Rock Mechanics and Rock Engineering (2021) 54:47–69
https://doi.org/10.1007/s00603-020-02259-0

ORIGINAL PAPER

Identification of Microseismic Events in Rock Engineering by a Convolutional Neural Network Combined with an Attention Mechanism

Shibin Tang[1] · Jiaxu Wang[1] · Chunan Tang[1]

Received: 21 February 2020 / Accepted: 22 September 2020 / Published online: 1 October 2020

Abstract

Microseismic technology has widely been used in many rock engineering applications to shield workers from engineering hazards and monitor underground construction. To avoid the heavy workloads imposed by the manual recognition of many microseismic signals, this study proposes a new end-to-end training network architecture to automatically identify microseismic events. A dataset including not only easily identifiable microseismic signals but also barely distinguishable nontypical data has been collected from a practical rock engineering project for training and testing the network model. The applicability of various networks for this task is discussed to select the best method for microseismic recognition. We modify the residual skip connections to make them more suitable for the signal classification task. Then, the novel depthwise spatial and channel attention (DSCA) module is proposed. This module can autonomously learn how to weight information with different levels of importance, similar to human attention, which greatly improves the network performance without incurring additional computational costs. Theoretically, it can be a useful tool to replace traditional denoising algorithms and model the interdependencies between the different channels of a multichannel signal. Furthermore, the DSCA module and the modified residual connections are combined with a traditional convolutional network to obtain a novel network architecture named ResSCA and the results of comparative experiments are presented. Finally, single- and multichannel models are constructed based on ResSCA, which achieved improved accuracy rates. Their advantages and drawbacks are analyzed. This study presents a modified network architecture suitable for identifying and classifying complex signals to enable intelligent microseismic monitoring, which is valuable for various rock engineering applications.

Keywords Microseismic monitoring · Signal recognition algorithm · Convolutional neural network · Attention mechanism · Deep learning

1 Introduction

The construction of deep and long tunnels is a major task in water diversion, transportation and other rock engineering applications around the world. Various rock engineering hazards such as rockbursts, soft rock deformation, and harmful gases have been frequently encountered during construction, and these hazards severely threaten the safety of personnel and equipment. Recently, microseismic techniques have gradually been widely adopted for practical monitoring and early warning technology; in particular, such techniques are widely used for various rock engineering applications in China, such as projects involving rock slopes, hydropower stations and high concrete arch dams (Dong et al. 2019; Ma et al. 2015; Zhuang et al. 2019). In addition, microseismic analysis has proven to be a very useful tool for understanding underground rock failure processes since the twentieth century (Ge et al. 2009; Ghosh and Sivakumar 2018; Milev and Spottiswoode 2002; Urbancic and Trifu 2000). A monitoring system is placed in an appropriate place from which microseisms can be detected and located. Sensors acquire signals that are generated by the release of seismic energy from a microseism and then mathematical algorithms are used to locate the event based on classified and labeled data acquired manually from the P-wave and S-wave arrivals detected by the sensors. The processing of microseismic data can be

✉ Shibin Tang
Tang_Shibin@dlut.edu.cn

[1] State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China

simply divided into three steps: waveform recognition, determination of a microseism's arrival, and event location.

In real rock engineering, data processing has mainly been conducted by microseismic experts and engineers. However, manual processing is time consuming and the discrimination ability greatly depends on the engineers' experience. Therefore, many automatic algorithms for event classification (Paul et al. 2015; Zhao et al. 2015) and arrival picking (Akram and Eaton 2016; Guo et al. 2011; Lee et al. 2017; Li et al. 2018; Song et al. 2010) have been developed to replace manual processing. Automatic arrival picking algorithms have seemed to attract more attention from researchers than recognition algorithms. Nevertheless, a robust recognition algorithm can accurately screen out events to be located and improve the efficiency of the subsequent data processing, such as arrival picking. Currently, several frequently used classification algorithms such as spectrum-based analysis methods, traditional machine learning algorithms and ANN-based models have achieved good results on academic datasets. However, most academic datasets contain only typical microseismic signals and easily distinguishable noise signals that are easy to classify. In real rock engineering construction and exploration, microseismic signals are often multichannel signals that are mixed with various types of background and stationary noise (Alvarez et al. 2013). These traditional algorithms cannot perform well when dealing with such nontypical and multidimensional data. Although some denoising algorithms have been developed by researchers (Liang et al. 2014; Rodriguez et al. 2012), using these noise reduction algorithms might result in underutilization of the original data and loss of information. Hence, the existing automatic methods cannot completely replace manual discrimination and there is a need to develop more powerful methods.

Microseismic signals are essentially one-dimensional time series data. Traditional time series analysis and machine learning methods for processing such data mainly depend on the implementation of feature engineering (selecting a few features, such as the peak and frequency to replace an entire signal). In other words, they are affected by the skill and experience of the engineers, and thus, the raw waveform data may not be fully utilized. As an alternative, a well-designed artificial neural network (ANN) can reasonably extract a large number of high-level features from the original data without any human intervention, thus making full use of the waveform information. Although previous researchers have tended to choose certain parameters to use in place of waveforms as the input to a neural network for classification (Dai and Macbeth 2007; Wang and Teng 1995; Zhao and Takano 1999), with the development of deep learning algorithms and the improvement of computer hardware capabilities, the original waveform data themselves can now be used as the neural network input, thereby avoiding the loss of information caused by manual feature selection. In addition, in the era of Industry 4.0, construction sites typically generate massive amounts of data every day. As the amount of available data grows, conventional machine learning algorithms are gradually reaching the upper limits of their capabilities. However, ANNs, as a data-driven product of the big data era, have almost no upper limits as long as the model architecture is suitable for the problem. In other domains of industrial production, ANNs have already entered common use. To handle problems in different fields, different neural network architectures have been designed and developed. For example, Gated Bidirectional Convolutional Network (Zeng et al. 2016) were specifically created for object detection tasks. Li (Li et al. 2019) developed a stereo region-convolutional network to detect 3D objects, which is very useful in autonomous driving. Google launched the new Transformer framework to replace traditional architectures for machine translation tasks and achieved good results (Vaswani et al. 2017). In the rock engineering field, some researchers have attempted to use neural networks to achieve intelligent monitoring. Many illustrative experiments have been conducted to compare the performance of neural networks and traditional methods, and the designed networks have performed well on the corresponding datasets (Lin et al. 2019; Shang et al. 2017; Wilkins et al. 2020). These studies' achievements demonstrate the feasibility and great potential of using neural networks in the field of rock engineering.

Nevertheless, there are some improvements that are needed in the network architecture currently used to process microseismic data. First, some network architectures are relatively simple and perform well only for simple data, whereas they show limited performance for complex datasets containing a large number of nontypical waveforms. In real-world engineering, due to environmental interference, such nontypical data are more common. In addition, the signals are often interrelated multichannel signals, which require more computing resources to process than single-channel signals. Second, to improve the ability of a neural network to handle more complex microseismic signals, engineers often choose to simply add more layers to the network, which will introduce problems. Simply stacking layers can lead to problems of bottlenecking and degradation (He et al. 2016) as well as gradient vanishing/explosion (Bengio et al. 1994; Glorot and Bengio 2010). Moreover, each additional layer leads to considerable growth in the number of parameters and increased computational complexity. However, in this study, we proved that such a complex structure is not actually needed to complete the task of waveform classification and simply deepening the network to improve its representation ability will result in excess computational costs. Finally, some researchers have combined neural networks with other mathematical algorithms to perform predenoising (Dai and Macbeth 2007) or to model the correlations and interdependence between multichannel signals (Lin et al. 2019).

However even if good results are achieved, the usability, and transfer ability of such a model may be reduced, increasing the difficulty of training. A more straightforward and easy-to-use method of achieving these goals is proposed in this study.

In this study, we first present a dataset consisting of the most complex data from the Hanjiang-to-Weihe River Diversion Project in China. It contains not only typical microseismic waveforms but also mostly nontypical microseismic waveforms and highly deceptive noise signals. Then, the basic type of neural network that is probably the most suitable for recognizing microseismic events is discussed. When one of the most popular network architectures that is commonly used in industrial production is reproduced and applied to our dataset, the results show the problems described above. To solve and avoid these problems, we propose several improvements to the residual blocks and convert them into the one-dimensional version, making them very easy to combine with the novel network module proposed in this study. The new network module is called the depthwise spatial and channel attention (DSCA) module; its aim is to improve the performance of a network model by refining the intermediate data flow in the network without increasing the number of layers. Furthermore, an attention module is used in place of the traditional denoising algorithms and can be used to model the interdependence among the different channels of a multichannel signal. This “lightweight” module can easily be inserted into many popular convolutional neural network (CNN) architectures, computing in parallel and improving the network performance without introducing additional parameters or incurring additional computational costs. Finally, experimental investigations are reported to demonstrate the reliability and potential of this network architecture in practical engineering applications. To test the application of the proposed architecture in engineering practice, we specifically trained a multichannel recognition model for application to the Hanjiang-to-Weihe River Diversion Project.

2 Data Description

2.1 Introduction to the Project

The microseismic monitoring data used in this study were all obtained from the Hanjiang-to-Weihe River Diversion Project, the purpose of which is to solve the problems of water shortages in the northwestern part of China. This project has a great impact on the overall planning of the South-to-North Water Transfer Project in China. It is located in Shanxi Province, crossing the Yellow River, the Yangtze River Basin and the Qinling Barrier. It has three major components: the Golden Gorge Water Conservancy Hub, the Qinling Water Conveyance Tunnel, and the Sanhekou Water Control Project, as shown in Fig. 1. The main purpose of the project is to transfer excess water from the Han River basin to Guanzhong’s water-scarce areas through the construction of water conservancy hubs and water transfer tunnels. Through the construction of two main reservoirs, Jinxia and Sanhekou, in the Hanjiang River Basin, coscheduled water sources will be directed to the Hanjiang–Weihe water conveyance tunnel, and finally enter the Huangchigou water transfer project to supply water to the Guanzhong area. The tunnel under construction has a large burial depth, high in situ stress, and a long length. Furthermore, during the construction process, a series of geological engineering problems such as rockbursts, water and mud inrush, high ground temperature, soft rock deformation, radioactivity and harmful gases occurred. Therefore, the microseismic monitoring technology is adopted for geological advanced predictions. Compared with other projects in China, the collaborative data processing by engineers and machines has greatly improved the efficiency.

2.2 Motivation for the Dataset and Its Composition

Large amounts of monitoring data are generated every day, and engineers can predict the conditions of underground structures by observing and analyzing these data. However, this task is very time consuming and labor intensive even for experienced interpreters. In addition, the monitoring data are always influenced by various types of noise due to the complex construction environment (Alvarez et al. 2013; Liang et al. 2014; Rodriguez et al. 2012). Even experienced engineers need to carefully scrutinize those complex data multiple times. Although some automatic recognition algorithms have been developed that perform well on standard datasets, they are often misled by various external factors, such as noise and signal strength. As a result, it is very important to develop new models that can efficiently classify nontypical microseismic data. To train such neural network, a dataset needs to be created that contains various complex microseismic waveforms and counterexamples, with coverage of the types of data that may be encountered that is as comprehensive as possible. Then, the neural network will endeavor to recognize the underlying relationships in these labeled data through a process that mimics the operation of the human brain. Once such a machine learning model has been trained on an initial dataset, another dataset is needed to test it. To this end, a large number of different types of waveform data are required.

