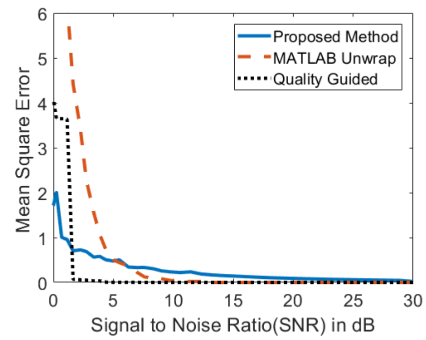


Figure 6: Error analysis of the unwrapped phase estimation performed using the proposed method, QGPU method and MATLAB's unwrap as a function of the SNR.

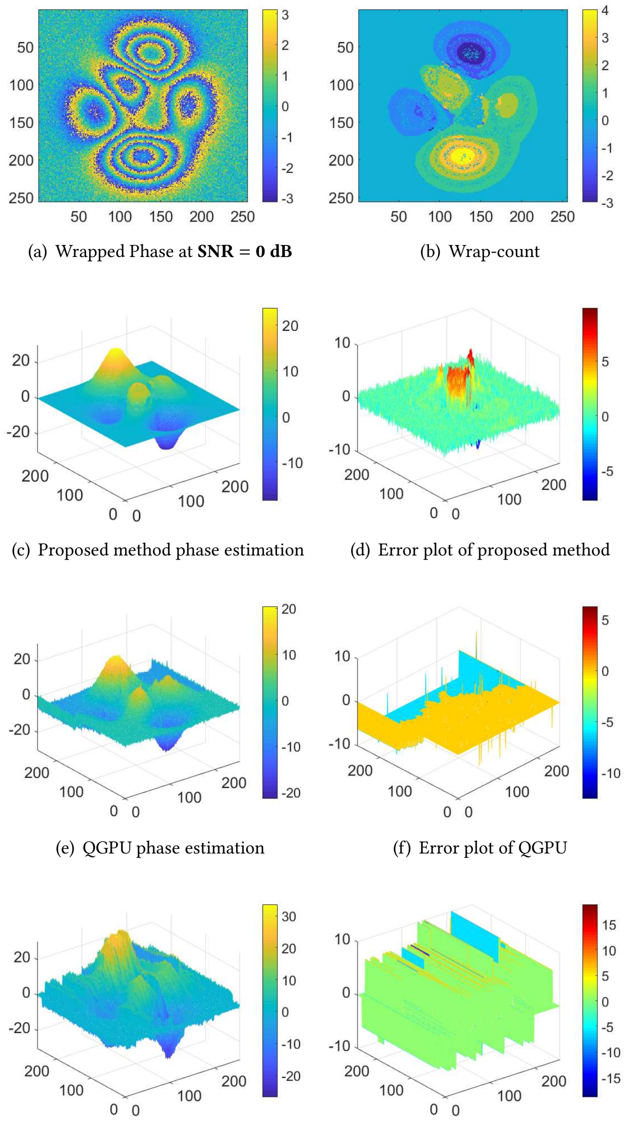
Table 2: Mean Square Error (MSE) & processing time for the PhaseNet, QGPU and MATLAB's unwrap function.

Method	MSE for SNR = 0 dB	Time in Seconds
PhaseNet	2	0.18
Quality-guided Unwrap	11	24
MATLAB's unwrap	17	0.05

Fig. 6 shows error analysis of the unwrapped phase estimation performed using the proposed method, QGPU method and MATLAB's unwrap as a function of the SNR. The error plot for **SNR = 0 dB** is plotted for the proposed method, QGPU and MATLAB's unwrap. Wrapped phase at SNR = 0 dB is shown in the fig. 7(a). Fig. 7(b) shows the wrap-count at each pixel obtained from the proposed method. The phase estimate for the proposed method, QGPU and MATLAB's unwrap shown in the Fig. 7(c)(e)(g). Fig. 7(f) shows the error plot of QGPU. From this error plot it can be seen that half of the reference plane is displaced because of high noise level. As QGPU is a path-following method the false estimation of phase map at one region results in propagation of error (almost cumulatively in spatial dimension) and that results in sudden rise of MSE at low SNRs, as the quality maps will not be accurate at least in few regions, in high noise case and the error propagates across the path. Fig. 7(h) shows the error plot of MATLAB's Unwrap function which has obvious error propagation along the path leading to inferior results. In contrast, the proposed method offers high noise immunity even thought it was not trained for higher noise levels and there is complete absence of error propagation.

