The Application of Machine Learning Algorithms on Triaxial Passive Seismic Data to Identify The Geological Location of The Signal Source

Muhammad Randy Azhari[1], Bagus Mahendro Wibowo Adhi[1], Evi Fazriati[1], and Yudi Rosandi[1*]

[1] Department of Geophysics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Jatinangor, Indonesia
rosandi@geophys.unpad.ac.id

Abstract. The characteristics of ground vibration are determined by the local geological and physical conditions of the Earth. Such vibrations can be detected using the passive seismic measurement. This research aims to create an advanced signal processing program to classify the local characteristic of ground vibration signals, applying the machine learning techniques, specifically the Convolutional Neural Network. The research process involved data acquisition, data preprocessing, model creation, model training, and model testing. Data acquisition was performed using a triaxial seismometer. The acquired data was converted into image representations in the form of axial spectrograms. The training data was divided into three directional components. The training process consists of two steps namely the component identification step and classification step. For the component identification we obtained accuracy of 87.8%. Whereas, for the classification step we obtain 90.8% using the horizontal model and 95.3% using the vertical model. Based on the confusion matrix evaluation, the model achieved an accuracy of over 85% in the overall classification. Furthermore, in the testing process correct classifications that matched the labels in all experiments was achieved. This work demonstrates the capability of classifying the local characteristic of ground vibration signals by means of the Convolutional Neural Network algorithm.

Keywords: CNN, Geological Classification, Machine Learning, Microtremor

Introduction

The local physical and geological conditions of every region have distinct characteristics, which is reflected in the ground vibrations patterns. These vibrations are mechanical waves that propagate through the soil, caused by natural sources or human activities. The propagation speed in a medium is determined by several factors, i.e. the source frequency, the material characteristics, and the environmental conditions [1]. Every medium has natural frequencies and harmonics, due to material characteristics such as molecular density and porosity. Ground vibration signals exhibit different spectral sig-

F. Khoerunnisa et al. (eds.), *Proceedings of the 9th Mathematics, Science, and Computer Science Education International Seminar (MSCEIS 2023)*, Advances in Social Science, Education and Humanities Research 860,
https://doi.org/10.2991/978-2-38476-283-5_10

nal patterns depending on the geological conditions [2]. Ground vibrations carry information through the frequencies they contain, and this information can be used to assess geological conditions [3].

Ground vibrations can be recorded using passive seismic methods. Passive seismic methods are also known as microtremor methods because they detect very small vibrations. Rock formations are important parameters that influence the ground vibration acceleration values in a specific region [4]. Research using geophone sensors to detect differences in subsurface materials [5]. This research revealed that variations in materials can be identified through the soil medium, which affects wave propagation velocity. Research to classify and characterize soil that could potentially trigger landslides, a natural disaster [6]. To understand the physical and geological characteristics of a region, it is necessary to employ methods capable of identification and classification. Therefore, studies utilizing machine learning methods are required to obtain efficiency and high accurate results. One of the popular machine learning methods for object identification and image classification is the Convolutional Neural Network [7].

Ground vibration signal data can be highly complex, containing intricate patterns and variations that may not be easily discernible to the human eye or conventional methods [8]. The CNN method is excellent at capturing complex patterns within data, making it suitable for analyzing complex signals such as ground vibrations [9]. CNN is adept at automatically extracting relevant features from raw data. In the context of ground vibration signals, CNN can discern subtle variations and extract features that are crucial for understanding the characteristics of the signals [10].

The Convolutional Neural Network (CNN) is a deep learning algorithm designed for processing two-dimensional data, such as sound or images. CNN can be employed for classifying labeled data using supervised learning methods, where training data and target variables are available, aiming to categorize data into existing data labels. Research to identify plant species using the CNN method, achieving an accuracy rate of 90% [11]. Another study focused on identifying fruit types using the CNN method with the MobileNetV2 architecture, achieving accuracy levels of up to 99% [12]. The CNN method is still rarely applied to cases involving ground vibration signal data. Therefore, there is a need for a study to apply machine learning methods using ground vibration signal data.

