

Deep Convolutional Neural Network-Based Robust Phase Gradient Estimation for Two-Dimensional Phase Unwrapping Using SAR Interferograms

Lifan Zhou[✉], Member, IEEE, Hanwen Yu[✉], Senior Member, IEEE, and Yang Lan[✉]

Abstract—Two-dimensional phase unwrapping (2-D PU) is one of the key processes in reconstructing the topography or displacement of the Earth surface from its interferometric synthetic aperture radar (InSAR) data. Estimating the absolute phase gradient information is an unavoidable step utilized by almost all the 2-D PU methods. Traditionally, the gradient estimation step relies on the phase continuity assumption, which requests that the observed area has spatial continuity. However, the abrupt topographic changes and system noise usually results in the failure of the phase continuity assumption in reality. Under this condition, it is difficult for the traditional 2-D PU to provide the correct absolute phase over the area with abrupt interferometric fringe change or with strong system noise. To solve the issue, we propose a novel deep convolutional neural network (DCNN), abbreviated as PGNet, to estimate the phase gradient information instead of the phase continuity assumption in this article. The major advantage of PGNet lies in its deep architecture to learn the characteristics of phase gradients from enormous training images with different noise levels and topographic features. Subsequently, the L^1 -norm objective function is used to minimize the difference between unwrapped phase gradients and the gradients estimated by PGNet for obtaining the final PU result. Taking the phase gradient pattern of the TerraSAR-X-TanDEM-X interferogram as the learning object, experimental results demonstrate the absolute phase gradient estimated by PGNet is more credible than that from the phase continuity assumption such that the corresponding PU result outperforms those obtained by the traditional 2-D PU methods.

Index Terms—2-D phase unwrapping (2-D PU), deep convolutional neural network (DCNN), interferometric synthetic aperture radar (InSAR), phase gradient estimation.

I. INTRODUCTION

INTERFEROMETRIC synthetic aperture radar (InSAR) is a powerful tool to measure the topography or displacement

Manuscript received August 15, 2019; revised November 27, 2019 and December 21, 2019; accepted December 23, 2019. Date of publication January 28, 2020; date of current version June 24, 2020. This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 61825105, in part by the National Natural Science Foundation of China under Grant 41501461 and Grant 61972059, and in part by the Six Talent Peaks Project in Jiangsu Province under Grant DZXX-027, XYDXX-057. (*Corresponding author: Hanwen Yu*)

Lifan Zhou is with the School of Computer Science and Engineering, Changshu Institute of Technology, Suzhou 215500, China.

Hanwen Yu is with the Department of Civil and Environmental Engineering, University of Houston, Houston, TX 77004 USA, and also with the National Center for Airborne Laser Mapping, University of Houston, Houston, TX 77004 USA (e-mail: yuhanwend@gmail.com).

Yang Lan is with the National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China.

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2020.2965918

of the Earth surface. InSAR system makes use of the phase difference between two or more SAR images covering the same scene to generate interferograms [1], [2]. The obtained wrapped phase is limited to a principle interval $(-\pi, \pi]$ so that the phase discontinuities occur. In order to remove the phase discontinuities and obtain the true phase values, phase unwrapping (PU), which resolves the correct multiples of 2π to each pixel in the wrapped phase image, is needed [3]. However, the 2-D PU problem is an ill-posed inverse problem, i.e., there will be infinite solutions. In order to obtain the unique PU solution, 2-D PU procedure needs the phase continuity assumption (also called the Itoh condition), which requires the absolute phase difference between any two neighboring pixels that cannot be greater than π [4], to estimate the absolute phase difference between neighboring pixels. In essence, the purpose of the phase continuity assumption is to request that the observed areas have spatial continuity. Unfortunately, the phase noise and rugged absolute phase themselves frequently fail to observe the phase continuity assumption in reality. In this case, the 2-D PU becomes an intractable problem.

In recent decades, researchers have proposed many 2-D PU methods. Yu *et al.* [4] provided a good review article of 2-D PU methods using SAR interferograms, which described that there are mainly two kinds of methods (i.e., path-following and optimization-based methods) usually be used to remove the phase discontinuities in the wrapped phase image. The path-following methods alleviate the phase discontinuities by choosing the reasonable integration paths of the estimated phase gradient, including Goldstein's Branch-cut algorithm [5], quality-guided algorithm [6], mask cut algorithm [7], and minimum discontinuity algorithm [8]. The optimization-based methods try to formulate objective functions to minimize the difference between true unwrapped phase gradients and estimated gradients as much as possible, such as the least square method [9], the minimum cost flow (MCF) method [10], SNAPHU method [11], and PUMA method [12]. It can be seen that the PU accuracy of these two categories of methods directly relies on the correctness of the estimated phase gradient information from the phase continuity assumption. In other words, when the phase continuity assumption is destroyed by the high system noise or abrupt terrain change, the incorrect phase gradient information will significantly affect the final PU result. Some researchers have already looked into this issue. Instead of the phase continuity assumption, Qian *et al.* [13] estimated the

phase gradient information by employing the edge detection techniques from the image processing field, but it is very sensitive to noise. In addition, a classification framework, using the Bayes classifier with six wrapped phase features, was proposed to detect the phase gradient information in [14]. Furthermore, Yu and Lan [15] used multiple interferograms with different normal baseline lengths to estimate the absolute phase gradient. Nevertheless, to date, the research achievements on how to estimate the phase gradient information for 2-D PU instead of the phase continuity assumption have been still quite limited.

In fact, in the traditional 2-D PU, the absolute phase gradient, i.e., the ambiguity number difference between neighboring pixels, mainly has three possible solutions (± 1 and 0). Under this condition, the phase gradient estimation problem can be considered as a three-class classification problem in which the phase gradient value between each pair of neighboring pixels in the wrapped phase image is supposed to belong to one of the three classes. That is to say, the phase gradient estimation problem can be potentially solved through a learnable method that takes the wrapped phase as input and gives the output as estimated phase gradients. To date, thanks to the deep architectures and the increased availability of computational resources, deep convolutional neural networks (DCNNs) can recognize the spatial relationships between different classes and directly extract high-level abstract image features from generalized input data, which significantly reduces the disturbance from noise or undesired features [16]. This ability is useful for estimating the phase gradient information when the input interferogram is polluted by strong phase noise or it contains the abrupt interferometric fringe change. Based on the aforementioned discussion, the definition of the research problem in this article is described as follows.

Problem Definition: We wish to use DCNN to learn the phase gradient information from the wrapped phase image such that a novel phase gradient estimation method can be designed instead of the phase continuity assumption.

It can be seen that the believed phase continuity assumption is from the PU algorithm designers' experience. However, recent studies have proven that, in some respects, artificial intelligence (AI) technique is better at accumulating experience than human [17]. Under this condition, we may start to consider to accumulate the correct phase gradient patterns from different study cases by machine learning. There are several learning-based PU methods that have been proposed. For example, back-propagation NN was firstly investigated on PU and the outcome was that it detected the phase wraps in 95% of the occasions in 1-D PU [18]. Also, Hopfield and Kohonen NNs have been experimented with and detected the phase wraps on approximately 50% of the occasions [19]. In addition, Schwartzkopf *et al.* [20] proposed a supervised feed-forward multilayer perceptron NN for 2-D PU based on the computation of probabilities over a local patch and achieved 90% success in identifying phase wraps. Spoorthi *et al.* [21] presented a deep learning-based PU framework that uses the fully convolutional network (FCN), and it is referred to as "PhaseNet," which was proved to be computationally fast. However, it is not effective when the absolute phase

rapidly varies or contains discontinuities, so it requires a clustering-based post-processing afterward.

All the aforementioned learning-based PU methods did not consider to learn the phase gradient information. In this article, we propose a novel DCNN to estimate the phase gradient information of the interferogram and is referred to as PGNet. The major advantage of the PGNet lies in its deep architecture to learn the characteristics of phase gradients from enormous training images with different noise levels and topographic features such that PGNet will recognize the patterns of the correct phase gradients rather than relying on the phase continuity assumption. Contrarily, the phase gradient information from the phase continuity assumption is mainly based on the PU algorithm designers' experience, which is not applicable in many situations, such as valleys, steep mountains, or noisy wrapped phases. To be specific, we design and train the PGNet on enormous patches of the simulated phase images with different noise levels and topographic features, and then utilize the trained PGNet to predict the phase gradients of the input wrapped phase image. Subsequently, the final PU result is computed using the L^1 -norm objective function to minimize the difference between the unwrapped phase gradients and the estimated gradients obtained by PGNet. The whole two-stage programming procedure is jointly abbreviated as the PGNet-PU method. Because the phase gradient information obtained by the PGNet is more correct, the accuracy of the PGNet-PU method is significantly higher than the traditional 2-D PU methods. It is worthwhile mentioning that the two-stage programming procedure applied in PGNet-PU is consistent with the framework of the traditional model-based 2-D PU method so that PGNet-PU can straightforwardly cooperate with many assistant PU components, such as applying the InSAR quality map as the PU weight.

This article is organized as follows. In Section II, the principles of the 2-D PU methods based on the phase continuity assumption are reviewed and the basic idea of how to detect the phase gradient information using the learning framework instead of the phase continuity assumption is introduced. Then, the design of PGNet is presented in Section III. The implementation details of the PGNet-PU method are described in Section IV. In Section V, the performance of the PGNet-PU method is tested both on simulated and real InSAR data sets. The conclusions are presented in Section VI.

