

## Research Article

# Object Detection in Ground-Penetrating Radar Images Using a Deep Convolutional Neural Network and Image Set Preparation by Migration

Kazuya Ishitsuka <sup>1</sup>, Shinichiro Iso <sup>2,3</sup>, Kyosuke Onishi <sup>4</sup>, and Toshifumi Matsuoka <sup>3</sup>

<sup>1</sup>Division of Sustainable Resources Engineering, Hokkaido University, Sapporo 065-0068, Japan

<sup>2</sup>School of Creative Science and Engineering, Waseda University, Tokyo 169-8555, Japan

<sup>3</sup>Fukada Geological Institute, Tokyo 113-0021, Japan

<sup>4</sup>Geology and Geotechnical Engineering Research Group, Public Works Research Institute, Tsukuba 305-8516, Japan

Correspondence should be addressed to Kazuya Ishitsuka; [ishitsuka@eng.hokudai.ac.jp](mailto:ishitsuka@eng.hokudai.ac.jp)

Received 25 June 2018; Accepted 8 November 2018; Published 27 November 2018

Academic Editor: Yun-tai Chen

Copyright © 2018 Kazuya Ishitsuka et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ground-penetrating radar allows the acquisition of many images for investigation of the pavement interior and shallow geological structures. Accordingly, an efficient methodology of detecting objects, such as pipes, reinforcing steel bars, and internal voids, in ground-penetrating radar images is an emerging technology. In this paper, we propose using a deep convolutional neural network to detect characteristic hyperbolic signatures from embedded objects. As a first step, we developed a migration-based method to collect many training data and created 53510 categorized images. We then examined the accuracy of the deep convolutional neural network in detecting the signatures. The accuracy of the classification was 0.945 (94.5%)–0.979 (97.9%) when using several thousands of training images and was much better than the accuracy of the conventional neural network approach. Our results demonstrate the effectiveness of the deep convolutional neural network in detecting characteristic events in ground-penetrating radar images.

## 1. Introduction

Ground-penetrating radar (GPR) is an effective technology for the nondestructive investigation of the shallow subsurface based on the transmitting and receiving of an electromagnetic wave. The radar system transmits a short pulse toward the target and records the signal that is backscattered by discontinuities in the dielectric permittivity. Because the dielectric permittivity relates to embedded subsurface features, GPR systems have been widely used for road inspection, mine detection, and geological/archaeological studies. Recent GPR systems with air-coupled antennas have been mounted onto vehicles for recording signals in a broader and faster way (e.g., [1, 2]). However, with the increasing data amount, the detection of target signatures becomes a challenging and time-consuming task.

For the automatic detection of characteristic features in GPR images, a neural network (NN)-based methodology

was developed [3, 4]. Frequency features of GPR images were used for detection using an NN [3] while spatial signatures of GPR images were used after binarization using a threshold [4]. The successful application of an NN has also been demonstrated (e.g., [5]). Despite these successful applications, the NN-based method is often unable to recognize complex features. Moreover, accurate recognition requires an ideal noise condition and appropriate preprocessing (e.g., clutter removal), whose assumptions sometimes fail.

Recently, the deep convolutional neural network (CNN) has been successfully used in pattern recognition (e.g., [6, 7]) and classification (e.g., [8, 9]). Compared with the conventional NN, the architecture of the deep CNN consists of several alternations of convolution and pooling layers, which is an analog of the receptive field. The architecture has a great advantage in terms of representing images with multiple levels of abstraction. The deep CNN improves the accuracy of

the target detection of GPR images and the versatility of the approach.

The present paper proposes the use of the deep CNN to increase the accuracy and versatility of target detection from GPR images. For versatility, we used the original pattern of GPR images as input images without conducting a prior feature extraction. Many training images are needed for the successful training of the deep CNN and improved accuracy. We therefore developed an algorithm based on the migration procedure for the first-step extraction of training images with target signatures.

## 2. Deep CNN

We first introduce an algorithm to prepare many training images from a two-dimensional section of GPR images in Section 2.1. We then describe the architecture and roles of each component of the deep CNN in Section 2.2. Sections 2.3 and 2.4 give the specifics of CNN architecture and a description of the accuracy index used in this study.

