Journal of
*Marine Science
and Engineering*

MDPI

Article

Hydraulic-Pump Fault-Diagnosis Method Based on Mean Spectrogram Bar Graph of Voiceprint and ResNet-50 Model Transfer

Peiyao Zhang [1,2], Wanlu Jiang [1,2,*], Yunfei Zheng [1,2], Shuqing Zhang [3], Sheng Zhang [1,2] and Siyuan Liu [1,2]

[1] Hebei Provincial Key Laboratory of Heavy Machinery Fluid Power Transmission and Control, Yanshan University, Qinhuangdao 066004, China; zhang_py@stumail.ysu.edu.cn (P.Z.); yunfeizheng2001@163.com (Y.Z.); zsq**@yau.edu.cn (S.Z.); liusiyuan@ysu.edu.cn (S.L.)
[2] Key Laboratory of Advanced Forging & Stamping Technology and Science, Ministry of Education of China, Yanshan University, Qinhuangdao 066004, China
[3] School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China; zhshq-yd@163.com
* Correspondence: wljiang@ysu.edu.cn

Abstract: The vibration signal of a pump is often used for analysis in the study of hydraulic-pump fault diagnosis methods. In this study, for the analysis, sound signals were used, which can be used to acquire data in a non-contact manner to expand the use scenarios of hydraulic-pump fault-diagnosis methods. First, the original data are denoised using complete ensemble empirical mode decomposition with adaptive noise and the minimum redundancy maximum relevance algorithm. Second, the noise-reduced data are plotted as mean spectrogram bar graphs, and the datasets are divided. Third, the training set graphs are input into the ResNet-50 network to train the base model for fault diagnosis. Fourth, all the layers of the base model are frozen, except for the fully connected and softmax layers, and the support set graphs are used to train the base model through transfer learning. Finally, a fault diagnosis model is obtained. The model is tested using data from two test pumps, resulting in accuracies of 86.1% and 90.8% and providing evidence for the effectiveness of the proposed method for diagnosing faults in hydraulic plunger pumps.

Keywords: complete ensemble empirical mode decomposition with adaptive noise; voiceprint information; ResNet-50; transfer learning; fault diagnosis; hydraulic plunger pump

check for updates

Citation: Zhang, P.; Jiang, W.; Zheng, Y.; Zhang, S.; Zhang, S.; Liu, S. Hydraulic-Pump Fault-Diagnosis Method Based on Mean Spectrogram Bar Graph of Voiceprint and ResNet-50 Model Transfer. *J. Mar. Sci. Eng.* **2023**, *11*, 1678. https://doi.org/10.3390/jmse11091678

Academic Editor: Dejan Brkić

Received: 4 August 2023
Revised: 20 August 2023
Accepted: 22 August 2023
Published: 25 August 2023

1. Introduction

The hydraulic pump is the power component of the hydraulic system, and the plunger pump has advantages such as a high rated pressure, compact structure, and high volumetric efficiency compared to other types of pumps. Therefore, they are widely used in heavy machinery, national defense equipment, and other fields. Plunger pumps typically operate under high-speed and heavy-load conditions, resulting in a shorter service life than other types of pumps. To prevent the occurrence of serious safety incidents, reduce hydraulic system maintenance costs, and shorten the maintenance duration, it is necessary to conduct research on fault diagnosis methods for plunger pumps.

Currently, vibration signals are commonly used for the fault diagnosis of rotating mechanical components such as bearings, gearboxes, and plunger pumps, and good research results have been obtained [1–4]. However, the collection of vibration signals is contact-based, which is inconvenient or impossible to achieve in some practical applications. In contrast, sound signals, which are generated by vibrations, not only contain rich equipment status information but can also be acquired in a non-contact-based manner, with less influence from spatial restrictions. The use of sound signals for fault diagnosis will significantly expand the application scope of fault diagnosis technologies. In addition, the use of a single sound-level meter to sample sound signals can aid in diagnosing faults in multiple components of equipment, while the collection of vibration signals often requires multiple sensors. Therefore, in this study, sound signals were analyzed, and a fault diagnosis was conducted for a constant-pressure variable-displacement axial plunger pump.

Traditional fault diagnosis techniques usually require complex signal processing tasks, such as noise reduction, filtering, and feature extraction, which require technical expertise and knowledge for implementation. However, with the development of deep-learning technology and the advent of the big data era, feature-based approaches that rely on manual selection will gradually be replaced by deep-learning algorithms. Deep-learning-based methods can adaptively extract features from equipment status data. These features not only contain well-known equipment information but also potential information that can help in identifying and locating faults [5]. Consequently, an increasing number of researchers have been interested in applying deep-learning methods in the field of fault diagnosis. In 2017, Qi Y.M. et al. proposed a fault diagnosis method based on a stacked sparse autoencoder and applied it to the diagnosis of rotating machinery faults [6]. In the same year, Verstraete D. et al. generated images from the time–frequency information of the raw data and inputted them into a deep convolutional neural network (CNN) for fault diagnosis. This method was tested using a publicly available rolling–bearing dataset [7]. In 2019, Mao W.T. et al. proposed a fault diagnosis method based on a generative adversarial network (GAN), considering the data imbalance problem in fault diagnosis. The proposed method was demonstrated to address the data-imbalance fault-diagnosis problem in bearing datasets [8]. In the same year, Peng D.D. et al. proposed a one-dimensional deep CNN based on a new residual block and tested it on a rolling–bearing dataset, thus achieving a good diagnostic performance [9]. In 2020, Li X.Q. et al. proposed an enhanced selective ensemble deep learning method and used a beetle antennae search algorithm to improve the performance of a rolling–bearing fault diagnosis algorithm. The experimental results showed that this method is more accurate and robust than baseline models and other ensemble learning methods [10]. In 2021, Li T.F. et al. proposed an effective intelligent fault diagnosis method based on a multireceptive field-graph convolutional network. This method has the advantages of strong data relationship mining and feature representation effects. By converting data samples into weighted graphs and extracting and fusing features from multiple receptive fields, this method solves the limitations of existing graph convolutional networks in terms of weights and receptive fields [11]. In 2022, Tang S.N. et al. used a Bayesian optimization algorithm to automatically select the hyperparameters of CNN models for the intelligent fault diagnosis of hydraulic axial plunger pumps. The results indicated that the proposed method provides excellent performance [12].

ResNet-50, which is used in this study, is a deep learning structure proposed by He K.M. et al., and along with it are other models with different network depths, such as ResNet-18, ResNet-34, ResNet-101, and ResNet-152 [13]. These models are composed of residual blocks and are therefore also known as deep residual networks (DRNs). Normalized initialization and intermediate normalization can largely solve the problems of vanishing gradients and exploding gradients caused by the increasing network depth. The DRN is designed to solve this degradation problem. The degradation problem is not overfitting, which refers to the phenomenon in which, as the network depth increases, the training accuracy quickly decreases after reaching saturation; in contrast, overfitting results in extremely high accuracy in the training set but low accuracy in the testing set.

However, deep learning has significant limitations, which can be summarized as follows. Firstly, the effectiveness of deep learning largely depends on large-scale samples, and the lack of prior knowledge may lead to results that deviate from human senses or expert knowledge. Secondly, deep learning is essentially mapping—i.e., a feature relationship between the input and output—and it lacks causal reasoning. Thirdly, deep learning lacks interpretability, as it is an end-to-end model that contains numerous neurons and parameters, and people cannot explain the meaning of each parameter clearly, which is also one of the greatest shortcomings of deep learning [14].

With the development of deep learning, the serious problem of dependence on large sample sizes has become apparent. Researchers have attempted to use models trained for a specific problem to solve a different but related problem. This learning method is known as transfer learning. Compared with training a separate model for the second problem, this approach can reduce the data and time of the required training. In 1993, Pratt L. proposed a discriminability-based transfer algorithm that can achieve transfer between neural networks and described in detail how to use transfer learning to leverage existing neural networks [15]. In 2020, Chen Z.Y. et al. proposed a transferable CNN to improve target tasks and applied this network to the fault diagnosis of rotating machinery [16]. In 2021, Deng Y.F. et al. proposed a method based on a double-layer attention-based GAN (DA-GAN) to address the problem of local transfer in the field of mechanical fault diagnosis. The DA-GAN method constructs two attention matrices that guide the model to focus on or ignore certain data components before domain adaptation [17]. In 2022, Wang Z.J. et al. proposed a fault diagnosis model called the subdomain adaptation transfer learning network, which not only extracts transferable features but also constructs target subdomains for each category using pseudo-label learning. It also reduces the bias of marginal and conditional distributions and adaptively adjusts the contribution of each network layer using dynamic weight terms [18].

The aim of this study is to investigate the constant-pressure variable-displacement axial plunger pump, and the noise reduction of the pump's sound signals is achieved through complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) and maximum relevance minimum redundancy (mRMR) algorithms. The denoised signals are then plotted as mean spectrogram bar graphs. Subsequently, the ResNet-50 model is used to extract features and classify the graphs generated in the previous step, thus completing the training of the source domain model for the fault diagnosis of a single pump. Finally, through transfer learning, the final diagnostic model is generalized and is able to diagnose faults in multiple pumps with minimal training.

The first part of this article is the introduction, which introduces the research content, research background, and research purpose. The second part is the theoretical background, which details the relevant information and mathematical derivation of the proposed algorithm, including CEEMDAN, the generation of the mean spectrogram bar graph, and the transfer learning algorithm based on the ResNet-50 model. The third part comprises the test and processes, wherein the pump fault simulation test and the dataset generated from it are introduced. The data are then input into the diagnostic algorithm, and the results of each step are presented and analyzed. The fourth part comprises the conclusions, which summarize this article and highlight the advantages of the proposed method. Finally, an outlook for future research on this topic is presented.

2. Theoretical Background

2.1. CEEMDAN

In 1998, Huang N.E. et al. proposed an empirical mode decomposition (EMD) algorithm, which can adaptively decompose signals into a finite number of intrinsic mode functions (IMFs) and is suitable for analyzing non-linear and non-stationary signals [19]. In 2009, Wu Z. et al. proposed an ensemble empirical mode decomposition (EEMD) algorithm to solve the endpoint effect and mode mixing problems in EMD by adding auxiliary white noise [20]. In 2011, Torres M.E. et al. proposed the CEEMDAN algorithm based on EEMD, which eliminates the noise residue caused by adding auxiliary white noise [21]. The EMD serves as the foundation for the EEMDAN algorithm; therefore, it is necessary to introduce the EMD before discussing the CEEMDAN.

2.1.1. EMD

The basic concept of the EMD algorithm is to decompose a signal sequence until the decomposition-stopping conditions are satisfied. The conditions are as follows: (1) The number of extreme points in the current component is less than two and (2) the remainder of the current component cannot be decomposed further. A flowchart of the EMD algorithm is presented in Figure 1.

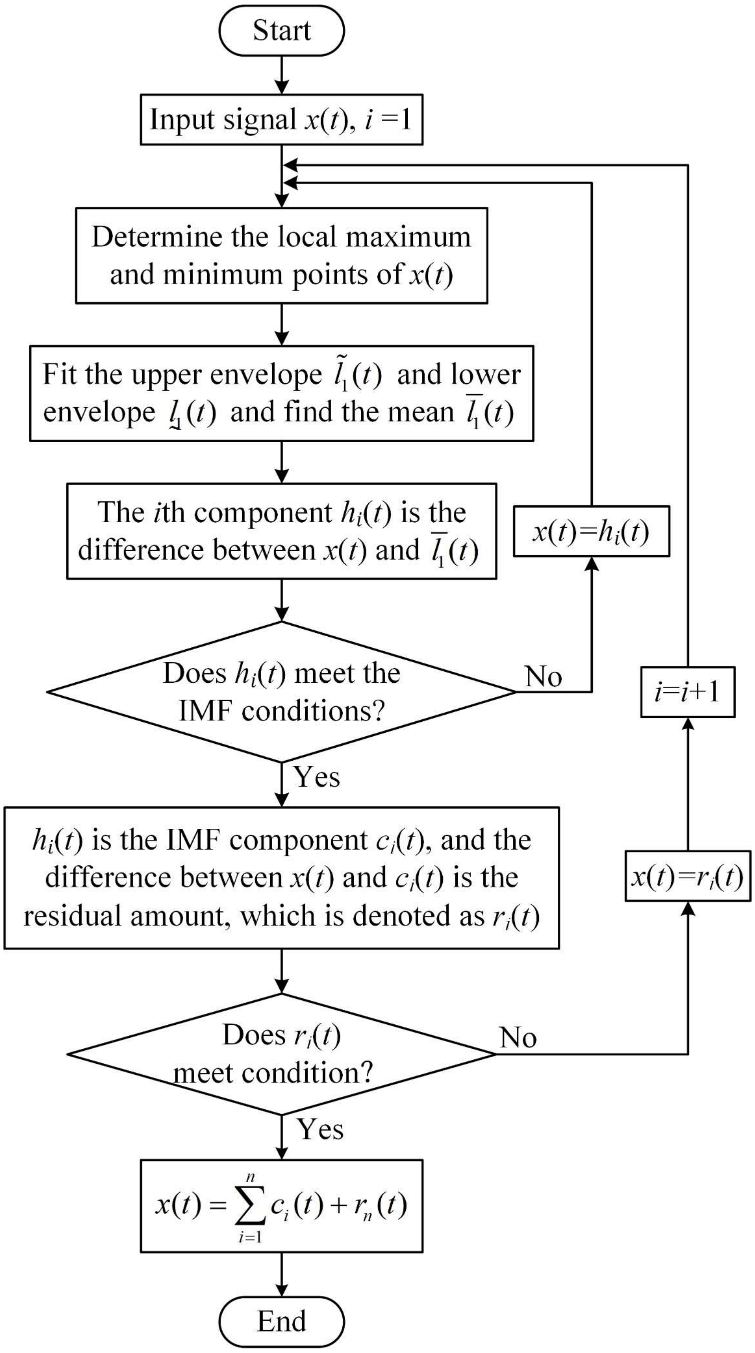

Start
Input signal $x(t)$, i =1
Determine the local maximum and minimum points of $x(t)$
Fit the upper envelope $\widetilde{l}_1(t)$ and lower envelope $\underset{\sim}{l}_1(t)$ and find the mean $\overline{l}_1(t)$
The ith component $h_i(t)$ is the difference between $x(t)$ and $\overline{l}_1(t)$
$x(t)=h_i(t)$
Does $h_i(t)$ meet the IMF conditions?
No
Yes
$i=i+1$
$h_i(t)$ is the IMF component $c_i(t)$, and the difference between $x(t)$ and $c_i(t)$ is the residual amount, which is denoted as $r_i(t)$
$x(t)=r_i(t)$
Does $r_i(t)$ meet condition?
No
Yes
$x(t)=\sum_{i=1}^{n} c_i(t)+r_n(t)$
End

Figure 1. Flowchart of the EMD algorithm.

The specific calculation steps of the EMD algorithm are as follows:

Step 1: All the local maximum and minimum points of the original signal sequence $x(t)$ are found, and spline functions are used to fit the upper envelope $\widetilde{l}_1(t)$ and lower envelope $\underset{\sim}{l}_1(t)$. The average line $\bar{l}_1(t)$ of the upper and lower envelopes are then calculated:

$$\bar{l}_1(t) = \frac{\widetilde{l}_1(t) + \underset{\sim}{l}_1(t)}{2}. \tag{1}$$

Step 2: The average line $\bar{l}_1(t)$ is subtracted from the original signal sequence $x(t)$ to obtain the new sequence $h_1(t)$:

$$h_1(t) = x(t) - \bar{l}_1(t). \tag{2}$$

Typically, $h_1(t)$ is not considered the first IMF component. Therefore, it is necessary to repeat the above steps k times until the average line $\bar{l}_{1k}(t)$ tends to zero and obtain the first IMF component $c_1(t)$:

$$c_1(t) = h_{1(k-1)}(t) - \bar{l}_{1k}(t). \tag{3}$$

Step 3: $c_1(t)$ is subtracted from the original sequence $x(t)$, which results in the residual sequence $r_1(t)$:

$$r_1(t) = x_1(t) - c_1(t), \tag{4}$$

The second IMF component $c_2(t)$ is obtained by repeating the first and second steps for $r_1(t)$. This process is repeated until the n-th residual sequence $r_n(t)$, which cannot be decomposed, is obtained as:

$$r_n(t) = r_{n-1}(t) - c_n(t). \tag{5}$$

At this point, the EMD decomposition process is completed, and $r_n(t)$ is called the residue. The original signal $x(t)$ can be reconstructed by the various IMF components and residue:

$$x(t) = \sum_{i=1}^{n} c_i(t) + r_n(t). \tag{6}$$

Since the emergence of the EMD algorithm, it has been widely applied in several nonlinear fields. However, it also suffers from the problems of endpoint effects and mode mixing. Methods such as the mirror method [22], the extrema extension method [23], the parallel extension method [24], and the boundary local characteristic scale extension method [25] have been developed to address the endpoint effects. To address the problem of mode mixing, Wu Z. et al. proposed the EEMD based on EMD, which solves the aforementioned problem by introducing Gaussian white noise, thus making the decomposition results more stable and reliable. However, because of the repeated addition of Gaussian white noise, the reconstructed signal contains more residual noise, which increases the number of IMF components in the decomposition results. To address this problem, Torres M.E. improved the EEMD method and proposed the CEEMDAN algorithm.