This research focuses on three locations, two in campus experiment fields (Lapangan Merah and the arboretum, Universitas Padjadjaran) and a landslide prone area at East part of Bandung basin (Desa Cihanjuang, Cimanggung, Sumedang, West Java). To ease the discussion the locations are denoted as L1, L2, and L3, respectively. They were selected due to the varying environmental conditions. L1 is on a plateau, L2 is situated in a valley, L3 lies inside a landslide prone area. We assumed that the selection would reveal distinct characteristics of ground vibration signals. Data acquisition was performed using a microtremor method with a triaxial seismometer, followed by data processing that is essential to eliminate noise and transform the raw data into spectrograms. This is a necessary step in the machine learning method to identify ground vibration signals using CNN. The spectrograms are put in an image format. The program was written in Python programming language, based on the pre-trained CNN model. This

research aims to develop a program that can identify geological locations from the ground vibration signals.

2 Methodology

This research is generally conducted in five stages: data acquisition, data preprocessing, model creation, model training, and model testing. Data acquisition was carried out at three different locations with distinctive local physical conditions. Figure 1 shows the research location, these locations were chosen according to different morphological conditions, i.e. a plateau (L1), a valley (L2), and a landslide-prone area (L3). The acquisition of ground vibration signal data was conducted using the microtremor method with a triaxial seismometer developed in the laboratory (SeismoLog devices). The used geophone type was the ST 4.5 N with a natural frequency of 4.5 Hz. The seismometer possesses a sensitivity of approximately 0.28 V/cm/s. Measurements are carried out 50 times sparsely distributed at each location with sampling time of 10 minutes.

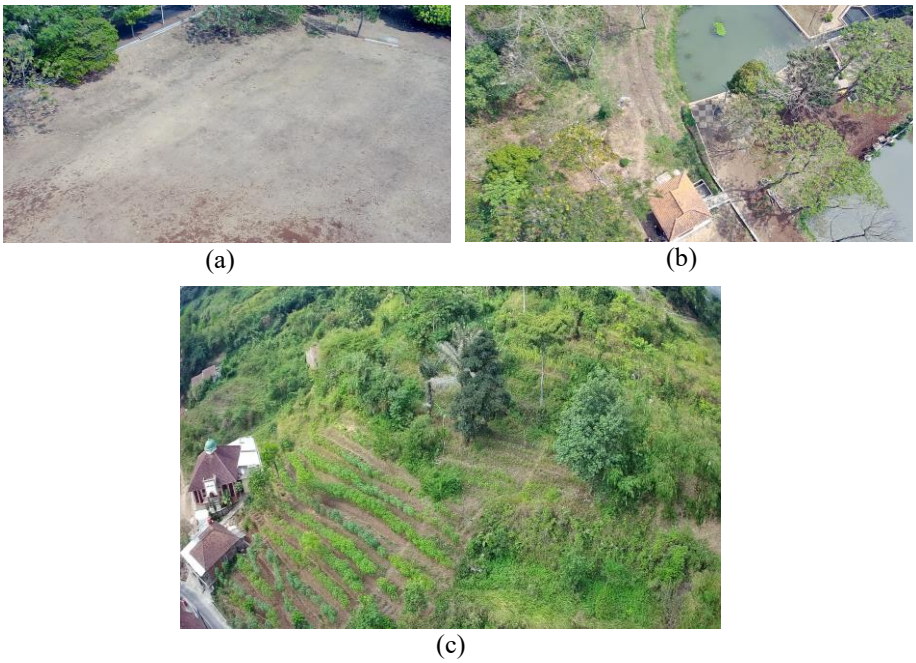


Fig. 1. The physical conditions of the research locations. (a) A plateau (L1), (b) A valley (L2), and (c) A landslide prone area (L3).

The data acquired using the SeismoLog device comprises three components that are mutually perpendicular. These components consist of two horizontal elements, X (east-west) and Y (north-south), as well as one vertical component, Z (up-down). To facilitate data transfer and processing, the acquisition data was stored in the JSON file format. JSON file format is easily readable and accessible by Python scripts due to its open

source [13]. The acquired data includes various measurement parameters, such as the data acquisition time, channel-00 signal data (horizontal component X), channel-01 signal data (horizontal component Y), and channel-02 signal data (vertical component Z). Table 1 illustrates the data format (.JSON) utilized by the SeismoLog instrument.