*2.1. Semi-automatic Collection of Training Images.* Normalization is a necessary preprocessing step for standardization in collecting training images and conducting CNN learning. This study applied local contrast normalization to input images. Resulting images thus have zero mean and a standard deviation of 1.

To extract hyperbolic signatures originating from embedded objects, we propose the following algorithm based on the difference between the envelope of the original signal (without migration) and the envelope of the signal after migration procedure. A signal reflected from an object has a characteristic hyperbolic curve in the GPR data section, because the distance between radar and target decreases as the radar approaches the target and the distance increases as the radar moves away from the target (Figure 1(a)). The migration procedure focuses the hyperbolic signal and reconstructs the true shape of an object at the true location. The shape of the signal therefore varies with migration.

The processing conducted to extract reflection events in this study is (1) frequency–wavenumber migration, (2) application of the Hilbert transform to signals with and without migration, (3) calculation of the envelope of signals with and without migration (Figures 1(b) and 1(c)), (4) calculation of the difference between the two envelopes, (5) extraction of changed areas by binarization of the difference map (Figure 1(d)), and (6) division of the original data into small batches and classification of them into two classes. The first class contains hyperbolic signatures (event images) while the second class does not contain these characteristics (nonevent images). The classification involves checking if the corresponding area of the binarized maps contains reflection events.

As a migration algorithm, we adopted a frequency–wavenumber migration algorithm [10] because it is an efficient algorithm with which to process a large amount of data. The migration algorithm calculates migrated signals through Fourier transformation and Stolt interpolation [11]. The envelope of the signal  $A(t)$  is defined by the magnitude

of the complex signal via the Hilbert transformation  $A(t) = \sqrt{f^2(t) + g^2(t)}$ , where the complex signal can be defined as  $A(t) \exp\{i \cdot \theta(t)\} = f(t) + i \cdot g(t)$  [12]. After the above processing, images classified into the wrong class were removed manually, because the above algorithm sometimes misclassified noisy images as images with the characteristic reflection events. Although fully automatic image set preparation is preferable, there is currently no perfect algorithm for GPR image classification. Therefore, the manual check is necessary as a final part of the image set preparation.

*2.2. Architecture of the Deep CNN.* The deep CNN is a feedforward NN that comprises a stack of alternating convolution layers and pooling layers and then fully connected layers and softmax layers [13] (Figure 2). Each convolutional layer receives inputs from a set of units located in a small neighborhood of the previous layer, referred to as the local receptive field. In each local receptive field, the dot product of the weights and input is calculated, and a bias term then added (Eq. (1)). This operation is called filtering in the convolution layer. The filter then moves along the input vertically and horizontally, repeating the same computation for each local receptive field. The feature represented by the filtering can be emphasized through the layer.

$$u_{i,j,k} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} x_{i+p,j+q} w_{p,q,k} + b_k \quad (1)$$

Here  $x_{i,j}$  and  $u_{i,j}$  denote  $i$ th and  $j$ th pixels of input and output images.  $w_{p,q,k}$  indicates the  $k$ th filter having a rectangular size of  $M \times N$ , which moves along the input space to convolute with the input image  $x_{p,q}$ .  $p$  and  $q$  denote a pixel within the rectangular area of the filter.  $b_k$  indicates the bias term corresponding to the  $k$ th filter. The rectified linear units (ReLU) activation function is applied after the convolutional layer (Eq. (2)).

$$y_{i,j,k} = \max\{0, u_{i,j,k}\} \quad (2)$$

Here  $y_{i,j,k}$  is pixel  $(i,j)$  of the output corresponding to the  $k$ th filter. The advantage of the ReLU function is its nonsaturating nonlinearity. In other words, the gradient of the ReLU function is not zero even for an extreme input value. It was also found that the computation is much faster than the computation of the conventional sigmoid function [14].

