Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.editorialmanager.com/rineng/Default.aspx

# Artificial intelligence system for supporting soil classification

Shinya Inazumi, Ph.D. [a,*], Sutasinee Intui, M.Eng. [b], Apiniti Jotisankasa, Ph.D. [c], Susit Chaiprakaikeow, Ph.D. [c], Kazuhiko Kojima [d]

[a] *Department of Civil Engineering, Shibaura Institute of Technology, 3-7-5 Toyosu, Koto-ku, Tokyo, 135-8548, Japan*
[b] *Graduate School of Engineering and Science, Shibaura Institute of Technology, Japan*
[c] *Civil Engineering Department, Kasetsart University, Thailand*
[d] *Otowa Co. Ltd., Japan*

ARTICLE INFO

*Keywords:*
Deep learning
Image recognition
Machine learning
Neural network
Soil classification

ABSTRACT

From the perspective of soil engineering, soil is uncertain and heterogeneous. Therefore, if an attempt is made to determine the soil classification of a soil without a precise test, for example, an engineer's individual judgement is often involved in making the determination based on his/her own experiences. In relation to acquiring vast and varied knowledge which is easily influenced by individual experiences, the purpose of this study is to gather the know-how of engineers and to create a certain index for use in making on-site judgments that are likely to be more inclusive of various data than those of individual engineers. This study discusses the potential of image recognition by artificial intelligence, using a machine learning technique called deep learning, for the purpose of expanding the cases which employ artificial intelligence. Deep learning was performed with a model using a neural network in this study. For three types of soil, namely, clay, sand, and gravel, an AI model was created that was conscious of the practical simplicity of the images used. It was shown that artificial intelligence, along with deep learning, can be applied to soil classification determination by performing simple deep learning with a model using a neural network.

## Credit author statement

Shinya Inazumi: Summary of research. Sutasinee Intui: Programming implementation. Apiniti Jotisankasa: Programming instruction and implementation. Susit Chaiprakaikeow: Programming instruction. Kazuhiko Kojima: Photography to enter into programming.

## 1. Introduction

In recent years, the working-age population in Japan has been declining, and it is estimated that it will continue to decline in the future. This means a shortage of human resources in every industry, and the civil engineering and construction industries are no exception [1]. The lack of human resources will not only result in a shortage of workers, but will also create an increase in the proportion of foreign workers in each profession. Under such circumstances, it is possible that the technologies and data held by the engineers involved in this work will disappear before they can be satisfactorily passed down to others. In addition, there is concern about certain inconveniences in business efficiency and communication due to the difference in the languages spoken by the engineers.

In terms of soil mechanics and geotechnical engineering, the ground is basically uncertain and heterogeneous (inhomogeneous) because the inside cannot be visually inspected and easy predictions are not possible. Therefore, if an attempt is made to classify a certain type of soil without a precise test, for example, an individual's experience may be involved in making the determination. This leads to it being difficult for engineers to make judgements on soil classification with the same degree of accuracy.

In relation to the uncertainties that make it nearly impossible to guarantee the succession of knowledge and a certain degree of accuracy, this study introduces a method for accumulating and passing down the knowledge of engineers, while also maintaining a certain level of accuracy. The purpose is to expand the use of artificial intelligence (AI) in the field of geotechnical engineering in order to create an index, and shows the possibility of image recognition by AI using machine learning in the field of geotechnical engineering.

* Corresponding author.
*E-mail addresses:* inazumi@shibaura-it.ac.jp (S. Inazumi), na20105@shibaura-it.ac.jp (S. Intui), fengatj@ku.ac.th (A. Jotisankasa), fengssck@ku.ac.th (S. Chaiprakaikeow), kojima@oto-wa.co.jp (K. Kojima).

https://doi.org/10.1016/j.rineng.2020.100188
Received 28 September 2020; Received in revised form 26 October 2020; Accepted 19 November 2020

As an example of previous studies in geotechnical engineering and related fields, Ito [2] prepared 150 sample photographs for three types of soil, namely, organic soil, gravel-mixed sand, and silt-mixed sand. As a result, it was shown that the rate of accuracy obtained by the model (the predictive value of the model against the real phenomenon) improved as the number of learning sessions was increased to 30, 100, 300, and 1000 times. On the other hand, when the 150 sample photographs were converted to grayscale, the black organic soil became almost black and the features of the images were lost. This led to the problem of the black organic soil images being erroneously recognized as silt-blended sand with few features. In addition, in the study by Kiso-Jiban Consultants Co., Ltd. (2019) [3], an AI program using deep learning was developed for three types of rock images, namely, granite, andesite, and mudstone, taken with a digital camera instead of a high-precision camera, such as a single-lens reflex camera. As a result, the rate of accuracy of the AI program was 88% for a total of 100 rock images consisting of 60 granite, 20 andesite, and 20 mudstone pieces. Furthermore, it was shown that the accuracy of this AI program determination exceeded that of geologists and soil engineers. In addition, Koszela et al. [4]; Pegalajar et al. [5]; and Shojaei et al. [6–8] are making advanced attempts to apply AI programs in the field of civil engineering, especially in the field of geoengineering.

In this study, deep learning was performed with a model using a neural network. For three types of soil, namely, clay, sand, and gravel, an AI model was created that was conscious of the practical simplicity of the images used. It is verified that the model misrecognition caused by the loss of image features reported in Ito's study [2] occurs even when another soil type is set as the discrimination target. In addition, although a digital camera was used in the study by Kiso-Jiban Consultants Co, Ltd. (2019), the authors will examine here whether a smartphone with different features from a digital camera can be applied to an AI program.

## 2. Literature review for imaging recognition and machine learning

### 2.1. Neural network

A neural network is a model that mathematically represents the nerve cells in the human brain and their connections. The neurons that make up the network are composed of four parts: the cell body, dendrites, axons, and synapses [9]. Among them, the dendrites function as input terminals that receive information, and the synapses function as output terminals that lead to the dendrites of other neurons. In the human brain, neurons transmit signals by changing potentials. Although there is a difference in potentials between a neuron and the extracellular fluid around it, the potential rises when a signal is input to the neuron, and when a certain threshold is exceeded, information is transmitted to the next neuron. In addition, the synaptic information transmission efficiency is different. The difference in transmission efficiency for each synapse is expressed by the connection weight given for each piece of input information. An arbitrary real value is given to the connection weight. The response of the activation function is obtained based on the input weighted by this connection weight. The response is returned as a numerical value by the activation function.

A neural network is a network that connects a large number of modeled neurons (hereinafter referred to as formal neurons) to process the input information, and it connects these formal neurons in layers. In the hierarchical neural network, the first layer, that is, the layer on the left side of Fig. 1, is called the input layer. It is the layer that receives the data information first. The last layer of the network, that is, the layer on the right side of Fig. 1, is called the output layer. It is the layer that outputs the final calculation results. The layer between the input layer and the output layer is called the hidden layer (or intermediate layer). It receives the output from the previous layer, performs the calculation, and outputs the results to the next layer. In the neural network, the structure of this middle layer has a high degree of freedom, which allows for flexible operations according to the purpose.

