

Seismic Data Quality Analysis Based on Image Recognition Using Convolutional Neural Network

Hapsoro Agung Nugroho¹, Siti Hasanah², Mahmud Yusuf²

¹Department of Instrumentation for MKG, School of Meteorology Climatology and Geophysics, Indonesia

²Indonesia Agency for Meteorology Climatology and Geophysics, Indonesia

¹hapsoro.agung@bmgk.go.id, ²siti.hasanah@bmgk.go.id, ³mahmud.yusuf@bmgk.go.id

Abstract - Seismometer monitoring and evaluation activities at the Indonesia Tsunami Early Warning System (InaTEWS) station can be carried out through a seismometer sensor calibration system with the use of the software of Seismic Data Quality Analysis. The software output is in the form of a spectrum image that represents the conditions of the seismometer following the spectrum results. The identification of the seismometer condition can be made by pattern recognition in the spectrum image. This study employed a neural network, specifically the Convolutional Neural Network (CNN), to analyse the pattern condition. The test results show that the performance of the system will be excellent if 1024 hidden layers are used. In addition, the epoch test shows that the system works well when given a maximum epoch value of 50. The test of image size gives the result that the system performance will result in good using input with a size of 30x20 pixels. The final results of the classification of spectrum images using CNN will exhibit the identification of seismometer. For the validation, the confusion matrix test shows that the corresponding findings are 80%, while the conflicting results are 20%.

Keywords: seismic quality data, seismograph, image recognition, convolutional neural network

I. INTRODUCTION

Indonesia Agency for Meteorology, Climatology, and Geophysics (BMKG) has a seismograph network installed throughout Indonesia for earthquake and tsunami monitoring in the Indonesian Tsunami Early Warning System (InaTEWS) seismic network. Equipment configuration on a seismograph network consists of a broadband seismometer, an accelerometer, and a 24-bit digitizer and uses a satellite connection for data transmission. Data from seismographs must be accurate and obtained in real-time, therefore the information service can be carried out correctly. BMKG has a significant task to maintain the operation of the equipment system to ensure it continues to work 24 hours a day. Observation equipment operated at the BMKG observation station must be operation-worthy, and the

operation's reliability is guaranteed by periodic calibration of the equipment. The Engineering Instrumentation and Geophysical Equipment Calibration Division has the main task to carry out an inventory, monitoring, and evaluation of geophysical equipment and its supporting devices, as well as to regularly calibrate geophysical equipment.

Seismometer monitoring and evaluation activities at the InaTEWS are carried out through a seismometer sensor calibration system using the Seismic Data Quality Analysis software or known as SQLX [1]. SQLX software produces digital images in the form of power spectrum density, also known as Power Spectral Density (PSD), and probability density, also called Probability Density Function (PDF), derived from seismometer measurements [2-3]. The digital image can describe the condition of the seismometer that operates following the spectrum result. Currently, the identification process of the seismometer condition is performed manually by the calibration officer. This officer analyzes the status of the seismometer by reviewing the spectrum image and checking for the presence of gaps, the existence of overlaps, RMS values, amplitude balance, the location of the spectrum on the Peterson Model, primary peak, and secondary peak [4-5]. This certainty can affect the calibration results because the identification process is subjective to the evaluation of each evaluator.

Convolutional neural network (CNN) is a development of the Multilayer Perceptron (MLP), which is designed to process two-dimensional data [6-7]. CNN is included in the Deep Neural Network type due to the high network depth and it is widely applied to image data [8]. In the case of image classification, MLP is unsuitable for use considering it does not store spatial information from image data and assumes each pixel is an independent feature resulting in unfavorable results [9]. CNN classifies the data labeled using the supervised learning method. Thus, it trained the data and from targeted variables. As such, the purpose of this method is to group data into existing data. Several studies related

to CNN are used implemented to introduce the characteristics of an object. CNN can recognize facial patterns [10-11], classify sounds that exist in the environment [12], recognize speech patterns associated with one's emotions [13], and be able to recognize human activity recognition from the accelerometer [14-15].