Each convolution layer possessing the ReLU function is followed by a max pooling layer having a certain pooling size. A max pooling layer extracts the maximum value within the pooling window from the output feature map generated by the previous convolutional layer (Eq. (3)). The pooling operation reduces arising information redundancy because the convolution windows of neighboring locations overlap and enhances the universality of slight position changes of an object.

$$z_{i,j,k} = \max_{(p,q) \in Q} \{y_{p,q,k}\} \quad (3)$$

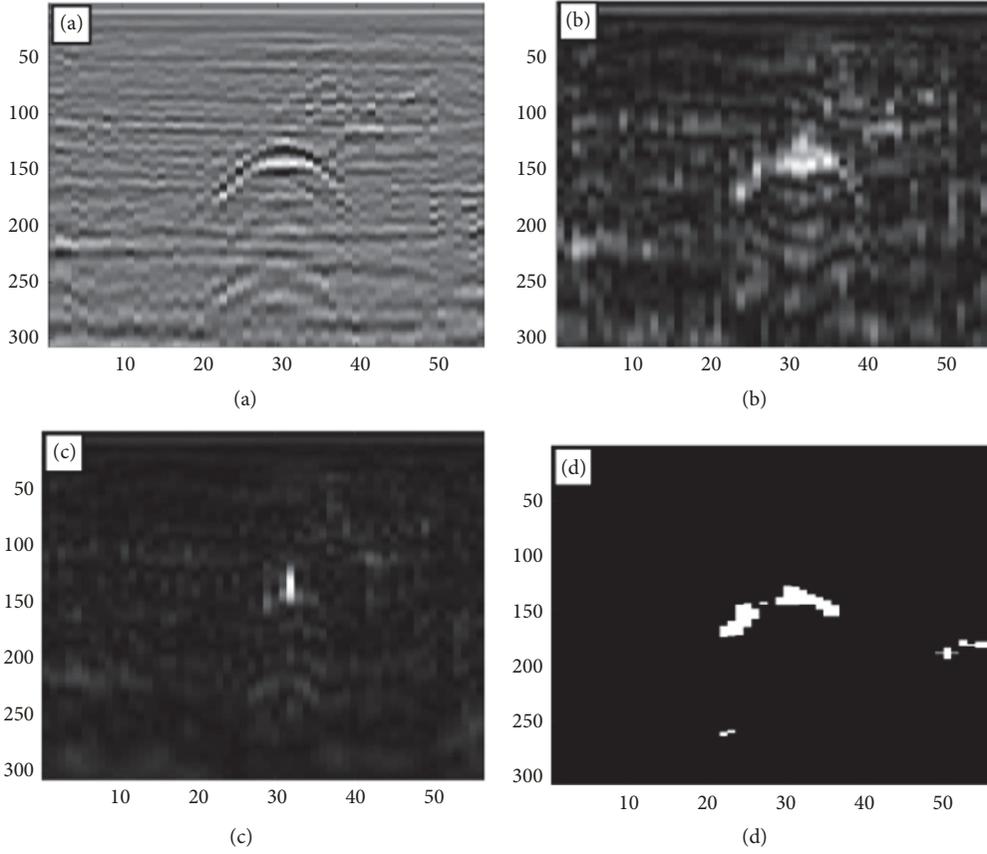


FIGURE 1: (a) Part of an original GPR image that contains the characteristic reflection event from a buried object. (b) Envelope of the GPR image of (a). (c) Envelope of the GPR image after migration. (d) Binarized image of the difference in the envelope without and with the migration procedure.

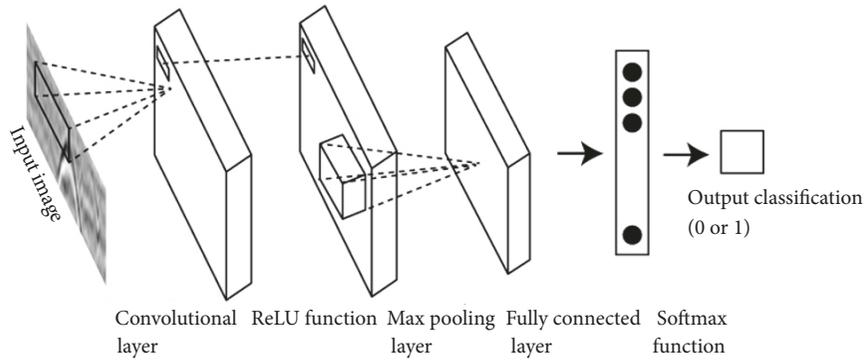


FIGURE 2: Conceptual diagram of image classification using the deep CNN.

Here  $Q$  is the pooling size and  $z_{i,j,k}$  is pixel  $(i, j)$  of the output of the pooling layer corresponding to the  $k$ th filter. A larger pooling size generally results in worse performance because the operation throws away too much information.

