ISSN 1392-1207. MECHANIKA. Vol. 32, No. 5, 2026: 402–411

Fault Diagnosis Based on Wavelet Transform and an Improved Octave Convolutional Neural Network

TianTong ZHANG*, JianPing YU, PeiYi YANG*****

*College of Locomotive and Rolling Stock, Zhengzhou Railway Vocational & Technical College, Zhengzhou 450046, China, E-mail: 405919311@qq.com

**East China Jiaotong University, Nanchang, 330013, China, E-mail: 2421113163@qq.com

***College of Locomotive and Rolling Stock, Zhengzhou Railway Vocational & Technical College, Zhengzhou 450046, China, E-mail: 771576680@qq.com

https://doi.org/10.5755/j02.mech.43853

1. Introduction

Bearings, as crucial components in rotating machinery, directly affect the safety and reliability of equipment [1]. Under long-term operational conditions, bearings are vulnerable to a variety of fault types, including spalling, cracks, wear, and rolling element damage, which are induced by factors such as impact loads, lubricant degradation, fatigue accumulation, and material defects [2]. In the absence of effective monitoring and diagnostic mechanisms, these faults can lead to significant performance degradation, potentially triggering severe system failures and substantial economic losses. Therefore, conducting research on high-precision and robust bearing fault diagnosis is of paramount importance [3].

Early fault diagnosis predominantly relied on traditional signal processing techniques, such as time-domain statistical analysis [4], Fast Fourier Transform (FFT) [5], envelope analysis [6], Empirical Mode Decomposition (EMD) [7], and Local Mean Decomposition (LMD) [8], among others. These methods are capable of effectively extracting fault features under specific operating conditions. However, they commonly exhibit limitations, such as restricted feature extraction capabilities, sensitivity to noise, and reliance on human expertise, which restrict their applicability in complex operating conditions and noisy environments [9].

With the advancement of pattern recognition technologies, machine learning-based bearing fault diagnosis methods have gained widespread attention, including models such as Support Vector Machines (SVM) [10], Random Forests (RF) [11], and K-Nearest Neighbors (KNN) [12]. These methods utilize manually extracted features for classification; however, their performance heavily depends on the quality of feature engineering. Since manually extracted features often fail to fully capture the non-stationarity and multi-scale characteristics of vibration signals, the generalization capability of these models is constrained [13].

Among various time-frequency analysis methods, wavelet transform is widely used due to its excellent time-frequency localization properties, making it effective for capturing transient impacts and modulation characteristics in bearing fault signals. The resulting time-frequency representations clearly display energy distributions across different scales, offering feature representations with stronger physical significance for deep learning model inputs [14].

On the other hand, while Convolutional Neural Networks (CNNs) possess powerful feature extraction capabilities, traditional convolutional structures often suffer from issues such as high feature redundancy and difficulties in simultaneously handling high-frequency and low-frequency information. Octave Convolution (Oct-CNN) addresses this by dividing feature maps into high-frequency and low-frequency components, allowing cross-frequency interactions [15]. This reduces computational complexity and enhances the model's ability to perceive multi-scale features. This structure is particularly advantageous for bearing signals that simultaneously contain impact information (high-frequency) and envelope modulation features (low-frequency). However, standard Octave Convolution still leaves room for optimization in terms of feature fusion methods and frequency distribution ratios between high and low-frequency components.

To address these challenges, this paper proposes a bearing fault diagnosis method that deeply integrates wavelet transform time-frequency representations with an improved Octave Convolutional Neural Network (Oct-CNN). The core idea of this method is to use wavelet transform to generate high-resolution, multi-scale time-frequency features, extracting fault-sensitive patterns from the signal's transient variations, frequency evolution, and local structures. Additionally, an improved Octave Convolution structure is introduced to establish efficient information exchange mechanisms between high and low-frequency feature flows, enabling the decomposition, association, and enhancement of multi-scale fault features. The main contributions of this paper are as follows:

1. a time-frequency feature generation module based on wavelet transform is developed to capture the multi-scale dynamic features of non-stationary fault components in bearing vibration signals;

2. an improved Octave Convolution CNN is designed, employing frequency-separation convolutions and cross-frequency feature fusion to effectively reduce computational redundancy while significantly enhancing the network's ability to express complex feature patterns;

3. experimental evaluations on the CWRU bearing dataset and the Qianpeng experimental test-rig dataset demonstrate that the proposed method achieves competitive diagnostic accuracy with relatively low computational requirements under the investigated conditions. These results provide preliminary evidence of its potential for bearing fault diagnosis applications.

2. Fault Diagnosis Method Based on Wavelet Transform and an Improved Octave Convolutional Neural Network

2.1. Framework of the method

The overall process of the proposed model is as follows: The original one-dimensional vibration signal is first transformed into a time-frequency representation using wavelet transform. It is then divided into training, validation, and test sets. The training set is input into a convolutional neural network (CNN) containing the improved octave convolutional blocks for training. The validation set is used to assess the model's generalization ability on unseen data during the training process, while the test set is used to evaluate the fault diagnosis performance of the model after it has converged.

2.1.1. Network Architecture

The network architecture proposed in this paper is illustrated in Figure 1. The input one-dimensional vibration sequence is transformed into a time-frequency image using wavelet transform, which is then fed into a network model consisting of four convolutional blocks. The final convolutional block is an improved octave convolution block. The model is trained using a sparse categorical cross-entropy loss function.

