

GNPENet: A Novel Convolutional Neural Network With Local Structure for Fault Diagnosis

Jinping Wang¹, Ruisheng Ran¹, and Bin Fang², *Senior Member, IEEE*

Abstract—With the development of modern industry, fault diagnosis has become an important research field. Currently, many methods for fault diagnosis have been proposed. As a method designed to overcome the high computational complexity of the convolutional neural network (CNN), PCANet is also used in the field of fault diagnosis. The core algorithm of the PCANet is the principal component analysis (PCA) algorithm. However, PCA is a global dimensionality reduction method, which cannot effectively analyze the local spatial geometry structure of data and may even weaken or destroy the local structure information. Furthermore, algorithms based on L2-norm are very sensitive to noise and outliers. To address the problems, a generalized neighborhood preserving embedding (GNPE) method based on Lp-norm is first proposed, and it is used to learn the convolutional filters of CNN. In this way, a novel CNN with local structure is proposed, called the GNPE Network (GNPENet). We conduct fault diagnosis experiments on five datasets of three types and comprehensively evaluate the proposed GNPENet method. The experimental results show that, compared with some state-of-the-art models, GNPENet has better feature extraction and generalization capabilities.

Index Terms—Deep learning (DL), fault diagnosis, generalized neighborhood preserving embedding network (GNPENet).

I. INTRODUCTION

FAULT diagnosis [1], [2] plays an important role in modern industry. In various complex working environments, the components of the machine are very fragile. In particular, the rotating components of machines, such as rotors, gears, and bearings, are usually the core of mechanical equipment, but they are prone to malfunctions. Accurate fault diagnosis is not only beneficial for equipment protection but also closely related to life safety. Numerous studies [3], [4], [5] have focused on how to improve the safety, reliability, and maintainability of machines. The data-driven fault diagnosis [6], [7] is

Manuscript received 22 August 2023; revised 1 October 2023; accepted 13 October 2023. Date of publication 1 November 2023; date of current version 29 December 2023. This work was supported in part by the Key Project for Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K202100505 and in part by the Chongqing Technology Innovation and Application Development Project under Grant cstc2020jscx-msxmX0190 and Grant cstc2019jscxmbdxX0061. The Associate Editor coordinating the review process was Dr. Huan Liu. (Corresponding author: Ruisheng Ran.)

Jinping Wang is with the College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China (e-mail: mw530@foxmail.com).

Ruisheng Ran is with the College of Computer and Information Science and the College of Intelligent Science, Chongqing Normal University, Chongqing 401331, China (e-mail: rshran@cqnu.edu.cn).

Bin Fang is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: fb@cqu.edu.cn).

Digital Object Identifier 10.1109/TIM.2023.3329156

a vital subspecialty of the fault diagnosis field, which is based on prior models and a large amount of historical data. There are mainly two components in the data-driven fault diagnosis methods: data and model. The data used in the data-driven fault diagnosis are often collected by the vibration sensors attached to the mechanical parts. The models can be roughly divided into two categories: machine learning (ML) and deep learning (DL) models. The tasks of the models are feature extraction and classification. A series of models are proposed to achieve more accurate fault diagnosis because of their vital role in data-driven fault diagnosis.

Some typical ML methods have been applied in the field of data-driven fault diagnosis and have achieved good results. The most representative ones are the principal component analysis (PCA) algorithm [8] based on subspace features and the fast Fourier transform (FFT) [9] algorithm based on frequency-domain features. Since the good feature extraction ability and usability of PCA, it is widely used in fault diagnosis. For example, Ali et al. [10] applied the multiscale PCA-signed directed graph to detect and diagnose the root cause and identify the fault propagation path. Zhang et al. [11] proposed a new method named subspace reconstruction-based robust kernel PCA to extract nonlinear features under discontinuous interference. Cao et al. [12] used PCA and the support vector machine theory and proposed a fault diagnosis method. FFT is also a widely used technology in the field of fault diagnosis due to its ability to convert the time-domain signal into the frequency domain. Many researchers use FFT and its varieties to extract the fault feature. For instance, Lu et al. [13] employed the variational mode decomposition and FFT to extract the feature of the wind turbine bearing. The short-FFT (SFFT) that not only keeps the resolution but also reduces the consumed computation and time is used for fault diagnosis of induction machines working under transient regimes [14]. Zhao et al. [15] proposed a bearing multifault diagnosis strategy that uses the Hilbert transform and the SFFT to calculate the time-frequency representation of the resonance frequency.

DL methods represented by the deep neural network (DNN), on the other hand, perform better in data-driven fault diagnosis. Convolutional neural network (CNN) is the most widely used DNN whose main idea is to add convolution operation into the network. The core layer of CNN mainly consists of three layers, convolutional layers, nonlinearity layers, and pooling layers. CNNs are also applied in fault diagnosis to extract more accurate features as their powerful feature extraction capability. For example, ResNet [16], whose main idea is to

add the residual structure to solve the problems of gradient disappearance and gradient explosion, is also widely used in fault diagnosis. Liang et al. [17] proposed an improved ResNet with a new pooling layer and a global singular value decomposition for fault diagnosis. Wei et al. [18] proposed a fault diagnosis method based on PCA and ResNet for fault diagnosis by denoising the telemetry signal of the spacecraft control system and extracting the deep-level feature. Gu et al. [19] proposed a novel hybrid model based on continuous wavelet transform and improved CNN (ICNN) for the rolling bearing defects diagnosis in spindle devices. Jin et al. [20] proposed a multi-scale CNN (MS-CNN) to enhance the robustness of the model under various noise conditions. Furthermore, Zhou et al. [21] proposed an evidence visual geometry group (EVGG) neural network and constructed a reliable and intelligent fault diagnosis framework. However, stochastic gradient descent (SGD) and backpropagation (BP) algorithms in CNN also face many problems, the most important of which is that more computations are needed to learn more representative features. Thus, the feedforward learning network is proposed to overcome the above challenge.

Feedforward learning networks represented by PCANet [22] have been proposed to overcome the problem. PCANet first introduced PCA, a data analysis method, to learn the convolutional filters from the image patches. The main idea of PCA is to transform the original data into a group of linearly independent representations of each dimension through linear transformation, which can be used to extract the main principal components of the data. Since the simplicity of PCANet, many variants are proposed, such as the nonlinear PCA network [23], mixtures of probabilistic PCA network [24], and 2-D PCA network [25]. Besides, PCANet and some of its variants are also employed for fault diagnosis. For example, a cascaded monitoring network motivated by PCANet is proposed to develop the monitoring model with concurrent analytics of temporal and spatial information [26]. Yang et al. [27] applied PCANet to detect the faults of the air conditioner and obtained excellent performance. Guo et al. [28] proposed an improved multilinear PCA network by introducing kernel transformation to better identify various faults of rolling bearings.

However, there are still some limitations to the learning process of PCANet and its variants. In particular, all variants improve the network performance by modifying the PCA algorithm. However, PCA, the core algorithm of PCANet, is a global dimensional reduction method, which cannot effectively analyze the spatial geometry structure of the data and may even weaken or destroy the local structure information [29]. Therefore, the features extracted by PCANet ignore the local information of the original data space, which leads to its failure to achieve better recognition results.