Typical microseismic waveforms (such as those depicted in Fig. 2) are included in the dataset. These signals are very pure, with high signal-to-noise ratios. They are largely undisturbed by noise. In addition, the arrival times of the pressure waves and shear waves, which are important

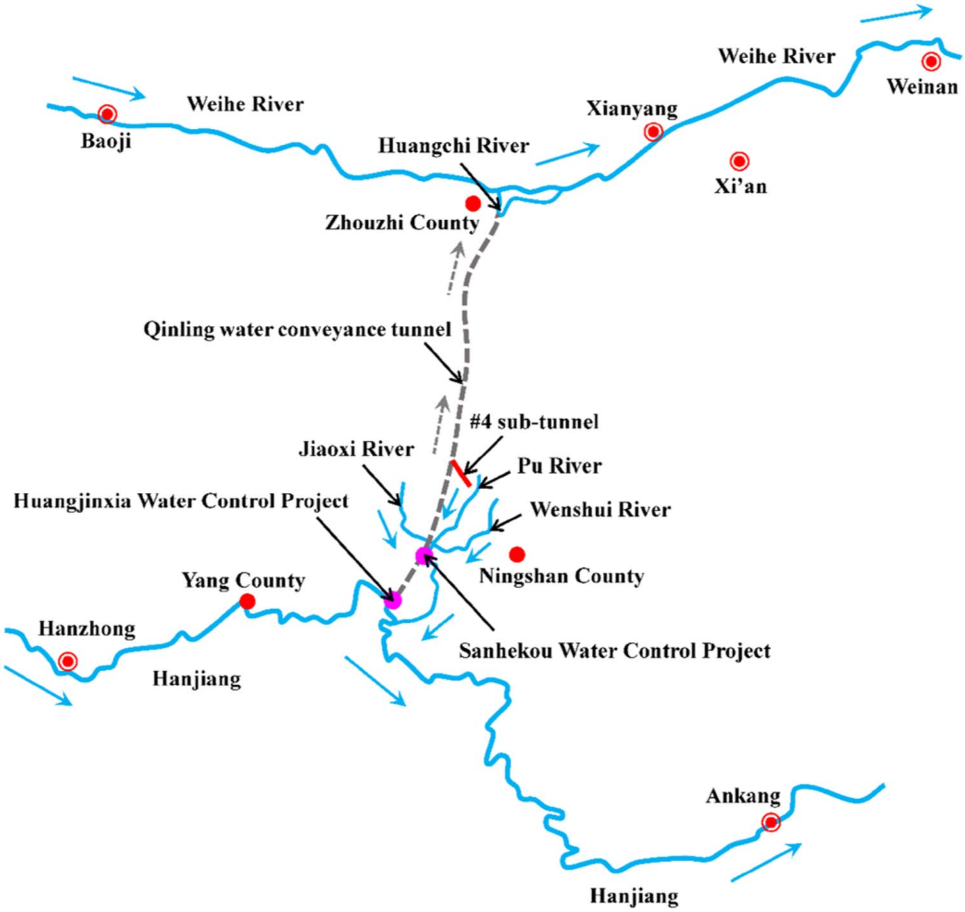
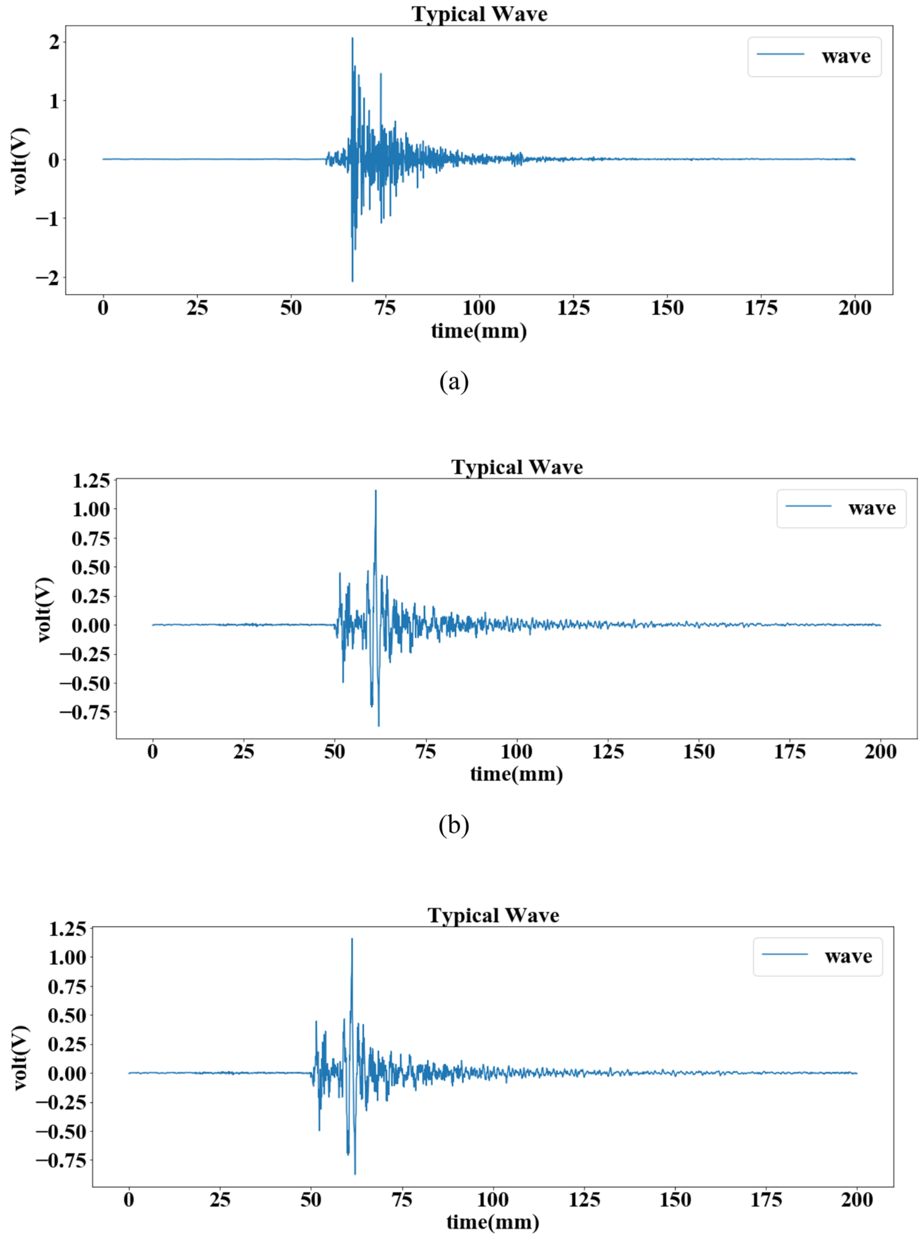


Fig. 1 The Hanjiang-to-Weihe River Diversion Project for addressing water shortages in the northwestern part of China (Liu et al. 2019)

features of microseismic signals, are very clear. Traditional algorithms perform well on such data, and classifying them is relatively easy for both human beings and machines. More importantly, some types of barely distinguishable waveforms have also been selected manually from a large amount of recorded data and included in our dataset. The method used to create the dataset is described below:

To create the two subsets (for training and testing) of our dataset, we first conducted many field experiments using manual knocking, blower vibrations and human voices, and the corresponding signals were acquired. We added the collected signals to the engineering database, which contains some previously observed microseismic waveforms triggered by rock failures. Human microseismic experts then examined the recorded data to identify events, and the experts made their decisions by comparison against waveforms representing definitive events recorded in the engineering database. During the process of manual events identification, we observed that the human experts sometimes based their decisions on the signals from all sensors and not just on one channel. This is because when a microseismic event occurs, it usually triggers multiple sensors, and the monitoring system records a corresponding multichannel waveform. For example, a six-channel waveform is presented in Fig. 3. When looking only at the second waveform, it is difficult to say whether it was triggered by a rock fracture. However, when all six channels are considered simultaneously, it is clear that they were triggered by a microseism. Similar observations were made by Wilkins et al. (2020) who noted that humans can simultaneously identify various general characteristics.

On this basis, we spent considerable time selecting many barely distinguishable waveforms for use in model training. Figure 4 presents some examples of such nontypical waveforms. Figure 4a, b show examples with low signal-to-noise ratios. Figure 4c present a segment of highly deceptive noise that is incorporated into the microseismic waveform, which can strongly affect the discrimination of machines. A more difficult case is depicted in Fig. 4d, which includes not only highly deceptive noise but also a low signal-to-noise ratio. Moreover, specific types of noise, such as waveforms containing blower vibration (Fig. 5a), manual knocking (Fig. 5b) or current interference (Fig. 5c), are also somewhat similar to microseismic events, which means they are more likely than ordinary noise to mislead machines, although human experts can easily distinguish them. We collected as many examples of these types of noise as we could and uniformly labeled them as the negative class to enable neural



(a)

(b)

(c)

Fig. 2 Examples of typical microseismic waveforms

networks to explore their differences from the positive class (waveforms containing microseismic events). Furthermore, the final model should be able to determine whether a snippet of a waveform in fact contains a microseismic event; therefore, a variety of waveforms were collected in the dataset to enable the models to learn from as many potential situations as possible.

The original waveforms were exported in two-dimensional coordinates from the monitoring system, where the abscissa represents time and the ordinate represents voltage. We retained only the voltage values to form time series data because the time intervals of these data are equal. We chose to provide these time series directly to the neural network, instead of selecting only certain parameters to replace a whole signal, because such artificial preprocessing and denoising algorithms may lose information and prevent the full utilization of the data.

One dataset was prepared with 10,244 single-channel data samples and another dataset was prepared with 1332 six-channel data samples. The collected data was used to train and test two models, i.e. a single-channel model and a six-channel model (six sensors are used to monitor and receive data in this project). Here, each six-channel data sample consists of 6 waveforms. Thus, the total number of waveforms in the six-channel dataset is 7992 (1332 × 6). Among the 10,244 single-channel data samples, there are

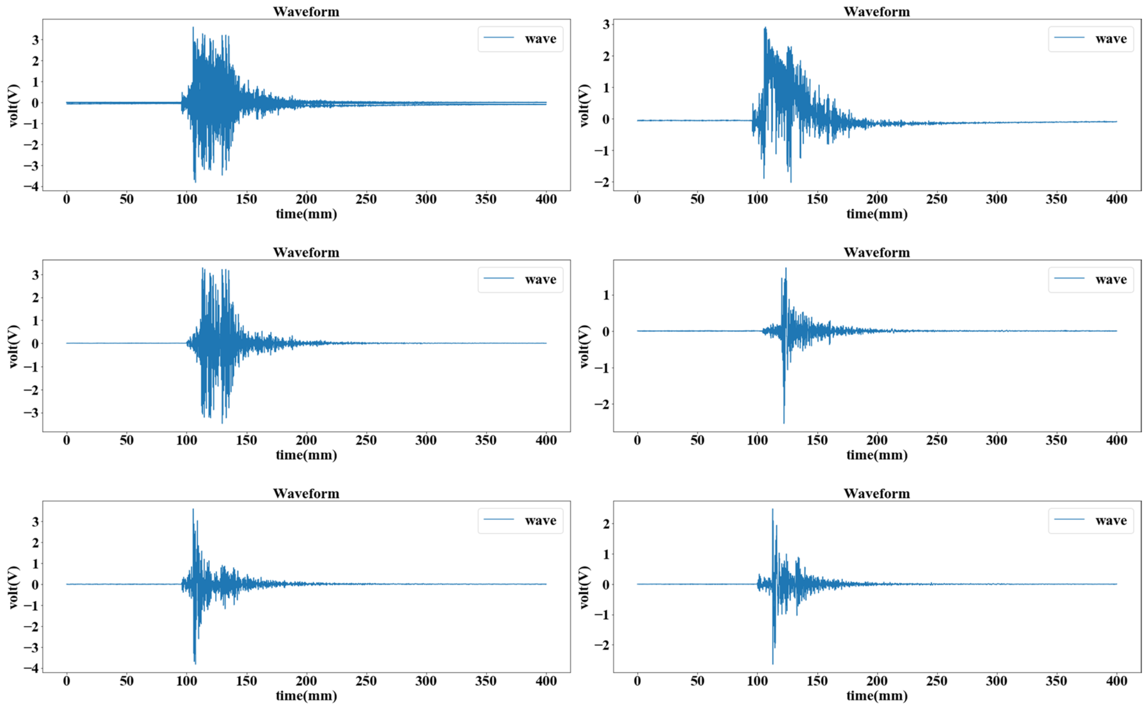


Fig. 3 An example of multichannel microseismic data

5001 microseismic waveforms triggered by rock fractures, and the other 5243 waveforms are all various types of noise. As previously noted, among the 5243 noisy waveforms, we deliberately selected many that show some similarities to microseismic waveforms to enable the network models to learn from sufficiently complex situations. Similarly, among the 1332 six-channel data samples, 647 waveforms were triggered by rock fractures. The remaining 685 waveforms contain only various types of noise. It can be predicted that for the single-channel model, it should be easy to implement transfer learning, which is introduced later in this paper. Therefore, the pretrained single-channel model can be easily applied for other waveform recognition tasks such as the recognition of quarry blasts or piling vibrations, as well as also directly for other rock engineering applications. By contrast, the six-channel model can only be used for projects with the same number of sensors, but it can achieve better accuracy than the single-channel model.

