Reconstruction of Missing Ground-Penetrating Radar Traces Using Simplified U-Net

Budiman P. A. Rohman, *Student Member, IEEE*, Masahiko Nishimoto, *Member, IEEE*, and Kohichi Ogata, *Member, IEEE*

***Abstract*— Ground-penetrating radar is an effective tool for the exploration of subsurface and obscured objects by exploiting the electromagnetic wave characteristics. However, some problems in the measurement process produce missing traces. The resulting incomplete data create difficulties in visualization, interpretation, and analysis. This article proposes a lightweight deep learning method to reconstruct the missing traces. The model is a simplification of a well-known convolutional neural network model, U-Net, which performs well in this context. We use a multi-and-wide kernel size, shallow convolutional layers, a reduced encoding–decoding step, and a minimized feature map size to minimize the model while maintaining high performance. In this study, we focus on the case of inspection of reinforced concrete, where the examined structure consists of a series of rebar and/or void cracks. The numerical and experimental field data results indicate that the method can reconstruct the missing traces with high accuracy. The proposed method also shows comparable performance with U-Net, although the model size is more than 45 times smaller, and its computation speed is more than three times faster. Thus, the proposed method is promising for use in a wide range of applications and devices.**

***Index Terms*— Convolutional neural network, deep learning (DL), ground-penetrating radar (GPR), image reconstruction.**

I. Introduction

GROUND-PENETRATING radar (GPR) is a nondestructive tool for exploring subsurface objects using electromagnetic waves. This tool is a mature technology that has been widely applied in the fields of geophysics, geology, humanitarian demining, and civil engineering [1]. GPR works by transmitting and receiving electromagnetic waves. The waves are distorted by the differences in electromagnetic properties, such as permittivity and conductivity, allowing identification of buried object characteristics.

In real applications, the trace data collected from the field are frequently incomplete for reasons such as construction conditions, technical failures, and the presence of obstacles [2], [3]. These missing and/or corrupt traces negatively affect the identification, interpretation, and analysis of GPR data. To address these issues, researchers commonly use simple linear interpolation methods or ignore vacant channels; each of these approaches has disadvantages. Some new methods have been proposed that use compressive sensing [2] and f-k migration-based [3] techniques. However, those methods require additional assumptions and information, such as the sparsity characteristic of the signal and medium permittivity information. Other innovative interpolation techniques have also been proposed [4], but such methods do not perform well when the lost traces are large, in consecutive traces, or when more than half of the B-scan data are missing.

The success of deep learning (DL) algorithms in many problems associated with computer vision, such as object detection [5], object classification [6], and segmentation [7], has motivated many researchers to implement this data-driven method in other applications. In GPR research, DL has been used for several purposes, including object detection and recognition [8]–[10]. Recently, researchers have also used this technique for solving the wave inversion problem and for reconstructing the internal structure of targeted materials. The translation of radar images using several types of DL was conducted by Alvarez and Kodagoda [11]. The use of DL to reconstruct the radar image was proposed in [12].

DL is now also being applied to the recovery of missing GPR and seismic data. The use of DL techniques in this context avoids theoretical assumptions regarding features such as linearity, sparsity, and band limitation [13]. However, most DL approaches, such as U-Net [13], generative adversarial network (GAN) [14], and conditional GAN [15], have a very large model size. It is known that very deep and large DL models have several disadvantages, especially in real applications, including large storage needs, high computational expense, high energy use, and difficulty in deploying on edge devices [16]. In some specific cases, such as large observation missions that produce a large amount of data, this kind of DL model can also lead to a critical issue in the processing time. The problem becomes even more complex when these methods are intended for deployment on limited-specification hardware, such as drone-mounted radar that requires on-site/borne and real-time processing, as in [17] and [18].

In this study, we propose a lightweight convolutional neural network method for the reconstruction of missing GPR traces, developed as a simplification of the U-Net model. Since the target reflection image on GPR with common-offset measurement is typically represented as a hyperbolic and linear segment pattern regardless of the actual shape of the buried object [19], we hypothesize that a data-driven method with a relatively simple structure will be sufficient to solve this problem. The simple structure of the DL model will make

Manuscript received March 18, 2021; accepted March 28, 2021. Date of publication April 26, 2021; date of current version December 20, 2021. *(Corresponding author: Budiman P. A. Rohman.)*

Budiman P. A. Rohman is with the Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan, on leave from the Research Center for Electronics and Telecommunication, Indonesian Institute of Sciences, Bandung 40135, Indonesia (e-mail: budi028@lipi.go.id).

