A New Fault Diagnosis of Rolling Bearing Based on Markov Transition Field and CNN

Mengjiao Wang [1,*], Wenjie Wang [1], Xinan Zhang [2] and Herbert Ho-Ching Iu [2]

[1] School of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, China; wwj291968051@163.com

[2] School of Electrical, Electronic and Computer Engineering, University of Western Australia, Crawley, Perth, WA 6009, Australia; xinan.zhang@uwa.edu.au (X.Z.); herbert.iu@uwa.edu.au (H.H.-C.I.)

[*] Correspondence: wangmj@xtu.edu.cn

Abstract: The rolling bearing is a crucial component of the rotating machine, and it is particularly vital to ensure its normal operation. In addition, the selection of different category features will add uncertainty and bias to the classification results. In order to decrease the interference of these factors to fault diagnosis, a new method that automatically learns the features of the data combined with Markov transition field (MTF) and convolutional neural network (CNN) is proposed in this paper, namely MTF-CNN. The MTF contributes to convert the original time series into corresponding figures, and the CNN is used to extract the deep feature information in the figure to complete the fault diagnosis. The effectiveness of the proposed method is verified by two public data sets. The experimental results show that MTF-CNN can classify different types of faults, and the highest accuracy rate can reach 100%. Likewise, the classification accuracy of this method is higher than some existing methods.

Keywords: feature extraction; Markov transition field; convolutional neural network; fault diagnosis; rolling bearing

1. Introduction

In the modern industrial system, rotating machinery has been widely used, so it is particularly important to ensure the stable operation of machinery and equipment [1,2]. Rolling bearings are a crucial part of rotating machinery. Due to the influence of its work environment and other factors, rolling bearings have a higher probability of failure than other components [3–5]. Hence, the accurate detection of rolling bearing faults has great practical value.

With the advancement of sensor technology and signal processing methods, detecting equipment failures through the vibration signals of rotating machines becomes increasingly popular [6–8]. The signal processing method can effectively extract the characteristics of the original vibration signal in the field domain [9–11]. The signal processing methods mainly include: wavelet transformation (WT) [12], local mean decomposition (LMD) [13], empirical mode decomposition (EMD) [14], variational mode decomposition (VMD) [15] and so on. After extracting the hidden features form the time series, the full fault diagnosis process is completed by the combinational usage of several artificial intelligence methods, such as random forest (RF) [16], support vector machines (SVM) [17] and k-nearest neighbor (kNN) [18]. Zheng et al. achieved fault classification by calculating the composite multiscale fuzzy entropy (CMFE) of the vibration signal based on ensemble support vector machines (ESVM) [19]. Yan et al. decompose the vibration signal through an improved VMD (IVMD) to obtain the components. They proposed instantaneous energy distribution-permutation entropy (IED-PE) to obtain the feature vectors. At the same time, combined with kNN, the fault diagnosis of rolling bearing is realized [20]; however, most of the traditional processing methods count on manually extracting the fault information from the vibration

signal, which requires tedious signal processing work. In fact, the expertise of people who carry out signal processing can greatly impact the quality of classification results.

Currently, with the rapid progress of mathematical theory and computer technology, deep learning (DL) has been widely used. DL can automatically extract the non-linear relationships between variables from the data, avoiding the manual extraction process [21]; therefore, it is feasible to judge the running state of rolling bearings by the DL method. Such as deep belief network (DBN) [22], deep neural network (DNN) [23], sparse auto-encoder [24], convolutional neural network (CNN) [25] and so on. CNN, as one of the most popular DL methods, can effectively extract deep features of data. Wang et al. enhanced the relevant characteristics of the fault and realized the classification of the fault with 1D-CNN [26]. Compared with the other DL methods, CNN has obvious advantages in processing pictures. The CNN model is specially designed for processing 2-D or 3-D data. In some studies, the raw data are converted into pictures, and then the methods of image processing are used to classify the pictures to complete the entire process. Li et al. predicted the rest life of the bearing by extracting the time–frequency information of the data sequence and using CNN to complete feature extraction [27]. Wen et al. realized fault diagnosis on multiple platforms such as rolling bearings by converting original vibration signals into figures and combining with CNN [28].

The main contributions of this paper are as follows. Firstly, a new method of converting time series data into figures using the Markov transformation field (MTF) [29] is introduced. MTF is based on the Markov transition matrix, which is obtained from the first-order Markov chain. Due to the memorylessness of the Markov chain, the current elements in the Markov matrix are independent of the previous element, which results in the loss of much information in the conversion process. By introducing the time axis, more time information can be retained during the conversion process; therefore, the MTF transforms time series into corresponding images, which can decrease information loss. Then, via the CNN model applied in this paper, by reducing the level of convolutional layers and optimizing the parameters of the fully connected layer, the operating efficiency is improved. Finally, by combining the MTF with the CNN model proposed (MTF-CNN), a new method for fault diagnosis of rolling bearings is proposed.