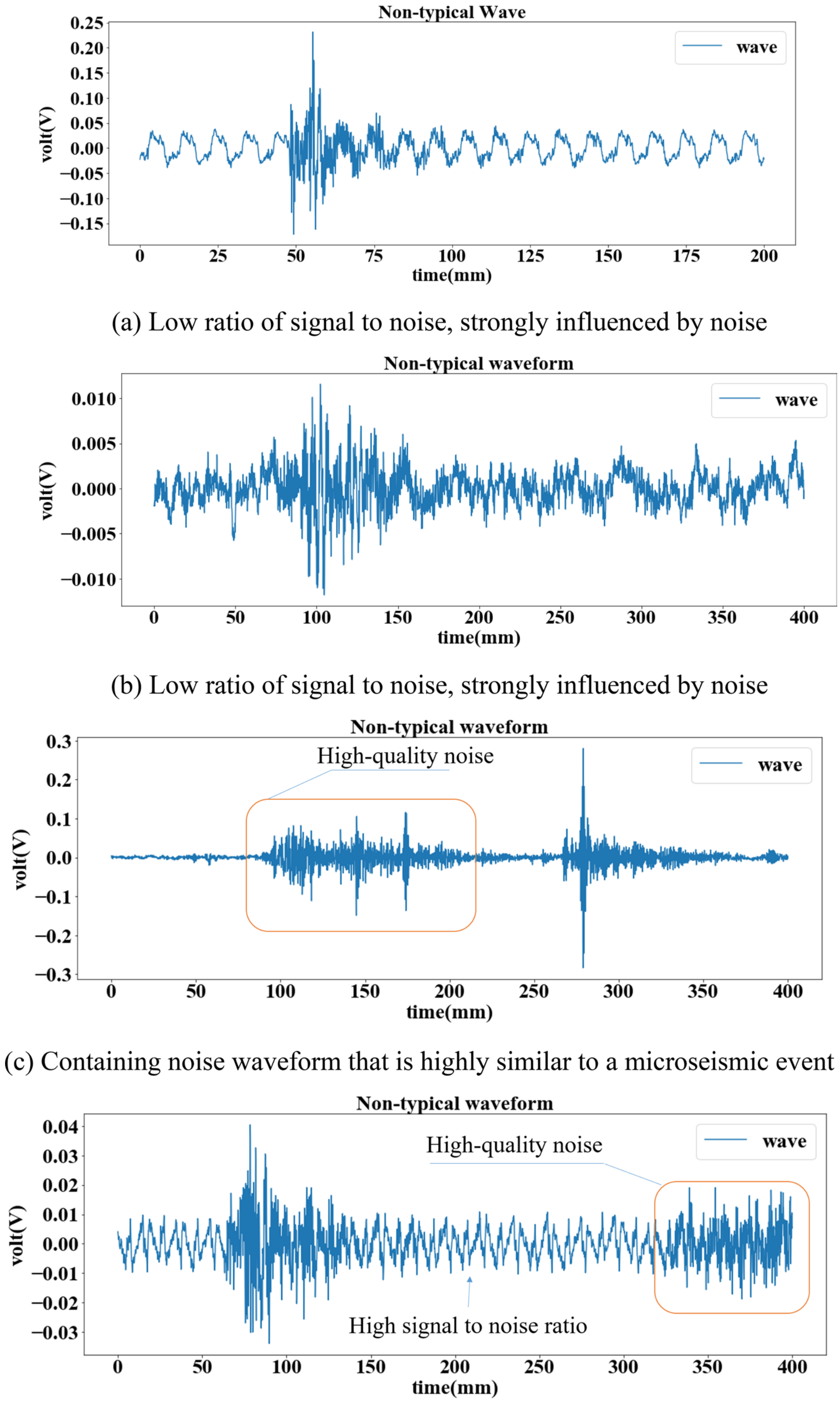
3 Methodology

3.1 Backbone Model Analysis of Neural Networks for the Microseismic Recognition Task

An ANN neural network can process various types of data, learn from data and update its own internal structure to improve its performance. Feedforward neural networks (FFNNs), recurrent neural networks (RNNs) and Convolutional neural network (CNNs) are the basic and typical architectures for deep neural networks. However, FFNNs and RNNs are not suitable for parsing superlong sequences of data because of the curse of dimensionality and gradient vanishing/explosion (Bengio et al. 1994; Elman 1990; Le et al. 2015).

CNNs are quite different from the other two types of networks. They are inspired by the organization of the animal visual cortex (Lecun et al. 1998). A CNN is not intended to parse data using a pixelwise approach. Instead, it tends to start with a “scanner”. For example, to take an image of 100 × 100 pixels as input, the input layer of an FFNN would need to contain 10,000 nodes. However, a CNN will create a scanner of 20 × 20 pixels, in which each pixel has its own weight, and move it in increments of one pixel from left to right (usually starting in the upper left corner). Each node concerns itself only with closely neighboring cells (how close depends on the implementation, but usually not more than a few). Moreover, the structure of a convolutional model relies on strong assumptions about local relationships in the data, which, when true, make it a good fit to the problem. Because the shaft invariance of images perfectly fits these assumptions, a CNN is ideal for images processing. For instance, when a CNN-based classification model examines an image to determine whether there is a dog in the picture, it does not matter where the dog is. Similarly, microseismic waveforms also

Fig. 4 Examples of nontypical microseismic waveforms



(a) Low ratio of signal to noise, strongly influenced by noise

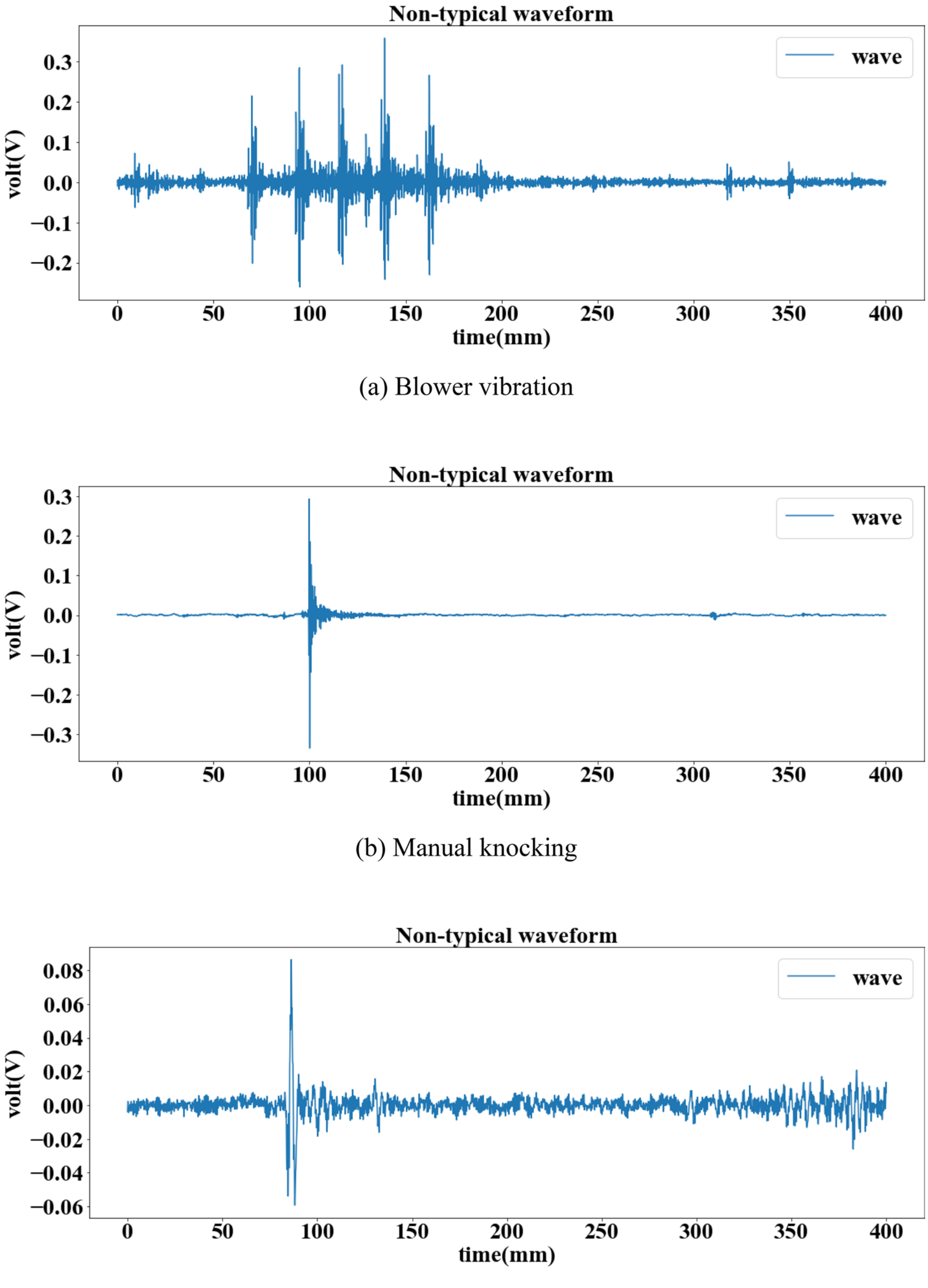
(b) Low ratio of signal to noise, strongly influenced by noise

(c) Containing noise waveform that is highly similar to a microseismic event

(d) Low signal-to-noise ratio and interfering noise

have this property, and the feature extraction process of CNN can be applied not only to two-dimensional images but also one-dimensional sequences of data. A one-dimensional CNN extracts features from a data sequences and maps the internal features of the sequence, which is an effective means of deriving features from fixed-length segments of an overall dataset in which the exact location of feature in a segment is not important.

Fig. 5 Examples of highly deceptive noise types



(a) Blower vibration

(b) Manual knocking

(c) Current interference

Overall, the architecture of the one-dimensional (1D) CNN is better for our recognition task than other basic architectures.

3.2 One-Dimensional Convolutional Neural Network (1D CNN)

A 1D CNN is basically identical to 2D CNN both mathematically and visually by setting the second dimension (either the vertical or horizontal one in visualization) to one. Accordingly, 1D filters can be applied in one dimension instead of using 2D filters spread over two dimensions. The basic properties of a CNN such as the local connectivity and shared weights are also independent of the number of dimensions. A deep 1D CNN consists of several component layer including the following four types:

(1) Convolutional

A one-dimensional convolutional layer consists of a line grid of neurons. Each neuron takes inputs from a line grid section of the previous input layer and the weights on this line section are the same for each neuron in the convolutional layer. In addition, there may be several grids in each convolutional layer, where each grid takes inputs from all grids in the previous layer, potentially using different filters. This process is expressed by Eq. (1):

$$y_i^l = f\left(\sum_{i=0}^{m-1} w_i y_{p+i}^{l-1}\right), \tag{1}$$

where w_i denotes the weight at position i in the filter, y^{l-1} is the output of the previous layer, and f denotes the activation function, which can be specified manually, and is usually the ReLU function (Glorot et al. 2011). Figure 6 shows a schematic sketch of convolution process where Seq represents a pending waveform and the yellow line grid represents the moving 1D filter (the convolution kernel), which aggregates information between adjacent nodes. w_1, w_2 and w_3 in Fig. 6 correspond to w_i in Eq. (1).

(2) Pooling layers

After each convolutional layer, there might be a pooling layer. The pooling layer takes small strip-shaped blocks from the preceding convolutional layer and subsamples each block to produce a single output. There are several ways to perform such pooling operation, such as taking the average, the maximum, or some learned linear combination of the neurons in the block.

(3) Batch normalization layers:

Batch normalization is a supervised learning technique that converts interlayer outputs of a neural network into a standard format, called normalizing. This effectively resets the distribution of the output of the previous layer to enable more efficient processing by subsequent layers (Ioffe and Szegedy 2015).

(4) Fully connected layers:

Real-world implementation of CNNs often glues a FFNN to the end of the network to further process the data, thus allowing for highly nonlinear abstractions. Other types of classifiers such as support vector machines (SVMs), logical regression models and other machine learning models, can also be applied for the postprocessing of the features extracted by CNNs (Girshick et al. 2014).

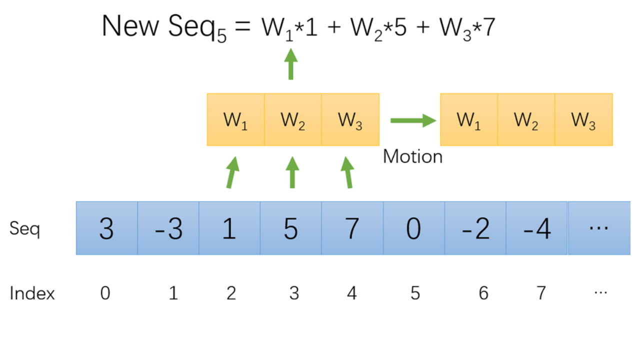


Fig. 6 The process of a one-dimensional convolution

When a deep network is being trained, gradient explosion/vanishing and model degradation often occur, increasing the difficulty of training. To overcome these obstacles, the concept of residual network (ResNets presented by He et al. (2016) is applied in our deep 1D CNN. This concept enables efficient performance improvement with very few additional parameters. Moreover, it is easy to combine with the attention module we have designed, which will be described Sect. 3.4.

3.3 Residual Learning

The conception of residual learning is essential to add skip connection. When a deep networks is able to start converging, a degradation problem might be exposed: as the network depth increases, the accuracy becomes saturated, which is not caused by overfitting (He and Sun 2015; Srivastava et al. 2015). This is unreasonable since as long as the neural network is able to fit an identity map, the performance of a deeper network should not be worse than that of a shallow one. In other words, deeper networks should be able to obtain potential advantages in identity mapping. Nevertheless, real observations tell us that neural networks are not good at fitting identity maps. To solve this problem, network architecture explicitly allows every few stacked layers to directly fit a residual mapping, instead of hoping that these layers will directly fit a desired base mapping. Equation (2) shows the relationship between these mappings:

$$x_{l+1} - x_l = F\left(x_l\right), \tag{2}$$

where x_{l+1} denotes the desired underlying mapping, and x_l is the output of previous layers. What the layers attempt to fit is $F\left(x_l\right)$, which is called the residual. In this way, the original mapping is recast as $x_l + F\left(x_l\right)$. The premise of this modification is that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping.