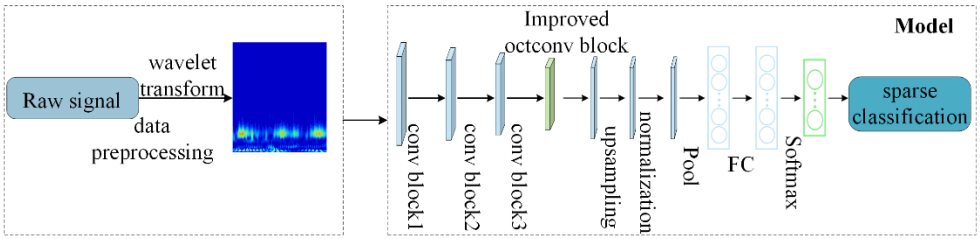

Fig. 1 CBOctconv-CNN

2.2. Wavelet transform

Wavelet Transform (WT) is a mathematical tool widely applied in fields such as signal processing, image analysis, and pattern recognition. Unlike traditional Fourier Transform, wavelet transform provides information about a signal in both the time and frequency domains, exhibiting excellent time-frequency localization properties. The core idea of wavelet transform is to perform multi-scale analysis on the signal to extract information from different frequency bands.

For a one-dimensional signal $x(t)$, the definition of the Continuous Wavelet Transform (CWT) is:

$$W(a,b)=\int_{-\infty}^{\infty}x(t)\psi^{*}\left(\frac{t-b}{a}\right)dt\,, \tag{1}$$

here, $\psi(t)$ represents the mother wavelet, a is the scale factor, b is the translation factor, and ψ^{*} denotes the complex conjugate of the mother wavelet. $W(a, b)$ is the wavelet transform coefficient, reflecting the characteristics of the signal at scale a and position b.

For a two-dimensional signal (such as an image), the extended form of the wavelet transform is given by:

$$W\left(a_x,a_y,b_x,b_y\right)=$$
$$=\int\int_{-\infty}^{\infty}f(x,y)\psi^{*}\left(\frac{x-b_x}{a_x},\frac{y-b_y}{a_y}\right)dxdy\,. \tag{2}$$

In this case, $f(x, y)$ represents the two-dimensional signal (image), a_x and a_y are the scale factors in the horizontal and vertical directions, respectively, and b_x and b_y are the translation factors in the horizontal and vertical directions.

The advantage of wavelet transform lies in its ability to simultaneously capture the local characteristics and multi-scale features of a signal, especially for the analysis of non-stationary signals. Moreover, wavelet transform is effective in extracting features such as edges and textures from images.

Therefore, in this paper, we adopt the Continuous Wavelet Transform (CWT) based on the Morlet mother wavelet for one-dimensional vibration data, generating high-resolution time-frequency representations as input for subsequent deep learning models. The time-frequency representations of normal signals and outer race faults after wavelet transform are shown in Figs. 2 and 3, respectively.

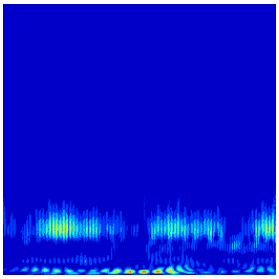

Fig. 2 Normal signal

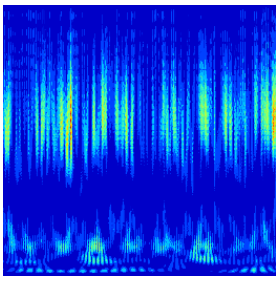

Fig. 3 Outer fault signal

2.3. Backbone network

The backbone network adopted in this paper is the Convolutional Neural Network (CNN). CNN is a deep learning model widely used in fields such as image processing and speech recognition. By simulating the working

principles of the biological visual cortex, CNN utilizes convolutional operations to extract local features from input data, thereby enabling effective pattern recognition and classification.

Let $X \in R^{H\times W\times C}$ represent the input data, where H and W are the height and width of the input, respectively, and C is the number of input channels. The fundamental operation in CNN is the convolution of the input data with a convolutional kernel $K \in R^{k_h\times k_w\times C}$. The mathematical expression for the convolution operation is:

$$Y = X * Y + b\,, \tag{3}$$

here, * denotes the convolution operation, b is the bias term, and Y is the result of the convolution.

The convolutional kernel K automatically learns to extract features from the image.

In CNNs, convolutional layers are typically alternated with pooling layers. The function of the pooling layer is to reduce the size of the feature maps through downsampling, thereby decreasing computational complexity while preserving important feature information. The most commonly used pooling operation is max pooling, and its mathematical expression is:

$$Y_{pool} = max\left(X\left[i,j\right]\right). \tag{4}$$

Pooling operations effectively reduce the number of parameters, thereby mitigating the risk of overfitting. Through successive layers of convolution and pooling, CNNs gradually extract high-level features from images, ultimately performing classification or regression tasks through fully connected layers. The remarkable capabilities of convolutional neural networks have led to revolutionary advancements in computer vision, with widespread applications in tasks such as object recognition and image segmentation.

Given that the time-frequency representations generated by wavelet transform inherently exhibit a multi-scale structure, they distinctly capture both the high-frequency impulsive components and low-frequency steady-state oscillatory characteristics of the signal across different scales. This provides explicit time-frequency information distributions for fault recognition models. To further enhance the deep convolutional network's ability to extract these multi-scale features, this paper replaces the final convolutional block of the traditional CNN architecture with an improved Octave Convolution (OctConv) structure, as illustrated in Fig. 1.

2.4. Octave convolution

In traditional Convolutional Neural Networks (CNNs), convolution operations typically involve performing a weighted sum over local regions of the input data. To enhance the model's ability to learn information from different frequency bands, the innovative method of Octave Convolution has been proposed. Octave Convolution divides the input feature map into two parts: a low-frequency component and a high-frequency component. This division reduces computational complexity while improving the model's ability to capture low-frequency information.

Specifically, let the input feature map be $X \in R^{H\times W\times C}$, where H and W are the height and width, respectively, and C is the number of channels. Through Octave Convolution, the input feature map is decomposed into a low-frequency part $X_L \in R^{H\times W\times C_L}$ and a high-frequency part $X_H \in R^{H\times W\times C_H}$, where C_L and C_H represent the number of channels for the low-frequency and high-frequency components, respectively, and satisfy $C_L + C_H = C$.