The rest sections of this paper are organized as follows. Section 2 introduces the theory of MTF. Section 3 is the background of CNN. Section 4 describes the fault diagnosis method MTF-CNN and the structure of the CNN adopted. Section 5 verifies the effectiveness of MTF-CNN and compared with other DL methods in accuracy. Finally, the conclusions are drawn in Section 6

2. MTF Theory

2.1. State Transition Probability Matrix

State refers to the current system status. For event S, when we assume that there are n different states during its development, and the i-th state in the t moment is S_t. Then, the development process of events can be represented by the following:

$$S = \{S_1, S_2, \cdots, S_n\} \tag{1}$$

The probabilities of reaching a state is expressed as $E_i(t)$.

$$E_i(t) = p\{X_t = S_t\}, t = 1, 2, \cdots, n \tag{2}$$

Therefore, the state space of the research object includes k parts as expressed in (3) below.

$$T_k = \{E_1(k), E_2(k), \cdots, E_n(k)\} \tag{3}$$

according to the probability theory, $E_i(k)$ satisfies the condition: $\sum_{i=1}^{n} E_i(k) = 1, E_i(k) \geq 0$.

The state of the research object will change over time. When the state of the research object transform, S_i to S_j, this state change is probabilistic, which is called state transition

probability. According to the number of steps of state transition, the state transition probability can be divided into two types: one-step and multi-step.

One-step transition probability refers to the probability of changing one state to the other state. The research subject is in S_i at $t = T$, and the probability of state change S_j at $t = T + 1$ is P_{ij}. Its expression is as follows:

$$P_{ij} = p\{S_i \mid S_j\} = p\{X_{T+1} = S_j \mid X_T = S_i\} \tag{4}$$

Multi-step transition probability indicates the probability of continuous atate changing for multiple steps. The research subject is in S_i at $t = T$, after k times of state changes, and the probability of state change S_j at $t = T + 1$ is P_{ij}. Its expression is as follows:

$$P_{ij} = p\{S_j \mid S_i\} = p\{X_T = S_j \mid X_{T+k} = S_i\}, k = 2, 3, \cdots \tag{5}$$

During the entire transition process, the range of the state transition probability value is $(0, 1)$ and satisfies the following relationship:

$$\sum_{i,j=1}^{n} P_{ij} = 1, (i, j = 1, 2, \cdots, n) \tag{6}$$

The matrix composed of these transformation probabilities is called the transformation probability matrix, and the expression is as follows:

$$P = (P_{ij})_{N \times N} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix} \tag{7}$$

2.2. Markov Transformation Process

For a data series $X = \{x_1, x_2, \cdots, x_n\}$, divide it equally into Q bins, and each x_i in the series has bin $q_j(j \in [1, Q])$ corresponding to it; therefore, a Markov matrix W of size $Q \times Q$ through a first-order Markov chain can be constructed. Its expression is as follows:

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1Q} \\ w_{21} & w_{22} & \cdots & w_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ w_{Q1} & w_{Q2} & \cdots & w_{QQ} \end{bmatrix} \tag{8}$$

$$w_{ij} = p\{x_t \in q_i \mid x_{t-1} \in q_j\} \tag{9}$$

where w_{ij} is the probability that the point x currently belongs to bin q_i will appear in bin q_j at the next time. Obviously, w_{ij} satisfies the relation: $\sum_{j=1}^{Q} w_{ij} = 1$. Due to the memorylessness of the Markov chain, the current elements in the matrix W are independent of the previous element, which results in the loss of much effective information in the process. In order to overcome this shortcoming, by introducing a time axis, the matrix W is extended to the Markov transform field M. The data set is divided into Q bins along the time axis, and the data point at time stamps i and j are belong corresponding bin: q_i and q_j. M_{ij} represents the transition probability of q_i to q_j. It is expressed as below:

$$M = \begin{bmatrix} w_{ij} \mid x_1 \in q_i, x_1 \in q_j & \cdots & w_{ij} \mid x_1 \in q_i, x_n \in q_j \\ w_{ij} \mid x_2 \in q_i, x_1 \in q_j & \cdots & w_{ij} \mid x_2 \in q_i, x_n \in q_j \\ \vdots & \ddots & \vdots \\ w_{ij} \mid x_n \in q_i, x_1 \in q_j & \cdots & w_{ij} \mid x_n \in q_i, x_n \in q_j \end{bmatrix} \tag{10}$$

After the matrix *M* is obtained, elements in the matrix as pixels, thereby completing the whole process of MTF conversion. The specific process of MTF is shown in Figure 1.
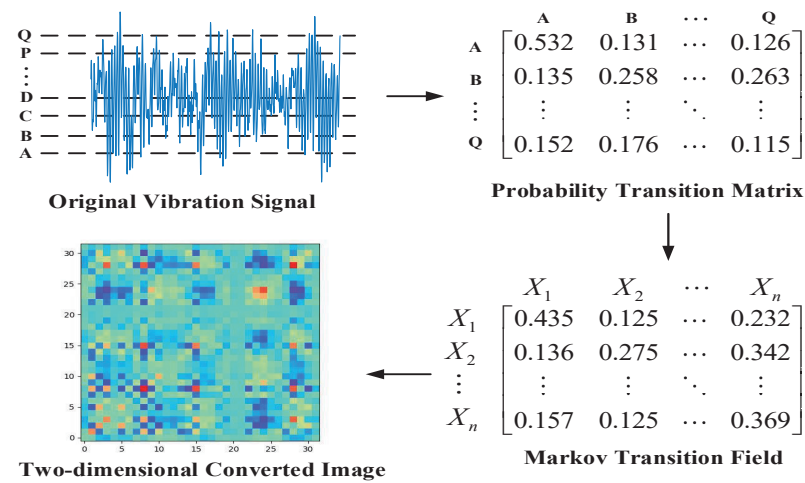

Figure 1. The flow chart of the MTF.