2.1.2. CEEMDAN

As in the case of the EEMD, the CEEMDAN adds white noise to the original signal $x(t)$. However, in contrast to EEMD, after the first addition of the complete Gaussian white noise, CEEMDAN adds the IMF components of the white noise obtained using EMD. A flowchart of the CEEMDAN algorithm is presented in Figure 2.

The specific steps of the CEEMDAN algorithm are as follows:

Step 1: The calculation parameters are initialized, including the noise standard deviation ε_i the and number of EMD executions l.

Step 2: Different Gaussian white noises $n_i(t)$ are added to the original signal $x(t)$ to obtain a new signal $x_i(t)$:

$$x_i(t) = x(t) + n_i(t), \tag{7}$$

where $i = 1, 2, \ldots, l$.

Step 3: The EMD is executed l times on $x_i(t)$, the average of l IMF components is calculated, and the first IMF component of the CEEMDAN $\widetilde{c}_1(t)$ is obtained:

$$\widetilde{c}_1(t) = \frac{1}{l} \sum_{i=1}^{l} c_{i1}(t). \tag{8}$$

$\widetilde{c}_1(t)$ is subtracted from the original signal $x(t)$ to obtain the first residue $\widetilde{r}_1(t)$:

$$\widetilde{r}_1(t) = x(t) - \widetilde{c}_1(t), \tag{9}$$

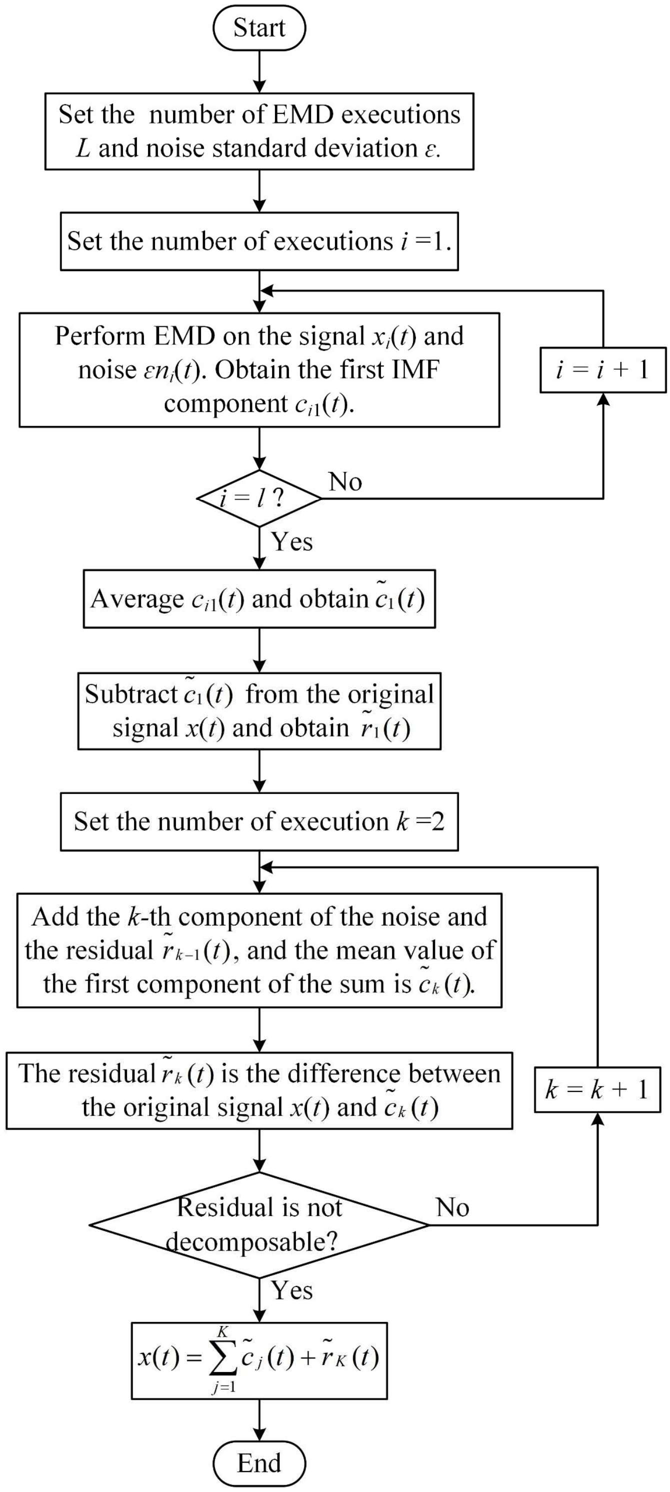

Start
Set the number of EMD executions L and noise standard deviation ε.
Set the number of executions i =1.
Perform EMD on the signal $x_i(t)$ and noise $\varepsilon n_i(t)$. Obtain the first IMF component $c_{i1}(t)$.
$i = l$?
No
$i = i + 1$
Yes
Average $c_{i1}(t)$ and obtain $\tilde{c}_1(t)$
Subtract $\tilde{c}_1(t)$ from the original signal $x(t)$ and obtain $\tilde{r}_1(t)$
Set the number of execution k =2
Add the k-th component of the noise and the residual $\tilde{r}_{k-1}(t)$, and the mean value of the first component of the sum is $\tilde{c}_k(t)$.
The residual $\tilde{r}_k(t)$ is the difference between the original signal $x(t)$ and $\tilde{c}_k(t)$
$k = k + 1$
Residual is not decomposable?
No
Yes
$x(t) = \sum_{j=1}^{K} \tilde{c}_j(t) + \tilde{r}_K(t)$
End

Figure 2. Flowchart of the CEEMDAN algorithm.

Step 4: EMD is executed on l Gaussian white noises $n_i(t)$. The first IMF components are added to $\widetilde{r}_1(t)$, and the l resulting sequences are subjected to EMD. The average of the first components is the second IMF component $\widetilde{c}_2(t)$:

$$\widetilde{c}_2(t) = \frac{1}{l}\sum_{i=1}^{l} E_1(\widetilde{r}_1 + \varepsilon_1 E_1(n_i(t))), \tag{10}$$

where $E_1(\cdot)$ denotes the first IMF component of the sequence in parentheses. Subtracting $\widetilde{c}_2(t)$ from $\widetilde{r}_1(t)$ results in the second residue $\widetilde{r}_2(t)$:

$$\widetilde{r}_2(t) = \widetilde{r}_1(t) - \widetilde{c}_2(t). \tag{11}$$

Step 5: Step 4 is repeated to calculate the remaining IMF components $\widetilde{c}_k(t)$ and residue $\widetilde{r}_k(t)$:

$$\begin{cases} \widetilde{c}_k(t) = \frac{1}{l}\sum\limits_{i=1}^{l} E_1(\widetilde{r}_{k-1} + \varepsilon_{k-1} E_{k-1}(n_i(t))) \\ \widetilde{r}_k(t) = \widetilde{r}_{k-1}(t) - \widetilde{c}_k(t) \end{cases}, \tag{12}$$

where $k = 3, 4, \ldots, K$. The K-th residue cannot be further decomposed. The original signal $x(t)$ is reconstructed from the obtained IMF components and residues:

$$x(t) = \sum_{j=1}^{K} \widetilde{c}_j(t) + \widetilde{r}_K(t). \tag{13}$$

The two important parameters in the CEEMDAN algorithm are the number of iterations l and the noise standard deviation ε_i. The number of executions l is generally set to a large value. Torres M.E. et al. set l as 500 in their study [21]. The standard deviation of the added noise ε_i can be used to select the signal-to-noise ratio (SNR) at each step. Wu Z. et al. suggested that adding small-amplitude noise signals during the decomposition process is beneficial [20]. As a result, ε_i is usually set to a small value, such as 0.02 [21]. In this study, all ε_i are set as the same value.

2.2. Spectrogram and Mean Spectrogram Bar Graph

The spectrogram was invented by researchers at Bell Laboratories in 1941. This is an image that displays sounds in three dimensions. A spectrogram typically presents the intensity of sound using colors, with time along the horizontal axis and frequency along the vertical axis. The spectrogram provides a visual representation of how the different frequency components of a sound signal change over time, thus making it easy to observe the frequency content of the signal at different moments. In 1945, Kesta L.G. et al. completed spectrogram matching using visual observation and first proposed the concept of a voiceprint. Spectrograms can be classified into two types: Wideband and narrowband spectrograms.

The wideband spectrogram is based on the windowed Fourier Transform technique. It divides a long signal into several time segments of equal length, performs a short-time Fourier transform (STFT) on each segment, and then concatenates the multiple frames of the spectra obtained by the STFT in the time dimension with overlap to form an image. Typically, the wideband spectrogram uses a wider frequency range and shorter time windows with shorter intervals between adjacent windows. Therefore, it allows for the observation of the transient properties of the signal in the time domain, thus presenting a more distinct visualization of high-frequency components and transient signals and providing a high temporal resolution.

Narrowband spectrograms are similar to wideband spectrograms, but usually select a relatively narrow frequency band with a high-frequency resolution, thus making them more suitable for analyzing the frequency domain details of audio signals. Considering that the state information of the equipment is more clearly reflected in the frequency domain, in this study, narrowband spectrograms are used for the analysis.

2.2.1. Generating Spectrogram

Generating a spectrogram involves several steps:

Step 1: Pre-emphasis is applied to the input audio signals. As the main analysis object of a spectrogram is human speech, a first-order high-pass filter is typically used to improve the SNR of the high-frequency part of the signal. This is related to the characteristics of speech itself and the weighting method used in the sound sampling process. In the fault diagnosis process, this step can be omitted to reflect the real-state information of the equipment.

Step 2: The pre-emphasized signals are divided into a series of overlapping frames and windowing is applied to each data frame. The frame length affects the temporal resolution of the spectrogram. Frame overlap can smooth the transition between frames and avoid signal leakage caused by window boundaries; it is usually half the frame length. Windowing can reduce the Gibbs phenomenon caused by frame division and is beneficial for accurately extracting the frequency characteristics of the signals in the subsequent Fourier transform. Commonly used window functions include the rectangular, Hamming, and Hann windows. In this study, the Hann window is used, and the window function is represented as follows:

$$w(i) = \begin{cases} 0.5(1-\cos\frac{2\pi i}{N-1}) & (0 \le i \le N-1) \\ 0 & (i < 0 \text{ or } i > N) \end{cases}, \tag{14}$$

where i is the index of the data points within the frame and N is the length of the frame.

Step 3: The STFT is performed on each segment to observe the frequency domain of the signals. The frequency-domain information $X(m,k)$ of the m-th time window at frequency band k is expressed as follows:

$$X(m,k) = \sum_{n=0}^{N-1} x(n)w(n-m)e^{-j2\pi kn/N}, \tag{15}$$

where $x(n)$ is the original signal, $0 \le m \le M-1$, M is the number of frames, and $0 \le k \le \frac{N}{2}-1$.

Step 4: The square of the amplitude of the STFT value is obtained and logarithmically processed. The purpose of logarithmic processing is to amplify low-amplitude components to observe signals masked by low-amplitude noise, the unit of which is dB. The processed signal $Y(m,k)$ is expressed as follows:

$$Y(m,k) = 10\log_{10}|X(m,k)|^2. \tag{16}$$

Step 5: The processed signal $Y(m,k)$ of each frame is concatenated to obtain the $\frac{N}{2} \times M$ spectrogram matrix $\boldsymbol{SP}$:

$$\boldsymbol{SP} = \begin{bmatrix} Y(0,\frac{N}{2}-1) & Y(1,\frac{N}{2}-1) & \cdots & Y(M-1,\frac{N}{2}-1) \\ \vdots & \vdots & \iddots & \vdots \\ Y(0,1) & Y(1,1) & \cdots & Y(M-1,1) \\ Y(0,0) & Y(1,0) & \cdots & Y(M-1,0) \end{bmatrix}. \tag{17}$$

The matrix is plotted as an image, which can be presented in grayscale or pseudocolor to indicate the magnitude of the power. In this article, grayscale images are used. The elements in the matrix $\boldsymbol{SP}$ are mapped to the range of 0–255, and a grayscale image is drawn, where 0 represents black and 255 represents white.

A traditional spectrogram is designed for human speech signals and used for speaker recognition and speech recognition. As the object of this study is mechanical equipment, it is necessary to improve the traditional spectrogram to satisfy the requirements of fault diagnosis.

2.2.2. Mean Spectrogram Bar Graph

Human voice signals have a higher energy in the low-frequency region; therefore, A-weighting is typically used to enhance the high-frequency region during the sampling process. The diagnostic object of this study is a hydraulic plunger pump that rotates periodically during operation. To obtain more realistic sound signals, a sound-level meter with Z-weighting is selected for the test process, which ensures that the sampled signals decay as little as possible over a wide frequency range. Therefore, when generating spectrograms for equipment fault diagnosis, the pre-emphasis processing of the sound signals can be omitted.

Steps 2–5 of the traditional spectrogram generation are retained to generate the ***SP*** matrix, which can be used to draw a mean spectrogram bar graph. To comprehensively monitor the health status of the plunger pump, the sample used to draw a spectrogram should contain at least one rotation period, and the sample length l_s should satisfy the following requirements:

$$l_s \geq \frac{f_s}{n}, \tag{18}$$

where f_s is the sampling frequency and n is the rotational frequency.

Based on the analysis of the sound signals of the plunger pump, its changes in the time domain are relatively small, while the frequency domain can better reflect the fault information. Therefore, after obtaining the matrix ***SP***, the mean value of each row element in the matrix is calculated to obtain the F order vector ***SSP***:

$$\boldsymbol{SSP} = \begin{bmatrix} \frac{1}{M}\sum\limits_{m=1}^{M} Y(m, F-1) \\ \vdots \\ \frac{1}{M}\sum\limits_{m=1}^{M} Y(m, 1) \\ \frac{1}{M}\sum\limits_{m=1}^{M} Y(m, 0) \end{bmatrix}. \tag{19}$$

The elements in the matrix ***SSP*** are mapped to a range of 0–255, and a grayscale image is drawn. This processing not only reduces the amount of data input into the deep learning model and speeds up the model calculation but also helps to eliminate occasional interference when l_s is long. Figure 3 presents the grayscale spectrogram and mean spectrogram bar graphs drawn using the sound data of the plunger pump. The mean spectrogram bar graph is stretched horizontally for better observation.

As shown in Figure 3, owing to the second mapping, the improved spectrogram has a stronger contrast, which can effectively improve the recognizability of the fault characteristics, reduce the learning difficulty, and improve the accuracy of the fault diagnosis.

2.3. Deep Transfer Learning Algorithm Based on ResNet-50

Deep learning, which originates from artificial neural networks, is a machine learning technique that uses a multilayer perceptron with multiple hidden layers to automatically extract abstract features from massive amounts of data, thereby establishing direct relationships between these features and the target outputs. However, deep learning often requires a large number of samples to achieve optimal results. In contrast, transfer learning allows the use of models trained for specific tasks for solving related but different problems. This article proposes a deep transfer learning method that combines deep learning with model-based transfer learning. By integrating these two methods, deep transfer can achieve stronger feature extraction capabilities and end-to-end diagnostic functions with better universality. Moreover, deep transfer learning can be used to efficiently extract features from the data, thereby addressing the problem of insufficient data for fault diagnosis.

