Title

A Framework of Rapid Regional Tsunami Damage Recognition From Post-event TerraSAR-X Imagery Using Deep Neural Networks

Journal

IEEE Geoscience and Remote Sensing Letters, 15(1)

ISSN

1545-598X

Authors

Bai, Yanbing

Gao, Chang

Singh, Sameer

et al.

Publication Date

2018

DOI

10.1109/lgrs.2017.2772349

A Framework of Rapid Regional Tsunami Damage Recognition From Post-event TerraSAR-X Imagery Using Deep Neural Networks

Yanbing Bai, Chang Gao, Sameer Singh, Magaly Koch, *Member, IEEE*, Bruno Adriano, Erick Mas, and Shunichi Koshimura, *Member, IEEE*

***Abstract*— Near real-time building damage mapping is an indispensable prerequisite for governments to make decisions for disaster relief. With high-resolution synthetic aperture radar (SAR) systems, such as TerraSAR-X, the provision of such products in a fast and effective way becomes possible. In this letter, a deep learning-based framework for rapid regional tsunami damage recognition using post-event SAR imagery is proposed. To perform such a rapid damage mapping, a series of tile-based image split analysis is employed to generate the data set. Next, a selection algorithm with the SqueezeNet network is developed to swiftly distinguish between built-up (BU) and nonbuilt-up regions. Finally, a recognition algorithm with a modified wide residual network is developed to classify the BU regions into wash away, collapsed, and slightly damaged regions. Experiments performed on the TerraSAR-X data from the 2011 Tohoku earthquake and tsunami in Japan show a BU region extraction accuracy of 80.4% and a damage-level recognition accuracy of 74.8%, respectively. Our framework takes around 2 h to train on a new region, and only several minutes for prediction.**

***Index Terms*— Deep neural networks, framework, post-event TerraSAR-X imagery, rapid, regional tsunami damage recognition.**

I. Introduction

NATURAL disasters, especially mega-tsunamis, are rapid and disastrous events that pose great threat to people's life and properties [1]. To support government's decision-making for postdisaster relief efforts, near real-time information of the building damage in affected areas is crucial [2]. Satellite remote sensing, especially active sensors such as the synthetic aperture radar (SAR), is a useful tool for building damage estimation because of its rapid and large-scale earth observation performance [3]. In earlier studies, a series of multitemporal SAR imagery-based change detection techniques was proposed for tsunami damage assessment [4], [5]. However, these methods are greatly limited when the predisaster SAR image is not available.

To generate a damage estimation method soon after a natural disaster with less dependence on preevent remote sensing data, there is an ongoing interest in developing building damage estimation techniques based on post-event SAR imagery. One approach aims to identify physical polarimetric SAR features [6], [7]. The advantage of this method is that it analyzes the damage from the essence of remote sensing by exploring the physical model of microwave scattering. However, the unavailability of fully polarimetric SAR data in real-word applications makes this method less practical. Another approach is based on a statistical learning method, where a series of texture features and polarimetric SAR features is employed to estimate building damage under the framework of machine learning [8], [9]. This method does achieve high accuracy; however, it requires manual and time-consuming extraction and selection of high-dimensional features, which limits the applicability of this method to meet the needs of a rapid disaster emergency response. In another method [10], bright curvilinear features derived from the geometry of man-made structures in SAR images are employed to detect building damage. These carefully outlined visual features are found to improve the accuracy of building damage recognition. This finding inspires us to explore the value of visual pattern information of SAR imagery for achieving high-precision damage recognition.

Considering the limitations of traditional methods, it is beneficial to develop a framework that not only enhances the speed and level of automation but also improves the efficiency of damage recognition. Deep learning has the potential to solve this problem because of its high ability of automatic feature learning and visual pattern recognition [11]. In addition, deep learning techniques have recently demonstrated a great potential in many different SAR imagery recognition tasks [12], [13]. With these inspirations in mind, this letter introduces a new framework of tsunami damage recognition. This framework is original as it is independent of preevent SAR and provides an automatic way to extract built-up (BU) area. Most importantly, this framework introduces a novel deep learning algorithm that achieves high accuracy and efficiency.

II. Data and Study Area

This letter focuses on the Pacific coast of the Tohoku region, Japan, which was severely damaged by the 2011 Tohoku earthquake tsunami, as shown in Fig. 1(a).