II. REVIEW AND ANALYSIS

In this section, we will review the basic principle of 2-D PU, especially how the traditional PU methods estimate the phase gradient information based on the phase continuity assumption. Then, we will introduce an idea of how to estimate the phase gradient information using the learning framework to replace the phase continuity assumption.

A. Basic Principle of 2-D PU

2-D PU can be regarded as estimating the unknown integral multiple of 2π to be added at each pixel in the wrapped phase image to restore the true phase, given by

$$\varphi(s) = \psi(s) - 2k(s)\pi \quad (1)$$

where $\phi(s)$ is the wrapped phase of the s th pixel, $\psi(s)$ is the unknown true phase of the s th pixel, and $k(s)$ is the unknown ambiguity number of the s th pixel, which is an integer. From (1), we can see that directly solving (1) is an ill-posed inverse problem because there are two unknowns in one equation, i.e., there will be infinite solutions to (1). Hence, almost all the 2-D PU methods make use of the phase continuity assumption to find the unique PU solution [4]. If the phase continuity assumption holds well, no ambiguity exists in the true phase difference between neighboring pixels, i.e., the phase gradient information should be correct. Under this condition, the PU result can be easily and uniquely achieved by a simple integration process [15].

The phase gradient information, i.e., the neighboring ambiguity number difference, can be obtained by

$$\hat{\Delta}_k(s, s-1) = \begin{cases} 0, & |\phi(s) - \phi(s-1)| \leq \pi \\ -1, & \phi(s) - \phi(s-1) > \pi \\ 1, & \phi(s) - \phi(s-1) < -\pi \end{cases} \quad (2)$$

where the indexes s and $s-1$ denote two neighboring pixels, and $\hat{\Delta}_k(s, s-1)$ is the estimated phase gradient information between the two neighboring pixels. It can be seen that there are two directions (vertical and horizontal) of phase gradients for the 2-D PU problem, i.e., the phase gradient information obtained by (2) is 2-D field and the value of the neighboring ambiguity number differences are restricted to ± 1 and 0. Unfortunately, high system noise and abrupt topographic changes result in the failure of the phase continuity assumption [4]. In other words, it is difficult to guarantee that there is a PU solution whose phase gradient between each pair of two neighboring pixels is exactly equal to the estimation obtained by (2). In this case, essentially, almost all the 2-D PU methods can be considered as a kind of criterion which minimizes the difference between the unwrapped phase gradient and that estimated by (2) to achieve the PU solution [15]. The generalized mathematical model of the 2-D PU method is shown as

$$\begin{aligned} \arg \min_{k(s)} \quad & \sum_{(s,s-1)} f(t(s, s-1)) \\ \text{s.t. } & k(s) - k(s-1) - \hat{\Delta}_k(s, s-1) = t(s, s-1) \\ & k(s) \in \text{integer} \end{aligned} \quad (3)$$

where $k(s)$ is the decision variable, $t(s, s-1)$ is the auxiliary variable, and $f(\cdot)$ is the generalized objective function. For example, if $f(\cdot)$ is the L^2 -norm, (3) will become the least square PU method [9], or if L^1 -norm, (3) will become the MCF method [10]. Finally, we can obtain the absolute phase $\psi(s)$ from the solution of (3), which is equal to $\phi(s) + 2k(s)\pi$. However, we can see that the credibility of the PU result from (3) is directly based on the correctness of $\hat{\Delta}_k(s, s-1)$. In other words, if the accuracy of $\hat{\Delta}_k(s, s-1)$ is too low, no matter what kind of $f(\cdot)$ is chosen, it could be impossible for the 2-D PU method to obtain the full correct PU solution.

B. Problem Analysis

According to (2), there are three classes of the absolute phase gradient, i.e., ± 1 and 0, in the 2-D PU. To improve the

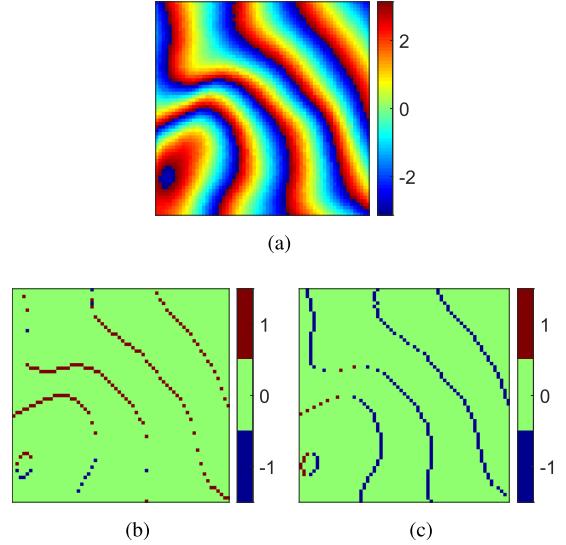


Fig. 1. Example between the wrapped phase image and the corresponding phase gradient images which have three classes. (a) Wrapped phase image. (b) Vertical phase gradient image of (a). (c) Horizontal phase gradient image of (a).

estimation accuracy of $\hat{\Delta}_k(s, s-1)$, in this article, we transform the phase gradient estimation problem as a three-class classification learnable problem, in which the phase gradient value between each pair of neighboring pixels in the wrapped phase image is supposed to belong to one of the three classes. An example between the wrapped phase image and the corresponding phase gradient images is illustrated in Fig. 1. Fig. 1(a) is the input interferogram. Fig. 1(b) and (c) are the vertical and horizontal phase gradient images of Fig. 1(a). In Fig. 1(b) and (c), the red pixels belong to the class whose neighboring ambiguity number difference is 1. Green belongs to the class whose neighboring ambiguity number difference is 0. Blue belongs to the class whose neighboring ambiguity number difference is -1 .

In this case, the corresponding three-class classification problem can be solved through a learning framework that takes the wrapped phase $\phi(s)$ as input and gives the output as the estimated phase gradient $\hat{\Delta}_k(s, s-1)$. Because the wrapped phase $\phi(s)$ can be simulated, the true phase $\psi(s)$ is available for training the learning framework. Based on the true phase $\psi(s)$, the true phase gradients $\Delta_k(s, s-1)$, as the ground truth, can be computed using the following equation:

$$\Delta_k(s, s-1) = k(s) - k(s-1) \quad (4)$$

where $k(s) = \text{round}((\psi(s) - \phi(s))/2\pi)$. It is worth mentioning that we assume $\Delta_k(s, s-1)$ in (4) belongs to ± 1 or 0. In other words, if the true phase gradient is less than -1 , we will set it to -1 . If larger than 1, we set it to 1 (it is possible that the actual phase gradient is bigger than 1, or smaller than -1). The reason why we restrict $\Delta_k(s, s-1) \in \{0, \pm 1\}$ is that this phase gradient range can cover most of the common situations. The case with a larger phase gradient range will be discussed later. Fig. 2 illustrates the six noisy wrapped phase images with different mean coherence from 0.4 to 0.9, and their corresponding true vertical and horizontal phase

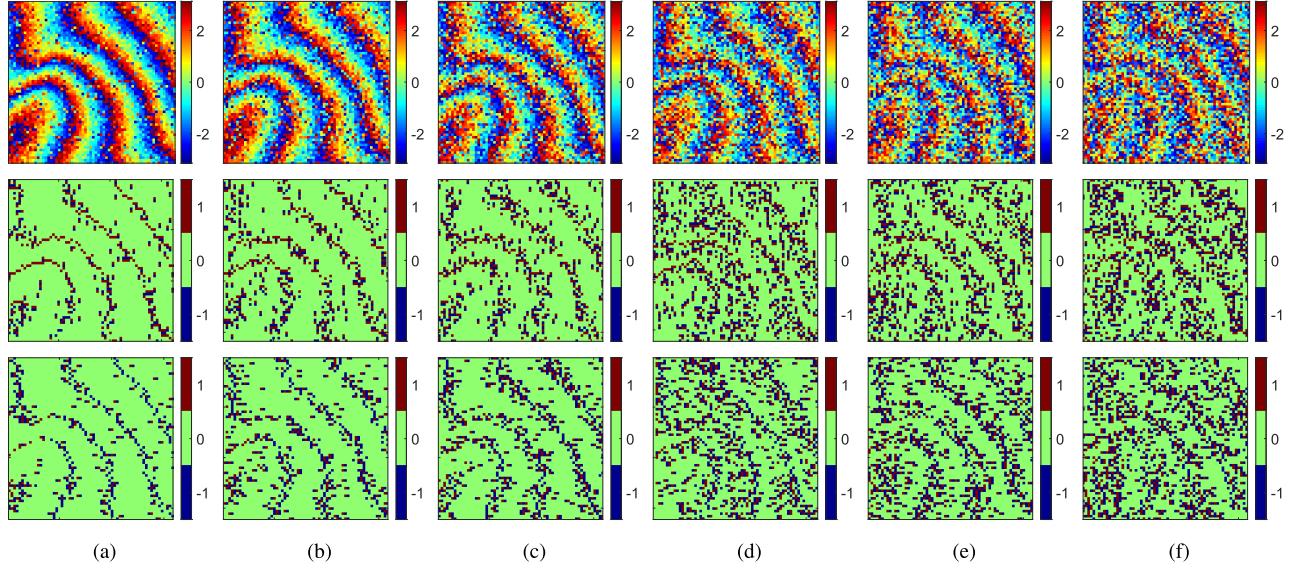


Fig. 2. Row 1: six noisy wrapped phase images with different mean coherences (a) 0.9, (b) 0.8, (c) 0.7, (d) 0.6, (e) 0.5, and (f) 0.4. Row 2: corresponding true vertical phase gradient images. Row 3: corresponding true horizontal phase gradient images.