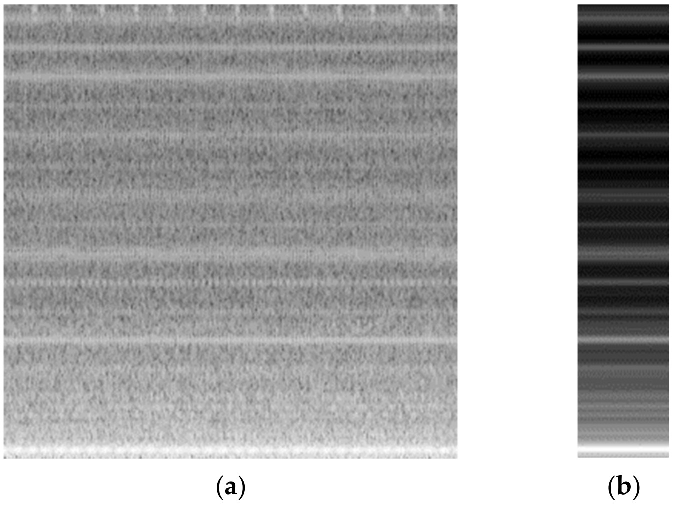


Figure 3. The grayscale spectrogram (**a**) and mean spectrogram bar graph (**b**) created from the sound signal of a plunger pump operating at normal conditions; the latter has undergone stretching processing.

2.3.1. ResNet-50 Deep Learning Model

ResNet-50 achieves interlayer connections through shortcuts, by adding the input to the output after convolution across layers, which fully trains the deeper-level network and improves the accuracy significantly with increasing depth, thus solving the model degradation problem. This structure is called a residual block. In this study, the convolutional residual block (CRB) and identity residual block (IRB) are used, as shown in Figure 4.

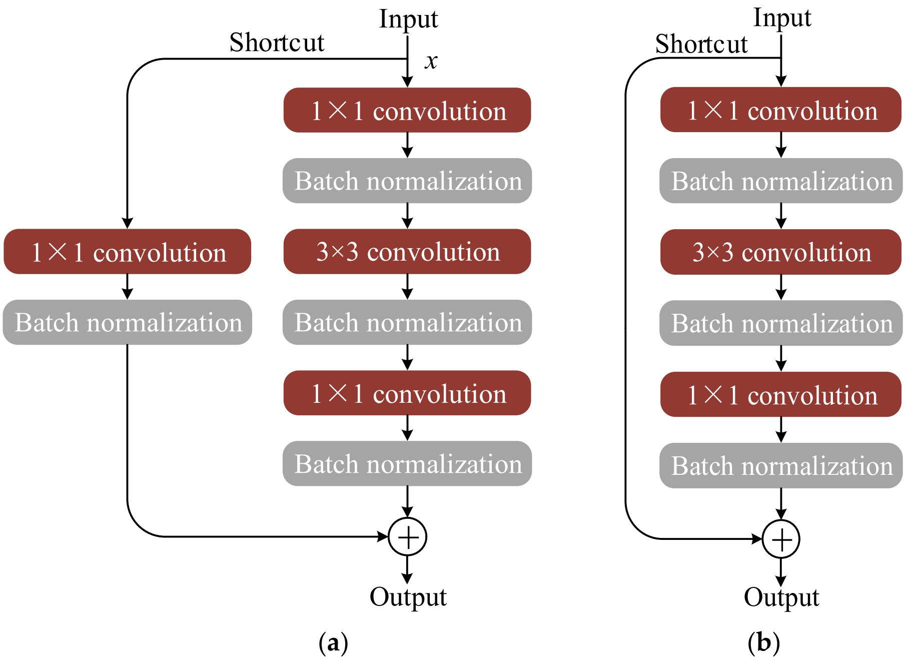


Figure 4. Structure of the CRB (**a**) and IRB (**b**).

The ResNet-50 model used in this study contains 49 convolution layers and 1 fully connected layer, and the activation function used is the rectified linear unit (ReLU). The model structure is presented in Figure 5.

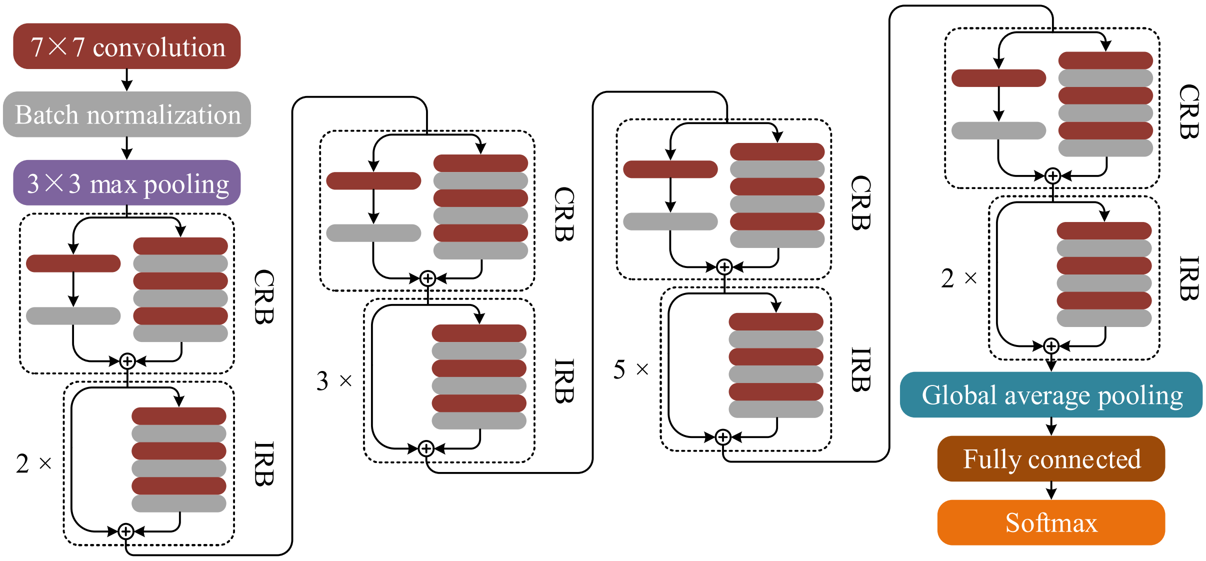


Figure 5. Network structure of ResNet-50.

Let the input sample be $\boldsymbol{x}$, and by computing the convolution of the convolution kernel $\boldsymbol{k}^{\text{inp}}$ and $\boldsymbol{x}$, the feature vector $\boldsymbol{f}^{\text{inp}}$ can be obtained:

$$\boldsymbol{f}^{\text{inp}} = \sigma_{\text{r}}(\boldsymbol{x} * \boldsymbol{k}^{\text{inp}} + \boldsymbol{b}^{\text{inp}}), \tag{20}$$

where σ_{r} represents the ReLU activation function and $\boldsymbol{k}^{\text{inp}}$ and $\boldsymbol{b}^{\text{inp}}$ are trainable parameters.

Batch normalization is performed on $\boldsymbol{f}^{\text{inp}}$ to improve the model training speed. Let $\boldsymbol{f}^{\text{inp}}$ be an $m \times l$ matrix and f_{ij} be the element in the i-th row and j-th column. The element n_{ij}^{inp} in the normalized result $\boldsymbol{n}^{\text{inp}}$ is represented as follows:

$$\begin{cases} \mu_f = \frac{1}{m \times l} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{l} f_{ij} \\ \sigma_f = \frac{1}{m \times l} \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{l} (f_{ij} - \mu_f)^2 \\ \hat{f}_{ij} = \frac{f_{ij} - \mu_f}{\sqrt{{\sigma_f}^2 + e}} \\ n_{ij}^{\text{inp}} = \gamma \hat{f}_{ij} + \beta \end{cases}, \tag{21}$$

where e is a small number added to prevent the mean squared error from becoming equal to zero. n_{ij}^{inp} is the element in the i-th row and j-th column. γ and β are trainable parameters. At this point, the distribution of $\boldsymbol{f}^{\text{inp}}$ is adjusted to a standard normal distribution, which allows the input data to fall into the sensitive region of the activation function and prevents the occurrence of the vanishing gradients problem. However, this can result in a decrease in the expressive power of the network, and the depth of the network loses its meaning. Therefore, it is necessary to perform an inverse operation on the transformed $\boldsymbol{n}^{\text{inp}}$ to enable the model to learn the optimal values of γ and β.

The normalized result is split into several non-overlapping one-dimensional segments, the maximum value of each segment element is returned, and maximum pooling is performed. This can reduce the feature dimensions and number of trainable parameters. The output of the max pooling layer is denoted as $\boldsymbol{P}^{\mathrm{inp}}$, which is a $q \times l$ matrix, where $p^{\mathrm{inp}}_{k,j}$ is the element in the k-th row and j-th column of $\boldsymbol{P}^{\mathrm{inp}}$:

$$p^{\mathrm{inp}}_{k,j} = \max\left\{ n^{\mathrm{inp}}_{i,j} \middle| s(k-1)+1 \le i \le sk \right\}, \tag{22}$$

where s represents the length of the non-overlapping segments.

Subsequently, the abstract features of $\boldsymbol{P}^{\mathrm{inp}}$ are extracted using multiple residual units. The residual unit calculates the sum of the residual function and input feature, where the residual function is a non-linear mapping $F_{\mathrm{r}}(\boldsymbol{X})$ consisting of three convolution layers:

$$F_{\mathrm{r}}(\boldsymbol{X}^{t-1};\boldsymbol{\theta}^t) = \left\{ \sigma_r[\sigma_r(\boldsymbol{X}^{t-1} * \boldsymbol{K}^t_1 + \boldsymbol{B}^t_1) * \boldsymbol{K}^t_2 + \boldsymbol{B}^t_2] * \boldsymbol{K}^t_3 + \boldsymbol{B}^t_3 \right\}, \tag{23}$$

where $\boldsymbol{\theta}^t = \{\boldsymbol{K}^t_1, \boldsymbol{B}^t_1, \boldsymbol{K}^t_2, \boldsymbol{B}^t_2, \boldsymbol{K}^t_3, \boldsymbol{B}^t_3\}$ denotes the trainable parameter set for the t-th residual unit. In this study, the number of residual units is 16 when $t = 1$ and $\boldsymbol{x}^0 = \boldsymbol{P}^{\mathrm{inp}}$.

In IRB, the shortcut connection performs element-wise addition between two features of equal dimensions, which is an identity mapping. The output $\boldsymbol{Y}^t$ of the t-th residual unit can be expressed as follows:

$$\boldsymbol{Y}^t = \sigma_r[F_{\mathrm{r}}(\boldsymbol{X}^{t-1};\boldsymbol{\theta}^t) + \boldsymbol{X}^{t-1}], \tag{24}$$

where $\boldsymbol{Y}^t$ denotes a multidimensional matrix. The IRB does not introduce additional parameters or computational complexity into the model, which has significant advantages in practical applications.

In the CRB, as the output channels of the convolution layers are modified, the dimensions of the features are not equal when they are added; therefore, dimension matching is required to be performed in the shortcut connection (implemented through a 1 × 1 convolution). This increases the network parameters and improves its performance. This process can be represented as follows:

$$\boldsymbol{Y}^t = \sigma_{\mathrm{r}}[F_{\mathrm{r}}(\boldsymbol{X}^{t-1};\boldsymbol{\theta}^t) + (\boldsymbol{X}^{t-1} * \boldsymbol{K}^t_4 + \boldsymbol{B}^t_4)], \tag{25}$$

where $\boldsymbol{K}^t_4$ and $\boldsymbol{B}^t_4$ are parameters to be trained.

After extracting the device fault feature matrix $\boldsymbol{Y}^{16}$ through a 16-layer residual network, Global Average Pooling (GAP) [26] is performed, followed by a fully connected layer. The GAP calculates the average value of each element in the feature matrix of every dimension of $\boldsymbol{Y}^{16}$, thus obtaining a feature vector $\boldsymbol{g}^{\mathrm{out}}$ with a length equal to the dimension of $\boldsymbol{Y}^{16}$. Compared with directly connecting to a fully connected layer, this approach reduces the network parameters and prevents overfitting.

Then, a fully connected layer is connected after the GAP layer, the features are mapped to the label space of the samples, and $\boldsymbol{fc}^{\mathrm{out}}$ is output as follows:

$$\boldsymbol{fc}^{\mathrm{out}} = F_{\mathrm{fc}}(\boldsymbol{g}^{\mathrm{out}};\boldsymbol{\theta}^{\mathrm{fc}}) = \sigma_{\mathrm{r}}(\boldsymbol{w}^{\mathrm{fc}} \cdot \boldsymbol{g}^{\mathrm{out}} + \boldsymbol{b}^{\mathrm{fc}}), \tag{26}$$

where $\boldsymbol{\theta}^{\mathrm{fc}} = \left\{\boldsymbol{w}^{\mathrm{fc}}, \boldsymbol{b}^{\mathrm{fc}}\right\}$ denotes the parameter set of the fully connected layer to be trained.

Finally, the softmax function is used to calculate the probability distribution of $\boldsymbol{fc}^{\mathrm{out}}$ in the label space, and the probability of belonging to the s-th fault type is given as:

$$P(y_{type} = s | \boldsymbol{fc}^{\mathrm{out}}; \boldsymbol{\theta}^{\mathrm{soft}}) = \frac{\exp(\boldsymbol{w}^{\mathrm{soft}}_s \cdot \boldsymbol{fc}^{\mathrm{out}} + \boldsymbol{b}^{\mathrm{soft}}_s)}{\sum\limits_{s=1}^{S} \exp(\boldsymbol{w}^{\mathrm{soft}}_s \cdot \boldsymbol{fc}^{\mathrm{out}} + \boldsymbol{b}^{\mathrm{soft}}_s)}, \tag{27}$$

where $\boldsymbol{\theta}^{\mathrm{soft}} = \left\{\boldsymbol{w}^{\mathrm{soft}}, \boldsymbol{b}^{\mathrm{soft}}\right\}$ denotes the set of parameters to be trained for the softmax layer. In this study, the total number of types $S = 8$. Thus, the device fault diagnosis task based on deep learning is completed.

2.3.2. Model Transfer Learning Based on ResNet-50

There are two basic concepts in transfer learning: Domains and tasks. The existing knowledge is called the source domain, which includes data knowledge and model knowledge. The source domain generates training samples containing rich annotation information. The new knowledge expected to be obtained is called the target domain, which generates test samples, usually without annotations or with only a small amount of annotation information. A task refers to a specific problem that is required to be solved, and a model is required to learn from a large amount of data. Transfer learning can transfer the annotated data and knowledge structure of the source domain to the target domain to complete the task of the target domain. Typically, the tasks of the source and target domains are identical; however, the data distributions are different. Pan et al. referred to this type of transfer learning as transductive transfer learning [27].

In model-based transfer learning, it is assumed that the source domain has a large amount of labeled data $\boldsymbol{D}_{\mathrm{s}} = \{\boldsymbol{x}_i^{\mathrm{s}}, y_i^{\mathrm{s}}\}_{i=1}^{ns}$, where $\boldsymbol{x}_i^{\mathrm{s}}$ represents the i-th sample in the source domain, and y_i^s is its label. ns is the number of samples in the source domain. It is also assumed that the target domain has a small amount of labeled data $\boldsymbol{D}_{\mathrm{t}} = \left\{\boldsymbol{x}_j^{\mathrm{t}}, y_j^{\mathrm{t}}\right\}_{j=1}^{nt}$ wherein $\boldsymbol{x}_j^{\mathrm{t}}$ represents the j-th sample in the target domain, y_j^{t} is its label, and nt is the number of samples in the target domain, where $nt \ll ns$. In model transfer, an attempt is made to transfer the abundant structural knowledge learned from the source domain data to the target domain and obtain a more accurate prediction model using only a small amount of labeled data from the target domain.

When transferring the model in this study, all the layer parameters, except for the fully connected layer, and softmax are frozen, which retains the knowledge of deep learning for the feature extraction of the device states. Transfer learning training is performed using the labeled data in the target domain. Parameter training comprises the use of a stochastic gradient descent with a momentum optimizer, which adds momentum to the stochastic gradient descent, and its iterative update formula for the parameters is as follows:

$$\begin{cases} m_t = \beta m_{t-1} + (1-\beta) g_t \\ \theta_t = \theta_{t-1} - \alpha m_t \end{cases}, \tag{28}$$

where m_t represents the momentum, β is the decay coefficient, typically, $\beta = 0.9$ [28], g_t is the gradient of the objective function with respect to the current parameters, θ_t is the optimized parameter, and α is the initial learning rate, which is obtained from optimization algorithms.