Nevertheless, sometimes, because the dimensionality of the data increases as the data flow deeper, a problem of unequal dimensions will arise when calculating $x_l + F\left(x_l\right)$. To solve this, we perform a linear transformation of x and project it into a new space with the same dimensions as $F\left(x_l\right)$.The parameters of the linear transformation can be obtained through backpropagation training. This is expressed in Eqs. (3) and (4).

$$x_{l+1} = h\left(x_l\right) + F\left(x_l\right), \tag{3}$$

$$h(x_l) = W_z x_l + b, \tag{4}$$

where $h(x_l)$ denotes the linear projection and W_z and b can be obtained through training.

Recursively, the relationship between the input layer and the Lth layer can be written as follows:

$$x_L = h(x_0) + \sum_{i=1}^{l-1} F(x_i), \tag{5}$$

where x_L is a correct mapping for layer L. $\sum_{i=1}^{l-1} F(x_i)$ represents all of the residuals from the first layer to the Lth layer and $h(x_0)$ denotes the linear transformation of x_0.

More generally, the relationship between layer A and layer B can be written as follows:

$$x_b = H(x_a) + \sum_{i=a}^{b-1} F(x_i). \tag{6}$$

Then, we consider a loss function u. The derivative of u with respect to x_b is expressed below:

$$\frac{\partial u}{\partial x_a} = \frac{\partial u}{\partial x_b}\frac{\partial x_b}{\partial x_a}. \tag{7}$$

By substituting in the expression for x_b from Eq. (6), this derivative can be expressed as shown in Eqs. (8) and (9).

$$\frac{\partial u}{\partial x_a} = \frac{\partial u}{\partial x_b}\left(1 + \frac{\partial}{\partial x_a}\sum_{i=a}^{b-1} F(x_i)\right), \tag{8}$$

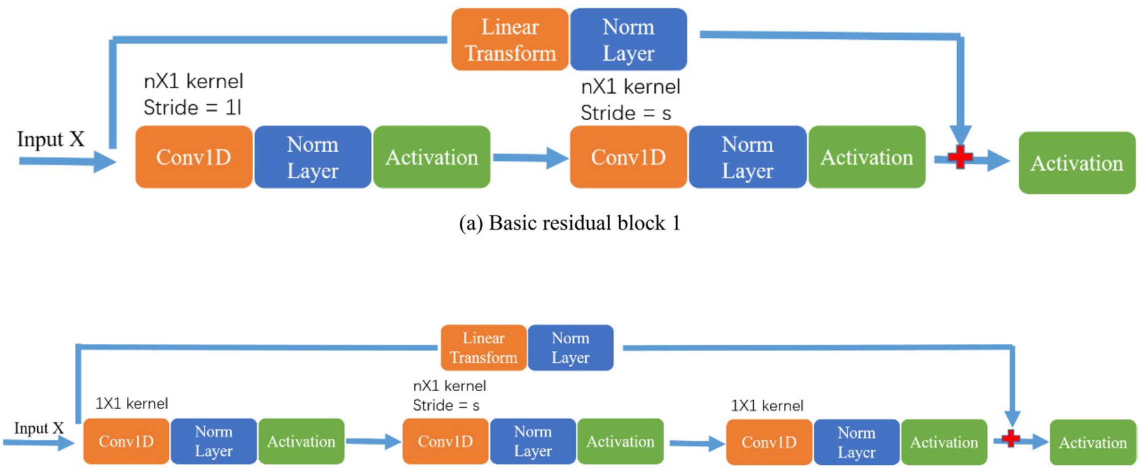
$$\frac{\partial u}{\partial x_a} = \frac{\partial u}{\partial x_b} + \frac{\partial u}{\partial x_b}\frac{\partial}{\partial x_a}\sum_{i=a}^{b-1} F(x_i). \tag{9}$$

As seen, from our analysis, gradient vanishing will not occur in ResNet because $\frac{\partial}{\partial x_a}\sum_{i=a}^{b-1} F(x_i)$ in Eq. (8) is never able to become zero throughout the entire training process. Furthermore, due to Eq. (9), the derivative of u with respect to x_b can be easily transferred to any previous layer. This is the reason why training a network with shortcut-like blocks is more effective and faster.

For our problem, two types of residual blocks are used. In the shallow layers of the model, the first module designed (Fig. 7a) is used when the dimensionality of the data is not very high. In other words, a shortcut connection is used once every two convolutional layers. However, to sufficiently extract waveform features, the number of channels needs to be a few hundred, especially for layers with smaller spatial inputs. In this case, the second residual block is used (Fig. 7b) to reduce the number of parameters. For a 1D convolution kernel, the number of parameters can be calculated using Eq. (10):

$$n = (l \times \mathrm{in}_c + 1) \times \mathrm{ou}_c, \tag{10}$$

where l denotes the length of the kernel, in_c is the dimensionality of the input data and ou_c is that of the output data. To alleviate the computational burden, the number of channels (dimensions) is reduced by applying a linear projection before the true convolutional layer. After the convolutional operation, the data will be transformed back into the original dimensional space. In detail, these transformations are implemented by means of a convolution kernel of length 1 (Fig. 7b). Furthermore, due to the presence of nonlinear activation functions such as the ReLU function after the linear transformation, the nonlinear fitting ability of the model is also enhanced.



(a) Basic residual block 1

(b) Bottleneck residual block 2

Fig. 7 The architectures of two types of residual blocks

In contrast to the traditional ResNet architecture (He et al. 2016) used in image processing, in which the convolution kernels have a constant size of 3 × 3, we designed our convolution kernels to be able to be freely specified.

Researchers have found that multiple stacked small convolutional kernels can achieve the same receptive field as a single larger convolution kernel, but with fewer parameters (Simonyan and Zisserman 2014). For example, it is easy to see that two 3 × 3 convolutional layers have an effective receptive field of 5 × 5, whereas three 3 × 3 layers have a 7 × 7 effective receptive field. A stack of two 3 × 3 convolutional layers is parametrized by $2 \times 3 \times 3 \times C^2 = 18C^2$, where C is the number of channels. However, a 5 × 5 layer has $5 \times 5 \times C^2 = 25C^2$ parameters, i.e., 39% more. Moreover, the addition of more nonlinear rectification layers can also make the decision function more discriminative.

However, for a 1D convolution, the number of parameters cannot be effectively decreased even by using a stack of layers with smaller kernels in place of a large with a larger receptive field. For instance, a 5 × 1 1D convolution kernel requires $5 \times C^2$ parameters. However, two 3 × 1 convolutional layers have $9 \times C^2$ parameters, an increase of over 80%. The effect is the opposite of what it is for a 2D CNN.

Overall, as long as the nonlinearity of the network is sufficient, larger convolution kernels should be chosen for the one-dimensional convolutional layers. Therefore, the residual blocks in our model should have kernels of different sizes, not merely 3 × 1 or 3 × 3, which should be manually determined through many experiments.

3.4 Depthwise Spatial and Channel Attention Module

In this subsection, we introduce our proposed DSCA module, which we have designed based on an “attention mechanism”. Originally, neural networks were developed by imitating the neural connections in human brains. Later, deep learning researchers took inspirations from neuroscience, cognitive science and other human behaviors to optimize network architecture. The concept of an attention mechanism is one of the ideas developed in this way. An important characteristic of human beings is that one is not inclined to process a whole scene in its entirety at once. Instead, people tend to pay selective attention to parts of what they can see to acquire the most necessary information and build up an internal representation of a scene to guide decision making processes. Google proposed a model that relies entirely on an attention mechanism to extract global dependencies for machine translation tasks, which achieved new state-of-the-art results in translation quality and reduced the required training and inference time (Vaswani et al. 2017). Squeeze-and-excitation modules can perform feature recalibration along the channel dimension based on an attention mechanism and a model based on this concept won the 2017 ImageNet Championship (Hu et al. 2018). Furthermore, other researchers have also studied the significance of attention (Jaderberg et al. 2015; Mnih et al. 2014; Xu et al. 2015). These findings all support our further exploration of the concept of attentions for our task.

(1) Spatial attention

When engineers conduct microseismic waveform analysis, they tend to focus on only a few parts of the waveform instead of dividing their attention equally among every part of the waveform data. For instance, when the microseismic wave shown in Fig. 8a is analyzed manually, more attention should be paid to the incoming microseismic event (indicated by the orange box), than to the deceptive noise (indicated by the blue circle) or to other unmarked parts that contain little information. If attention is assigned to unreasonable areas, it will be difficult to complete the waveform analysis task. Hence, the influence of noise can be effectively suppressed through an attention mechanism, which can replace traditional denoising algorithms to avoid the loss of information caused by noise removal. For the same reason, focusing the limited available computational resources on the proper parts of a scene can not only improve performance by ignoring irrelevant noise but also reduce the task complexity since the object of interest can be placed at the center of attention. Of course, this is true not just for the input waveform data but also for the inputs to all of the intermediate layers of a CNN.

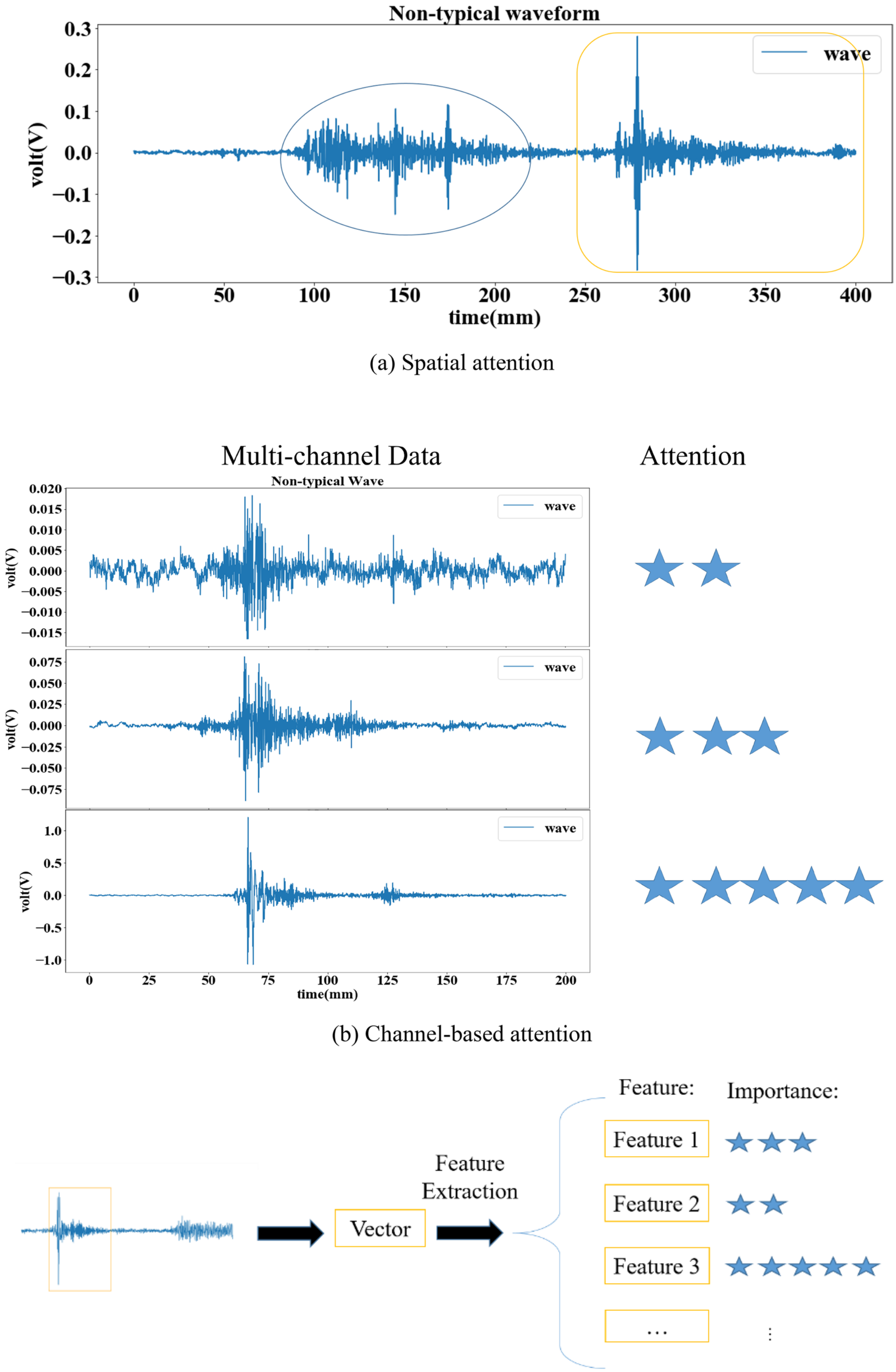
(2) Channel-based attention

When multichannel waveform data are analyzed simultaneously, experts will also unconsciously give priority to channels with lower noise (Fig. 8b); hence, the “attention” can be applied not only in terms of space but also in modeling the channel-based interrelationships of features.