The core operation of Octave Convolution involves performing convolutions separately on the low-frequency and high-frequency parts, followed by the fusion of the results. Let the convolutional kernels be $W_L \in R^{k_L\times k_L\times C_L}$ and $W_H \in R^{k_H\times k_H\times C_H}$, where k_L and k_H are the sizes of the low-frequency and high-frequency kernels, respectively. The convolution operation for the low-frequency part can be expressed as:

$$X_L^{'} = X_L * W_L\,. \tag{5}$$

The convolution operation for the high-frequency part is:

$$X_H^{'} = X_H * W_H\,. \tag{6}$$

Finally, the convolution results of the low-frequency and high-frequency components are fused by weighted summation or concatenation, producing the final output feature map:

$$X_{output} = Concat\left(X_H^{'}, X_H^{'}\right). \tag{7}$$

Octave Convolution effectively reduces computational load by allocating computation to low-frequency and high-frequency features, while simultaneously enhancing the model's ability to capture multi-scale information within the data.

2.4.1. Improved octave convolution

Building upon traditional Octave Convolution, this paper further introduces an attention mechanism and designs an adaptive fusion strategy for low- and high-frequency features, enhancing the model's sensitivity to key frequency components and improving its overall representational capacity. The classic Octave Convolution performs branch convolutions and simple fusion between high-frequency and low-frequency features. However, in practical signal or image data, the importance of different frequency bands for discriminative tasks varies, and a fixed fusion strategy may dilute the effective features. To address this, this paper embeds both channel attention and spatial attention modules (Convolutional Block Attention Module, CBAM) into the branches of Octave Convolution, reweighting the importance of low- and high-frequency channels before fusion to enhance feature selectivity.

Specifically, the input features are first divided into high-frequency branch X_H and low-frequency branch X_L according to a scaling factor a. During the convolution phase, the high-frequency channels focus on capturing local discontinuities and rapid variations, while the low-frequency channels emphasize structural, smooth, and global semantic features. After convolution, both branches pass through the

CBAM module, where the channel attention module adaptively strengthens the most discriminative channel features. The spatial attention module further filters the spatial responses within each branch, ensuring optimal enhancement of the high- and low-frequency features before fusion. Compared to the original Octave Convolution, this improvement significantly reduces redundant feature propagation and enhances the directionality and effectiveness of feature representation.

During the fusion phase, considering that the high- and low-frequency branches have different spatial resolutions, this paper employs nearest-neighbor interpolation to upsample the low-frequency branch to the same resolution as the high-frequency branch:

$$X_L^{up} = UpSample\left(X_L\right). \tag{8}$$

Subsequently, the enhanced high-frequency features $X_H^{'}$ are concatenated with the upsampled low-frequency features $X_L^{up'}$, resulting in the fused feature representation:

$$X = Concat\left(X_H^{'}, X_L^{up'}\right). \tag{9}$$

The fused features are then passed through batch normalization and the Swish activation function to enhance the stability of gradient propagation and the network's ability to express non-linearities. This improvement strategy not only preserves the advantages of Octave Convolution in separating features across different frequency bands, but also incorporates an attention mechanism and an adaptive fusion path. This enables the network to intelligently emphasize the most effective frequency components, significantly improving the model's discriminative accuracy in tasks such as fault diagnosis or image classification. The improved Octave Convolution structure is illustrated in Fig. 4.

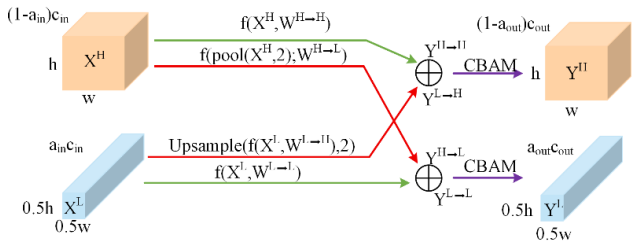

Fig. 4 Improved octave convolution

2.5. CBAM attention mechanism

CBAM (Convolutional Block Attention Module) is a lightweight and efficient attention mechanism composed of channel attention and spatial attention, designed to enhance the feature selection capability of Convolutional Neural Networks (CNNs) for key information.

The channel attention mechanism allocates learnable weights to each channel based on its importance, adjusting the feature responses. This can overcome the information redundancy interference in traditional convolutions, allowing the network to focus on crucial channels while suppressing irrelevant information. The mathematical expression is as follows:

$$\begin{aligned} M_c(F) &= \sigma\left(MLP\left(F_{avgpool}\right) + MLP\left(F_{max\ pool}\right)\right) = \\ &= \sigma\left(W_1(W_0\left(F_{avg}^c\right)\right) + W_1\left(W_0\left(F_{max}^c\right)\right), \end{aligned} \tag{10}$$

here, F represents the feature map input to the network, W_0 and W_1 are the fully connected layers, and σ denotes the sigmoid function.

The spatial attention mechanism empowers the model to uncover the intrinsic relationships between feature maps, enabling different regions of the feature map to exhibit differentiated behavior during feature recognition. It adaptively locates key regions for targeted processing. The feature map is split into several regions, and through average pooling and max pooling, two feature maps $F_{avg}^s \in R^{1\times H\times W}$ and $F_{\max}^s \in R^{1\times H\times W}$ are generated. These are then passed through a convolutional operation to produce the spatial attention map. This process aggregates channel-wise statistical features and learns the fusion, allowing the model to adaptively focus on critical spatial regions. The mathematical expression is as follow:

$$\begin{aligned} M_s(F) &= \sigma\left(f^{7\times7}\left(\left[F_{avgpool}; F_{max\ pool}\right]\right)\right) = \\ &= \sigma\left(f^{7\times7}\left(\left[F_{avg}^s; F_{max}^s\right]\right)\right). \end{aligned} \tag{11}$$

In the lightweight Convolutional Neural Network designed in this paper, the CBAM module is embedded after the Octave Convolution to further optimize the frequency-divided features. The improved Octave Convolution introduces an attention re-weighting mechanism in both the channel and spatial dimensions. CBAM can adaptively regulate the contribution of different frequency components based on frequency band division, thereby enhancing the Octave Convolution's ability to select information across frequency bands. At the same time, the introduction of spatial attention enables the network to more precisely highlight discriminative regions of the feature map before fusion, allowing the multi-scale features extracted by Octave Convolution to focus more on their spatial distribution. Through this attention-enhanced strategy, Octave Convolution maintains its low complexity advantage while providing a more selective and targeted representation of both high- and low-frequency features, effectively improving the model's performance in multi-scale information modeling and fault recognition tasks.