Fully connected layers correspond to a standard multi-layer NN. The layer consists of a stack of several perceptrons that represents a nonlinear relationship between the weighted sums of inputs and outputs. Each perceptron has the weights and biases as unknown parameters, which are optimized in the learning process.

The softmax function is applied to the output layer. The softmax function calculates posterior probabilities ( $p_l$ ) over each class (Eq. (4)). In this study, a class is defined as whether the input figure contains the target signatures or not.

$$p_l = \frac{\exp(a_l)}{\sum_{k=1}^K \exp(a_k)} \quad (4)$$

Here  $a_l$  indicates the weighted sum of inputs to the  $l_{th}$  unit in the output layer.

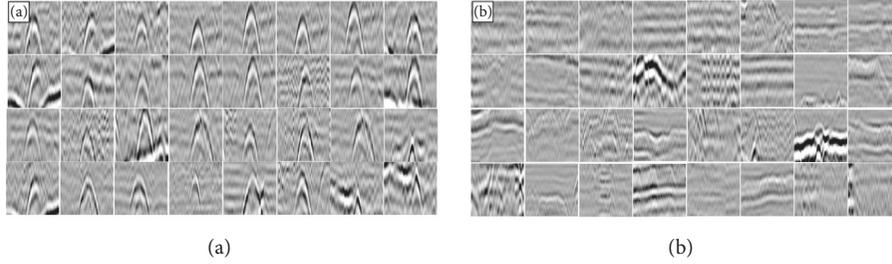


FIGURE 3: Examples of extracted GPR images with characteristic reflection events: (a) training images categorized into reflection events, and (b) training images categorized into nonreflection events.

The CNN contains unknown parameters: the filtering coefficients and biases in convolution layers and the weights and biases in fully connected layers. The learning process optimizes these parameters via a backpropagation algorithm. The backpropagation algorithm iteratively estimates parameters using the gradient of an objective function. In the case of classification, cross entropy is usually adopted as the objective function (Eq. (5)).

$$C = -\sum_{l=1}^n d_l \log_e p_l \quad (5)$$

Here  $d_l$  is 1 if the classified result is correct and zero if not.

**2.3. Configuration.** The input layer has dimensions of  $65 \times 65$  corresponding to the size of the extracted original image (e.g., Figure 3). Figure 3 shows typical images of two classes (i.e., reflection event/no reflection event). For the convolutional layer, we used 20 convolution filters having a window size of  $5 \times 5$  as a default setting, resulting in 20 feature maps with a size of  $61 \times 61$ . In this case, the numbers of weights and biases are respectively  $5 \times 5 \times 20$  and 20. The accuracy of the CNN depends on the window size; this is evaluated in Section 4. No spatial zero padding is used in the convolution filters, and the convolution stride is fixed to 1 pixel. A max pooling layer with a pooling size of  $2 \times 2$  has a stride of 2 pixels. The size of the resulting 20 feature maps is  $30 \times 30$ . In this study, we used a simple configuration: one convolution layer and one max pooling layer. This is because the accuracy did not significantly increase in our data when the number of these layer increases. A fully connected layer is specified as having an output size of 2. Therefore, the numbers of weights and biases are respectively  $30 \times 30 \times 2$  and 10. The softmax function divides an input image into two classes; i.e., it categorizes on the basis of whether an input image contains target signatures (reflection events) or not.

**2.4. Accuracy Evaluation of the Deep CNN.** We quantified the accuracy of the deep CNN using the total accuracy ( $Ac$ ) as follows:

$$Ac = \frac{c_{est}}{c_{val}} \quad (6)$$

Here  $c_{est}$  is the number of correctly categorized images in the validation data and  $c_{val}$  is the number of validation images.

The accuracy is ranged from 0 to 1, and a higher value of the index indicates higher accuracy of overall classification. In addition, we calculated the true positive ratio ( $TPR$ ) and the true negative ratio ( $TNR$ ) to investigate how the deep CNN can detect reflection events correctly (Eq. (7) and (8)).

$$TPR = \frac{c_{est,e}}{c_{val,e}} \quad (7)$$

$$TNR = \frac{c_{est,n}}{c_{val,n}} \quad (8)$$

Here  $c_{est,e}$  and  $c_{est,n}$  are the number of correctly categorized images as reflection events and nonreflection event in the validation data, while  $c_{val,e}$  and  $c_{val,n}$  are the number of validation images with reflection events and without reflection events. The  $TPR$  is the accuracy in detecting reflection images, and the  $TNR$  is the accuracy in detecting nonreflection images.