As the data pass deeper into a CNN model, the model can produce many feature maps and the importance of different types of features should be different. Through channel attention, the model should be able to learn from training to selectively highlight more meaningful features while suppressing less useful ones by applying suitable weights to the feature vectors extracted in the intermediate layers (Fig. 8c).

(3) DSCA module

In this study, we have implemented an attention mechanism by applying weights in both the spatial and channel dimensions. Our DSCA module can learn how to weight the feature maps along the spatial and channel dimensions separately, and it can be easily used as a plug-and-play module



(a) Spatial attention

(b) Channel-based attention

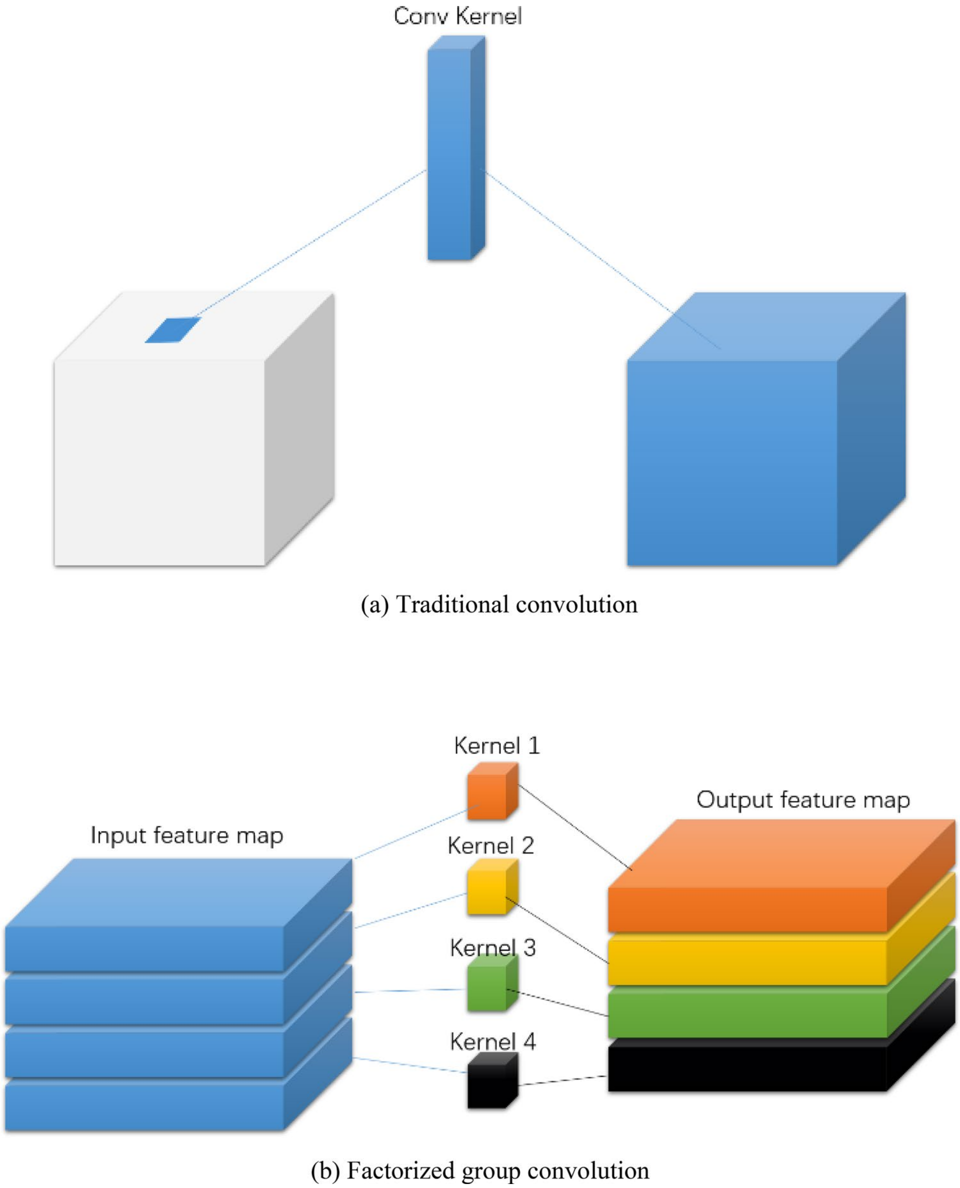
(c) Attention for intermediate information

Fig. 8 Attention diagram

in combination with pre-existing base CNN architectures such as residual blocks. Hence, a combination of DSCA and residual blocks was created to perform the microseismic wave classification task considered in this study.

Two different types of mini-networks were designed and embedded into the original neural model to recalibrate the importance of information along the spatial and channel dimensions. The concept of group convolution (Wang et al. 2017) was used to model spatial attention. Specifically, we adopted depthwise separable convolution or depthwise convolution, which is a special case of group convolution. Traditional convolution attempts to learn filters in a 3D space with two spatial dimensions (width and height) and a third dimension consisting of all channels (Fig. 9a). In other words, in traditional convolution, intra-channel spatial convolution and linear channel transformation are performed simultaneously. However, many studies (Chollet 2017; Wang et al. 2017; Zhang et al. 2017) on the computational redundancy of convolution have shown that a convolution operation can be divided into several groups along the channel dimension, where in each group, only a subset of the channels are convolved.. By stacking such factorized convolution groups sequentially to revert to the predefined output channels, similar or better accuracy can be achieved with significantly fewer computations. This is due to the reduction in the input channel dimensionality of the grouped convolution filters and the following hypothesis: the mapping of the cross-channel correlations and spatial correlations in the feature maps can be decoupled. An "extreme" version of group convolution is known as the depthwise separable convolution. In this operation, the number of groups is equal to

Fig. 9 Comparison between normal convolution and grouped convolution



the number of channels, i.e., each group contains only one channel (Fig. 9b).

We explicitly define a convolution filter for each channel to model the spatial attention weights. Different from other spatial attention mechanisms (Chen et al. 2017; Woo et al. 2018), aggregation operations are not used before computing the spatial attention weights because they can make features with high values more prominent and it is obvious that each feature map has a different spatial representation, which needs to be separately computed. Furthermore, depthwise spatial attention does not require high computational redundancy because most of the computation resources are consumed by the channel projection and our model performs only intra-channel spatial convolution. As long as the convolution is performed with an odd kernel size, which can be specified by the users, setting the stride equal to 1 and the zero padding equal to $(k-1)/2$ guarantees that the output will have the same spatial dimension as the input. The operation to produce weights for each feature map is expressed in Eq. (11). Then, weights are applied to the corresponding layers using Eq. (12).

$$w_s(x,c) = \sum_{l=1}^{k} W_k(u,v)F(x+l-1,c), \tag{11}$$

$$F_\mathrm{s} = W_\mathrm{si} * F_i, \tag{12}$$

where k is the length of the convolution kernel. $w_\mathrm{s}(x,c)$ denotes the spatial attention weight at $(x,1)$ in the cth channel. W_{si} denotes the concatenation of the $w_\mathrm{s}(x,c)$ of each feature map along the channel dimension; $*$ represents elementwise multiplication. W_k is the parameter matrix of the convolution kernel, obtained through training; F_i represents the output of the previous convolutional layer and F_s is the version of F_i. For our waveform classification task, k is usually specified as 7 or 9.

Since each channel can be regarded as a feature detector (Zeiler and Fergus 2014), channel attention can be used to determine which features are more meaningful given an input sequence. Similar to SE-Net (Hu et al. 2018), we compress the spatial information by adopting an average pooling operation to compute the channel attention. Although the pooling operation can produce an embedding of the global distribution of the channelwise feature responses, we argue that simple average pooling discards some important information about distinctive spatial features for each map. To solve this, the spatial attention weights are calculated before pooling to allow a finer the aggregation process. This is simply a matter of computing order without requiring any additional computation. It is easy to parallelize because the convolution is divided into many groups. Equation (13) expresses the spatial aggregation operation.

$$V_c = \frac{1}{L}\sum_{l=1}^{L} W_\mathrm{s}(x,y,c)F(l,c), \tag{13}$$

where V_c is the cth element of the aggregated vector. L is the feature map length (the feature maps of a 1D CNN have only one dimension). W_s denotes the abovementioned spatial attention weight matrix, and F denotes the cth original feature map. After the spatial information is compressed to obtain a vector with a global receptive field, a mini-network with 3 layers is embedded to model the importance weights of the input channels (Fig. 10). In this architecture, the number of neurons in the hidden layer can be set to any number greater than 6. The overall equation for recalibrating the output from the convolutional layers based on these two attention weights can be summarized as follows:

$$F_a = \mathrm{concat}\left(W_\mathrm{s} * F\right) * V_a, \tag{14}$$

where W_s is the spatial attention matrix, the "concat" operation consists of sequentially stacking the feature maps weighted by W_s along the channel dimension, and V_a is the channel attention vector generated by the mini-network shown in Fig. 10.

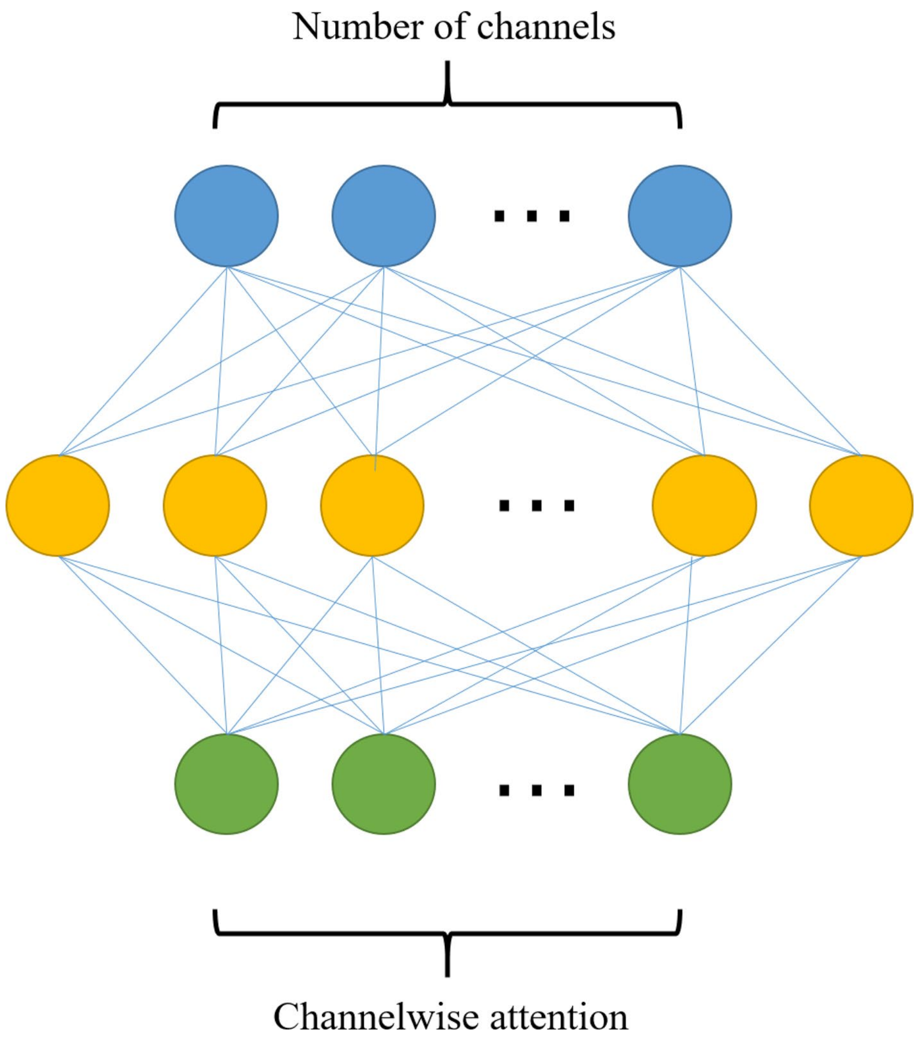
The whole attention process is shown in Fig. 11. After the data pass through the depthwise spatial attention block, the weights are generated and applied to the original information flow. In Fig. 11, darker-colored elements represent larger attention weights, whereas lighter ones correspond to smaller weights. After the channel attention matrix is calculated, it is multiplied by the spatial attention results. The DSCA module is flexible and can be directly incorporated into standard CNN architectures such as ResNet. To achieve the best results in the identification of microseismic events, we construct the aforementioned 1D residual architecture by adding a DSCA module before the residual sum operation in Fig. 7 following each convolution. Mathematically, the transformation F in Eq. (14) is taken as an entire residual block. Eventually, the complete ResSCA network is obtained by making this change for each such module (see Fig. 12) in the architecture.