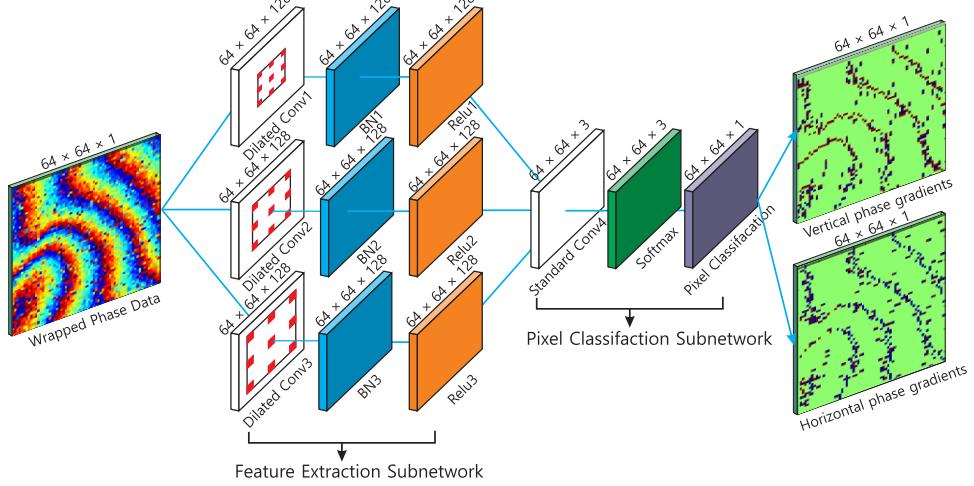


Fig. 3. Illustration of the PGNet architecture. Wrapped phase image is given as input, and vertical and horizontal phase gradients at each pixel are outputs.

gradient images obtained by (4). From Fig. 2, we observe that, while the coherence coefficient is getting lower, and the vertical and horizontal phase gradient patterns will be more destroyed. By owning the diversity of phase gradient patterns, the robustness of the learning framework will be significantly increased.

Recent studies indicate that DCNN, as a powerful learning framework, has been extensively used in semantic segmentation, and has been achieved state-of-the-art performance [22], [23]. Furthermore, DCNN has also gained increasing interest in semantic segmentation of remote sensing images [24]–[27]. Its powerful ability lies in the fact that it directly extracts high-level abstract image features from generalized input data and significantly reduces the disturbance from noise or undesired features. This ability is useful for estimating the phase gradient information when the input interferogram is polluted by strong phase noise or it contains the abrupt interferometric fringe change. However, to date,

to our knowledge, there is no study that uses DCNN to estimate the phase gradients of the wrapped phase image. In Section III, we will introduce the proposed PGNet for the phase gradient estimation.

III. DESIGN OF PGNET

The detailed network architecture of the PGNet is illustrated in Fig. 3, and the details of the network configuration are shown in Table I. From Fig. 3, we can see that the proposed network contains two subnetworks: the feature extraction subnetwork and the pixel classification subnetwork. In the feature extraction subnetwork, some feature maps are extracted to describe the characteristics of phase gradient information in the wrapped phase image. In the pixel classification subnetwork, the three feature maps generated by the first subnetwork are utilized to classifying the gradient value between each pair of neighboring pixels in wrapped phase images into one of the three classes obtained by (4). It is worth mentioning that

TABLE I
MAJOR PARAMETERS OF PGNET

Layer Name	#Filters	Filter Size	Output Size
Wrapped phase input	-	-	$64 \times 64 \times 1$
Dilated convolution1	128	$3 \times 3 \times 1$	$64 \times 64 \times 128$
BN1	-	-	$64 \times 64 \times 128$
Relu1	-	-	$64 \times 64 \times 128$
Dilated convolution2	128	$3 \times 3 \times 1$	$64 \times 64 \times 128$
BN2	-	-	$64 \times 64 \times 128$
Relu2	-	-	$64 \times 64 \times 128$
Dilated convolution3	128	$3 \times 3 \times 1$	$64 \times 64 \times 128$
BN3	-	-	$64 \times 64 \times 128$
Relu3	-	-	$64 \times 64 \times 128$
Standard convolution4	3	$1 \times 1 \times 128$	$64 \times 64 \times 3$
Softmax	-	-	$64 \times 64 \times 3$
Pixel classification	-	-	$64 \times 64 \times 1$

there are two PGNet networks (shown in Fig. 3) are trained by the vertical and horizontal phase gradient information, respectively. In other words, the proposed PGNet is applied to predict the vertical and horizontal gradients independently. In the following, each layer in PGNet will be introduced in detail.

A. Introduction of Network Architecture

1) *Dilated Convolution Layer*: The dilated convolution is used to apply sliding convolutional filters to the wrapped phase images, which convolves the input by moving the filters along the input vertically and horizontally and computing the dot product of the weights and the input and then adding a bias term. The main idea of dilated convolution is to insert “holes” (zeros) between pixels in convolutional kernels to systematically aggregate multiscale contextual information without losing resolution [28]. However, the traditional convolution layer integrates multiscale contextual information via successive pooling and subsampling layers which will reduce the resolution. Therefore, it is difficult for the traditional convolution layer to deal with the conflicting demands of multiscale reasoning and full-resolution prediction. From Fig. 1(b) and (c), it can be seen that the pattern of the phase gradient on an interferogram looks like the continuous curve. If any pixel on the gradient curve is missed or its location is a few pixels away, it will generate the residues from the coming integration step. Residue is a classic concept in 2-D PU [5], which is computed by the loop integration of the gradients on each 2×2 neighboring pixels and indicates the information of the discontinuity phase gradient. Usually, the less number of residues usually provide us a more credible PU result [4]. For the output d th feature map of the s th pixel $y(s, d)$, we formulate dilated convolution over the input wrapped phase image $\varphi(s)$ as follows:

$$y(s, d) = \sum_{e=1}^E \varphi(s + r \cdot e) \cdot \theta(d, e) + b(s) \quad (5)$$

where r is the dilated rate, E denotes the filter size, $\theta(d, e)$ is the e th parameter of the d th filter, and $b(s)$ is the bias term at the s th pixel. It is worth mentioning that the number of the output feature maps obtained by (5) equals the number of filters D used in the dilated convolutions. The dilated rate r

corresponds to the stride with which we sample from the input wrapped phase image. We can adaptively modify the receptive field of the filters by changing the dilated rate value. To capture multiscaled feature maps of phase gradient information in the wrapped phase image, we design three dilated convolutions with 1 stride and r padding, which can enlarge the receptive field by increasing the dilated rate. Several major parameters of the three dilated convolutions, e.g., number of filters D , filter size E , and dilation factor r , need to be chosen, which requires empirical analysis and hyperparameter tuning. To determine the parameter values of PGNet, the validation data are used to compare the performance of our network that is created based on the training data using different values of parameters. The experimental results indicate that $D = 128$, $E = 3$, and three increasing dilation factors ($r_1 = 2$, $r_2 = 4$, $r_3 = 6$) seems to give the highest validation accuracy. It is worth mentioning that the dilated convolution is equivalent to standard convolution when $r = 1$. Usually, the standard convolution layers aggregate multiscale contextual information by increasing the filter size E . Under this condition, the number of parameters associated with each layer grows drastically, which results in increasing the amount of computation. In addition, the standard convolution operator is usually followed by pooling and subsampling, which will affect the resolution of the feature maps. Compared to the standard convolution operator, there are two key advantages of the dilated convolutions applied in our network. To begin with, the dilated convolution operator can apply the same filter at different ranges using different dilation factors, which supports the exponential expansion of the receptive field without increasing the amount of computation. In addition, the dilated convolution operator does not need pooling or subsampling operation, which will keep the resolution and the coverage of the feature maps. Of late, many semantic segmentation networks, such as DeepLab [29], make use of the advantages of the dilated convolution to reliably increase the learning accuracy.

2) *BN and ReLU Layers*: Each dilated convolution is followed by element-wise batch normalization (BN) and rectified linear nonlinearity operation (ReLU) layers, respectively. The BN layer normalizes each input feature map across a mini-batch, which speeds up training of the network and reduces the sensitivity to network initialization [30]. Then, it calculates the normalized activations as

$$y_{\text{BN}}(s, d) = \frac{y(s, d) - \mu}{\sqrt{\sigma^2 + \varepsilon}} \quad (6)$$

where $y(s, d)$ is the input d th feature map of the s th pixel, μ and σ^2 are the mean and variance over a mini-batch of each input feature map, and ε is the property Epsilon, which improves numerical stability when the mini-batch variance is very small. The ReLU layer performs a threshold operation to each element of the input BN layer, where any value less than zero is set to zero [31], that is,

$$y_{\text{ReLU}}(s, d) = \begin{cases} y_{\text{BN}}(s, d), & y_{\text{BN}}(s, d) \geq 0 \\ 0, & y_{\text{BN}}(s, d) < 0. \end{cases} \quad (7)$$

3) Standard Convolution, Softmax, and Pixel Classification

Layers: The pixel classification subnetwork, working on these three feature maps, is employed to classify each pixel into one of the three classes according to (4). Under this condition, the subnetwork has three classes, i.e., the neighboring ambiguity number differences to be ± 1 or 0. This subnetwork includes a standard convolution layer, with three filters with size 1×1 , a stride of 1, and a padding of 1, is created to combine the three feature maps with size $64 \times 64 \times 128$ down to the three classes maps with size $64 \times 64 \times 3$.