### 3. GPR Images

We used image data acquired by a vehicle-mounted GPR system. The system acquired GPR trace data with 1-cm spacing along a road, recording 305 samples during 29.79 ns in the depth direction. In this configuration, GPR data were acquired along four lines (namely 8009-1, 8009-2, 8009-3 and 8009-4). The corresponding numbers of traces were 548,700, 556,980, 552,580 and 569,940.

To prepare training data, we first extracted square areas within a window size of  $65 \times 65$ . The size corresponds to 65 cm in the horizontal direction and 6.35 ns in the vertical direction. Then, according to the method described in Section 2.1, the square areas were automatically classified according to whether there were characteristic reflection events. Images with noisy areas were sometimes also classified as images having a reflection event, and we checked the horizontal and vertical continuity of the difference in envelopes with and without migration and discarded the misclassified images that did not show spatial continuity. Subsequently, the extracted areas were manually checked, and the images of reflection were reclassified from target objects and images without such events (e.g., inclined geological boundaries). About 39% of images were reclassified. Examples of classified images with characteristic reflection events are shown in Figure 3. The number of extracted images was 21,879 (1875 for reflection events / 20,004 for nonreflection events) on line

TABLE 1: Accuracy of the cross-validation test using the deep CNN. One line was used for validation images while other lines were used for training images.

The line for validation images	Accuracy	True Positive Ratio (TPR)	True Negative Ratio (TNR)
8009-1	0.945	0.794	0.959
8009-2	0.978	0.719	0.989
8009-3	0.958	0.702	0.991
8009-4	0.979	0.741	0.980

TABLE 2: Accuracy of the deep CNN depending on the window size of the convolutional layer.

The window size	Accuracy	True Positive Ratio (TPR)	True Negative Ratio (TNR)
3×3	0.928	0.789	0.941
5×5	0.945	0.794	0.959
11×11	0.945	0.828	0.956
15×15	0.949	0.799	0.963
20×20	0.947	0.801	0.961

8009-1, 8702 (501/8201) on line 8009-2, 13,293 (1532/11,761) on line 8009-3, and 9636 (263/9423) on line 8009-4.

#### 4. Results of Classification

For accuracy evaluation, we used one of the four lines as validation data and the other three lines as training data. In other words, we calculated four accuracies corresponding to the four lines as validation images. During the training of the deep CNN, we monitored the learning curve to avoid overfitting. Among the four obtained accuracies, the accuracy was a maximum (0.979) when 8009-4 was used as the validation images and a minimum (0.945) when 8009-1 was used as the validation images (Table 1). The TPR had a maximum of 0.794 and a minimum of 0.702, while the TNR had a maximum of 0.959 and a minimum of 0.991 (Table 1). The accuracy of reflection image detection (TPR) was inferior to that of nonreflection image detection (TNR), probably owing to the variation in reflection features and fewer training images.

Visualization of the output images of the convolutional and pooling layers clarified the characteristics of the deep CNN. Figures 4(a) and 4(d) shows one of the input images with and without reflection events. The optimal filtering of the convolutional layers produced the images shown in Figures 4(b) and 4(e). Figures 4(b) and 4(e) shows that the filtered images had emphasized features in a certain preferential direction. After application of ReLU functions and the max pooling layer, the characteristic parts were exhibited by emphasizing areas with larger positive amplitude (Figures 4(c) and 4(f)).

The window size may affect accuracy because the convolutional layer extracts the features of input data. We therefore examined accuracy depending on the architecture of the deep CNN. Specifically, we tested the deep CNN with the convolution layer having a window size of 3, 5, 11, 15, and 20. Lines 8009-2, 8009-3, and 8009-4 were used as training data while line 8009-1 was used as validation data. The result of the examination is presented in Table 2. The table shows that the accuracy for a window size of 15 was 0.949, which

was superior to the accuracy for other window sizes. The total accuracy decreased with the window size decreasing from 15 and also with the window size increasing above 15 (Table 2). The correlation of the total accuracy with the window size was due to the TNR value; i.e., TNR was a maximum when a window size of 15 was used, and the accuracy monotonically decreased with the window size decreasing from 15 (Table 2). Since the size of training images is 65×65, the window size of 15 is 23 % of the total image size. Meanwhile, there was no apparent correlation of TPR with the window size (Table 2).