Manuscript received June 1, 2017; revised August 19, 2017 and September 18, 2017; accepted October 23, 2017. Date of publication December 4, 2017; date of current version December 27, 2017. This work was supported in part by JST CREST, Japan, under Grant JPMJCR1411 and in part by the China Scholarship Council. *(Corresponding author: Yanbing Bai.)*

Y. Bai is with the Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan (e-mail: ybbaipku@gmail.com).

C. Gao is with the Department of Computer Science, The University of Hong Kong, Hong Kong.

S. Singh is with the Department of Computer Science, University of California at Irvine, Irvine, CA 92697-3435 USA.

M. Koch is with the Center for Remote Sensing, Boston, MA 02215 USA.

B. Adriano, E. Mas, and S. Koshimura are with the International Research Institute of Disaster Science, Tohoku University, Sendai 980-0845, Japan.

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

Digital Object Identifier 10.1109/LGRS.2017.2772349

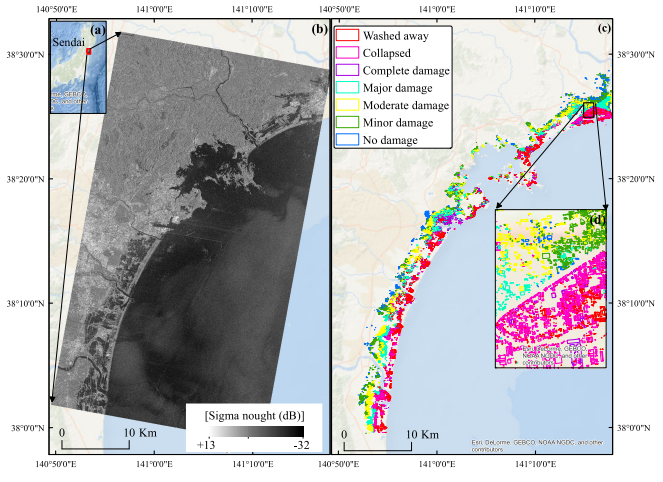

Fig. 1. Study area and data set used in this letter. (a) Location of the Tohoku region in Japan. (b) Post-event TerraSAR-X data from Tohoku region in Japan. (c) GTD of tsunami affected buildings in the Tohoku region in Japan. (d) Zoomed-in affected area.

The TerraSAR-X data used in this letter were acquired on March 12, 2011 (UTC) covering the Pacific coast of the Miyagi prefecture, Japan. The data were acquired in the StripMap mode with HH polarization in a right looking descending path. The center incident angle was 37.3°. The data were an enhanced ellipsoid-corrected product resampled into a 1.25-m square pixel size. The ground truth data (GTD) were provided by the Ministry of Land Infrastructure, Transport and Tourism (MLIT) [1] in a building footprint shape format with seven damage categories ("no damage," "minor damage," "moderate damage," "major damage," "complete damage," "collapsed," and "washed away"), as shown in Fig. 1(c). A zoomed-in affected area of the GTD in Ishinomaki city is shown in Fig. 1(d).

III. Damage Recognition Framework

We propose a framework based on deep convolutional neural networks to achieve rapid building damage recognition from SAR imagery. The framework is shown in Fig. 2. It consists of three major procedures: data preprocessing, BU region selection, and regional damage-level recognition.

A. Data Preprocessing

To generate proper and labeled image data for training and testing, a series of steps for data preprocessing is required. Our data preprocessing includes four steps: processing of SAR imagery, tile-based image split analysis [see Fig. 2(a)], label generation for image data [see Fig. 2(b)], and train-test data split.

1) Processing of SAR Imagery: The raw TerraSAR-X data is first transformed from digital numbers to sigma nought (dB). Then, each image is processed using the Lee filter with a kernel size of 3 × 3 pixels to reduce the image noise. Next, each image is subsampled into a pixel size of 1.25 m followed by a normalization of pixel values into a range of 0–255.