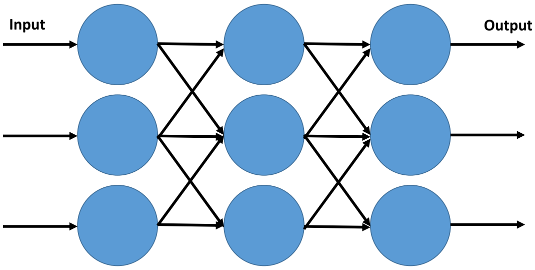


**Fig. 1.** Image of hierarchical neural network.

In this study, a learning model is constructed by adopting the convolutional neural network [10], which is a type of hierarchical neural network that has been successful in tasks related to vision.

### 2.2. Deep learning

Deep learning is a machine learning technique; it involves using a model constructed by stacking many hidden layers of a neural network [11,12]. The network consists of three types of layers: input layer, hidden layer, and output layer. In the network shown in Fig. 2, the left side is the input layer, the right side is the output layer, and the hidden layer is between them. Many of the conventional neural networks have few hidden layers, but the number of hidden layers can be increased, deepening the whole hierarchy, which is the reason why it is called deep learning.

The features of deep learning are that the number of hidden layers can be increased, which makes it possible to perform more complex function approximations than a conventional neural network with few hidden layers. Moreover, the features of the objects to be recognized, such as images, can be automatically extracted, as will be explained in the following. The term "features" is used here to refer to the unique characteristics of an image. As an example, if the image is one of a human face, there will be "one unique shape at the upper left and at the upper right of the image (with eyes)" and "one unique shape at the bottom of the image (with a mouth)" etc. It is the property of the data itself.

If deep learning is not used, it will be necessary to manually extract the features of the object to be recognized, such as an image, and to input it as a set with the images. For this reason, difficulties will be encountered, such as an increase in the time and effort required for the work and different ways of capturing the features extracted to increase the recognition rate. If a human being can easily think "Where are the features?", like of the human face, the latter difficulty can be disregarded to some extent. When it comes to treating soil, however, such as in this study, it becomes difficult to determine what the characteristics or features are. It is hard to decide what to consider or examine.

With deep learning, however, this feature-extraction work can be performed without human processing, and images can be recognized. In other words, it is possible to achieve high accuracy even for images from which it is difficult to extract features. This is also a reason why deep learning is used in this study, namely, because it is compatible with

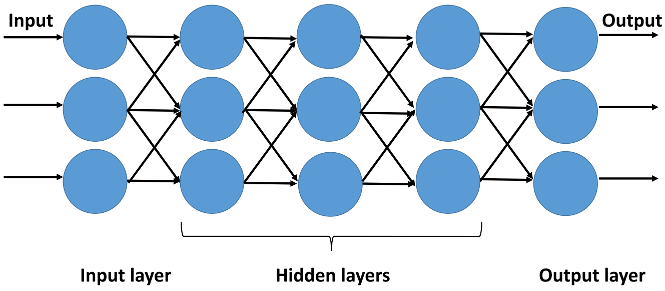


**Fig. 2.** Image of model used in deep learning.

things such as sand and clay for which it is difficult for humans to objectively define what to extract. Looking at the conventional method as an example, the method called support vector machine (hereinafter abbreviated as SVM) is advantageous in that the number of required learning sessions is small because there are few parameters to be optimized, but the accuracy of various data is measured. Therefore, it is necessary to perform a process called cross validation. This process involves dividing the data collected as a sample into training and verification, and then learning and verifying the model. In order to divide the data so that all the data are selected as either training data or verification data, as much as possible, it is necessary to perform the learning and verification steps repeatedly by changing the method multiple times. Therefore, as the amount of data increases, the computational cost becomes enormous. Therefore, addressing a way to limit the computational cost was also significant in this study.

## 3. Conduct model learning

### 3.1. Learning method

The connection weight is applied as the input to the formal neuron. As can be expressed by Eq. (1), firstly, the sum of the products of the input and the connection weight (called the net value, u) is obtained. The sigmoid function in Eq. (2) is used as the activation function for finding the output of the unit based on the net value. In this study, the sigmoid function was selected in consideration of the ease of calculation and the expression for backpropagation to be described later.

$$\mathrm{u} = X_1W_1 + X_2W_2 \ldots + X_nW_n = \sum_{i=1}^{n} X_iW_i \qquad (1)$$

where X is the input and W is the coupling weight.

$$\mathrm{f(x)} = \frac{1}{1+e^{-x}} \qquad (2)$$

Using sigmoid function f(x) for net value u, to represent the final output of the neuron, the following form is obtained for Eq. (3):

$$\mathrm{y} = \mathrm{f}(u) \qquad (3)$$

The "error" in this study is the sum of the squares of the difference between the true value (called the teacher signal), based on the prepared data, and the output of the model expressed by Eq. (3), etc. This is called the error function, namely, error function E when there are n formal neurons in the output layer as can be expressed in Eq. (4).

$$\mathrm{E} = \frac{1}{2}\sum_{i=1}^{n}(o_i - t_i)^2 \qquad (4)$$

where o is the output and t is the teacher signal.

"Learning" refers to changing the value of the coupling weight in order to minimize the value of the error function. In this study, backpropagation is used for learning. This is because "the data input from the input layer is transmitted to the intermediate layer and output is performed at the output layer. This is done by updating the coupling load between the two and updating the coupling load between the middle layer and the input layer" [13,14]. A method called the steepest descent method is used when updating the connection weight.

In the steepest descent method, the error and the joint weight (an arbitrary real value) are considered, as shown in Fig. 3. From this figure, it can be seen that the error varies depending on the value of the joint load. In the steepest descent method, the gradient in Fig. 3 is calculated by Eq. (5), and by multiplying this by −1 and the learning coefficient, the value that modifies the coupling load in the opposite direction to the gradient can be calculated [13,14].

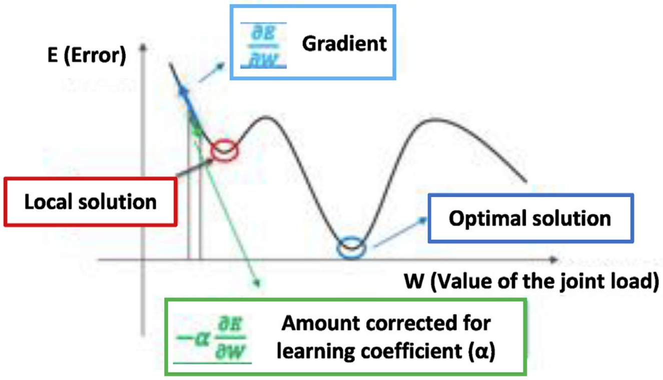

E (Error)
$\frac{\partial E}{\partial W}$ Gradient
Local solution
Optimal solution
W (Value of the joint load)
$-\alpha\frac{\partial E}{\partial W}$ Amount corrected for learning coefficient (α)

**Fig. 3.** Schematic of steepest descent method [13,14].

$$\frac{\partial \mathrm{E}}{\partial \mathrm{W}} \qquad (5)$$

where E is the error and W is the coupling weight.