To show the advantage of the deep CNN, we compared the results of the deep CNN with those of the conventional feedforward NN. The feedforward NN consists of layers of perceptrons, with perceptrons between layers being fully connected [15]. Because the performance of the NN depends on the number of layers and perceptrons, we examined the accuracy for several architectures: 10 perceptrons per layer (one hidden layer), 20 perceptrons per layer (one hidden layer), 5 perceptrons per layer (two hidden layers), and 10 perceptrons per layer (two hidden layers). For the NN with two hidden layers, we used the same number of perceptrons per layer for simplicity. In the examination, data on the 8009-1 line were used as validation data and the other data (of lines 8009-2, -3, and -4) were used as training data to optimize the neural network.

The results of the conventional NN approach are shown in Table 3. The accuracy of the overall classification was highest (0.707) for 10 perceptrons per layer and one hidden layer and lowest (0.513) for 10 perceptrons per layer and two hidden layers (Table 3). In the classification of data without a characteristic reflection event, TNR ranged 0.518–0.752 (Table 3). Meanwhile, the TPR ranged 0.237–0.466 (Table 3). The total accuracy of the deep CNN exceeded 0.9, showing that the deep CNN performed well. In addition, the results imply that the classification accuracy of the NN highly depends on the architecture of the NN in contrast with the case for the deep CNN.

It is widely known that the deep CNN requires many training images. We thus examined the accuracy depending on the number of training GPR images. As stated above, there

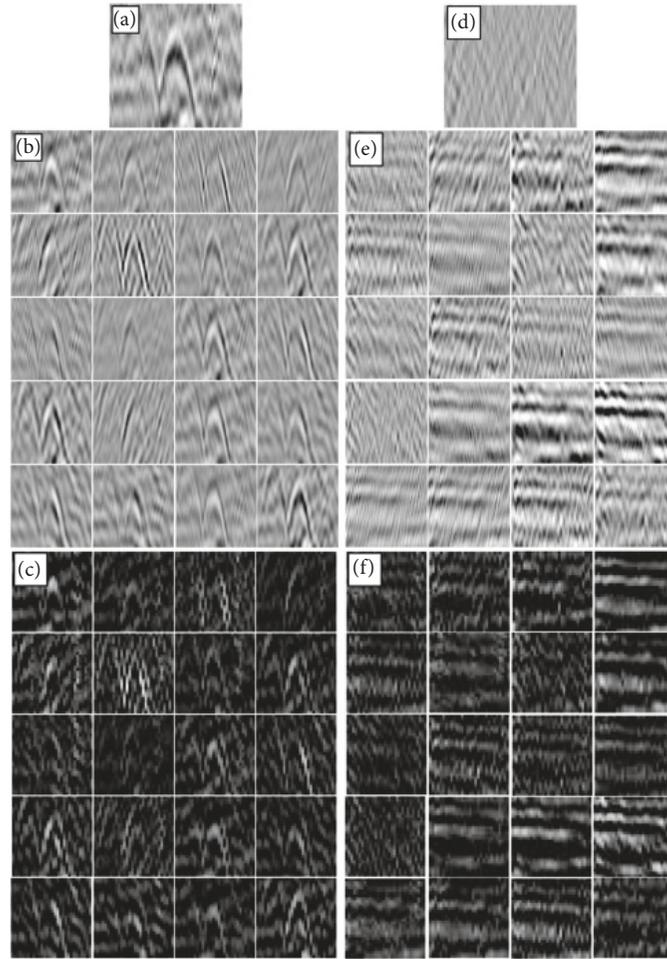


FIGURE 4: Visualization of images for the deep CNN: (a) and (d) inputs of reflection and nonreflection images, (b) and (e) examples of the images following the convolutional layer, and (c) and (f) examples of the images following the max pooling layer.

TABLE 3: Accuracy using the NN depending on the number of perceptrons and layers.