II. METHOD

The initial stage of designing the system is the construction of a system block diagram. A block diagram system is a graphical representation of the system being built. Fig. 1 shows a block diagram for a digital image processing application design.

As is shown in Fig. 2, the system input is made in the form of a spectrum image. Images show the seismographs recorded that have three orientations, one vertical direction (z component) and two horizontal directions (east-west component and north-south component). In the figure (a), shows seismograph records for the vertical direction or Z component (b),

seismograph records for the east-west direction or E-W component, and (c) seismograph records for the north-south direction or N-S component. Spectrum images were obtained from the process of recorded seismometer data at each InaTEWS station using SQLX software. Input in the form of a spectrum image is then subjected to image processing. Thus, the initial stage of image processing can be cropped.

Cropping is the process of cutting the image to get only the needed objects. The cropping process in this system uses the thresholding method of Hue, Saturation, and Value (HSV) values [16]. Spectrum images that have gone through the stages of cropping are then converted into grayscale image forms to clarify spectrum patterns. The grayscale image then undergoes a resizing process or resized to 30x20 pixels. The resize process is done because the used spectrum image data has different sizes, therefore there is a need for a change in size to obtain the same size of image data which results in streamlining the next process.

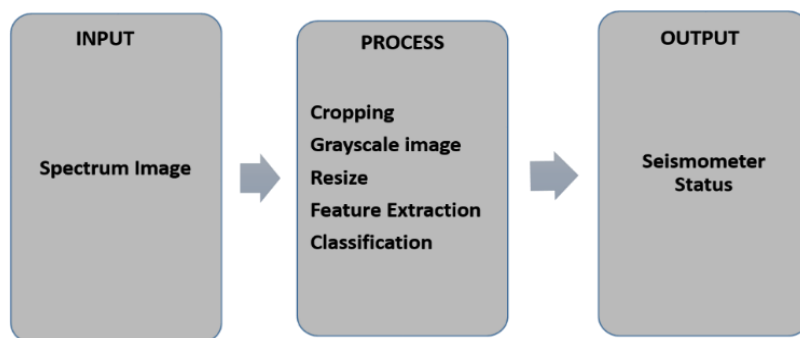


Fig. 1 Diagram block of the seismic quality data image recognition

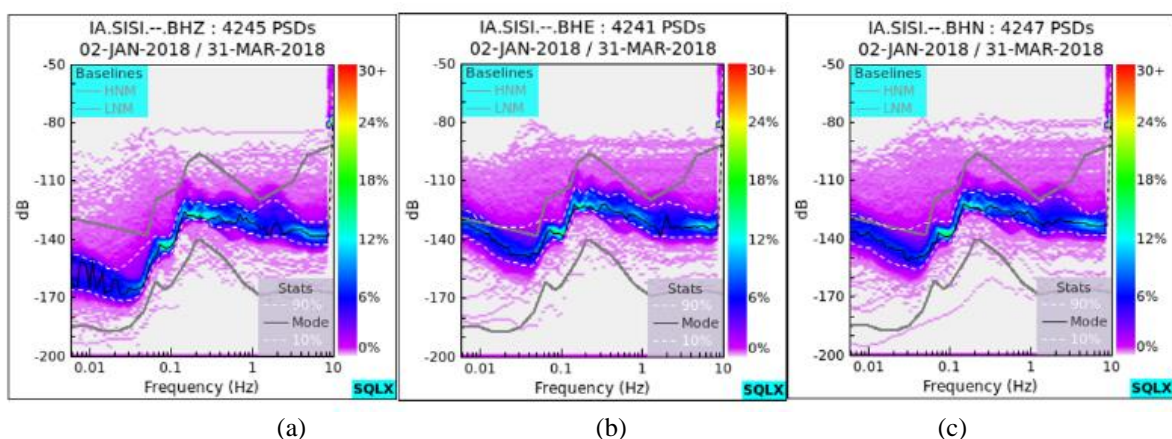


Fig. 2 Spectrum image as input (a) Z component (b) E-W component (c) N-S component