Whereafter, this 1×1 standard convolution layer is followed by the softmax and pixel classification layers. The softmax layer applies a softmax function to make the input classes maps lie between zero and one and to sum to one [32], given by

$$y_{\text{soft}}(s, j) = \frac{\exp(y_{\text{class}}(s, j))}{\sum_{j=1}^K \exp(y_{\text{class}}(s, j))} \quad (8)$$

where $y_{\text{class}}(s, j)$ is the input j th classes map of the s th pixel, and K is the number of classes, i.e., $K = 3$ in PGNet. The pixel classification layer takes the values from the softmax function and assigns each input to one of the three classes using the cross entropy function [32], that is,

$$\begin{aligned} \text{loss} = - \sum_{s=1}^N & \left(\sum_{j=1}^K h(s, j) \cdot \ln(y_{\text{soft}}(s, j)) \right. \\ & \left. + \lambda \cdot \sum_{d=1}^D \sum_{e=1}^E \theta(d, e)^2 \right) \end{aligned} \quad (9)$$

where N is the number of pixels, $h(s, j)$ is the indicator that the s th pixel belongs to the j th class from the ground truth map. In addition, we add a L^2 -norm regularization term for the weights to the loss function to reduce overfitting, where λ is the regularization coefficient. In short, the softmax and pixel classification layers are combined to predict the phase gradient information for each pair of neighboring pixels in the input wrapped phase image.

B. Generation of Training Data

One of the compelling requisites for training our network is to have a large labeled data set. In our network, for detecting vertical and horizontal phase gradient information in the wrapped phase image, there is a definitive input–output relationship between the wrapped phase images and the estimated phase gradient images, which is shown in Fig. 3. As described above, the true phase gradient images can be derived from the true phase images according to (4), which can be simulated by the existing digital elevation model (DEM) database, such as Shuttle Radar Topography Mission (SRTM) DEM, according to the simulated InSAR system which has the identical parameters of the real InSAR system. Under this condition, the phase noise can be added to generate wrapped phase data sets with different coherence coefficients through employing a probability density function of the noise wrapped phase in [33]. This enables the network to learn the correct phase gradient information with any kind of noise level and

topographic feature rather than limiting it to a certain situation. In Section V, we will specifically take the phase gradient pattern of the TerraSAR-X-TanDEM-X interferogram as the learning object to validate the effectiveness of this strategy.

C. Implementation of Training

We choose the stochastic gradient descent with momentum (SGDM) as an optimizer with an initial learning rate of 10^{-3} and a momentum of 0.99. The learning rate uses a piecewise schedule, which is reduced by a factor of 0.3 every 10 epochs. This allows the network to learn efficiently with a higher initial learning rate while being able to find a solution close to the local optimum once the learning rate drops. In addition, our network takes image patches as inputs with size 64×64 and a mini-batch with 64 patches at each iteration is used. It is worth mentioning that the size of the input patch used in our network is not only limited to 64×64 , which can be assigned to other size as well, e.g., 128×128 or 256×256 , but the smaller input patch size usually reduces more memory usage of training. Furthermore, our network is trained by the simulated wrapped phase patches and the corresponding true phase gradient patches, in which 70% of the data are selected in random and used to train the network, and the rest of them are used for validation. There are two benefits of the validation patches in our network. To begin with, our network is tested against the validation patches every epoch which prevents the network from overfitting on the training patches. In addition, the validation data are also employed to determine the optimal parameters of the dilated convolutions used in our network. Furthermore, being different from the training and validation phase images, extra wrapped phase images are used as the test data to validate the performance of our network. Finally, our network is trained on an NVIDIA™ Titan X with 12 GB of GPU memory. Although the training is very time-consuming, the network can be fully parallelized during offline training.

IV. PGNET-PU METHOD

In the PGNet-PU method, the phase gradient information is predicted by PGNet first. Then, the L^1 -norm objective function is used to minimize the difference between the unwrapped phase gradients and the estimated gradients from PGNet for obtaining the final PU result. Hence, there will be two processing stages in the PGNet-PU method.

A. Stage 1: Phase Gradient Information Prediction

When PGNet has been trained, it can be employed to predict the phase gradient information of the wrapped phase image. Because the trained PGNet could learn the phase gradient characteristics with any kind of noise level and topographic feature from enormous training images, it can recognize the credible phase gradient information from the complicated and noisy interferogram. In addition, the running time of the gradient estimation step using the trained PGNet is very rapid because the computation of the forward propagation in our network is straightforward. Regarding memory usage, the peak memory consumption of this step just requires $m \times n$ bytes, where m is the number of pixels in the wrapped phase image, and n is the number of filters used in the network.

B. Stage 2: Unwrapping Using Phase Gradients From PGNet

Stage 2 of the PGNet-PU method is the unwrapping step using the phase gradients from PGNet because we cannot guarantee that the vertical and horizontal phase gradient information obtained by PGNet is an irrotational field. That is to say, we cannot directly use a simple flood-fill integration process to unwrap the interferogram [4]. In this case, a 2-D PU model is still required. In fact, almost all the existing 2-D PU model can be used in this step. In the following discussion, the L^1 -norm PU model is selected to obtain the absolute phase of the interferogram as shown in the following:

$$\begin{aligned} \arg \min_{k(s)} \sum_{(s,s-1)} w(s, s-1) \cdot |t(s, s-1)| \\ \text{s.t. } k(s) - k(s-1) - \tilde{\Delta}_k(s, s-1) = t(s, s-1) \\ k(s) \in \text{integer} \end{aligned} \quad (10)$$

where $w(s, s-1)$ is the weighted coefficient, which can be derived from any kind of quality map in InSAR [34], and $\tilde{\Delta}_k(s, s-1)$ is the phase gradient estimated by PGNet. Because the physical meaning of (10) is the MCF model, there are several strong polynomial algorithms that can be applied to solve (10) (e.g., minimum mean cycle-canceling algorithm and network simplex algorithm [35]). Furthermore, some studies [36]–[38] indicate that we can use the divide-and-conquer criterion to further reduce the computational and peak memory consumption of (10). Therefore, it can be seen that the total time and space complexities of PGNet-PU is practical.

V. PERFORMANCE ANALYSIS

In this section, we conduct the performance evaluation and illustration of the proposed PGNet-PU method in comparison with several representative 2-D PU methods through three experiments. The first experiment demonstrates the robustness of PGNet in estimating the phase gradients on the simulated InSAR data set compared with the traditional phase continuity assumption. The second experiment is also performed on the simulated InSAR data set, which validates the effectiveness of the PGNet-PU method using the phase gradients from PGNet for obtaining the final PU result. The third experiment is performed on the real InSAR data set for comparing PGNet-PU with the traditional representative 2-D PU methods. We shall start with a brief introduction to the training setting.

A. Introduction of Training Setting

The training data used in this experiment is from the SRTM DEM, which covers the mountainous area around Lhasa, Tibet autonomous region, China, which is shown in Fig. 4(a). The unit of all the DEMs in this section is meters. The true unwrapped and wrapped phase images with size 6336×4032 are simulated by Fig. 4(a) according to the parameters of the simulated system listed in Table II. The reasons why we choose these system parameters and Fig. 4(a) to generate the training sample pool are shown as follows. To begin with, the topographic features of Fig. 4(a) is similar to those of the interferograms applied for testing the PU performance in the following experiments. In addition, the system parameters

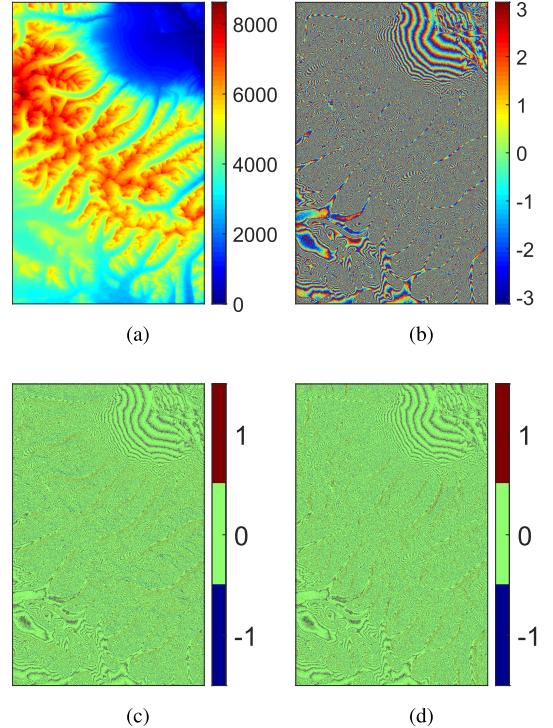


Fig. 4. (a) Reference DEM. (b) Simulated interferogram with 0.8 coherence coefficient. (c) True vertical phase gradient image of (b). (d) True horizontal phase gradient image of (b).