4 Experimental Study

4.1 Results Comparison and Signal Channel Model Analysis

To confirm the effectiveness and performance of the proposed ResSCA architecture for recognizing microseismic signals, three networks were tested in comparative experiments: a relatively shallow classic neural network, a deep residual network and a deep residual network with the DSCA module integrated into it. Next, a six-channel

Fig. 10 The mini-network architecture for modeling the channel dependency



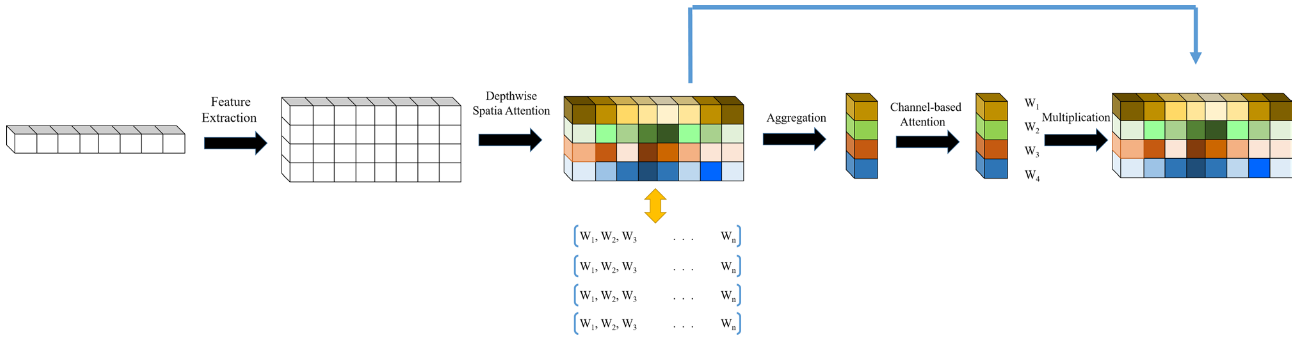


Fig. 11 The depthwise spatial and channel attention (DSCA) module

model was constructed specifically to achieve the optimal accuracy for the Hanjiang-to-Weihe River Diversion Project as a practical demonstration. All networks in this study were implemented using the Pytorch framework (Paszke et al. 2019). The dataset introduced in the second section of this paper was divided into two parts at a ratio of 2:8 between the training set and the test set. The training set was used for end-to-end training and the test set was used for model testing. Finally, a fivefold cross-validation experiment was conducted to further demonstrate improved better performance of ResSCA.

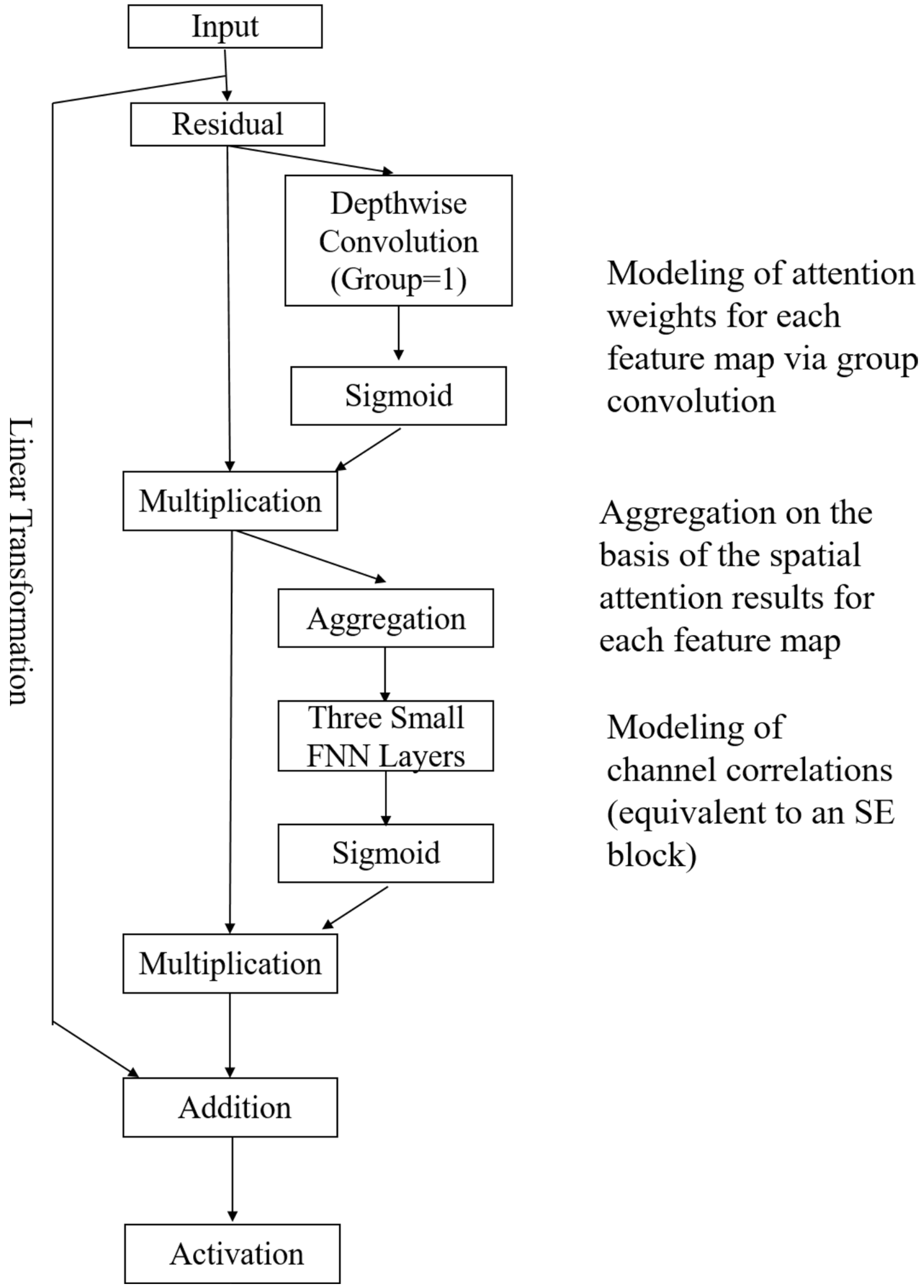

Input
Residual
Depthwise Convolution (Group=1)
Sigmoid
Multiplication
Aggregation
Three Small FNN Layers
Sigmoid
Multiplication
Addition
Activation
Linear Transformation
Modeling of attention weights for each feature map via group convolution
Aggregation on the basis of the spatial attention results for each feature map
Modeling of channel correlations (equivalent to an SE block)

Fig. 12 The Combination of the DSCA module and a residual skip connection

First, a 1D CNN with a common architecture was implemented and applied to our dataset. It was constructed by directly stacking different types of layers and a fully connected neural network was used as the classifier at the end of the model. The flow chart of the network is depicted in Fig. 13a. The mini-batch gradient descent algorithm was used to train the model and the batch size was set to 64. To accelerate convergence and avoid some controversial questions regarding the Adam, the newest AdamW optimizer (Loshchilov and Hutter 2018) was used for all networks tested in this paper. Moreover, a cross-entropy loss function, which is the most common type of loss function for classification tasks, was used as the loss function for all experiments reported in this paper. The accuracy curves on both the training data and the test data were plotted in Fig. 14a.

The final accuracy on the training set stabilizes at approximately 79.0% and that on test set was lower, which was approximately 75.0%. This indicates that the fitting ability of the model is insufficient. Therefore, it is desirable to continue to increase the depth of the model, because the higher the number of layers is, the more high-level features that are extracted. Meanwhile, to avoid gradient explosions/vanishing and model degradation and to shorten the training time,

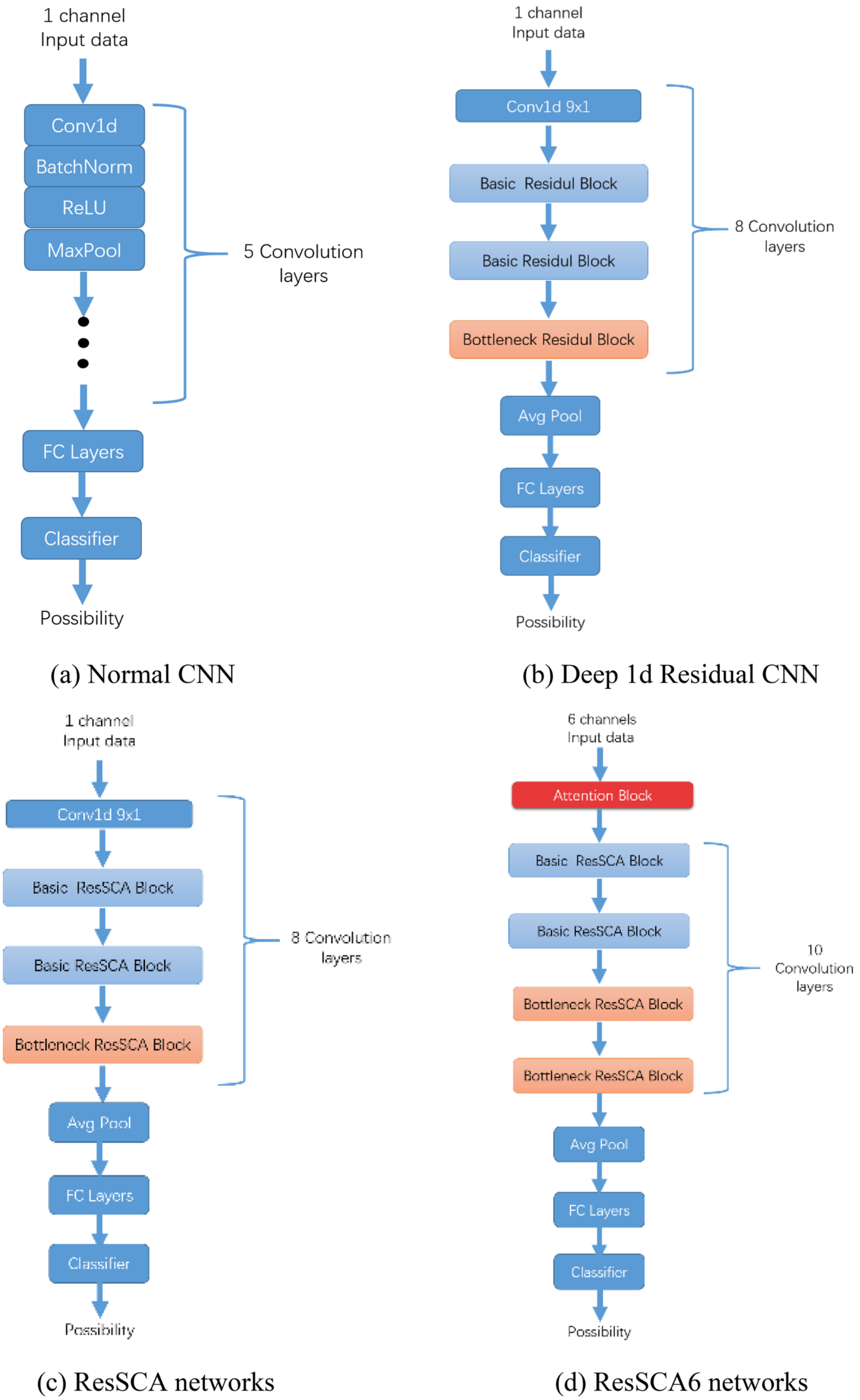
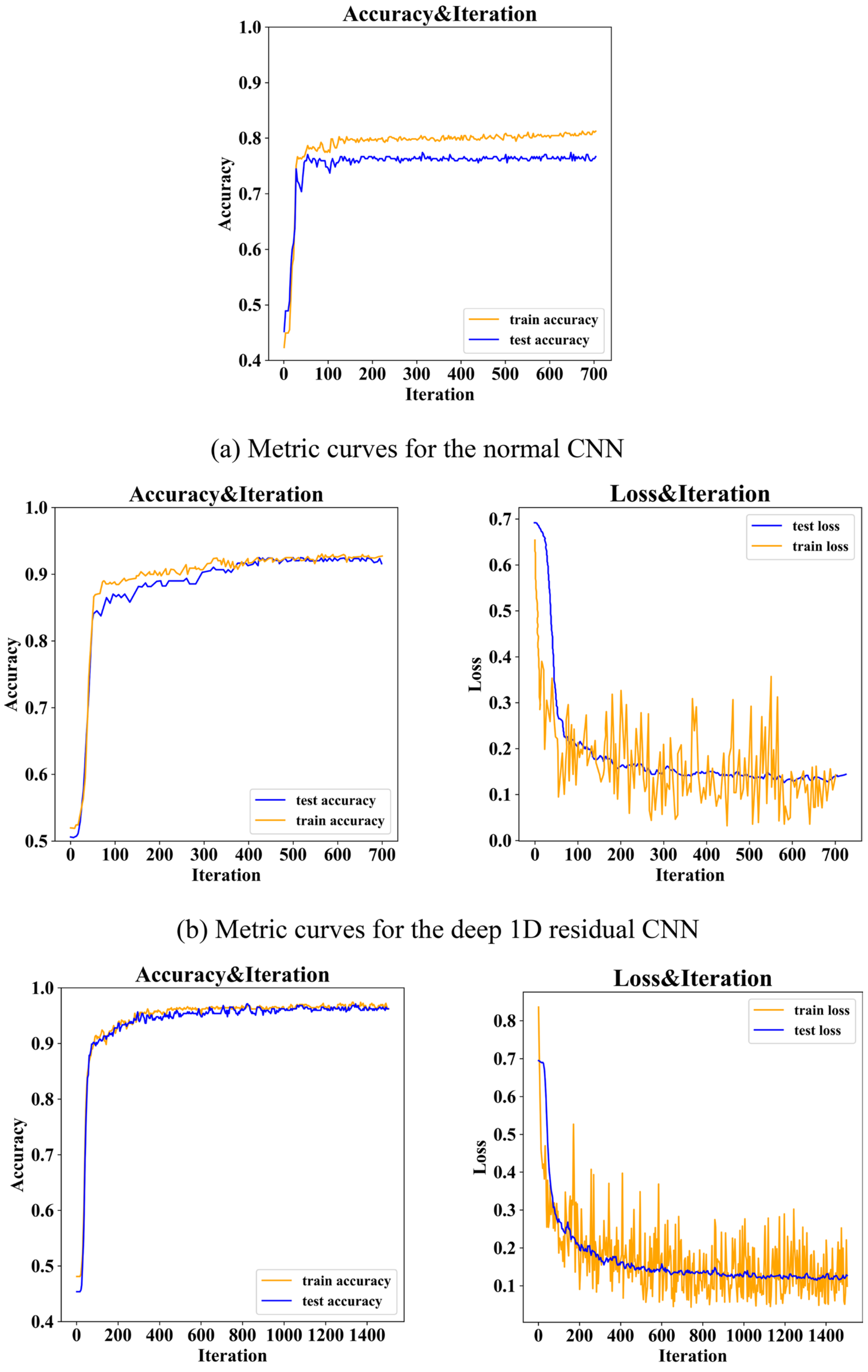


