

Classification of Rock Core Sensing Images Using Convolutional Neural Network Methods

Jaehong Hwang and Jaehyuk Lee*

Korea Institute of Geoscience and Mineral Resources (KIGAM),
124 Gwahakro, Yuseong-gu, Daejeon, 34132, Republic of Korea

(Received May 16, 2022; accepted August 16, 2022; online published September 26, 2022)

Keywords: borehole core, deep learning, convolutional neural network, rock classification

The development of underground spaces such as tunnels, subways, logistics warehouses, and complex facilities is continuing. However, owing to poor planning and reckless expansion, there has also been an increase in underground accidents. As such, it is important to obtain accurate geotechnical data on underground spaces for optimal construction outcomes and to ensure the safety of workers. Borehole cores contain essential geological information towards achieving these ends; however, rock classification using borehole cores takes a long time and the classification depends on the interpreter. To address these issues, we performed rock classification based on borehole sensing images using a convolutional neural network (CNN) combined with deep learning techniques. The data used for the training were collected from images of borehole cores in Hang-dong, Guro-gu, Seoul, and Hyeol-dong, Taebaek, Republic of Korea. We used the collected two datasets: a rod dataset labeled by the rock type of the borehole core rod unit and a grid dataset labeled by the rock type unit. The rock types were classified into basalt, gneiss, limestone, mudstone, and shale. In addition, mixed-rock and loss classes were added to the classifications. For the image classification process, we proposed three methods: general deep-learning-based image classification, multiregion image classification, and multiregion image classification using a scoring process. An experiment was conducted to validate these methods. A maximum accuracy of 99.02% was achieved in the validation process. The proposed methods introduced here are expected to reduce the time and costs associated with creating geotechnical databases.

1. Introduction

Recently, the development of underground tunnels, subways, logistics warehouses, and complex facilities has expanded markedly in range and scope. However, owing to reckless development, underground accidents are also increasing, as the geotechnical data necessary for planning and safe implementation are lacking. Since a borehole image contains various geological information, it is very useful for obtaining geotechnical information. Various types of methods have been used to collect borehole core data. Research on deep learning technology for obtaining geotechnical information is also being actively conducted.^(1–7)

*Corresponding author: e-mail: bksro1254@kigam.re.kr
<https://doi.org/10.18494/SAM3970>

The geotechnical information currently available has some inaccuracies due to investigator bias; the data are dependent on the investigator and the nature of the borehole survey. In addition, there are issues regarding the utilization of currently available geological borehole data. To address these issues, we apply deep learning technology to borehole sensing information; this technology can collect accurate geotechnical data on borehole constituents via an easy-to-use, accessible framework. This framework is useful for practical applications, such as underground space development, social overhead capital, and construction design and planning. In addition, at the policy level, sustainable and systematic real-time visualization of geological data for underground works is necessary to ensure safety and efficiency. This study was conducted to address the need to improve borehole surveys and the transparency of geotechnical information to aid construction planning and mitigate safety issues.

Various studies on the automatic classification of rocks have been performed. Prior to the development of deep learning, studies using machine learning to classify rocks were performed. These included a study to classify minerals using a multilayer perceptron neural network,⁽⁸⁾ studies to classify and analyze rocks using a support vector machine,^(9,10) and a study to classify limestone using a probabilistic neural network.⁽¹¹⁾ Recently, with the development of graphics processing units, various deep learning technologies have been developed to improve computational power. Therefore, various studies applying deep learning to rock classification have been conducted.^(12–15)

In this study, we use deep learning for the rock classification of borehole core images. Among the deep learning technologies, a convolutional neural network (CNN) uses a convolution filter to extract features such as the color, texture, and shape of an image. Therefore, it shows good performance in image classification, and CNNs are widely used for image classification.⁽¹⁶⁾ Currently, various CNN models exist, with different models having different features. In this study, various CNN models were compared to find a model suitable for borehole core image classification.

Deep learning networks generally improve model performance through their deeper layers; however, problems can arise, one of the most common being overfitting. When overfitting occurs, the details of the training data are memorized such that the accuracy tends to be high, but the accuracy decreases for the actual test data. This issue has been addressed using the GoogLeNet (Inception-V1) model, which has a network with 22 layers based on the network-in-network method.^(17,18) GoogLeNet uses sparse connections to increase the depth of the network while also effectively reducing the number of dimensions. The sparse connections connect highly correlated layers and control the number of channels by calculating the correlations between them. Also, an inception module is used to reduce dimensionality by applying multiple 1×1 convolutions to one input. After that, the operation is performed in parallel and then concatenated. GoogLeNet has 12 times fewer parameters than AlexNet but still shows higher performance.⁽¹⁹⁾

Gradient vanishing is a problem that can occur when deep learning layers are configured for a network. When gradient vanishing occurs, the weights of the input layer are not updated; thus, the optimal parameters cannot be found. A residual network (ResNet) uses a shortcut connection to solve this problem.⁽²⁰⁾ In general, the value of the input layer is changed as it passes through

the layer. The shortcut connection maintains the weight of the input layer by adding its value to that of the output layer. Therefore, gradient vanishing does not occur even if the network layer is deep. In addition, since the addition operation is used, the learning time does not increase significantly. However, even if a shortcut connection is applied, the amount of computation increases if the layer is deep. In ResNet, 1×1 convolution is applied to the shortcut connection to adjust the dimensions and reduce the amount of computation required to solve the problem.