The specific steps are shown below:

- Divide the vibration signal into *Q* parts.
- Obtain the probability conversion matrix of $Q \times Q$ size.
- Convert the probability matrix to MTF.
- Transform MTF into a 2-D image.

3. The Brief Introduction of CNN

CNN is a multi-level neural network. Generally speaking, it includes the following structures: convolutional layer, pooling layer and fully connected layer. Next, each type of layer is described below.

3.1. Convolutional Layer

The convolution (Conv) layer is named after the convolution operation, but it usually uses the cross-correlation operation between matrices. The Conv layer consists of several convolution units, and the output is generated by the activation unit. The main function of the Conv layer is to extract the hidden features in the input image by calculation.

The convolutional layer has two important hyperparameters: padding (P) and stride (S). Their values affect the size of the output image of the Covn. Padding refers to filling elements in both sides of the height and width of the input matrix (usually filled elements are 0). Padding can increase the width and height of the output image, so that the output and input images have the same width and height. The stride refers to the number of rows and columns of each sliding of the convolution kernel window, which can reduce the width and height of the output image; therefore, after the image passes through the convolution layer, the size of the output picture can be expressed as:

$$O = \frac{N - F + 2P}{S} + 1 \tag{11}$$

where *N* and *F* are the sizes of the original input image and convolution kernel, respectively. *P* and *S* are the value of padding and stride. By default, the padding is 0 and the stride is 1.

The activation unit can increase the robustness and nonlinear expression ability of the CNN, which is conducive to better train the model. The common activation functions including Sigmoid, tanh, ReLu and so on. Among them, the ReLu function can improve the running speed of the model and accelerate the convergence of the model. We assume

that the convolutional layer *l* uses the convolution kernel *K*, then the entire convolutional layer process is described as follows:

$$
\begin{aligned}
y_l^i &= ReLu(K_l^i * X_{l-1}) \\
&= ReLu(\sum_j x_{r-1}^j * \omega_l^i + b_l^i)
\end{aligned}
\tag{12}
$$

where y_l^i means the output of the *i*-th convolution kernel, ω_i^l and b_i^l are the weight and bias of the *i*-th convolution kernel of layer *l*, respectively. X_{l-1} indicates the output of the previous convolutional layer, and its size is *j*. The symbol $*$ represents the operation between the local area and the convolution kernel.

3.2. Pooling Layer

The function of the pooling layer is to decrease the size of the feature value space and alleviate the excessive sensitivity of the convolutional layer to position. Compared with the convolutional layer to that calculates the correlation between input and convolutional kernel, The pooling layer mainly keeps the maximum value or average value of elements in the pooling window. The calculation process is also known as maximum pooling or average pooling. In this paper, the operation of pooling layer is maximum pooling, and its expression is as follows:

$$
H_{l+1}^n = \max_{(i-1)r+1 \leq t \leq ir} y_l^n(t)
\tag{13}
$$

where, H_{l+1}^n represents the output of the maximum pooling layer, *i* is the *i*-th area selected by the pooling window and *r* is the size of the area.

3.3. Fully Connected Layer

The fully connected (FC) layer is usually located at the end of the entire CNN. Its main contribution is to flatten the learned feature H_{l+1}^n into a vector V_{m-1}, and input the obtained vector into the classifier (such as the softmax classifier). It can be expressed as follows:

$$
\begin{aligned}
V_n &= g(\beta_n) \\
\beta_n &= (\delta_n)^T V_{n-1} + \mu_n
\end{aligned}
\tag{14}
$$

where δ_m and μ_m are the basic parameters of the full connection layer, and *n* is the quantity of them. *g* indicates the nonlinear activation function in the full connection layer.

3.4. Proposed CNN Structure

The CNN structure used in this paper is shown in Figure 2, which mainly includes four convolutional layers and four maximum pooling layers. A full connection layer and an output layer (the output layer includes a full connection layer and a Sotfmax classifier). Table 1 shows the specific network structure parameters.

The convolution kernel sizes of the convolutional layers *L2*, *L4*, *L6* and *L8* are all 2×2, the input images of each layer are filled with zero elements. According to the Equation (11), the size of the output image of each convolutional layer can be calculated. The activation unit used in the convolution layer and pool layer are ReLU. The activation function introduces nonlinearity to the CNN, which is advantageous in solving complex problem. In the *FC*1 layers, the number of neurons set are 256. The output layer includes a FC layer and a softmax classifier.