Masahiko Nishimoto and Kohichi Ogata are with the Faculty of Advanced Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan (e-mail: nisimoto@cs.kumamoto-u.ac.jp; ogata@cs.kumamoto-u.ac.jp).

Digital Object Identifier 10.1109/LGRS.2021.3072028

1558-0571 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on May 13,2023 at 12:19:27 UTC from IEEE Xplore. Restrictions apply.

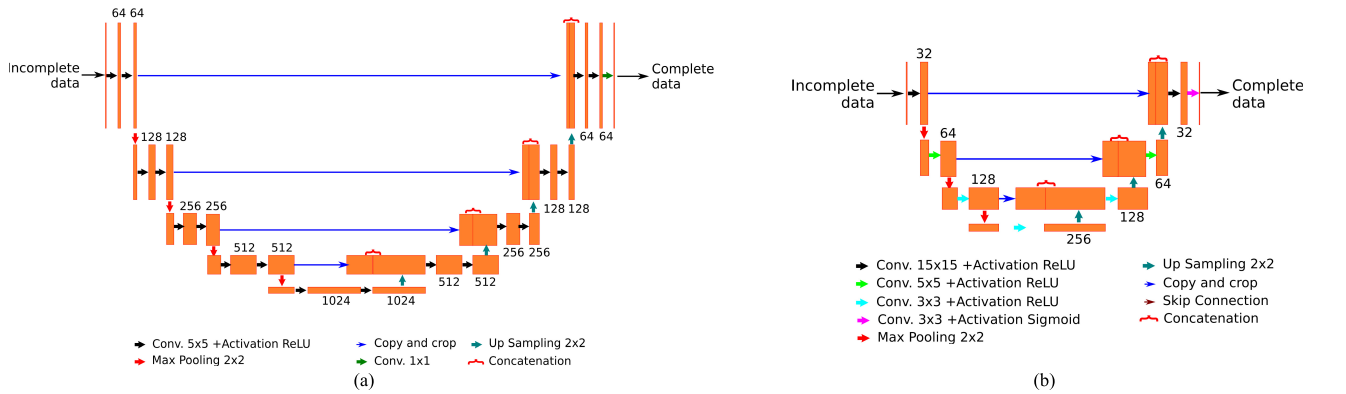


Fig. 1. CNN model. (a) U-Net [12]. (b) Proposed model.

the method computationally efficient and applicable for a wide range of possible devices, such as drone-mounted GPR. In addition, because the training process entails memorizing possible hyperbolic pattern features via DL, this algorithm may be able to address wide consecutive missing traces that cannot be solved by the existing methods.

II. Methodology

DL is a subset of machine learning that works by processing multiple layers to extract the high-level features to build a nonlinear parametrized mapping. The backbone of this algorithm is the artificial neural network that imitates human neurons [20]. If DL is a function of f

$$\boldsymbol{Y_P} = f(\boldsymbol{X}, \boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{Y_P}$ is the prediction value, $\boldsymbol{X}$ is the data input, and $\boldsymbol{\theta}$ is the DL parameter that commonly consists of but is not limited to weight and bias values. In the learning process, the parameter of the neural network is iteratively optimized to minimize the error estimation between the estimated and target values.

Convolutional neural networks are the most popular architecture of DL algorithms and are commonly used for image processing. This model commonly consists of multiple building blocks, including convolution layers, pooling layers, and fully connected layers. It is designed to automatically learn the spatial hierarchies of features through a backpropagation algorithm.

A. U-Net

U-Net is a model of a fully convolutional neural network that was originally proposed for semantic segmentation in biomedical images [21]. It has since been shown [13] to also work well for the reconstruction of missing seismic traces.

The architecture of U-Net is illustrated in Fig. 1(a). The model consists of two symmetrical parts: an encoding path and a decoding path. The encoding path consists of two repeated 5×5 convolutions, where each convolution is followed by a rectified linear unit (ReLU) activation and a 2×2 max pooling operation with the same padding size for the downsampling process. Then, every step of the decoding path consists of upsampling followed by a 5×5 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map in the encoding path, and two 5×5 convolutions, each followed by an ReLU. These copy step or skip connections provide local information to the global level during upsampling. In the last step, a 1×1 convolution is used to map the feature vector to the output image. This model uses 19 convolutional layers.