The Xception module is similar to the existing inception module, with some modifications; its name refers to an “extreme” version of the inception module.⁽²¹⁾ In the inception module, a 1×1 convolution is used to perform cross-channel correlations, and then $N \times N$ convolutions are used to perform spatial correlations. However, the Xception module calculates cross-channel and spatial correlations independently. The Xception module is a variant of depthwise separable convolution, which adjusts the number of channels via a 1×1 convolution after performing depthwise convolution for each channel. The Xception module divides the number of channels into N segments by performing a 1×1 convolution, where N is a hyperparameter. For each segment divided in this way, a 3×3 convolution is performed. By this method, cross-channel and spatial correlations can be calculated independently. Also, Xception does not use a nonlinear function because information loss occurs when using the ReLU nonlinear function. The Xception module reduces the amount of computation and achieves higher accuracy than the existing inception module.

InceptionResNet-V2 is the second version of InceptionResNet-V1 in the inception model series.^(22,23) InceptionResNet uses InceptionResNet modules that combine the shortcut connection of ResNet with the inception module. Inception-V4, which has an improved inception module, has high accuracy but also a high learning curve. InceptionResNet-V2 shows slight differences from Inception-V4 in terms of accuracy; however, the learning rate is faster.

In consideration of the features of the deep learning models above, in this study GoogLeNet, ResNet, Xception, and InceptionResNet-V2 were selected as representative models. In addition, image processing was used to improve the deep learning performance.

In this study, CNN techniques were used to analyze borehole core images. As a first step, red–green–blue (RGB) images of borehole cores in Hang-dong, Guro-gu, Seoul, and Hyeoldong, Taebaek, Republic of Korea, were collected. The data were then configured as a “borehole core box” including various rock types. Individual rocks were classified and labeled by dividing the samples into rod units, each unit representing one borehole core. The labeled data were then used to construct a pre-trained deep learning model. Image processing technology was applied to improve performance and for comparative analysis. The trained model was evaluated using fourfold cross-validation, a confusion matrix, and various evaluation metrics. The main contributions of this work are as follows.

- A method of constructing a structured training dataset is proposed.
- The rock in borehole images is classified using various methods (deep learning and image processing).
- Multiple classification of mixed rocks in borehole images
- A comparative analysis of deep learning networks and various methods is performed.

The proposed method can reduce the time and cost required to obtain geotechnical information.

2. Data

2.1 Data collection

Borehole core images were acquired from the Geoscience Data Repository (GDR) of Korea Institute of Geoscience and Mineral Resources (KIGAM).⁽²⁴⁾ Original images were collected from boreholes in Hang-dong, Guro-gu, Seoul, and Hyeol-dong, Taebaek, Republic of Korea. The borehole cores were composed of hard rock strata with a depth of 10.5–502.1 m, excluding the 0.0–10.5-m-deep laminated and buried layers. The top parts were covered with sedimentary rocks and buried layers (gravel mixed with sand, boulders) up to 10.5 m below the surface, and hard rock appeared at the bottom of borehole core images. Most of the rocks were in good condition but were partially in clay form due to erosion. Moreover, closely spaced cracks had developed on the surface of the cores. The rocks were mainly basalt, gneiss, limestone, mudstone, and shale.

The borehole core images were in JPG format. A total of 120 RGB borehole core images were acquired. Specifically, there were 103 images of the TB-19A borehole (depth, 10.5–502.1 m), eight images of the BH-08 borehole (surface, ~60.0 m), and nine images of the BH-13 borehole (surface, ~63.2 m).

2.2 Data preprocessing

Owing to the geological characteristics of Korea, many types of rock are mixed. Given that a borehole contains a mixture of rock types, a process for labeling and classifying each rock to obtain training data was required. In this study, training data were constructed by first creating a diagram of the data structure, as shown in Fig. 1.

First, rod units, each representing one borehole core in the borehole core box image, were distinguished. The pixel size of the cropped images was 3679×187 ; severely damaged parts were excluded. The cropped images were classified on the basis of the rocks in the borehole core; these included basalt, gneiss, limestone, mudstone, and shale. In the case of borehole cores comprising more than one rock type, the labeling was based on the predominant rock type. An example of a rod-labeled dataset is shown in Fig. 2(a).