Based on the above description, the local information of the space should be taken into consideration. The neighborhood preserving embedded (NPE) algorithm [30] is a data dimensionality reduction method that considers the local structure of the manifold and then maintains the overall manifold structure by overlapping the locality. Since it considers local features, its performance is often better than PCA [31], [32]. Then,

consider that raw data for fault diagnosis often contain a lot of noise and outliers. However, the original NPE algorithm based on the L2-norm is very sensitive to noise and outliers. This affects the accuracy of the algorithm in fault diagnosis scenarios. Therefore, a novel NPE algorithm based on the Lp-norm named generalized NPE (GNPE) is innovatively proposed to overcome the problems. Due to the flexibility of Lp-norm, different norms constraints can be used depending on the scenario. For example, using the L1-norm makes the algorithm more robust and reduces the interference of noise and outliers. Or use the L2-norm to get high generalization performance in transfer scenarios. Based on the GNPE algorithm proposed above, a novel CNN with local structure named GNPE network (GNPENet) is proposed. The main idea of GNPENet is to apply the GNPE algorithm to learning the convolutional filters. Moreover, considering that the raw fault data are a continuous signal, which results in fault features that may be distributed over different scales [20]. Therefore, to extract features at multiple scales, spatial pyramid pooling (SPP) is added to the proposed GNPENet to obtain better performance.

The main contributions of this article are summarized as follows.

- 1) Considering the drawbacks of PCA and NPE, a novel data dimensionality reduction method based on the Lp-norm, which is named GNPE, is proposed.
- 2) Based on the GNPE algorithm proposed above, a novel CNN with local structure named GNPENet for fault diagnosis is proposed.
- 3) Considering that the raw fault diagnosis signals are multiscale, SPP is added to GNPENet to extract more representative features at multiple scales.
- 4) Extensive experiments on five fault diagnosis datasets have shown that the proposed GNPENet method not only has a high feature extraction ability but also a high generalization ability.

The rest of this article proceeds as follows. In Section II, we review the Lp-norm and SPP. In Section III, we introduce the proposed GNPE and GNPENet. In Section IV, we conduct experiments to evaluate the new methods. The conclusion and future work are given in Section V.

II. REVIEW OF RELATED WORKS

A. Lp-Norm

It is known that the definition of distance is a broad concept that can be called distance as long as it satisfies the nonnegative, reflexivity, and triangle inequality. The norm is a strengthened concept of distance, which has one more rule the multiplicative rule than the distance in its definition. In ML algorithms, the norm is often used as a measure of distance between samples or regularization. Then, we will explain the L2-, L1-, and Lp-norms in terms of regularization restrictions.

In data dimensionality reduction, the L2-norm is usually used to define the distance between sample points, of which the Euclidean distance is a typical L2-norm. L2-norm is often used as regularization terms for optimization objective functions to prevent the model from becoming too complex to

cater to the training set resulting in overfitting, thus improving the generalization ability of the model.

The L1-norm is widely used as an alternative to the L2-norm as a way to improve the model performance. The advantage of the L1-norm is that it produces sparse solutions, making the model easier to interpret and understand. Compared with the L2-norm, the L1-norm is stable and robust and has better adaptability to some outliers and noise. However, the L1-norm restrictions objective is hard to solve since it is not derivable near zero.

All of the above widely used norms can be seen as special cases of the L_p -norm. Supposed that the norm value $0 < p < 1$, it can be viewed as a tradeoff between the L0- and L1-norms, with the expectation of obtaining the advantages of both. Although, in this case, the L0-norm is nonconvex and does not satisfy the definition of a norm, it can be somewhat closer to the L0-norm in the event that the L0- and L1-norms are not equal. When $1 < p < 2$, it can be seen as a tradeoff between L1- and L2-norms. Since the result of the L1-norm is sparse, the features can be extracted more accurately in a particular case. However, this sparsity also limits the generalization performance of the model to a certain extent, so the L_p -norm for $1 < p < 2$ can improve the generalization performance of the model while retaining the original sparsity to a certain extent to improve the classification accuracy.

B. Spatial Pyramid Pooling

SPP is a pooling strategy that can handle images of different sizes without requiring image editing, such as warping or cropping. It can also extract features at different scales, capturing various spatial information that helps in better understanding the context and structure of the objects in an image. Moreover, it can pool features into multilevel spatial bins, instead of relying on a single-window size. This allows SPP to adapt to various object sizes and capture richer and more informative representations of the objects. SPP works well with the spatial pyramid machine and the bag-of-words model, which are effective and widely used algorithms in the field of computer vision. By using SPP, the feature dimensions are reduced, which reduces the model complexity and improves the training time. Furthermore, SPP preserves critical visual details that may be lost or changed by image editing, leading to improved object recognition and localization. Therefore, SPP has proven to be an effective and versatile pooling strategy in the field of computer vision and a valuable tool in various image processing and classification tasks.

III. PROPOSED METHODS

A. Generalized Neighborhood Preserving Embedding

The original NPE restriction on the objective function is based on the L2-norm and only applies a looser L2-norm restriction on the projection matrix. However, the L2-norm restriction alone does not work well for various application scenarios. As stated in Section II-A, the matrices under the L1-norm and L2-norm restrictions have different properties.

The L1-norm constraint makes the matrix sparse, which enables automatic feature selection and improves the

algorithm's feature extraction ability by discarding noisy features. Sparsity also makes the model more interpretable, as the features with a coefficient of 1 directly influence the result.

The L2-norm constraint does not make the matrix sparse, and it is more sensitive to anomalies, outliers, and noise, which lowers the model's accuracy. L2-norm usually improves the model's generalization performance more than L1-norm, which may be because L1-norm selects features more selectively, but the feature distribution in different data scenarios may not be consistent. L2-norm is better for cross-data scenario transfer, as our experiments and other literature confirm [33], [34], [35].

Based on the above analysis, a generalized neighborhood preserving embedding (GNPE) algorithm based on the L_p -norm is proposed. Specifically, we impose the L_p -norm on the projection matrix to obtain the corresponding properties. At the same time, a generalized norm number constraint is also imposed on the objective function. Thus, the proposed algorithm can be adapted to different application scenarios. For example, when on the same dataset, the precise selection of features to improve the accuracy of prediction is desired, so a norm (e.g., L1-norm) that can produce sparse features can be chosen. When the generalization ability of the model is the primary consideration, a norm that prevents overfitting (e.g., L2-norm) can be chosen. Therefore, compared to the original algorithm, the proposed method is more flexible and can extract different aspects of features according to different scenarios, thus better utilizing the NPE algorithm.

Moreover, to better match the network structure we propose in the following, GNPE is actually a method for processing 2-D images, that is, the input to the GNPE algorithm is the complete 2-D image, not a 1-D vector as in the original NPE. Thus, suppose that there are N images denoted as X_1, X_2, \dots, X_N , where $X_i \in \mathbb{R}^{m \times n}$, $i = 1, 2, \dots, N$, m and n are the height and the width of the images, respectively.

1) *Algorithm Flow*: The steps of the GNPE algorithm are first given as follows.

a) *Construct the adjacency graph*: An adjacency graph \mathbf{G} containing l nodes is employed to represent the adjacency relationship of the nodes, i.e., the element G_{ij} is 1 if the node i is adjacent to the node j and zero otherwise. There are two algorithms to construct \mathbf{G} .

- 1) *k Nearest Neighbor (KNN)*: Nodes i and j are marked as adjacent if i is among the k closest neighbors of j .
- 2) *ε -Neighborhood*: Nodes i and j are marked as adjacent if $\|X_i - X_j\|_F^2 \leq \varepsilon$.

b) *Compute the weights*: The element W_{ij} of the weight matrix \mathbf{W} denotes the weight of the edge connecting node i and node j , which is 0 if nodes are not adjacent. The weight matrix \mathbf{W} can be obtained by minimizing the following object function:

$$\sum_i^N \left\| X_i - \sum_j^N W_{ij} X_j \right\|_F \quad (1)$$

with constraint

$$\sum_j^N W_{ij} = 1. \quad (2)$$

This step maps each point and its nearest neighbors to a locally linear patch on the manifold. The patch geometry and point reconstruction are represented by the weight matrix \mathbf{W} .

c) *Computing the projection:* Suppose that Z_i denotes the optimal projection of the data point X_i from high-dimensional space $\mathbb{R}^{m \times n}$ to low-dimensional space $\mathbb{R}^{m \times d}$ ($d \ll n$) and it is computed by a linear transformation

$$Z_i = X_i A \quad (3)$$

where $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r]$ is the projection matrix and \mathbf{a}_j ($j = 1, 2, \dots, r$) is the j th projection vector.