TABLE II
MAJOR INTERFEROMETRIC PARAMETERS OF SIMULATED InSAR SYSTEM

Orbit Altitude	Incidence Angle	Wavelength	Normal Baseline
6885km	46°	0.031m	268.324m

listed in Table II are the same as those of the real interferogram employed in the following experiments as well. The 12 training wrapped phase images with the coherence coefficients ranging from 0.4 to 0.95 with 0.05 as the interval are simulated by employing the probability density function of the noisy wrapped phase given in [33]. It is worth mentioning that, in our simulation, we use the general coherence coefficient to jointly express all the decorrelation components. Using noisy wrapped phase images with different noise levels can make the trained network more robust for estimating the phase gradient information of the wrapped phase images under different noise situations. Fig. 4(b) illustrates the wrapped phase image with 0.8 coherence coefficient, which is one of the training wrapped phase images. Fig. 4(c) and (d) illustrates the corresponding ground truth maps [i.e., the vertical and horizontal absolute phase gradient images obtained by (4)] of Fig. 4(b). For reducing the peak memory usage while training, we cut the whole simulated interferograms into patches (the size is 64×64) as inputs. The total number of the groups of the training patches (i.e., simulated wrapped phase patch and its corresponding true vertical and horizontal phase gradient patches) are 74844. 52391 training patch groups are used to train our network, and the rest of them are used as the validation data set to prevent the network from overfitting on

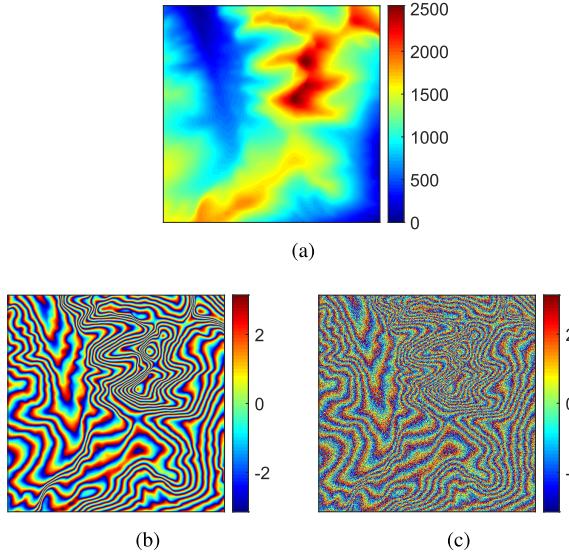


Fig. 5. (a) Reference DEM. (b) Simulated noise-free wrapped phase image. (c) Simulated wrapped phase image with 0.5 coherence coefficient.

the training patches and determine the optimal parameters of the network.

B. Evaluation of Robustness of PGNet Phase Gradient Estimation

In the first experiment, we test the robustness of PGNet in detecting the phase gradients on the simulated InSAR data. Fig. 5(a) shows the reference terrain height which is also from the SRTM DEM at Lhasa, Tibet autonomous region, China. However, Fig. 5(a) is different from the data used in the training procedure. By using the parameters listed in Table II, we can obtain the simulated noise-free wrapped phase image of Fig. 5(a) (the size is 512×512), as shown in Fig. 5(b). The 12 wrapped phase images for testing are then generated with different coherence coefficients between 0.4 and 0.95 with 0.05 as the interval. Fig. 5(c) illustrates one tested wrapped phase image with a 0.5 coherence coefficient. From Fig. 5(c), it can be found that, due to the low coherence, the phase fringes are very blurry and the phase gradient pattern is very difficult to recognize. In this experiment, the phase continuity assumption utilized in (2) is used as a baseline method to compare with the proposed PGNet.

The visual results obtained by the phase continuity assumption and the PGNet of Fig. 5(c) are illustrated in Fig. 6. Fig. 6(a) and (b) reveals the vertical and horizontal phase gradient results using the traditional phase continuity assumption. Fig. 6(c) and (d) shows the vertical and horizontal phase gradient results using PGNet. According to phase gradients, we can compute the residue distribution. Fig. 7(a) is the residue distribution obtained by Fig. 6(a) and (b) (47428 residues). Fig. 7(b) is the residue distribution obtained by Fig. 6(c) and (d) (17690 residues). In Fig. 7(a) and (b), the value of each pixel represents the corresponding residue polarity (the polarity is discretely limited to $\{0, \pm 1\}$ [4]). Fig. 7(c) and (d) shows the magnified images of the red squares in Fig. 7(a) and (b). Compared Fig. 7(c) with Fig. 7(d),

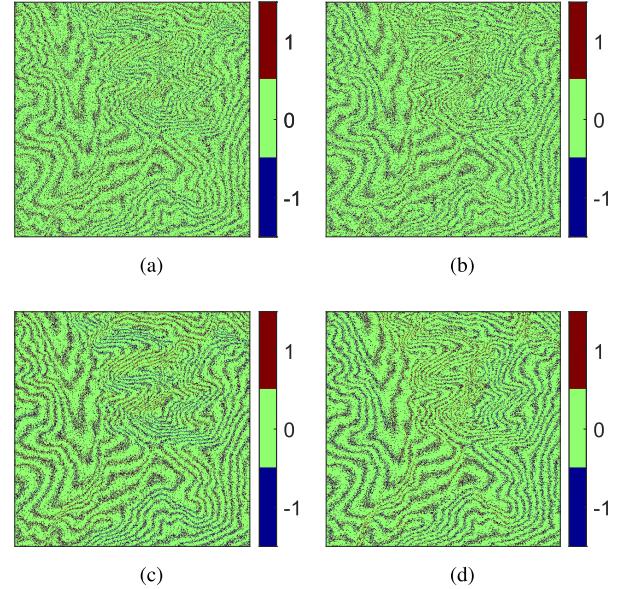


Fig. 6. (a) Vertical phase gradient image obtained by the phase continuity assumption. (b) Horizontal phase gradient image obtained by the phase continuity assumption. (c) Vertical phase gradient image obtained by PGNet. (d) Horizontal phase gradient image obtained by PGNet.

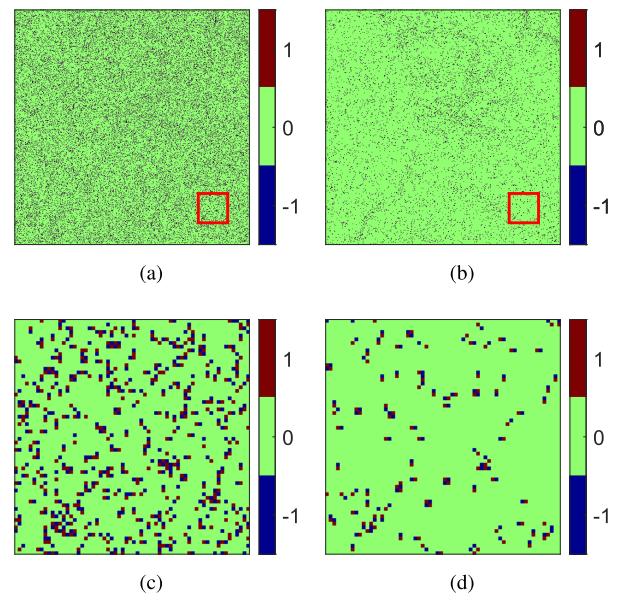


Fig. 7. (a) Residue distribution obtained by Fig. 6(a) and (b). (b) Residue distribution obtained by Fig. 6(c) and (d). (c) Magnified image of the red square in (a). (d) Magnified image of the red square in (b).

it can be clearly found that the residue distribution obtained by PGNet is much sparser than that obtained by the phase continuity assumption.

In addition to the visual demonstration, two most commonly used measurements, the Accuracy and the intersection over union (IoU) [39], are adopted in the 12 tested wrapped phase images for the quantitative analysis. For each class, Accuracy is the ratio of correctly classified pixels to the total number of pixels in that class, according to the ground truth map. IoU, known as the Jaccard similarity coefficient, is the ratio of

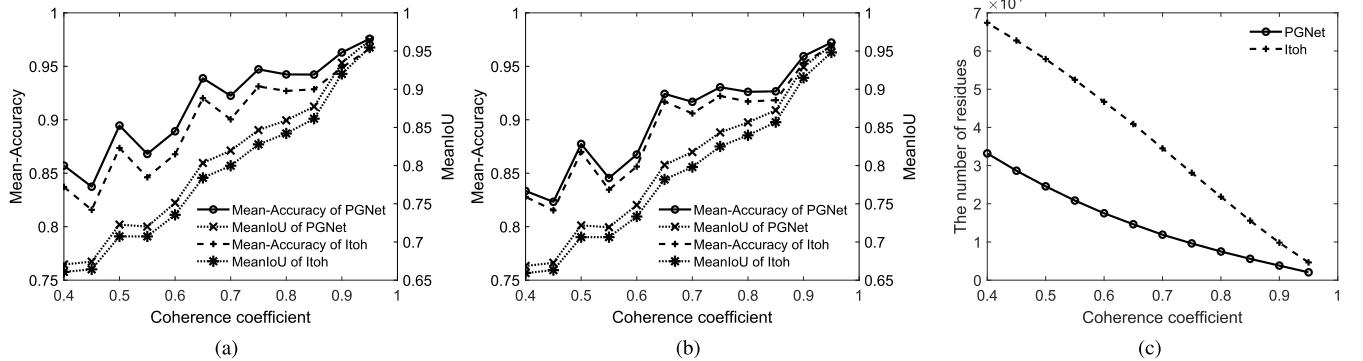


Fig. 8. Relationship between the Mean-Accuracy and MeanIoU of the estimated (a) vertical phase gradient results and (b) horizontal phase gradient results obtained by the two methods and the coherence coefficient. (c) Relationship between the number of residues obtained by the two methods and the coherence coefficient.