Data Construction

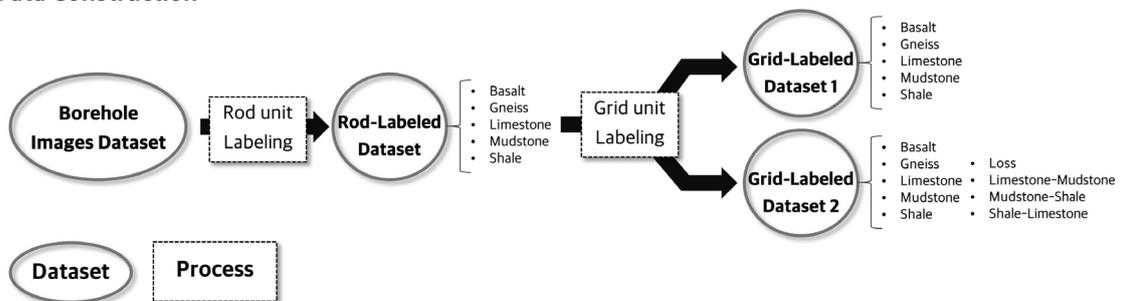


Fig. 1. Proposed data construction method.

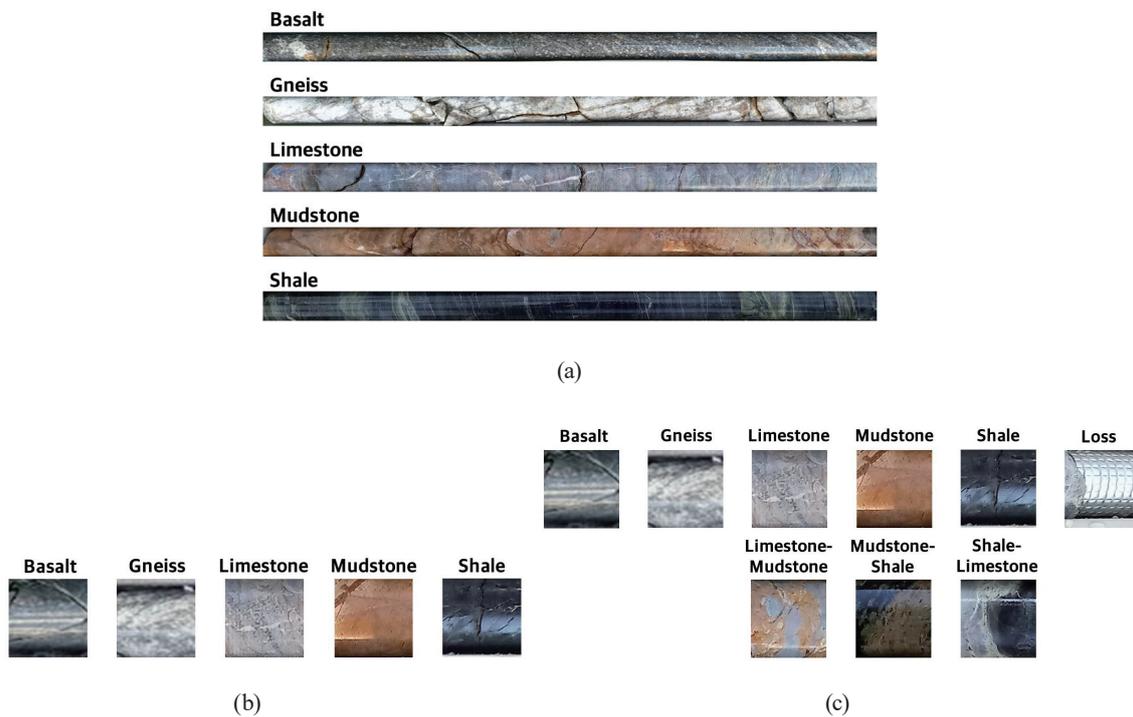


Fig. 2. (Color online) Examples of datasets: (a) rod-labeled dataset, (b) grid-labeled dataset 1, and (c) grid-labeled dataset 2.

Second, specific pieces of rock were demarcated by grid units; the pixel size of the cropped image was 187×187 . The cropped image also comprised five rock types (basalt, gneiss, limestone, mudstone, and shale). Grid-labeled dataset 1 is shown as an example in Fig. 2(b).

To solve the problem of mixed rock, mixed rock classes were added (e.g., limestone–mudstone, mudstone–shale, and shale–limestone). In addition, the borehole core box was not in an intact state; it was fractured due to external forces generated during the borehole process. A filler was used to prevent further damage, and a “loss” class was added for the core. Grid-labeled dataset 2 consisted of nine classes: basalt, gneiss, limestone, limestone–mudstone, mudstone, mudstone–shale, shale, shale–limestone, and loss; an example of a dataset is shown in Fig. 2(c). The rod-labeled dataset, grid-labeled dataset 1, and grid-labeled dataset 2 contained 205, 6692, and 6692 images, respectively.