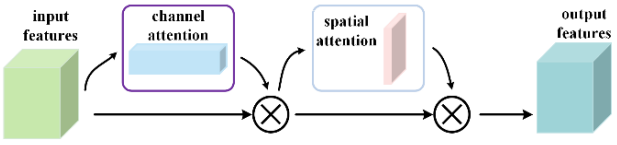

Fig. 5 Convolutional attention mechanism

2.6. Loss function

In the multi-class fault recognition task of this paper, the model is required to learn representative discriminative features from various bearing fault patterns. To this end, Sparse Categorical Cross-Entropy is adopted as the optimization objective, measuring the difference between the predicted distribution and the true class labels. This loss function directly uses category indices rather than one-hot

encoded vectors to represent the labels, which reduces label storage overhead while maintaining the mathematical form consistent with standard cross-entropy. This makes it particularly suitable for deep learning scenarios with a large number of categories or a large sample size. Its mathematical expression can be written as:

$$L_{sparse} = -log\left(\frac{e^{w_y^T f}}{\sum_{k=1}^{N} e^{w_y^T f}}\right), \quad (12)$$

here, N represents the total number of categories, f is the high-dimensional representation output by the feature extraction network, k denotes the classification weight parameter for the k class, and y is the true class index of the sample. The loss function maximizes the logarithmic probability of the true class, gradually guiding the network to converge to a more discriminative feature representation space. While maintaining clarity in the representation, Sparse Categorical Cross-Entropy reduces the dimensional redundancy introduced by one-hot encoding, thereby improving training efficiency and reducing memory overhead. Overall, the combination of Sparse Categorical Cross-Entropy with the Softmax classifier not only stably directs the network's update path but also enhances the separability of multi-class tasks within the feature space. This plays a crucial role in improving the training stability and generalization ability of fault recognition models.

3. Experimental Validation and Analysis

3.1. Dataset

This paper uses the Case Western Reserve University (CWRU) bearing dataset and a proprietary industrial dataset (Qianpeng Dataset) for experimental analysis. The data include normal bearings, outer race faults, inner race faults, and rolling element faults, which are categorized into five health states.

The Qianpeng dataset was acquired from the QPZZ-II rotating machinery fault simulation test bench developed by Jiangsu Qianpeng Diagnostic Engineering Co., Ltd. The QPZZ-II test bench mainly consists of four subsystems, including a mechanical drive system, a loading system, a fixation system, and auxiliary components. The overall structure and main components of the QPZZ-II test bench are shown in Fig. 6.

Owing to its modular configuration and flexible adjustment of component installation positions, the platform can simulate various rotating machinery fault conditions under different operating scenarios. To reproduce realistic bearing damage, artificial defects were introduced using electrical discharge machining. Specifically, single-point spalling defects were introduced at three representative contact locations, including the inner-race contact surface, the outer-race contact surface, and the rolling element surface. Accordingly, the dataset contains three typical bearing fault categories, namely inner race faults, outer race faults, and rolling element faults, with different defect severities and operating conditions. Together with the normal condition, these fault configurations constitute the five health states used in this study. The detailed fault sizes, loading conditions, labels, and sample distributions are summarized in Table 2.

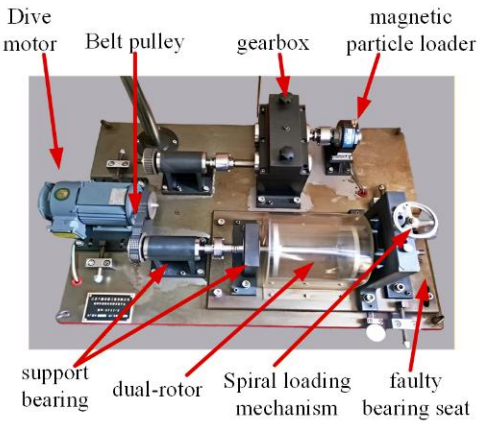

Fig. 6 QPZZ-II test rig

To strictly prevent potential data leakage, a source-level chronological splitting protocol was implemented prior to data generation. Specifically, each continuous 1D raw vibration signal from the drive end was sequentially partitioned along the time axis into training, validation, and test segments in an 8:1:1 ratio. After this physical isolation, sliding windows were independently applied within each data subset to generate time-frequency images. Furthermore, to maintain a balanced sample size across different categories (as shown in Tables 1 and 2), an adaptive overlapping strategy was utilized during the windowing process. Overlapping windows were permitted only within the same data subset, whereas no window could cross the boundary between the training, validation, and test partitions. This rigorous setup completely eliminates temporal overlap between the training and test sets, ensuring that the test samples are strictly unseen data.

