Detection of Deep Low-Frequency Tremors from Continuous Paper Records at a Station in Southwest Japan About 50 Years Ago Based on Convolutional Neural Network for Seismogram Images

Hiromichi Nagao[1,1], Ryosuke Kaneko[2,2], Shin-Ichi Ito[1,1], Hiroshi Tsuruoka[1,1], and Kazushige Obara[1,1]

[1]Earthquake Research Institute
[2]Graduate School of Information Science and Technology

November 30, 2022

Abstract

The establishment of the High Sensitivity Seismograph Network (Hi-net) in Japan has led to the discovery of deep low-frequency tremors. Since such tremors are considered to be related to large earthquakes adjacent to tremors on the same subducting plate interface, it is important in seismology to investigate tremors before establishing modern seismograph networks that record seismic data digitally. We propose a deep learning method to detect evidence of tremors from seismogram images recorded on paper more than 50 years ago. In our previous study, we constructed a convolutional neural network (CNN) based on the Residual Network (ResNet) structure and verified its performance through learning with synthetic images generated based on past seismograms. In this study, we trained the CNN with seismogram images converted from real seismic data recorded by Hi-net. The CNN trained by fine-tuning achieved an accuracy of 98.64% for determining whether an input image contains tremors. The Gradient-weighted Class Activation Mapping (Grad-CAM) heatmaps to visualize model predictions indicate that the CNN successfully detects tremors without affections of a variety of noises, such as teleseisms. The trained CNN was applied to the past seismograms recorded at the Kumano observatory, Japan, operated by Earthquake Research Institute, The University of Tokyo. The CNN shows the potential to detect tremors from past seismogram images for broader applications, such as publishing a new tremor catalog.

Detection of Deep Low-Frequency Tremors from Continuous Paper Records at a Station in Southwest Japan About 50 Years Ago Based on Convolutional Neural Network for Seismogram Images

Ryosuke Kaneko[1,2], Hiromichi Nagao[2,1], Shin-ichi Ito[2,1], Hiroshi Tsuruoka[2], and Kazushige Obara[2]

[1]Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan
[2]Earthquake Research Institute, The University of Tokyo, Tokyo, Japan

Key Points:

- Convolutional neural network model for detection of deep low-frequency tremors from seismogram images is proposed
- The model trained with seismogram images converted from real seismic data successfully detects tremors
- The detection performances of the trained model for the paper records at the Kumano observatory in southwest Japan are discussed

Corresponding author: Hiromichi Nagao, `nagaoh@eri.u-tokyo.ac.jp`

Abstract

The establishment of the High Sensitivity Seismograph Network (Hi-net) in Japan has led to the discovery of deep low-frequency tremors. Since such tremors are considered to be related to large earthquakes adjacent to tremors on the same subducting plate interface, it is important in seismology to investigate tremors before establishing modern seismograph networks that record seismic data digitally. We propose a deep learning method to detect evidence of tremors from seismogram images recorded on paper more than 50 years ago. In our previous study, we constructed a convolutional neural network (CNN) based on the Residual Network (ResNet) structure and verified its performance through learning with synthetic images generated based on past seismograms. In this study, we trained the CNN with seismogram images converted from real seismic data recorded by Hi-net. The CNN trained by fine-tuning achieved an accuracy of 98.64% for determining whether an input image contains tremors. The Gradient-weighted Class Activation Mapping (Grad-CAM) heatmaps to visualize model predictions indicate that the CNN successfully detects tremors without affections of a variety of noises, such as teleseisms. The trained CNN was applied to the past seismograms recorded at the Kumano observatory, Japan, operated by Earthquake Research Institute, The University of Tokyo. The CNN shows the potential to detect tremors from past seismogram images for broader applications, such as publishing a new tremor catalog.

1 Introduction

Deep low-frequency tremors are well-recognized as stress release processes along the subducting plate interface. In the Nankai and Cascadia subduction zones, active tremor episodes frequently occur and are associated with slow slip events in the downdip region adjacent to the large earthquake seismogenic zones (Schwartz & Rokosky, 2007). Therefore, we expect tremors to be related to stress accumulation in large earthquake source regions (Obara & Kato, 2016). It is important to clarify the history of tremor activity during at least one cycle of an interseismic period of large earthquakes.

In southwest Japan, tremors were discovered for the first time in the downdip neighboring the seismogenic zone (Obara, 2002). This discovery was made possible by the High Sensitivity Seismograph Network (Hi-net) established by the National Research Institute for Earth Science and Disaster Prevention (NIED) in the Japan Islands after the 1995 Hyogo-ken Nanbu Earthquake (Okada et al., 2004). In this region, megathrust earthquakes have periodically occurred at intervals of 100–200 years due to subduction of the Philippine Sea Plate beneath the Eurasian Plate from the Nankai Trough. The most recent megathrust earthquakes occurred in 1944 and 1946. The available tremor catalog is currently limited to only 20 years, due to the limitations of continuous digital seismic data. Analog records must be found and used to reveal the tremor activity before the digital era (Nagai et al., 2001; Kano & Kano, 2019). In the Kii Peninsula, close to the Nankai megathrust earthquake seismogenic zone, Earthquake Research Institute (ERI), The University of Tokyo deployed a microearthquake observation network in the 1960s. Seismic data at each station were recorded continuously as daily seismograms drawn with pens on drum-roll papers. Our visual inspection of a few daily seismograms revealed signals similar to deep low-frequency tremors at a station near the tremor source region. Therefore, developing a systematic tremor detection method based on daily continuous seismogram images is required to reveal the history of tremor activity.

This study aims to detect tremors from past seismogram images by using a convolutional neural network (CNN), a deep-learning model that exhibits high performance for image recognition. A CNN can automatically tune its internal parameters by learning the features of targets from input images without requiring prior knowledge or manual adjustment of the parameters. The history of CNN started with Neocognitron (Fukushima, 1980), a neural network model for visual pattern recognition constructed based on the cognitive model of the receptive field with simple cells and complex cells (Hubel & Wiesel, 1962). In 1989, ConvNet (LeCun et al., 1989) was proposed, which was recognized as the first CNN that laid the foundation for structures and algorithms used in modern CNNs. In 2012, AlexNet (Krizhevsky et al., 2012) showed an outstanding performance in the ILSVRC, a competition of algorithms for image recognition, compared to the conventional methods, which triggered the CNN to obtain the spotlight. With improvements in the computational power of CPUs and GPUs and development of learning datasets, CNN has become one of the most popular models and is applied in various fields, such as time-series analysis and natural language processing. Recently, some seismological studies have used CNNs to analyze seismic phenomena. Previously proposed CNNs mainly target at earthquakes, *e.g.*, ConvNetQuake determines whether a waveform in a time window is an earthquake or noise (Perol et al., 2018), PhaseNet measures the arrival times of P- and S-waves of an earthquake (Zhu & Beroza, 2018), and a CNN with graph partitioning detects small seismic events with low signal-to-noise ratios utilizing spatial correlations between observatories (Yano et al., 2021). Some CNNs aim to detect tremors by transforming the raw waveform data into spectrogram images to ease the detection (Nakano et al., 2019; Rouet-Leduc et al., 2020). However, no previous study has addressed the detection of tremors from seismogram images, especially past ones.

In our previous study, we conducted numerical experiments to train a CNN using synthetic images generated based on past seismogram images (Kaneko et al., 2021). We verified that a CNN with shortcut connections of the Residual Network (ResNet) (He et al., 2016) is effective for detecting tremors based on synthetic images. This paper reports the subsequent learning with real data of the Hi-net and its application to past seismogram images of the Kumano observatory (KUM), Japan, operated by ERI.

2 Convolutional neural network

A neural network is a representative deep learning model. A basic neural network has several structures called layers. The i-th layer works as a function $\boldsymbol{f}_i = \boldsymbol{f}_i(\cdot;\boldsymbol{\theta}_i)$, where $\boldsymbol{\theta}_i$ is a vector of the parameters. An input $\boldsymbol{x}$ passes through the layers in order, and finally, the output $\boldsymbol{y}$ is obtained after the last N-th layer. Therefore, the output $\boldsymbol{y}$ for an input $\boldsymbol{x}$ is represented as

$$\boldsymbol{y} = \boldsymbol{f}_N(\cdots \boldsymbol{f}_2(\boldsymbol{f}_1(\boldsymbol{x};\boldsymbol{\theta}_1);\boldsymbol{\theta}_2)\cdots;\boldsymbol{\theta}_N)\ . \tag{1}$$

Letting $\boldsymbol{z}_i$ be the output of i-th layer, the above formula is represented as

$$\boldsymbol{z}_i = \boldsymbol{f}_i(\boldsymbol{z}_{i-1};\boldsymbol{\theta}_i) \quad i = 1, 2, \ldots, N\ , \tag{2}$$

where $\boldsymbol{z}_0$ is defined as the input $\boldsymbol{x}$. The characteristics of the neural network depend on the design of $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N$.