2) Tile-Based Image Split Analysis: First, a tile-based image split analysis technique is employed to divide TerraSAR scene into quadratic subimages with the predefined tiles

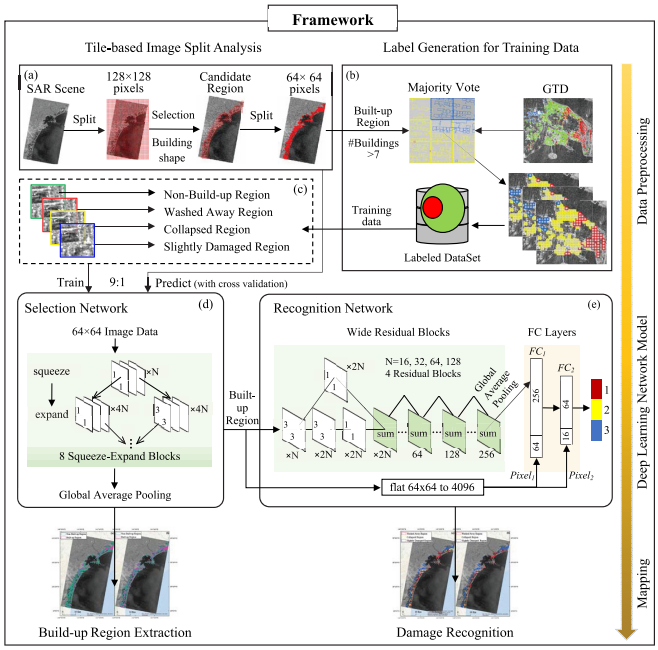

Fig. 2. Framework of the study. Descriptions in Section III.

of 128 × 128 pixels. Then, the ancillary data (i.e., building footprint vector layer) containing the position and spatial distribution of the affected buildings are used for tiles selection. In this step, only the tiles that intersect or contain at least one building are kept. Those areas are then split into subsamples with a tile size of 64 × 64 pixels. A larger quadratic size would result in too many buildings in one tile, and a smaller quadratic size would increase the number of buildings split into multiple tiles.

3) Label Generation for Image Data: A tile with only limited number of buildings is not sufficient to be used as a suitable BU sample, and the feature maps to determine BU and nonbuilt-up (NBU) regions are different from those of building damage patterns. Hence, we first select BU regions with one neural network, and then conduct damage-level mapping only on those regions with another neural network.

Considering the average building area size and coverage ratio, to select tiles with adequate building information and coverage, in this letter, tiles containing and/or intersecting no less than seven buildings are defined as BU regions, otherwise as NBU regions. For the BU regions, the GTD is used to generate a sample database with different damage levels. The MLIT building damage data are reclassified into three classes: "washed away building," "collapsed building," or "slightly damaged building" based on the similarity of damage degree [see Fig. 2(c)]. Tiles where the majority of building are "washed away" are labeled as "washed away regions," where the majority of buildings are "collapsed" are labeled as "collapsed region," and others as "slightly damaged regions." Such a majority criterion is regarded as a logical solution focusing more on the damage information that counts as the largest proportion within each tile.

4) Train-Test Data Split: For training and testing purposes, we adopt a tenfold cross validation [14], [15]: split 90% of data as training set and 10% as testing (validation) set, and create

TABLE I
STRUCTURE OF OUR SQUEEZENET

group	size change	parameter (w×h×channel×stride)
conv1	$64^2 \rightarrow 32^2$	3×3×16×2 conv, 3×3×16×1 pool
conv2	$32^2 \rightarrow 16^2$	two B^1(8,16,16), 3×3×32×2 pool
conv3	16^2	two B(16,32,32)
conv4	$16^2 \rightarrow 8^2$	two B(24,48,48), 3×3×96×2 pool
conv5	8^2	two B(32,64,64), dropout 0.5
conv6	$8^2 \rightarrow 2$	1×1×2×1 conv, global average

[1] $B(a, b, c)$ denotes a squeeze-expand block as defined in the original paper [18] with a squeeze channel, b expand channel of 1×1 convolution and c expand channel of 3×3 convolution.