Since the size of the filter used in the CNN pooling layer is 2×2, after the image passes through the pooling layer, the output size of the image will be decreased to $1/2$ of the original size. The size of input image should be 2^n. Based on this factor and the limitation of the amount of experimental data, the size of the original input image set is 32×32.
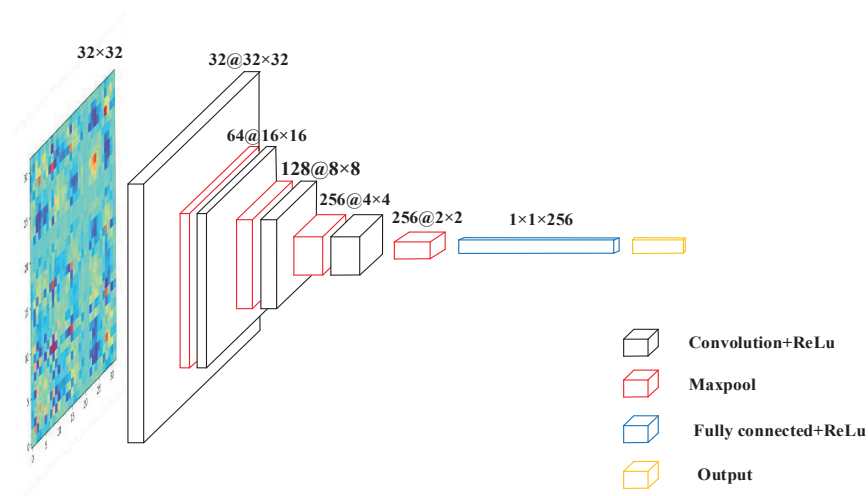
Figure 2. CNN model structure.

Table 1. The parameters of CNN.

Layer	CNN Models	Kernel Size	Padding	Stride
*Layer*1	Conv	3×3	1	1
*Layer*2	Maxpool	2×2	No	1
*Layer*3	Conv	3×3	1	1
*Layer*4	Maxpool	2×2	No	1
*Layer*5	Conv	3×3	1	1
*Layer*6	Maxpool	2×2	No	1
*Layer*7	Conv	3×3	1	1
*Layer*8	Maxpool	2×2	No	1
*Layer*9	FCl	256	-	-

4. The Process of MTF-CNN

Based on the theory of MTF and CNN, a method called MTF-CNN is proposed, which can accurately and effectively complete the fault classification. The process of MTF-CNN is illustrated in Figure 3, and the specific steps are shown below:

- Collect the vibration signals.
- Convert 1-D vibration signals into MTF images.
- Input the images into CNN for classification.
- Obtain the result of fault classification.
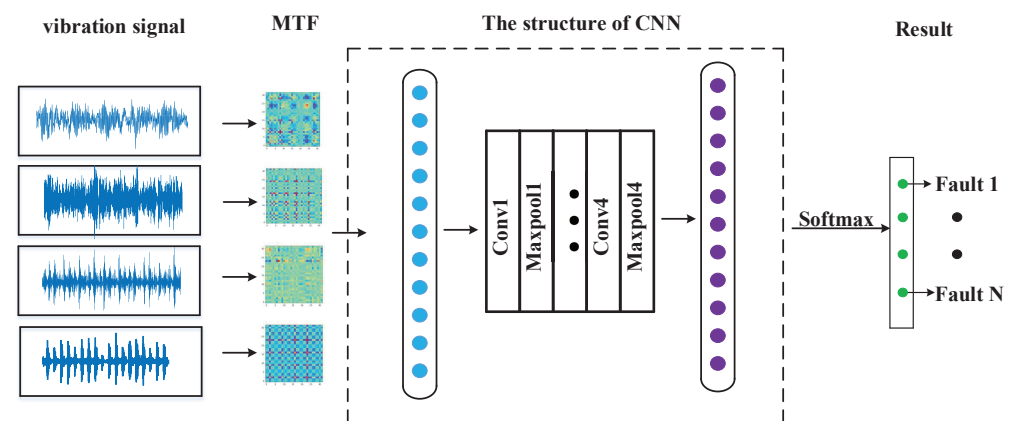
**Figure 3.** The flow chart of the MTF-CNN.

5. Experiment Test

In this part, two data sets were used to verify the feasibility of MTF-CNN. The CNN model was written in Python 3.7 with PyCharm, and the experiment was applied on a computer with a Core I5 7th generation CPU and 8 GB memory.

5.1. Experiment 1

5.1.1. Data Description

The data sets used in experiment 1 were obtained from the Bearing Data Center of Case Western Reserve University (CWRU). In this data set, the original vibration signals are obtained by artificially setting fault points in different parts of the rolling bearings, and the experimental platform is illustrated in Figure 4.
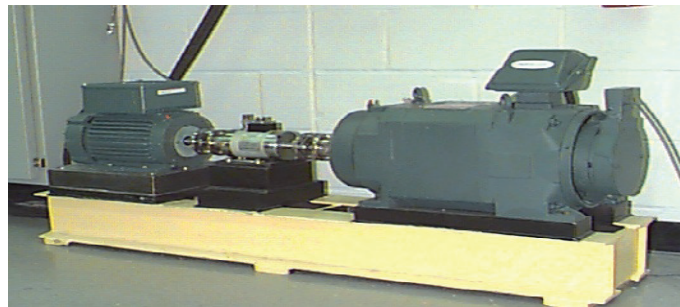
Figure 4. CWRU bearing platform.