The cost function J_{MSE} uses the mean squared error, which is expressed as:

$$J_{MSE} = \frac{1}{nt} \sum_{j=1}^{nt} (y_j^t - \widetilde{y}_j^t)^2, \tag{29}$$

where $\widetilde{y}_j^t$ is the predicted label for sample $\boldsymbol{x}_j^{\mathrm{t}}$ in the j-th sample of the target domain. L2 regularization is applied to the cost function to prevent overfitting.

3. Test and Calculation

3.1. Fault Simulation Test of the Hydraulic Plunger Pump

In the hydraulic plunger pump fault simulation test, the normal components of the pump are replaced with artificial faulty components to simulate seven types of faults: Shoe wear, port plate wear, loose shoe, plunger wear, and support bearing faults (outer ring, inner ring, and rolling element faults). The test also distinguishes the degree of some faults, which is, however, not introduced in this study. The faulty components used in the tests are presented in Figure 6.

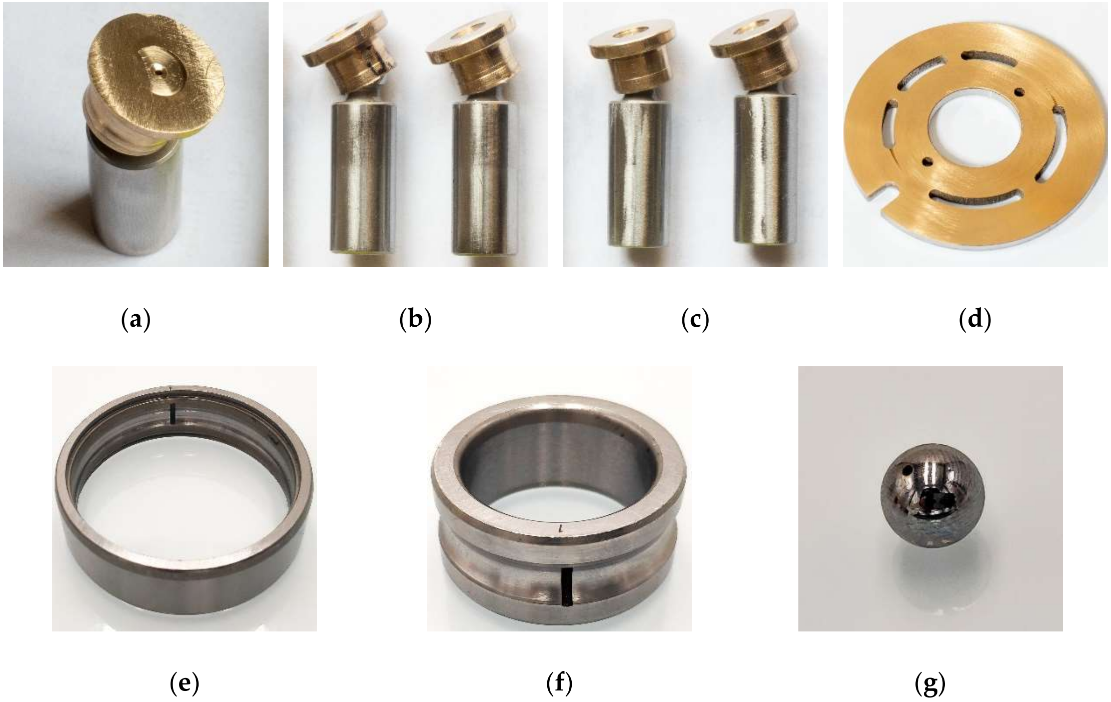


Figure 6. Photographs of various faulty components of plunger pumps used for the testing. (**a**) Wear of shoe; (**b**) loose shoe; (**c**) wear of plunger; (**d**) wear of port plate; (**e**) fault of outer ring; (**f**) fault of inner ring; (**g**) fault of rolling element.

To produce worn shoe, plunger, and port plate components, we used 80-grit sandpaper to polish the surface of the components, and the weight of the former two decreased by 0.4 g while the latter decreased by 1 g. To manufacture the loose-shoe component, we employed a tool to stretch the shoe and plunger, resulting in a 0.3 mm increase in the distance between the two. The three types of fault-bearing components are obtained using electrical discharge machining. For the faulty inner and outer rings of the bearings, a through groove with a width of 1 mm and a depth of 1 mm is machined on the rolling track. For the faulty rolling elements of the bearings, a small pit with a diameter of 1 mm and a depth of 1 mm is machined on a rolling element.

The fault simulation test bench of the hydraulic plunger pump is designed based on a constant-pressure variable-displacement axial plunger pump (see (2) in Figure 7), which uses a proportional relief valve (see (2) in Figure 4) to control the pressure of the test system and adjusts the opening size of the throttle valve (see (2) in Figure 7) to change the load pressure and flow rate. A hydraulic schematic of the test bench is presented in Figure 7.

During the test, various types of fault signals are generated by replacing normal components with faulty components to simulate faulty pumps. In addition to the flow, pressure, and temperature sensors indicated in Figure 7, a sound level meter is also placed 0.5 m directly above the pump to collect sound signals during the pump operation. In the test, the vibration signals from the pump are also sampled; however, this is not discussed in detail in this article. A photograph of the test bench is presented in Figure 8, and the main components and their performance indicators are listed in Table 1.

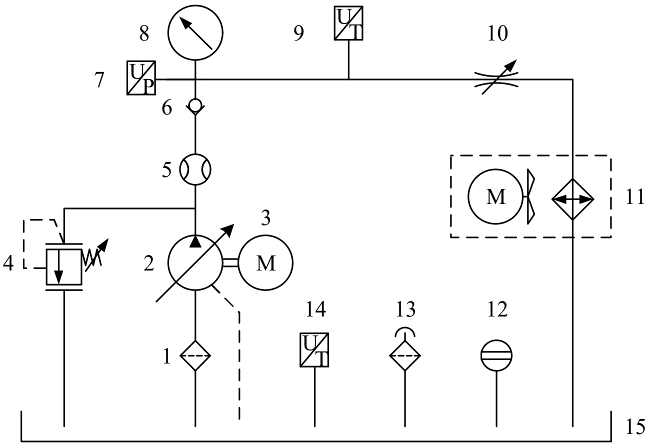


Figure 7. Schematic of Hydraulic system of the hydraulic-plunger-pump fault-simulation test bench. The numbered identifiers in the figure correspond to different members of the system. 1: Oil absorption filter; 2: Constant-pressure variable-displacement axial plunger pump; 3: Electric motor; 4: Proportional relief valve; 5: Flowmeter; 6: Check valve; 7: Pressure sensor; 8: Pressure gauge; 9, 14: Temperature sensor; 10: Throttle valve; 11: Air-cooled unit; 12: Liquid level gauge; 13: Air filter; 15: Oil tank.

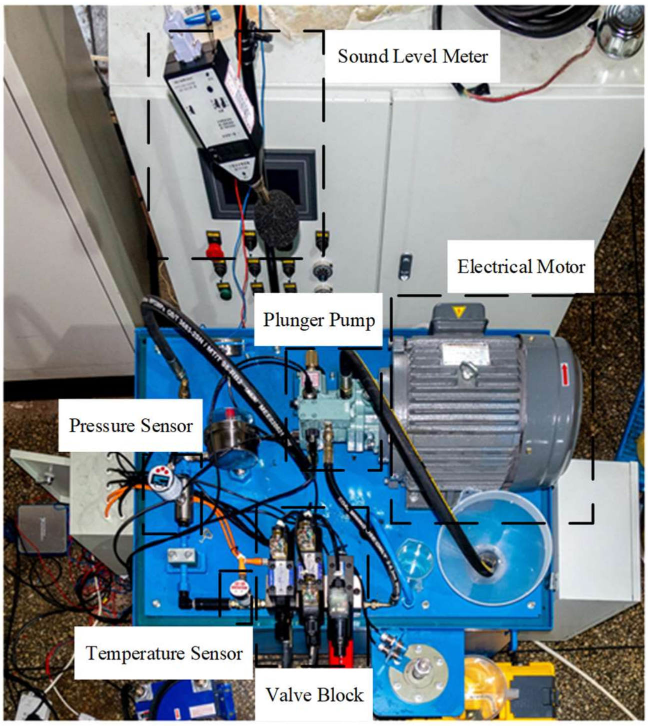
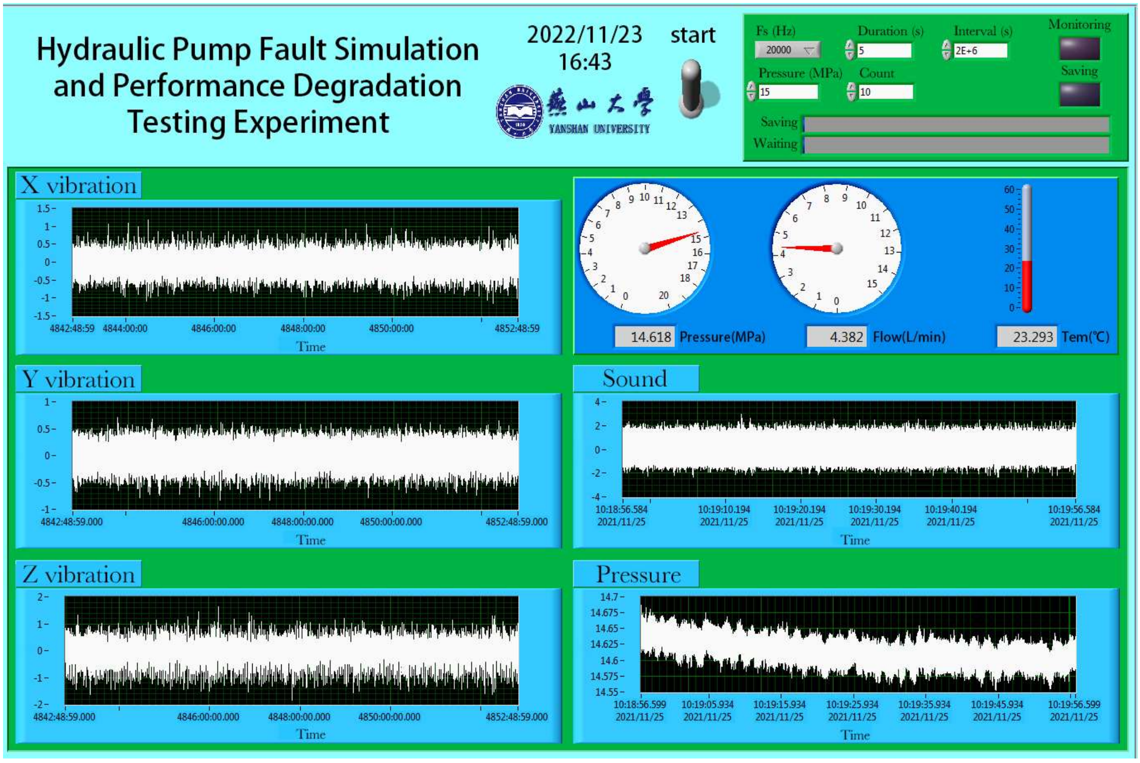


Figure 8. Photograph of test bench. The main components are indicated by black dotted line boxes in the picture and are accompanied by text explanations.

Table 1. Main components of test bench and their performance indicators.

Name	Model Performance	Parameters
Hydraulic plunger pump (KOMPASS Mechanical and Electrical Co., Ltd., Shanghai, China)	PVS08-B3-F-R	Theoretical displacement: 8 mL/min, pressure regulation range: 3–21 MPa, speed range: 500–2000 r/min, number of plungers: 7,
Electric motor (Chyun Tseh Motor Industrial Co., Ltd., Taiwan, China)	C07-43BO	Rated power: 5.5 kW, rated speed: 1440 r/min
Flowmeter (Katu Electronics Co. Ltd., Suzhou, China)	FM120-010DCA3KQ	Measuring range: 0.2–1.2 m^3/h
Temperature sensor (Katu Electronics Co. Ltd., Suzhou, China)	TS300-A3R14MM005D6P	Measuring range: 0–150 °C
Pressure sensor (Katu Electronics Co. Ltd., Suzhou, China)	PS300-B250G1/2MA3P	Measuring range: 0–250 bar
Sound level meter (Aihua Instruments Co. Ltd., Zhejiang, China)	AWA5661	Measuring range: 35–130 dB, sensitivity: 40 mV/Pa, frequency range: 10 Hz–16 kHz
Analog signal acquisition card (National Instruments Corp. Ltd., Austin, TX, USA)	NI PXIe-6363	Sampling rate: 2 MS/s, number of channels: 16, resolution: 16 bit
Data Acquisition Controller (National Instruments Corp. Ltd., Austin, TX, USA)	NI PXIe-8135	Processor: 2.3 GHz Quad Core Intel Core i7-3610QE, Memory: 4 GB

The signals obtained using the various sensors are transmitted to the data acquisition card, and the data are displayed and stored in real time using the developed LabVIEW program. The front panel of the LabVIEW program is presented in Figure 9.



Hydraulic Pump Fault Simulation and Performance Degradation Testing Experiment

2022/11/23
16:43

Fs (Hz): 20000
Duration (s): 5
Interval (s): 2E+6
Pressure (MPa): 15
Count: 10

14.618 Pressure(MPa)
4.382 Flow(L/min)
23.293 Tem(℃)

Figure 9. Front panel of LabVIEW data acquisition program. On the left side of the panel are three channel vibration signal monitoring panels, and in the upper right corner is the interface for setting acquisition parameters. Below that are visual displays of system pressure, flow, and temperature, with the real-time monitoring panel for sound and pressure located at the bottom.

During the test, the oil temperature is maintained at 35–40 °C by controlling the air-cooled unit to reduce the influence of changes in oil viscosity on the test. The tests are conducted at different pressures (5, 10, and 15 MPa) and flow rates (3, 6, and 9 L/min). Three different sampling frequencies (20, 30, and 40 kHz) are used to continuously sample the pump 10 times under each state, with each sampling duration being 5 s. Six pumps with the same model are individually installed on the bench, and seven faulty components are replaced to collect eight types of state data for the six pumps.

3.2. Test Dataset

The data-collection conditions for the sound signal dataset used in this study are as follows: A sampling frequency of 40 kHz, system pressure of 10 MPa, and flow rate of 9 L/min. The dataset includes data from six pumps, each with eight state signals (normal N, wear of shoe WS, wear of port plate WPP, loose shoe LS, wear of plunger WP, fault of bearing outer ring FOR, fault of bearing inner ring FIR, and fault of bearing rolling element FRE). Figure 10 presents the corresponding time-domain and frequency-domain graphs of the pump under these eight states.

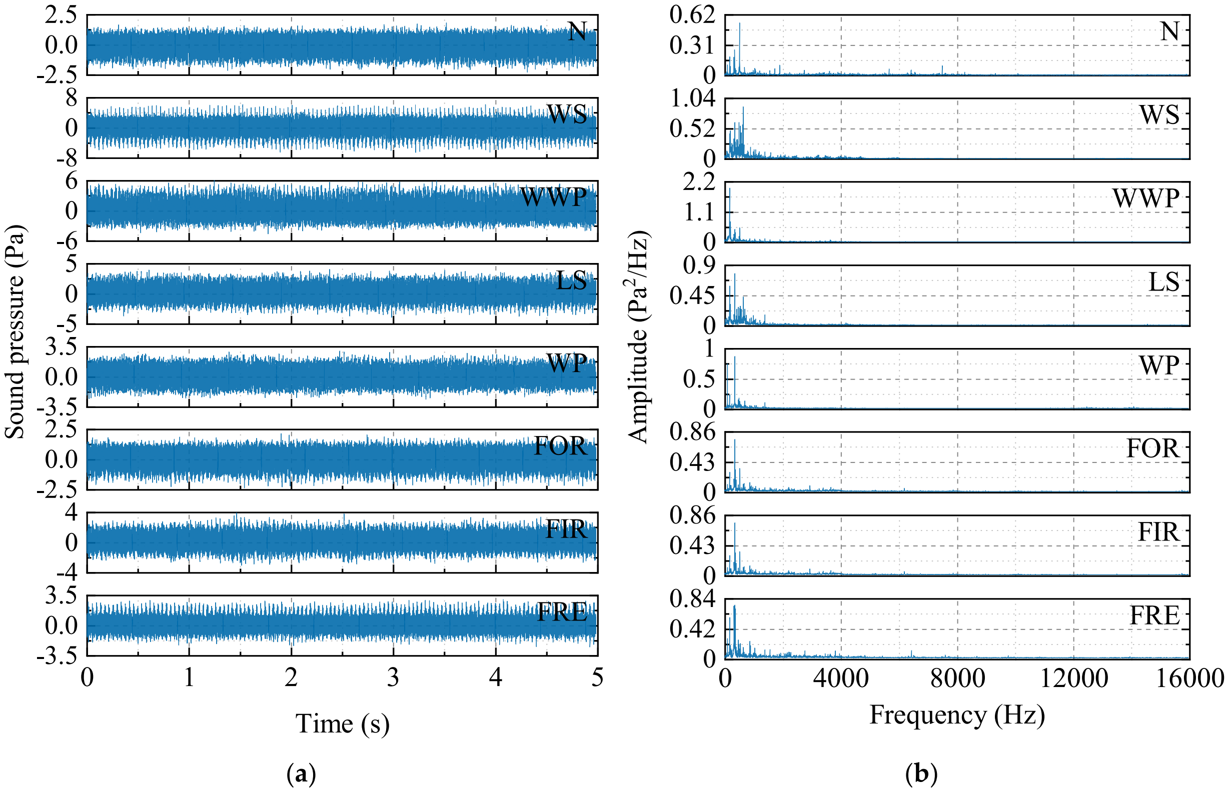


Figure 10. The sound signals of a plunger pump in normal and seven faulty states. (**a**) Time domain; (**b**) frequency domain. Due to the limitations of the sound level meter, the effective frequency range is 0 Hz–16 kHz.