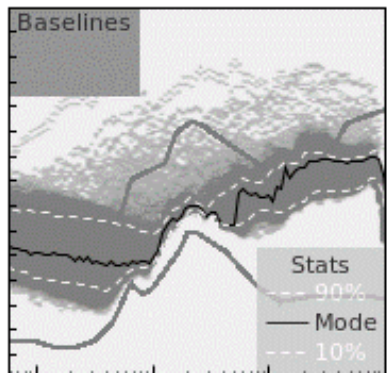


Fig. 3 Spectrum image that has been converted into a grayscale image and through the process of cropping

As is shown in Fig. 3, the grayscale image has been resized further through the feature extraction step. The use of feature extraction is to obtain special features possessed by spectrum images. The unique feature of a spectrum image is the spectrum pattern in that image itself. This research uses CNN therefore the process of feature extraction and image classification is carried out simultaneously as opposed to separately using the CNN algorithm. The CNN classification process produces the final result in the form of a seismometer. This study provides five indicators for each seismometer condition: (i) green indicates right seismometer conditions, (ii) yellow indicates poor seismometer conditions, (iii) red indicates damaged seismometer conditions, (iv) grey indicates metadata errors, and (v) black indicates seismometer conditions are not in operation.

A. Convolutional Neural Network

The convolutional network consists of three layers: the convolutional layer, the pooling layer, and the fully connected layer [17]. The convolutional layer is a layer that has a collection of filters to study the input image. Features will be extracted at this layer and then proceed to the next layer to remove more complex features [18]. The pooling layer, or subsampling, is a reduction in the size of the matrix by using pooling operations that are performed after the convolutional layer [19]. This CNN employs two types of pooling, i.e., average pooling and max pooling. One of the characteristics of average pooling is the value taken the average value, while the feature of max-pooling is the value received at the maximum value [20]. The pooling method that widely used in CNN architecture is max pooling. Max pooling divides the convolutional output layer into several grids, and then each filter shift will take the most significant value from each grid. The image generated at the pooling layer is a small part of its original size, which is useful to reduce the dimensions of the data. As a result, it will reduce the number of parameters in the next step. The

fully connected layer is a layer where all the activation neurons from the previous layer are all connected with neurons in the next layer [21]. This layer is usually used in MLP, which aims to transform data dimensions to classify the data linearly. The fully connected layer will take the input from the output pooling layer in the form of a feature map [22]. The previous feature map is still in the way of a multidimensional array. Thus, this layer will reshape the feature map and produce as many n-dimensional vectors.

Convolution operations are calculated based on two real-valued argument functions [23]. This operation applies the output function as a feature map of the image input. Furthermore, these inputs and outputs can be seen as two real-valued arguments. Convolution operations can be written with the following formula use (1).

$$s(t) = (x * w)(t) \quad (1)$$

The $s(t)$ function gives a single output in the form map, and the first argument is the input, which is x , and the second argument was the kernel or filter. Input is a two-dimensional image as such t can be expressed as pixels and replaced with i and j . Operations for convolution to inputs with more than one dimension can be written as (2) and (3).

$$S(i, j) = (K * I)(i, j) = \sum I(I + m, j + n) K(m, n) \quad (2)$$

$$S(i, j) = (K * I)(i, j) = \sum I(I - m, j - n) K(m, n) \quad (3)$$

Equations (2) and (3) are the basic calculations in convolution operations where i and j are the pixels of the image. The calculation is commutative and appears when K is the kernel, I is the input, and the kernel is reversible relative to the input. Convolution operations can be seen as a matrix multiplication between the input image and the kernel, whereas the output can be calculated by dot product [24].

The input image on the CNN model uses a 30x20x1 image. The size of 30x20x1 indicates that the entered image has a size of 30x20 pixels, and the number one indicates that the entered image has one gray-scale color channel (greyscale image). The input image will then be processed first through the convolution process and pooling process at the feature learning stage. The number of convolution processes in this design has three convolution layers. Each convolution has a different number of filters but uses the same kernel size. The next process is the flatten process, or the process of changing the feature map of the pooling layer results in vector shapes. This process is usually called the fully connected layer stage. Fig. 4 shows the design of the CNN architecture in this study.