U-Net has several advantages. First, this model uses the global location and context simultaneously. Second, it can work with very few training samples. U-Net's end-to-end processing allows the model to preserve the full context of the input.

B. Simplified U-Net

In this study, we propose a simplified version of U-Net from [13] to reconstruct missing GPR traces. The objective is to minimize the size of the model while maintaining the high performance of U-Net to allow it to be applied to a wide range of applications and devices.

We apply four strategies for this simplification. First, we take a multikernel size in each layer that is wider than the original U-Net. This multikernel size approach is taken to anticipate the differences in the spatial resolution of the recorded radar images. In the convolution process, we use a relatively large kernel size to gain as much context information as possible from the neighboring pixels since GPR images commonly contain hyperbolic and linear patterns. This large kernel size is also used to anticipate any wide missing traces in the consecutive pixels that might appear in cases with substantial missing data. The structure of U-Net is suggested to maintain the detail of the image feature [21] so that with this kernel size, the object image detail can be preserved. Second, we simplify the repeated double convolutional layers to be single. This step is taken to optimize the feature number, which otherwise causes the model to grow large and makes computation expensive. Considering that our purpose is to reconstruct the image, instead of segmentation, the shallower layer will be sufficient. Thus, we prune the encoding–decoding paths because we expect that a very large structure is not needed to solve GPR target images. Finally, we minimize the feature map size to be one-fourth the size of the U-Net in all convolution layers.

Our proposed model is shown in detail in Fig. 1(b). Similar to the common U-Net model, our model consists of encoding and decoding paths. Every step of the encoding path consists of a single convolution followed by an ReLU and

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on May 13,2023 at 12:19:27 UTC from IEEE Xplore. Restrictions apply.

a 2 × 2 max pooling. In this path, the kernel size of the convolution operation in the first and second steps is 15 × 15 and 5 × 5, respectively. The remaining encoding step uses a 3 × 3 kernel size. This size configuration is set to mirror the decoding paths in the corresponding step. The skip connection is also used and works as in the original U-Net. In the final layer, a 3 × 3 convolution with activation function sigmoid is used to map the feature vector to the image. Our proposed model uses a total of eight convolution layers.

C. Hyperparameter

In the training process, we use an Adam optimizer derived from adaptive moment estimation. Then, we use the mean square error (MSE) for the loss function, defined as

$$\text{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(Y_T(i,j) - Y_P(i,j))^2 \tag{2}$$

where Y_T is the target output, m and n are the sizes of the rows and columns of data, respectively, and i and j are the elements of the data, respectively.

D. Evaluation Metrics

To more quantitatively evaluate the performance of our proposed model, we also use the mean absolute error (MAE) and peak signal-to-noise ratio (PSNR). The MAE is a measure of the average of the absolute value of discrepancy between the estimated and actual values, calculated as

$$\text{MAE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\|Y_T(i,j) - Y_P(i,j)\|. \tag{3}$$

The PSNR is the ratio between the maximum power of a signal and the power of distorted noise. The PSNR is defined as

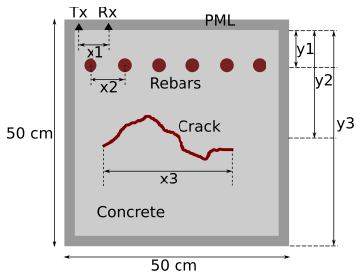
$$\text{PSNR} = 10.\log_{10}\left\{\frac{\max(\boldsymbol{Y_P})^2}{\text{MSE}}\right\}. \tag{4}$$

III. Experimental Study and Results

A. Data Set

For training and testing of the method, we use synthetic data generated by gprMax, a finite-difference time-domain-based electromagnetic simulation software [22]. Our simulation works using a spatial discretization of 1 mm. For the radar antenna, we use a theoretical Hertzian dipole antenna with a range of 4 cm between transmitter and the receiver. The transmitted waveform is a monocycle pulse with a center frequency of 2 GHz.