3. System Development

3.1 Proposed system

The purpose of this study was to develop a borehole core rock classification system based on a CNN deep learning model. To construct and train the deep learning model and for transfer learning, pre-trained deep learning models (GoogLeNet, ResNet, Xception, and InceptionResNet-V2) were used. For transfer learning, the rod-labeled dataset, grid-labeled

dataset 1, and grid-labeled dataset 2 were used. Each dataset was divided into three parts for testing, with each part corresponding to a specific method (Methods 1–3). Method 1 classified borehole core rocks through general image classification. Method 2 classified borehole core rocks through multiregion image classification based on the borehole core images. Method 3 was a modified and improved version of Method 2. A flowchart for each method is shown in Fig. 3; a detailed description is given in Sect. 3.2.

During the model training process, hyperparameters of the learned elements should be set. In this study, the hyperparameters were determined using GoogLeNet, a deep learning model with a relatively fast learning rate. Fourfold cross-validation was used; the training dataset was divided into four parts with alternation between the training and test datasets for training on all datasets. In this manner, generalized results could be obtained without overfitting.

For a deep learning model, a loss function is used to evaluate performance during training. In general, models perform best when the loss function is minimized. Therefore, it is necessary to

System Design

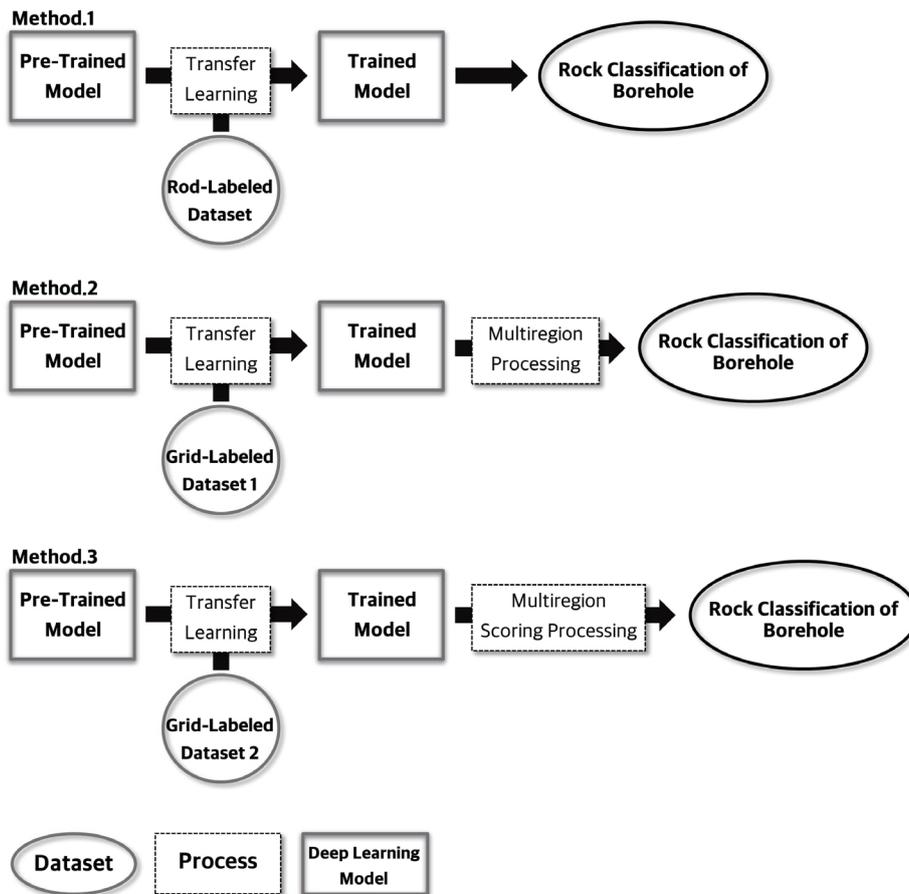


Fig. 3. Steps of the proposed methods for rock classification of the borehole core.

determine and optimize a weight parameter that minimizes the loss function. In this study, the Adam optimizer was used for optimization.⁽²⁵⁾ The number of learning cycles was set to 50 or 64 depending on the epoch and batch size. The initial learning rate was 0.1% with the Adam optimizer, which was decreased by 10% every 10 epochs. An NVIDIA TITAN RTX graphics processing unit was used for the training (NVIDIA, Santa Clara, CA, USA).

3.2 Application of proposed methods

Method 1 classifies borehole core rocks through general image classification. First, the rod-labeled dataset consisting of five rock types (basalt, gneiss, limestone, mudstone, and shale) was input into the pre-trained deep learning model. The transfer-learning-trained deep learning model classified the borehole core rocks.

Method 2 classifies borehole core rocks via multiregion processing of borehole core images. Grid-labeled dataset 1 was input into the pre-trained deep learning model. Grid-labeled dataset 1 included five rock types (basalt, gneiss, limestone, mudstone, and shale). The multiregion process divided the borehole core into multiple images to identify the dominant rock types.