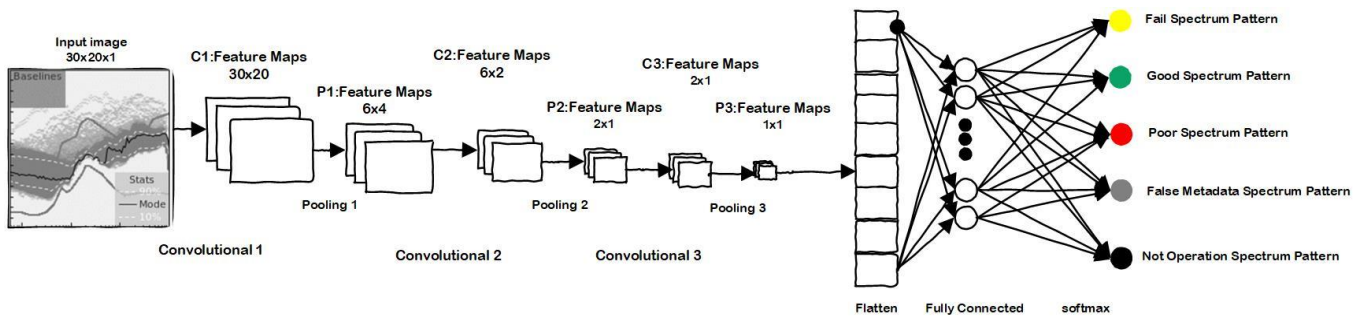


Fig. 4 Convolutional neural network architecture

The size of the image used in this study are 30x20, 50x50, 64x64, and 100x100, besides the number of hidden layers used are 32, 64, 256, 512, and 1024 to determine performance the best classification system. Fig. 4 shows the CNN model design using a 30x20 image size and 512 hidden layers. The first convolution used 32 filters and a kernel with a 5x5 matrix, and then the pooling process was done by using a 5x5 pooling size. The next stage was the second convolution phase using 50 filters and a kernel with a 5x5 matrix. The third convolution stage was carried out after the second convolution phase is completed by using 80 filters and a kernel with a 5x5 matrix. Subsequently, it was then proceed with the flatten process that changes the output of the convolution process in the form of a matrix into a vector. The vector will be continued in the classification process using MLP with the number of neurons in the hidden layer that has been determined, which is 512 layers. The class of images was then classified based on the value of neurons in the hidden layer employing the softmax activation function.

B. Feature Extraction

The system will train the input as image files. The training image will be cropped to get the object needed, then the image that has been cut will be converted to a grayscale image and resized to 30x20 pixels. For the next process, the system will provide labeling to the image following the file name of each inserted image. The labeling process uses five categories with the following provisions: (i) files with the name yellow_fail will be labeled as a fail spectrum pattern, (ii) files with the name green_good will be labeled as a good spectrum pattern, (iii) files with the name red_poor will be labeled as a poor spectrum pattern, (iv) files with the name gray_false_metadata will be labeled as an false metadata spectrum pattern, and (v) files with the name black_no_operation will be labeled as no operation spectrum pattern. The results of the training can be used

if the training process produces greater accuracy equal to 90%. If the training accuracy is less than 90%, then the system will not be processed to the next step.

III. RESULTS AND DISCUSSION

Spectrum image data for the training process amounted to 700 spectrum images, consisting of 140 good category spectrum images, 140 damaged category spectrum images, 140 poor category spectrum images, 140 wrong category metadata spectrum images, and 140 dead category spectrum images. Furthermore, spectrum image data for the testing process amounted to 225 spectrum images which consists of 45 categories of good spectrum images, 45 categories of fail spectrum images, 45 types of poor spectrum images, 45 categories of false metadata spectrum, and 45 categories of not operation spectrum images. The training data testing was undertaken by testing the image size, hidden layer testing, and epoch testing employing the confusion matrix. Image size testing was performed to determine the performance of the system at 512 hidden layers and a maximum epoch of 50, however different sized images were used, such as 30x20 pixels, 50x50 pixels, 64x64 pixels, and 100x100 pixels. The test results are shown in Table I.