The optimization process of the model parameters is called learning. In this process, we only require a dataset of inputs whose correct outputs are known, called learning data. By inputting the learning data, a neural network automatically optimizes the parameters to minimize the value of loss function. The loss function $L = L(\boldsymbol{y};\boldsymbol{x})$ is designed by users to quantify the misfit between the model output and corresponding answer. In general, the loss becomes zero at a minimum when all model outputs are completely consistent with the correct answers, whereas it increases as the outputs include large or many differences from correct answers. The minimization of loss is conducted using a gradient method that optimizes the parameter of the i-th layer $\boldsymbol{\theta}_i$ as

$$\boldsymbol{\theta}_i \mapsto \boldsymbol{\theta}_i + \alpha \boldsymbol{d}\ , \tag{3}$$

where $\boldsymbol{d}$ is a direction calculated based on $\frac{\partial L}{\partial \boldsymbol{\theta}_i}$, and the positive number α is the learning rate that determines the distance $\boldsymbol{\theta}_i$ moves. For example, gradient descent, which is the most basic gradient method, defines $\boldsymbol{d} = -\frac{\partial L}{\partial \boldsymbol{\theta}_i}$. From equation (2), $\frac{\partial L}{\partial \boldsymbol{\theta}_i}$ is calculated using the chain rule as

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial L}{\partial \boldsymbol{z}_i}\frac{\partial \boldsymbol{z}_i}{\partial \boldsymbol{\theta}_i} = \frac{\partial L}{\partial \boldsymbol{z}_i}\frac{\partial \boldsymbol{f}_i(\boldsymbol{z}_{i-1};\boldsymbol{\theta}_i)}{\partial \boldsymbol{\theta}_i} \ . \tag{4}$$

Similarly, $\frac{\partial L}{\partial \boldsymbol{z}_i}$ is calculated as

$$\frac{\partial L}{\partial \boldsymbol{z}_i} = \frac{\partial L}{\partial \boldsymbol{z}_{i+1}}\frac{\partial \boldsymbol{z}_{i+1}}{\partial \boldsymbol{z}_i} = \frac{\partial L}{\partial \boldsymbol{z}_{i+1}}\frac{\partial \boldsymbol{f}_{i+1}(\boldsymbol{z}_i;\boldsymbol{\theta}_{i+1})}{\partial \boldsymbol{z}_i} \ . \tag{5}$$

Starting with $\frac{\partial L}{\partial \boldsymbol{z}_N} = \frac{\partial L}{\partial \boldsymbol{y}}$, $\frac{\partial L}{\partial \boldsymbol{z}_i}$ is calculated for $i = N, N-1, \ldots, 1$ using equation (5). Therefore, $\frac{\partial L}{\partial \boldsymbol{\theta}_i}$ is calculated efficiently in the same order using equation (4), which is called backpropagation. In summary, the learning of a neural network optimizes the parameters based on the gradients of loss calculated by backpropagation. The process is iterated until the parameters converge to a local optimum. A high computational cost is generally required to calculate the loss for all inputs of learning data at every iteration. Therefore, it is popular to use a mini-batch stochastic gradient method such as AdaGrad (Duchi et al., 2011), Adam (Kingma & Ba, 2017), and AMSGrad (Reddi et al., 2018). This method divides the learning data into many small groups called batches and calculates the loss for each batch instead of one for all data.

Excessive learning may result in overfitting, a phenomenon in which a neural network loses adaptability to inputs other than learning data. Therefore, the learning data are generally split into training and validation data to avoid overfitting. For each iteration of the learning, a neural network optimizes the parameters using the training data, and the performance is then estimated using the validation data. Accuracy and loss are often used as performance metrics. Accuracy refers to the probability that the model outputs are correct, whereas the loss is the value of the loss function. The performance is considered to improve as the accuracy increases and the loss decreases. Some neural networks contain additional layers, such as dropout layers (Srivastava et al., 2014) and batch-normalization layers (Ioffe & Szegedy, 2015) to prevent overfitting and improve the learning efficiency. A dropout layer allows some randomly-chosen parameters to be zeros during training. A batch-normalization layer applies a standardization to elements in the vectors to convert their mean to zero and variance to one for every batch.

Convolutional neural networks (CNN) have exhibited superior performance in image and handwriting recognition tasks. A CNN contains two types of distinctive layers, convolutional layers and pooling layers, which allow the CNN to extract features from the input images (Figure 1). Hereafter, each $\boldsymbol{z}_i$ is assumed to be a two-dimensional vector, and $\boldsymbol{z}_i(h,w)$ denotes the (h,w)-th component of $\boldsymbol{z}_i$. The function of the i-th convolutional layer is represented as

$$\boldsymbol{z}_i(h,w) = \sigma_i\left(\sum_{(p,q)\in D_i} \boldsymbol{h}_i(p,q)\boldsymbol{z}_{i-1}(h+p, w+q) + \boldsymbol{b}_i(h,w)\right) , \tag{6}$$

where $\boldsymbol{h}_i$ is a set of parameters called a filter that has a real value for each (p,q) in the domain D_i, which is generally a square area, $\boldsymbol{b}_i$ is a vector representing the bias, and σ_i is the activation function responsible for the conclusive output of the layer. An activation function is generally defined as a non-linear function such as a sigmoid function:

$$\varsigma(x) = \frac{1}{1+e^{-x}} \tag{7}$$

or a hyperbolic tangent function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \ . \tag{8}$$

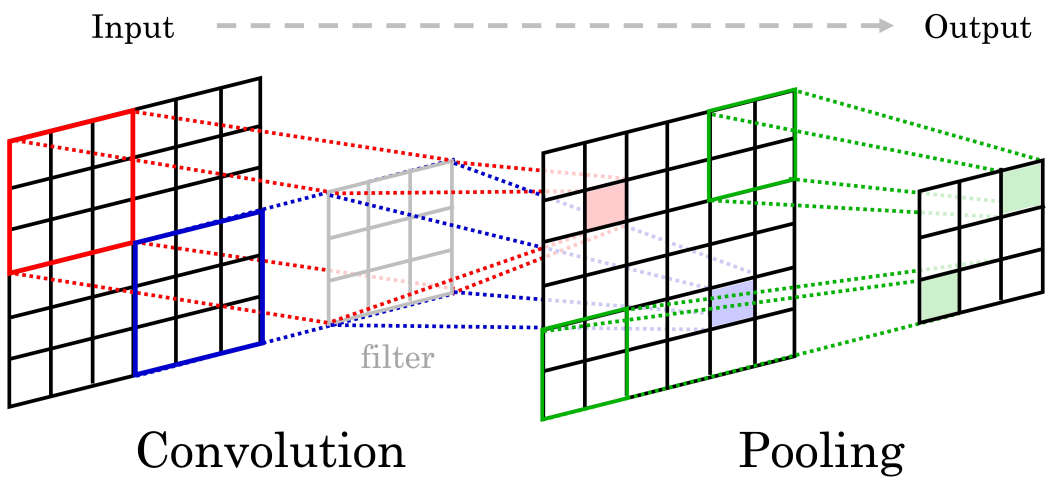

Figure 1. Schematic of convolutional layer and pooling layer. The area surrounded by red solid lines passes through the convolutional filter and outputs the value shown in light red. The blue indicates the convolution for another area. Each area surrounded by solid green lines outputs the value shown in light green after the pooling.