The CWRU platform mainly contains three parts. The left end is a two horsepower motor, the middle part is a torque sensor and the right end is a power meter. The electrical discharge machining technology is used to punch out the fault point on the bearing, and its fault range is 0.007 inches to 0.040 inches. We collected a data set containing the normal (NOR) conditions of the rolling bearing and the other three failure conditions (inner race (IR), outer race (OR) and ball (B)). The bearing data parameters used in this paper: sampling frequency is 12 kHz, bearing speed is 1797 rpm, load is 0HP. A total of 1024 points in the vibration signal are taken as a sample, and the sampling time is 0.0853 s. Figure 5 shows the original vibration images and corresponding spectra in the CUWR dataset under four different working conditions.


| (**a**) | (**b**) |
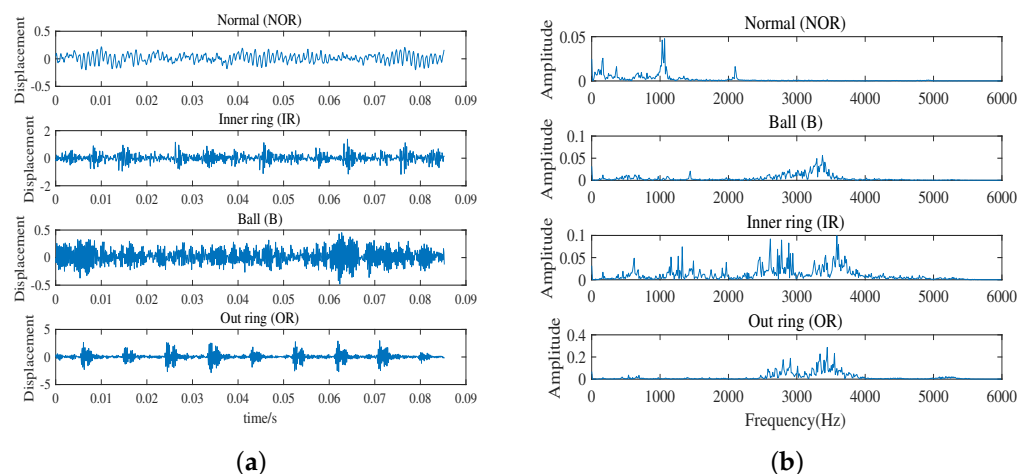
Figure 5. Waveforms of vibration signal in four conditions. (**a**) The diagram of time domain. (**b**) The diagram of frequency domain.

5.1.2. Result Analysis

The vibration signals in the four cases are converted into MTF, and the conversion results are shown in Figure 6. It can be seen that under different fault conditions, these

converted images are totally different from each other, and these images can be intuitively classified.

**(a)** NOR

(b) IR

(c) B

(d) OR
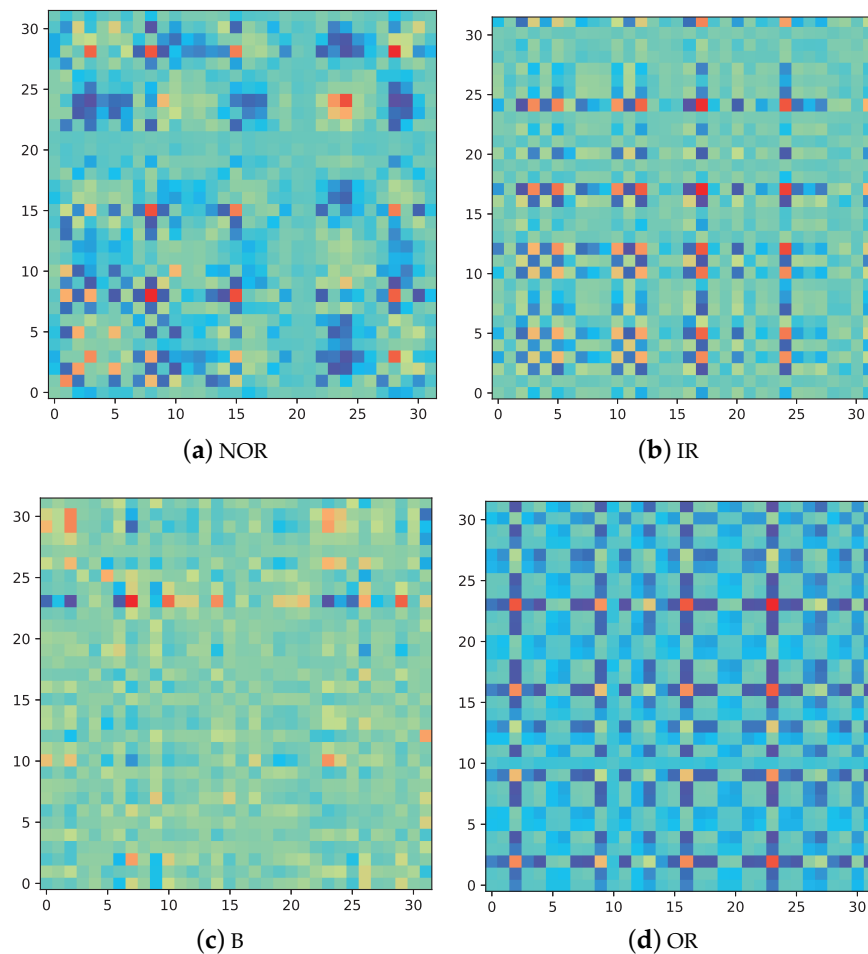
Figure 6. The converted images on four fault conditions.

We collected 100 samples under each working condition, for a total of 400 samples. The algorithm randomly selects 70% of the total data sets as the training set and the rest 30% as the test set. A total of 10 rounds of training were carried out, and the number of epochs of each training is 100. The purpose of this approach is to avoid the special data partition, which leads to the deviation of classification results.