TABLE II
STRUCTURE OF OUR WRN

group	size change	layer blocks
conv1	64^2	3×3×16×1 conv
res1	$64^2 \rightarrow 32^2$	$B^1(2, 32)$
res2	32^2	$B(1, 64)$
res3	$32^2 \rightarrow 16^2$	$B(2, 128)$
res4	$16^2 \rightarrow 8^2$	$B(2, 256)$
pool5	$8^2 \rightarrow 256$	*Global Average*[2]
fc6	$256 + 64 \rightarrow 64$	$FC^3(64, 256)$
fc7	$64 + 16 \rightarrow 3$	$FC(16, 64)$

[1] $B(a, b)$ denotes a residual block: a $3 \times 3 \times a \times b$ convolutional layer a_1 followed by a $3 \times 3 \times 1 \times b$ convolutional layer a_2; a $1 \times 1 \times a \times b$ convolutional layer b; and an element-wise sum layer of a_2 and b. A batch normalization [23] layer and a rectified linear unit (relu) layer are set before a_1. A batch normalization layer, a relu layer, and a dropout (ratio = 0.3) layer are set between a_1 and a_2.

[2] *Global Average* denotes a group of batch normalization layer, a relu layer, and a global average pooling layer.

[3] $FC(a, b)$ denotes a fully-connected block: a $Pixel$ layer with a neurons from the pixel-wise input image is concatenated with a FC layer with b neurons connected from the previous layer; a batch normalization layer and a dropout = 0.3 layer is added after them.

TABLE III
TEST ACCURACY OF DIFFERENT SELECTION NETWORK STRUCTURES

Network	Configuration	Accuracy
SqueezeNet	8 squeeze-expand blocks	80.4%
WRN-16-4	16 layers, 4 times channels	**81.9**%
Resnet-50	50 convolutional layers	74.3%
AlexNet	5 convolutional layers	75.1%

tenfold of such a selection to make ten nonoverlapping testing sets covering all data. Each fold is to be trained independently. Using subsets of the whole database as validation sets, cross validation serves as an effective method to relieve overfitting problems and provide an insight on how the trained model will generalize to other independent data sets. To alleviate the problem of insufficient training samples and sample imbalance, we adopted data augmentation techniques [16], [17] of mirroring (upside-down and left-to-right) and rotating (90°, 180°, and 270°) images. Such methods are able to balance classes with insufficient samples to the same size as others, and enlarge the whole data set by applying data augmentation to all images.

B. Built-Up Region Selection

We adopt SqueezeNet [18] as our selection network [see Fig. 2(d)] to rapidly extract the BU regions. Two main reasons account for our choice. First, it has far fewer parameters and thus could be trained much faster than other popular networks, even 50 times fewer parameters than AlexNet [19]. Second, since SAR images generally contain less information than high-resolution optical images due to spatial resolution limitations, improving the complexity of deep neural networks will not significantly improve the prediction result, and may even worsen it. Table I shows the structure of our SqueezeNet.

Our SqueezeNet is characterized by the eight squeeze-expand blocks and a global average polling layer [20]. These squeeze-expand blocks greatly reduce parameters in the convolution structure. After the eight blocks, instead of using fully connected layers, we use a global average polling layer to combine feature responses from the convolutional layers, which further reduces the parameters and helps improve prediction accuracy. The output of the global average pooling layer is a binary number indicating whether the input image should be dropped before recognition or not.

C. Regional Damage-Level Recognition

We adopt a modified version of wide residual network (WRN) [21] for recognition, as shown in Fig. 2(e). The WRN is different from the original residual network [22] (Resnet) in that it has wider convolutional channels and fewer convolutional layers, yet provides better overall prediction accuracy. Our model has nine convolutional layers with four times width of convolution channels compared with Resnet. Moreover, we introduce extra pixel-level image encoding layers, $Pixel_1$ and $Pixel_2$ in our model. They are fully connected layers that directly encode pixelwise value from the input image. After the residual blocks and a global average pooling layer, we add two additional fully connected layers FC_1 and FC_2 to the network. Then, we concatenate FC_1 with $Pixel_1$ and FC_2 with $Pixel_2$. This helps extract pixel-level information. Table II shows our model (WRN-9-4 with pixel-level encoding).

The argmax output of the final fully connected layer represents the damage level of the corresponding input region.

IV. RESULT AND DISCUSSION

A. Network Structures for Built-Up Region Selection

Table III shows the comparison of test accuracy among different network structures for BU region selection with tenfold cross validation. We use mirroring and rotation to balance classes with insufficient samples.