Fig. 13 Comparison of four types of architectures: normal CNN, deep 1D residual CNN, ResSCA, ResSCA6

we can deepen the network by stacking residual blocks. As described in the third section, the convolution kernels in the residual blocks should be set as large as possible when the nonlinearity of the model is sufficient. To ensure fair comparisons, the hyperparameter settings, loss function and optimizer should be the same as in other experiments. The flow chart of the 1D deep residual network is shown in Fig. 13b, and Table 1 gives detailed information about its internal hyperparameters. The kernel and stride in Table 1 refer to *n* and *s* in Fig. 7a, b, respectively. The results are presented in Fig. 14b. After 550 iterations, the training and test losses tended to become stable. Small gains in convergence

Fig. 14 Graphs of the metrics for the three experiments



(a) Metric curves for the normal CNN

(b) Metric curves for the deep 1D residual CNN

(c) Metric curves for the ResSCA

speed and large gains in accuracy are evident. The accuracy on both the training and test data was increased to approximately 92.0%, by increasing the number of layers and embedding residual connections. This performance is much better than that of the normal CNN models commonly used for microseismic waveform recognition. Moreover, the

Table 1 Configuration of the deep 1D residual CNN

Layer types	In/out channels	Kernel	Stride	Parameters
Convolution Block	1/32	9	3	288
Basic Block	32/64	5	2	32,768
Basic Block	64/160	3	2	117,760
Bottleneck Block	160/304	5	2	112,784
FFNN1	304/96	/	/	29,184
FFNN2	96/2	/	/	192

total number of parameters is 292,976, fewer than in a traditional 2D CNN for image processing.

To obtain state-of-the-art results, we can continue to deepen the neural network with skip connections, and the performance of the model ought to improve further. However, the number of channels in the new layers needs to be a few hundred. Consequently, adding only one layer will increase the number of parameters by hundreds of thousands. Our aim is to build a lightweight model and a key requirement is high usability. Therefore, making the internal structure more finely tuned rather than adding more layers is another way to achieve better performance.

Thus, in our third experiment, the proposed DSCA module was embedded in the deep 1D residual network. This module is straightforward and can be directly used in existing CNN architectures by inserting it behind each residual block as shown in Fig. 12; the resulting structure is called a counterpart block. The details of this architecture are presented in Fig. 13c and Table 2. The difference is that the residual block is replaced with our ResSCA block and the other network components remain the same. The column labeled “attention kernel” in Table 2 refers to k in Eq. (11). The total number of parameters is 312,480. Figure 14c plots the results for the ResSCA network. After full convergence before overfitting, 97.5% accuracy was achieved on both the training and test sets, approximately 6% higher more than the accuracy of the residual network. Meanwhile, there is only a slight increase in the number of parameters (19,504 additional parameters, representing 6% of the total), and the additional computational overhead can be neglected in most cases.

Thus, it is shown that, the lightweight DSCA module can successfully refine substantial features in both the channel and spatial dimensions and help the information flow smoothly without introducing redundant computations and parameters. Hence, the ResSCA architecture can achieve state-of-the-art results on our complex microseismic signal dataset. In addition, the pretrained model can be easily reused because it is trained simply to accept single-channel waveform input and is not limited by the number of channels (in other words, the number of sensors). The only thing that needs to be done is to freeze the parameters of the convolutional layers to serve as a feature extractor and train a new classifier for these features to replace the last two fully connected layers; this new classifier may be an SVM, an XGBoost model or a different FFNN.

4.2 K-fold Cross Validation

We used k-fold cross validation to compare the performance of ResSCA and ResNet (a common residual network without attention blocks) to prove the ability of theDSCA module and eliminate the possible impacts of specific datasets. The normal CNN was not included in the k-fold (fivefold) cross-validation experiment since it showed the worst performance in the previous experiments. K-fold cross-validation is a common type of cross validation that is widely used in machine learning to enable better utilization of data. The steps of k-fold cross-validation are as follows: (1) Partition the original training data set into k equal subsets. Each subset is called a fold. (2) Reserve one fold as the validation set and use all the remaining $k-1$ folds as training set. (3) Train models on each pair of train-test set and get k results. (4) Analyze the results.

In this study, we used fivefold cross validation to ensure that each test set would contain more than 2000 data samples and the results are given in Fig. 15. Figure 15 shows the test set accuracy after 100 and 500 training iterations for each model, respectively. As shown in the legend, the filled circles on the blue and green lines represent the test accuracy rates in iterations 100 and 500 when ResSCA was trained on each fold. Similarly, the stars on the yellow and red lines represent the same for ResNet. It can be observed from this figure that

Table 2 Configuration of the ResSCA network

Layer types	In/out channels	Kernel	Stride	Attention kernel	Parameters
Convolution Block	1/32	9	3	9	704
Basic Block	32/64	5	2	9	33,856
Basic Block	64/160	3	2	7	122,080
Bottleneck Block	160/304	5	2	7	126,464
FFNN1	304/96	/	/	/	29,184
FFNN2	96/2	/	/	/	192

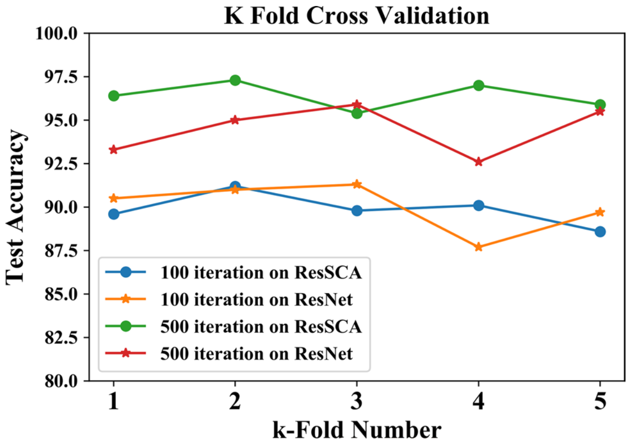


Fig. 15 Results of fivefold cross validation

there was not much difference in performance between the two models in the initial stage of training. However, as the number of training rounds increased, the performance of ResSCA gradually exceeded that of ResNet. After 500 training iterations, the accuracy rates of ResSCA were almost all higher than those of ResNet. These findings indicate that networks with DSCA better capture nonlinear relationships among data as long as they are sufficiently trained.

4.3 Transfer Learning and Reusability

The pretrained ResSCA network can also be applied for other rock engineering applications. If we wish to apply the ResSCA model trained in this study to other rock engineering projects, such as a rock slope project, we do not need to train a new model from scratch. Instead, we can fine-tune the pretrained ResSCA model with only a few hundred labeled data samples from the new projects. The fine-tuning step is necessary because the characteristics of microseismic signals are different in different projects due to differences in the lithology. This technique of transferring models to new applications is called transfer learning, which has proven to be very practical in deep learning (Lei 2019).

Transfer learning makes use of the knowledge gained while solving one problem by applying it to a different but related problem (Zhang et al. 2019; Lei 2019). Usually, many data are needed to train a neural network from scratch, but access to those data is not always available—this is where transfer learning comes in handy. With transfer learning a suitable machine learning model can be built with comparatively little training data because the model is already pretrained. For example, knowledge gained when learning to recognize microseisms in a tunneling project can be used to some extent, to recognize events in other rock engineering projects, such as slope engineering or mining projects. When we trained the network on our dataset, we trained all parameters of the neural network to obtain the final model. This may take hours of computing time and require a lot of available labeled data. However, if we wish to reuse the ResSCA, we can directly load our pretrained model and parameters and feed the model a new dataset (collected from another rock engineering project) to fine-tune the pretrained ResSCA. Any new dataset consisting of microseismic monitoring data will be highly similar to the original dataset used for pretraining. Since the new dataset is similar, the same weights can be used for extracting the features from the new dataset. Here, two fine-tune methods are recommended:

(1) If the new dataset is very small, it is better to train only the classifier of the network to avoid overfitting, keeping all other layers fixed. For this purpose, the final fully connected layers (the classifier in ResSCA is a fully connected network) of the pretrained network should be removed. Then, new layers should be added to replace the old ones. Only the new layers should be retrained.
(2) If the new dataset is very large, the whole network can be retrained with the weights initially set to the values from the pretrained model.

4.4 Practical Application and Multichannel Model Analysis

Following the previously introduced concept of ResSCA, we trained a six-channel model specifically for application to the Hanjiang-to-Weihe River Diversion Project. Six sensors are used to monitor a certain section of this project. When a microseism occurs, these sensors will be triggered by the same signal source but will generate different waveforms due to the different local conditions of each sensor such as the location and burial depth. To allow the model to accept the six-channel input and generate better results than the common ResSCA in this specific project, two modifications were made. First, we stacked every six single-channel waveforms reflecting the same event along the channel dimension, analogous to an RGB image. In addition, the shape of the input data matrix was changed from (n, 1, 4001) to (n/6, 6, 4001) where n is the number of samples and the second dimension denotes the number of channels, from 1 to 6. In this study, n refers to the number of waveforms, which is 7992, and n/6 is the number of six-channel data samples, which is one-sixth of the number of waveforms, i.e., 1332. The second entries in (n, 1, 4001) and (n/6, 6, 4001) represent the number of channels. Finally, the value of 4001 indicates that all waveforms consist of 4001 sampling points. Because the input now has the form of six-channel signals, the attention block shown in Fig. 11 is inserted before the first convolutional layer to model the interdependencies among the six channels

Table 3 Configuration of the multichannel ResSCA network (ResSCA6)

Layer types	In/out channels	Kernel	Stride	Attention kernel	Parameters
Attention Module	6/6	/	/	9	126
Basic Block	1/32	9	3	9	41,792
Basic Block	32/64	5	2	7	84,864
Bottleneck Block	64/160	3	2	7	79,616
Bottleneck Block	160/304	3	2	7	198,512
FFNN1	304/96	/	/	/	29,184
FFNN2	96/2	/	/	/	192

Table 4 Comparison of results and complexity among different models

Architecture	Accuracy	Number of parameters
Normal CNN	75.0	/
Residual network	90.0	292,276
ResSCA	97.5	312,480
ResSCA6	99.3	434,286

and enable a more comprehensive analysis. Second, because the input data are more complex than the single-channel signals used in ResSCA, the neural network has one more layer than the ResSCA network. The architecture of the ResSCA6 networks is depicted in Fig. 13d and the hyperparameters are given in Table 3. Finally, the trained ResSCA6 model achieved 99.3% accuracy. This result proves that considering the different channels simultaneously, as an expert dose can improve the performance. A comparison of the four experiments is reported in Table 4.