Table 1
Case Western Reserve University Bearing Dataset

Location	Size, inch	Speed, rpm	Label	Train	Val	Test
Normal	0	1796	F1	320	40	40
Outer	0.021	1771	F2	320	40	40
Roller	0.021	1729	F3	320	40	40
Inner	0.021	1752	F4	320	40	40
Outer	0.014	1749	F5	320	40	40

Table 2
Qianpeng Dataset

Location	Size. mm	Load, kg	Label	Train	Val	Test
Normal	0	0	C1	320	40	40
Outer	2	80	C2	320	40	40
Roller	1	110	C3	320	40	40
Inner	2	0	C4	320	40	40
Outer	0.5	0	C5	320	40	40

3.2. Experimental details

Prior to model training, the raw one-dimensional signal was preprocessed and converted into a two-dimensional time-frequency representation using Continuous Wavelet Transform. To begin, the input signal was normalized by subtracting its mean and dividing by its standard deviation, where a tiny numerical offset was included to prevent division by zero. A complex Morlet wavelet with a bandwidth parameter of 1.5 and a center frequency of 1.0

was selected as the mother wavelet. The transformation covered a frequency range spanning from a minimum frequency of 10.0 Hz to a maximum of 6000.0 Hz, which represents the Nyquist frequency under a sampling rate of 12000.0 Hz. This frequency range was discretized into 100 linearly spaced bins. The scale parameters for the transform were computed based on the center frequency of the selected wavelet. The squared magnitude of the resulting wavelet coefficients was calculated to obtain the power spectrum, which was subsequently converted to decibels on a logarithmic scale with a small stabilizing constant to generate the final time-frequency scalograms.

Table 3

Architecture and parameter configuration of the proposed OctConv-CBAM network

Stage	Layer	Output size	Configuration
Input	Input	224×224×3	RGB image
Block1	Conv-BN-Swish	224×224×32	3×3Conv, L2=0.002
	Conv-BN-Maxpool-Dropout	112×112×32	3×3Conv, Pool=2×2, p=0.15
Block2	Conv-BN-Swish	112×112×64	3×3Conv, L2=0.002
	Conv-BN-Maxpool-Dropout	56×56×64	Pool=2×2, p=0.2
Block3	Conv-BN-Swish	56×56×128	3×3Conv, L2=0.002
	Conv-BN-Maxpool-Dropout	28×28×128	Pool=2×2, p=0.25
Frequency Split	1×1Conv + AvgPool	HF:28×28×64 LF:14×14×64	α=0.5
Block4	Octconv+CBAM	HF:28×28×128 LF:14×14×128	Channels=256, Reduction=8, Kernel=7×7
Fusion	UpSampling+ Concat	28×28×256	Nearest interpolation
	BN+Swish	28×28×256	Feature normalization
Classifier	GAP	256	
	FC+Swish+Dropout	256	p=0.5
	FC+Swish+Dropout	128	
Output	FC + Softmax	5	Classification

These scalograms were then utilized as inputs for the proposed network. As summarized in Table 3, the proposed model consists of three conventional convolutional blocks, an Octave Convolution (OctConv) block integrated with the Convolutional Block Attention Module (CBAM), and a classifier. The first three convolutional blocks adopt a combination of convolution, batch normalization, Swish activation, max-pooling, and Dropout to progressively extract hierarchical features while reducing the spatial resolution. After the third convolutional block, the extracted features are decomposed into high-frequency and low-frequency branches with a channel allocation ratio of 0.5. The two branches are subsequently processed by the OctConv-CBAM module to facilitate cross-frequency feature interaction and adaptive feature refinement through channel and spatial attention mechanisms. The enhanced low-frequency features are then upsampled and concatenated with the corresponding high-frequency features, followed by batch normalization and Swish activation to obtain fused feature representations. Finally, global average pooling and two fully connected layers with Dropout and L2 regularization are employed before the Softmax classifier to generate the final fault classification results.

In this experiment, the Adam optimizer was employed to optimize the network parameters. Initially, three different learning rates, namely 0.01, 0.001, and 0.0001, were evaluated. The experimental results demonstrated that the model achieved the highest classification accuracy and the fastest convergence when the learning rate was set to 0.0001. Therefore, the initial learning rate was set to 0.0001. The sparse categorical cross-entropy loss function was adopted for model optimization, while the learning rate decay factor was set to 0.5, and the maximum number of training epochs was fixed at 200.

The batch size was set to 64 for all experiments. Five-fold stratified cross-validation was employed, with the random seed for data partitioning fixed at 42 to ensure reproducibility. During each fold, model selection was performed exclusively according to the validation accuracy. The model achieving the highest validation accuracy was saved as the best model, while early stopping with a patience of 120 epochs was adopted to avoid unnecessary training. The best-performing model of each fold was subsequently evaluated on the independent test set, which was not involved in model training or model selection. The final classification performance was reported in terms of accuracy, macro-averaged precision, recall, and F1-score. The random seed was fixed at 42 for Python, NumPy, TensorFlow, and cross-validation partitioning to improve experimental reproducibility

3.3. Evaluation metrics

To comprehensively assess the performance of the model, four metrics were selected for this experiment: accuracy, precision, recall, and F1 score. Additionally, a confusion matrix was employed for visual analysis of the classification results. The mathematical definitions of these metrics are provided below:

1. *Accuracy* quantifies the proportion of correctly predicted samples relative to the total number of samples, serving as an overall indicator of the model's classification performance. It is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{13}$$

2. *Precision* measures the correctness of the model when predicting a sample to belong to a particular class.

Specifically, it quantifies the proportion of true samples belonging to the class among all samples predicted as that class. It is defined as follows:

$$Precision = \frac{TP}{TP + FP}. \quad (14)$$

3. *Recall* measures the model's ability to identify samples of a specific class. Specifically, it quantifies the proportion of true samples belonging to the class that are correctly identified. It is expressed as follows:

$$Recall = \frac{TP}{TP + FN}. \quad (15)$$

4. *F*1-*Score* is the harmonic mean of precision and recall, used to achieve a balance between the two metrics. It is particularly useful for classification tasks with imbalanced class distributions. It is defined as follows:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (16)$$

5. *Confusion Matrix* visually describes the classification performance of the model by statistically analyzing the correspondence between the predicted and true labels. The confusion matrix not only showcases the model's correct identification of each class but also reveals potential confusions between different fault types. This provides valuable insights for subsequent diagnostic result analysis.