We observe that SqueezeNet is a desirable balance between accuracy and speed, since other methods have similar results but far more parameters. For the training details, we set learning rate as 0.01 for the first 50 000 steps, 0.001 for the next 50 000 steps for SqueezeNet, and similar steps with other nets depending on their structure. We use a batch size of 32, a momentum of 0.9, and a weight decay of 0.0005. It generally takes around 1 h to train our SqueezeNet on a GTX TITAN X GPU under the Caffe framework [24]. We use mean shift, mirroring, and rotation for data augmentation.

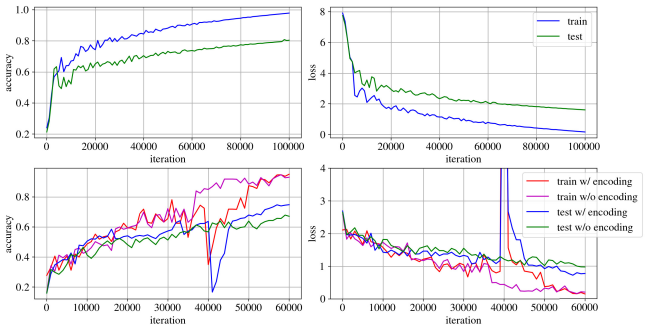

Fig. 3. Accuracy and loss curves. (a) Accuracy of our SqueezeNet model. (b) Loss of our SqueezeNet model. (c) Accuracy of WRN-9-4 models. (d) Loss of WRN-9-4 models. Training and testing of (c) and (d) were conducted independently. All curves are generated as mean values of cross validation.

TABLE IV
TEST ACCURACY OF DIFFERENT RECOGNITION NETWORK STRUCTURES

Network	Configuration	Encoding	Accuracy
WRN	9 layers, ×2 width	No	62.23%
		Yes	65.12%
	9 layers, ×4 width	No	67.12%
		Yes	**74.80**%
	16 layers, ×4 width	No	60.07%
		Yes	66.11%
Resnet	20 layers	No	59.93%
	50 layers	No	57.23%
AlexNet	-	No	60.77%

The second fastest model is AlexNet, which takes 2–5 h depending on channel sizes. Resnet and WRN take around 6 h. Fig. 3(a) and (b) shows the accuracy and loss curve of our model.

B. Network Structures for Damage-Level Recognition

Table IV shows the test accuracy among different network structures for regional damage-level recognition with tenfold cross validation. We modify the layer stride in some models to adjust input size to 64×64. We use mirroring and rotation to balance classes with insufficient samples.

For models with extra encoding layers, we first train the original model only, then add the additional fully connected encoding layers, and freeze the parameters of all convolutional layers. More specifically, for our model (WRN-9-4 with pixel-level encoding), we train 40 000 steps with learning rate as 0.001 for the original model and 20 000 steps with learning rate as 0.0001 for encoding layers. We adopt the same hyperparameters as BU region selection. It takes around 1 h to train this model on the same platform and GPU. Due to overfitting, some complex models, such as Resnet, could not even achieve a result over 60%. Since our model compresses the residual layers and adds encoding with high dropout to help extract features while maintaining sparsity, the overfitting problem is significantly reduced. As shown in Fig. 3(c) and (d), our model achieves a relatively high testing accuracy, and the encoding layers helps reduce the total loss by around 20%, which proves their ability to relieve overfitting problems.

TABLE V
ASSESSMENT OF BU REGION SELECTION

Result of our modified SqueezeNet					
		Prediction			
		BU	NBU	Total	P.A.(%)
GTD	BU	3948	743	4691	84.2
	NBU	2720	10309	13029	79.1
	Total	6668	11052	κ = 0.56	
	U.A.(%)	59.2	93.3	Overall = 80.5%	

κ, P.A., U.A., and Overall denote kappa coefficient, producer's accuracy, user's accuracy, and overall accuracy respectively.