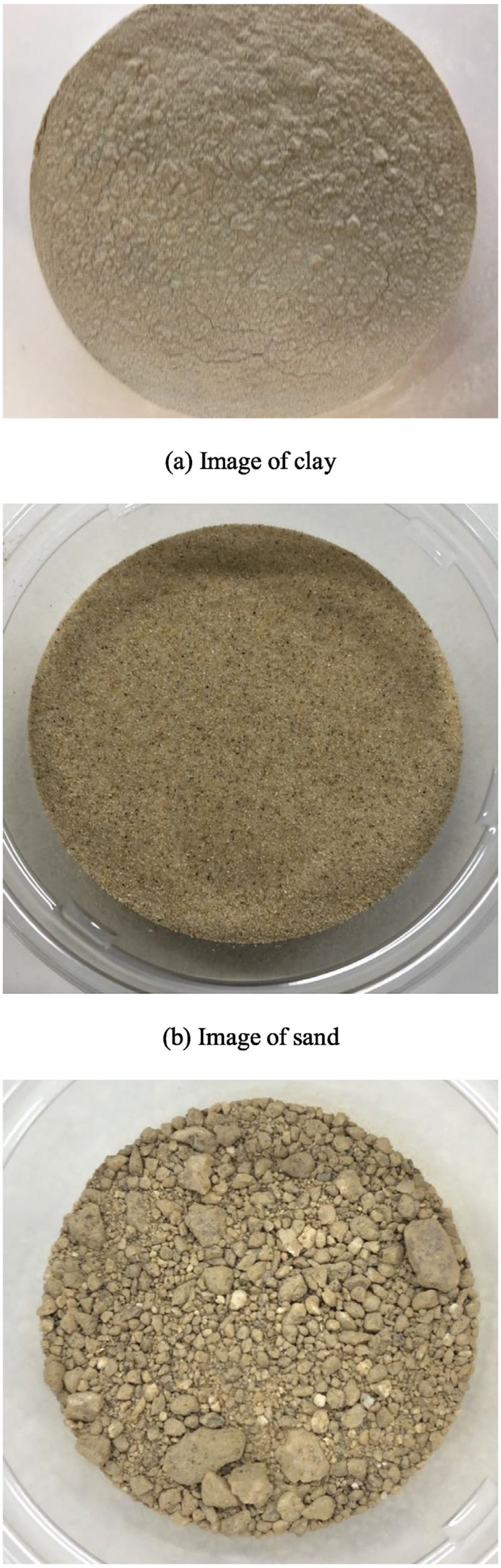
The connection weight can be updated by applying the value at the present time to the connection weight, and this updating process is called learning. The steepest descent method is used to obtain the connection weight that minimizes the error by updating the connection weight in this way.

Because this correction amount is obtained by multiplying the learning coefficient by the gradient in Eq. (5), the coupling weight is updated by increases or decreases depending on the size of the learning coefficient. Therefore, if the learning coefficient is too large, there is a possibility of jumping over the coupling weight (called the optimal solution) where the error becomes the smallest. On the other hand, if the learning coefficient is too small, the update width in one learning session becomes small, and the number of learning sessions required to reach the optimum solution increases. Not only that, but if there is a value (local solution) for which the error function takes a minimum value before the optimal solution, the update of the connection weight may stop at the local solution and the optimal solution may not be reached. Considering these factors, it is necessary to find a learning coefficient suitable for the model and the data used for learning.

In addition, when using the sigmoid function as the activation function, it is necessary to pay attention to the problem of gradient disappearance. The maximum value of the gradient of the function used for parameter adjustment, that is, the derivative value, is 0.25 in the sigmoid function. Therefore, in backpropagation in which the coupling load is adjusted from the output layer to the input layer, a value that is less than 0.25 is multiplied in each layer that is processed later, and the amount of fluctuation tends to be small. Therefore, if there are many layers that use the sigmoid function, the gradient required for adjustment may almost be eliminated, especially when there are four or more layers. The above is an outline of the gradient vanishing problem. In this study, the influence of the vanishing gradient problem is reduced by using three layers and employing the sigmoid function.

#### 3.1.1. Image used

The purpose of this study is to create the basis for a model to be used to make judgements on soil classification that can be utilized on a site-by-site basis. For this reason, the soils to be classified were divided into three types: clay (D50 = 0.008 mm), sand (D50 = 0.7 mm), and gravel (D50 = 4 mm) with the water content adjusted to 0 for simplicity. Clay, sand, and gravel, whose particle sizes were adjusted by conducting a sieving test, were put in a transparent plastic cup as shown in Fig. 4, considering this practice as a difference from the previous studies [2,3,15]. The images were taken with a smartphone (iPhone 7) camera. Comparing the performances of the smartphone camera and the digital camera (FUJIFILM X-T4) released at the time of the experiment, the iPhone 7 has 12 million



(a) Image of clay

(b) Image of sand

(c) Image of gravel

**Fig. 4.** Examples of soil image used to develop AI program for soil classification.

pixels and the digital camera has 26 million pixels. In addition, the size of the image sensor related to image noise (the larger the image, the more faithful the image to the subject) is 4.8 × 3.6 mm for the iPhone 7 and 23.5 × 15.6 mm for the digital camera. It is seen that the photographs were shot indoors. As for the clay, 200 photographs were taken both with and without lighting (400 in total). As for the sand, 200 photographs were also taken with and without lighting (400 in total). As for the gravel, 100 photographs were taken with and without lighting (200 in total). In order to see the influence of the difference in the amount of data on the learning results, only the gravel had a different amount of data. In order to secure the data variation, adjustments such as switching the presence or absence of lighting during shooting and adding vibration or rotation to the cup to change the appearance of the surface were added. Vibration was applied so that the arrangement of the particles on the sample surface would be completely different, and rotation was performed in 90-degree units. These processes were combined to prevent the data from becoming uniform. This is because, even if a large number of identical images are copied and prepared, the effect of the learning model will be weak.

Table 1 presents a list of prepared images. These 1000 images were used as the learning data, and a total of 60 images of each randomly selected 20 images were used as model accuracy verification data.

### *3.1.2. Parameters used*

In this study, the soil images were learned as a model using deep learning by the convolutional neural network and by the steepest descent method. At that time, each parameter was set as shown in Table 2.

Regarding the learning coefficient, 1e-1 to 1e-9 were tested. Finally, 1e-6 was chosen; it was the only coefficient which yielded a significant result in the model of this study. With the other learning coefficients, the accuracy did not improve even if the number of learning sessions was increased, and only the accuracy of the learning data became extremely high, and the accuracy of the model for unknown data, such as verification data, did not improve. What occurred was the phenomenon of overlearning and the learning did not proceed normally.