Table 1. The recorded data result (.JSON) in the acquisition process.

Field	Number of Data/Second	Data Type
Time	594	array
Channel-00 (X)	594	array
Channel-01 (Y)	594	array
Channel-02 (Z)	594	array
config	-	JSON Object

In this research, the identification of geological conditions was conducted using a machine learning method known as Convolutional Neural Network (CNN). This method was chosen for its effectiveness in extracting features from spatial data. Data fed into the CNN method is typically in the form of digital images. Therefore, a data preprocessing process is needed to transform the raw data, which initially comes in JSON format, into images in the form of spectrograms.

Data related to ground vibrations must also pass through several stages in signal preparation to eliminate interference signals and adapt the dataset to the required parameters. This process was carried out using the Python programming language with the assistance of commonly used signal processing libraries, namely Numerical Python (NumPy) and Scientific Python (SciPy). The preprocessing was divided into four stages: filtering, acquisition data sampling, spectrogram creation, and data augmentation.

The filtering stage aims to eliminate unwanted noise, such as vehicle activity and tree roots. The filtering method used was the Butterworth low-pass filter. Research that demonstrating that the Butterworth filter produces an almost flat passband amplitude response without ripples, making it relatively superior compared to other filters, such as Chebyshev and Elliptic [14]. This filter allows low-frequency signals to pass through with a cutoff frequency of 20 Hz. This is because the dominant frequency at the measurement point is less than 20 Hz, so signals above this cutoff frequency are considered as the interference. The number of data points in each JSON file varies, typically around 355,000 data points. This variation is due to differences in the natural frequencies during acquisition. To address this issue, an acquisition data sampling process was carried out. The purpose of this acquisition data sampling process was to prevent data imbalance.

Spectrograms were created using data from the X, Y, and Z components at each location. The spectrograms underwent a data augmentation process to increase the quantity of datasets that will be used during model training. Data augmentation can enhance the accuracy of the trained CNN model because the model gains additional data, enabling it to generalize better [15]. This results in a dataset of spectrograms consisting of 50 for each component at each location. Data augmentation was performed

by dividing the spectrograms into time segments of 300s, 150s, 100s, 75s, and 60s. As a result, each original spectrogram data generated 31 spectrogram data points. In this way from 50 spectrogram data points acquired from every location we obtained 1,550 smaller spectrogram data chunks for each component.

Table 2. The distribution of training and test data for the component, vertical, and horizontal models.

Model	Component		Vertical			Horizontal		
Label	HOR	VER	L1	L2	L3	L1	L2	L3
Training Data	3.720	3.720	1.240	1.240	1.240	2.480	2.480	2.480
Test Data	930	930	310	310	310	620	620	620

In this research, the model was built using the MobileNetV2 architecture, which divides the process into depthwise convolution and pointwise convolution. MobileNetV2 introduces two new features: linear bottlenecks and shortcut connections between bottlenecks. The MobileNetV2 architecture was chosen for its relatively fast training duration and high accuracy [16]. In total, three models are created in this research: the component model, the horizontal model, and the vertical model. The component model was responsible for identifying the components from the input data, while the horizontal and vertical models are responsible for identifying the geological locations from the input data.

Table 2 explains the labels for each model, as well as the training and testing datasets that each model will utilize. The dataset distribution was divided into 80% for training and 20% for testing, selected randomly. Previous study elucidates that the 80%-20% dataset distribution generally yields better accuracy compared to the 60%-40% and 70%-30% distributions [17].