Since the training sets and test sets of the CNN model in each round of training are random, the classification results of each round will be different. In the course of ten rounds of training, the highest classification accuracy is 100%, and the lowest is 99.18%. The average accuracy rate is 99.88%. The confusion matrices of the CNN classification result are shown in Figure 7.

In order to show the advantages of MTF-CNN in classification accuracy, other deep learning methods were selected for comparison under the same conditions. The details are shown in Table 2. The LSTM model is a time recursive network. The classifier used for data processing is softmax and the controller is Adam. The learning rate of the LSTM model used in this article is 0.001. The batch-size is 27 and the epochs are 100. SVM is a learning model that can be used for classification. The SVM in this paper uses GridSearchCV (5−fold cross verification parameters). The kernel function uses a radial basis kernel function (RBF). The feature values of LSTM and SVM are directly composed of points in the data sets. At the same time, the fuzzy entropy (FE) is introduced as the feature value of the two classification algorithms to compare the classification effects.
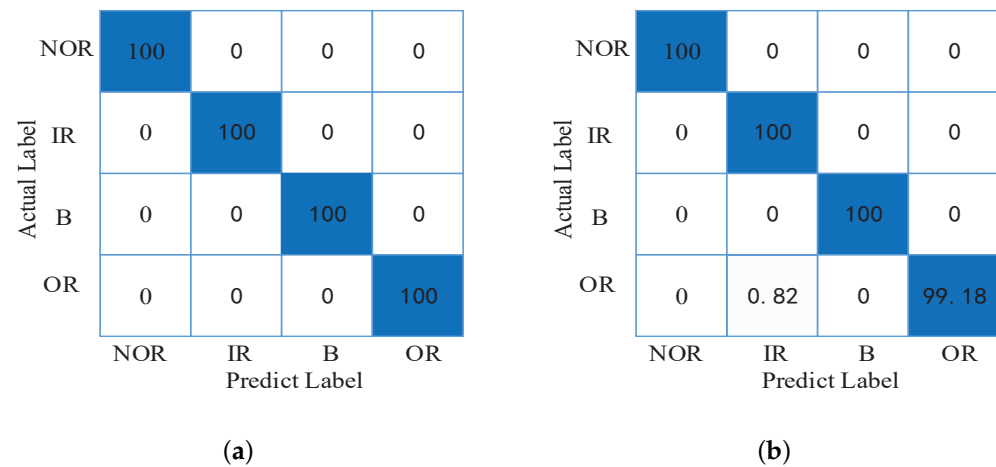
Figure 7. The confusion matrix of CNN in CWRU data set. (**a**) The highest classification result. (**b**) The lowest classification result.

From the Table 2, we can see that the MTF-CNN is far better than LSTM and SVM in terms of classification effect. Among them, the average accuracy rate of MTF-CNN is about 99.88%. For the classification effect of LSTM and SVM, it is better to use fuzzy entropy as the feature value than to directly use the points in the data set as the feature value. Moreover, during the ten experiments of MTF-CNN, the fluctuation of classification accuracy is smaller than that of the other two methods.

Table 2. The accuracy of six methods.

Method	Highest Accuracy	Lowest Accuracy	Mean
MTF-CNN	100%	99.18%	99.88%
LSTM	80.4%	60.3%	73.8%
FE-LSTM	91.4%	88.7%	90.3%
SVM	90.3%	83.5%	89.4%
FE-SVM	98.75%	96.25	97.5%

In [30], the author analyzes the application of the CWRU data set in bearing fault diagnosis. According to [30], the data are divided into three categories: diagnosable (category Y), partially diagnosable (category P) and non-diagnosable (category N). At the same time, the literature holds that the new fault diagnosis method should be verified by P-category or N-category data sets. The normal data and outer ring data used in this paper belong to the Y-category, and the ball data belongs to the N-category; therefore, the data set used in this paper leads to similar conclusions.