The following reconstruction function from high dimension to low dimension is being minimized to compute the linear transformation A :

$$\min_A \sum_i^N \left\| Z_i - \sum_j^N W_{ij} Z_j \right\|_s^s. \quad (4)$$

The orthogonal constraint $\|\mathbf{a}\|_p^p = 1$ is imposed on (4) to ensure that the projected low-dimensional features are as independent of each other as possible, where $s \geq 1$ and $p > 0$.

Equation (4) can be transformed into an eigenvalue problem and the projection vector can be obtained by combining the eigenvectors corresponding to the minimum few eigenvalues of this eigenvalue problem. Therefore, in order to solve the problem more easily, (4) can be converted to the form of (5)

$$\begin{aligned} \min_A \sum_{i=1}^N \left\| Z_i - \sum_j^N W_{ij} Z_j \right\|_s^s \\ = \min_A \sum_{i=1}^N \left\| X_i A - \sum_j^N W_{ij} X_j A \right\|_s^s \\ = \min_A \sum_{i=1}^N \left\| (X_i - \sum_j^N W_{ij} X_j) A \right\|_s^s \\ = \min_A \sum_{i=1}^N \left\| \widetilde{X}_i A \right\|_s^s \end{aligned} \quad (5)$$

with constraint $\|\mathbf{a}\|_p^p = 1$, where $\widetilde{X}_i = X_i - \sum_j^N W_{ij} X_j$.

Since optimizing the above objective function is a more complex problem that requires the use of theories such as the MM framework [36], [37], and the solution process of the algorithm is not the major part of this article, we have given it inside the Appendix (5) can be easily transformed into a problem solvable in [36] and will not be repeated here. The solution to the above objective is directly given as follows.

Let $\mathbf{v}^k = \sum_{i=1}^N \widetilde{X}_i^T [|\widetilde{X}_i \mathbf{a}^k|^{s-1} \circ \text{sign}(\widetilde{X}_i \mathbf{a}^k)]$, where k is the iteration number, \mathbf{a}^k is the projection vector of the k th step, $\text{sign}(\cdot)$ means the sign function, and $|\cdot|$ denotes the absolute value; the solution can be divided into four cases.

1) *Case 1:* When $0 < p < 1$, the solution can be obtained by the following two steps:

$$\mathbf{u}^k = |\mathbf{a}^k|^{2-p} \circ \mathbf{v}^k \quad (6)$$

$$\mathbf{a}^{k+1} = \frac{\mathbf{u}^k}{\|\mathbf{u}^k\|_p}. \quad (7)$$

2) *Case 2:* When $p = 1$, the solution is

$$a_i^{k+1} = \begin{cases} \text{sign}(v_j^k), & i = j \\ 0, & i \neq j \end{cases} \quad (8)$$

where $j = \arg \max |v_i^k|$ and $i \in [1, 2, \dots, w]$, i.e., $|v_j^k|$ is the largest value of $|v^k|$.

3) *Case 3:* When $p > 1$ and $p \neq \infty$, the solution can be obtained by the following two steps:

$$\mathbf{u}^k = |\mathbf{v}^k|^{q-1} \circ \text{sign}(\mathbf{v}^k) \quad (9)$$

$$\mathbf{a}^{k+1} = \frac{\mathbf{u}^k}{\|\mathbf{u}^k\|_p} \quad (10)$$

where $1/p + 1/q = 1$.

4) *Case 4:* When $p = \infty$, the solution is

$$\mathbf{a}^{k+1} = \text{sign}(\mathbf{v}^k). \quad (11)$$

B. Generalized Neighborhood Preserving Embedding Network

In this section, the structure of the proposed GNPENet is presented. The structure of GNPENet can be generally divided into three stages. The main components are the first two convolution stages. Then, an output stage containing hasing, histogram operations, and SPP is followed. Besides, the structure of GNPENet is shown in Fig. 1. The details of the structure are described in the following subsections.

1) *First Stage (GNPE Convolution): The Following Items Hold:*

a) *Take patches:* At this step, the overlapping patches are taken to extract the better features of the images. Assume that there are N input images with the size of $m \times n$ denoted by $\{\mathbf{I}_i\}_{i=1}^N$. Then, the patches' sizing $k_1 \times k_2$ is taken from the raw images around each pixel. Finally, the patches are vectorized and expressed as $\{\mathbf{x}_{i,j}\}_{j=1}^{\tilde{m}\tilde{n}}$, where $\tilde{m} = m - \lceil k_1/2 \rceil$, $\tilde{n} = n - \lceil k_2/2 \rceil$, and $\lceil x \rceil$ means the upward integer of x .

b) *Mean removal (MR):* It is a very important preprocess strategy. MR means subtracting the average gray value of the image from the gray value of each pixel in the image, thereby eliminating the common components of the image, enhancing the contrast of the image, and highlighting the details and textures of the image. MR is often used as a preprocessing step to provide better input for subsequent image analysis or processing.

The mean-removal operation can be expressed as $\bar{\mathbf{x}}_{i,j} = \mathbf{x}_{i,j} - (\mathbf{1}^T \mathbf{x}_{i,j} / k_1 k_2) \mathbf{1}$, where $\mathbf{1}$ represents an all-one vector with the same dimension as $\mathbf{x}_{i,j}$.

c) *Recombine the input image:* After the above operations, the input images of the model are recombined by the mean-removed patches. The process of recombining a single input image can be denoted as $\bar{\mathbf{X}}_i = [\bar{\mathbf{x}}_{i,1}, \bar{\mathbf{x}}_{i,2}, \dots, \bar{\mathbf{x}}_{i,\tilde{m}\tilde{n}}]$. Finally, the input data matrix is constructed with all recombined single images by the following formula:

$$\mathbf{X} = [\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \dots, \bar{\mathbf{X}}_N] \in \mathbb{R}^{k_1 k_2 \times N \tilde{m} \tilde{n}}. \quad (12)$$

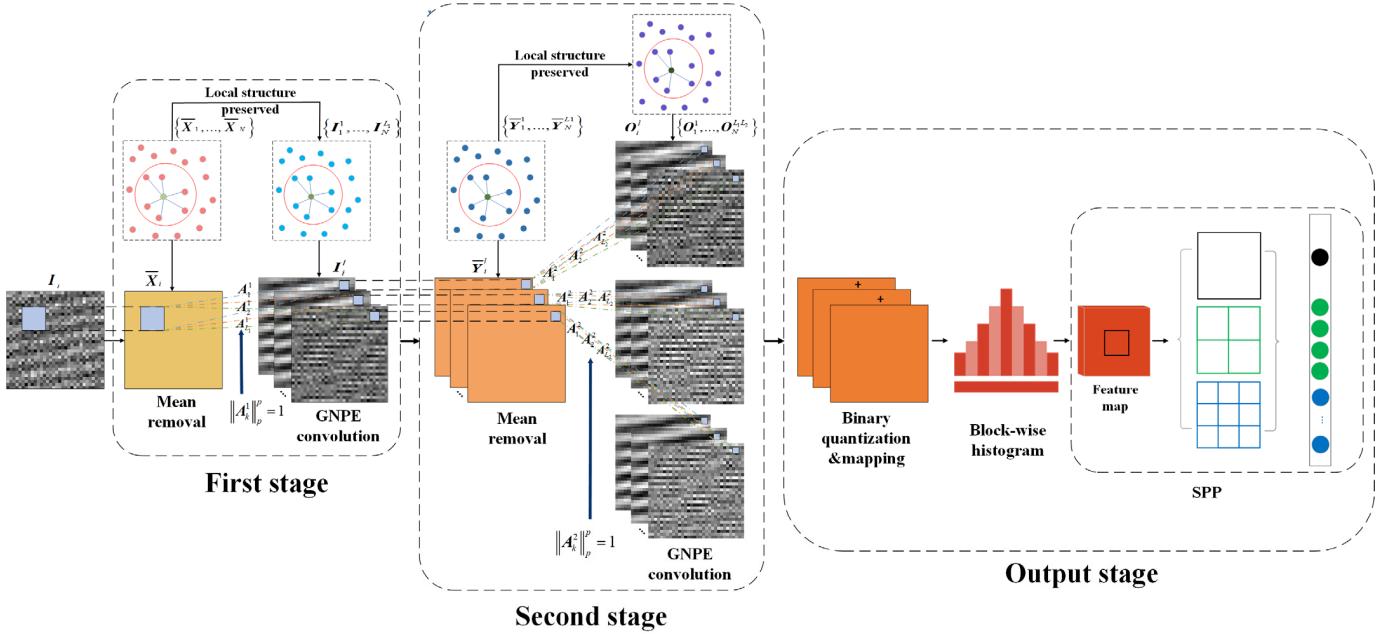


Fig. 1. Structure of GNPENet.