Table I indicates that the image size, is proportional to the time needed to carry out the training process as such the training process is inefficient. An actual image used for the training process in this study is an image with a size of 30x20 pixels because the training process that uses an image size of 30x20 can produce training data with an accuracy of 1 in 34.52 seconds. Hidden layer testing was performed to determine the training performance using the same image size at 30x20 pixels. Subsequently, different epochs at 25, 50, and 100 and several different hidden layers at 32, 64, 256, 512, and 1024 were used to complete the system training. The test results are shown in Table II - V.

TABLE I
THE PERFORMANCE OF IMAGE SIZE TESTING RESULTS

No.	Size (pixel)	Number of Hidden Layer	Number of Epoch	Accuracy	Time (second)
1.	30x20	512	50	1	34,52
2.	50x50	512	50	1	60,37
3.	64x64	512	50	1	120,42
4.	100x100	512	50	1	360,02

TABLE II
THE PERFORMANCE OF HIDDEN LAYER TEST RESULTS WITH A MAXIMUM EPOCH OF 25

No.	Number of Hidden Layer	Accuracy	Time (second)
1.	32	0,7200	20,00
2.	64	0,9551	20,64
3.	256	0,9943	20,97
4.	512	0,9886	20,26
5.	1024	0,7653	19,84

TABLE III
THE PERFORMANCE OF HIDDEN LAYER TEST RESULTS WITH A MAXIMUM EPOCH OF 50

No.	Number of Hidden Layer	Accuracy	Time (second)
1.	32	0,9971	38,60
2.	64	0,9985	38,77
3.	256	1	38,41
4.	512	1	38,70
5.	1024	1	29,75

TABLE IV
THE PERFORMANCE OF HIDDEN LAYER TEST RESULTS WITH A MAXIMUM EPOCH OF 100

No.	Number of Hidden Layer	Accuracy	Time (second)
1.	32	1	59,00
2.	64	1	55,83
3.	256	1	52,53
4.	512	1	53,90
5.	1024	1	60,02

TABLE V
THE PERFORMANCE OF EPOCH TESTING RESULTS

No.	Size (pixel)	Number of Hidden Layer	Number of Epoch	Accuracy	Time (second)
1.	30x20	25	1024	0,7653	19,84
2.	30x20	50	1024	1	29,75
3.	30x20	100	1024	1	60,02

Table V shows that the number of epochs affects the training process based on the magnitude of the accuracy value obtained from the training process. The higher the epoch value used, the greater the accuracy value received, however the system efficiency cannot be seen based on its accuracy value alone. It needs to consider the time

spent during the training process. A number of good epochs used in this study were an epoch of 50 because it produced training data with an accuracy of 1 in 19.84 seconds. Image training will provide both loss and accuracy values in a range of values from 0 to 1.

Fig. 5 shows a graph for the comparison of loss values with epoch values when the training process was run using 700 training data, the epoch number of 50, and the learning rate value of 0.001. The small loss value is proportional to the number of epochs when the number of iterations has done.

As shown in Fig. 6, the accuracy value will increase in proportion to the number of epochs when the number of iterations was undertaken. Image training in this study is concluded to be successful when the accuracy value is more significant than 0.9. In contrast, when the accuracy value shows the number of 0.9, it can not be used at the next step for the spectrum image testing process.

Furthermore, the test data was validated using spectrum image data from 75 InaTEWS stations. They have spectrum images representing BHE, BHN, and BHZ components. In addition, reference data derived from seismometer condition analysis reports on DokuWiki with the address <https://geof.bmkg.go.id>

/dokuwiki/ were used to validate data as shown in Table 6. The value of the test data following the reference data is 80%, while the value that does not follow the reference data is 20%. This value was obtained using the following calculation (4) and (5) [25].

$$Accuracy\ True = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (4)$$

$$Accuracy\ False = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (5)$$

True Positive (TP) : category images that were correctly classified as category images

True Negative (TN) : non-category images that were correctly classified as non-category images

False Positive (FP) : non-category images that were incorrectly labeled as category images