The absolute value of the derivative of each of these functions is less than one (if $x \neq 0$ for equation (8)). Therefore, $|\frac{\partial L}{\partial \boldsymbol{\theta}_i}|$ approaches zero based on the chain rule shown in equations (4) and (5) if a neural network has many layers. In this case, the parameters are hardly updated during the learning process shown in equation (3), which is called a vanishing gradient problem. To avoid this problem, it is popular to use ReLU

$$\mathrm{ReLU}(x) = \max\{0, x\} \tag{9}$$

or innovative functions such as Swish (Ramachandran et al., 2017) and Mish (Misra, 2020). A convolutional layer can be used to extract specific local patterns. Meanwhile, the function of the i-th pooling layer is expressed as

$$\boldsymbol{z}_i(h, w) = \mathrm{pooling}_i(\{\boldsymbol{z}_{i-1}(h+p, w+q) \mid (p,q) \in D_i\}) \ , \tag{10}$$

where $\mathrm{pooling}_i$ is a pooling function, *e.g.*, max pooling

$$\boldsymbol{z}_i(h, w) = \max_{(p,q)\in D_i} \boldsymbol{z}_{i-1}(h+p, w+q) \tag{11}$$

or average pooling

$$\boldsymbol{z}_i(h, w) = \frac{1}{S_i} \sum_{(p,q)\in D_i} \boldsymbol{z}_{i-1}(h+p, w+q) \ , \tag{12}$$

where S_i is the size of the domain D_i. A pooling layer is generally located after convolutional layers to reduce the position sensitivity of pattern extraction by convolution. A basic CNN contains convolutional layers and pooling layers alternately, and some fully-connected layers in the end whose function is represented as

$$\boldsymbol{z}_i = \sigma_i(\boldsymbol{A}_i \boldsymbol{z}_{i-1} + \boldsymbol{b}_i) \ , \tag{13}$$

where $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ are the matrix and vector of parameters, respectively.

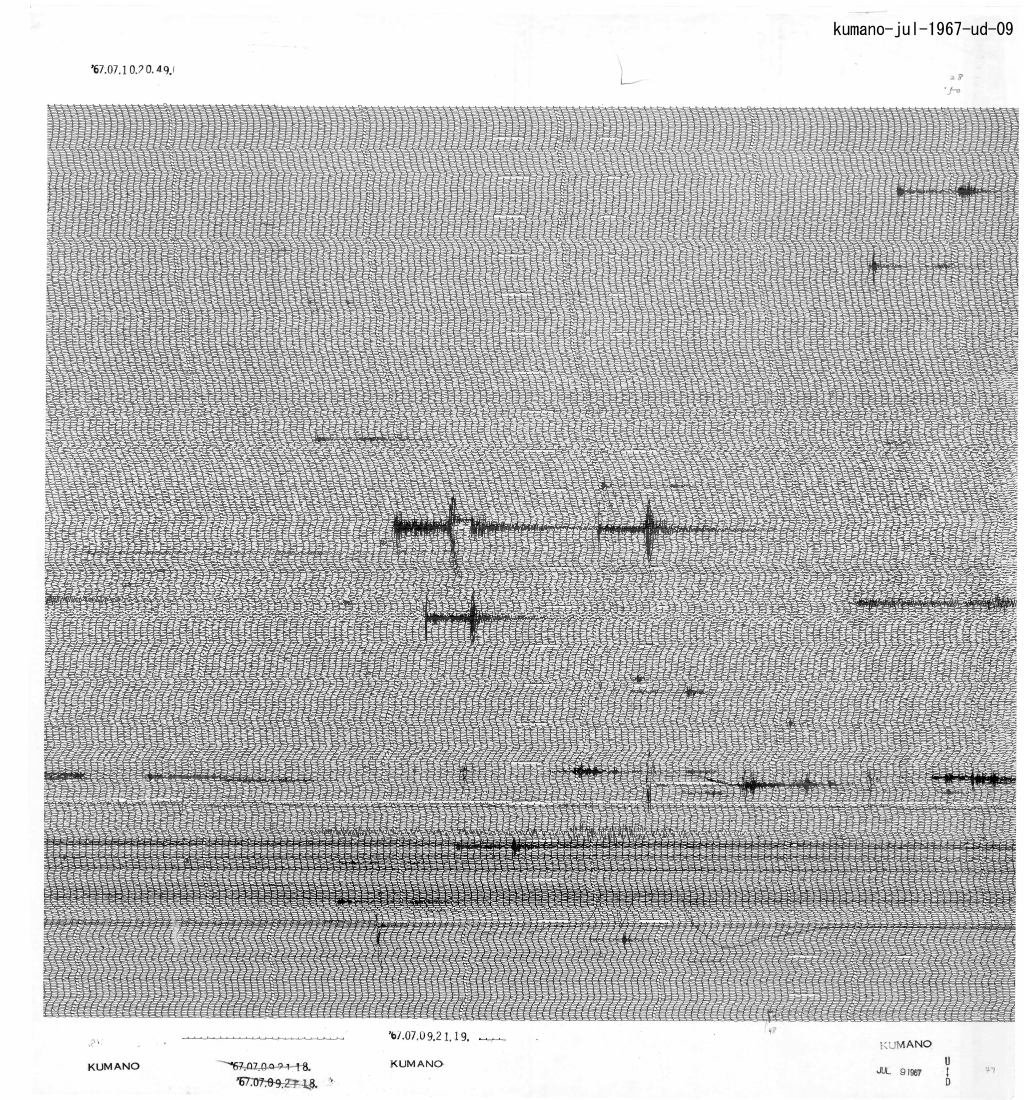

Figure 2. The PSI from 21:19 9 July 1967 to 20:49 10 July 1967 at KUM.

3 Datasets

More than 50 years ago, the seismic observation system drew waveforms directly on paper at each station location. The ERI archived seismograms recorded by the Wakayama seismological network, Kii Peninsula, Japan, and some of them have been scanned as TIFF images (Satake et al., 2020). Hereafter, we refer to the past seismogram image as PSI. This study uses 3,630 PSIs recorded at KUM from 1966 to 1977. Figure 2 shows an example of a PSI containing a number of continuous waveforms. The time length of each waveform drawn from left to right was approximately 2.5 minutes. One paper contains 500–600 waveforms stacked vertically from the bottom to the top in chronological order, corresponding to a continuous daily record. Artificial marks are inserted to indicate the time, *e.g.*, the triangles at every second and the steps at every 30 seconds. The times stamped at the bottom and top of the paper indicate the start and end times of recording, respectively. Each PSI is cropped and resized to a size of 7,000 × 7,000 pixels by removing the surrounding margins, and the grayscales are adjusted to reduce the effects of noise due to the scanning process on the original paper records as much as possible.

Digitization tracing the waveforms is effective for investigating ordinary earthquakes because their waveforms are extractable, even when a couple of waveforms overlap. In contrast, unrealistically high-precision tracing is required for tremors because of their small amplitudes and long durations. Visual detection by experts is also an effective method for detecting seismic phenomena such as earthquakes. However, such visual detection of tremors is unrealistic, unlike the case of ordinary earthquakes, because similar waveforms caused by such as passages of low atmospheric pressure and artificial noises are easily included. Another reason why visual detection should be avoided is that the number of available PSIs is too large, exceeding several tens of thousands. Therefore, we adopt a CNN-based image recognition as a promising alternative to detect tremors from PSIs.

Training the CNN with learning data is necessary to detect tremors from PSIs using a CNN. The PSIs would be ideal for the learning data if the hypocenters and occurrence times of tremors were known. However, a catalog of tremors before the establishment of Hi-net has not yet been published. As the alternative, this study uses two datasets for learning data: synthetic seismogram images (SSIs) and modern seismogram images (MSIs). The SSIs are artificial images containing fewer types of waveforms than the PSIs. The detailed explanations of SSIs are available in Kaneko et al. (2021). The SSIs comprise 100 images labeled as "none" that contain no tremors and another 100 images labeled as "tremor" that contain tremors. The MSIs are seismogram images converted from the Hi-net continuous waveform data (National Research Institute for Earth Science and Disaster Resilience, 2019). We referred to the tremor catalog published by NIED (Maeda & Obara, 2009; Obara et al., 2010) to select the dates and Hi-net observatories to be plotted as none or tremor images. The catalog contains information on events such as occurrence dates, hypocenters, and magnitudes of tremors in southwest Japan detected based on the Hi-net data from January 2001 to April 2019. For the tremor images, we require tremors to have magnitudes larger than 1.0 and hypocenters located at east of 135° E (Figure 3). We generate MSIs from the records of the three nearest stations to the epicenter of each event. For the none images, we selected dates where no tremors occurred for three days before and after, and then generated MSIs from the records of the Totsukawa-Nishi observatory (N.TKWH), which is located near KUM, to reduce the differences in observation environments as much as possible. We used all three components, *i.e.*, east-west (EW), north-south (NS), and up-down (UD) components, of the records to generate MSIs in the same manner as the SSIs. The MSIs consisted of 405 none images and 405 tremor images.