The hyperparameters used in this study encompass the number of epochs, set at 20, a learning rate of 0.001, and a batch size of 128. Hyperparameter tuning focuses on ensuring that the model does not experience underfitting or overfitting to the training dataset, while also learning the data structure [18]. The selection of the number of epochs and learning rate was based on research [19], which achieved an accuracy rate of 98.18%. Furthermore, the models underwent testing using batch sizes of 8, 16, 32, 64, and 128, with the results indicating that a batch size of 128 yielded the highest accuracy and the shortest training duration. The optimization technique employed was Adam. Adam optimization was chosen due to its ability to expedite the training process towards convergence [20]. To gauge the disparity between the values predicted by the model and the actual values in the label data, a loss function in the form of CrossEntropyLoss was employed. The primary objective of model training was to minimize the loss value, enabling the model to furnish precise predictions in line with the actual data.

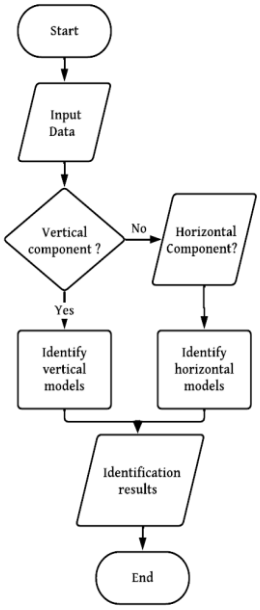


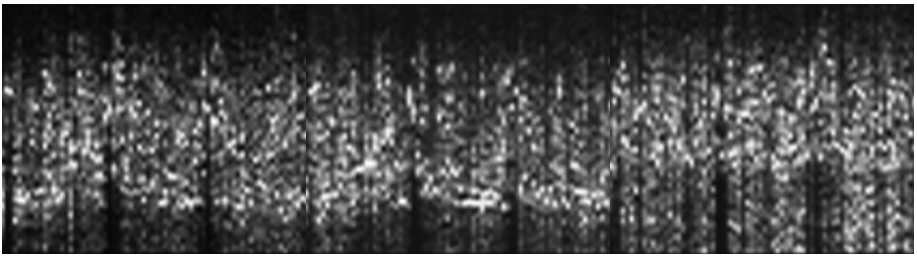
Fig. 2. Flowchart of the identification program for testing the Component Model, Horizontal Model, and Vertical Model.

The model testing was performed by creating a confusion matrix using the testing dataset. This confusion matrix serves to assess the comparison between the model's predictions, resulting from the training process, and the actual data in the labels. The confusion matrix can indicate the model's performance in classifying correct values in the predicted classes. Previous study utilized a confusion matrix to evaluate the performance of the trained model, achieving an accuracy rate of 86.2% [21]. In addition to using the confusion matrix, testing was also conducted by developing a program capable of processing input data into outputs that represent the identification of geological conditions. The algorithm for this program can be seen in Figure 2.

Based on this algorithm, the program will be able to identify whether the input consists of horizontal or vertical component data using the component model. When the input has a horizontal component, the program will proceed with the identification process using the horizontal model to determine the geological location, which could be L1, L2, or L3. However, if the input was a vertical component, the program will continue the identification process using the vertical model to determine the same geological locations: L1, L2, or L3. This program was implemented in the form of a script using the Python programming language with a (.py) format, enabling it to be executed directly from the command line in the console. The data input into the program was in (.JSON) file format because the program was designed to integrate with the Seismo-Log seismometer, allowing input data to be processed to generate component and location predictions.

3 Result and Discussion

The raw data that has undergone preprocessing results in spectrogram data. These spectrograms are generated using three components: X, Y, and Z. This data was then utilized to build the CNN model. Figure 3 shows the spectrograms for each component from location L2. Each spectrogram exhibits distinct characteristic patterns, primarily due to variations in local frequency and magnitude values recorded by the geophone sensors.



(a) (b) (c)

Fig. 3. Spectrogram at location L2. (a) Component X, (b) Component Y, and (c) Component Z

Based on the training results shown in Table 3, it can be observed that the training duration for the component model was 27 minutes and 38 seconds, for the horizontal model was 28 minutes and 59 seconds, and for the vertical model was 13 minutes and 27 seconds. Meanwhile, the accuracy of the component model reaches 87.8%, the horizontal model achieves 90.8%, and the vertical model attains 95.3%. These training results indicate that accuracy is not significantly influenced by the training duration, as the vertical model, with the shortest training duration among the three models, has the highest accuracy.