As mentioned before, we focus on reinforced concrete inspection where we consider three cases: concrete with rebar (case I); concrete with cracks (case II); and concrete with rebar and cracks (case III). The condition of each component is configured with various depths, radii, and spaces. Approximately 2000 training, 500 validation, and 500 test data were used in each case. The missing traces were selected randomly, and the value was filled with the mean value of recorded traces. We consider three percentages of missing traces: 20%, 50%, and 70%. Thus, we have 18 000 images for training, 4500 for testing, and 4500 for validation. The recorded B-scan is resized so that the size of the image is 160 × 160 pixels. Fig. 2 illustrates our data set for the input of our model.



Component	ϵ_r	σ	Dimension (cm)
Concrete	6–8	0.001	width (w) = 50 height (y3) = 49
Rebar (PEC)	1	∞	radius (r) = 0.5–1.5 space (x2) = 4r–10 depth (y1) = 5–10
Crack	1	0	length (x3) = 5–30 thickness (t) = 0.5–2.0 depth (y2) = 10–20

Fig. 2. Simulated model for generating synthetic data.

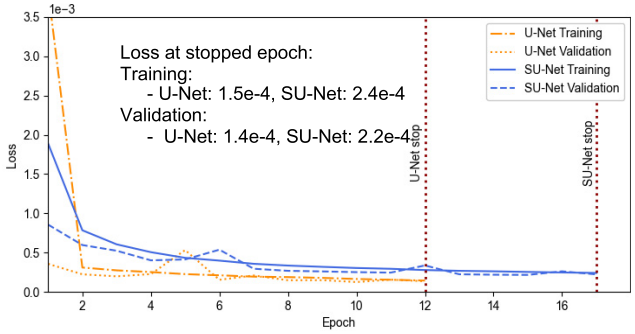


Fig. 3. Comparison of loss variation during training between baseline and proposed methods.

B. Implementation

The proposed model was implemented on our workstation running a 64-bit Windows 10 Pro Intel Core i7-8700K CPU with a clock in each 3.70-GHz core (12 CPUs), 32-GB RAM, and an NVIDIA RTX 2070 Super GPU. The code was written in Python version 3.6, with TensorFlow as the backend and Keras as the framework. We ran the training with a minibatch of size 8. The learning rate was initially 1^{-4} with a decay of 1^{-5}. For a fair comparison, we used identical parameters for the learning process of the proposed and baseline methods. To avoid overfitting, we used an early-check function on the validation loss to end the training when the performance stopped improving with a delay of ten epochs.

The training process for both the methods is shown in Fig. 3. It can be confirmed that no overfitting occurred. In the stopped epoch, the loss value of U-Net in both the training and validation sets is slightly lower, less than 1^{-4}, than that of our model (SU-Net).

C. Comparisons

A comparison of the results of the testing data in case III (selected randomly) is shown in Fig. 4. Our model can reconstruct the missing data. Qualitatively, the result of our proposed model is almost the same as the result of U-Net.

Table I compares the metrics of reconstruction using U-Net and our method in terms of the MSE, MAE, and PSNR. As shown, the recovery performance slightly degrades as the percentage of missing trace data increases. This is

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on May 13,2023 at 12:19:27 UTC from IEEE Xplore. Restrictions apply.

TABLE I

COMPARISON OF RECONSTRUCTION OF SYNTHETIC DATA

Case	%	MSE		MAE		PSNR (dB)	
		U-Net	SU-Net	U-Net	SU-Net	U-Net	SU-Net
I	20	0.00123	**0.00118**	0.02175	**0.02115**	29.297	**29.457**
	50	**0.00126**	0.00128	**0.02237**	0.02247	**29.171**	29.126
	70	**0.00140**	0.00163	**0.02448**	0.02674	**28.752**	28.161
II	20	**0.00171**	**0.00171**	0.02137	**0.02090**	28.859	**28.960**
	50	**0.00177**	0.00180	0.02242	**0.02221**	**28.642**	28.615
	70	**0.00197**	0.00204	0.02567	**0.02555**	**28.014**	27.893
III	20	0.00124	**0.00120**	0.02169	**0.02125**	29.231	**29.361**
	50	**0.00129**	0.00131	**0.02255**	0.02288	**29.063**	28.984
	70	**0.00145**	0.00169	**0.02502**	0.02737	**28.584**	27.985
Average		0.00148	0.00153	0.02304	0.02339	28.846	28.727