The batch size represents the number of learning images divided into groups and used in one learning session. Because processing, such as updating the parameters of the connection weight, is performed for each group, the number of parameter updates increases and the learning time increases when the batch size is small, that is, the number of groups is large. In consideration of the balance between the required time and the number of trials, the batch size was set to 20 in this study.

The number of times of learning is literally set to how many times to learn, but if the number of times is too small, sufficient accuracy cannot be obtained, and if it is too large, a phenomenon called overlearning occurs depending on the learning coefficient. Here, overlearning means that the model is over-optimized to the characteristics of the training data and loses its versatility with respect to other unknown data such as verification data and new data to be judged after the training. This versatility for various data is called generalization performance, which is the performance that should be emphasized when creating a model. When compared with humans, it is similar to the situation in which the scores of other tests are low compared to a certain one, although all the scores for the test are all very high, because the students studied very

**Table 1**
Types and numbers of images taken.

| | Clay | Sand | Gravel | (Total) |
|---|---|---|---|---|
| Presence of lighting | 200 | 200 | 100 | 500 |
| Absence of lighting | 200 | 200 | 100 | 500 |
| (Total) | 400 | 400 | 200 | 1000 |

**Table 2**
Parameters used.

| Learning coefficient | 1e-06 |
|---|---|
| Batch size | 20 |
| Number of times of learning | 70 |
| Image size during learning (px) | 56 × 56 |

diligently for the test. When the model overlearns in this study, the accuracy of the image classification for clay, sand, and gravel used for learning becomes very high. However, when the completed model is put into practical use, the classification accuracy with other captured images of clay, sand and gravel will be low.

To prevent overlearning, the model first learns the characteristics of the data, and then terminates the learning when the accuracy has increased moderately. For example, if 100 learning sessions causes overlearning, it means that less than 90 or 80 sessions will complete the learning process. In addition, it is effective to increase the sample data for learning. In the first place, over-learning means that the model is optimized for sample data with limited variations and cannot be applied to new data, that is, unknown variation data. Therefore, it is also important to suppress over-learning by increasing the variation of the data prepared as a sample (soil quality in the case of the model of this study) as much as possible and reducing the variation unknown to the model.

Also, the number learning sessions is directly related to the time required for learning. The PC used in this study required about 2 h for 70 learning sessions, so the study was forced to continue for 70 sessions. In this study, the CPU was Intel Core i5-3320 M 2.6 GHz, and the internal GPU was an Intel HD Graphics 4000 notebook PC.

Also, in order to reduce the processing load on the program, the number of pixels in the image was reduced. In this study, the image was resized to 56 × 56 pixels and then processed. Regarding the number of pixels, when initially learning with 26 × 26 pixels, the model sometimes could not identify the characteristics of sand and gravel, and judged all the gravel images as sand. After that, when the size was changed to 56 × 56 pixels, an improvement was found in its ability to discriminate between sand and gravel. In this study, therefore, learning was finally conducted with images of 56 × 56 pixels.

### 3.2. Learning flow and implementation

#### 3.2.1. Flow of learning

In this study, Python (Ver. 3.6.6) was used as the programming language and Tensorflow (Ver. 1.9.0) [16–18], published by Google Inc., was used as the library of functions required for learning.

The model constructed in this study mainly consists of two convolutional layers, two pooling layers, and a fully connected layer. In Fig. 5, “conv” corresponds to the convolutional layers and “pool” corresponds to the pooling layers. These layers sequentially detect and process image features and pass them on to the next layer. Furthermore, after detecting the features in all layers, the prediction of the learned model and the true value of the image are compared, and if there is an error, the weight of the connection weight is updated according to the error. At this time, the algorithm for correcting the connection weight, shown in Fig. 3 and based on the error, is a training algorithm called AdamOptimizer, provided by Tensorflow. Adam, shown in Fig. 5, represents this. The procedure is repeated to reduce the error. It should be noted that “init” in Fig. 5 represents the process of initializing the parameters and starting the calculation associated with learning, and that “gradients” represents the process associated with the gradients of the error function.

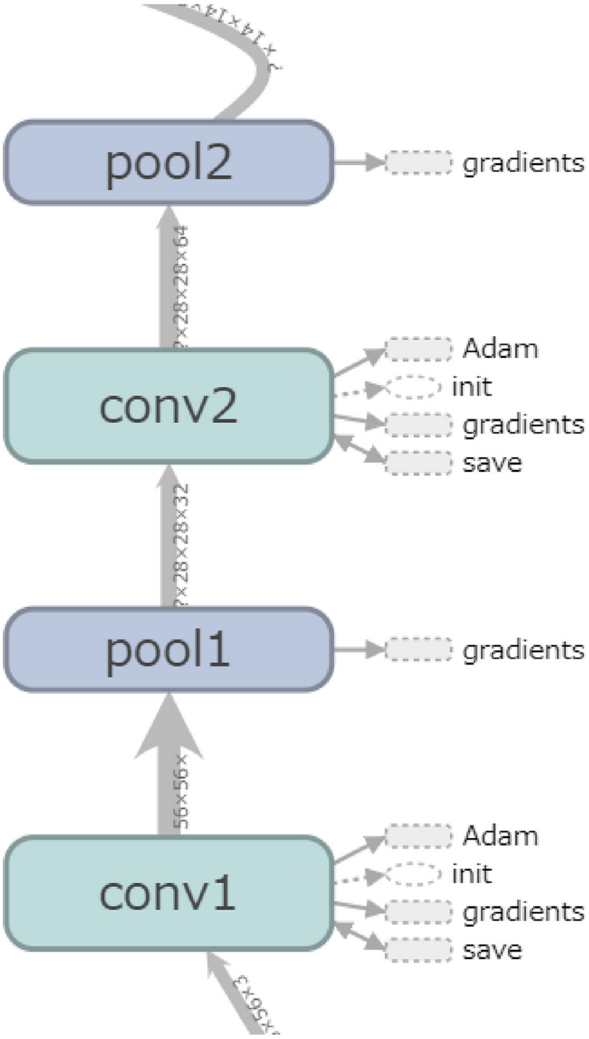
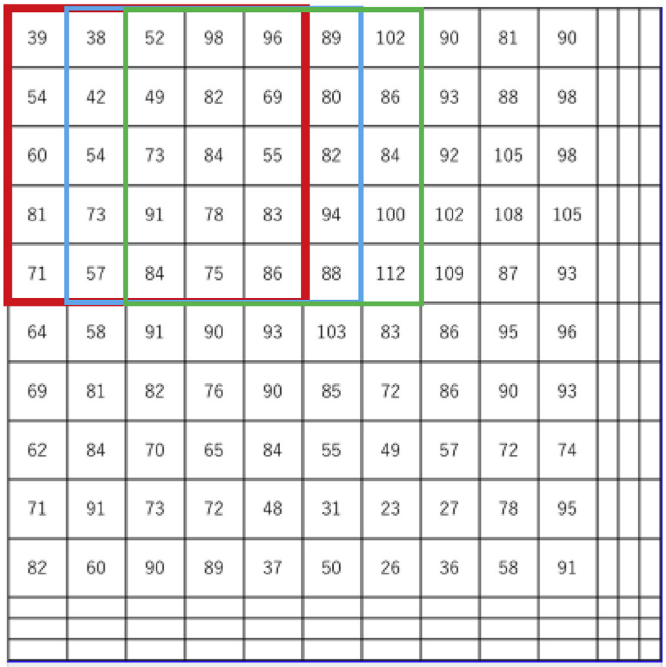