3.4. Experimental

To validate the effectiveness of combining wavelet transform with the improved octave convolution in fault diagnosis tasks, the following approach was undertaken: The images transformed using CWT, GADF, GASF, and MTF were individually input into the CNN, octave convolutional CNN, and improved octave convolutional CNN models for fault diagnosis experiments. The analysis of the experimental results is presented below.

3.4.1. CWRU Dataset

Based on the results presented in Tables 4, 5, and 6, when wavelet transform (CWT) is consistently used as the input image, all three models demonstrate significant advantages in accuracy, achieving approximately 99.47%, 98.00%, and 99.60%, respectively. Compared to the GADF, GASF, and MTF image input methods, CWT exhibits the best performance in terms of precision, recall, and F1 score. This indicates that wavelet transform offers a distinct advantage in extracting fault features, as it captures deeper fault-related information. In comparison to other image transformation methods, CWT provides a more stable and reliable input to the model, thereby effectively enhancing the overall performance of the model.

Further analysis of the data in Tables 3, 4, and 5 reveals that when CWT is fixed as the input image, the CBOctconv-CNN model performs the best among the three models, achieving an accuracy of approximately 99.60%. Compared to the Octconv-CNN and CNN models, the accuracy improves by approximately 1.6% and 0.13%, respectively. This indicates that the CNN model incorporating the improved octave convolution demonstrates superior network representation capability and enhanced sensitivity to critical fault features.

Additionally, the results in Table 7 further demonstrate that the CBOctconv-CNN model significantly reduces the model's parameter count while maintaining a high accuracy. Specifically, the parameter count of CBOctconv-CNN is 709,771, compared to 1,275,429 for CNN, resulting in a reduction of approximately 44%. This substantial parameter compression provides CBOctconv-CNN with clear advantages in computational efficiency and resource consumption.

Regarding training time, the CBOctconv-CNN model has a training duration of 59.07 seconds, which is significantly shorter than the 122.49 seconds of Octconv-CNN

Table 4

CWRU-CBOctconv-CNN

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
GADF	97.50±1.30	97.77±1.12	97.50±1.30	97.49±1.31
GASF	91.40±3.40	93.01±2.25	91.40±3.40	91.22±3.56
MTF	81.50±2.70	82.51±7.54	81.50±2.70	79.28±4.87
CWT	99.60±0.58	99.62±0.54	99.60±0.58	99.60±0.58

Table 5

CWRU-Octconv-CNN

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
GADF	98.30±0.81	98.39±0.77	98.30±0.81	98.29±0.82
GASF	90.60±1.85	92.89±1.34	90.60±1.85	90.44±1.69
MTF	81.10±3.95	82.54±8.49	81.10±3.95	78.13±6.49
CWT	99.30±0.68	99.34±0.3	99.30±0.68	99.30±0.67

Table 6

CWRU-CNN

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
GADF	97.60±0.49	97.87±0.40	97.60±0.49	97.61±0.49
GASF	92.10±4.73	94.03±3.01	92.10±4.73	91.53±5.34
MTF	80.70±2.27	86.04±1.86	80.70±2.27	77.92±4.98
CWT	98.80±0.75	98.87±0.70	98.80±0.75	98.80±0.75

and 66.09 seconds of CNN. This further indicates that CBOctConv-CNN achieves competitive diagnostic accuracy with fewer trainable parameters and shorter training time under the evaluated experimental settings. These findings suggest potential advantages for computationally constrained applications.

Table 7

Performance comparison

Model	Parameter	Training time, s
CBOctconv-CNN	709,771	59.07
Octconv-CNN	701,093	122.49
CNN	1,275,429	66.09

Fig. 7 illustrates the clustering results based on CWT and the CBOctconv-CNN model. It is evident that the data points exhibit a clear clustering structure, with points of different colors distinctly grouped into multiple clusters, and significant separation between the clusters. This indicates that the proposed CBOctconv-CNN method effectively differentiates between data from different categories in both feature extraction and classification tasks. The clustering results also suggest high similarity within each category and significant distinguishability between different categories. The t-SNE visualization shows relatively compact intra-class clusters and distinguishable inter-class distributions, suggesting that the proposed model learns discriminative feature representations for the evaluated fault categories.

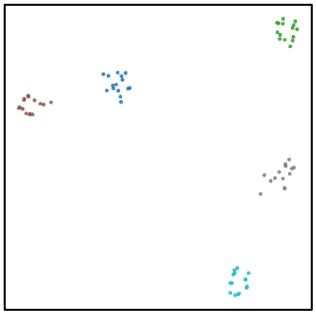

Fig. 7 CWT-CBOctconv-CNN Clustering

Table 8 presents a quantitative comparison between the proposed CBOctconv-CNN model and several state-of-the-art deep learning methods. The experimental results demonstrate that the CBOctconv-CNN achieves consistently better overall performance than InceptionV3, MobileNetV2, and Transformer across all evaluation metrics. Specifically, the proposed model attains an average diagnostic accuracy of 99.30%, which outperforms InceptionV3, MobileNetV2, and Transformer by 0.80%, 0.80%, and 0.90%, respectively. Furthermore, the precision, recall, and F1-score of the CBOctconv-CNN all exceed 99.30%, verifying its outstanding classification capacity and well-balanced diagnostic performance for various bearing fault categories. Notably, as shown in Table 9 and Table 8, ResNet18 exhibits a marginal advantage of 0.10% in average accuracy over the proposed method; however, it requires approximately 15.75 times more parameters and 1.77 times longer training time than CBOctconv-CNN, indicating that the proposed model achieves a better trade-off between diagnostic performance and computational efficiency.

These results fully validate the competitive fault diagnostic performance of the proposed CBOctconv-CNN. The superior performance can be attributed to the efficient feature extraction capability of octave convolution and the adaptive feature refinement function of the convolutional block attention module (CBAM). This further confirms the effectiveness of the designed network architecture in extracting discriminative time-frequency features from bearing vibration signals, thereby significantly improving the accuracy and reliability of mechanical fault diagnosis.