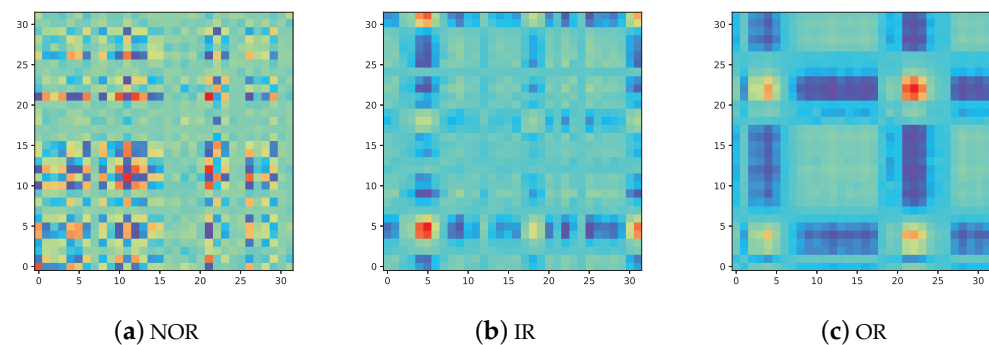## 5.2. Experiment 2

5.2.1. Data Description

This data set comes from Mechanical Failure Prevention Technology (MFPT), including the normal (NOR) condition of the fixed load, outer ring (OR) failure and inner ring (IR) failure under various loads. The parameters of the normal data set used in this paper are as follows: the load is 270 lbs and the input shaft speed is 25 Hz. Outer and inner fault data parameters: load 300 lbs and input shaft speed 25 Hz. A total of 1024 vibration signal data points were taken as a sample. The original vibration images and corresponding spectra in the MFPT dataset under three different working conditions are shown in Figure 8.
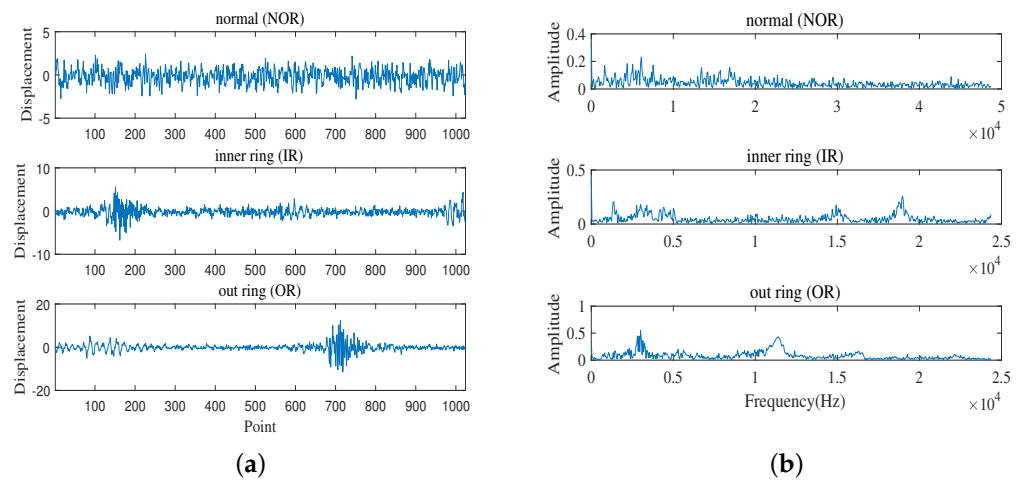
Figure 8. Waveforms of vibration signal in three conditions. (**a**) The diagram of time domain. (**b**) The diagram of frequency domain.

5.2.2. Result Analysis

Convert different types of vibration signals into the corresponding MTF images and the conversion results are shown in Figure 9. It can be seen from the figure that there are obvious differences in pixel distribution among different types of MTF images.

Figure 9. The converted images on three fault conditions.

A total of 100 samples were collected under each working condition, for a total of 300. The data set division is the same as experiment 1, and the train set and test set account for 70% and 30% of the total number, respectively. A total of ten rounds of experiments were conducted, and each round was run for 100 epochs. The purpose of this practice is to avoid biased results due to special training and test sets. In ten rounds of experiments, the CNN classification accuracy under this data set has reached 100%, and the classification results do not have obvious fluctuations for different test sets. Its confusion matrix is shown in Figure 10.

Same as experiment 1, the classification accuracy of MTF-CNN and other methods is compared under MPTF data set, and the detailed comparison results are shown in Table 3. As can be seen from the table, the classification accuracy of MTF-CNN is higher than that of other methods. At the same time, the accuracy rate did not fluctuate greatly in multiple experiments, which shows that MTF-CNN has better robustness.

Figure 10. The confusion matrix of CNN in MFPT data set.

Table 3. The accuracy of six methods.

Method	Highest Accuracy	Lowest Accuracy	Mean
MTF-CNN	100%	100%	100%
LSTM	87.6%	71.5%	79.8%
FE-LSTM	95.4%	90.7%	93.3%
SVM	92.6%	88.5%	91.5%
FE-SVM	99%	97%	98.6%

In summary, two experiments prove that MTF-CNN can accurately and effectively identify pictures with different labels, thus realizing the classification and diagnosis of faults. Meanwhile, compared with other methods such as LSTM, MTF-CNN is superior in classification accuracy.

6. Conclusions

In this paper, a method for diagnosing rolling bearing faults based on MTF-CNN is proposed. It eliminates the tedious signal processing work that is required to extract the fault information from the original data and reduces the manpower cost due to its autonomous classification. Furthermore, compared to the other popular machine learning algorithms, the proposed method produces better accuracy with reduced number of data samples. The effectiveness of the MTF-CNN is verified by experimental results. The number of data samples used in this paper is less than that of general papers; however, by optimizing the structure of CNN, the dependence of CNN model on training samples can be reduced.

Author Contributions: Conceptualization, M.W. and W.W.; methodology, M.W.; software, W.W.; validation, M.W. and W.W.; formal analysis, M.W. and W.W.; investigation, M.W.; resources, W.W.; data curation, W.W.; writing—original draft preparation, W.W.; writing—review and editing, M.W., X.Z. and H.H.-C.I. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experimental data are available from the Bearing Data Center of Case Western Reserve University deposited in https://engineering.case.edu/bearingdatacenter/welcome (accessed on 1 May 2022) and the Mechanical Failure Prevention Technology deposited in https://www.mfpt.org/fault-data-sets/(accessed on 1 May 2022).