The bearings in the hydraulic pump are deep groove ball bearings, with model number 6205. The pitch diameter is 39.04 mm, and there are 9 rolling elements with a diameter of 7.94 mm. The rated speed of the driving motor is 1440 r/min, and the actual measured speed is 1488 r/min, which is equivalent to 24.8 Hz. The characteristic frequency coefficients for the three types of bearing faults are 5.4152 (FIR), 3.5848 (FOR), and 2.3567 (FRE). The corresponding characteristic frequencies are 134.3 Hz, 88.9 Hz, and 58.4 Hz. The hydraulic pump has a total of seven plungers, which results in a pressure shock at 173.6 Hz. We enlarged the frequency domain plot to display the range of 0–700 Hz, as shown in Figure 11.

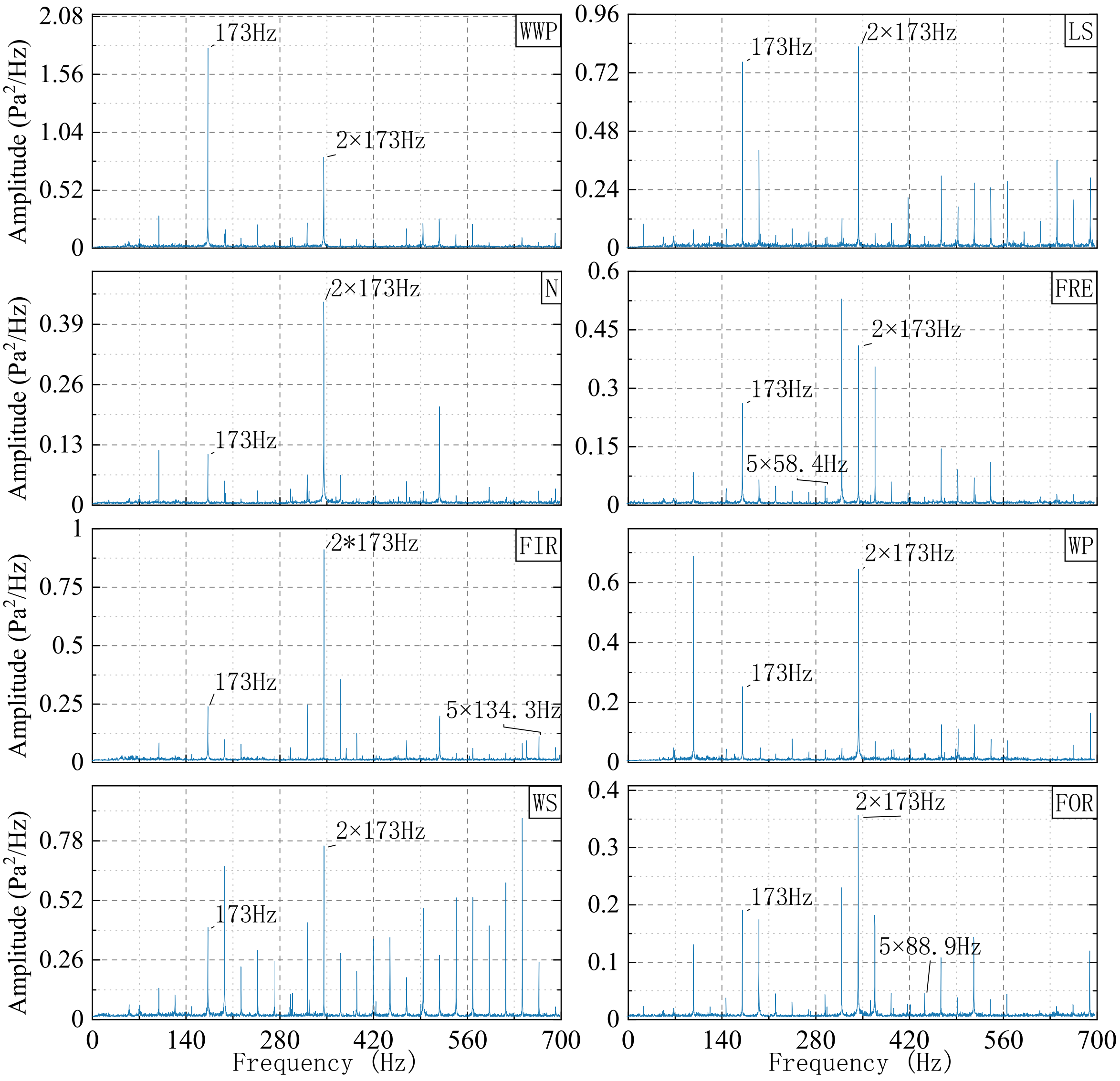


Figure 11. Amplified plot of the frequency domain signal.

In Figure 11, peak values at the reference frequency of the pressure shock and its harmonics can be clearly observed. The fifth harmonic of the fault frequency of three types of bearing faults can be observed.

The most commonly used method for measuring sound signals is the sound pressure level (dB). The measurement for the signals obtained in this test is that of sound pressure in Pa. The conversion formula between the two is:

$$SPL = 20\lg(\frac{P}{P_{\text{ref}}}), \tag{30}$$

where *SPL* represents the sound pressure level, *P* is the sound pressure, and P_{ref} is the reference sound pressure. As the minimum sound pressure perceived by the human ear is 20 μPa, the reference sound pressure in air is generally considered 20 μPa. As shown in Figure 10, the sound pressure amplitude of the time domain in the normal state is less than that in the other states, and there are slight differences in the frequency domain graphs corresponding to different states; however, it is difficult to recognize their corresponding state categories.

In transfer learning, the dataset is divided into four parts: Training, validation, support, and query sets. The training and validation sets are obtained from the source domain, and the samples are labeled; the support set comprises the training samples in the target domain, and the samples are labeled; and the query set comprises the test samples in the target domain, and the samples are unlabeled. Each data file collected through the test contains 5 s of sound signals, with 200 k data points, and each file is divided into 47 samples through overlapping partitioning. The dataset divisions used in this study are listed in Table 2.

Table 2. Dataset Partitioning.

Data Set	Data Details	Files	Samples
Training set	Pump_1, normal + 7 types of faults	80	3760
	Pump_2, normal + 7 types of faults	80	3760
	Pump_3, normal + 7 types of faults	80	3760
	Pump_4 normal	10	470
Verification set	Pump_4, 7 types of faults	70	3290
Support set	Pump_5, normal	10	470
	Pump_6, normal	10	470
Query set	Pump_5, 7 types of faults	70	3290
	Pump_6, 7 types of faults	70	3290

The data obtained from Pump_1–Pump_4 are used as the source domain data, while the data obtained from Pump_5 and Pump_6 are used as the target domain data. Considering that in practical engineering applications, normal data are usually abundant while failure data are scarce, all the normal samples in the target domain are selected as the support set.

3.3. Fault Diagnosis Calculation Process

3.3.1. CEEMDAN Noise Reduction

Before denoising, the original data are zero-centered. CEEMDAN is then used to decompose the sound signal into multiple modal components and selected components for reconstruction using the mRMR algorithm to complete the denoising process. For example, we demonstrated the denoising process of a sound signal in the normal state of Pump_1. The other signals are processed similarly. Figure 12 presents a comparison of the modal components after the EEMD and CEEMDAN are performed.

The EEMD produced 17 IMF components and 1 residual, while the CEEMDAN produced 13 IMF components and 1 residual. Compared with EEMD, there are fewer signal components after the CEEMDAN, which indicates that CEEMDAN can effectively avoid the interference of white noise that is added multiple times by the EEMD to the original signal.

In the mRMR algorithm, the mutual information $I(\widetilde{c}_j;x)$ between each component $\widetilde{c}_j(s)$ and the original signal $x(t)$ is first calculated:

$$I(\widetilde{\boldsymbol{c}}_j;\boldsymbol{x}) = \sum_{t=0}^{T-1}\sum_{s=0}^{S-1} p[\widetilde{c}_j(s),x(t)]log_2\frac{p[\widetilde{c}_j(s),x(t)]}{p[\widetilde{c}_j(s)]p[x(t)]}, \tag{31}$$

where $p[\widetilde{c}_j(s),x(t)]$ represents the probability of their joint occurrence; $p[\widetilde{c}_j(s)]$ and $p[x(t)]$ represent their individual occurrence probabilities; and S and T are equal, both of which are the number of signal points.

The higher the mutual information, the higher the degree of dependence between the two random variables. The features are then sorted according to their mutual information values. Next, the first feature is selected as the most relevant feature from a sorted list of features. Subsequently, the mutual information of the remaining features with the selected feature is calculated, and their correlation with the selected feature is computed. The scores are then totaled to obtain a comprehensive score $mRMR_j$:

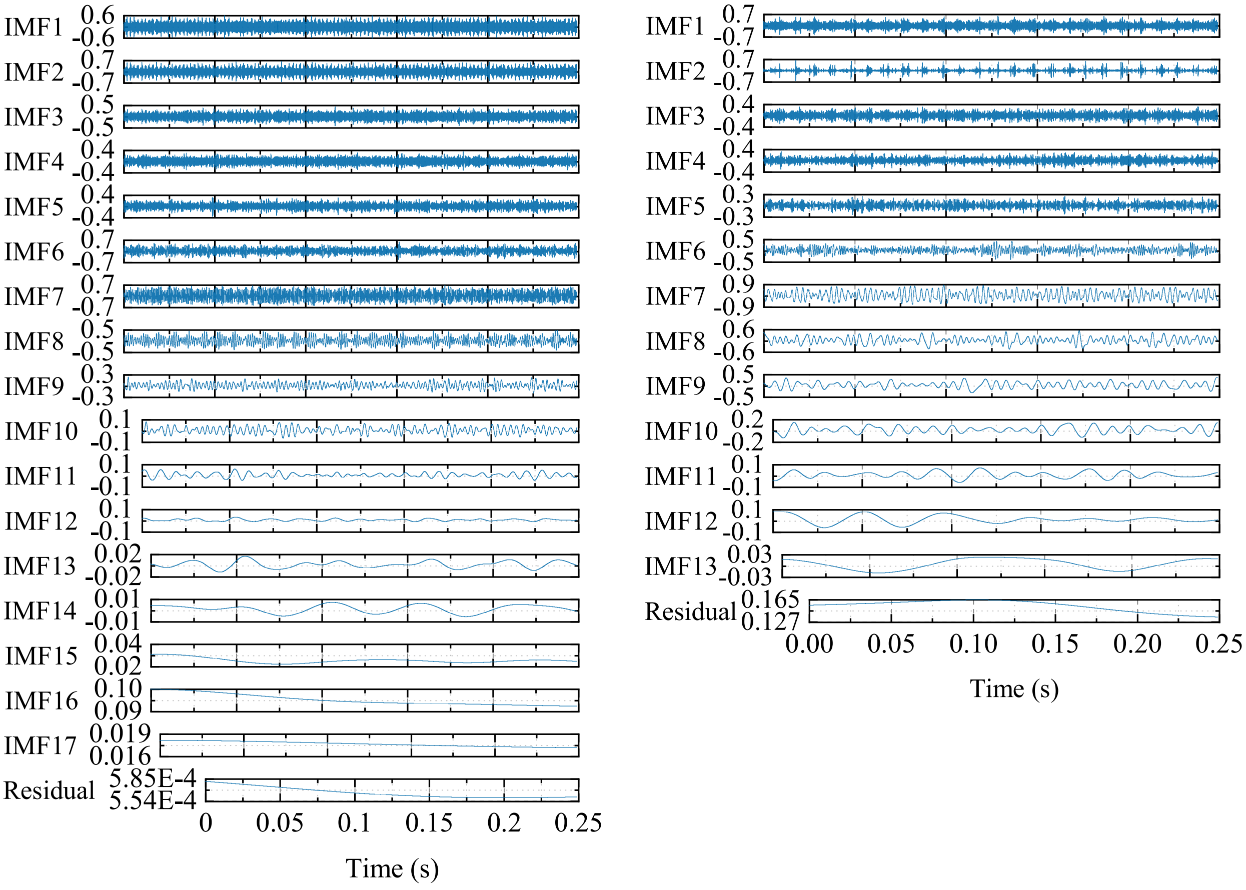
$$mRMR_j = I(\widetilde{\boldsymbol{c}}_j; \boldsymbol{x}) - \frac{1}{k}\sum_{i=1}^{k} I(\widetilde{\boldsymbol{c}}_j; \widetilde{\boldsymbol{c}}_i), \tag{32}$$

where k denotes the number of selected features. Based on this, the features are sorted and the feature with the highest score is selected as the next feature. This process is repeated until sufficient features are selected or all the features are selected [29]. After sorting all the IMFs using the mRMR algorithm, the correlation coefficients between the signals are reconstructed with different numbers of selected IMFs and the original signal. In this study, the Pearson correlation coefficient is used and is calculated as follows:

$$CC(\boldsymbol{x}, \boldsymbol{y}) = \frac{cov(\boldsymbol{x}, \boldsymbol{y})}{\sqrt{var(\boldsymbol{x})var(\boldsymbol{y})}}, \tag{33}$$

where $cov(\boldsymbol{x}, \boldsymbol{y})$ represents the covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$, and $var(\boldsymbol{x})$ represents the variance of $\boldsymbol{x}$.

Subsequently, the calculated correlation coefficient is used to plot the curve, as shown in Figure 13.



(**a**) (**b**)

Figure 12. Normal sound signal of Pump_1 decomposed via EEMD (**a**) and CEEMDAN (**b**).

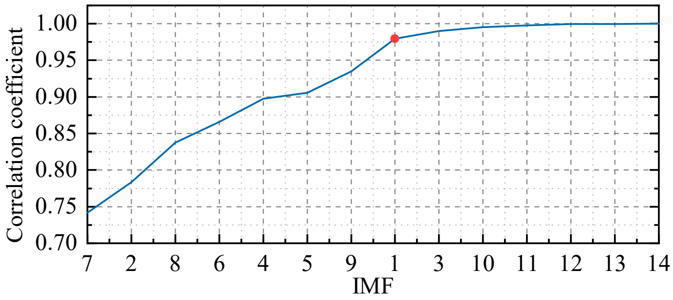


Figure 13. Correlation coefficients between the reconstructed signal with added IMFs and the original signal. The signal comes from the normal state of Pump_1.

In Figure 13, the horizontal axis represents the ordering number of the IMFs obtained using the mRMR algorithm. The closer to the left end, the higher the mRMR comprehensive score of the component, with 14 indicating a residual. To elaborate, the score of IMF7 is the highest, with a correlation coefficient of 0.74 with the original signal; the score of IMF2 ranks second, and the signal reconstructed using IMF2 and IMF7 has a correlation coefficient of 0.78 with the original signal. Thus, the signal reconstructed using all the components and residual has a correlation coefficient of 1 with the original signal. The slope of the curve is relatively steep in the first half of the curve, and the curve growth slows after the addition of IMF1, at the location of the red dot marker. Therefore, the first eight IMFs (7/2/8/6/4/5/9/1) are selected to reconstruct the signal. The frequency-domain graphs of the reconstructed and original signals are presented in Figure 14.