Method 3 is an improved version of Method 2. Grid-labeled dataset 2 was input into the pre-trained deep learning model. This dataset included nine rock types (basalt, gneiss, limestone, limestone–mudstone, mudstone, mudstone–shale, shale, shale–limestone, and loss). The borehole core was divided into multiple images, as per Method 2. We used the divided images to calculate the prediction accuracy through deep learning. The prediction accuracy was substituted into Eq. (1) to obtain $Score_{class}$.

$$Score_{class} = \begin{cases} Acc_{Limestone} + 0.5Acc_{Limestone-Mudstone} & \text{if class = Limestone} \\ Acc_{Mudstone} + 0.5Acc_{Limestone-Mudstone} & \text{if class = Mudstone} \\ Acc_{Shale} + 0.5Acc_{Mudstone-Shale} + 0.5Acc_{Shale-Limestone} & \text{if class = Shale} \\ 0 & \text{if class = Loss} \\ Acc_{class} & \text{otherwise} \end{cases} \quad (1)$$

Here, Acc_{class} is the prediction accuracy for the variable class. Each $Score_{class}$ obtained through Eq. (1) is substituted into Eq. (2) to obtain $Total Score_{class}$.

$$Total Score_{class} = \sum_{k=1}^n Score_{class,k} \quad (2)$$

Here, n is the number of images cut through Method 2. The maximum value among $Total Score_{class}$ obtained through Eq. (2) is finally predicted. With the above method, the mixed area and loss area are reflected in the results. An example of the application of Method 3 is shown in Fig. 4. The results were evaluated through validation testing, as discussed in Sect. 4.

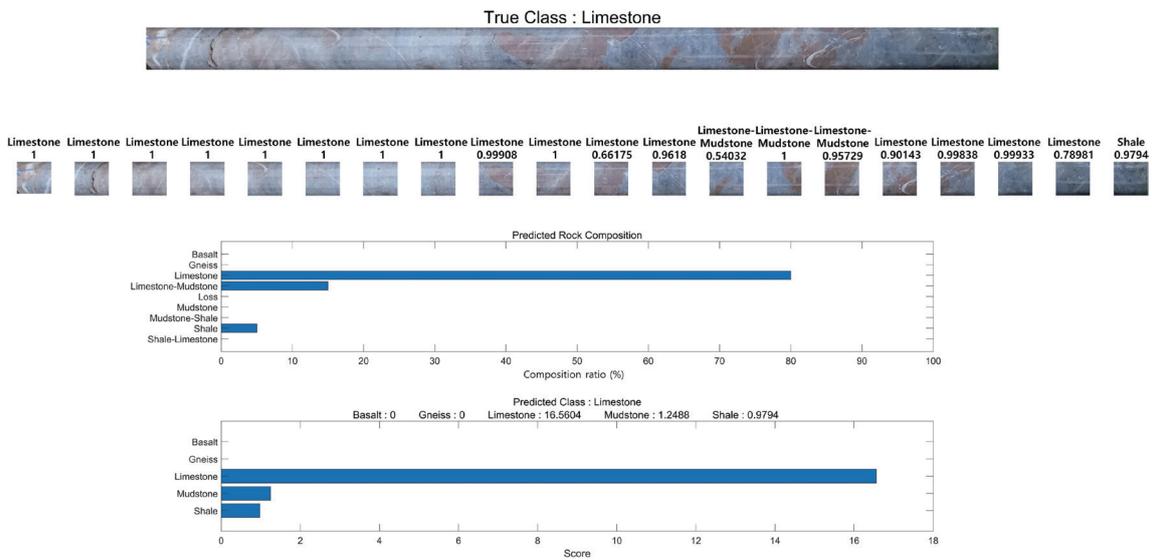


Fig. 4. (Color online) Example of application of Method 3.

4. Results

4.1 Evaluation method

To evaluate the results of the deep-learning-based rock classification system, a confusion matrix and evaluation metrics were used. The classification evaluation metrics in this study were accuracy, recall, precision, and F1 score.

Accuracy rates can be calculated to evaluate the performance of classification models. Here, accuracy was determined as the ratio between true positives and true negatives in the error matrix. However, when the number of data points is unbalanced among classes, the classification model becomes biased. Therefore, accuracy should be used in conjunction with other evaluation metrics. Precision and recall are complementary and inversely related evaluation metrics. Precision is given by the ratio between true positives and false positives and reflects the actual values. Recall is the ratio between true positives and false negatives and reflects the modeled values. However, since both indicators have different strengths and weaknesses, evaluations of model performance are biased if only one of the indicators is used. Therefore, we evaluated the model performance using the F1 score, which is the harmonic average of precision and recall.

4.2 Validation of Method 1

Method 1 classifies the borehole core rocks through general image classification. The rod-labeled dataset was used for model training and validation. Fourfold cross-validation was also performed. The confusion matrices are displayed in Fig. 5, and the classification evaluation metrics are listed in Table 1.