The differences in training duration and accuracy are attributed to variations in the datasets used to train each model. The component and horizontal models have more diverse input datasets. The component model covers three different locations for each label, while the horizontal model encompasses components X and Y for each label. On the other hand, the vertical model has a more homogeneous dataset since each label consists only of component Z.

Table 3. The training results of the model with MobileNetV2 architecture.

Model	Batch size	epoch	Learning rate	Duration	Accuracy
Component	128	20	0.001	27 minutes 38 seconds	87.84%
Horizontal	128	20	0.001	28 minutes 59 seconds	90.80%
Vertical	128	20	0.001	13 minutes 27 seconds	95.26%

The results of model training can also be analyzed by examining the differences between the loss curve and accuracy curve on the training dataset and the testing dataset. Loss and accuracy curves are two important indicators commonly used to monitor and analyze a model's performance in machine learning. The loss curve illustrates how well or poorly the model manages prediction errors at each training epoch. Meanwhile, the accuracy curve measures how accurately the model can make predictions on the training dataset.

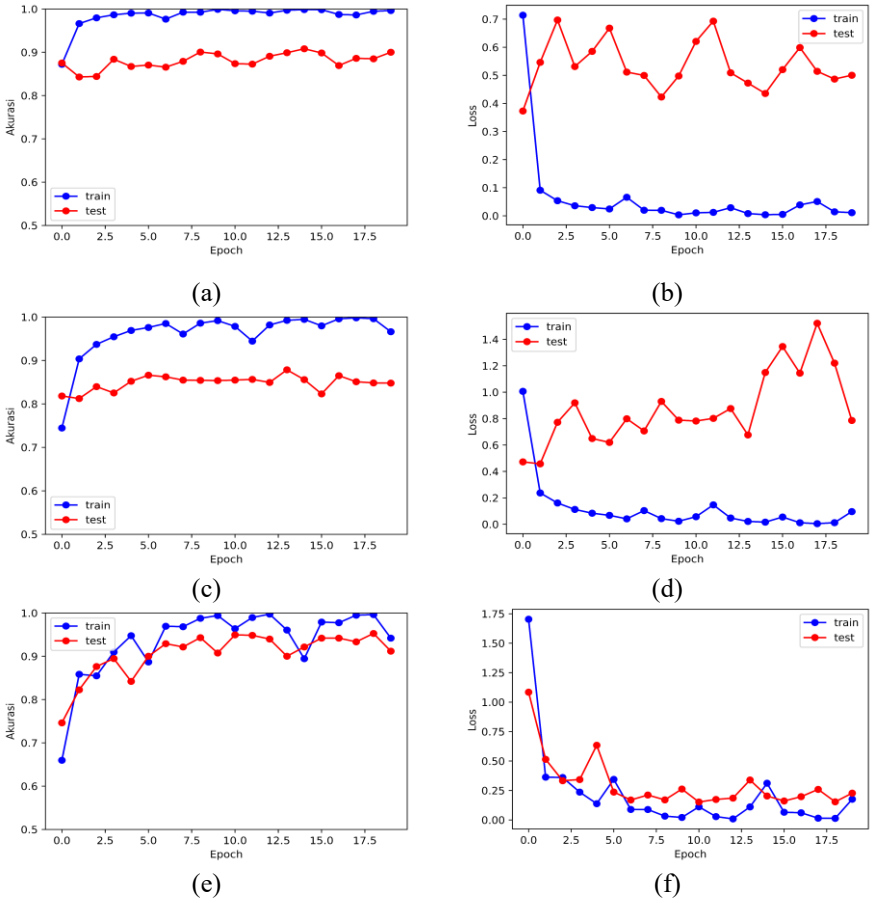


Fig. 4. Results of the loss and accuracy curves on the training and testing data. (a,b) Loss curve and accuracy curve graphs for the component model. (c,d) Loss curve and accuracy curve graphs for the horizontal model. (e,f) Loss curve and accuracy curve graphs for the vertical model.