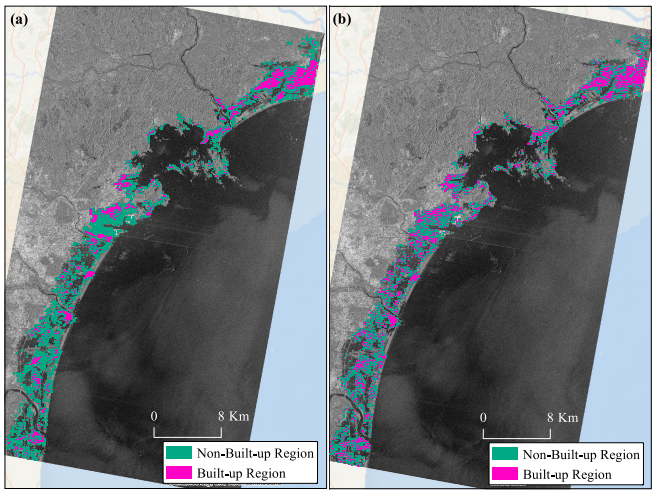

Fig. 4. Comparison of BU region. (a) Reference map of BU regions. (b) Predicted BU regions.

C. Accuracy Assessment of Built-Up Region Selection

Table V and Fig. 4 demonstrate the BU region selection result with our modified SqueezeNet. Our model has an overall accuracy of 80.5% and a kappa coefficient of 0.56. From the viewpoint of disaster response, our results are ideal because of the high producer accuracy, which could help the responser better grasp the disaster situation. Moreover, a bidirectional check was conducted to grasp the reason for erroneous selections. We found that most of the misclassified NBU regions are characterized by complex scatterers that look similar to buildings, such as forests and paddy field, and that most of the misclassified BU regions are intersected by roads and rivers, which hinder the recognition.

D. Accuracy Assessment of Regional Damage Mapping

The regional damage mapping results with our modified WRN are described in Table VI and displayed in Fig. 5 with comparison to the reference data generated from the GTD. Our method has an overall accuracy of 74.8% and a kappa coefficient of 0.60. It can be observed that the bias problem that may result from insufficient data is controlled by using rotated and mirrored images, and P.A. result of the three classes only varies less than 2

E. Time Assessment and Reproduction

As described in Section IV, the training of our modified SqueezeNet and WRN takes around 1 h on a GTX TITAN X GPU. Thus, if we aim to train on a completely new

TABLE VI
ASSESSMENT OF REGIONAL DAMAGE MAPPING

Result of recognition algorithm with our WRN						
		Prediction				
		WAR	CR	SDR	Total	P.A.(%)
GTD	WAR	1016	168	164	1348	75.4
	CR	106	560	94	760	73.7
	SDR	339	310	1934	2583	74.9
	Total	1461	1038	2192	κ = 0.60	
	U.A.(%)	69.5	53.9	88.2	Overall = 74.8%	

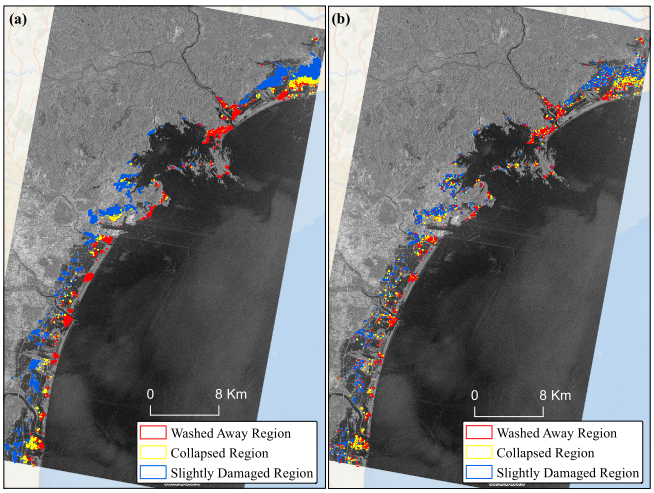

Fig. 5. Comparison of regional damage mapping. (a) Damage mapping result generated from the GTD. (b) Damage mapping result generated from the recognition model.

region, our framework can be reproduced in around 1 h with concurrent training on two GPUs, or 2 h on a single GPU. If we aim to test over a new region, given that it has a similar geometric outlook as a pretrained model, it takes less than 2 min to finish the whole prediction process if the data set size is similar to ours. This validates the speed of our framework for both training and prediction.

V. CONCLUSION

In this letter, we provided a practical and rapid solution to the problem of tsunami damage mapping at a regional scale. We introduced a deep learning-based framework for SAR data preprocessing, rapid BU region extraction, and automatic building damage mapping. We combined popular structures of deep neural networks, with special designs to extract most important features and reduce computational time requirements. Experiments on the 2011 Tohoku earthquake and tsunami area validate that our framework is operational and fast in training and prediction calculations.