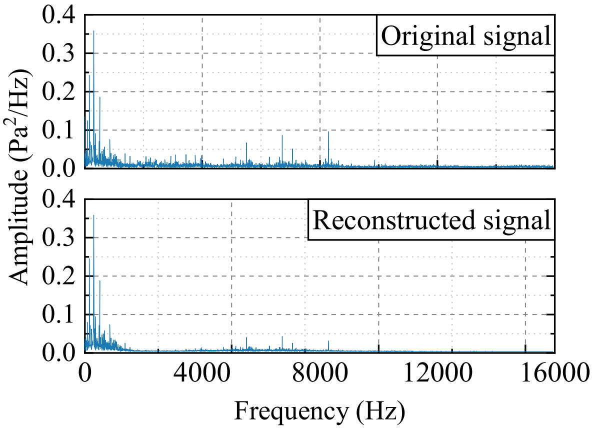


Figure 14. Frequency-domain diagram of original and reconstructed signals. The signal comes from the normal state of Pump_1.

From Figure 14, it can be observed that, compared with the original signal, the reconstructed signal retains the data in the 0–2000 Hz frequency band, and the two frequency bands with high amplitudes of 5000–9000 Hz have some attenuation, but their peaks are preserved. The other frequency bands are suppressed to lower levels.

3.3.2. Mean Spectrogram Bar Graphs Generation

The reconstructed signal is linearly normalized, and the data are mapped to [0, 1], which is expressed as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{34}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of all the data, respectively.

The normalized data are plotted as a spectrogram. It is worth noting that converting a spectrogram generated by a false color to a black-and-white image will result in the loss of a considerable amount of information and produce the same file size as a grayscale spectrogram. In this study, grayscale images are directly generated using raw data, which can compress the file size, reduce the network computing power, and fully record the device state information. For example, a comparison of the three visualizations using the normal signal of Pump_1 is presented in Figure 15.

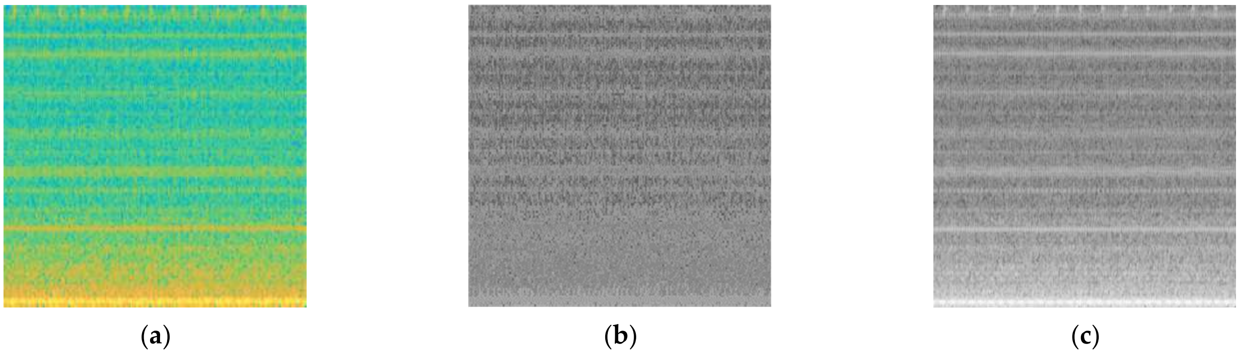


Figure 15. Three spectrograms drawn using normal state data of Pump_1. (**a**) Pseudocolor spectrogram; (**b**) pseudocolor converted to black and white; (**c**) grayscale spectrogram.

In the field of deep learning, color images can provide more feature information than grayscale images and are therefore widely used. However, this requires a premise: The RGB three-color channels of a color image contain different feature information. The spectrogram drawn using pseudocolors uses a three-dimensional tensor to express a two-dimensional matrix, which means that the feature information contained in the pseudocolor and grayscale spectrograms is identical. Using color images as the input to the network will inevitably result in a multiplication of the computational load owing to the increase in the number of channels. Therefore, in this study, grayscale images are used instead of pseudocolor images as network inputs. Figure 16 presents the grayscale spectrograms corresponding to the data of Pump_1 in the eight different states considered.

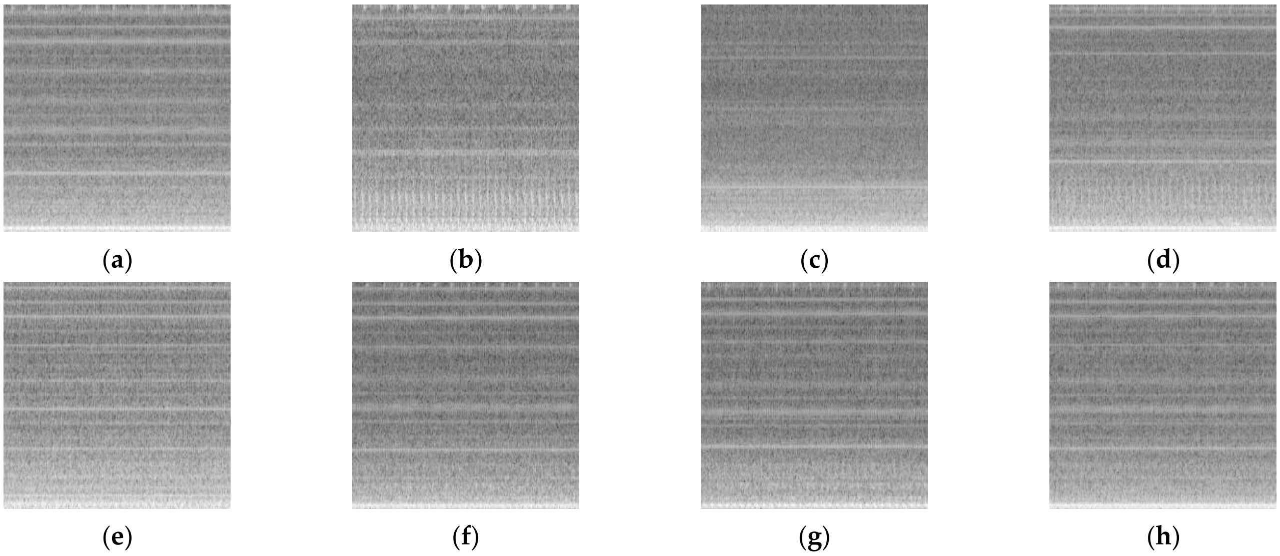


Figure 16. Grayscale spectrogram of Pump_1 corresponding to eight different states. (**a**) N; (**b**) WS; (**c**) WWP; (**d**) LS; (**e**) WP; (**f**) FOR; (**g**) FIR; (**h**) FRE.

From Figure 16, it can be observed that the grayscale spectrogram has more obvious horizontal stripes. This indicates that the frequency domain has a better feature representation ability for the pump-state data. Therefore, in this study, grayscale spectrograms are compressed into mean frequency spectrum bar graphs as inputs to the network. Figure 17 presents the mean spectrogram bar graphs corresponding to the eight states of Pump_1. To better observe the compressed images, they are stretched horizontally.

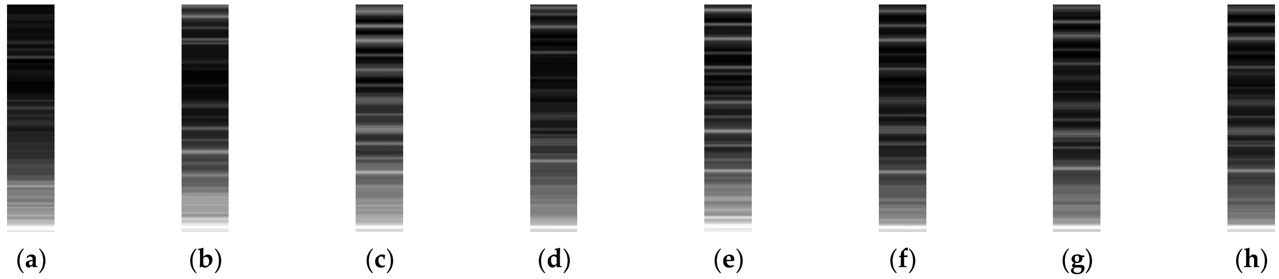


Figure 17. Mean spectrogram bar graph for eight states of Pump_1. (**a**) N; (**b**) WS; (**c**) WWP; (**d**) LS; (**e**) WP; (**f**) FOR; (**g**) FIR; (**h**) FRE. The image has undergone stretching processing.

From Figure 17, it can be observed that the mean spectrogram bar graphs corresponding to each state are significantly different from those of the other states. The mean spectrum bar graphs of the three types of bearing faults are relatively similar; however, some differences can still be observed. Compared with the grayscale spectrogram, in addition to compressing the file size, the mean spectrogram bar graph has a higher contrast and more obvious differences between the images of each state. These significant differences can reduce the difficulty in learning the feature information and improve the accuracy of the diagnostic model.

3.3.3. Use of Deep Transfer Learning for Fault Location

Previously, the dataset used in many studies on fault-diagnosis equipment, such as bearings and hydraulic plunger pumps, is obtained using a single device. Therefore, the similarity between the datasets was extremely high and the diagnostic accuracy was often close to 100%. This study demonstrates this in the same manner by dividing the Pump _1 data into training, validation, and test sets in a 5:2:3 ratio; the corresponding sample sizes are divided in the ratio of 1880:752:1128. Via overlapping segmentation of the original data, 47 mean spectrogram bar graphs are generated for each data file. The mean spectrogram bar graphs generated from the training and validation sets are imported into the ResNet-50 model for training. The training process parameters are listed in Table 3.

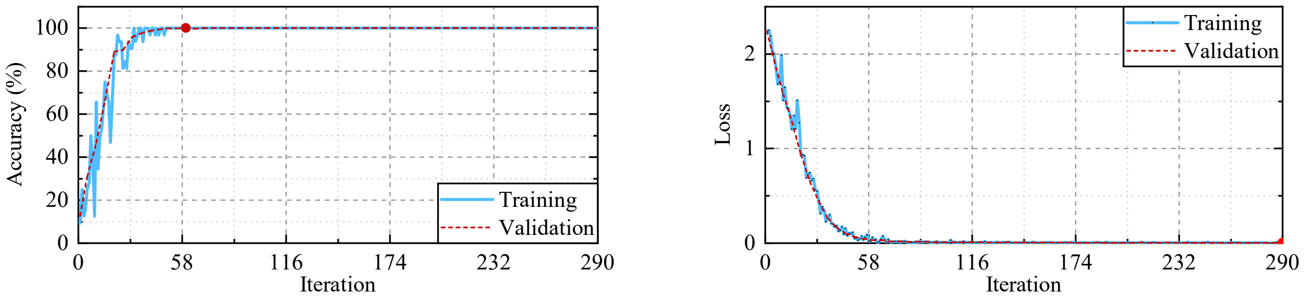
Table 3. ResNet-50 Model Training Parameters.

Parameter	Parameter Value
Initial learn rate	0.001
Validation frequency	5
Max epochs	5
Mini batch size	32
L2 regularization	0.0001
Learn rate drop factor	0.1
Learn rate drop period	10

The relationship between the number of training epochs n_{epochs} and batch size s_{batch} is as follows:

$$n_{\text{epochs}} = \frac{t_{\text{iteration}} \cdot s_{\text{batch}}}{n_{\text{samples}}}, \tag{35}$$

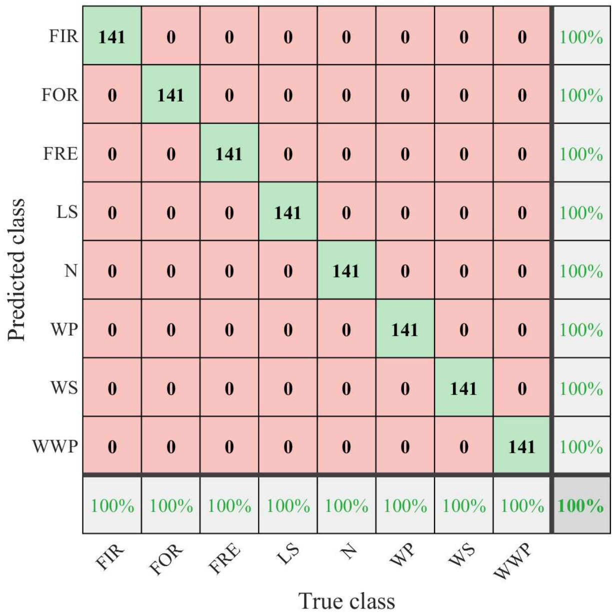
where $t_{\text{iteration}}$ represents the number of training iterations, and n_{samples} represents the total number of training samples. Therefore, according to the parameter settings, 32 samples are involved in the training per iteration; after 58 iterations, all the samples (1880) participated in one round of model training, and after 290 iterations, five rounds of training are completed. The training process curve is presented in Figure 18.



(**a**) (**b**)

Figure 18. Process curves for training the ResNet-50 model using Pump_1 data. (**a**) Accuracy curves; (**b**) Loss curves.

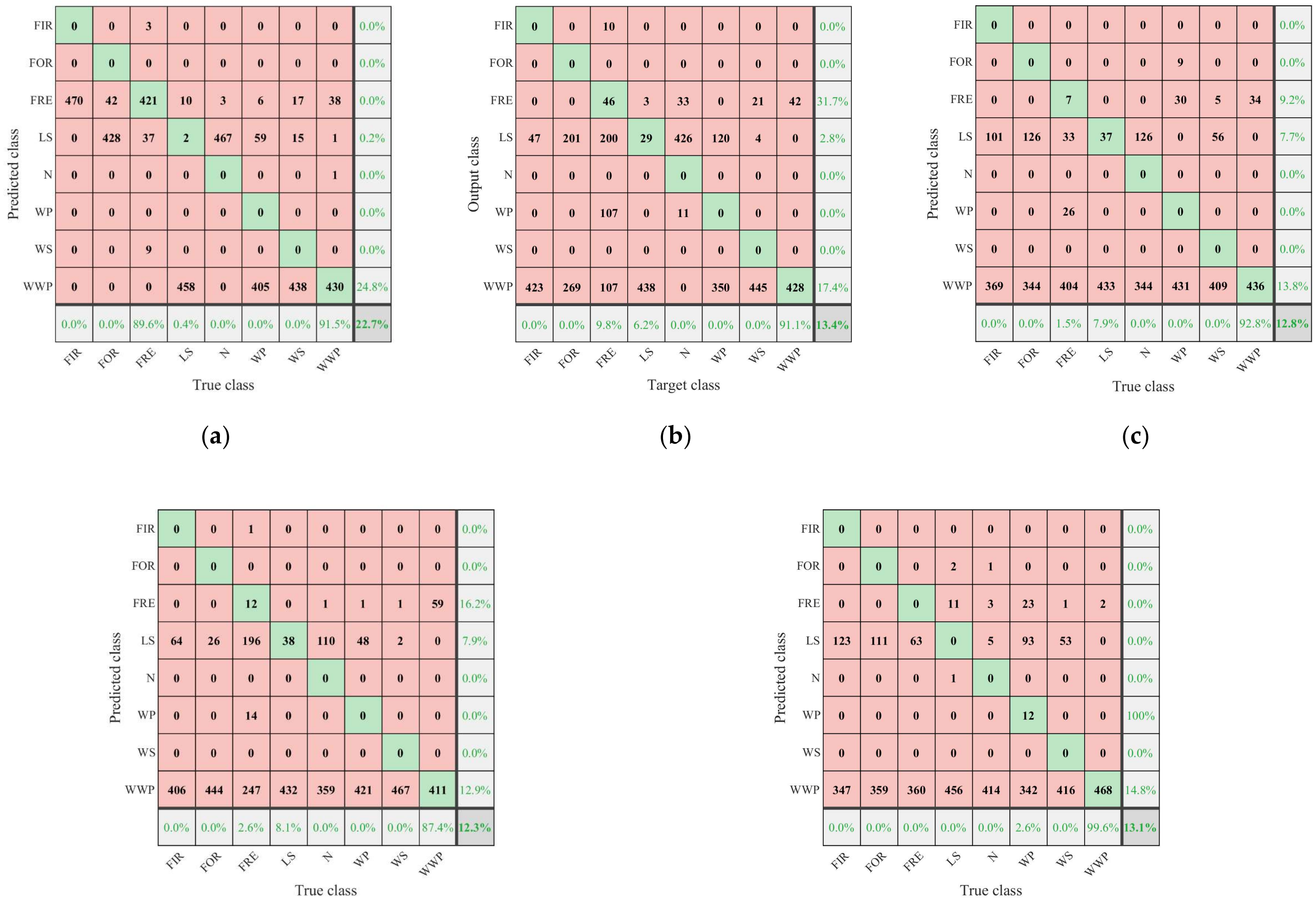
The curves show that the ResNet-50 model requires only one round of training, and the validation accuracy can reach 100% after 60 iterations, at the location of the red dot marker. The validation accuracy remains stable at 100% after 70 iterations. The validation loss is 0.032 after 60 iterations and decreases to 0.0026 after 290 iterations. The test set data are input into the trained model for testing, and the confusion matrix is plotted in Figure 19.