To further evaluate the computational efficiency of the proposed model, the number of trainable parameters and training time were compared with those of several representative baseline models. As shown in Table 9, the proposed CBOctconv-CNN contains only 709,771 trainable parameters and requires 59.07 s for training, both of which are the lowest among all compared models. Compared with InceptionV3, MobileNetV2, ResNet18, and Transformer, the parameter count is reduced by approximately 96.74%, 68.18%, 93.65%, and 12.43%, respectively. Meanwhile, the corresponding training time is reduced by approximately 91.28%, 54.19%, 43.44%, and 35.12%. These results indicate that CBOctconv-CNN achieves substantially lower computational overhead while maintaining a compact network structure, demonstrating its advantages in computational efficiency and potential suitability for deployment in resource-constrained fault diagnosis applications.

Table 8

Performance Comparison of Different Fault Diagnosis Models

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
InceptionV3	98.50±0.55	98.53±0.55	98.50±0.55	98.49±0.55
MobileV2	98.50±0.32	98.57±0.30	98.50±0.32	98.50±0.31
Resnet18	99.40±0.58	99.43±0.54	99.40±0.58	99.40±0.58
Transformer	98.40±0.20	98.50±0.21	98.40±0.20	98.40±0.20
CBOctconv-CNN	99.30±0.68	99.34±0.3	99.30±0.68	99.30±0.67

Table 9

Computational efficiency comparison of different fault diagnosis models

Model	Parameter	Training time, s
InceptionV3	21,795,813	677.78
MobileV2	2,230,277	128.94
Resnet18	11,179,077	104.43
Transformer	810,501	91.04
CBOctconv-CNN	709,771	59.07

3.4.2. QianPeng Dataset

Based on the experimental results presented in Tables 10,11, and 12, we conduct an in-depth analysis of the different models, image transformation methods, and their respective performance.

First, the CBOctconv-CNN model demonstrates higher accuracy across all image transformation methods, showing a significant improvement compared to the traditional CNN (Table 9). Notably, under the GADF method,

the accuracy increased from 72.00% to 82.43%. This performance enhancement is attributed to the introduction of the improved octave convolution (CBOctConv) structure, which separates the processing of high-frequency and low-frequency information. By reducing redundant features while effectively preserving the signal details, this structure enhances the model's expressive capability and computational efficiency.

Furthermore, in the CBOctconv-CNN model, the CWT method achieved 100% accuracy, precision, recall, and F1 score, demonstrating its ability to precisely extract time-frequency features in fault detection tasks. In contrast, other image transformation methods such as GADF, GASF, and MTF showed significantly lower performance in terms of accuracy and F1 score, further validating the superiority of wavelet transform in capturing the instantaneous features of non-stationary signals.

In summary, the model combining the CB-Octconv structure with CWT wavelet transform not only enhances the effectiveness of octave convolution in feature extraction but also fully leverages the advantages of CWT in time-frequency domain feature extraction.

Table 10

QP-CBOctconv-CNN

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
GADF	79.70±2.48	81.50±2.19	79.70±2.48	79.46±2.51
GASF	60.10±2.71	69.08±2.01	60.10±2.71	58.07±3.99
MTF	55.90±1.62	57.97±4.19	55.90±1.62	52.18±1.60
CWT	98.10±0.22	98.12±0.24	98.10±0.22	98.09±0.22

Table 11

QP-Octconv-CNN

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
GADF	78.80±3.36	81.91±2.68	78.80±3.36	78.33±3.44
GASF	62.30±4.98	69.41±2.21	62.30±4.98	61.30±5.77
MTF	57.60±4.57	61.14±7.24	57.60±4.57	55.57±5.86
CWT	97.50±0.63	97.52±0.64	97.50±0.63	97.48±0.64

Table 12

QP-CNN

Method	*Accuracy*, %	*Precision*, %	*Recall*, %	*F*1 *Score*, %
GADF	78.60±2.75	81.06±2.09	78.60±2.75	78.20±2.63
GASF	61.70±2.46	68.88±2.21	61.70±2.46	60.62±2.76
MTF	57.80±2.93	55.10±1.56	57.80±2.93	54.37±2.39
CWT	96.90±0.66	96.97±0.66	96.90±0.66	96.90±0.67

4. Conclusions

1. This study proposes a bearing fault diagnosis method combining continuous wavelet transform (CWT) with a CBAM-enhanced Octave Convolutional Neural Network (CBOctConv-CNN). CWT is employed to generate multi-scale time-frequency representations from vibration signals, while the proposed network integrates octave convolution and attention mechanisms to refine high- and low-frequency features. This architecture is designed to improve the extraction of discriminative fault features while limiting computational complexity.

2. Experimental evaluations on the Case Western Reserve University (CWRU) bearing dataset and the Qianpeng experimental test-rig dataset demonstrate the effectiveness of the proposed method under the investigated conditions. When CWT representations are used as input, the proposed model achieves classification accuracies of 99.60% and 98.10% on the CWRU and Qianpeng datasets, respectively. These results indicate that the combination of wavelet-based time-frequency representations and attention-enhanced octave convolution can provide effective feature representations for the evaluated bearing fault diagnosis tasks.

3. The proposed CBOctConv-CNN demonstrates a favorable balance between diagnostic accuracy and computational requirements in the conducted experiments. Compared with several representative baseline models, it achieves competitive classification performance with fewer trainable parameters and shorter training time. These findings suggest its potential for applications with limited computational resources.

4. Future research will focus on evaluating the proposed method under a wider range of operating conditions, including varying rotational speeds, loads, and noise levels. Cross-domain validation and experiments using additional real-world industrial data will be conducted to assess generalization and robustness more comprehensively. Furthermore, deployment-oriented evaluations, including inference latency, memory requirements, and computational performance on embedded platforms, will be necessary to investigate its practical industrial applicability.