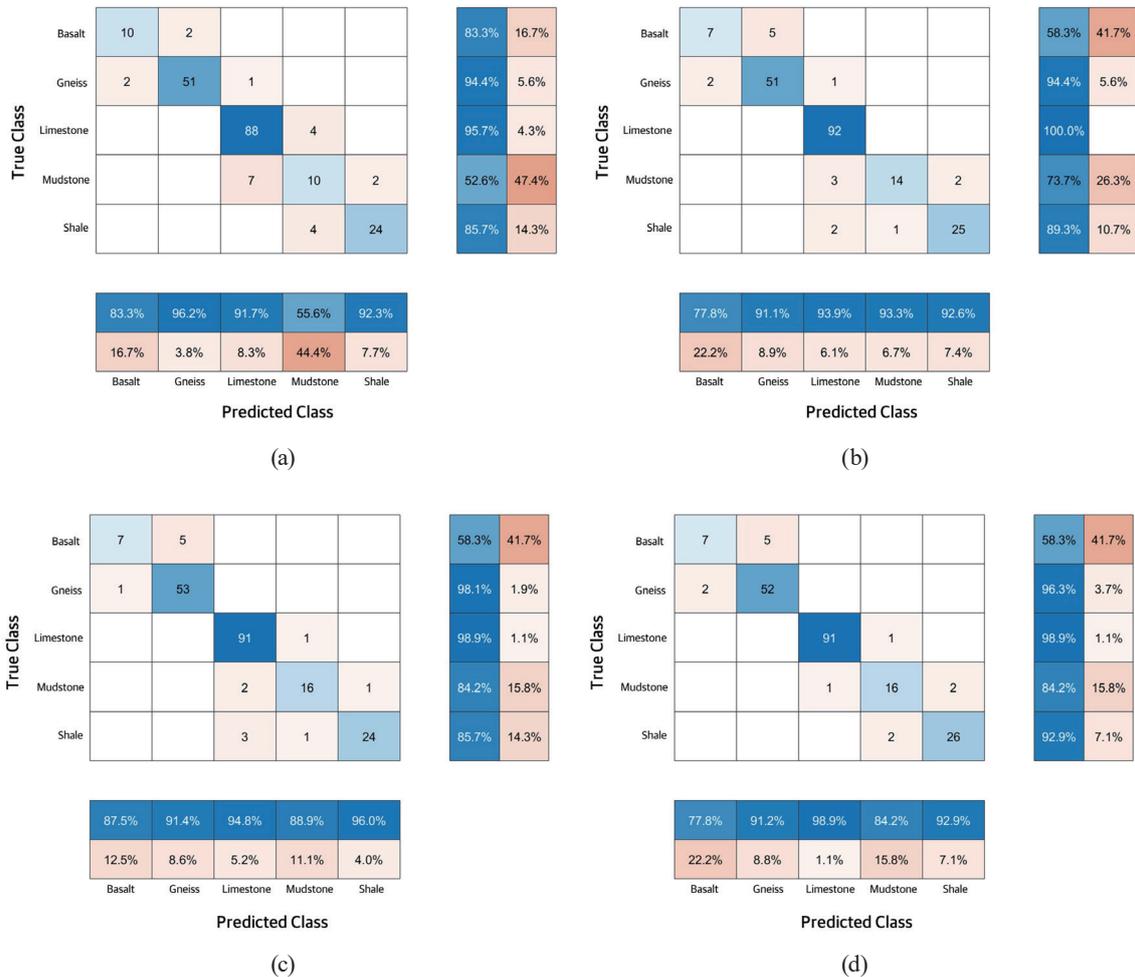


Fig. 5. (Color online) Confusion matrices for Method 1: (a) GoogLeNet, (b) ResNet, (c) Xception, and (d) InceptionResNet-V2.

Table 1
Classification evaluation metrics of Method 1.

	GoogLeNet	ResNet	Xception	InceptionResNet-V2
Accuracy	0.8927	0.9220	0.9366	0.9317
Recall	0.8236	0.8315	0.8612	0.8506
Precision	0.8382	0.8973	0.8900	0.9171
F1 score	0.8308	0.8631	0.8754	0.8826

The validation test revealed an average accuracy of 92.08% for Method 1. Xception had the highest accuracy at 93.66%. However, the F1 score of InceptionResNet-V2 was 88.26%, which was 0.72% higher than that of Xception.

4.3 Validation of Method 2

Method 2 classifies borehole core rocks via multiregion classification of borehole core images. Grid-labeled dataset 1 was used for training, and the rod-labeled dataset was used for validation. Fourfold cross-validation was also performed. The classification evaluation metrics of Method 2 are listed in Table 2.

The validation results showed that the average accuracy of Method 2 was 97.08% (an increase of 5% compared with Method 1). There was no significant difference in accuracy among the deep learning models. GoogLeNet showed the highest accuracy at 97.56%.