TABLE II

COMPARISON OF THE MODEL CHARACTERISTICS

Characteristic	U-Net	SU-Net	Ratio
Total convolutional layer	19	8	2.1:1
Total parameters	87.1 million	1.9 million	45.8:1
Model size on storage	1 GB	0.02 GB	46.5:1
Processing time in:			
- Training (18000 data)	9799 s (12 epoch)	2880 s (17 epoch)	3.4:1
- Testing (22500 data)	138.64 s	39.61 s	3.5:1

understandable since a larger percentage of missing data results in a more difficult problem. We can also see the comparison of performance between our method and U-Net proposed in [13]. The results indicate that our method's performance can keep up with that of U-Net. For our method, the performance did degrade, but the difference was not substantial. In addition, in other cases, especially in cases with 20% missing data, our method consistently outperforms U-Net. From the average value perspective, our method's performance is lower than that of U-Net, but the difference is minor. Our method is a far simpler version of U-Net that reduces some structure and parameters so that the information obtained may also be slightly lost.

Table II compares the model size and computation time of the methods. Compared with U-Net, our model is much simpler with fewer than half the convolutional layers. The total parameter number and storage size of our model are far smaller than those of U-Net by a factor of more than 45. In the training process, even though the epoch number for U-Net is smaller, the time required is longer than that of our method. In addition, in the inference processing time, our method is more than three times faster than U-Net. This comparison shows that our method will be feasible to apply in some applications where U-Net cannot be efficiently used, especially in limited specification devices such as older or cheaper workstations/CPUs, mobile devices, and aerial vehicle radar. Our method is also more applicable for large observation areas.

D. Evaluation of Real Radar Data

To validate the practicality of the method on real radar B-scan images, we check the functioning of our method with two real cases. The experimental setup is shown in Fig. 5. For the radar, we use ultrawideband impulse radar from Xethru with a frequency bandwidth of 1.5-6 GHz [23]. The transmitted pulse is a monocycle pulse with a sampling frequency of 39 GS/s.

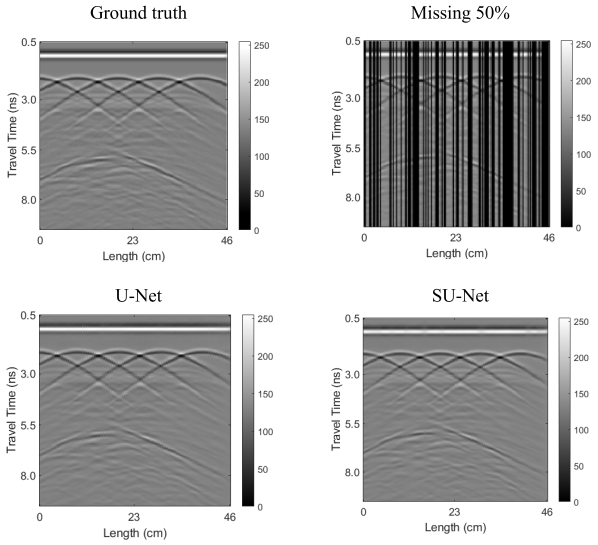


Fig. 4. Result of reconstruction on randomly selected synthetic testing data in case III, with 50% missing traces.

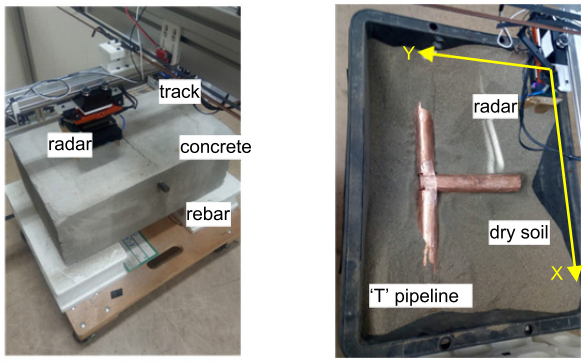


Fig. 5. Experimental setup using UWB radar on the case. (a) Reinforced concrete. (b) T-shaped pipeline buried under soil.