**Fig. 5.** Image of model’s multi-layer structure [16–18].

As a concrete flow, the feature of the image is detected in the first convolutional layer (conv1). In the first place, faster images are innumerable squares with slightly different colors when enlarged. Each of these squares is called a pixel. In this study, the image is resized to 56 × 56 pixels and treated; the images with 56 × 56 squares are lined up and should be considered. Each pixel has a value of 0–255 as data for each of the three colors of red, green, and blue. In this study, the image is converted to grayscale. Therefore, the color data held by each pixel is a numerical value with 256 levels from 0 to 255. It is thought that the numbers from 0 to 255 are lined up in 56 × 56 pieces. Fig. 6 illustrates this. In this way, while numerical data are arranged for each pixel, in the first convolution layer, the numerical values are compressed in the range of 5 × 5 pixels from the upper left portion of the image. At this time, as shown in Fig. 7, the weight parameter in the range of 5 × 5 (this size is arbitrary), called the kernel, is applied to the range of 5 × 5 pixels from the upper left portion of Fig. 6, and the corresponding cells are multiplied by a numerical value. In this case, the numerical value in the upper left portion (first row, first column) in the image data of Fig. 6 is 39, and the corresponding value of the kernel is also 0 in the first row, first column. 39 is multiplied by 0 which yields 0 as the product as the first solution. Next, looking to the right, the number at the first row and second column in Fig. 6 is 38, and the corresponding number in Fig. 7 is −1. If these are multiplied, −38 is obtained as the second solution. This operation is performed in the range of 5 × 5, which is the size of the kernel, and the sum of all the 25 products obtained is 28. This total becomes the numerical data in the first row and first column of the compressed image. After that, the same calculation is repeated by shifting the range to which the kernel is applied by 1 pixel. This series of operations is called convolution. In the model of this study, two convolution layers are used.

After convolution in the convolution layers, the work of blurring the image features is performed in each pooling layer. Here, the numerical



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 39 | 38 | 52 | 98 | 96 | 89 | 102 | 90 | 81 | 90 |
| 54 | 42 | 49 | 82 | 69 | 80 | 86 | 93 | 88 | 98 |
| 60 | 54 | 73 | 84 | 55 | 82 | 84 | 92 | 105 | 98 |
| 81 | 73 | 91 | 78 | 83 | 94 | 100 | 102 | 108 | 105 |
| 71 | 57 | 84 | 75 | 86 | 88 | 112 | 109 | 87 | 93 |
| 64 | 58 | 91 | 90 | 93 | 103 | 83 | 86 | 95 | 96 |
| 69 | 81 | 82 | 76 | 90 | 85 | 72 | 86 | 90 | 93 |
| 62 | 84 | 70 | 65 | 84 | 55 | 49 | 57 | 72 | 74 |
| 71 | 91 | 73 | 72 | 48 | 31 | 23 | 27 | 78 | 95 |
| 82 | 60 | 90 | 89 | 37 | 50 | 26 | 36 | 58 | 91 |

**Fig. 6.** Image of digitizing images [10].

| | | | | |
|---|---|---|---|---|
| 0 | -1 | -1 | -1 | 0 |
| -1 | 0 | 3 | 0 | -1 |
| -1 | 3 | 0 | 3 | -1 |
| -1 | 0 | 3 | 0 | -1 |
| 0 | -1 | -1 | -1 | 0 |

**Fig. 7.** 5 × 5 kernel (Ng et al., 2020).

data of the image input to each pooling layer is compressed to 1 × 1 pixel every 2 × 2 pixels from the edge [19]). That is, when processing is applied to a 4 × 4 pixel image, it is output as 2 × 2 pixel data. In this way, the number of pixels in the image is halved as it passes through each pooling layer.

After detecting the features of the image in these layers, the output of the neural network is finally converted into the probability by the Softmax function to calculate what kind of error exists between the model prediction and the true value for the image. Here, the true value is the name of the soil featured in the image given to the model, for example, in this study. As a concrete example, if an image showing sand is given to the model, the true value is "sand", and if the prediction result of the model is "sand", the answer is correct; otherwise it is incorrect. After these processes, the weight of the connection weight is updated according to the calculated error.

The multi-layer structure of the convolutional layers and the pooling layers, shown in Fig. 5, can be expressed in an actual program, as shown in Fig. 8. In the code in this figure, four groups separated by blank lines correspond to the first and second convolutional layers and the first and second pooling layers, respectively. Because these structures can be freely modified, such as by increasing or decreasing the number of convolutional/pooling layers, it is possible to make them into a suitable form according to the type of data to be classified, if necessary.

```
with tf.name_scope('conv1') as scope:
  W_conv1 = weight_variable([5, 5, 3, 32])
   b_conv1 = bias_variable([32])
   h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)

with tf.name_scope('pool1') as scope:
   h_pool1 = max_pool_2×2(h_conv1)

with tf.name_scope('conv2') as scope:
  W_conv2 = weight_variable([5, 5, 32, 64])
   b_conv2 = bias_variable([64])
   h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)

with tf.name_scope('pool2') as scope:
   h_pool2 = max_pool_2×2(h_conv2)
```

**Fig. 8.** Part of the program.

### 3.2.2. Implementation of learning

The model was trained using the model of the structure shown in Fig. 5 and the parameters shown in Table 2. The personal computer used in this study required about 2 h for 70 learning sessions.

The parameters finally adopted in this study are summarized in Table 2, but multiple cases were also conducted here with different learning coefficients. Although 1e-1 to 1e-2, 1e-3 … and 1e-9 were tried, significant results were obtained only in the case of 1e-6. Examples of unsuccessful cases were cases in which the accuracy did not improve even after repeated learning sessioins, as shown in Fig. 9, or overlearning, as shown in Fig. 10.

## 4. Learning results and considerations

### 4.1. Results of learning

Fig. 11 shows the transition of accuracy with respect to the number of learning sessions. The transition of the error is shown in Fig. 12. From these figures, it can be seen that the accuracy improves and the error decreases as the number of learning sessions increases. The error referred to here is the one defined by Eq. (4). Finally, an accuracy of about 86% was recorded for the image data for learning and about 77% for the data for verification. Here, the accuracy is the rate at which the images of clay, sand, and gravel used for learning can be accurately identified. In other words, out of a total of 1000 learning data images, about 86% (about 860) of the images were correctly classified.