TABLE III
QUANTITATIVE EVALUATION OF PHASE GRADIENT ESTIMATION PERFORMANCE ON 12 TEST INPUTS

Methods	Vertical Phase Gradients		Horizontal Phase Gradients		The Total Number of Residues
	Mean-Accuracy	MeanIoU	Mean-Accuracy	MeanIoU	
Phase Continuity Assumption	0.82974	0.74814	0.82848	0.74673	442,604
PGNet	0.88565	0.81321	0.86978	0.80582	179,936

correctly classified pixels to the total number of ground truths and predicted pixels in that class. These measuring metrics are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (11)$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (12)$$

where TP is the number of true-positives, TN is the number of true-negatives, FN is the number of false-negatives, and FP is the number of false-positives. The values of Accuracy and IoU both range from 0 to 1, and the closer to 1 means the better performance. For 12 tested phase images, Mean-Accuracy is the average Accuracy of three classes in all the tested wrapped phase images, and MeanIoU is the average IoU score of three classes in all the tested wrapped phase images. We use Mean-Accuracy in conjunction with MeanIoU for a complete evaluation of phase gradient detection results. The quantitative results of the phase continuity assumption and PGNet are tabulated in Table III. From Table III, it can be observed that the PGNet has the superior performance of estimating the phase gradients to that of the phase continuity assumption under both quantitative indexes. In addition, we also note that the number of the residues generated by PGNet is just 40% around that using the phase continuity assumption.

In order to reveal the performance of the two methods with varying noise levels, the relationship between the two quantitative measures (Mean-Accuracy and MeanIoU) of the estimated phase gradient results obtained by the two methods and the coherence coefficient is shown in Fig. 8(a) and (b). The behavior of these curves in Fig. 8(a) and (b) illustrates that the two quantitative measures of PGNet have higher values than those of the phase continuity assumption against the variation of the coherence coefficient, which demonstrates that the performance of the PGNet suffers less with the degradation

of the coherence coefficient. A similar conclusion can also be noted in Fig. 8(c), where the PGNet has lower numbers of residues compared with that obtained by the phase continuity assumption in the majority of the coherence coefficient scale, especially in the situation of low coherence. It can be seen that PGNet can provide more correct phase gradient information to facilitate the following PU step, i.e., stage 2 of PGNet-PU.

C. Simulation-Based 2-D PU Performance Analysis

In the second experiment, we will examine the PU effectiveness of the PGNet-PU method by using Fig. 5(c) as the test interferogram. The PGNet-PU method is compared with the well-known Branch-cut [5], MCF [10], SNAPHU [11], and PUMA methods [12], respectively. The source code of the Branch-cut method is provided by the book [40], and that of the Delaunay triangulation version of the MCF method is supplied by the GAMMA software [41]. The source codes of SNAPHU and PUMA are both from their algorithm designers [42], [43]. In this experiment, the statistical-cost mode of SNAPHU is “TOPO”, and the clique potential exponent of PUMA is 0.5 (to be kept in experiment 3). Fig. 9(a)–(e) shows the DEM solutions, obtained by the PU results of the Branch-cut, MCF, SNAPHU, PUMA and PGNet-PU methods, of Fig. 5(c), respectively. To increase the PU difficulty and sufficiently distinguish the performance difference on the phase gradient estimation caused by the phase continuity assumption and PGNet, we unweight all the 2-D PU methods on the purpose and no filtering processing is performed for the wrapped phase image before PU (to be kept in experiment 3). Fig. 9(f)–(j) shows the corresponding errors between Fig. 5(a) and Fig. 9(a)–(e). It is worth mentioning that, in order to effectively and fairly compare the results, the same reference point, scale, and range of the color bar are used in the DEMs shown in Fig. 9(a)–(e) and the corresponding errors shown in Fig. 9(f)–(j) (similarly hereinafter in experiment 3). It can

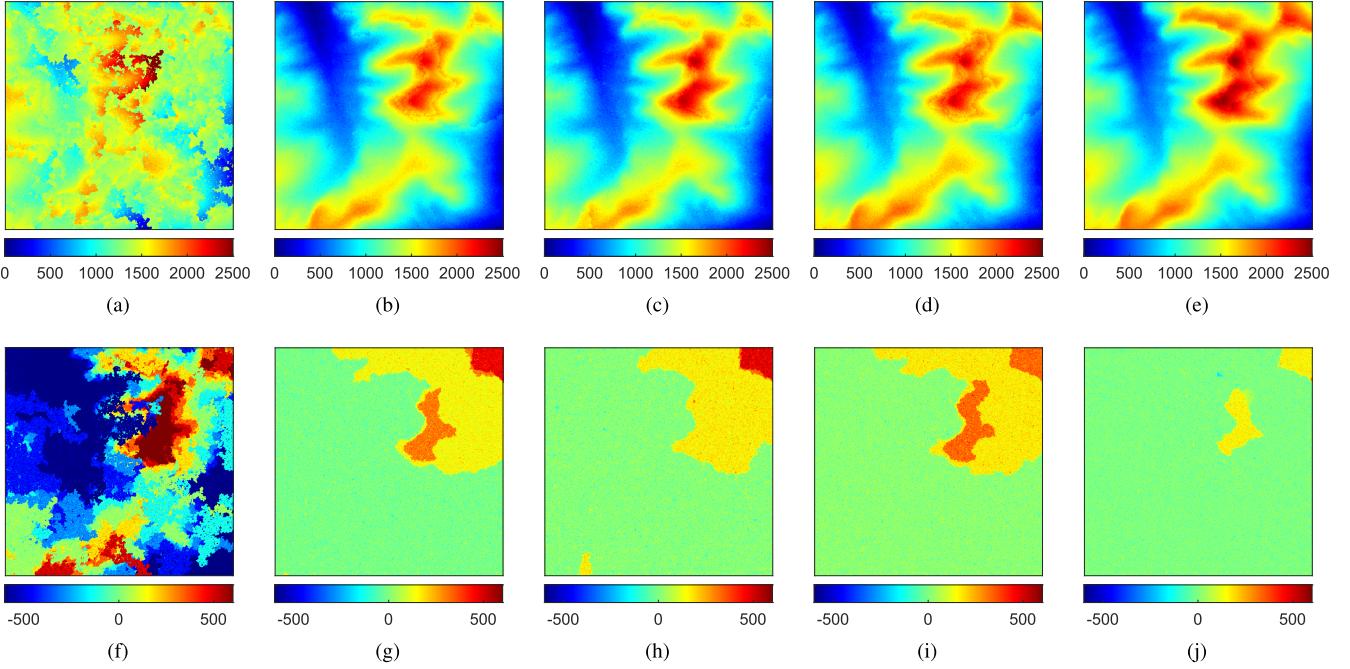
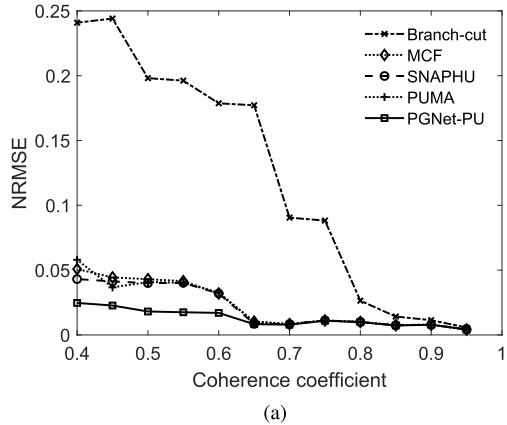


Fig. 9. (a) DEM estimated by the Branch-cut method. (b) DEM estimated by MCF. (c) DEM estimated by SNAPHU. (d) DEM estimated by PUMA. (e) DEM estimated by PGNet-PU. (f) Errors between (a) and Fig. 5(a). (g) Errors between (b) and Fig. 5(a). (h) Errors between (c) and Fig. 5(a). (i) Errors between (d) and Fig. 5(a). (j) Errors between (e) and Fig. 5(a).



(a)

Fig. 10. Relationship between the NRMSE of each method and the coherence coefficient.

be seen that the Branch-cut method fails completely, and the reason is that the incorrect gradient information from the phase continuity assumption will cause the Branch-cut method cannot effectively choose the integration path. In addition, we find that the MCF, SNAPHU, and PUMA methods can all guarantee the PU accuracy in the high-quality regions of Fig. 5(c), but they still produce some unwrapping errors in the right up corner area of the image. This is because, although the applied optimization models in these methods can improve their PU performance by using the global gradient information, the accuracy of these methods can still be decreased by the wrong gradient information from the phase continuity assumption. However, it can be found that the PGNet-PU method avoids most of the unwrapping errors. The reason is that PGNet applied in stage 1 of PGNet-PU can accumulate

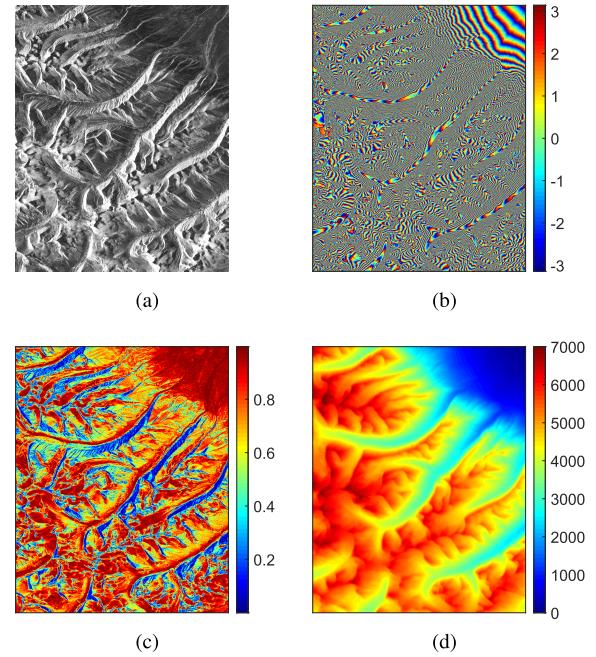


Fig. 11. (a) SAR intensity image. (b) Flattened real interferogram. (c) Coherence map of (b). (d) Reference DEM of (b).

and learn the patterns of the correct phase gradients, leading to more correct phase gradient information compared with the phase continuity assumption.