For preprocessing, we divided each image of the SSIs, MSIs, and PSIs vertically into five rectangle images, reducing the size to 2,000 × 400 pixels. This preprocessing has three advantages; (1) the number of learning data increases, such as data augmentation, (2) the small image reduces the number of model parameters and consequently saves computational cost, and (3) the vertical division can reduce the influence of temporary noise without affecting the identification of tremors with long durations. We carefully determined the reduced size of the images not to eliminate the features of the tremors. Hereafter, we call the square image before the preprocessing as the "original image" and the rectangle image after preprocessing as the "divided image."

4 Construction and Learning of CNN

We constructed our CNN based on ResNet, which adopts shortcut connections to realize residual learning and exhibites high performance with a simple structure (Figure 4). A residual block, the unit structure of ResNet, comprises convolutional layers with a filter size 2×2 followed by batch normalization layers and a shortcut connection. We combined residual blocks, pooling, and fully-connected layers to construct a model. A divided image was input into the CNN as a vector of pixel values with a size of 2,000 × 400. For each input $\boldsymbol{x}$, the CNN outputs two non-negative values, $\boldsymbol{y} = (p_\mathrm{n}, p_\mathrm{t})$, where p_n and p_t are the probability that $\boldsymbol{x}$ is a none and tremor image, respectively. Finally, the CNN

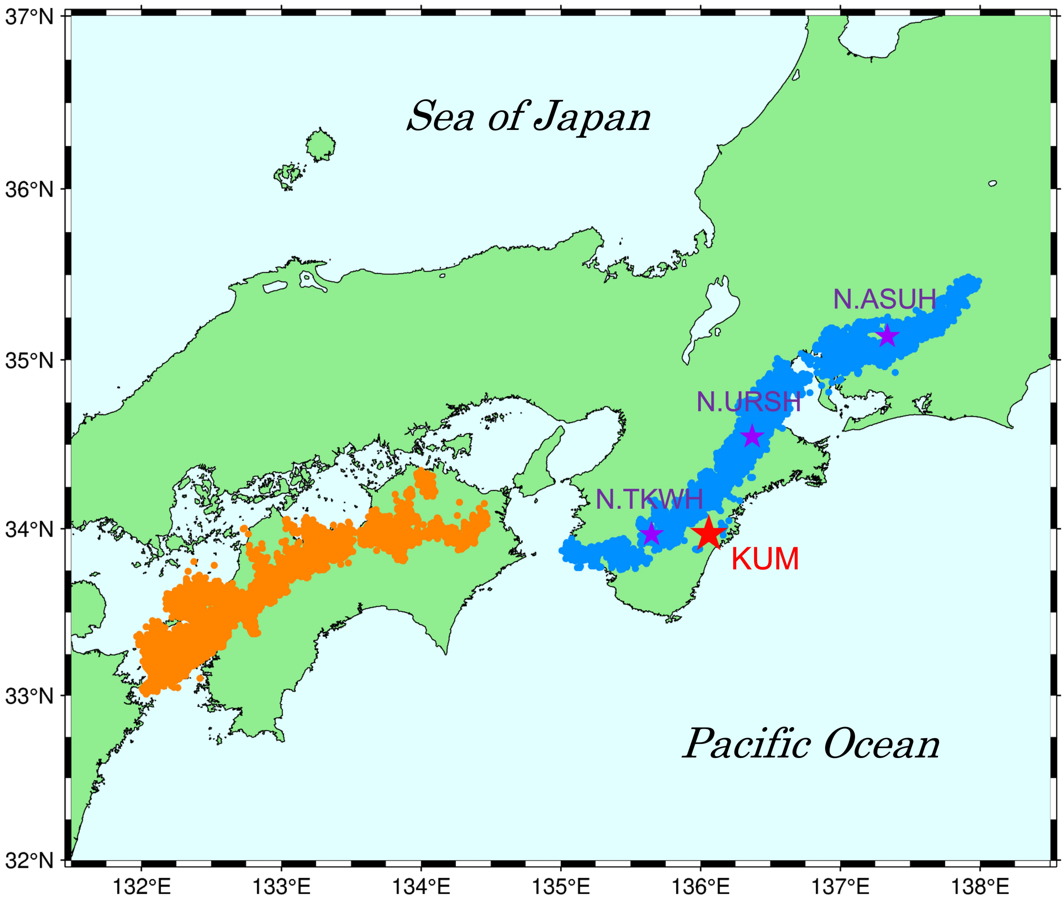

Figure 3. Epicenter distribution of the tremors listed in the NIED catalog. The blue and orange circles indicate the epicenters of the tremors that occurred east and west of 135° E, respectively.

determines whether $\boldsymbol{x}$ is none or tremor by comparing p_{n} and p_{t}. We used ReLU (equation (9)) as the activation function, except for the output layer in which a softmax function was used. Letting z_{n} and z_{t} be the values before passing through the softmax function, respectively, the outputs are represented as

$$p_{\mathrm{n}} = \frac{e^{z_{\mathrm{n}}}}{e^{z_{\mathrm{n}}} + e^{z_{\mathrm{t}}}} , \quad p_{\mathrm{t}} = \frac{e^{z_{\mathrm{t}}}}{e^{z_{\mathrm{n}}} + e^{z_{\mathrm{t}}}} . \tag{14}$$

Therefore, the softmax function ensures that p_{n} and p_{t} are non-negative and their sum is one. The hyperparameters, such as the fliter size of the convolutional layers, are determined carefully by comparing some cases. The number of model parameters was approximately one million.

We conducted learning of the CNN through fine-tuning. Fine-tuning is a popular learning method that uses a pre-trained model with another dataset for learning the target data instead of a model whose parameters are initialized randomly. It is known to improve model performance and learning efficiency. The effect of the fine-tuning has been discussed in previous studies (Kornblith et al., 2019; Yosinski et al., 2014). We trained the CNN with the SSIs and subsequently with the MSIs. In the following, we discribe the learning with the MSIs; see Kaneko et al. (2021) for the pretraining with the SSIs. The MSIs were split into 80% of data for training and the remaining 20% for validation. At every epoch, which is the unit of learning iterations, the CNN optimizes the param-

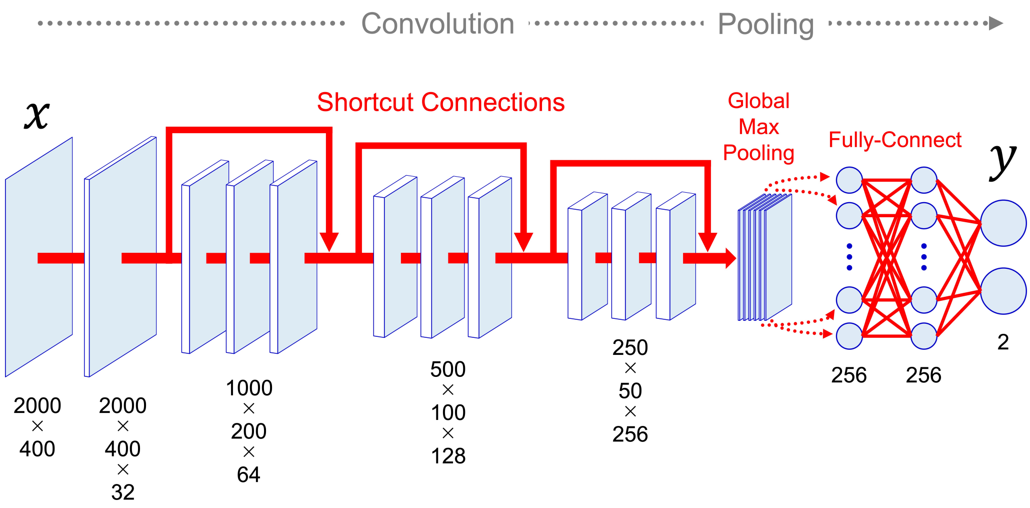

Figure 4. Structure of the proposed CNN. The numerical values indicate the size of vectors on the layers.

eters with the training data, and its performance is evaluated using the validation data. The iterations were repeated until the CNN performance converged. We used the cross-entropy loss as the loss function and optimized it using Adam with a batch size of 16. Letting $t = 0$ if $\boldsymbol{x}$ is a none image and $t = 1$ if $\boldsymbol{x}$ is a tremor image, the cross-entropy loss $L = L(\boldsymbol{y}; \boldsymbol{x})$ is defined as follows:

$$L(\boldsymbol{y}; \boldsymbol{x}) = -(1 - t) \log p_\mathrm{n} - t \log p_\mathrm{t} \ . \tag{15}$$

Adam is an improved stochastic gradient method that uses the momentum method and step-size tuning for fast convergence to the solutions. We calculated the loss and accuracy of both training and validation data at each epoch. Accuracy is calculated by considering that the outputs are correct if $p_\mathrm{n} > p_\mathrm{t}$ for none images and $p_\mathrm{n} < p_\mathrm{t}$ for tremor images. We generated heatmaps to visualize the model predictions based on the Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017). The Grad-CAM calculates the importance of each area in the input image based on the gradients of the output values with respect to values at the last convolutional layer. The heatmap using the gradient of p_t (tremor heatmap) shows the contribution to p_t. The tremor heatmap intuitively shows where the model focuses its attention on an input image to determine the existence of the tremors. After training with the SSIs and MSIs, we applied the CNN to the PSIs.