The number of perceptrons per a layer	Accuracy	True Positive Ratio (TPR)	True Negative Ratio (TNR)
10	0.707	0.237	0.752
20	0.594	0.393	0.613
5×5	0.589	0.387	0.609
10×10	0.513	0.466	0.518

were 2296 training images for reflection events and 29,385 for nonreflection events when lines 8009-2, 3 and 4 were used as training images. We then examined the accuracy when using 90% (2066 images for reflection and 26,447 for nonreflection), 70% (1607 and 20,570), 50% (1148 and 14,692), 30% (668 and 8816) and 10% (230 and 2939) of training data. The 8009-4 line was used as validation data. Table 4 shows that the total accuracy decreased with a decreasing number of training images: the accuracy was 0.945 when 90% and 70% of training data were used, and the values respectively decreased to 0.937 and 0.941 when using 30% and 10% of images (Table 4). An examination of the TPR and TNR shows that the decrease in the total accuracy is attributed to TPR.

In particular, the decrease in accuracy was notable when the number of images fell below about 1000 (30%–50%) (Table 4). Meanwhile, TNR was almost constant over the percentages of images tested in this study. Although the total accuracy was lower for smaller number of training images used, the total accuracy remained high (>0.9).

The visualization and examination of correctly and incorrectly classified images by the deep CNN further clarifies the characteristics of the deep CNN approach. The correctly classified images apparently contain or do not contain reflection events (Figures 5(a) and 5(c)). On the other hand, incorrectly classified images display an incomplete signature of reflection events, which is difficult to be categorized even by visual

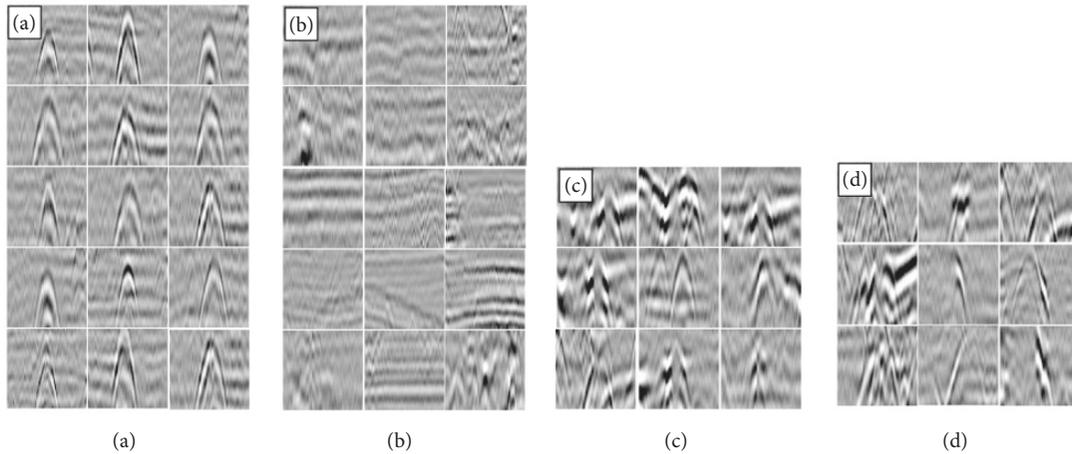


FIGURE 5: Classified images using the deep CNN: (a) images originally categorized into images with reflection events and correctly classified by the deep CNN, (b) images originally categorized into images without reflection events and correctly classified by the deep CNN, (c) images originally categorized into images with reflection events but recognized as nonreflection images by the deep CNN, and (d) images originally categorized into images without a reflection event but recognized as reflection images by the deep CNN.

TABLE 4: Accuracy depending on the number (the percentage) of training images in the deep CNN. The percentage is that of all training data.

The percentage	Accuracy	True Positive Ratio (TPR)	True Negative Ratio (TNR)
90 %	0.945	0.741	0.964
70 %	0.945	0.765	0.962
50 %	0.944	0.740	0.963
30 %	0.937	0.725	0.957
10 %	0.941	0.612	0.972

inspection (Figures 5(b) and 5(d)). The facts demonstrate that the algorithm has successfully classified the apparent reflection events or nonreflection event images. Further strategies for the recognition of the incomplete reflection signatures would increase the applicability of the deep CNN approach.