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

References

1. Heng, A.; Zhang, S.; Tan, A.; Mathew, J. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mech. Syst. Signal Proc.* **2009**, *23*, 724–739. [CrossRef]
2. Shao, H.; Jiang, H.; Wang, F.; Zhao, H. An enhancement deep feature fusion method for rotating machinery fault diagnosis. *Knowl.-Based Syst.* **2017**, *119*, 200–220. [CrossRef]
3. Zhang, Y.; Randall, R. Rolling element bearing fault diagnosis based on the combination of genetic algorithms and fast kurtogram. *Mech. Syst. Signal Proc.* **2009**, *23*, 1509–1517. [CrossRef]
4. Xu, M.; Feng, G.; He, Q.; Gu, F; Ball, A. Vibration characteristics of rolling element bearings with different radial clearances for condition monitoring of wind turbine. *Appl. Sci.* **2020**, *14*, 4731. [CrossRef]
5. Liu, Z.; Zhang, L. A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings. *Measurement* **2020**, *149*, 107002. [CrossRef]
6. Dybała, J.; Zimroz, R. Rolling bearing diagnosing method based on empirical mode decomposition of machine vibration signal. *Appl. Acoust.* **2014**, *77*, 195–203. [CrossRef]
7. Ying, Y.; Li, J.; Chen, Z.; Guo, J. Study on rolling bearing on-line reliability analysis based on vibration information processing. *Comput. Electr. Eng.* **2018**, *69*, 842–851. [CrossRef]
8. Wang, J.; Peng, Y.; Qiao, W. Current-aided order tracking of vibration signals for bearing fault diagnosis of direct-drive wind turbines. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6336–6346. [CrossRef]
9. Miao, Y.; Zhao, M.; Lin, J.; Lei, Y. Application of an improved maximum correlated kurtosis deconvolution method for fault diagnosis of rolling element bearings. *Mech. Syst. Signal Proc.* **2017**, *92*, 173–195. [CrossRef]
10. Wang, L.; Shao, Y.; Cao, Z. Optimal demodulation subband selection for sun gear crack fault diagnosis in planetary gearbox. *Measurement* **2018**, *125*, 554–563. [CrossRef]
11. Zhang, M.; Jiang, Z.; Feng, K. Research on variational mode decomposition in rolling bearings fault diagnosis of the multistage centrifugal pump. *Mech. Syst. Signal Proc.* **2017**, *93*, 460–493. [CrossRef]
12. Li, Y.; Liang, X.; Xu, M.; Huang, W. Early fault feature extraction of rolling bearing based on ICD and tunable Q-factor wavelet transform. *Mech. Syst. Signal Proc.* **2017**, *86*, 204–223. [CrossRef]
13. Li, X.; Ma, J.; Wang, X.; Wu, J.; Li, Z. An improved local mean decomposition method based on improved composite interpolation envelope and its application in bearing fault feature extraction. *ISA Trans.* **2020**, *97*, 365–383. [CrossRef] [PubMed]
14. Riaz, F.; Hassan, A.; Rehman, S.; Niazi, I.; Dremstrup, K. EMD-based temporal and spectral features for the classification of EEG signals using supervised learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2015** , *24*, 28–35. [CrossRef]
15. Zhang, X.; Miao, Q.; Zhang, H.; Wang, L. A parameter-adaptive VMD method based on grasshopper optimization algorithm to analyze vibration signals from rotating machinery. *Mech. Syst. Signal Proc.* **2018**, *108*, 58–72. [CrossRef]
16. Quiroz, J.; Mariun, N.; Mehrjou,. M.; Izadi, M.; Misron, N.; Radzi, M. Fault detection of broken rotor bar in LS-PMSM using random forests. *Measurement* **2018**, *116*, 273–280. [CrossRef]
17. Zheng, J.; Pan, H.; Yang, S.; Cheng, J. Generalized composite multiscale permutation entropy and Laplacian score based rolling bearing fault diagnosis. *Mech. Syst. Signal Proc.* **2018**, *99*, 229–243. [CrossRef]
18. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Wang, R. Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 1774–1785. [CrossRef]
19. Zheng, J.; Pan, H.; Cheng, J. Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines. *Mech. Syst. Signal Proc.* **2017**, *85*, 746–759. [CrossRef]
20. Yan, X.; Jia, M.; Zhao, Z. A novel intelligent detection method for rolling bearing based on IVMD and instantaneous energy distribution-permutation entropy. *Measurement* **2018**, *130*, 435–447. [CrossRef]
21. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
22. Shao, H.; Jiang, H.; Zhang, H.; Duan, W.; Liang, T.; Wu, S. Rolling bearing fault feature learning using improved convolutional deep belief network with compressed sensing. *Mech. Syst. Signal Proc.* **2018**, *100*, 743–765. [CrossRef]
23. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech. Syst. Signal Proc.* **2016**, *72*, 303–315. [CrossRef]
24. Wen, L.; Gao, L.; Li, X. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Trans. Syst. Man Cybern. -Syst.* **2017**, *49*, 136–144. [CrossRef]
25. Xia, M.; Li, T.; Xu, L.; Liu, L.; De, S.C. Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks. *IEEE-ASME Trans. Mechatron.* **2017**, *23*, 101–110. [CrossRef]

26. Wang, H.; Liu, Z.; Peng, D.; Qin, Y. Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5735–5745. [CrossRef]
27. Li, X.; Zhang, W.; Ding, Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 208–218. [CrossRef]
28. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5990–5998. [CrossRef]
29. Wang, Z.; Oates, T. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In Proceedings of the Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
30. Smith, A.; Robert, B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mech. Syst. Signal Proc.* **2015**, *64*, 100–131. [CrossRef]