The qualitative assessment of the proposed method on various test data is shown in the Fig. 8. It can be seen that the proposed method produces accurate results for different range and shapes. Fig. 9 shows the results of the proposed method performance test at SNR level of 60 dB, 20 dB, 5 dB, 2 dB and -2 dB.

The processing time for obtaining the unwrapped estimation required by the proposed method and quality-guided Unwrap method are provided in table 2. From the table it can be seen that proposed method is extremely fast and efficient especially at high noise levels



(a) Wrapped Phase at **SNR = 0 dB** (b) Wrap-count

(c) Proposed method phase estimation (d) Error plot of proposed method

(e) QGPU phase estimation (f) Error plot of QGPU

(g) MATLAB's unwrap phase estimation (h) Error plot of MATLAB's unwrap

Figure 7: Simulation results of proposed method, quality-guided phase unwrapping algorithm and MATLAB's unwrap function at SNR = 0 dB

and the processing time is less than a second, whereas quality-guided method takes more time making it difficult to apply for real time application.

4 CONCLUSION

This paper presents a novel 2D phase unwrapping method which uses DCNN to unwrap the phase. The main motivation was to design an efficient architecture that could overcome the drawbacks of the existing conventional methods and obtain accurate results with less computational time for practical applications. The proposed model addresses the problem of error propagation in low SNR conditions. We compared the performance of the our proposed method with two of the well know unwrapping techniques and the results demonstrated that the proposed method can produce acceptable results even in the presence of severe noise. The performance of the method can be further enhanced by fine tuning the network with application specific data set. The other issues of phase unwrapping such as rapid phase changes and discontinuities can be further explored.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015).

[2] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*. 647–655.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2016. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38, 2 (2016), 295–307.

[4] Jianwu Dong, Feng Chen, Dong Zhou, Tian Liu, Zhaofei Yu, and Yi Wang. 2017. Phase unwrapping with graph cuts optimization and dual decomposition acceleration for 3D high-resolution MRI data. *Magnetic resonance in medicine* 77, 3 (2017), 1353–1358.

[5] David Eigen and Rob Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*. 2650–2658.

[6] Thomas J Flynn. 1997. Two-dimensional phase unwrapping with minimum weighted discontinuity. *JOSA A* 14, 10 (1997), 2692–2701.

[7] Dapeng Gao and Fuliang Yin. 2012. Mask cut optimization in two-dimensional phase unwrapping. *IEEE Geoscience and Remote Sensing Letters* 9, 3 (2012), 338–342.

[8] Yandong Gao, Shubi Zhang, Tao Li, Qianfu Chen, Shijin Li, and Pengfei Meng. 2018. Adaptive Unscented Kalman Filter Phase Unwrapping Method and Its Application on Gaofen-3 Interferometric SAR Data. *Sensors (Basel, Switzerland)* 18, 6 (2018).

[9] Dennis C Ghiglia and Mark D Pritt. 1998. *Two-dimensional phase unwrapping: theory, algorithms, and software*. Vol. 4. Wiley New York.

[10] Dennis C Ghiglia and Louis A Romero. 1996. Minimum Lp-norm two-dimensional phase unwrapping. *JOSA A* 13, 10 (1996), 1999–2013.

[11] Richard M Goldstein, Howard A Zebker, and Charles L Werner. 1988. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio science* 23, 4 (1988), 713–720.

[12] Rafael C Gonzalez, Richard E Woods, Steven L Eddins, et al. 2004. *Digital image processing using MATLAB*. Vol. 624. Pearson-Prentice-Hall Upper Saddle River, New Jersey.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[14] Zhangqiang He, Junning Cui, Jiubin Tan, Xingyuan Bian, and Wenxue Jiang. 2018. Discrete fringe phase unwrapping algorithm based on Kalman motion estimation for high-speed I/Q-interferometry. *Optics express* 26, 7 (2018), 8699–8708.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks.. In *CVPR*, Vol. 1. 3.