TABLE III

COMPARISON OF RECONSTRUCTION OF REINFORCED CONCRETE DATA

%	MSE		MAE		PSNR (dB)	
	U-Net	SU-Net	U-Net	SU-Net	U-Net	SU-Net
20	0.00107	**0.00084**	0.02282	**0.01537**	29.687	**30.748**
50	**0.00106**	0.00110	0.02203	**0.02056**	**29.759**	29.564
70	**0.00122**	0.00115	**0.02520**	0.02640	**29.129**	28.201

In the first case [Case A, Fig. 5(a)], we inspect the reinforced concrete with a range between radar and concrete of approximately 4 cm and a spatial step of 4 mm. Metal rebar 1.5 cm in diameter is embedded inside the concrete at a depth of approximately 3 cm. In Case B [Fig. 5(b)], we use the same radar to inspect a T-shaped pipeline buried under dry soil to form a 3-D image. The pipeline is cylindrical paper coated with copper tape with a diameter of 4 cm buried approximately 4 cm below the surface. The radar-soil distance is approximately 5 cm, and the spatial step is approximately 4 and 3 mm for the x- and y-axes, respectively.

A comparison of reconstruction using our method and U-Net in Case A is shown in Fig. 6, which confirms that our proposed method works on real radar data. Even though the center frequency of radar in our training step was different, our method was still effective. The performance of

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on May 13,2023 at 12:19:27 UTC from IEEE Xplore. Restrictions apply.

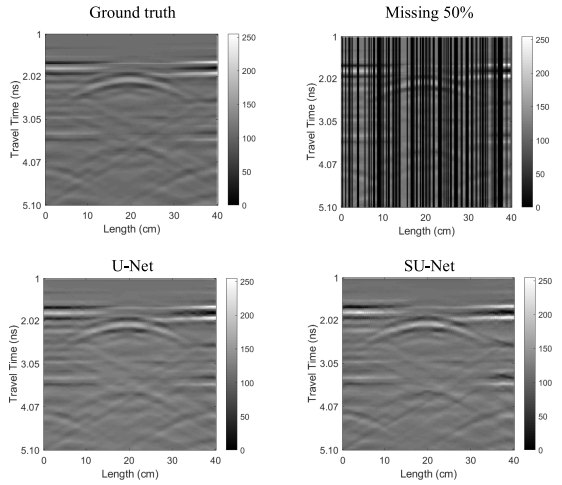


Fig. 6. Result of reconstruction on the radar image of reinforced concrete.

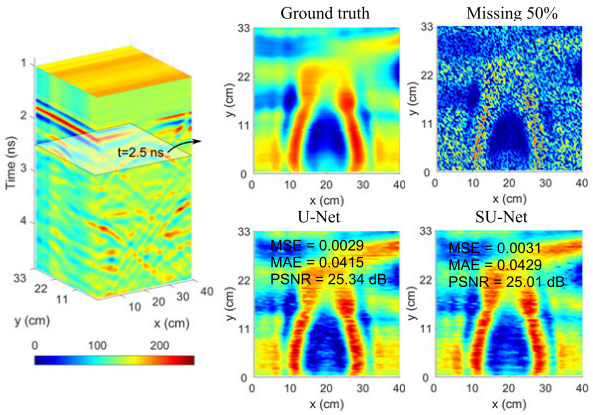


Fig. 7. Result of reconstruction effect on horizontal surface of 3-D radar imaging of buried pipeline with 50% missing traces.

the reconstruction is comparable to that of U-Net, with only minor discrepancies. These visual evaluations are confirmed by numerical analysis (Table III). The comparison on the horizontal slice (at a travel time of approximately 2.5 ns) of the 3-D image in Case B is shown in Fig. 7. This figure shows that our method works almost similar to U-Net. Thus, we can confidently confirm that our method works and is comparable to U-Net when applied to field data.

IV. Conclusion

In this letter, we propose the use of a lightweight convolutional neural network for the recovery of missing GPR traces. The proposed method simplifies the U-Net model by several minimization strategies: using multiple and larger kernel sizes to optimize information collection, reducing the number of convolution layers, pruning the encoding–decoding path, and minimizing the feature map size. An evaluation of both synthetic and experimental data reveals that the model works accurately in the reconstruction of missing traces. The proposed method shows comparable performance with the baseline method although the model size is more than 45 times smaller, and its training and inference computation is more than three times faster. Based on these results, our method is feasible and promising for several applications, especially in cases that require accuracy, small storage, lightweight computation, and fast reconstruction.