These loss and accuracy curves are highly significant as they help us understand the behavior of the model throughout the training process. By analyzing these curves, we can determine whether the model is experiencing overfitting (overly adapting to the training data and failing to generalize to the testing data) or underfitting (insufficiently adapting to the training data). Additionally, we can identify when the model reaches a

stable level of performance. The information obtained from the analysis of loss and accuracy curves is valuable for optimizing and improving models during the training phase [22].

The results of the loss and accuracy curves during the training of the component model (Figure 4a, 4b), horizontal model (Figure 4c, 4d), and vertical model (Figure 4e, 4f) generally show that the loss values fluctuate with each epoch, but the accuracy values tend to converge as the epochs progress. Additionally, the loss values for the training dataset are lower than those for the test dataset, and the accuracy values for the training dataset are higher than those for the test dataset.

Based on these results, it can be concluded that all three models are experiencing overfitting. Overfitting occurs when a model becomes overly tailored to the training data, resulting in good performance on the training data but poor performance on unseen test data or data it has not encountered before. This can be observed from the differences between the accuracy and loss values for the training and test datasets. Higher accuracy on the training dataset suggests that the model has "memorized" the training data but fails to generalize to the test data.

To address overfitting, several steps can be taken, including the use of regularization techniques like dropout or the addition of more diverse training data [23]. By doing so, the model can better generalize to previously unseen data and achieve improved overall performance.

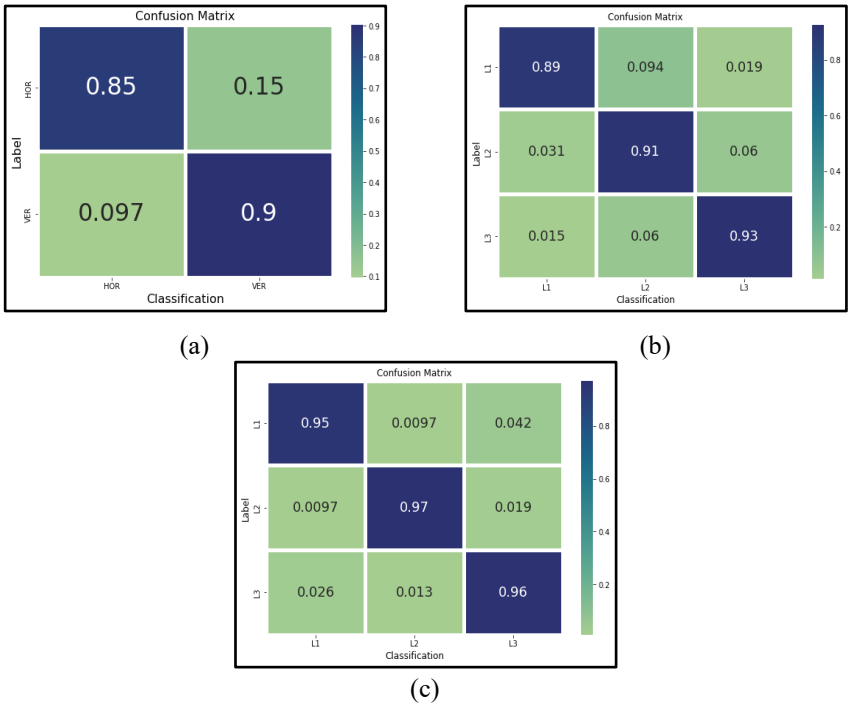


Fig. 5. Confusion matrices based on test data. (a) Component Model, (b) Horizontal Model, and (c) Vertical Model.

In this study, model testing was conducted using two methods: confusion matrices and an identification program. Confusion matrices are used to observe the comparison between the model's predictions from training and the actual data in the labels. The identification program assesses the model's performance in identifying geological locations.

The results of the confusion matrix are presented in Figure 5. In the component model (Figure 5a), it can be observed that the model was able to correctly identify the HOR label with an accuracy of 85% and the VER label with an accuracy of 90%. In the horizontal model (Figure 5b), the confusion matrix results show that the model was able to correctly identify the L1 label with an accuracy of 89%, correctly identify the L2 label with an accuracy of 91%, and correctly identify the L3 label with an accuracy of 93%. Meanwhile, in the vertical model (Figure 5c), the confusion matrix results show that the model was able to correctly identify the L1 label with an accuracy of 95%, correctly identify the L2 label with an accuracy of 97%, and correctly identify the L3 label with an accuracy of 96%. Based on the results of these three confusion matrices, it can be observed that all three models are able to identify geological conditions with an accuracy rate of over 85%. The vertical model has the highest success rate in performing identifications among the three models.