False Negative (FN) : category images that were incorrectly marked as non-category images

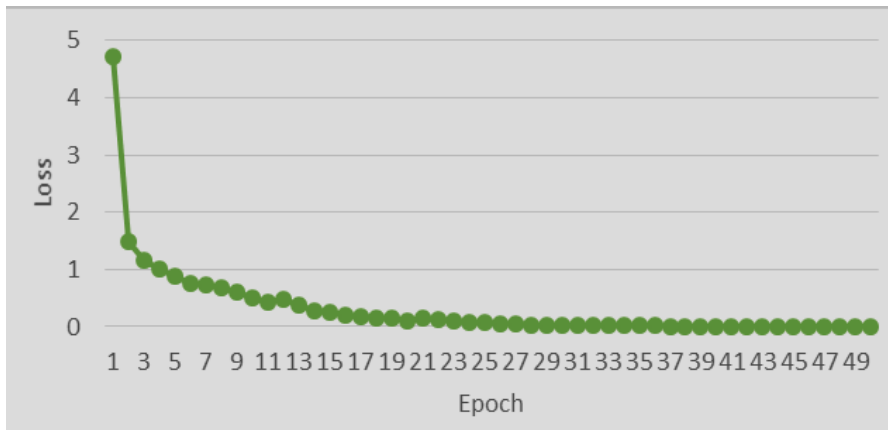


Fig. 5 Comparison of loss with epoch

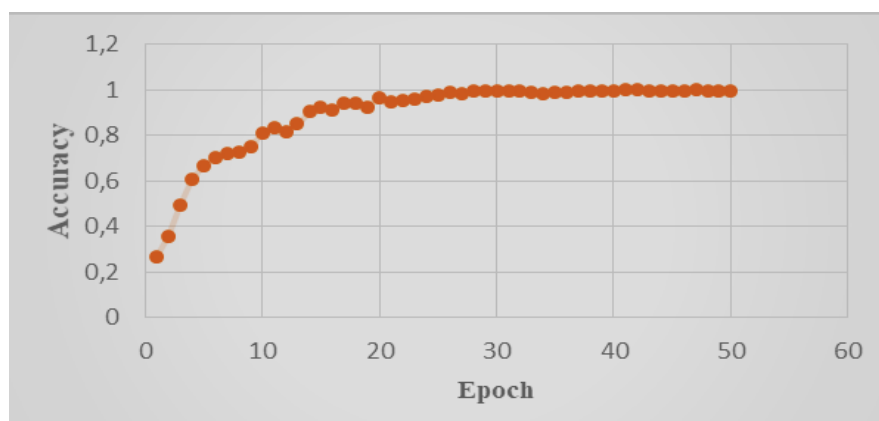


Fig. 6 Comparison of accuracy with epoch

The calculation results show that the work of the system in classifying the conditions of a seismometer gives a good performance up to 80%. This value indicates that the system can distinguish the types of spectrum patterns in each category, while the value of 20% suggests the failure of the system in classifying spectrum patterns in the test data used.

Table VI shows that there are data that do not follow the reference data during the testing stage. For instance, the data that were entered in either category is classified by the system as incorrect metadata categories. The results of the classification are not appropriate in regards to the different size of the spectrum image used in this study. Different sizes of images used in the training and testing process can affect cropping results affecting classification results that can possibly diverge.

Data with functional categories can be classified into the wrong metadata category because it has the same side size. Data with functional groups have different sizes

therefore images that have the same side size will be classified into incorrect metadata data when tested for good categories, as shown in Fig. 7. In the figure (a), shows seismograph records for the east-west component or E-W component (b), seismograph records for the north-south component or N-S component, and (c) seismograph records for the vertical component or Z component.

The shortcome of this system that is it is only able to classify the condition of the seismometer based on spectrum patterns in the trained image and unable to integrate with gap data, overlap data, RMS value data, or amplitude data, as such the accuracy of the classification results was only 80%. The system will work better if the seismometer condition classification process is not only based on the recognition of spectrum image patterns but also integrated with gap data, overlap data, RMS value data, and amplitude data.