References

[1] H. M. Jol, *Ground Penetrating Radar Theory and Applications*, 1st ed. Oxford, U.K.: Elsevier, 2008, pp. 4–40.
[2] J. Xu, Z. Shen, and Z. Tian, "GPR data reconstruction method based on compressive sensing and K-SVD," *Near Surf. Geophys.*, vol. 16, no. 1, pp. 13–21, Feb. 2018.
[3] L. Yi, K. Takahashi, and M. Sato, "A fast iterative interpolation method in f-k domain for 3-D irregularly sampled GPR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 1, pp. 9–17, Jan. 2016.
[4] G. Safont, A. Salazar, A. Rodriguez, and L. Vergara, "On recovering missing ground penetrating radar traces by statistical interpolation methods," *Remote Sens.*, vol. 6, no. 8, pp. 7546–7565, Aug. 2014.
[5] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.
[6] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2203–2215, Sep. 2018.
[7] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[8] M.-T. Pham and S. Lefevre, "Buried object detection from B-scan ground penetrating radar data using faster-RCNN," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 6804–6807.
[9] Z. Tong, J. Gao, and H. Zhang, "Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks," *Construction Building Mater.*, vol. 146, pp. 775–787, Aug. 2017.
[10] H. Liu, C. Lin, J. Cui, L. Fan, X. Xie, and B. F. Spencer, "Detection and localization of rebar in concrete by deep learning using ground penetrating radar," *Autom. Construct.*, vol. 118, Oct. 2020, Art. no. 103279.
[11] J. K. Alvarez and S. Kodagoda, "Application of deep learning image-to-image transformation networks to GPR radargrams for sub-surface imaging in infrastructure monitoring," in *Proc. 13th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, May 2018, pp. 611–616.
[12] B. Liu *et al.*, "GPRInvNet: Deep learning-based ground-penetrating radar data inversion for tunnel linings," *IEEE Trans. Geosci. Remote Sens.*, early access, 13, Jan. 2021, doi: 10.1109/TGRS.2020.3046454.
[13] X. Chai, H. Gu, F. Li, H. Duan, X. Hu, and K. Lin, "Deep learning for irregularly and regularly missing data reconstruction," *Sci. Rep.*, vol. 10, no. 1, pp. 1–18, Feb. 2020.
[14] D. Chang, W. Yang, X. Yong, and H. Li, "Generative adversarial networks for seismic data interpolation," in *Proc. SEG Workshop, SEG Maximizing Asset Value Through Artif. Intell. Mach. Learn., Beijing, China, 17–19 September*, Dec. 2018, pp. 40–43.
[15] D. A. B. Oliveira, R. S. Ferreira, R. Silva, and E. V. Brazil, "Interpolating seismic data with conditional generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 12, pp. 1952–1956, Dec. 2018.
[16] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
[17] M. Garcia Fernandez *et al.*, "Synthetic aperture radar imaging system for landmine detection using a ground penetrating radar on board a unmanned aerial vehicle," *IEEE Access*, vol. 6, pp. 45100–45112, 2018.
[18] B. P. A. Rohman, M. B. Andra, H. F. Putra, D. H. Fandiantoro, and M. Nishimoto, "Multisensory surveillance drone for survivor detection and geolocalization in complex post-disaster environment," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 9368–9371.
[19] Q. Dou, L. Wei, D. R. Magee, and A. G. Cohn, "Real-time hyperbola recognition and fitting in GPR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 51–62, Jan. 2017.
[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, Munich, Germany, 2015, pp. 234–241.
[22] C. Warren, A. Giannopoulos, and I. Giannakis, "GprMax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Comput. Phys. Commun.*, vol. 209, pp. 163–170, Dec. 2016.
[23] *Cayenne Radar Development Kit Bow Tie*. Accessed: Sep. 26, 2019. [Online]. Available: https://store.flatearthinc.com/products/cayenne-radar-development-kit

Authorized licensed use limited to: Harbin Institute of Technology. Downloaded on May 13,2023 at 12:19:27 UTC from IEEE Xplore. Restrictions apply.