d) Obtain the convolutional filters by GNPE: This is the vital step that calculates the convolutional filters by the GNPE algorithm.

The filters are computed by the following formula:

$$A_l^1 = \text{mat}_{k_1, k_2}(\text{GNPE}_l(X)) \in \mathbb{R}^{k_1 \times k_2}, \quad l = 1, 2, \dots, L_1 \quad (13)$$

where $\text{mat}_{k_1, k_2}(v)$ is a map function that maps the $v \in \mathbb{R}^{k_1 k_2 \times 1}$ to a matrix $A \in \mathbb{R}^{k_1 \times k_2}$ and $\text{GNPE}_l(X)$ is the function that computes the first l mapping vectors ordered by the corresponding eigenvalue from smallest to largest.

e) Convolution operation: The input images are processed through convolution operation with the GNPE filters. The process formula is shown as follows:

$$I_i^l = I_i * A_l^1, \quad i = 1, 2, \dots, N \quad (14)$$

where the operation $*$ means the 2-D convolution and I_i^l is the result of the convolution operation. To satisfy the data format, I_i^l is supposed to obtain the same size of I_i . Thus, a zero-padding operation is processed to the matrix I_i .

2) Second Stage (GNPE Convolution): After the first GNPE convolution, $N \times L_1$ images are obtained. At this stage, another GNPE convolution layer is followed. The process of this stage is the same as the first GNPE convolution layer. The patches of I_i^l are first taken, and then, the MR process is followed. The recombined input image of the first convolution image is expressed as $\bar{Y}_i^l = [\bar{y}_{i,l,1}, \bar{y}_{i,l,2}, \dots, \bar{y}_{i,l,\tilde{m}\tilde{n}}] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$, where $\bar{y}_{i,l,j}$ is the j th mean-removed patch of I_i^l . The l th image result can be expressed as $Y^l = [\bar{Y}_1^l, \bar{Y}_2^l, \dots, \bar{Y}_N^l] \in \mathbb{R}^{k_1 k_2 \times N \tilde{m}\tilde{n}}$. Finally, the input matrix is denoted as

$$Y = [Y^1, Y^2, \dots, Y^{L_1}] \in \mathbb{R}^{k_1 k_2 \times L_1 N \tilde{m}\tilde{n}}. \quad (15)$$

A GNPE operation proceeds with the input matrix Y , and the details are given as follows.

The GNPE filters are obtained, and the formula can be written as

$$A_l^2 = \text{mat}_{k_1, k_2}(\text{GNPE}_l(Y)) \in \mathbb{R}^{k_1 \times k_2}, \quad l = 1, 2, \dots, L_2. \quad (16)$$

Notably, the corresponding eigenvalue e_l , which is computed by the initialization eigen decomposition of the covariance of the input data, is applied in the following quantization step.

Then, the final output images are obtained by a convolution operation:

$$O_i^l = \{I_i^l * A_l^2\}_{l=1}^{L_2}. \quad (17)$$

3) Output Stage (Hasing and Histograms): It is easy to compute that there are $N \times L_1 \times L_2$ images obtained after the convolution stage. At this stage, the images are all processed by hasing and histogram operations.

a) Hasing: The images are binarized first by $\{H(O_i^l)\}$, where $H(x)$ means is a Heaviside operation step (like) function, whose output is one if the input equals or is bigger than zero and zero otherwise. The binarized images of each convolution filter are accumulated. This operation converts I_i^l into a single integer-valued “image.” Besides, the weights are also needed for each different filter. The following formula describes the process:

$$T_i^l = \sum_{l=1}^{L_2} \tilde{e}_{l-1} H(O_i^l) \quad (18)$$

where \tilde{e} is the normalized e whose range is between 0 and 2^{L_2-1} .

It is reasonable to use the eigenvalues of the corresponding convolutional filters as the weight in the quantization stage. Because the eigenvalue reflects the importance of the corresponding eigenvector among all the eigenvectors to some

extent. Compared with the fixed weights in PCANet, eigenvalue weights can better reflect the importance of different convolutional kernels for more accurate feature extraction.

b) *Histograms*: The block-wise histograms are computed at this stage. To be specific, T_i^l is partitioned into B blocks with 2^{L_2} bins that can be overlapping or not. The histogram vectors are expressed as $Bhist(T_i^l)$. Finally, the feature of the original image I_i is constructed by concatenating all vectors

$$\mathbf{b}_i = \left[Bhist(T_i^l), \dots, Bhist(T_i^{L_1}) \right]^T \in \mathbb{R}^{(2^{L_2})L_1B}. \quad (19)$$

c) *SPP*: Finally, the obtained feature image by the above histogram is put into the SPP to further extract multiscale features

$$\mathbf{f}_i = SPP(\mathbf{b}_i). \quad (20)$$

SPP plays an important role in feature extraction, especially in the application scenario of fault diagnosis. The raw data in the fault diagnosis are a series of continual signals and the signal is split into fixed pieces. However, the size of fixed pieces may be not the optimal scale for feature extraction. Thus, the ability to extract features from multiscale can solve the problem to obtain more accurate representative features.

IV. EXPERIMENTS

In this section, we conduct experiments on five datasets belonging to three types from different aspects to evaluate the proposed GNPENet. To be specific, the rotor dataset is used to evaluate the performance under different filter numbers and sizes. The gear dataset is used to evaluate the performance under different hyperparameters of the GNPE algorithm. The bearing datasets are used to evaluate the generalization ability.

In addition, the accuracy is applied as the indicator to evaluate the performance of the algorithms in the following experiments. Furthermore, the definition of accuracy is described as

$$ACC = \frac{N^{CC}}{N}$$

where N is the total number of the classified subjects, N^{CC} is the total number of the correctly classified subjects, and ACC is the accuracy.

To comprehensively evaluate the proposed algorithm, we compared the proposed algorithm with five different methods. In detail, the original PCANet and other four state-of-the-art CNN models, ResNet (ResNet18) [16], ICNN [19], MS-CNN [20], and EVGG [21], are compared.

The algorithm with MATLAB R2023a runs on a PC with AMD Ryzen 7 5800X 8-Core Processor@3.80 GHz, 16-GB memory under the Windows 10 operation system.

A. Experimental Datasets

The rotor dataset is from Wuhan University [38]. It is captured from an experimental rotating machinery system, which consists of a rotor laboratory bench, a speed controller, a front-end processor, and a computer. Acquisition device is two eddy current sensors fixed on the sensor bracket. There are four kinds of states: normal, rubbing, unbalanced, and

misaligned. Besides, the rotor speed is set to 1200 r/min, and the frequency is set to 2048 Hz with a duration of 1 s. At the preprocessing stage, the samples are resampled to the 2-D image sizing 32×32 .