To quantitatively evaluate the robustness of the PGNet-PU method, the normalized root-mean-square error (NRMSE) [44] is defined as

$$\text{NRMSE} = \sqrt{\frac{\|\hat{h} - h\|^2}{\|h\|^2}} \quad (13)$$

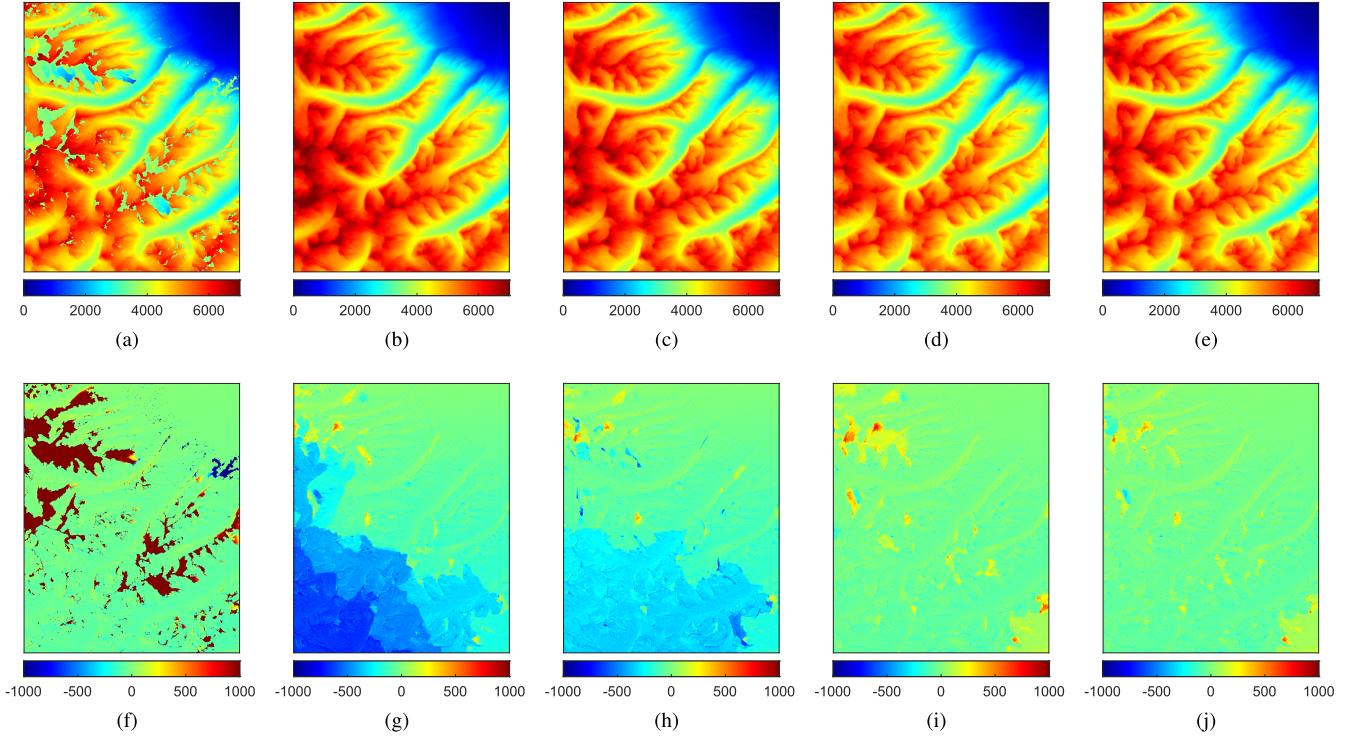


Fig. 12. (a) DEM estimated by the Branch-cut method. (b) DEM estimated by MCF. (c) DEM estimated by SNAPHU. (d) DEM estimated by PUMA. (e) DEM estimated by PGNet-PU. (f) Errors between (a) and Fig. 11(d). (g) Errors between (b) and Fig. 11(d). (h) Errors between (c) and Fig. 11(d). (i) Errors between (d) and Fig. 11(d). (j) Errors between (e) and Fig. 11(d).

TABLE IV
NRMSEs OF FIG. 9(f)–(j)

PU methods	Figures	NRMSEs
Branch-cut	Fig. 9(f)	0.1981
MCF	Fig. 9(g)	0.0430
SNAPHU	Fig. 9(h)	0.0400
PUMA	Fig. 9(i)	0.0408
PGNet-PU	Fig. 9(j)	0.0181

where h is the vector collecting from the reference terrain height, \hat{h} is the vector collecting from the estimated terrain height, and $\|\cdot\|^2$ is the quadratic norm. The NRMSEs of Fig. 9(f)–(j) are shown in Table IV. In Table IV, it can be observed that the Branch-cut method causes the highest NRMSE. In addition, the NRMSEs of the MCF, SNAPHU, and PUMA methods are similar, and they are much lower than those of Branch-cut. However, by owning more correct phase gradient information obtained by PGNet, the NRMSE of the PGNet-PU method is notably lower than those of the other methods.

Fig. 10 reveals the relationship between the NRMSE and coherence coefficient of each PU method on all 12 tested wrapped phase images employed in Section V-B. From Fig. 10, it can be observed that the Branch-cut method generates the highest NRMSE throughout the whole coherence coefficient scale. In the case with coherence above 0.7, the other four methods have similar PU accuracy performance. However, when the coherence is lower than 0.7, the NRMSE of the PGNet-PU method is lower than those of the other competing methods. This indicates that the gradient information

TABLE V
NRMSEs OF FIG. 12(f)–(j)

PU methods	Figures	NRMSEs
Branch-cut	Fig. 12(f)	0.1054
MCF	Fig. 12(g)	0.0423
SNAPHU	Fig. 12(h)	0.0283
PUMA	Fig. 12(i)	0.0107
PGNet-PU	Fig. 12(j)	0.0088

obtained by PGNet has better robustness than that obtained by the phase continuity assumption.

D. Validation of PU Performance on Real Case

The third experiment tests the performance of the PGNet-PU method on the real interferogram. In this experiment, the TerraSAR-X-TanDEM-X interferometry image covering the Lhasa, Tibet autonomous region, China, acquired on March 30, 2013, is employed. It is worth mentioning that the trained PGNet in Section V-A is directly applied in this experiment. The SAR intensity image of the study area is shown in Fig. 11(a) (the size is 2500×2000). In Fig. 11(a), the number of multilooks on range and that on azimuth are both 4, and the corresponding resolution of each pixel is approximately 12×12 m. Fig. 11(b) is the flatten interferogram. Fig. 11(c) is the coherence coefficient map of Fig. 11(b). Because the study area is geographically featured by a great deal of rugged and mountainous terrain, the coherence coefficients in some regions are significantly low caused by the layover and shadow effects. Under this condition, the phase continuity assumption cannot be performed

correctly, leading to incorrect PU results using the traditional 2-D PU methods. Fig. 11(d) is the reference DEM of Fig. 11(b) provided by the SRTM DEM.

Fig. 12(a)–(e) shows the DEM solutions, generated by the PU results of the Branch-cut, MCF, SNAPHU, PUMA, and PGNet-PU methods, of Fig. 11(b), respectively. Fig. 12(f)–(j) shows the corresponding errors between Fig. 11(d) and Fig. 12(a)–(e). Table V illustrates the NRMSEs of Fig. 12(f)–(j). Fig. 12(f) reveals several significant errors caused by the wrong PU. The reason is that the phase gradient information estimated by the phase continuity assumption is so inaccurate that the Branch-cut algorithm cannot effectively choose the integration path. In addition, although the MCF method using the L^1 -norm optimization model to estimate the absolute phase, from Fig. 12(g), we still can observe the PU errors spread in the left bottom corner area of the image, which could be caused by the wrong phase gradient information from the phase continuity assumption as well. From Fig. 12(h), it can be found that the SNAPHU method relatively outperforms the Branch-cut and MCF methods, because it uses a statistical-cost, network-flow model. However, Fig. 12(h) still shows the agglomerative DEM errors in the low part of the image. From Fig. 12(i), it can be noted that the PUMA method alleviates most of unwrapping errors in the result. The reason is that the PUMA method employs a 0.5 clique potential exponent, which has the discontinuity preserving ability. However, comparing Fig. 12(i) with Fig. 12(j), we can see that the DEM reconstructed by PUMA is still worse than that from PGNet-PU. In other words, the DEM reconstructed by the PGNet-PU method outperforms the other four methods, even if the PGNet-PU method applied the same PU objective function to that of the MCF method (i.e., L^1 -norm). This is because in stage 1 of PGNet-PU, the trained PGNet can predict the phase gradient information much more correctly than that from the traditional phase continuity assumption.