Overall, a multichannel model adapted for a specific project can achieve the best results but is slightly more complex. By contrast, the single-channel model is more advantageous for transfer learning and is not limited by the number of sensors.

5 Conclusion

This study has proposed a new lightweight network architecture for efficiently recognizing microseismic events, which is based on concepts of the residual skip connections and an attention mechanism in deep learning. The performance of the model was tested using a monitoring signal dataset we created, which consists of complex nontypical waveforms and highly deceptive noise samples selected from the Hanjiang-to-Weihe River Diversion Project in China.

To improve the representational ability of neural networks and allow such models to perform better for real rock engineering applications, the novel DSCA module based on an attention mechanism has been designed, and two types of residual blocks have been modified to their 1D version. The lightweight module can easily be inserted into pre-existing popular CNN architectures to successfully learn which parts of the information should be emphasized or suppressed along both the channel and spatial dimensions. In addition, theoretically, this module can also be used to model the interdependence among multichannel signals and suppress noise by applying the attention weights obtained through training. We integrated residual connections and the DSCA module into a normal deep CNN, obtaining a new architecture named ResSCA, which achieved state-of-the-art results on our dataset. To achieve the best results for the specific project considered in this study, a six-channel model named ResSCA6 was constructed based on the proposed network concept. The basic single-channel model is obviously more amenable to transfer learning to adapt it to different projects and tasks, but a suitably constructed multichannel model can achieve higher performance for a specific project due to the consideration of the interdependencies among different sensors.

In conclusion, the proposed lightweight DSCA module can improve network performance without requiring many additional parameters or incurring much additional computational cost by refining the intermediate information in a CNN. Furthermore, the ResSCA network performs well at identifying microseismic signals, and this concept has great potential for processing other waveform data obtained from construction activities. With further improvements to the neural network algorithm, it can be effectively applied in other rock engineering applications for intelligent monitoring.

Acknowledgements This project was financially supported by the National Key Research and Development Program (2017YFC1503102) and the National Natural Science Foundation of China (41941018, 51874065 and U1903112).

Data Availability All the monitoring data that support this study are available from the corresponding author upon reasonable request.

Compliance with Ethical Standards

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

Akram J, Eaton DW (2016) A review and appraisal of arrival-time picking methods for downhole microseismic data. Geophysics 81:KS71–KS91

Alvarez I, Garcia L, Mota S, Cortes G, Benitez C, De la Torre A (2013) An automatic P-Phase picking algorithm based on adaptive multiband processing. IEEE Geosci Remote Sens Lett 10:1488–1492. https://doi.org/10.1109/lgrs.2013.2260720

Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult IEEE transactions on neural. Networks 5:157–166

Chen L, Zhang H, Xiao J, Nie L, Shao J, Liu W, Chua T-S (2017) SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning. In: 30th IEEE conference on computer vision and pattern recognition, CVPR 2017, July 21, 2017–July 26, 2017, Honolulu, HI, United states, 2017. Proceedings—30th IEEE conference on computer vision and pattern recognition, CVPR 2017. Institute of Electrical and Electronics Engineers Inc., pp 6298–6306. https://doi.org/10.1109/CVPR.2017.667

Dai H, Macbeth C (2007) Automatic picking of seismic arrivals in local earthquake data using an artificial neural network. Geophys J Int 120:758–774

Dong L, Yang Y, Qian B, Tan Y, Sun H, Xu N (2019) Deformation analysis of large-scale rock slopes considering the effect of microseismic events. Appl Sci 9:3409. https://doi.org/10.3390/app9163409

Elman JL (1990) Finding structure in time. Cogn Sci 14:179–211

Ge M, Mrugala M, Iannacchione AT (2009) Microseismic monitoring at a limestone mine. Geotech Geol Eng 27:325–339

Ghosh GK, Sivakumar C (2018) Application of underground microseismic monitoring for ground failure and secure longwall coal mining operation: a case study in an Indian mine. J Appl Geophys 150:21–39

Girshick R, Donahue J, Darrell T, Malik JR (2014) feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2014, Columbus, OH, United states, June 23, 2014–June 28 2014. Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, pp 580–587. https://doi.og/10.1109/CVPR.2014.81

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statisticss, AISTATS 2010, Sardinia, Italy, May 13, 2010–May 15, 2010 2010. Journal of Machine Learning Research. Microtome Publishing, pp 249–256

Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, AISTATS 2011, Fort Lauderdale, FL, United states, April 11, 2011–April 13 2011. Journal of Machine Learning Research. Microtome Publishing, pp 315–323

Guo X, Li Z, Qin N, Jin W (2011) Adaptive picking of microseismic event arrival using a power spectrum envelope. Comput Geosci 37:158–164. https://doi.org/10.1016/j.cageo.2010.05.022

He K, Sun J (2015) Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, United states, June 7, 2015 - June 12, 2015. Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, pp 5353–5360. https://doi.org/10.1109/CVPR.2015.7299173

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, United states, June 26, 2016–July 1, 2016. Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, pp 770–778. https://doi.org/10.1109/CVPR.2016.90

Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: 31st meeting of the IEEE/CVF conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, United states, June 18, 2018 - June 22 2018. Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, pp 7132–7141

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift ArXiv

Jaderberg M, Simonyan K, Zisserman A (2015) Spatial transformer networks. Adv Neural Infor Process Syst 2015:2017–2025

Le QV, Jaitly N, Hinton G (2015) A simple way to initialize recurrent networks of rectified linear units ArXiv

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE Conf Comput Vis Pattern Recogn 86:2278–2324

Lee M, Byun J, Kim D, Choi J, Kim M (2017) Improved modified energy ratio method using a multi-window approach for accurate arrival picking. J Appl Geophys 139:117–130. https://doi.org/10.1016/j.jappgeo.2017.02.019

Lei Z (2019) Transfer adaptation learning a decade survey ArXiv

Li Y, Ni Z, Tian Y (2018) Arrival-time picking method based on approximate negentropy for microseismic data. J Appl Geophys 152:100–109. https://doi.org/10.1016/j.jappgeo.2018.03.012

Li P, Chen X, Shen S (2019) Stereo R-CNN based 3D Object detection for autonomous driving. In: 32nd IEEE/CVF conference on computer vision and pattern recognition, CVPR 2019, June 16, 2019–June 20, 2019, Long Beach, CA, United states, 2019. Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, pp 7636–7644. https://doi.org/10.1109/CVPR.2019.00783

Liang Z, Peng S, Zheng J (2014) Self-adaptive denoising for microseismic signal based on EMD and mutual information entropy. Comput Eng Appl 50:7–11

Lin B, Wei X, Junjie Z (2019) Automatic recognition and classification of multi-channel microseismic waveform based on DCNN and SVM. Comput Geosci 123:111–120. https://doi.org/10.1016/j.cageo.2018.10.008

Liu F, Ca T, Ma T, Tang L (2019) Characterizing rockbursts along a structural plane in a tunnel of the Hanjiang-to-Weihe river diversion project by microseismic monitoring rock. Mechan Rock Eng 52:1835–1856. https://doi.org/10.1007/s00603-018-1649-0

Loshchilov I, Hutter F (2018) Fixing weight decay regularization in adam ICLR 2018 conference

Ma TH, Tang CA, Tang LX, Zhang WD, Wang L (2015) Rockburst characteristics and microseismic monitoring of deep-buried tunnels for Jinping II Hydropower Station. Tunn Undergr Space Technol 49:345–368. https://doi.org/10.1016/j.tust.2015.04.016

Milev AM, Spottiswoode SM (2002) Effect of the rock properties on mining-induced seismicity around the ventersdorp contact reef Witwatersrand Basin, South Africa. Mech Induc Seism 159:165–177

Mnih V, Heess N, Graves A, Kavukcuoglu K (2014) Recurrent models of visual attention. In: 28th annual conference on neural information processing systems 2014, NIPS 2014, December 8, 2014–December 13, 2014, Montreal, QC, Canada, 2014. Advances in neural information processing systems. Neural information processing systems foundation, pp 2204–2212

Paszke A et al (2019) PyTorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 32:8024–8035

Paul BQ, Pierre G, Yoann C, Munkhuu U (2015) Detection and classification of seismic events with progressive multi-channel correlation and hidden Markov models. Comput Geosci 83:110–119. https://doi.org/10.1016/j.cageo.2015.07.002

Rodriguez IV, Bonar D, Sacchi M (2012) Microseismic data denoising using a 3C group sparsity constrained time-frequency transform. Geophysics 77:21–29

Shang X, Li X, Morales-Esteban A, Chen G (2017) Improving microseismic event and quarry blast classification using artificial neural networks based on principal component analysis. Soil Dyn Earthq Eng 99:142–149. https://doi.org/10.1016/j.soildyn.2017.05.008

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition ArXiv

Song F, Kuleli HS, Toksöz MN, Ay E, Zhang H (2010) An improved method for hydrofracture-induced microseismic event detection and phase picking. Geophysics 75:A47–A52

Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks arXiv preprint arXiv:00387

Urbancic T, Trifu C-I (2000) Recent advances in seismic monitoring technology at Canadian mines. J Appl Geophys 45:225–237

Vaswani A et al (2017) Attention is all you need. In: 31st annual conference on neural information processing systems, NIPS 2017, December 4, 2017–December 9, 2017, Long Beach, CA, United states, 2017. Advances in neural information processing systems. Neural information processing systems foundation, pp 5998–6008

Wang J, Teng TL (1995) Artificial neural network-based seismic detector. Bull Seismol Soc Am 85:308–319

Wang M, Liu B, Foroosh H (2017) Factorized convolutional neural networks. In: 16th IEEE international conference on computer vision workshops, ICCVW 2017, October 22, 2017–October 29, 2017, Venice, Italy, 2017. Proceedings - 2017 IEEE international conference on computer vision workshops, ICCVW 2017. Institute of Electrical and Electronics Engineers Inc., pp 545–553. https://doi.org/10.1109/ICCVW.2017.71

Wilkins AH, Strange A, Duan Y, Luo X (2020) Identifying microseismic events in a mining scenario using a convolutional neural network. Comput Geosci 137:104418. https://doi.org/10.1016/j.cageo.2020.104418

Woo S, Park J, Lee J-Y, Kweon IS (2018) CBAM: convolutional block attention module. In: 15th european conference on computer vision, ECCV 2018, September 8, 2018 - September 14, 2018, Munich, Germany, 2018. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Springer, New York, pp 3–19. https://doi.org/10.1007/978-3-030-01234-2_1

Chollet F Xception (2017) Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2017, Honolulu, HI, United states, July 21, 2017–July 26 2017. Institute of Electrical and Electronics Engineers Inc., pp 1800–1807. https://doi.org/10.1109/CVPR.2017.195

Xu K et al (2015) Show, attend and tell: Neural image caption generation with visual attention. In: 32nd international conference on machine learning, ICML 2015, July 6, 2015–July 11, 2015, Lile, France, 2015. 32nd international conference on machine learning, ICML 2015. International Machine Learning Society (IMLS), pp 2048–2057

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: 13th European conference on computer vision, ECCV 2014, September 6, 2014–September 12, 2014, Zurich, Switzerland, 2014. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Springer, New York, pp 818–833. https://doi.org/10.1007/978-3-319-10590-1_53

Zeng X, Ouyang W, Yang B, Yan J, Wang X (2016) Gated Bi-directional CNN for Object Detection. In: Computer vision—14th European conference, ECCV 2016, Proceedings, 2016. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Springer, New York, pp 354–369. https://doi.org/10.1007/978-3-319-46478-7_22

Zhang T, Qi G-J, Xiao B, Wang J (2017) Interleaved group convolutions. In: 16th IEEE international conference on computer vision, ICCV 2017, Venice, Italy, 2017. Proceedings of the IEEE international conference on computer vision. Institute of Electrical and Electronics Engineers Inc., pp 4373–4382. https://doi.org/10.1109/ICCV.2017.469

Zhang J, Li W, Ogunbona P (2019) <Transfer Learning For Cross-Dataset Recognition A Survey.pdf> ArXiv

Zhao Y, Takano K (1999) An artificial neural network approach for broadband seismic phase picking. Bull Seismol Soc Am 89:670–680

Zhao GY, Ma J, Dong LJ, Li XB, Hui CG, Zhang CX (2015) Classification of mine blasts and microseismic events using starting-up features in seismograms. Trans Nonferrous Metals Soc China 25:3410–3420

Zhuang D, Ma K, Tang C, Cui X, Yang G (2019) Study on crack formation and propagation in the galleries of the Dagangshan high arch dam in Southwest China based on microseismic monitoring and numerical simulation. Int J Rock Mech Min Sci 115:157–172. https://doi.org/10.1016/j.ijrmms.2018.11.016

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.