Predicted class
True class

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	141	0	0	0	0	0	0	0	100%
FOR	0	141	0	0	0	0	0	0	100%
FRE	0	0	141	0	0	0	0	0	100%
LS	0	0	0	141	0	0	0	0	100%
N	0	0	0	0	141	0	0	0	100%
WP	0	0	0	0	0	141	0	0	100%
WS	0	0	0	0	0	0	141	0	100%
WWP	0	0	0	0	0	0	0	141	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%

Figure 19. Confusion matrix of diagnostic results of ResNet-50 model when both training and test sets are from Pump_1 data.

The numbers in the confusion matrix represent the number of samples, with the last row indicating the recall rate for each true class and the last column indicating the precision rate for each predicted class. The bottom-right corner presents the testing accuracy of the proposed model. It can be observed that the diagnostic model's accuracy reached 100% when tested with samples obtained from Pump_1, but the accuracy decreased to an unacceptable level when tested with other pump samples. The testing accuracy of the model for the samples obtained from Pump_2–Pump_6 is 22.7%, 13.4%, 12.8%, 12.3% and 13.1%, respectively. The confusion matrices shown in Figure 20 illustrate the performance of the model when tested using samples from different pumps.



Predicted class; True class

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	0	0	3	0	0	0	0	0	0.0%
FOR	0	0	0	0	0	0	0	0	0.0%
FRE	470	42	421	10	3	6	17	38	0.0%
LS	0	428	37	2	467	59	15	1	0.2%
N	0	0	0	0	0	0	0	1	0.0%
WP	0	0	0	0	0	0	0	0	0.0%
WS	0	0	9	0	0	0	0	0	0.0%
WWP	0	0	0	458	0	405	438	430	24.8%
	0.0%	0.0%	89.6%	0.4%	0.0%	0.0%	0.0%	91.5%	22.7%

(**a**)

Output class; Target class

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	0	0	10	0	0	0	0	0	0.0%
FOR	0	0	0	0	0	0	0	0	0.0%
FRE	0	0	46	3	33	0	21	42	31.7%
LS	47	201	200	29	426	120	4	0	2.8%
N	0	0	0	0	0	0	0	0	0.0%
WP	0	0	107	0	11	0	0	0	0.0%
WS	0	0	0	0	0	0	0	0	0.0%
WWP	423	269	107	438	0	350	445	428	17.4%
	0.0%	0.0%	9.8%	6.2%	0.0%	0.0%	0.0%	91.1%	13.4%

(**b**)

Predicted class; True class

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	0	0	0	0	0	0	0	0	0.0%
FOR	0	0	0	0	0	9	0	0	0.0%
FRE	0	0	7	0	0	30	5	34	9.2%
LS	101	126	33	37	126	0	56	0	7.7%
N	0	0	0	0	0	0	0	0	0.0%
WP	0	0	26	0	0	0	0	0	0.0%
WS	0	0	0	0	0	0	0	0	0.0%
WWP	369	344	404	433	344	431	409	436	13.8%
	0.0%	0.0%	1.5%	7.9%	0.0%	0.0%	0.0%	92.8%	12.8%

(**c**)

Predicted class; True class

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	0	0	1	0	0	0	0	0	0.0%
FOR	0	0	0	0	0	0	0	0	0.0%
FRE	0	0	12	0	1	1	1	59	16.2%
LS	64	26	196	38	110	48	2	0	7.9%
N	0	0	0	0	0	0	0	0	0.0%
WP	0	0	14	0	0	0	0	0	0.0%
WS	0	0	0	0	0	0	0	0	0.0%
WWP	406	444	247	432	359	421	467	411	12.9%
	0.0%	0.0%	2.6%	8.1%	0.0%	0.0%	0.0%	87.4%	12.3%

(**d**)

Predicted class; True class

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	0	0	0	0	0	0	0	0	0.0%
FOR	0	0	0	2	1	0	0	0	0.0%
FRE	0	0	0	11	3	23	1	2	0.0%
LS	123	111	63	0	5	93	53	0	0.0%
N	0	0	0	1	0	0	0	0	0.0%
WP	0	0	0	0	0	12	0	0	100%
WS	0	0	0	0	0	0	0	0	0.0%
WWP	347	359	360	456	414	342	416	468	14.8%
	0.0%	0.0%	0.0%	0.0%	0.0%	2.6%	0.0%	99.6%	13.1%

(**e**)

Figure 20. Confusion matrix of diagnostic results of ResNet-50 model when the training set is from Pump_1 data and the test set is from the data of the other five pumps. (**a**) Pump_2; (**b**) Pump_3; (**c**) Pump_4; (**d**). Pump_5; (**e**) Pump_6.

For practical applications, a well-trained diagnostic model should be able to diagnose the majority of pumps in the same model and have a satisfactory diagnostic effect. Therefore, it is necessary to conduct research on transfer learning to increase the generalization of the diagnostic model.

The ResNet-50 diagnostic model is trained on a dataset that is divided as shown in Table 2. The parameters used in the first training process are the same as those in Table 3, but the maximum number of epochs is adjusted to 40 to increase the training time of the model. The model training process curves are presented in Figure 21.

Although the training accuracy quickly reached 100%, it can be observed that the validation accuracy can only reach a maximum of 65.1%, at the location of the red dot marker and the minimum validation loss is 1.32, at the location of the red dot marker. Because the selected dataset for the validation set comprises data on the seven types of faults of Pump_4, which is quite different from the data of Pump_1–Pump_3 used for training. Therefore, Bayesian optimization is used to optimize the initial learning rate of the model between [0.00001, 0.1], a logarithmic scale is used for the horizontal axis, and 30 attempts are made. The loss and accuracy corresponding to different initial learning rates during the optimization process are presented in Figure 22.

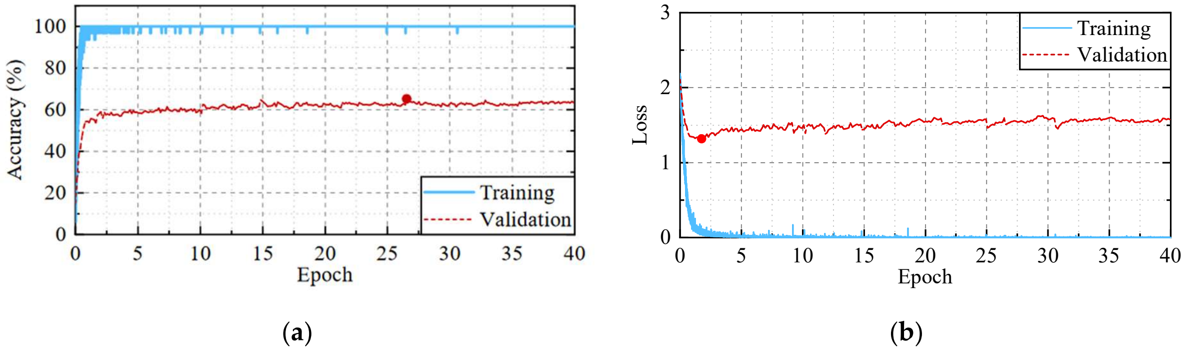


Figure 21. Process curve of ResNet-50 model training with samples obtained from Pump_1–Pump_3. (**a**) Accuracy curves; (**b**) loss curves.

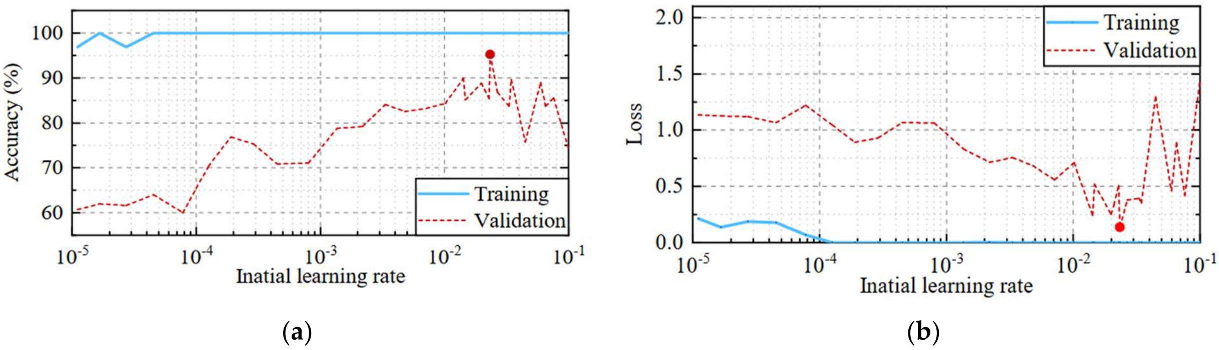


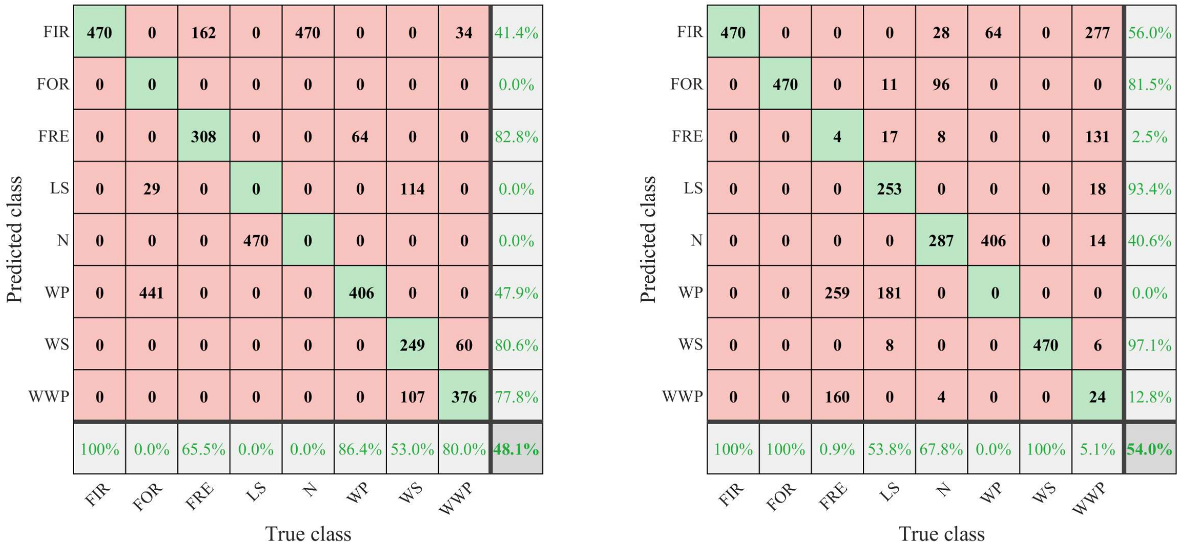
Figure 22. Curves of loss and accuracy corresponding to different initial learning rates. (**a**) Accuracy curves; (**b**) loss curves.

As shown in Figure 22, in the 30 attempts, the validation accuracy realized is the highest at 95.2%, at the location of the red dot marker and the validation loss is 0.13, at the location of the red dot marker when the initial learning rate is set as 0.023. The model trained at this learning rate, which had the minimum validation loss, is exported as the base model for the transfer learning. The confusion matrices of the diagnostic results obtained for the data of Pump _5 and Pump _6 using the base model are presented in Figure 23.

As can be observed from Figure 23, owing to the increase in the training data and parameter optimization, the accuracy of the base diagnostic model reached 48.1% and 54% for the Pump_5 and Pump_6 samples, respectively. The accuracy increased by 35.8% and 40.9%, respectively, compared to the model trained with the Pump _1 data. This indicates that expanding the training data can improve the generalization ability of the diagnostic model.

Subsequently, a model-based transfer learning model training was performed. With the exception of the fully connected and softmax layers, the parameters of the remaining layers of the base model are frozen, and the frozen model is trained using the support set. In this study, the frozen model is referred to as the baseline model. The initial learning rate is consistent with the parameters obtained from the optimization of the base model. The last 30% of the support set is divided into the validation sets for the training process. The training process is illustrated in Figure 24.

During the training process, the highest validation accuracy is 99.5%, at the location of the red dot marker and the minimum validation loss is 0.28, at the location of the red dot marker. The model with the minimum validation loss is selected as the final diagnostic model and is used to diagnose the Pump_5 and Pump_6 data. The confusion matrices for the diagnostic results are presented in Figure 25.



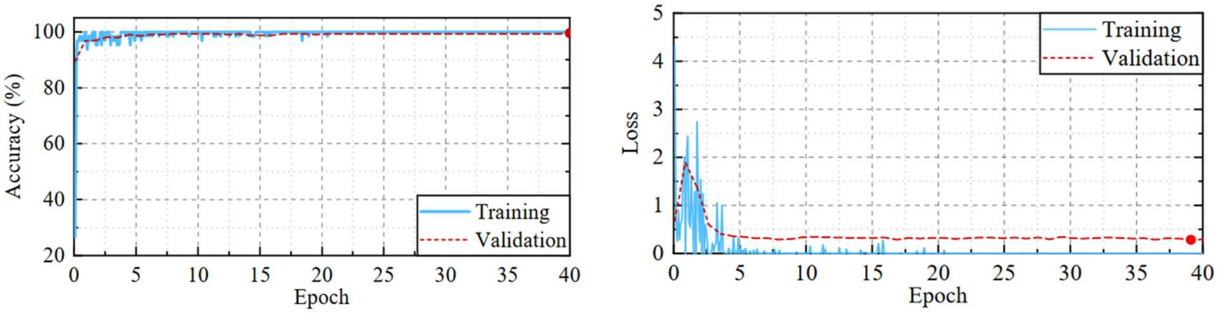
	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	470	0	162	0	470	0	0	34	41.4%
FOR	0	0	0	0	0	0	0	0	0.0%
FRE	0	0	308	0	0	64	0	0	82.8%
LS	0	29	0	0	0	0	114	0	0.0%
N	0	0	0	470	0	0	0	0	0.0%
WP	0	441	0	0	0	406	0	0	47.9%
WS	0	0	0	0	0	0	249	60	80.6%
WWP	0	0	0	0	0	0	107	376	77.8%
	100%	0.0%	65.5%	0.0%	0.0%	86.4%	53.0%	80.0%	**48.1%**

(**a**)

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	470	0	0	0	28	64	0	277	56.0%
FOR	0	470	0	11	96	0	0	0	81.5%
FRE	0	0	4	17	8	0	0	131	2.5%
LS	0	0	0	253	0	0	0	18	93.4%
N	0	0	0	0	287	406	0	14	40.6%
WP	0	0	259	181	0	0	0	0	0.0%
WS	0	0	0	8	0	0	470	6	97.1%
WWP	0	0	160	0	4	0	0	24	12.8%
	100%	100%	0.9%	53.8%	67.8%	0.0%	100%	5.1%	**54.0%**

(**b**)

Figure 23. Confusion matrix of diagnostic model obtained by enlarging the training dataset without transfer. (**a**) Pump_5; (**b**) Pump_6. Pump_5 and Pump_6 data were not involved in the training.