5 Results

5.1 Learning with MSIs

Figure 5 shows the variations in accuracy and loss during the learning with the MSIs. The metrics show high performance even at the initial stage of learning, which is considered an effect of fine-tuning. The accuracy and loss for the validation data finally converged at approximately 0.99 and 10^{-2}, respectively. Table 1 shows the prediction results of the CNN when the loss of the validation data was the lowest during learning. This shows that the CNN can almost certainly determine whether an MSI contains tremors or not. The p_t values are extreme, *i.e.*, either $p_\mathrm{t} \approx 0$ or $p_\mathrm{t} \approx 1$ for most of the divided MSIs.

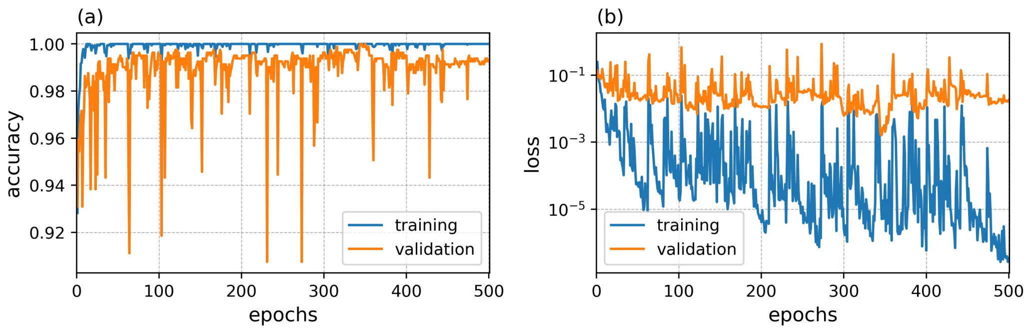

Figure 5. (a) Accuracy and (b) loss with respect to the number of epochs during the learning with the MSIs.

Figure 6 shows the original tremor MSI and tremor heatmaps for the divided images. The MSI (Figure 6(a)) contains various waveforms, *e.g.*, the tremor from 2:00 to 3:00, and earthquakes around 5:00, 15:00, and 23:00. The heatmaps (Figure 6(b)) show that the CNN outputs $p_\mathrm{t} \approx 1$ for all divided images and reacts strongly only to the tremor. Figure 7 shows another original tremor MSI and tremor heatmaps of the divided images. The MSI (Figure 7(a)) contains tremors around from 8:00 to 19:00. The heatmaps (Figure 7(b)) show reactions corresponding to the long duration of the tremors. The strength of the reactions seemed to vary according to the amplitudes of the tremors. An earthquake overlapping the tremors around 16:00 is considered to have little influence on the predictions of the CNN. These results for the tremor MSIs (Figures 6 and 7) indicate that we can precisely determine the occurrence times and durations of tremors from the heatmaps. Figure 8 shows the original none MSI and tremor heatmaps for the divided images. MSI (Figure 8(a)) does not contain tremors but some teleseisms at approximately 6:00, 13:00, and 15:00. The waveform of teleseism is sometimes similar to that of a tremor, depending on its magnitude and duration. It may cause a false positive for the CNN. The heatmaps (Figure 8(b)) show that the CNN does not misidentify the teleseisms as tremors outputting $p_\mathrm{t} \approx 0$ for all the divided images. This result indicates that the CNN was not affected by the teleseisms.

Table 1. *Prediction Summary for the MSIs*

	True: none	True: tremor	Accuracy
Prediction: none	2,023 (405)	43 (11)	97.92% (97.36%)
Prediction: tremor	2 (0)	1,982 (394)	99.90% (100.0%)
Accuracy	99.90% (100.0%)	97.88% (97.28%)	98.89% (98.64%)

Note. The values outside and inside of parentheses in each cell indicate the results for all data (training and validation data) and only validation data, respectively. The value in column “True: xxx” of row “Prediction: yyy” indicates the number of images with the label xxx predicted as yyy by the CNN. Each accuracy shows the percentage that the predictions of CNN are correct for images of the corresponding column or row. Especially, the bottom-right cell indicates the total accuracy.

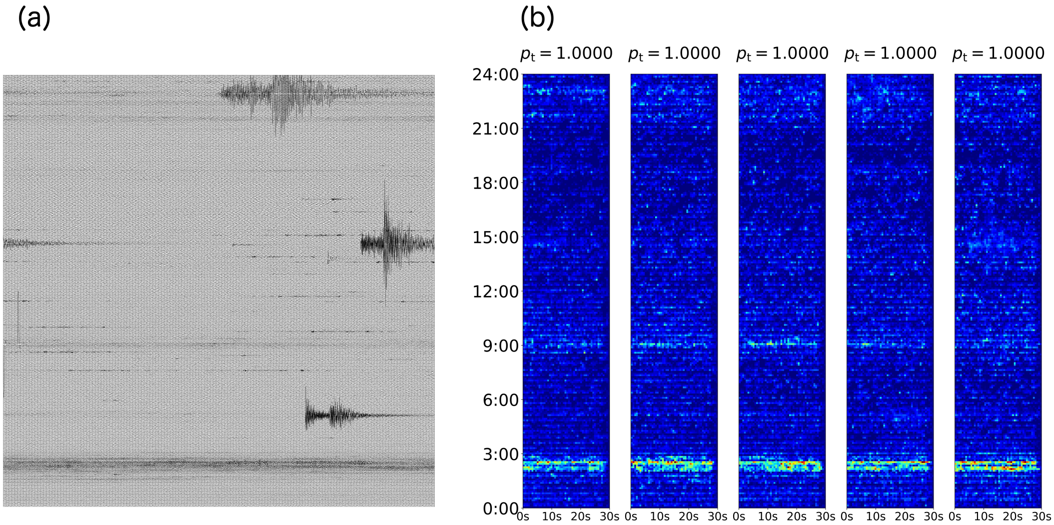

Figure 6. (a) The"tremor" MSI generated from the UD component at Ureshino observatory (N.URSH) on 24 April 2017, and (b) "tremor" heatmaps. The bright colors in (b) indicate the places in (a) on which the CNN strongly focused, while the blue means no interest. The p_{t} values for the divided images are shown above the heatmaps.

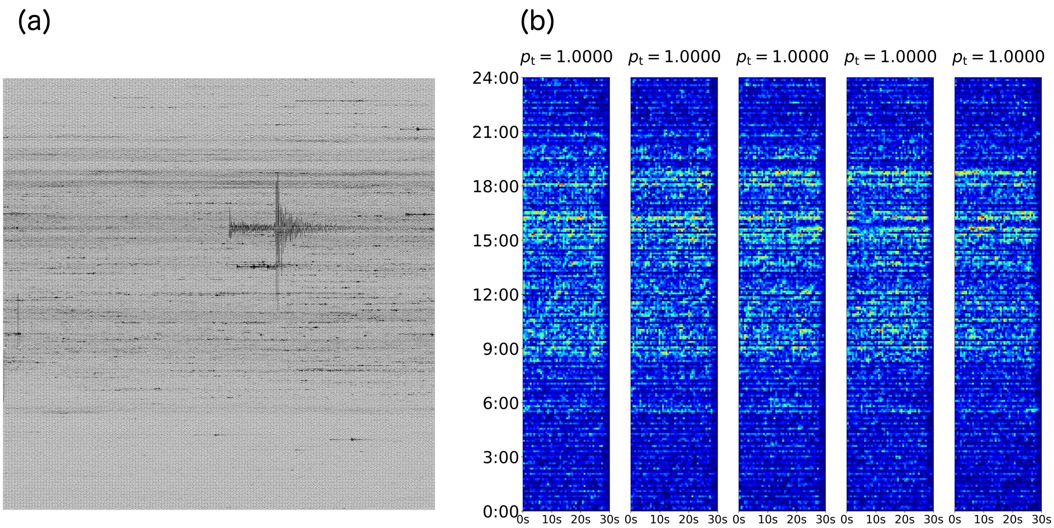

Figure 7. (a) The "tremor" MSI generated from the UD component at Asuke observatory (N.ASUH) on 3 September 2008, and (b) "tremor" heatmaps.