The gear dataset is provided by Pei Cao at the University of Connecticut [39]. The data are collected from a benchmark two-stage gearbox. A magnetic brake controlled by the input voltage supplies the torque. The vibration signals with the sampling frequency of 20 kHz are measured by an accelerometer and then recorded through a dSPACE system. In the experiment, the dataset is resampled into different sample sizes and tests the effect of it.

The first bearing dataset is from the Case Western Reserve University Bearing Data Center [40]. The experiment machine mainly consists of a 2-hp Reliance motor, a torque sensor, an electric dynamometer, and a control device. Furthermore, there are three fault diameters: 7", 14", and 21", and two collection frequencies: 12 000 samples/s and 48 000 samples/s. After our preprocessing, the original data were divided into seven states: normal, inner race (fault diameter 7"), outer race (fault diameter 7"), ball (fault diameter 7"), inner race (fault diameter 14"), outer race (fault diameter 14"), and ball (fault diameter 14"), which contains 118 samples for each state. In the experiment, the data are resampled to the size of 32×32 .

The second bearing dataset is the University of Ottawa bearing dataset [41]. The vibration signals of the data are collected from different health conditions under time-varying rotational speed conditions. There are three health conditions: health, fault with an inner race defect, and fault with an outer race defect. The vibration signals measured by the accelerometer are sampled at 20 kHz. Besides, the data are also resampled 2-D image sizing 32×32 .

The third bearing dataset was produced jointly by Mid Sweden University and Svenska Cellulosa Aktiebolaget [42]. The most distinctive feature of this dataset is that the raw data is not derived from standard laboratory test models but is actually collected in an industrial environment. Thus, it is possible to simulate the performance of the algorithm in a real industrial environment. The condition of each fault signal is different since they are from real industrial environments and the details can be viewed in [42]. In the experiment, we choose five different bearing data with the resampled size of 32×32 .

The details of the datasets are listed in Table I.

B. Hyperparameter Settings

In the experiment, PCANet and GNPENet are two stages, which means that they both have two convolution layers. The parameters are set to default as follows. The patch size of all stages is set to 7. The filter number of all stages is set to 8, which is inspired by the Gabor filters. The histogram block size is set to 7. The block overlapping rate is set to 0.5. For the GNPE algorithm, the optimal neighborhood parameter k is obtained by the adaptive algorithm.

Since the parameters p and s have a crucial impact on the performance of the algorithm, we refer to two aspects when selecting them. One aspect is mainly based on the test results

TABLE I
DETAILS OF THE DATASETS

Datasets	Abbreviation	Class Number	Sample Number/class	Sample Size	Environment	Purposes
Rotor (Wuhan University)	R	4	45	32×32	Laboratory	Test filter size and number
Gear (Connecticut University)	G	9	104	32×32	Laboratory	Test the hyper-parameters
Bearing (Case Western Reserve University)	B1	7	118	32×32	Laboratory	Train the generic filters
Bearing (Ottawa University)	B2	3	300	32×32	Laboratory	Test the generic filters
Bearing (Mid Sweden University)	B3	3	100	32×32	Industry	Test the generic filters & test the real industrial data

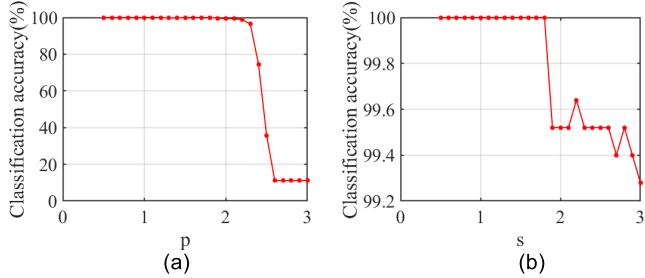


Fig. 2. Classification accuracy of GNPENet versus the (a) and (b) parameters p and s on dataset G.

in Fig. 2, and on the other hand, we also refer to the set values in the previous literature [36], [43]. Based on the above discussion, the parameters p and s are set to 0.7 and 1.5 as default in the next experiments, respectively.

Since hyperparameters p and s of the proposed GNPE algorithm play a significant role in the GNPENet, thus, we investigate the influence of the hyperparameters p and s on classification accuracy, represented by dataset G. The line graphs are shown as follows.

From the above line graphs in Fig. 2, it can be concluded that the optimal value of the parameter p tends to be smaller than 2 since the classification accuracy decreases sharply with the increase of p when p is larger than 2.2. Besides, when p gets to 2.6, the extracted features tend to be ineffective as the classification accuracy then stays at a very low level. For parameter s , when it increases, the classification accuracy also decreases slightly.

For traditional CNN (ResNet, ICNN, MS-CNN, and EVGG), the max epoch is set to 500, and the mini-batch size is set to 128.

Furthermore, the training sample size denotes as α , which is set to 10% as default. The rest of the data are used as the test set.

C. Data Preprocessing

Since the original data are a 1-D time series. To perform the convolution operation in the model more effectively, the original data information needs to be converted into a 2-D image-like format. In the experiments, the split-superposition resampling method is used for signal transformation. The main process is given as follows. First, the 1-D sample is divided into M $m \times n$ subsequences, and then, each subsequence is divided into m sequences with the length of n . Finally,

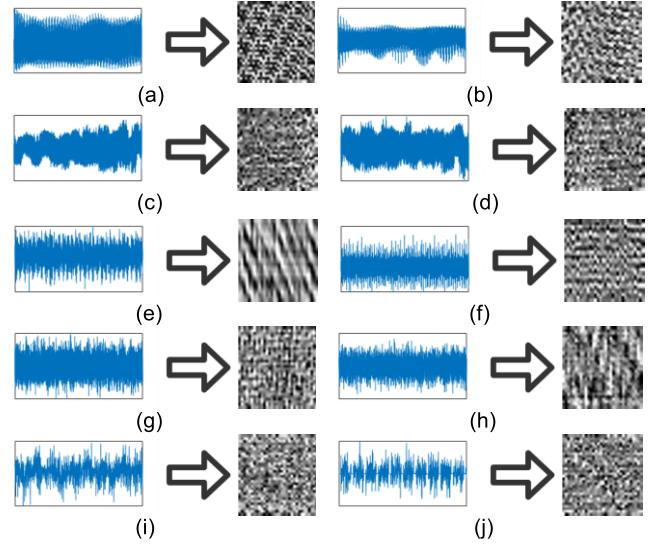


Fig. 3. Vibration signals before and after conversion. (a) R-1, (b) R-2, (c) G-1, (d) G-2, (e) B1-1, (f) B1-2, (g) B2-1, (h) B2-2, (i) B3-1, and (j) B3-2.

the corresponding 2-D samples are obtained by superposition. Some samples used in the experiments are shown in Fig. 3 and each subfigure represents a data type, e.g., R-1 represents the first type of data in the dataset R.

D. Experiment on Dataset R

Dataset R has four types of status: normal, rubbing, unbalanced, and misaligned, which are abbreviated as NM, RU, UB, and MA. The dataset is used to evaluate the performance of the model on the rotor dataset while detecting the influence of filter size and number on the performance of the algorithm.

First, the accuracies of the classification of the algorithms on dataset R are listed in Table II. To better evaluate the proposed algorithm and compare it with other algorithms, the training sample size α is increased from 5% to 70%. The accuracy of the algorithms on each type is given.