## 5. Conclusions

We examined the accuracy of the deep CNN in detecting a characteristic reflection pattern in GPR images. To prepare training images, we used the difference in envelopes obtained without and with a migration procedure. We found the classification accuracy of the deep CNN ranged 0.945–0.979. The accuracy was slightly improved by a few percentage points by tuning the window size of the convolutional layer. Comparison with the conventional NN showed the high accuracy of the deep CNN. Our results demonstrate that a large number of training data and an effective methodology improve the effectiveness of object detection in GPR images.

## Data Availability

GPR data used in this study were provided from Canaan Geo Research. For more detail, please contact the corresponding author.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

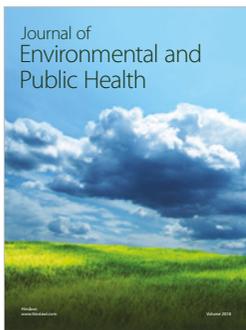
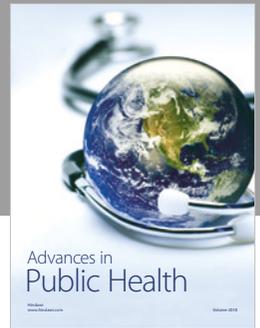
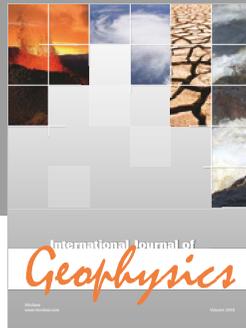
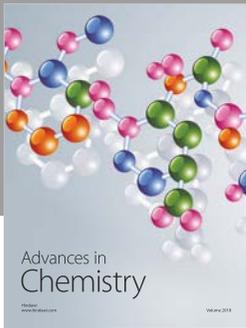
## Acknowledgments

The authors acknowledge Canaan Geo Research for providing GPR data used in this study. This study was supported in part by general account budgets of the universities and the institutions the authors are affiliated with.

## References

- [1] T. Saarenketo and T. Scullion, "Road evaluation with ground penetrating radar," *Journal of Applied Geophysics*, vol. 43, no. 2–4, pp. 119–138, 2000.
- [2] A. Benedetto and S. Pensa, "Indirect diagnosis of pavement structural damages using surface GPR reflection techniques," *Journal of Applied Geophysics*, vol. 62, no. 2, pp. 107–123, 2007.
- [3] W. Al-Nuaimy, Y. Huang, M. Nakhkash, M. T. C. Fang, V. T. Nguyen, and A. Eriksen, "Automatic detection of buried utilities and solid objects with GPR using neural networks and pattern recognition," *Journal of Applied Geophysics*, vol. 43, no. 2–4, pp. 157–165, 2000.
- [4] P. Gamba and S. Lossani, "Neural detection of pipe signatures in ground penetrating radar images," *IEEE Transactions on*

- Geoscience and Remote Sensing*, vol. 38, no. 2, pp. 790–797, 2000.
- [5] M. R. Shaw, S. G. Millard, T. C. K. Molyneaux, M. J. Taylor, and J. H. Bungey, “Location of steel reinforcement in concrete using ground penetrating radar and neural networks,” *NDT & E International*, vol. 38, no. 3, pp. 203–212, 2005.
- [6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.
- [7] J. Ding, B. Chen, H. Liu, and M. Huang, “Convolutional neural network with data augmentation for SAR target recognition,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 364–368, 2016.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Proc. Advances in Neural Information Processing Systems*, vol. 25, pp. 1090–1098, 2012.
- [9] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, “Target classification using the deep convolutional networks for SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806–4817, 2016.
- [10] R. H. Stolt, “Migration by fourier transform,” *Geophysics*, vol. 43, no. 1, pp. 23–48, 1978.
- [11] N. Smitha, D. R. Ullas Bharadwaj, S. Abilash, S. N. Sridhara, and V. Singh, “Kirchhoff and F-K migration to focus ground penetrating radar images,” *International Journal of Geo-Engineering*, vol. 7, no. 1, 2016.
- [12] M. T. Tarner, F. Koehler, and R. E. Sheriff, “Complex seismic traces analysis,” *Geophysics*, vol. 44, pp. 1485–1501, 1979.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
- [14] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 315–323, 2011.
- [15] M. Van Der Baan and C. Jutten, “Neural networks in geophysical applications,” *Geophysics*, vol. 65, no. 4, pp. 1032–1047, 2000.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