Table 4. The classification results with input data at the L1.

Input	Component		Geological Classification		
Data	Horizontal	Vertical	L1	L2	L3
Component X	99.99%	0.003%	99.99%	0.0023%	0.0000004%
Component Y	97.54%	2.44%	99.83%	0.12%	0.006%
Component Z	0.00%	100%	99.99%	0.0001%	0.000001%

The identification program was tested 9 times under various conditions. The input data consisted of components X, Y, and Z from the locations L1, L2, and L3. The results of the testing of the identification program using input data at the L1 location are shown in Table 4. When the input data was component X, the program predicts that the input is a horizontal component with a confidence level of 99% and identifies it as L1 with a confidence level of 99%. When the input was component Y, the program predicts that the input is a horizontal component with a confidence level of 97% and identifies it as L1 with a confidence level of 99%. Meanwhile, when the input data was component Z, the program predicts that the input is a vertical component with a confidence level of 100% and identifies it as L1 with a confidence level of 99%.

Table 5. The classification results with input data at the L2 location.

Input	Component		Geological Classification		
Data	Horizontal	Vertical	L1	L2	L3
Component X	98.75%	1.02%	2.68%	79.07%	20.92%
Component Y	99.98%	0.01%	2.84%	99.66%	0.33%
Component Z	0.0001%	99.99%	0.18%	99.66%	22.93%

The results of the identification program with input data at the L2 are shown in Table 5. When the input data was component X, the program predicts that the input is a horizontal component with a confidence level of 98% and identifies it as L2 with a confidence level of 79%. When the input was component Y, the program predicts that the input is a horizontal component with a confidence level of 99% and identifies it as L2 with a confidence level of 99%. Meanwhile, when the input data was component Z, the program predicts that the input is a vertical component with a confidence level of 99% and identifies it as L2 with a confidence level of 99%.

Table 6. The classification results with input data at the L3 location.

Input Data	Component		Geological Classification		
	Horizontal	Vertical	L1	L2	L3
Component X	99.99%	0.0006%	0.00000007%	0.00002%	99.99%
Component Y	99.95%	0.04%	0.00000007%	0.0001%	99.99%
Component Z	0.50%	99.48%	2.07%	0.38%	97.44%

The results of the testing of the identification program using input data at the L3 location are shown in Table 6. When the input data was component X, the program predicts that the input is a horizontal component with a confidence level of 99% and identifies it as L3 with a confidence level of 99%. When the input was component Y, the program predicts that the input is a horizontal component with a confidence level of 99% and identifies it as L3 with a confidence level of 99%. Meanwhile, when the input data was component Z, the program predicts that the input is a vertical component with a confidence level of 99% and identifies it as L3 with a confidence level of 97%.

4 Conclusion

Based on the results of training the models using the MobileNetV2 architecture, it can be concluded that the component model has an accuracy of 87.8%, the horizontal model has an accuracy of 90.8%, and the vertical model has an accuracy of 95.3%. There are differences in accuracy values and training duration among these three models. These differences can be attributed to variations in the datasets used to train each model, where the dataset for the component and horizontal models is more diverse, while the dataset for the vertical model is more uniform. The testing results using the confusion matrix show that the program was capable of classification with an accuracy rate of more than 85%. The testing results using the program with input data from three different locations, each of which has three different components, demonstrate that the program was able to provide correct predictions 9 out of 9 times. Based on these results, it can be concluded that the program has successfully identified geological locations from the ground vibration signals.