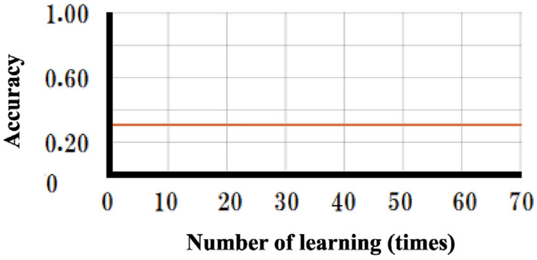


**Fig. 9.** Transition of accuracy with learning coefficient 1e-4.

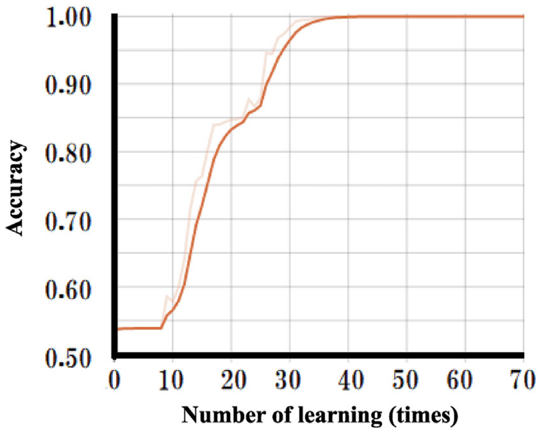


**Fig. 10.** Transition of accuracy with learning coefficient 1e-5.

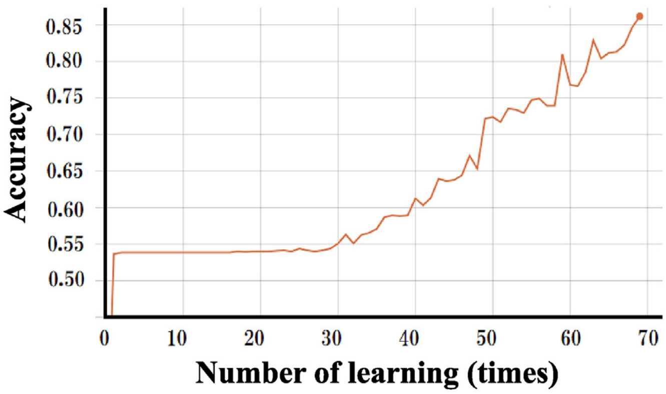


**Fig. 11.** Transition of accuracy with learning coefficient 1e-6.

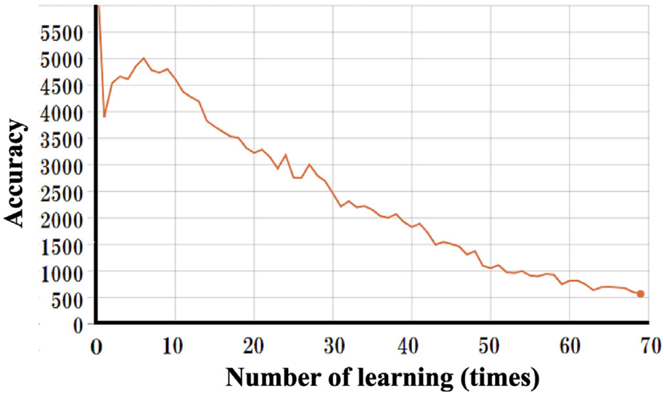


**Fig. 12.** Transition of error in learning coefficient 1e-6.

In order to confirm whether the model can correctly recognize the training data, Table 3 shows the judgment results for each soil type among the training data. The table shows how the model determined each soil type. For example, as a result of inputting 400 learning images showing clay as a true value into the model, 313 of the images were determined to be clay. At the same time, 87 of the images had the true value of clay, but the model misidentified them as sand. From the table, it can be seen that all the sand was correctly recognized and only about half of the gravel was correctly recognized. In particular, the gravel and clay were often mistaken for sand.

For these results, the evaluation index for the trained model is applied. Equation (6) represents the recall.

$$\frac{x}{n} = \frac{x}{x+y} \tag{6}$$

where n is the number of data of a specific class C, x is the number of data classified into class C, and y is the number of data classified into other classes.

The recall is an index that focuses on the true class of the images to be classified, and refers to the ratio of the images in which the model can be accurately recognized as sand, for example, of the images in which what is truly sand is shown. When this model is evaluated using the recall, the recall of sand is 1, and it seems that sand can be recognized very accurately. On the other hand, the recall rate of gravel is about 0.54; thus it seems that gravel cannot be recognized accurately.

Now, an attempt is made to find the precision rate, which is another index. The precision is calculated by Eq. (7).

$$\frac{w}{w+z} \tag{7}$$

where w is the number of data samples that truly belongs to class C among the data classified into class C, and z is the number of data samples that does not belong to class C among the data classified into class C.

The relevance ratio is an index that focuses on the previous class into which the images have been classified and, for example, refers to the ratio of images that are truly clay in the image group that the model has determined to be clay. In other words, it is an index showing how carefully the clay is classified, and whether non-clay has been classified as clay. When the model is evaluated using the precision rate, the precision rate is 1 for gravel, which had a low recall of 0.54. In other words, the gravel images are classified into another class at a rate of about 46%, but all the images recognized as the remaining gravel were truly gravel. In this way, recall and precision are in a trade-off relationship, and if either increases, the other decreases. Therefore, it is difficult to evaluate the model satisfactorily using either the recall rate or the precision rate. In this study, the F value, which is an index for integrating these two rates (recall and precision), is used. The F value is the harmonic mean of recall and precision, and is calculated by Eq. (8) [20,21].

A model that increases this F value is considered to be good [22].

$$\mathrm{F-measure} = \frac{2{*}Pre{*}Rec}{Pre + Rec} \tag{8}$$

where Pre is the precision and Rec is the recall.

Table 4 shows the recall, precision, and F value.

### 4.2. *Considerations of learning*

When the learning session results were summarized, the accuracy of the learning images was about 86% and the accuracy of the verification data was about 77% through 70 learning sessions.

When the evaluation index was applied to the 1000 sheets of data used for learning, the sand recall rate was extremely high at 1. This

**Table 3**
Judgment results for each soil type.

| True value | Clay | | | Sand | | | Gravel | | |
|---|---|---|---|---|---|---|---|---|---|
| Class | Clay | Sand | Gravel | Clay | Sand | Gravel | Clay | Sand | Gravel |
| Judgment result | 313 | 87 | 0 | 0 | 400 | 0 | 10 | 83 | 107 |

**Table 4**
Results of learning with 56 pixels (Each evaluation index).

| | Number of data images | Correctly classified number of data images | Recall | Precision | F value |
|---|---|---|---|---|---|
| Clay | 400 | 313 | 0.78 | 0.97 | 0.87 |
| Sand | 400 | 400 | 1 | 0.70 | 0.82 |
| Gravel | 200 | 107 | 0.54 | 1 | 0.70 |

**Table 5**
Results of learning with 28 pixels (Each evaluation index).

| | Number of data images | Correctly classified number of data images | Recall | Precision | F value |
|---|---|---|---|---|---|
| Clay | 400 | 392 | 0.98 | 1 | 0.99 |
| Sand | 400 | 400 | 1 | 0.74 | 0.85 |
| Gravel | 200 | 0 | 0 | 0 | 0 |

means that all the sand images could be identified as sand. On the other hand, the reproducibility of gravel and clay was low, with the reproducibility of gravel at only 0.54. To address this low rate, it is thought that the model could not learn the characteristics of clay and gravel satisfactorily, and thus, mistakenly recognized them as another soil type. The number of pixels after resizing the image was too small, and the original features of the image were destroyed, so they could not be detected satisfactorily. In addition to this, it is possible that the number of learning sessions was small, especially for gravel, and that the number of learning images was also small. From Table 3, almost all the false positives were for sand.