References

1. **Zhang, L.; Qiu, J.; Liu, J.; Xiao, Q.; Wang, C.; Zhou, P.; Zhang, Y.; Sun, C.; Luo, Y.** 2025. CTNet: an interpretable convolutional neural network for bearing fault diagnosis, Structural Health Monitoring: 14759217251382151. https://doi.org/10.1177/14759217251382151.
2. **Zhang, L.; Zhang, K.; Liu, J.; Wang, C.; Song, Y.; Pan, S.** 2026. A bearing fault diagnosis method using bitmap image encoding and multimodal feature fusion, Measurement 258: 119430. https://doi.org/10.1016/j.measurement.2025.119430.

3. **Xu, Z.; Chen, X.; Xu, J.** 2025. Multi-modal multi-sensor feature fusion spiking neural network algorithm for early bearing weak fault diagnosis, Engineering Applications of Artificial Intelligence 141: 109845. https://doi.org/10.1016/j.engappai.2024.109845.
4. **Ciabattoni, L.; Ferracuti, F.; Freddi, A.; Monteriu, A.** 2018. Statistical Spectral Analysis for Fault Diagnosis of Rotating Machines, IEEE Transactions on Industrial Electronics 65(5): 4301-4310. https://doi.org/10.1109/TIE.2017.2762623.
5. **Fang, X.; Zheng, J.; Jiang, B.** 2025. A rolling bearing fault diagnosis method based on vibro-acoustic data fusion and fast Fourier transform (FFT), International Journal of Data Science and Analytics 20: 2377-2386. https://doi.org/10.1007/s41060-024-00609-7.
6. **Cheng, F.; Qu, L.; Qiao, W.; Wei, C.; Hao, L.** 2019. Fault Diagnosis of Wind Turbine Gearboxes Based on DFIG Stator Current Envelope Analysis, IEEE Transactions on Sustainable Energy 10(3): 1044-1053. https://doi.org/10.1109/TSTE.2018.2859764.
7. **Sun, Y.; Li, S.; Wang, X.** 2021. Bearing fault diagnosis based on EMD and improved Chebyshev distance in SDP image, Measurement 176: 109100. https://doi.org/10.1016/j.measurement.2021.109100.
8. **Han, M.; Wu, Y.; Wang, Y.; Liu, W.** 2021. Roller bearing fault diagnosis based on LMD and multi-scale symbolic dynamic information entropy, Journal of Mechanical Science and Technology 35: 1993-2005. https://doi.org/10.1007/s12206-021-0417-3.
9. **Yi, C.; Wang, H.; Ran, L.; Zhou, L.; Lin, J.** 2022. Power spectral density-guided variational mode decomposition for the compound fault diagnosis of rolling bearings, Measurement 199: 111494. https://doi.org/10.1016/j.measurement.2022.111494.
10. **Li, Y.; Song, L.; Sun, Q.; Xu, H.; Li, X.; Fang, Z.; Yao, W.** 2022. Rolling bearing fault diagnosis based on quantum LS-SVM, EPJ Quantum Technology 9: 18. https://doi.org/10.1140/epjqt/s40507-022-00137-y.
11. **Tang, M.; Liang, Z.; Wu, H.; Wang, Z.** 2021. Fault Diagnosis Method for Wind Turbine Gearboxes Based on IWOA-RF, Energies 14(19): 6283. https://doi.org/10.3390/en14196283.
12. **Bing, W.; Xiong, H.** 2025. Fault diagnosis technique based on multi-domain features and RelifF-bayes-KNN in rolling bearing, Nonlinear Dynamics 113: 12985-13000. https://doi.org/10.1007/s11071-024-10212-x.
13. **Dong, Z.; Zhao, D.; Cui, L.** 2024. An intelligent bearing fault diagnosis framework: one-dimensional improved self-attention-enhanced CNN and empirical wavelet transform, Nonlinear Dynamics 112: 6439-6459. https://doi.org/10.1007/s11071-024-09389-y.
14. **Wang, D.; Zhao, Y.; Yi, C.; Tsui, K.-L.; Lin, J.** 2018. Sparsity guided empirical wavelet transform for fault diagnosis of rolling element bearings, Mechanical Systems and Signal Processing 101: 292-308. https://doi.org/10.1016/j.ymssp.2017.08.038.
15. **Liu, Z.; Meng, L.; Tan, Y.; Zhang, J.; Zhang, H.** 2021. Image compression based on octave convolution and semantic segmentation, Knowledge-Based Systems 228: 107254. https://doi.org/10.1016/j.knosys.2021.107254.

T.T. Zhang, J.P. Yu, P.Y. Yang

FAULT DIAGNOSIS BASED ON WAVELET TRANSFORM AND AN IMPROVED OCTAVE CONVOLUTIONAL NEURAL NETWORK

S u m m a r y

This study proposes a bearing fault diagnosis method combining continuous wavelet transform (CWT) with a CBAM-enhanced Octave Convolutional Neural Network (CBOctConv-CNN). CWT is used to transform one-dimensional vibration signals into multi-scale time-frequency representations, while the proposed network integrates octave convolution and attention mechanisms to enhance the extraction of discriminative features. Experimental evaluations on the CWRU bearing dataset and the Qianpeng experimental test-rig dataset demonstrate competitive diagnostic performance, with classification accuracies of 99.60% and 98.10%, respectively, using CWT inputs. Furthermore, the proposed network achieves a relatively low parameter count and shorter training times compared with several baseline models, indicating a favorable balance between diagnostic performance and computational requirements. These findings demonstrate the effectiveness of combining wavelet-based time-frequency representations with attention-enhanced octave convolution for bearing fault diagnosis under the evaluated experimental conditions and suggest its potential for computationally efficient diagnostic applications.

Keywords: wavelet transform, octave convolution, fault diagnosis, attention mechanism, convolutional neural network.

Received December 16, 2025
Accepted October 15, 2026



This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).