References

1. Fazriati, E. Algoritma Analisis Sinyal Mikrotremor 3 Komponen (Studi Kasus: Daerah Karst, Citatah). Universitas Padjadjaran, (2022).
2. Sitorus, N., Purwanto, S., Utama, W. Analisis Nilai Frekuensi Natural Dan Amplifikasi Desa Olak Alen Blitar Menggunakan Metode Mikrotremor HVSR. Jurnal Geosaintek, **3**(2), 89-92 (2017).
3. Simanjuntak, A. V. H., Asnawi, Y., Umar, M., Rizal, S., Syukri, M. A microtremor survey to identify seismic vulnerability around Banda Aceh using HVSr analysis. Elkawnie: Journal of Islamic Science and Technology, **6**(2), 342-358 (2020).
4. Burton, P. W., Xu, Y., Tselentis, G. A., Sokos, E., Aspinall, W. Strong ground acceleration seismic hazard in Greece and neighboring regions. Soil Dynamics and Earthquake Engineering, **23**(2), 159-181 (2003).
5. Sishananto, C. A., Murti, M. A., Priramadhi, R. A. Pendeteksi Struktur Tanah Dengan Menggunakan Sensor Geophone. eProceedings of Engineering, **7**(1), (2020).
6. Gazali, I., Purwanto, M. S., Warnana, D. D. Estimasi Kecepatan Gelombang Geser (Vs) Berdasarkan Inversi Mikrotremor Spectrum Horizontal to Vertikal Spectral Ratio (HVSR) Studi Kasus: Tanah Longsor Desa Olak-Alen, Blitar. Jurnal Teknik ITS, **6**(2), C383-C387 (2018).
7. Guo, T., Dong, J., Li, H., Gao, Y. Simple convolutional neural network on image classification. In 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA) (pp. 721-724). IEEE (2017).
8. Zhong, Z., Li, H. Recognition and prediction of ground vibration signal based on machine learning algorithm. Neural Computing and Applications, **32**, 1937-1947 (2020).
9. LeCun, Y., Bengio, Y., Hinton, G. Deep learning. nature, **521**(7553), 436-444 (2015).
10. Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, **25** (2012).
11. Ilahiyah, S., Nilogiri, A. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia), **3**(2), 49-56 (2018).
12. Miftahuddin, Y. Perbandingan Metode EfficientNetB3 dan MobileNetV2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun: Metode EfficientNetB3 dan MobileNetv2. Jurnal Ilmiah Teknologi Infomasi Terapan, **9**(1), (2022).
13. Ooms, J. The jsonlite package: A practical and consistent mapping between json data and r objects. arXiv preprint arXiv:1403.2805, (2014).
14. Laghari, W. M., Baloch, M. U., Mengal, M. A., Shah, S. J. Performance analysis of analog butterworth low pass filter as compared to Chebyshev type-I filter, Chebyshev type-II filter and elliptical filter. Circuits and Systems, **2014**, (2014).
15. Perez, L., Wang, J. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621, (2017).
16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520), IEEE (2018).
17. Shrivastava, V. K., Pradhan, M. K., Minz, S., Thakur, M. P. Rice plant disease classification using transfer learning of deep convolution neural network. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **42**, 631-635 (2019).
18. Patterson, J., Gibson, A. Deep learning: A practitioner's approach. " O'Reilly Media, Inc.", (2017).

19. Hendriyana, H., Maulana, Y. H. Identification of Types of Wood using Convolutional Neural Network with Mobilenet Architecture. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), **4**(1), 70-76 (2020).
20. Tong, Q., Liang, G., Bi, J. Calibrating the adaptive learning rate to improve convergence of ADAM. Neurocomputing, **481**, 333-356 (2022).
21. Ehtisham, R., Mir, J., Chairman, N., Ahmad, A. Evaluation of pre-trained ResNet and MobileNetV2 CNN models for the concrete crack detection and crack orientation classification. In Proceedings of the 1st International Conference on Advances in Civil and Environmental Engineering, Taxila Pakistan (pp. 22-23), (2022).
22. Brownlee, J. Better deep learning: train faster, reduce overfitting, and make better predictions. Machine Learning Mastery, (2018)
23. Kotsilieris, T., Anagnostopoulos, I., Livieris, I. E. Special Issue: Regularization Techniques for Machine Learning and Their Applications. Electronics **2022**, (2022).