From Table II, we can see that GNPENet outperforms other compared algorithms a lot under a variety of conditions. From the perspective of training sample size, GNPENet obtains 100% accuracy on all fault types only with 30% of training samples. Furthermore, GNPENet achieves an accuracy of 95.23% with 5% training sample size, which is nearly 15% higher than the second-best algorithm, PCANet, while other traditional CNNs whose core are SGD and BP algorithms,

TABLE II
CLASSIFICATION ACCURACY OF DIFFERENT α VALUES (IN PERCENT) ON
DATASET R

Method	5	10	30	50	70
PCANet	79.76	93.75	100.00	100.00	100.00
ResNet	60.12	61.25	71.77	90.91	98.08
ICNN	77.98	77.50	99.19	100.00	100.00
MS-CNN	77.38	76.25	85.48	100.00	100.00
EVGG	50.60	58.75	83.87	93.18	96.15
GNPENet	95.23	98.75	100.00	100.00	100.00

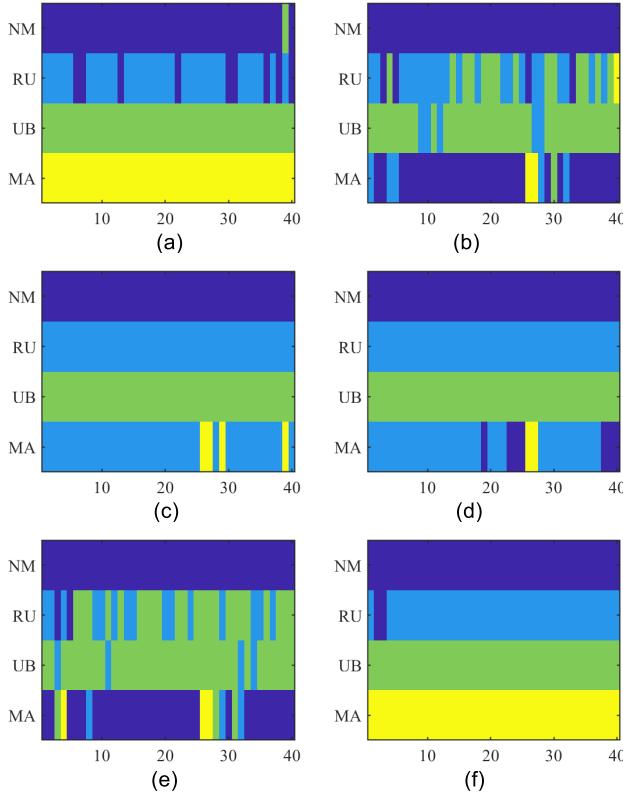


Fig. 4. Colormap on dataset R. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

such as ResNet and ICNN, tend to overfit under lower training sample sizes. Thus, their accuracies decline sharply with the reduction of training sample size. However, GNPENet still obtains higher accuracy with a lower training size. Above all, GNPENet has a higher feature extraction ability as it outperforms other algorithms under different training sample sizes.

To demonstrate the experiment results more clearly from an intuitionistic and comprehensive perspective, the colormaps are also given in Fig. 4. The horizontal coordinate of the colormaps is the number of samples and the vertical coordinate is the type of the sample. It is worth stating that in order to make the following results more differentiated, the figures in all the experiments below are under the condition of training sample size $\alpha = 10$.

As shown in Fig. 4, the results are consistent with Table II. GNPENet only misclassified a few samples of the RU and classified all other samples accurately, while other algorithms

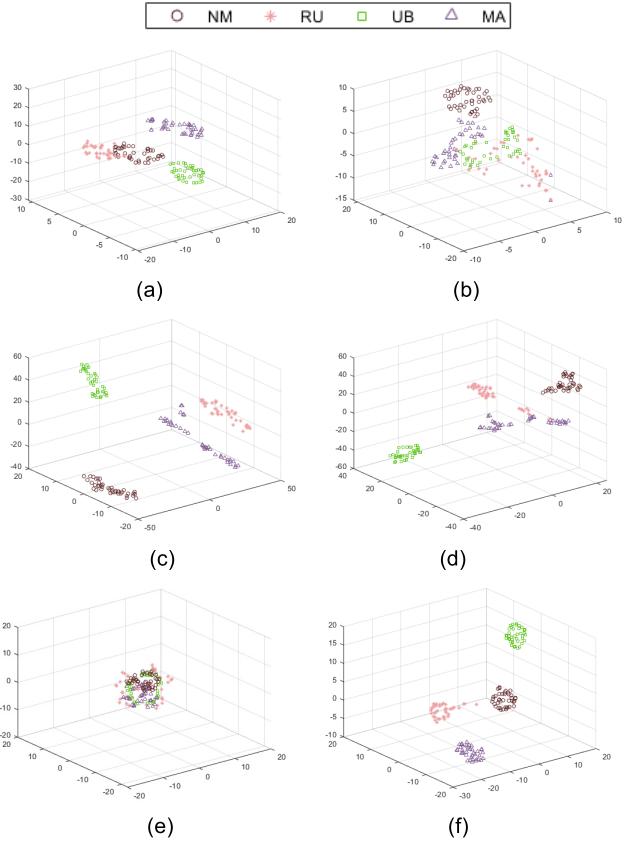


Fig. 5. Feature visualization on dataset R. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

did not perform so well in comparison. In detail, accurate classification of MA samples is always a hard task for traditional CNNs. ICNN and MS-CNN both classify MA samples as RU, and ResNet and EVGG both classify MA samples as NM.

To analyze the feature extraction ability better, the extracted features of the above six algorithms are projected into 3-D space with t-SNE [44] (with the default parameter setting). The results are shown in Fig. 5.

Fig. 5 shows the advantage of GNPENet over other methods in separating different class samples. In detail, GNPENet effectively clusters the data points into distinct classes with high similarity within each class and large differences between classes, thereby improving the classification performance. Only some RU samples are close to the area of fault NM, which also corroborates with the results of the colormap. For other algorithms, accurate classification of the samples of MA is a hard task. ICNN and MS-CNN almost classify all of them as RU. However, ResNet and EVGG both recognize them as the NM.

Finally, we evaluate the influence of the number and size of the filters on the performance of GNPENet. Specifically, the filter numbers of the two layers are taken from a range of 2–14 with a step of 2. The filter sizes are taken from the range of 3–15 with a step of 2. When the number of filters and size of one of these changes, the other one remains fixed as the default value. The 3-D bar figures are shown in Fig. 6.

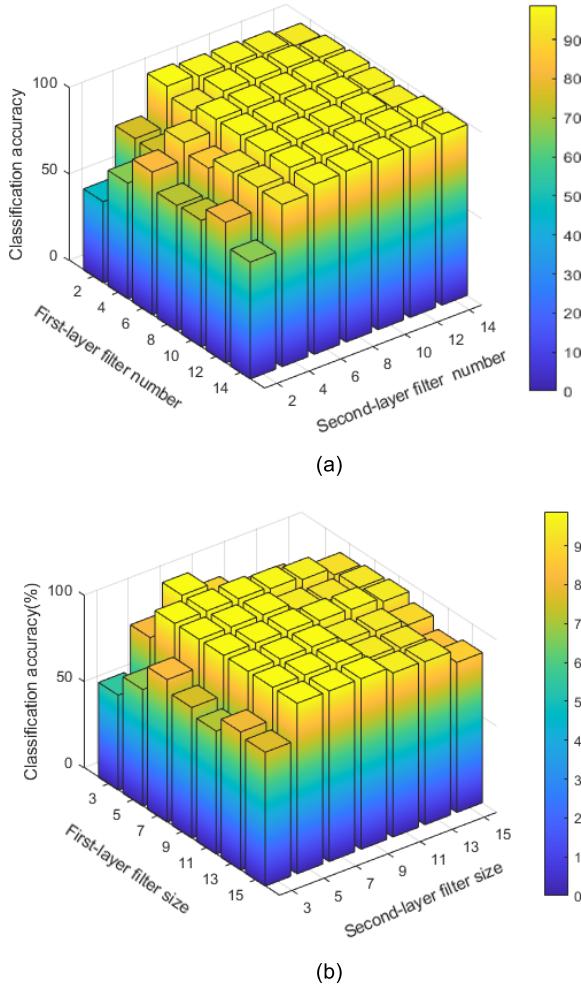


Fig. 6. Classification accuracy versus the filter number and filter size on dataset R. (a) Filter number. (b) Filter size.