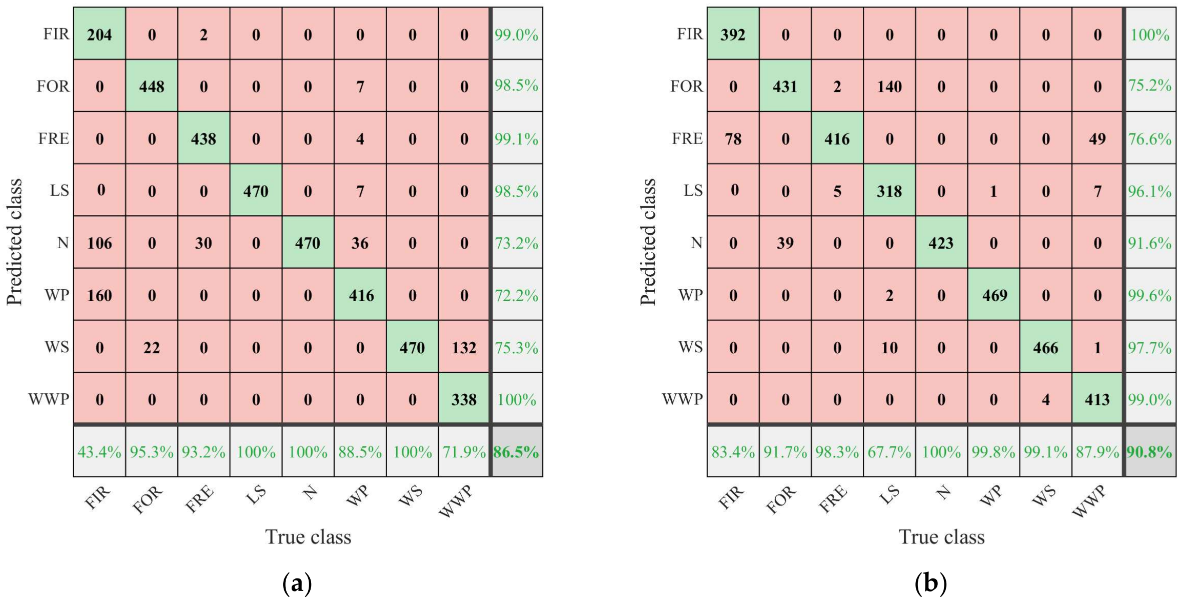
(**a**) (**b**)

Figure 24. Process curve for training of the baseline model with normal data obtained from Pump_5 and Pump_6. (**a**) Accuracy curves; (**b**) loss curves.

It can be observed that the transferred model can also provide better diagnosis results for the new data. The diagnostic accuracies of Pump_5 and Pump_6 are 86.5% and 90.8%, respectively. The diagnostic results are 38.4% and 36.8% higher than those of the baseline model, respectively. The diagnostic accuracies of the different models used in this study are listed in Table 4.

Table 4. Diagnostic accuracy of the model at different stages.

Diagnostic Object	Diagnostic Accuracy of Pump_1 Data Training Model	Baseline Model Diagnostic Accuracy	Transferred Model Diagnostic Accuracy
Pump_5	12.3%	48.1%	86.1%
Pump_6	13.1%	58%	90.8%



	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	204	0	2	0	0	0	0	0	99.0%
FOR	0	448	0	0	0	7	0	0	98.5%
FRE	0	0	438	0	0	4	0	0	99.1%
LS	0	0	0	470	0	7	0	0	98.5%
N	106	0	30	0	470	36	0	0	73.2%
WP	160	0	0	0	0	416	0	0	72.2%
WS	0	22	0	0	0	0	470	132	75.3%
WWP	0	0	0	0	0	0	0	338	100%
	43.4%	95.3%	93.2%	100%	100%	88.5%	100%	71.9%	86.5%

(**a**)

	FIR	FOR	FRE	LS	N	WP	WS	WWP	
FIR	392	0	0	0	0	0	0	0	100%
FOR	0	431	2	140	0	0	0	0	75.2%
FRE	78	0	416	0	0	0	0	49	76.6%
LS	0	0	5	318	0	1	0	7	96.1%
N	0	39	0	0	423	0	0	0	91.6%
WP	0	0	0	2	0	469	0	0	99.6%
WS	0	0	0	10	0	0	466	1	97.7%
WWP	0	0	0	0	0	0	4	413	99.0%
	83.4%	91.7%	98.3%	67.7%	100%	99.8%	99.1%	87.9%	90.8%

(**b**)

Figure 25. Confusion matrixes of diagnostic results by the transferred model. (**a**) Pump_5; (**b**) Pump_6.

In this paper, a transfer learning study based on SqueezeNet and GoogleLeNet is also conducted. Using the same dataset mentioned previously, the accuracy of the fault diagnosis with different transfer models is presented in Table 5.

Table 5. Diagnostic accuracy of different transfer models.

Diagnostic Object	SqueezeNet	GoogleLeNet	ResNet-50
Pump_5	67.7%	77.5%	86.1%
Pump_6	68.4%	84.1%	90.8%

From Table 5, it can be observed that the hydraulic-pump fault-diagnosis model based on the transfer of ResNet-50 has higher diagnostic accuracy than the other two models.

In addition, to observe whether the model in this study exhibits the overfitting phenomenon, models trained with different numbers of epochs are used to diagnose the data obtained from Pump_6, and the result is presented in Figure 26.

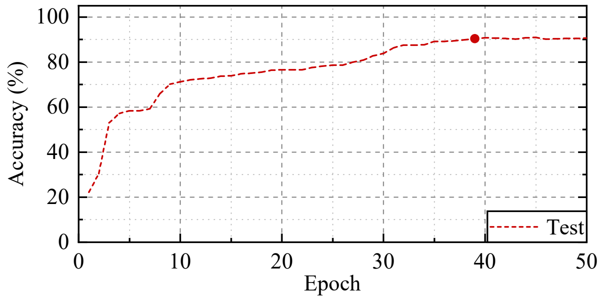


Figure 26. Impact of various training epochs on diagnostic accuracy.

From Figure 26, it can be observed that, with an increase in the number of model epochs, the test accuracy realized using data obtained from Pump_6 increases significantly when the number of training epochs is increased from one to five. Subsequently, the growth becomes slower, and two plateau periods are observed: [4,7] and [9,26]. Finally, when the number of training epochs reaches 39, the test accuracy exceeds 90%, at the location of the red dot marker and remains unchanged thereafter. It can be considered that the model did not exhibit overfitting owing to the increase in the number of training epochs.

In addition, there is one point that needs to be recognized. The sampled signals contain environmental noise, the sound generated by various components of the hydraulic system, and specific frequency sounds caused by faults. Environmental noise can be effectively suppressed through CEEMDAN and mRMR. However, the sound generated by the operation of the hydraulic system cannot be eliminated by noise-reduction algorithms, and the sound signals will also change when the working pressure or spatial arrangement changes. It can be expected that this will have an adverse effect on the accuracy of fault diagnosis. Fortunately, this problem can be solved. By using re-sampled sound signals to conduct transfer learning, the fault diagnosis algorithm can be adapted to the changed hydraulic system.

4. Conclusions

This article first elaborates on the CEEMDAN and mRMR denoising algorithms, the method of drawing mean spectrogram bar graphs based on sound signals, and the transfer learning algorithm based on the ResNet-50 model used in the research process. The test settings and sampled data are then described, and the data calculation process and diagnostic results are displayed. The following conclusions are thus drawn:

1. Compared with the IMF components obtained via EEMD, the CEEMDAN can provide fewer IMF components and can effectively denoise the signal by combining it with the mRMR algorithm.
2. The mean spectrogram bar graph significantly compresses the input diagnostic model images while ensuring the diagnostic accuracy of the diagnostic model.
3. A diagnostic model trained using samples from a single pump cannot be used to diagnose the faults in other pumps. In this study, the diagnostic accuracy of the diagnostic model trained using data from Pump_1 for diagnosing other pumps is found to be less than 22.7%.
4. The baseline diagnostic model trained directly with samples from multiple pumps has a higher diagnostic accuracy when applied with other pumps than the diagnostic model trained with samples obtained from a single pump. In this study, the baseline diagnostic model trained using samples from Pump_1–Pump_3 achieved diagnostic accuracies of 48.1% and 58% for Pump_5 and Pump_6, respectively, which reflect increases of 35.8% and 44.9%, respectively, compared with the diagnostic accuracy of the model trained with samples obtained from a single pump.
5. After the transfer learning, the diagnostic model has a higher diagnostic accuracy than the baseline diagnostic model trained using samples from multiple pumps. The former model can achieve a diagnostic accuracy of 86.1% and 90.8% for Pump_5 and Pump_6, respectively, which is an increase of 38.4% and 36.8%, respectively, compared with the diagnostic accuracy of the latter model.
6. Compared with SqueezeNet and GoogleLeNet, the hydraulic-pump fault-diagnosis model based on the transfer of ResNet-50 has higher diagnostic accuracy. The diagnostic accuracies obtained for the Pump_5 validation data are 67.7%, 77.5%, and 86.1% for the three models, respectively. The accuracies obtained for the Pump_6 data are 68.4%, 84.1%, and 90.8%, respectively.

Research on fault diagnosis based on sound signals has expanded the applications of fault diagnosis methods. In the future, signals such as sound, vibration, and pressure can be fused and used for the fault diagnosis of hydraulic equipment. As inputs to the model, the three channels of the image can be used separately to store different input information. The mean spectrogram bar graph presented in this study can significantly compress the input image file size required for fault diagnosis and can be promoted in appropriate fields. Transfer learning can be used to effectively solve the problem of insufficient data in fault diagnosis and other research fields and has great research value. Ensemble learning, which combines multiple deep-learning models, has been proven to have better generality than a single model [30]. Each model can be specialized for processing specific data subsets during the training phase [31], which can aid in fully leveraging the performances of different models. Therefore, future research should be focused on this direction.

Author Contributions: Methodology, P.Z.; Software, P.Z.; Validation, P.Z.; Formal analysis, P.Z.; Investigation, S.Z. (Sheng Zhang); Resources, W.J. and S.Z. (Sheng Zhang); Data curation, P.Z.; Writing—original draft, P.Z. and Y.Z.; Writing—review & editing, W.J.; Visualization, P.Z.; Supervision, W.J.; Funding acquisition, W.J., S.Z. (Shuqing Zhang) and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China, China (Grant Nos. 52275067, 51875498, and 52275069) and the Natural Science Foundation of Hebei Province, China (Grant No. F2020203058).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data sharing available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, H.; Sun, M.; Deng, W.; Yang, X. A New Feature Extraction Method Based on EEMD and Multi-Scale Fuzzy Entropy for Motor Bearing. *Entropy* **2017**, *19*, 14. [CrossRef]
2. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals. *Sensors* **2017**, *17*, 425. [CrossRef] [PubMed]
3. Zhu, Y.; Tang, S.; Yuan, S. Multiple-signal Defect Identification of Hydraulic Pump Using an Adaptive Normalized Model and S Transform. *Eng. Appl. Artif. Intel.* **2023**, *124*, 106548. [CrossRef]
4. Tang, S.; Zhu, Y.; Yuan, S. An Improved Convolutional Neural Network with an Adaptable Learning Rate towards Multi-signal Fault Diagnosis of Hydraulic Piston Pump. *Adv. Eng. Inform.* **2021**, *50*, 101406. [CrossRef]
5. Jiang, W.; Li, Z.; Lei, Y.; Zhang, S.; Tong, X. Deep learning based rolling bearing fault diagnosis and performance degradation degree recognition method. *J. Yanshan Univ.* **2020**, *44*, 526–536.
6. Qi, Y.; Shen, C.; Wang, D.; Shi, J.; Jiang, X.; Zhu, Z. Stacked Sparse Autoencoder-Based Deep Network for Fault Diagnosis of Rotating Machinery. *IEEE Access* **2017**, *5*, 15066–15079. [CrossRef]
7. Verstraete, D.; Ferrada, A.; Droguett, E.; Meruane, V.; Modarres, M. Deep Learning Enabled Fault Diagnosis Using Time-Frequency Image Analysis of Rolling Element Bearings. *Shock Vib.* **2017**, *2017*, 5067651. [CrossRef]
8. Mao, W.; Liu, Y.; Ding, L.; Li, Y. Imbalanced Fault Diagnosis of Rolling Bearing Based on Generative Adversarial Network: A Comparative Study. *IEEE Access* **2019**, *7*, 9515–9530. [CrossRef]
9. Peng, D.; Liu, Z.; Wang, H.; Qin, Y.; Jia, L. A Novel Deeper One-Dimensional CNN With Residual Learning for Fault Diagnosis of Wheelset Bearings in High-Speed Trains. *IEEE Access* **2019**, *7*, 10278–10293. [CrossRef]
10. Li, X.; Jiang, H.; Niu, M.; Wang, R. An enhanced selective ensemble deep learning method for rolling bearing fault diagnosis with beetle antennae search algorithm. *Mech. Syst. Signal Process.* **2020**, *142*, 106752. [CrossRef]
11. Li, T.; Zhao, Z.; Sun, C.; Yan, R.; Chen, X. Multireceptive Field Graph Convolutional Networks for Machine Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2021**, *68*, 12739–12749. [CrossRef]
12. Tang, S.; Zhu, Y.; Yuan, S. Intelligent fault diagnosis of hydraulic piston pump based on deep learning and Bayesian optimization. *ISA Trans.* **2022**, *129*, 555–563. [CrossRef] [PubMed]
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *Microsoft* **2016**, 770–778. [CrossRef]
14. Wang, C.; Jiang, W.; Yue, Y.; Zhang, S. Research on Prediction Method of Gear Pump Remaining Useful Life Based on DCAE and Bi-LSTM. *Symmetry* **2022**, *14*, 1111. [CrossRef]
15. Pratt, Y.L. Discriminability-Based Transfer between Neural Networks. NIPS. Morgan-Kaufmann. 1992. Available online: https://proceedings.neurips.cc/paper/1992/hash/67e103b0761e60683e83c559be18d40c-Abstract.html (accessed on 16 February 2023).
16. Chen, Z.; Gryllias, K.; Li, W. Intelligent Fault Diagnosis for Rotary Machinery Using Transferable Convolutional Neural Network. *IEEE Trans. Ind. Inform.* **2020**, *16*, 339–349. [CrossRef]
17. Deng, Y.; Huang, D.; Du, S.; Li, G.; Zhao, C.; Lv, J. A double-layer attention based adversarial network for partial transfer learning in machinery fault diagnosis. *Comput Ind.* **2021**, *127*, 103399. [CrossRef]
18. Wang, Z.; He, X.; Yang, B.; Li, N. Subdomain Adaptation Transfer Learning Network for Fault Diagnosis of Roller Bearings. *IEEE Trans. Ind. Electron.* **2022**, *69*, 8430–8439. [CrossRef]
19. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q. The empirical mode decomposition and the Hilbert spectrum for non-linear and non-stationary time series analysis. *R. Soc.* **1998**, *454*, 903–995. [CrossRef]

20. Wu, Z.; Huang, N.-E. Ensemble empirical mode decomposition: A noise-assisteddata analysis method. *Adv. Data Sci. Adapt.* **2009**, *1*, 1–41. [CrossRef]
21. Torres, M.-E.; Colominas, M.-A.; Schlotthauer, G.; Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, 22–27 May 2011; pp. 4144–4147. [CrossRef]
22. Huang, N.-E.; Shen, Z.; Long, S.-R. A New View of Nonlinear Water Waves: The hilbert Spectrum. *Annu. Rev. Fluid. Mech.* **1999**, *31*, 417–457. [CrossRef]
23. Wu, Z.; Huang, N.; Long, S.; Peng, C. On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 14889–14894. [CrossRef] [PubMed]
24. Su, D.; Zheng, H. A boundary extension method for empirical mode decomposition end effect. *Acta Aeronaut. Astronaut. Sin.* **2016**, *37*, 960–969. [CrossRef]
25. Wu, J.; Wu, L.; Sun, M.; Lu, Y.; Han, Y. Application of Boundary Local Feature Scale Adaptive Matching Extension EMD Endpoint Effect Suppression Method in Blasting Seismic Wave Signal Processing. *Shock Vib.* **2021**, *2021*, 2804539. [CrossRef]
26. Lin, M.; Chen, Q.; Yan, S. Network In Network. *arXiv* **2013**, arXiv:1312.4400.
27. Pan, S.-J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
28. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the Importance of Initialization and Momentum in Deep Learning. *PME30* **2013**, *28*, III-1139–III–1147.
29. Peng, H.-C.; Long, F.-H.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef]
30. Ju, C.; Bibaut, A.; Van-der-Laan, M. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *J. Appl. Stat.* **2018**, *45*, 2800–2818. [CrossRef]
31. Lee, S.; Purushwalkam, S.; Cogswell, M.; Ranjan, V.; Crandall, D.; Batra, D.; Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; et al. (Eds.) *Stochastic Multiple Choice Learning for Training Diverse Deep Ensembles*; Virginia Tech.: Blacksburg, VA, USA, 2016.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.