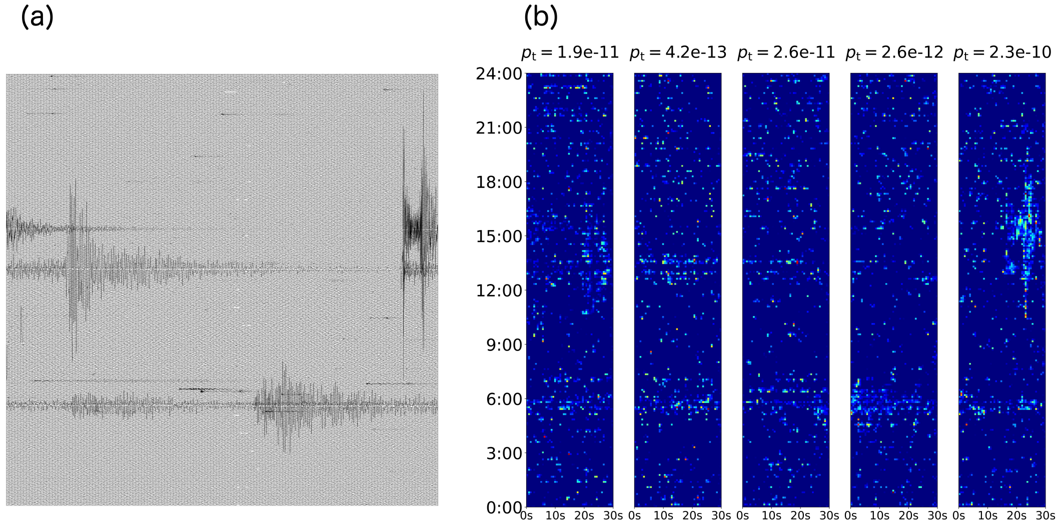

Figure 8. (a) The "none" MSI generated from the UD component at Totsukawa-Nishi observatory (N.TKWH) on 9 December 2001, and (b) "tremor" heatmaps.

5.2 Application to PSIs

Figure 9 shows the Gantt chart of the number of divided images for each day that are determined as tremors by the trained CNN. Deep red indicates that tremors are expected to have occurred with a high probability during this period. There are periods during which tremors are determined to have occurred intermittently, *e.g.*, from August 1974 to November 1974, and those during which tremors are determined to have occurred continuously, *e.g.*, from October 1971 to June 1972. We call these two periods as the "intermittent period" and "continuous period," respectively.

To verify the plausibility of the result, we compared the obtained detection frequency of tremors in the PSIs, *i.e.*, Figure 9, with the occurrence frequency recorded in one of the tremor catalogs widely accepted in the community. Figure 10 shows the Gantt chart indicating tremor occurrences recorded in the NIED catalog between January 2001 and April 2019 . Focusing on the intermittent periods, the detection frequency seems to be probable considering that the occurrence frequency of tremors determined in the NIED catalog is roughly a couple of times a month. Figures 11 and 12 show the original PSIs and corresponding tremor heatmaps for the divided images on September 19 and 17, 1974, respectively, both of which were included in an intermittent period. The PSI (Figure 11(a)) contains the events considered to be a tremor from 16:00 to 21:00 and a tremor with a short duration at approximately 10:00. The heatmaps (Figure 11(b)) show the reactions for both tremors with $p_\mathrm{t} = 1$ for all divided images. The PSI (Figure 12(a)) contains tremors ranging from 11:00 to 15:00. Although the second divided image from the left was determined as none ($p_\mathrm{t} = 0.1761$), the other four images were determined as tremor with large p_t values (Figure 12(b)). This result supports the importance of preprocessing to divide an original image vertically. It reduces the risk of missing tremors by integrating the predictions for multiple samples generated from one image. These results verify that the CNN successfully detects tremors from PSIs in intermittent periods without reacting to the waveforms contaminated by noise.

Next, we focused on tremor heatmaps in the continuous periods. The tremor heatmaps show strong reactions across the entire image, outputting $p_\mathrm{t} \approx 1$ for all of the divided images, although the original images obviously do not contain tremors. We pursued why

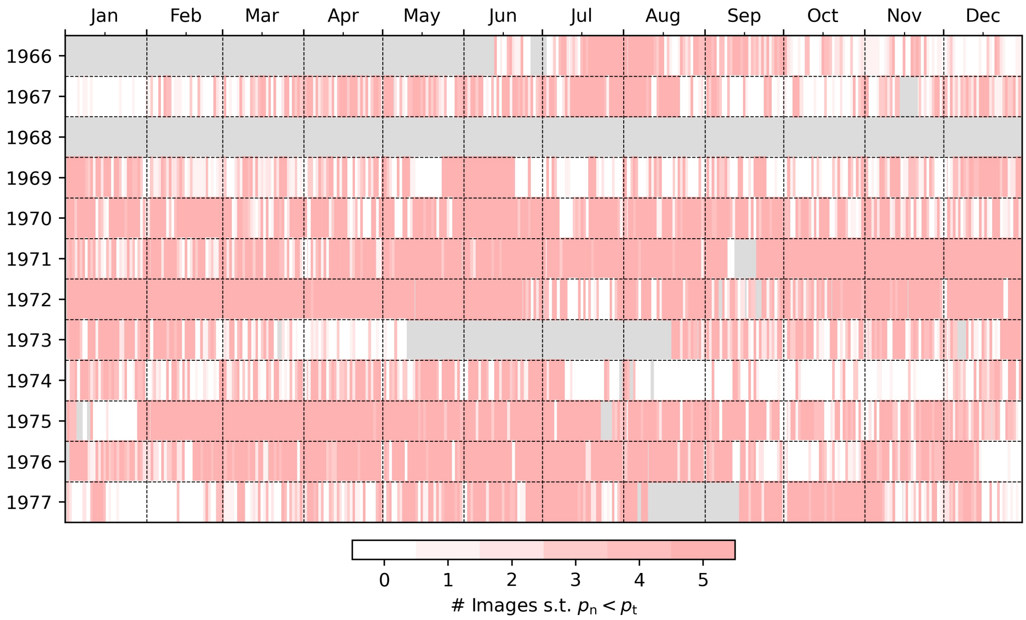

Figure 9. Gantt chart for predictions to the PSIs. The depth of red represents the number of divided images such that $p_{\mathrm{n}} < p_{\mathrm{t}}$ holds. The gray area represents that the PSIs are missing in the period.

the CNNs work well for the PSIs in the intermittent periods but not the continuous periods. One possible cause is the thickness of the plotting pen. Figure 13 shows the enlarged waveforms of Figures 11(a) and one of the PSIs in the continuous period. The waveform lines in the continuous periods were thicker (Figure 13(b)) than those in the intermittent periods (Figure 13(a)). This indicates that the CNNs performed well when the waveform lines were thin, which is probably coincident with the thickness of the waveforms in the training dataset. The reason why the thickness in the PSIs changed so drastically with time is uninvestigable.

Augmentation of learning data is the most practical and effective method to construct a CNN universally applicable to any seismogram images that overcomes the problem of the thickness of waveform lines, For additional learning data, only the generation of MSIs with thicker waveform lines is considered sufficient for the first step. In addition, the MSIs in which the recording directions of waveforms are intentionally changed, *i.e.*, right to left, are to be learned because the PSIs sometimes include upside-down images, as shown in Figure 13(a). This augmentation is expected to increase the amount of data several times to dozens of times, which requires a much higher computational cost. We are considering installing GPUs with higher computational performance than the current ones to realize learning with the augmented datasets.