TABLE VI
THE PERFORMANCE OF CONFUSION MATRIX

		Prediction				
		Good	Fail	False metadata	Not operation	Poor
Actual	Good	10	0	2	0	3
	Fail	0	12	2	1	0
	False metadata	0	0	15	0	0
	Not operation	0	1	0	14	0
	Poor	1	4	1	0	9

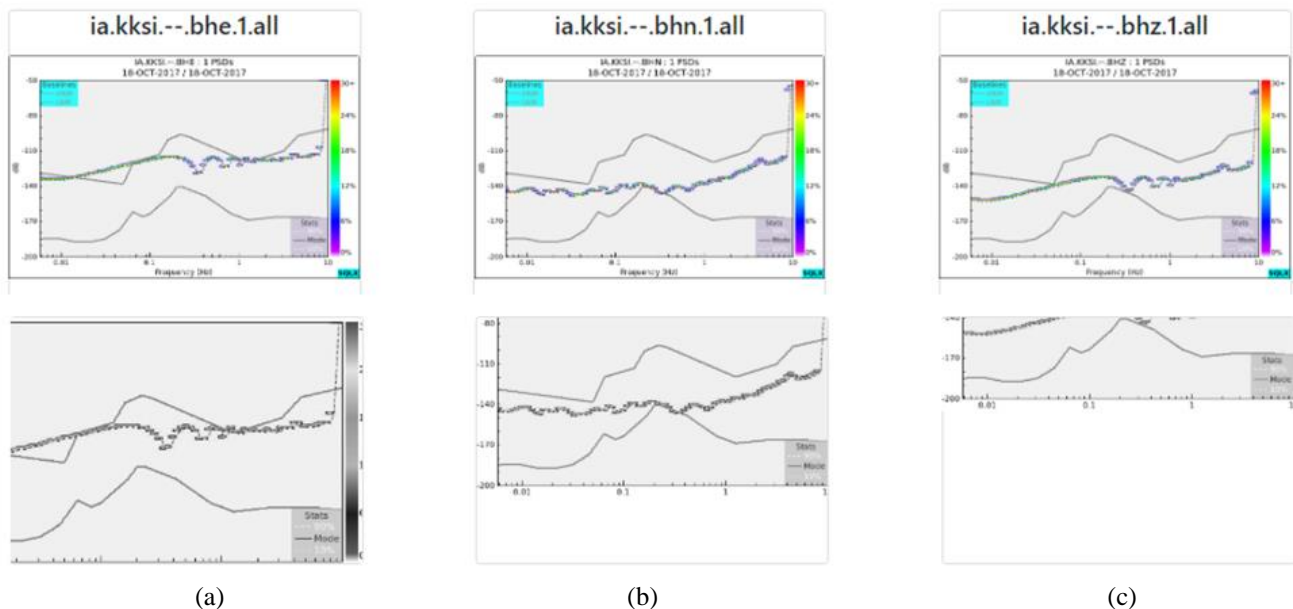


Fig. 7 The system reported the results of the fail category (above) classification with the not operating (bottom) category test data. (a) E-W component (b) N-S component (c) Z component

IV. CONCLUSION

The large image size causes the amount of time needed to carry out the training process resulting in the training process to be inefficient. The accuracy value is not only influenced by the number of hidden layers but also by the maximum amount of the epoch. A considerable epoch value will produce a substantial accuracy value, however time needed for the learning process will increase rapidly. The results will give better performance when the results of the recognition of spectrum pattern data are integrated with the gap data, overlap data, RMS value data, and amplitude data. These integrations will increase the accuracy value of the seismometer condition classification. The next system development is expected to be able to identify seismometer damage accompanied by an analysis of the causes of the failed seismometer.

ACKNOWLEDGEMENT

We would like to thank the Engineering Instrumentation and Geophysical Equipment Calibration Division of the Indonesia Agency for Meteorology Climatology and Geophysics (BMKG) and Division for Research and Community Service (UP2KM) School of Meteorology Climatology and Geophysics (STMKG) for their support to publish this article.