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[17] Diederik Kingma and J Adam Ba. [n. d.]. A method for stochastic optimization. arXiv 2014. *arXiv preprint arXiv:1412.6980* ([n. d.]).

[18] Yihang Liu, Yu Han, Fengjiao Li, and Qican Zhang. 2018. Speedup of minimum discontinuity phase unwrapping algorithm with a reference phase distribution. *Optics Communications* 417 (2018), 97–102.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[21] Jarle Strand, Torfinn Taxt, and Anil K Jain. 1999. Two-dimensional phase unwrapping using a block least-squares method. *IEEE Transactions on Image Processing* 8, 3 (1999), 375–386.

[22] P Ram Sukumar, Rahul G Waghmare, Rakesh Kumar Singh, Gorthi RK Sai Subrahmanyam, and Deepak Mishra. 2015. Phase unwrapping with Kalman filter

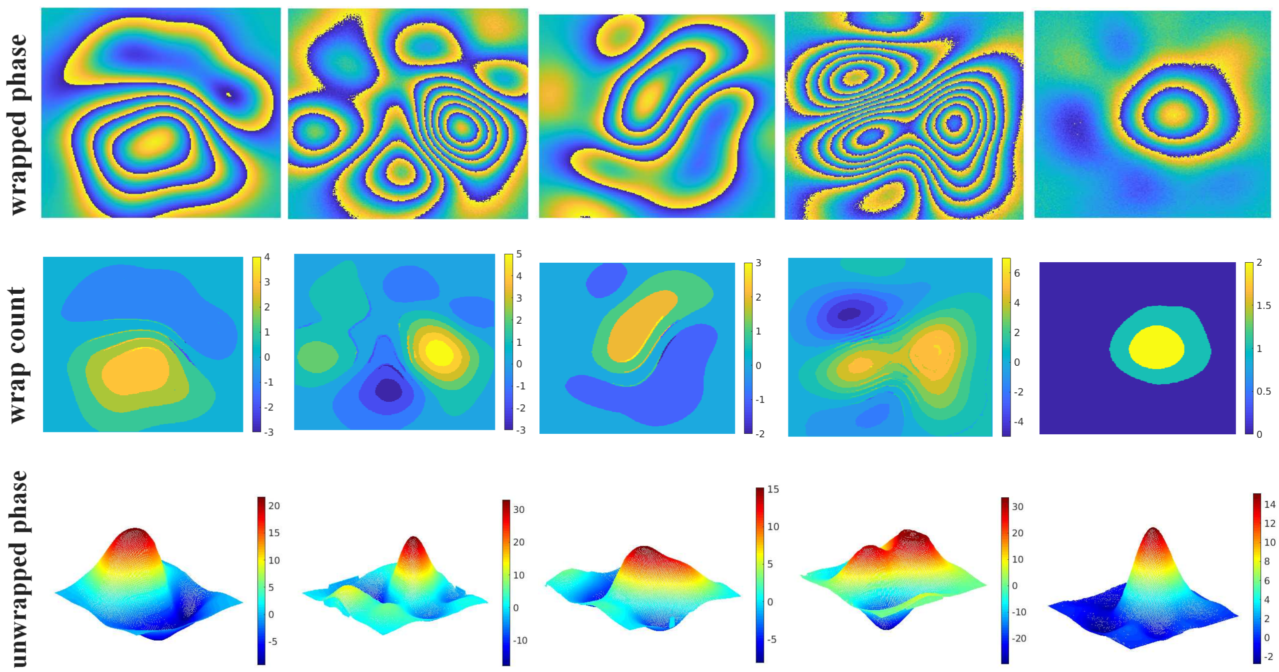
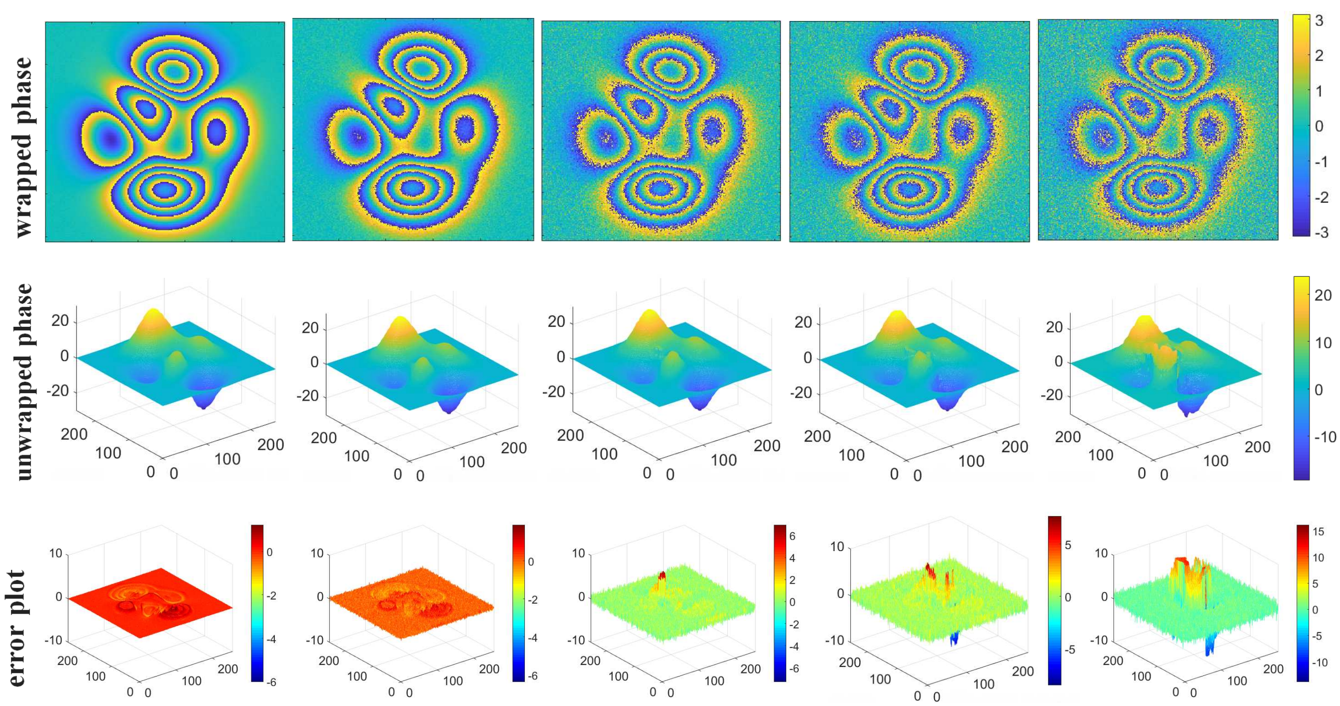


Figure 8: Qualitative assessment of the proposed method results on test dataset. The proposed method is able to predict wrap-count accurately for different patterns.



(a) SNR=60 dB (b) SNR=20 dB (c) SNR=5 dB (d) SNR=2 dB (e) SNR=-2 dB

Figure 9: Results of the proposed methods performance test at various SNR levels. The proposed method gives good results even under low SNR conditions.

based denoising in digital holographic interferometry. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*. IEEE, 2256–2260.

[23] Junyi Xu, Daoxiang An, Xiaotao Huang, and Pan Yi. 2016. An efficient minimum-discontinuity phase-unwrapping method. *IEEE Geoscience and Remote Sensing Letters* 13, 5 (2016), 666–670.

[24] Yudong Zhang, Shuihua Wang, Genlin Ji, and Zhengchao Dong. 2014. An improved quality guided phase unwrapping method and its applications to MRI. *Progress In Electromagnetics Research* 145 (2014), 273–286.

[25] Heping Zhong, Jinsong Tang, Sen Zhang, and Ming Chen. 2011. An improved quality-guided phase-unwrapping algorithm based on priority queue. *IEEE Geoscience and Remote Sensing Letters* 8, 2 (2011), 364–368.