6 Conclusions

We proposed a CNN-based method to detect tremors from seismogram images and evaluated its performance on the MSIs and PSIs. We confirmed that the CNN trained by fine-tuning successfully detected tremors from the MSIs. The CNN trained with the MSIs showed sufficient potential to detect tremors from the PSIs, while its performance depended on the width of waveform lines. Augmentation of the learning data is consid-

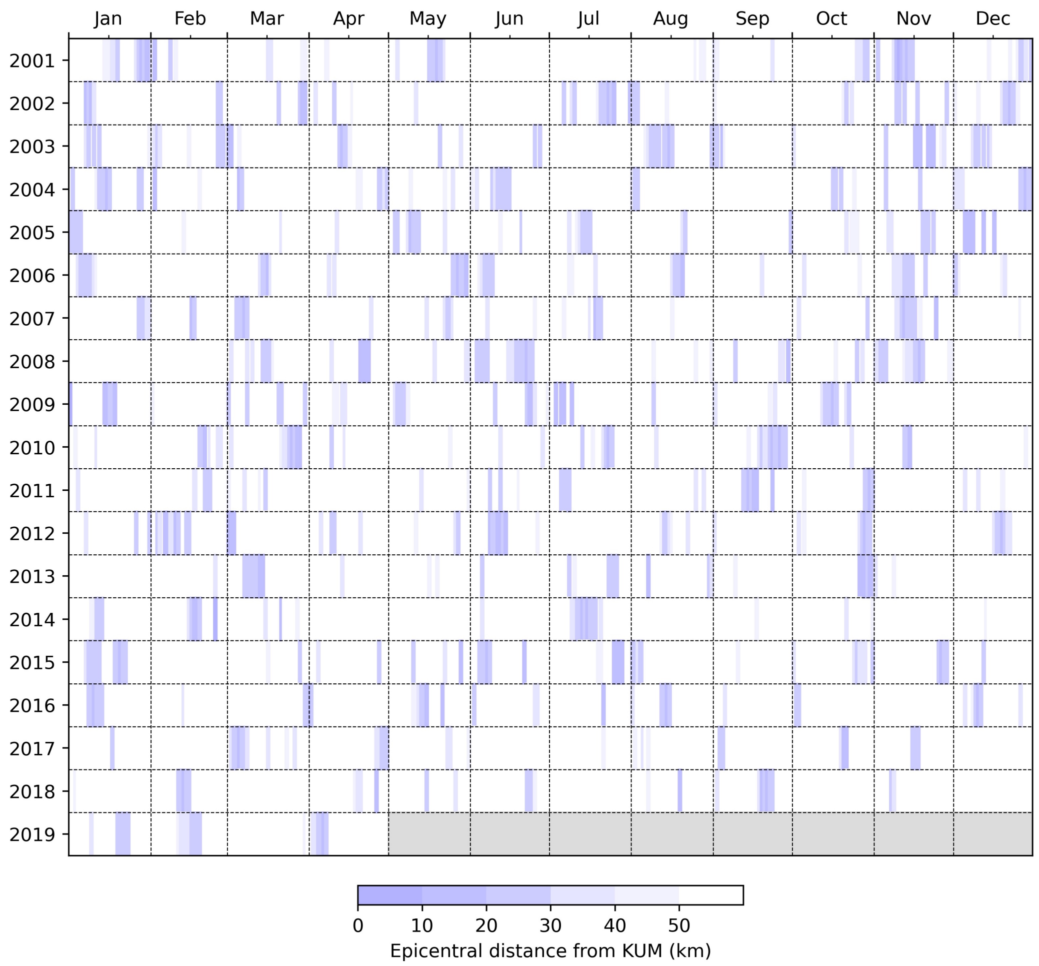

Figure 10. Gantt chart for tremors listed in the NIED catalog. The deep blue indicates the short epicentral distance from KUM implying that the tremors are probably observable at KUM.

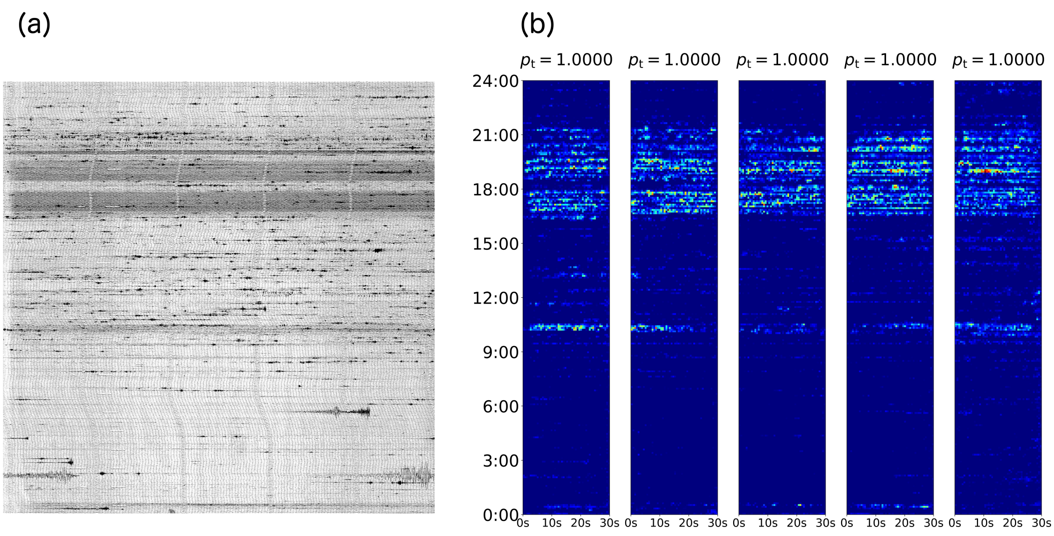

Figure 11. (a) The PSI from 21:29 19 September 1974 to 20:44 20 September 1974, and (b) "tremor" heatmaps.

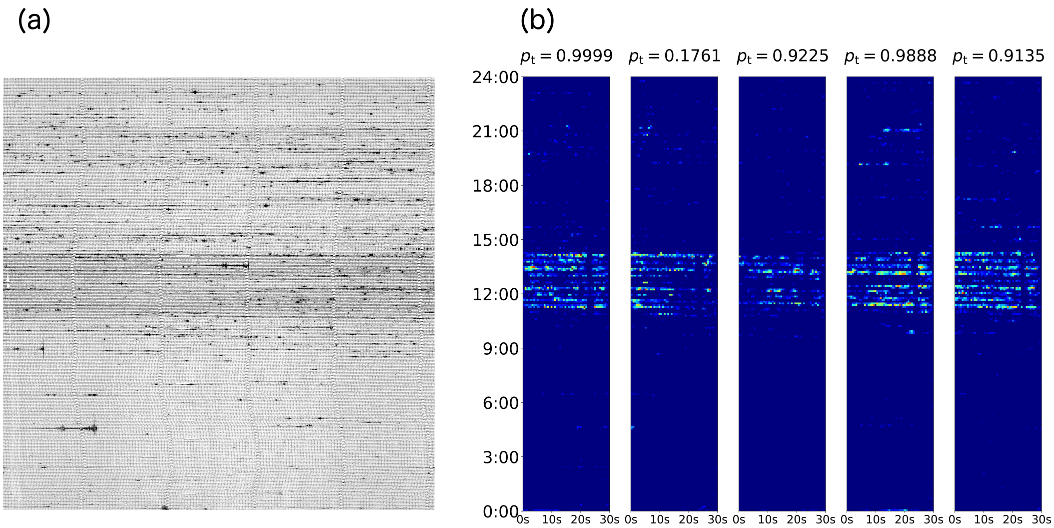

Figure 12. (a) The PSI from 21:02 17 September 1974 to 20:42 18 September 1974, and (b) "tremor" heatmaps.