4.4 Validation of Method 3

Method 3 is a modified version of Method 2, as stated above. Grid-labeled dataset 2 was used for model training, and the rod-labeled dataset was used for model validation. Fourfold cross-validation was also performed. The classification evaluation metrics of Method 3 are listed in Table 3.

The average accuracy of Method 3 was 98.66%, which was 1.58% higher than that of Method 2. There was no significant difference in accuracy among the deep learning models. GoogLeNet and InceptionResNet-V2 showed the highest accuracy at 99.02%.

4.5 Discussion

The performances of the methods proposed in this study can be seen in Table 4. Methods 2 and 3 showed higher accuracy than Method 1, which is a general image classification method. The higher accuracy is expected to be due to the features of the borehole core image. The borehole core has an axially elongated shape. However, to use the pre-trained deep learning

Table 2
Classification evaluation metrics of Method 2.

	GoogLeNet-based Method 2	ResNet-based Method 2	Xception-based Method 2	InceptionResNet-V2- based Method 2
Accuracy	0.9756	0.9707	0.9659	0.9707
Recall	0.9845	0.9824	0.9716	0.9824
Precision	0.9410	0.9327	0.9292	0.9327
F1 score	0.9623	0.9589	0.9500	0.9589

Table 3
Classification evaluation metrics of Method 3.

	GoogLeNet-based Method 3	ResNet-based Method 3	Xception-based Method 3	InceptionResNet-V2- based Method 3
Accuracy	0.9902	0.9854	0.9805	0.9902
Recall	0.9941	0.9904	0.9882	0.9941
Precision	0.9746	0.9614	0.9524	0.9746
F1 score	0.9843	0.9757	0.9700	0.9843

Table 4
Classification evaluation metrics of methods.

	Method 1	Method 2	Method 3
Accuracy	0.9366	0.9756	0.9902
Recall	0.8612	0.9845	0.9941
Precision	0.8900	0.9410	0.9746
F1 score	0.8754	0.9623	0.9843

models, they are transformed into squares through image preprocessing. Therefore, it is expected that the features of the borehole core are damaged and the accuracy is lowered. Methods 2 and 3 can reflect all the features of the borehole core, so the accuracy is expected to increase.

It can be seen that Method 3 has 1.46% higher accuracy than Method 2. Method 3 has two advantages over Method 2. First, it is possible to reduce the weight of a region with low accuracy. In general, a part with low prediction accuracy has a high probability of error. Therefore, it was possible to improve the final classification accuracy by reducing the proportion of the parts with low accuracy. Second, mixed rock can be predicted. In Method 2, the mixed rock was predicted to be a rock class with a high accuracy. In Method 3, the class of the mixed part was additionally learned, and the score of the mixed part could be calculated. Therefore, the final classification accuracy was improved.

In this study, the types of rock in the borehole core were classified with an accuracy of up to 99.02%. The main cause of error is impurities in the borehole core. The borehole core is covered with mud during the drilling process. In this case, it is possible to classify rocks when they are identified by humans. However, when making predictions using deep learning, the color of the surface of the borehole core changes, and the type of rock is recognized incorrectly. Therefore, the prediction accuracy is lowered. It is expected that the prediction accuracy can be improved if the training data of the part with impurities is increased.

The methods proposed in this study do not require additional equipment except for a camera. Therefore, they can be applied quickly in the field. In addition, the methods showed high prediction accuracy even for mixed rocks. The limitation of this study is that the deep learning model learns incorrectly due to the color change of the surface of the drilling core. Therefore, the prediction accuracy is lowered. Moreover, if the borehole core is severely damaged, the error rate increases owing to the characteristic of the CNN. It is expected that the above shortcomings will be resolved if additional geological characteristics are studied.

5. Conclusions

In this study, three methods were proposed for the rock classification of borehole core images using a deep learning method. Training data were obtained by processing the original borehole core box images. First, the training data were divided into a rod unit and grid unit in accordance with the image shape. The training image was labeled in these two units; each unit cropped image was classified on the basis of the type of rock

To validate the proposed methods, an experiment was performed using training data, fourfold cross-validation, a confusion matrix, and classification evaluation metrics. The performance

indicators were accuracy, recall, precision, and F1 score. The maximum accuracies of Methods 1–3 were 93.66, 97.56, and 99.02%, respectively. The conclusions of this study are as follows.

- The rock classification of the borehole core methods proposed in this study showed up to 99.02% accuracy.
- Our system can be expected to reduce the time and cost required to obtain geotechnical information.

This study had several limitations. First, the image data of a limited area were used for model training. However, the proposed system should be generally applicable if data from other regions are included during the training process. The use of additional metrics such as the strength, particle size, and humidity of borehole cores should further improve the classification accuracy. Second, rocks were classified only by general type for efficiency. Through semantic segmentation in future studies, the constituent components of borehole core rocks should be classifiable.