REFERENCES

- [1] E. Y. Arias, B. C. Beaudoin, R. Woodward, K. E. Anderson, and A. M. Reusch, "Relative noise level comparison of portable broadband seismometer installation techniques used by PASSCAL and flexible array," in *AGU Fall Meeting Abstracts*, 2014.
- [2] B. Weber and S. Herrnkind, "Automatic Classification of Station Quality by Image Based Pattern Recognition of Ppsd Plots," in *AGU Fall Meeting Abstracts*, 2017.
- [3] B. Grecu, C. Neagoe, D. Tataru, F. Borleanu, and B. Zaharia, "Analysis of seismic noise in the Romanian-Bulgarian cross-border region," *J. Seismol.*, vol. 22, no. 5, pp. 1275–1292, 2018.
- [4] A. T. Ringler and C. R. Hutt, "Self-noise models of seismic instruments," *Seismol. Res. Lett.*, vol. 81, no. 6, pp. 972–983, 2010.
- [5] Y. El Fellah, A. E.-A. K. A. El-Aal, M. Harnafi, and A. Villaseñor, "New comprehensive standard seismic noise models and 3D seismic noise variation for Morocco territory, North Africa, obtained using seismic broadband stations," *Explor. Geophys.*, vol. 48, no. 3, pp. 272–283, 2017.
- [6] S. Savalia and V. Emamian, "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks," *Bioengineering*, vol. 5, no. 2, p. 35, 2018.
- [7] I. K. Gunawan, I. P. A. Bayupati, K. S. Wibawa, I. M. Sukarsa, and L. Adi, "Indonesian Plate Number Identification Using YOLACT and Mobilenetv2 in the Parking Management System," *JUITA J. Inform.*, vol. 9, no. 1, pp. 69–76, 2021.
- [8] G. J. Scott, M. R. England, W. A. Starns, R. A. Marcum, and C. H. Davis, "Training deep convolutional neural networks for land-cover classification of high-resolution imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 4, pp. 549–553, 2017.
- [9] S. H. Supangkat and F. F. Lubis, "Moving Image Interpretation Models to Support City Analysis," in *2018 International Conference on ICT for Smart Society (ICISS)*, 2018, pp. 1–5.
- [10] B. Hasani and M. H. Mahoor, "Facial expression recognition using enhanced deep 3D convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 30–40.
- [11] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: coping with few data and the training sample order," *Pattern Recognit.*, vol. 61, pp. 610–628, 2017.
- [12] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, 2017.
- [13] D. Bertero and P. Fung, "A first look into a convolutional neural network for speech emotion detection," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 5115–5119.
- [14] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2017, pp. 131–134.
- [15] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, 2016.
- [16] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, "Automatic crop detection under field conditions using the HSV colour space and morphological operations," *Comput. Electron. Agric.*, vol. 133, pp. 97–107, 2017.
- [17] L. Jing, M. Zhao, P. Li, and X. Xu, "A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox," *Measurement*, vol. 111, pp. 1–10, 2017.
- [18] V. Bui and L.-C. Chang, "Deep learning architectures for hard character classification," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, 2016, p. 108.

- [19] M. Abadi *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [20] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, “A dynamic recurrent model for next basket recommendation,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 729–732.
- [21] B. A. M. Ashqar and S. S. Abu-Naser, “Image-Based Tomato Leaves Diseases Detection Using Deep Learning,” *Int. J. Acad. Eng. Res.*, vol. 2, no. 12, pp. 10–16, 2019.
- [22] C. M. P. Cases *et al.*, “Quality Assessment and Prediction of Philippine Mangoes: A Convolutional Neural Network Approach,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 6, pp. 2128–2133, 2019.
- [23] X. Zhu, Y. Xu, H. Xu, and C. Chen, “Quaternion convolutional neural networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–647.
- [24] L. Lai, N. Suda, and V. Chandra, “Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus,” *arXiv Prepr. arXiv1801.06601*, 2018.
- [25] A. M. Abdel-Zaher and A. M. Eldeib, “Breast cancer classification using deep belief networks,” *Expert Syst. Appl.*, vol. 46, pp. 139–144, 2016.