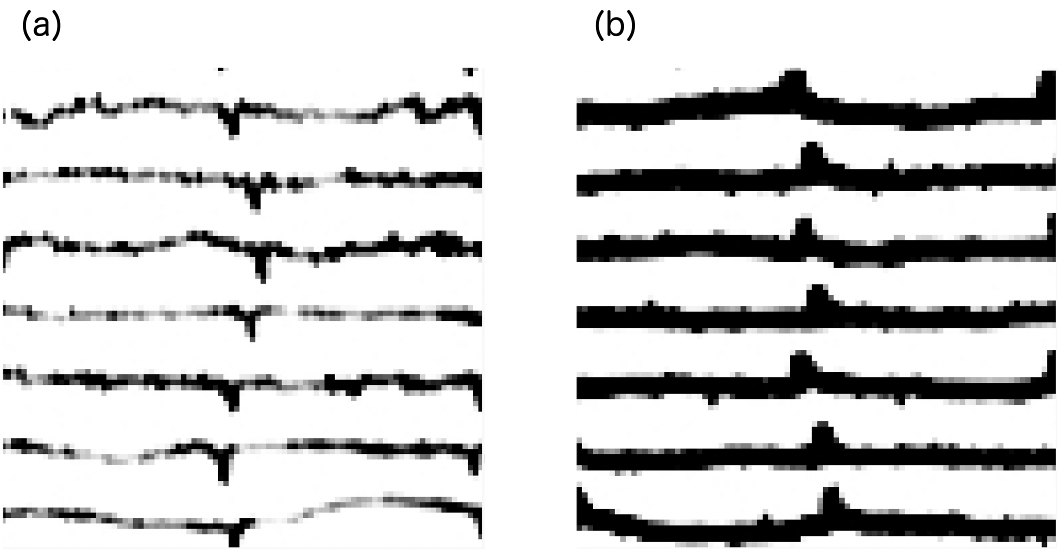

Figure 13. Enlargement of the waveforms in (a) Figure 11(a) and (b) the PSI on 9 January 1972.

ered effective at improving the adaptability of the CNNs for PSIs, as discussed in the Section 5.2. It should be noted that it is difficult to validate whether the waveforms detected by the CNN are tremors. A concrete strategy at this moment is to verify the candidate tremor images through the eyes of experts who have much experience in detecting tremors based on seismic waveforms. If the CNN achieves sufficiently high performance, it can be used in the PSIs of seismic observatories other than KUM. The accuracy of tremor detection can be improved by using seismograms in neighboring stations simultaneously. Subsequently, publishing a new catalog that contains tremors in the past is possible, which undoubtedly contributes to seismology.

Acknowledgments

The past seismogram data used in this study are available via the website (`http://wwweic.eri.u-tokyo.ac.jp/wakayama/`). TensorFlow (Abadi et al., 2015) was used to construct a CNN and its learning. OpenCV (OpenCV, 2015) was used to generate and preprocess the seismogram images. Matplotlib (Hunter, 2007) and PyGMT (Uieda et al., 2022) were used to create the figures presented herein.

This study was supported by MEXT Project for Seismology toward Research Innovation with Data of Earthquake (STAR-E) Grant Number JPJ010217. The key ideas in this study were derived from the activities of JST CREST under Grant Numbers JPMJCR1763 and JPMJCR1761, JSPS KAKENHI Grant-in-Aids for Scientific Research (B) No. 17H01703, 17H01704, 18H03210, Grant-in-Aid for Scientific Research (S) No. 19H05662, Grant-in-Aid for Challenging Research (Exploratory) No. 20K21785, and Earthquake Research Institute Joint Research ERI JURP 2022-A-03, 2021-B-01, and 2022-B-06.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from `https://www.tensorflow.org/` (Software available from tensorflow.org)

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(61), 2121-2159.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*, 193-202.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 ieee conference on computer vision and pattern recognition* (p. 770-778). doi: 10.1109/CVPR.2016.90

Hubel, D., & Wiesel, T. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, *160*, 106-154.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on international conference on machine learning* (Vol. 37, p. 448–456).

Kaneko, R., Nagao, H., Ito, S., Obara, K., & Tsuruoka, H. (2021). Convolutional neural network to detect deep low-frequency tremors from seismic waveform images. In *Lecture notes in computer science* (Vol. 12705, p. 31-43). doi: 10.1007/978-3-030-75015-2_4

Kano, M., & Kano, Y. (2019, 9). Possible slow slip event beneath the Kii Peninsula, southwest Japan, inferred from historical tilt records in 1973. *Earth, Planets and Space*, *71*. doi: 10.1186/s40623-019-1076-9

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do better ImageNet models transfer better? In *2019 ieee/cvf conference on computer vision and pattern recognition* (p. 2656-2666). doi: 10.1109/CVPR.2019.00277

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems* (Vol. 1, p. 1097–1105).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541-551. doi: 10.1162/neco.1989.1.4.541

Maeda, T., & Obara, K. (2009). Spatiotemporal distribution of seismic energy radiation from low-frequency tremor in western Shikoku, Japan. *Journal of Geophysical Research: Solid Earth*, *114*(B10). doi: 10.1029/2008JB006043

Misra, D. (2020). Mish: A self regularized non-monotonic activation function. In *31st british machine vision conference.*

Nagai, R., Kikuchi, M., & Yamanaka, Y. (2001). Comparative study on the source processes of recurrent large earthquakes in sanriku-oki region: the 1968 tokachi-oki earthquake and the 1994 sanriku-oki earthquake. *Zisin (Journal of the Seismological Society of Japan. 2nd ser.)*, *54*(2), 267-280. doi: 10.4294/zisin1948.54.2_267

Nakano, M., Sugiyama, D., Hori, T., Kuwatani, T., & Tsuboi, S. (2019). Discrimination of seismic signals from earthquakes and tectonic tremor by applying a convolutional neural network to running spectral images. *Seismological Research Letters*, *90*(2A), 530-538. doi: 10.1785/0220180279

National Research Institute for Earth Science and Disaster Resilience. (2019). *NIED Hi-net, National Research Institute for Earth Science and Disaster Resilience.* doi: 10.17598/NIED.0003

Obara, K. (2002). Nonvolcanic deep tremor associated with subduction in southwest Japan. *Science*, *296*(5573), 1679-1681. doi: 10.1126/science.1070378

Obara, K., & Kato, A. (2016). Connecting slow earthquakes to huge earthquakes. *Science*, *353*(6296), 253-257. doi: 10.1126/science.aaf1512

Obara, K., Tanaka, S., Maeda, T., & Matsuzawa, T. (2010). Depth-dependent activity of non-volcanic tremor in southwest Japan. *Geophysical Research Letters*, *37*(13). doi: 10.1029/2010GL043679

Okada, Y., Kasahara, K., Hori, S., Obara, K., Sekiguchi, S., Fujiwara, H., & Yamamoto, A. (2004). Recent progress of seismic observation networks in Japan —Hi-net, F-net, K-NET and KiK-net—. *Earth, Planets and Space*, *56*(8), xv–xxviii. doi: 10.1186/BF03353076

OpenCV. (2015). *Open source computer vision library.*

Perol, T., Gharbi, M., & Denolle, M. (2018). Convolutional neural network for earthquake detection and location. *Science Advances*, *4*(2). doi: 10.1126/sciadv.1700578

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). *Searching for activation functions.*

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of Adam and beyond. In *6th international conference on learning representations.*

Rouet-Leduc, B., Hulbert, C., McBrearty, I. W., & Johnson, P. A. (2020). Probing slow earthquakes with deep learning. *Geophysical Research Letters*, *47*(4). doi: 10.1029/2019GL085870

Satake, K., Tsuruoka, H., Murotani, S., & Tsumura, K. (2020). Analog seismogram archives at the Earthquake Research Institute, the University of Tokyo. *Seismological Research Letters*, *91*(3), 1384-1393. doi: 10.1785/0220190281

Schwartz, S. Y., & Rokosky, J. M. (2007). Slow slip events and seismic tremor at circum-Pacific subduction zones. *Reviews of Geophysics*, *45*(3). doi: 10.1029/2006RG000208

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 ieee international conference on computer vision* (p. 618-626). doi: 10.1109/ICCV.2017.74

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929-1958.

Uieda, L., Tian, D., Leong, W. J., Jones, M., Schlitzer, W., Grund, M., ... Wessel, P. (2022, April). *PyGMT: A Python interface for the Generic Mapping Tools.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.6426493` doi: 10.5281/zenodo.6426493

Yano, K., Shiina, T., Kurata, S., Kato, A., Komaki, F., Sakai, S., & Hirata, N. (2021). Graph-partitioning based convolutional neural network for earthquake detection using a seismic array. *Journal of Geophysical Research: Solid Earth*, *126*(5). doi: 10.1029/2020JB020269

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Proceedings of the 27th international conference on neural information processing systems* (Vol. 2, p. 3320–3328).

Zhu, W., & Beroza, G. C. (2018). PhaseNet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, *216*(1), 261-273.