Acknowledgments

This research was supported by a Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 22DCRU-B158151-03) and a Basic Research Project (22-3118) of Korea Institute of Geoscience and Mineral Resources (KIGAM) funded by the Ministry of Science, ICT and Future Planning of Korea.

References

- 1 K. K. Bradbury, J. P. Evans, J. S. Chester, F. M. Chester, and D. L. Kirschner: *Earth Planet. Sci. Lett.* **310** (2011) 131. <https://doi.org/10.1016/j.epsl.2011.07.020>
- 2 A. Azimian: *Rock Mech. Rock Eng.* **49** (2016) 1559. <https://doi.org/10.1007/s00603-015-0789-8>
- 3 F. Tian, X. Lu, S. Zheng, H. Zhang, Y. Rong, D. Yang, and N. Liu: *Open Geosci.* **9** (2017) 266. <https://doi.org/10.1515/geo-2017-0022>
- 4 A. R. Kim, D. Kim, Y. S. Byun, and S. W. Lee: *J. Korea Geotech. Sci.* **34** (2018) 145. <https://doi.org/10.7843/kgs.2018.34.12.145>
- 5 H. Lin, W. H. Kang, J. Oh, I. Canbulat, and B. Hebblewhite: *Int. J. Min. Sci. Technol.* **30** (2020) 623. <https://doi.org/10.1016/j.ijmst.2020.05.019>
- 6 M. Shahriari, D. Pardo, A. Picón, A. Galdran, J. Del Ser, and C. Torres-Verdín: *Comput. Geosci.* **24** (2020) 971. <https://doi.org/10.1007/s10596-019-09859-y>
- 7 W. Zhang, H. Li, Y. Li, H. Liu, Y. Chen, and X. Ding: *Artif. Intell. Rev.* **54** (2021) 5633. <https://doi.org/10.1007/s10462-021-09967-1>
- 8 N. A. Baykam and N. Yilmaz: *Comput. Geosci.* **36** (2010) 91. <https://doi.org/10.1016/j.cageo.2009.04.009>
- 9 C. A. Perez, J. A. Saravia, C. F. Navarro, D. A. Schulz, C. M. Aravena, and F. J. Galdames: *Int. J. Miner. Process.* **144** (2015) 56. <https://doi.org/10.1016/j.minpro.2015.09.015>
- 10 C. L. Bérubé, G. R. Olivo, M. Chouteau, S. Perrouy, P. Shamsipour, R. J. Enkin, and R. Thiémondge: *Ore Geol. Rev.* **96** (2018) 130. <https://doi.org/10.1016/j.oregeorev.2018.04.011>
- 11 A. K. Patel and S. Chatterjee: *Geosci. Front.* **7** (2016) 53. <https://doi.org/10.1016/j.gsf.2014.10.005>
- 12 P. Y. Zhang, J. M. Sun, Y. J. Jiang, and J. S. Gao: 79th EAGE Conf. Exhibition 2017 (EAGE, 2017) 1–5.
- 13 X. Ran, L. Xue, Y. Zhang, Z. Liu, X. Sang, and J. He: *Mathematics* **7** (2019) 755. <https://doi.org/10.3390/math7080755>
- 14 X. Zou, C. Wang, H. Zhang, and S. Chen: *Appl. Sci.* **11** (2021) 10490. <https://doi.org/10.3390/app112110490>
- 15 X. Zou, H. Song, and C. Wang: *Rock Mech. Rock Eng.* **54** (2021) 5945. <https://doi.org/10.1007/s00603-021-02588-8>

- 16 W. Rawat and Z. Wang: *Neural Comput.* **29** (2017) 2352. https://doi.org/10.1162/neco_a_00990
- 17 M. Lin, Q. Chen, and S. Yan: *arXiv* (2013). <https://doi.org/10.48550/arXiv.1312.4400>
- 18 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and A. Rabinovich: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2015)* 1–9.
- 19 A. Krizhevsky, I. Sutskever, and G. E. Hinton: *Adv. Neural Inf. Process. Syst.* **60** (2017) 84. <https://doi.org/10.1145/3065386>
- 20 K. He, X. Zhang, S. Ren, and J. Sun: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016)* 770–778.
- 21 F. Chollet: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2017)* 1251–1258.
- 22 C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016)* 2818–2826.
- 23 C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi: *Proc. AAAI Conf. Artificial Intelligence (AAAI)* **31** (2017).
- 24 Geoscience Data Repository of KIGAM: Development of Underground Information Construction and Analysis Technology based on Artificial Intelligence through Machine Learning, <https://gdr.kigam.re.kr/gdr/> (accessed 8 September 2021).
- 25 D. P. Kingma and J. Ba: *arXiv* (2014). <https://doi.org/10.48550/arXiv.1412.6980>
- 26 D. L. Olson and D. Delen: *Advanced Data Mining Techniques* (Springer Science & Business Media, 2008) 1st ed., p. 138.