VI. CONCLUSION

Estimating the absolute phase gradient information is an important process utilized by almost all the 2-D PU methods. Traditionally, the phase gradient estimation relies on the phase continuity assumption, which is sensitive to the violent terrain change or the system noise. Under this condition, how to design a more robust phase gradient estimation method to replace the phase continuity assumption will be an important problem for 2-D PU. In this article, we presented a novel DCNN (i.e., PGNet) to estimate the phase gradient information instead of the phase continuity assumption. It opens up new possibilities and opportunities to involve DCNN into the PU domain. To be specific, the phase continuity assumption is based on the PU algorithm designers' experience, but PGNet can accumulate and learn the patterns of the correct phase gradients from enormous interferograms with different noise levels and topographic features such that leads to more correct phase gradient information. Subsequently, the PGNet-PU method is proposed, which applies the L^1 -norm objective function to minimize the difference between the unwrapped phase gradients and the gradients from PGNet for obtaining

the final PU result. The experimental results are shown to verify that the PGNet-PU method is an effective and efficient 2-D PU method.

In (4), we restrict the true phase gradient range in $[-1, 1]$. However, we can extend this range to $[-l, l]$ (l is an integer and $l > 1$) by increasing the number of the possible classes from three to $2 \cdot l + 1$. Under this condition, the number of the filters in the standard convolution layer and the value of K in (8) in the pixel classification subnetwork of PGNet both need to change to $2 \cdot l + 1$. To decide the value of l , we could use some prior knowledge, such as approximate terrain information of the observed area. To be specific, if the terrain change of the observed area is large, l should be large as well. It is worth mentioning that the incorrect estimated phase gradient caused by the improper value of l would not affect the PU result directly but generate some extra residues. It means that the optimization model used in stage 2 of PGNet-PU, i.e., (10), still has a chance to choose a reasonable integration path to minimize the effect of the improper value of l .

REFERENCES

- [1] R. Bamler and P. Hartl, "Synthetic aperture radar interferometry," *Inverse Problems*, vol. 14, no. 4, pp. R1–R54, Aug. 1998.
- [2] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013.
- [3] D. C. Ghiglia, G. A. Mastin, and L. A. Romero, "Cellular automata method for phase unwrapping," *J. Opt. Soc. Amer.*, vol. 4, no. 1, pp. 267–280, 1987.
- [4] H. Yu, Y. Lan, Z. Yuan, J. Xu, and H. Lee, "Phase unwrapping in InSAR: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 1, pp. 40–58, Mar. 2019.
- [5] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," *Radio Sci.*, vol. 23, no. 4, pp. 713–720, Jul. 1988.
- [6] W. Xu and I. Cumming, "A region-growing algorithm for InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 124–134, Jan. 1999.
- [7] D. Gao and F. Yin, "Mask cut optimization in two-dimensional phase unwrapping," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 338–342, May 2012.
- [8] T. J. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 14, no. 10, p. 2692, Oct. 1997.
- [9] D. C. Ghiglia and L. A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 11, no. 1, p. 107, Jan. 1994.
- [10] M. Costantini, "A novel phase unwrapping method based on network programming," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 3, pp. 813–821, May 1998.
- [11] C. W. Chen and H. A. Zebker, "Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 18, no. 2, p. 338, Feb. 2001.
- [12] J. M. Bioucas-Dias and G. Valadao, "Phase unwrapping via graph cuts," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 698–709, Mar. 2007.
- [13] Q. Lin, J. Vesely, and H. Zebker, "New approaches in interferometric SAR data processing," *IEEE Trans. Geosci. Remote Sens.*, vol. 30, no. 3, pp. 560–567, May 1992.
- [14] A. Ahmad and Y. Lu, "Identifying the phase discontinuities in the wrapped phase maps by a classification framework," *Opt. Eng.*, vol. 55, no. 3, Mar. 2016, Art. no. 033104.
- [15] H. Yu and Y. Lan, "Robust two-dimensional phase unwrapping for multi-baseline SAR interferograms: A two-stage programming approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5217–5225, Sep. 2016.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

- [17] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.
- [18] D. J. Tipper, D. R. Burton, and M. J. Lalor, "A neural network approach to the phase unwrapping problem in fringe analysis," *Nondestruct. Test. Eval.*, vol. 12, no. 6, pp. 391–400, Jan. 1996.
- [19] S. Hamzah, J. D. Pearson, P. J. Lisboa, and C. A. Hobson, "Phase unwrapping in 3-D shape measurement using artificial neural networks," in *Proc. Int. Conf. Image Process. Appl.*, Dublin, Ireland, Jul. 1997, pp. 680–683.
- [20] W. Schwartzkopf, T. E. Milner, J. Ghosh, B. L. Evans, and A. C. Bovik, "Two-dimensional phase unwrapping using neural networks," in *Proc. 4th IEEE Southwest Symp. Image Anal. Interpretation*, Austin, TX, USA, Apr. 2000, pp. 247–250.
- [21] G. E. Spoorthi, S. Gorthi, and R. K. S. S. Gorthi, "PhaseNet: A deep convolutional neural network for two-dimensional phase unwrapping," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 54–58, Jan. 2019.
- [22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.
- [23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [24] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, "Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 59–69, Apr. 2019.
- [25] M. Shahzad, M. Maurer, F. Fraundorfer, Y. Wang, and X. X. Zhu, "Buildings detection in VHR SAR images using fully convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1100–1116, Feb. 2019.
- [26] L. Mou and X. Zhu, "RiFCN: Recurrent network in fully convolutional network for semantic segmentation of high resolution remote sensing images," *CoRR*, May 2018. [Online]. Available: <https://arxiv.org/abs/1805.02091>
- [27] J. Kang, M. Körner, Y. Wang, H. Taubenböck, and X. X. Zhu, "Building instance classification using street view images," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 44–59, Nov. 2018.
- [28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.* San Juan, Puerto Rico: Caribe Hilton, May 2016, pp. 1–13.
- [29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
- [30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Mach. Learn. Res.*, Lille, France, Jul. 2015, pp. 448–456.
- [31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 807–814.
- [32] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [33] J.-S. Lee, K. Hoppel, S. Mango, and A. Miller, "Intensity and phase statistics of multilook polarimetric and interferometric SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 5, pp. 1017–1028, Sep. 1994.
- [34] G. Liu, R. Wang, Y. Deng, R. Chen, Y. Shao, and Z. Yuan, "A new quality map for 2-D phase unwrapping based on gray level co-occurrence matrix," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 2, pp. 444–448, Feb. 2014.
- [35] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [36] H. Yu, M. Xing, and Z. Bao, "A fast phase unwrapping method for large-scale interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4240–4248, Jul. 2013.
- [37] H. Yu, Y. Lan, J. Xu, D. An, and H. Lee, "Large-scale L^0 -norm and L^1 -norm two-dimensional phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4712–4728, Aug. 2017.
- [38] H. Yu, Y. Zhou, S. S. Ivey, and Y. Lan, "Large-scale multibaseline phase unwrapping: Interferogram segmentation based on multibaseline envelope-sparsity theorem," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9308–9322, Nov. 2019.
- [39] G. Csurka, D. Larlus, and F. Perronnin, "What is a good evaluation measure for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, Bristol, U.K., Sep. 2013, pp. 32.1–32.11.
- [40] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*. New York, NY, USA: Wiley, 1998.
- [41] *Interferometric SAR Processing: GAMMA Remote Sensing*, GAMMA Remote Sensing AG, Gmunden, Switzerland, 2019.
- [42] SNAPHU. (2019). [Online]. Available: <https://web.stanford.edu/group/radar/softwareandlinks/sw/snaphu/>
- [43] PUMA. (2019). [Online]. Available: <http://www.lx.it.pt/bioucas/code.htm>
- [44] F. Baselice, G. Ferraioli, V. Pascazio, and G. Schirinzi, "Contextual information-based multichannel synthetic aperture radar interferometry: Addressing DEM reconstruction using contextual information," *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 59–68, Jul. 2014.



Lifan Zhou (Member, IEEE) was born in Suzhou, Jiangsu, China, in 1984. He received the B.S. degree in geographic information system from the Wuhan University, Wuhan, China, in 2006, and the Ph.D. degree in cartography and geographic information system from the Zhejiang University, Hangzhou, China, in 2014.

Since 2014, he has been a Lecturer with the School of Computer Science and Engineering, Changshu Institute of Technology, China. In 2016, he was a Visiting Scientist with The Chinese University of Hong Kong, Hong Kong. His main research interests are in the fields of phase unwrapping, algorithm design, and machine learning.



Hanwen Yu (Senior Member, IEEE) was born in Xi'an, Shaanxi, China, in 1985. He received the B.S. and Ph.D. degrees in electronic engineering from the Xidian University, Xi'an, in 2007 and 2012, respectively, and the M.S. degree in computer science from the University of Memphis, Memphis, TN, USA, in 2015. He is currently a Post-Doctoral Research Fellow with the Department of Civil and Environmental Engineering, National Center for Airborne Laser Mapping, University of Houston, Houston, TX, USA. He has authored or coauthored more than 20 research articles in high-impact peer-reviewed journals, such as the *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING* and the *IEEE TRANSACTIONS ON IMAGE PROCESSING*, and *Remote Sensing of Environment*. His research interests include phase unwrapping, algorithm design, and synthetic aperture radar interferometry signal processing and application.

Dr. Yu serves as a Guest Editor of the Special Issue "InSAR in Remote Sensing" of the *IEEE Geoscience and Remote Sensing Magazine* and the Special Issue "InSAR Signal and Data Processing" of *Sensors*.



Yang Lan was born in Shaanxi, China. She received the B.S. degree in communication engineering and the M.S. degree in information and communication engineering from the Xidian University, Xi'an, China, in 1997 and 2005, respectively, where she is currently pursuing the Ph.D. degree with the National Laboratory of Radar Signal Processing.

Her research interests include phase unwrapping and synthetic aperture radar interferometry (InSAR) signal processing.