ACKNOWLEDGMENT

The authors would like to thank the Pasco Corporation for providing the TerraSAR-X data.

REFERENCES

[1] MLIT. (2014). *Survey of Tsunami Damage Condition*. Accessed: Sep. 2017. [Online]. Available: http://www.mlit.go.jp/toshi/toshi-hukkou-arkaibu.html

[2] L. Ge, A. H.-M. Ng, X. Li, Y. Liu, Z. Du, and Q. Liu, "Near real-time satellite mapping of the 2015 Gorkha earthquake, Nepal," *Ann. GIS*, vol. 21, no. 3, pp. 175–190, 2015.

[3] A. Suppasri, D. Kamthonkiat, H. Gokon, M. Matsuoka, and S. Koshimura, *Application of Remote Sensing for Tsunami Disaster*. Rijeka, Croatia: InTech, 2012.

[4] S.-W. Chen and M. Sato, "Tsunami damage investigation of built-up areas using multitemporal spaceborne full polarimetric SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 1985–1997, Apr. 2013.

[5] H. Gokon *et al.*, "A method for detecting buildings destroyed by the 2011 Tohoku earthquake and tsunami using multitemporal TerraSAR-X data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 6, pp. 1277–1281, Jun. 2015.

[6] W. Zhai and C. Huang, "Fast building damage mapping using a single post-earthquake PolSAR image: A case study of the 2010 Yushu earthquake," *Earth, Planets Space*, vol. 68, no. 1, p. 86, Dec. 2016.

[7] X. Li, H. Guo, L. Zhang, X. Chen, and L. Liang, "A new approach to collapsed building extraction using RADARSAT-2 polarimetric SAR imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 4, pp. 677–681, Jul. 2012.

[8] L. Shi, W. Sun, J. Yang, P. Li, and L. Lu, "Building collapse assessment by the use of postearthquake Chinese VHR airborne SAR," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2021–2025, Oct. 2015.

[9] Y. Bai, B. Adriano, E. Mas, H. Gokon, and S. Koshimura, "Object-based building damage assessment methodology using only post event ALOS-2/PALSAR-2 dual polarimetric SAR intensity images," *J. Disaster Res.*, vol. 12, no. 2, pp. 259–271, 2017.

[10] P. T. B. Brett and R. Guida, "Earthquake damage detection in urban areas using curvilinear features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4877–4884, Sep. 2013.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[12] L. Wang, K. A. Scott, L. Xu, and D. A. Clausi, "Sea ice concentration estimation during melt from dual-Pol SAR scenes using deep convolutional neural networks: A case study," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4524–4533, Aug. 2016.

[13] C. Bentes, A. Frost, D. Velotto, and B. Tings, "Ship-iceberg discrimination with convolutional neural networks in high resolution SAR images," in *Proc. 11th Eur. Conf. Synth. Aperture Radar, (EUSAR)*, Jun. 2016, pp. 1–4.

[14] R. C. Sharma, K. Hara, and H. Hirayama, "A machine learning and cross-validation approach for the discrimination of vegetation physiognomic types using satellite based multispectral and multitemporal data," *Scientifica*, vol. 2017, 2017, Art. no. 9806479.

[15] G. Seni and J. F. Elder, "Ensemble methods in data mining: Improving accuracy through combining predictions," *Synthesis Lect. Data Mining Knowl. Discovery*, vol. 2, no. 1, pp. 1–126, 2010.

[16] T. Kooi *et al.*, "Large scale deep learning for computer aided detection of mammographic lesions," *Med. Image Anal.*, vol. 35, pp. 303–312, Jan. 2017.

[17] B. Leng, K. Yu, and J. Qin, "Data augmentation for unbalanced face recognition training sets," *Neurocomputing*, vol. 235, pp. 10–14, Apr. 2017.

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. (Feb. 2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size." [Online]. Available: https://arxiv.org/abs/1602.07360

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[20] M. Lin, Q. Chen, and S. Yan. (Dec. 2013). "Network in network." [Online]. Available: https://arxiv.org/abs/1312.4400

[21] S. Zagoruyko and N. Komodakis. (May 2016). "Wide residual networks." [Online]. Available: https://arxiv.org/abs/1605.07146

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[24] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.