If the image size during the learning session is too small, the features of the image cannot be extracted satisfactorily, which is unsuitable. However, if the image size is too large, the time and mechanical cost for the learning session will increase, so care must be taken when setting the image size during learning. In this study, a policy was set to gradually increase the image size from a small value of 28 × 28 pixels while checking the results. As a result, movement was seen in the second 56 × 56 pixels; and thus, the value of 56 × 56 pixels was used here. However, from the viewpoint of optimal parameter setting, it is also important to continue to increase the size. If the computer used for learning has a high performance, it is considered to be effective if the image size is started from a large value and then adjustments are made based on the results, such as decreasing the size.

In addition, before using the parameters shown in Table 2, learning was performed with an image size of 28 × 28 pixels. At that time, as shown in Table 5, the model mistakenly recognized all the gravel images as sand. After that, when learning was performed with the image size set to 56 × 56 pixels, as shown in Table 4, about half of the gravel images were accurately discriminated.

From these, the reason why the soil characteristics cannot be accurately recognized is that the image size during learning is small. Because the learning rate was improved when the size was expanded from the first learning with 28 pixels to 56 pixels, it can be assumed that increasing the image size for gravel will lead to an improvement in the recall rate …

A comparison of the clay recall and the precision rate in the 28 pixel case and the 56 pixel case shows a slight decrease in the 56 pixel case. The direct reason for the decrease in recall was that the number of false positives for clay as sand increased. This is thought to be due to the difference in the number of learning sessions. In the case of 28 pixels, the learning session is performed 100 times, while in the case of 56 pixels, the learning session is performed 70 times. It can be said that the number of cases in which clay images were correctly classified can be said to be large because the characteristics of clay can be learned relatively well in the case of 28 pixels by the difference of 30 times. The reason for the decrease in the precision is that the characteristics of the image of gravel can be recognized normally and, as a result, gravel is sometimes mistaken for clay. That is, in the case where the number of pixels was small, all 200 images of gravel were mistaken for sand, but in the case where the number of pixels was large, 83 images of gravel were mistaken for sand and 10 for clay. These 10 pieces reduce the compatibility of clay.

There are differences in the F value between sand/clay and gravel, which indicate that the characteristics of gravel are not sufficiently detected. The reason is thought to be that the number of pixels for learning is small and the amount of gravel learning data is less than that of clay or sand. The data belonging to such a data group in which the numbers are not balanced are called imbalanced data. When learning with imbalanced data, the features obtained from each data set are weighted. However, it is necessary to take measures, such as learning the features of the small amount of data well. Because this processing was not performed this time, it is considered that the difference in the amounts of data affects the difference in the F value. The overall balance of the F values can be improved by increasing the number of pixels during the learning sessions, as described above, and by making the amounts of learning data sets sufficiently uniform and even.

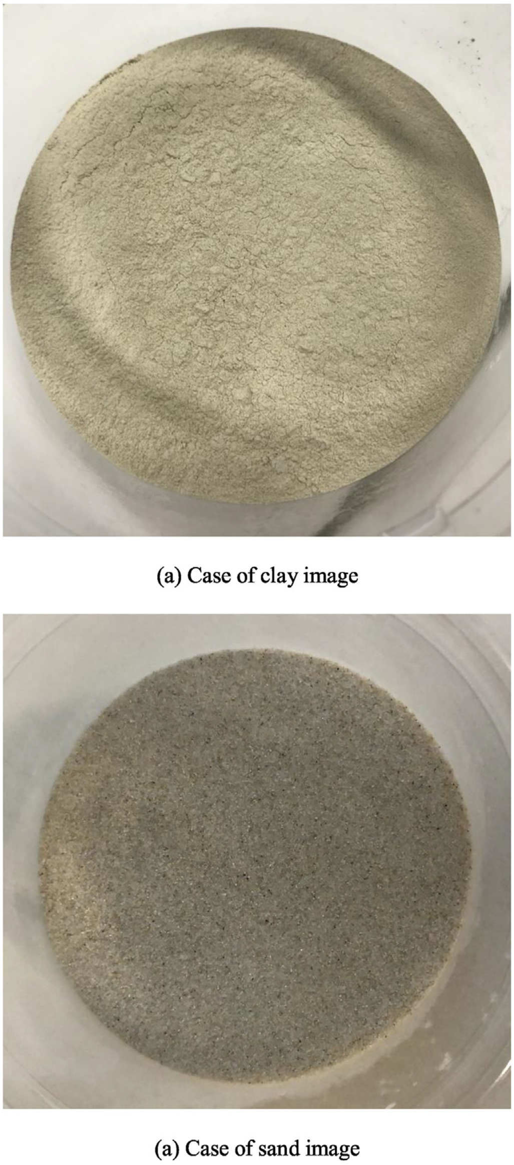
As for sand, the results of this study show that the precision rate of sand is particularly low. This is because there were many cases where the images of clay and gravel were mistakenly recognized as sand. The false recognition was based on the above consideration, as well as on the number of pixels in the image during learning, the number learning sessions, and the number of data images for learning. It is thought that this is a structure in which many images look like sand to the model because the elements on the model side, such as balance, and the elements on the image side, such as grain size (feature distribution) and shadow, did not mesh well.

The low precision means that the models classified as A are mixed with those different from B or C …. If a model with such a defect were to be put into practical use, naturally there would be a problem with reliability, but as in the model of this study, the precision rate was low only in a specific class, and the precision rate was low in other classes. If the model tends to be high, even if it is put to practical use as it is, it can be used to some extent by being captured by the user. In other words, if only the sand has a low matching rate, the reliability of the model in judging gravel or clay is guaranteed to have a certain accuracy. The user would then need to be careful when judging whether or not it is sand. However, variations in recall, precision, and the F value are not sufficient indicators, so it is important to review the learning parameters and to create a model with a better evaluation index, such as the F value.

The overall level of accuracy can be expected to improve to a certain degree by increasing the number of learning sessions. In this study, learning sessions were performed 70 times in order to balance the time required by the PC (used) and the number of trials, but it is highly possible that the learning was completed before the accuracy was deemed satisfactory at 70 sessions. Therefore, it is thought that increasing the number of learning sessions, while paying attention to prevent overlearning, will lead to the improvement of the model.