According to Fig. 6, the accuracy versus the filter size and filter number is clear. From the perspective of the number of filters, the proposed GNPENet achieves a lower accuracy on the condition of a small size of filter number. In detail, when the numbers of first- and second-layer filters are lowest, the classification accuracy is also lowest. However, the accuracy increases steadily with the increases of the first-layer filter number, while accuracy increases sharply with the increases of the second-layer filter number. Thus, it can be concluded that the second-layer filters have greater usefulness, and the best second-layer number greater than 6. From the perspective of the size of the filters, the results are almost the same as the filter number, which means that the second-layer filter plays a more important role in the algorithm. Furthermore, the filter size of the first-layer filter tends to be greater than 5 since the accuracy gets higher with the increase of the first-layer filter size.

E. Experiment on Dataset G

Gears are key components in industrial machinery. Thus, they are also prone to be damaged due to their high-intensity working environment. In this section, we test the performance of the algorithms on the gear dataset G. There are nine

TABLE III
CLASSIFICATION ACCURACY OF DIFFERENT α VALUES
(IN PERCENT) ON DATASET G

Method	5	10	30	50	70
PCANet	96.25	99.16	99.85	100.00	100.00
ResNet	96.37	96.77	99.86	100.00	100.00
ICNN	81.74	96.53	100.00	99.78	100.00
MS-CNN	96.03	99.76	100.00	100.00	100.00
EVGG	91.04	97.84	99.85	99.79	99.64
GNPENet	99.65	100.00	100.00	100.00	100.00

statuses: healthy, missing, crack, spall, chip5a, chip4a, chip3a, chip2a, and chip1a, which are represented as HE, MI, CR, SP, C5, C4, C3, C2, and C1 in this section.

As in the rotor experiments, we first test the performance of the different methods under varying sample sizes. The training sample size α is also obtained from 5%, 10%, 30%, 50%, and 70%.

The classification accuracies are first presented in Table III to show the direct performance on dataset G.

According to Table III, it can be seen that GNPENet achieved very impressive results under different training sample sizes. Specifically, GNPENet still achieves 99.65% and 100% classification accuracies when the training sample size is relatively small, such as 5% or 10%. This achievement is 3.28% and 0.24% higher than the second-best method. In addition, with the training sample increase, GNPENet still achieves 100% classification accuracy, which is also still higher or equal to the compared CNNs. In other words, GNPENet is not only able to extract features accurately under small sample conditions but also able to perform better than traditional CNNs under large training sample sizes.

It can be observed in Fig. 7 that samples CR and HE are relatively easy to misidentify. In detail, ResNet incorrectly classifies some of the CR and C3 samples as C1 samples. As for ICNN, HE samples and M1 samples are often misclassified. Specifically, many HE samples were misclassified as C3 samples, while many HE samples were classified as C1 and C2 samples. The performance of EVGG, on the other hand, is worse than other methods. However, GNPENet also performs well on the gear datasets. Specifically, it can be easily concluded from the colormaps that GNPENet achieved 100% classification accuracy in all classes of samples.

According to the visualization graphs in Fig. 8, the visualization results also correspond to the colormaps. GNPENet classifies the different class samples into the corresponding regions, which is also corroborated by the 100% classification accuracy. For other CNNs, such as ResNet, a portion of the CR samples are too close to the C1 samples, causing them to be misclassified as C1.

F. Experiment on Dataset B1

The bearing dataset is a typical fault dataset that is often used as a benchmark to evaluate the algorithms. There are seven fault types in dataset B1: normal, inner race (fault diameter 7"), outer race (fault diameter 7"), ball (fault diameter

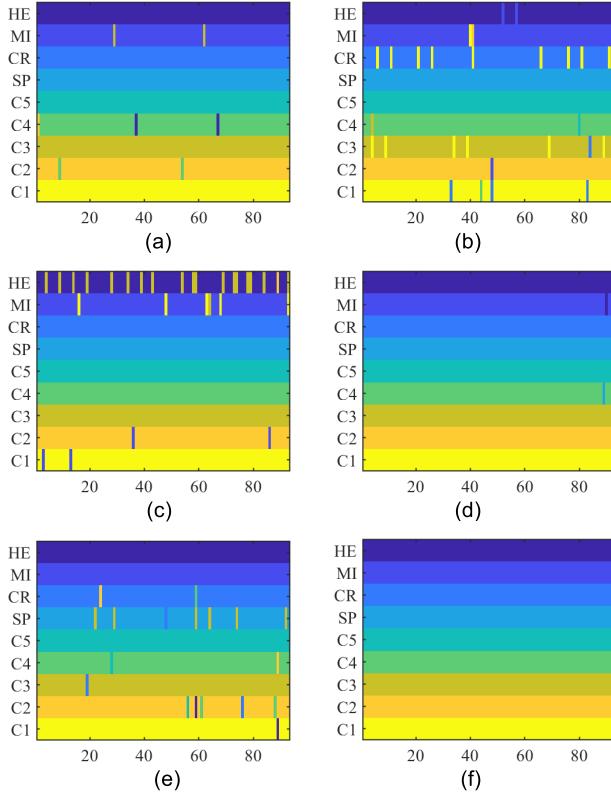


Fig. 7. Colormap on dataset G. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPNENet.

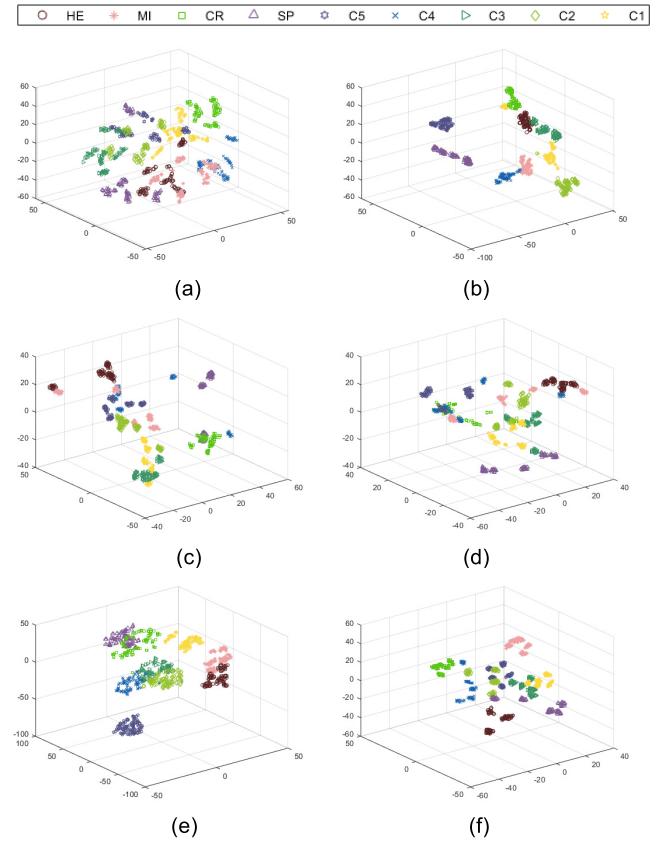


Fig. 8. Feature visualization on dataset G. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPNENet.

TABLE IV

CLASSIFICATION ACCURACY OF DIFFERENT α VALUES
(IN PERCENT) ON DATASET B1

Method	5	10	30	50	70
PCANet	82.65	96.09	96.68	95.88	98.77
ResNet	70.81	87.46	95.12	99.27	100.00
ICNN	73.72	77.49	85.36	95.15	99.59
MS-CNN	59.43	72.50	86.75	95.15	99.18
EVGG	43.36	77.08	83.62	71.42	70.61
GNPNENet	96.68	97.70	99.65	100.00	100.00

7”), inner race (fault diameter 14”), outer race (fault diameter 14”), and ball (fault diameter 14”), which are abbreviated as NM, IR1, OR1, BA1, IR2, OR2, and BA2. We use the dataset to train a set of generic filters to test the generalization ability of the proposed GNPNENet. In particular, we first performed the same experiments as above on this dataset to test the performance of the algorithm on the rotor dataset. Then, all the samples are used to train generic filters, which are applied to the next two bearing datasets to test the generalization ability.