In addition, learning and classification were performed in this study by targeting only three types of soil. However, considering further development, not only the adjustment of parameters but the expansion of corresponding soil is the first issue. As shown in Fig. 13, clay and sand, which look very similar to human eyes, can be discriminated by the model. It is possible to obtain a generalization performance for various types of soils by preparing several types, securing a sufficient amount of data for learning, and conducting the learning sessions with a sufficiently large number of pixels. It is thought that this will soon be possible.

In addition, considering the practicality in the field, if it becomes possible to roughly determine the classification and mixing state of gravel, sand, silt and clay, it can be said that the utility as a rudimentary index will be created. Furthermore, instead of a model that simply discriminates soil types, a model that is specialized only for soils with a specific tendency will come into view. For example, it would be very beneficial if there were a model that could determine whether or not a certain kind of sandy soil, for example, had a high risk of liquefaction, in order to determine the risk of liquefaction.



(a) Case of clay image

(a) Case of sand image

**Fig. 13.** Examples of images that could be accurately identified by the developed AI program.

## 5. Conclusions

In this study, deep learning was performed with a model using a neural network. For three types of soil, namely, clay, sand, and gravel, an AI model was created that was conscious of the practical simplicity of the images used. It was shown that this AI model can be applied to make judgments on soil classification. As a result, a high recall rate of 1 was obtained for sand. This means that all the sand images could be identified as sand. On the other hand, a high matching rate was obtained for clay and gravel. This means that images of clay and gravel can be carefully discriminated without much mixing of different types of images. Regarding the parameters, if the number of pixels in the image during the learning sessions is too small, the features of the images cannot be detected sufficiently and the accuracy decreases. If the number of pixels is increased, the features can be detected and the accuracy increases.

There are two issues to be addressed in the future: the further adjustment of the parameters related to learning, such as the above-mentioned number of pixels, and securing a sufficient number of learning sessions and verification data. Regarding the parameters, only the four parameters shown in Table 2 were adjusted in this study. However, in addition to the initial values of the coupling load and the parameters, the number and the structure of the convolutional layers and the pooling layers, etc. are among the many things that can be adjusted. Regarding the number of data samples, an unbalanced number of images was used for learning in this study, so it will be necessary to secure a sufficient number of data samples for the learning of clay, sand, and gravel to correct this imbalance.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] S. Kusayanagi, To build up the appropriate education program for construction management in Japan, Journal of Construction Management 11 (2004) 281–292.

[2] M. Ito, Possibility of automated soil quality judgment by machine learning, Proceedings of ZENCHIREN Technology Forum 2017 Asahikawa, http://www.takuhoku-chika.co.jp/machinelearning/resume.pdf, 2017.

[3] S. Utsuki, M. Nakaya, R. Tsuruta, Construction of a geological information management system using AI, CIM and image processing technology, Journal of Japan Society of Engineering Geology 58 (6) (2018) 408–415.

[4] K. Koszela, J. Przybył, S. Kujawa, R.J. Kozłowski, K. Przybył, G. Niedbała, P. Idziaszek, P. Boniecki, M. Zaborowicz, IT system for the identification and classification of soil valuation classes, Proceedings of the Eighth International Conference on Digital Image Processing (ICDIP 2016) 10033 (2016) 100332J.

[5] M.C. Pegalajar, L.G.B. Ruiz, M. Sánchez-Marañón, L. Mansilla, A munsell colour-based approach for soil classification using fuzzy logic and artificial neural networks, Fuzzy Set Syst. 401 (15) (2020) 38–54.

[6] S. Shojaei, M.A.H. Ardakani, H. Sodaiezadeh, M. Jafari, S.F. Afzali, Optimization using response surface method (RSM) to investigate the compaction of mulch, Modeling Earth Systems and Environment 5 (4) (2019a) 1553–1561.

[7] S. Shojaei, M.A.H. Ardakani, H. Sodaiezadeh, Optimization of parameters affecting organic mulch test to control erosion, J. Environ. Manag. 249 (No. 1) (2019b) 109414.

[8] S. Shojaei, M.A.H. Ardakani, H. Sodaiezadeh, Simultaneous optimization of parameters influencing organic mulch test using response surface methodology, Sci. Rep. 10 (1) (2020) 1–11.

[9] W. Ng, B. Minasnya, M. Montazerolghaem, J. Padarian, R. Ferguson, S. Bailey, A.B. McBratneya, Convolutional neural network for simultaneous prediction of several soil properties using visible/near-infrared, mid-infrared, and their combined spectra, Geoderma 352 (2019) 251–267, https://doi.org/10.1016/j.geoderma.2019.06.016.

[10] W. Ng, B. Minasny, A. McBratney, Convolutional neural network for soil microplastic contamination screening using infrared spectroscopy, Sci. Total Environ. 702 (2020) 134723, https://doi.org/10.1016/j.scitotenv.2019.134723.

[11] J. Padarian, B. Minasny, A.B. McBratney, Using deep learning to predict soil properties from regional spectral data, Geoderma Regional 19 (2019), e00198 ttps://doi.org/10.1016/j.geodrs.2018.e00198.

[12] A. Azizi, Y.A. Gilandeh, T.M. Gundoshmian, A.A.S. Bigdeli, H.A. Moghaddam, Classification of soil aggregates: a novel approach based on deep learning, Soil Tillage Res. 199 (2020) 104586, https://doi.org/10.1016/j.still.2020.104586.

[13] J. Tani, N. Fukumura, Embedding a grammatical description in deterministic chaos: an experiment in recurrent neural learning, Biol. Cybern. 72 (4) (1995) 365–370.

[14] J. Yang, R. Horie, A Computer Interface Realized by a Recurrent Neural Network and a Natural User Interface Based on Tracking Hand Motion, Life2014, 2014.

[15] Kiso-Jiban Consultants Co.,Ltd.. https://www.kisojiban.com, 2019.

[16] M. Abadi, TensorFlow: learning functions at scale, Proceedings of the 21st ACM SIGPLAN International Conference on Functional ProgrammingSeptember, https://doi.org/10.1145/2951913.2976746, 2016, 1.

[17] M.A. Abu, N.H. Indra, A.H.A. Rahman, N.A. Sapiee, I. Ahmad, A study on image classification based on deep learning and Tensorflow, Int. J. Eng. Res. Technol. 12 (4) (2019) 563–569.

[18] E.P.B. Guidang, Classifying soil texture images using transfer learning, IOP Conf. Ser. Mater. Sci. Eng. 482 (2019), 012042.

[19] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016.

[20] D.H. Buckley, T.M. Schmidt, Diversity and dynamics of microbial communities in soils from agro-ecosystems, Environ. Microbiol. 5 (6) (2003) 441–452, https://doi.org/10.1046/j.1462-2920.2003.00404.

[21] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep Learning Approach for Network Intrusion Detection in Software Defined Networking, Proceedings of 2016 International Conference on Wireless Networks and Mobile Communications, WINCOM), 2016, pp. 258–263, https://doi.org/10.1109/WINCOM.2016.7777224.

[22] E. Alpaydin, Introduction to Machine Learning, Massachusetts Institute of Technology, 2020.