From Table IV and Fig. 9, we can conclude that GNPNENet still shows its superiority over other algorithms. Especially, on the condition of small training sample sizes, GNPNENet achieves remarkable results, i.e., GNPNENet obtains 96.68% accuracy with just 5% training samples. Thus, GNPNENet excels in the case of small samples. Furthermore, GNPNENet also performs well on normal-sized training sets as it also obtains 100% accuracy with 50% and 70% training sample

sizes. From the perspective of fault types, NM samples are easier to be recognized, and it is successfully recognized by all methods. IR1 and OR1, however, are more difficult to recognize, and CNNs generally have lower classification accuracies on these two fault classes. In contrast, GNPNENet, which recognizes all types of samples well, misclassifies only a small number of samples.

Similarly, the results are also projected to the 3-D space to visualize to extracted features.

The results of the visualization figures in Fig. 10 also confirm the classification accuracy of each type in the colormaps. In detail, ResNet, which performs well among CNNs, clusters samples of the same type well. However, it does not separate OR1 and OR2 samples far enough, which results in some OR2 samples being classified as OR1 samples, and this result is displayed in the colormap. As for ICNN and MS-CNN, the NM samples are easily recognized and classified into separate regions, but the samples of other types are not well separated, which leads to their poor performance on other types. For GNPNENet, the samples of the same classes are well clustered in a region together, and there are proper distances between different fault regions. Only a small number of samples break away from their own regions, which is also consistent with the colormap.

To evaluate the generalization ability of the algorithm, the filters learned from dataset B1 are applied to the other two rotor datasets. The filters learned are shown in Fig. 11. As for

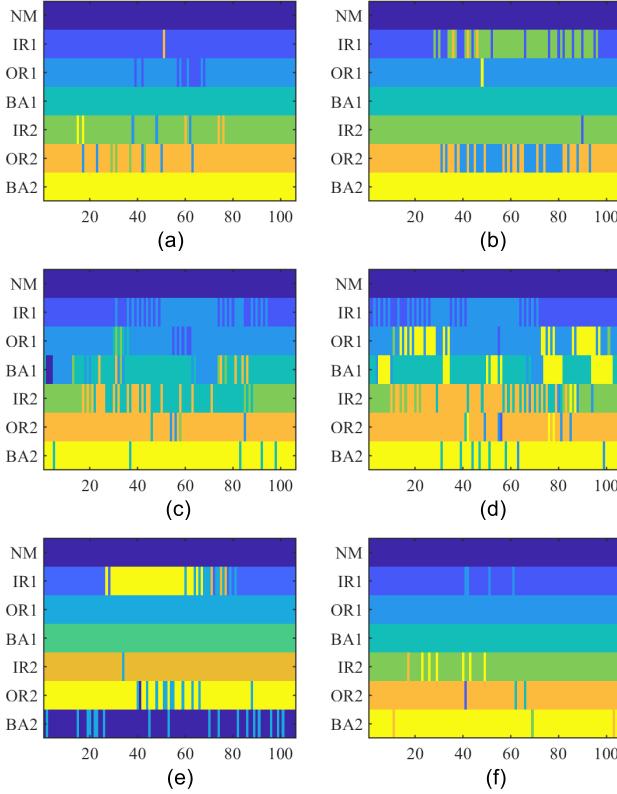


Fig. 9. Colormap on dataset B1. (a) PCANet. (b) ResNet. (c) ICNN. (e) MS-CNN. (e) EVGG. (f) GNPENet.

other algorithms, we also use a similar method. Since smaller p -values make the projection matrix sparser, it also makes it more specialized for a single dataset during feature extraction. In order to increase the generalization performance, we took a suitable value of 2.1 to learn the convolutional filters so that the learned convolutional filters can achieve better performance on other datasets as well.

To compare it with other algorithms more fairly, we conduct the following experiments with a similar method. In detail, for the other four traditional CNNs, ResNet, ICNN, MS-CNN, and EVGG, the transfer learning strategy is employed. In particular, the parameters are frozen except for the FC layer and the classification layer. Then, the part-frozen models are retrained with the datasets B2 and B3 in the next experiments.

G. Experiment on Dataset B2

There are three sample types: healthy, inner race, outer race. They are abbreviated as HE, IR, and OR for a better description. The dataset B2 is also used for testing the generalization ability of GNPENet. The classification accuracies of the algorithms on dataset B2 are listed in Table V.

From Table V, it can be concluded that GNPENet achieves excellent results, achieving 100% classification accuracy at all different training sample sizes. With a training sample size of 5%, GNPENet outperforms the other methods (PCANet, ResNet, ICNN, MS-CNN, and EVGG) by 19.19%, 12.99%, 39.42%, 36.14%, and 7.14%, respectively. Meanwhile, GNPENet consistently maintains 100% classification accuracy as the training sample size increases, while the other

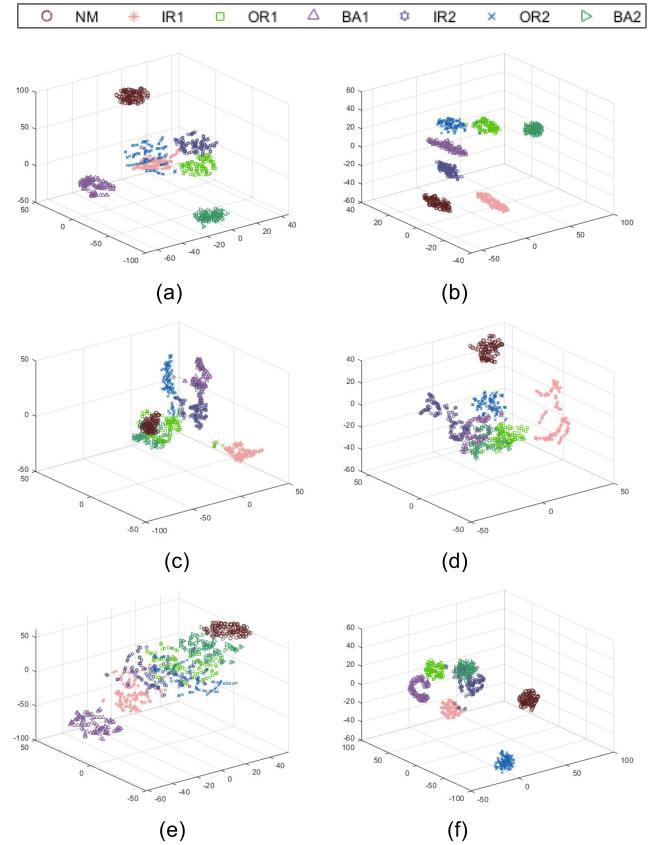


Fig. 10. Feature visualization on dataset B1. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

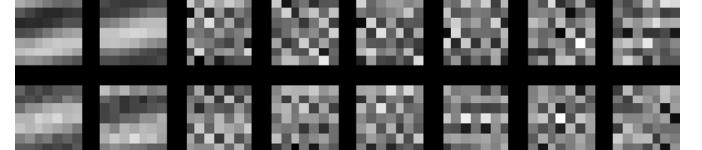


Fig. 11. Filters learned from dataset B1.

TABLE V
CLASSIFICATION ACCURACY OF DIFFERENT α VALUES
(IN PERCENT) ON DATASET B2

Method	5	10	30	50	70
PCANet	80.81	80.98	96.19	98.66	100.00
ResNet	87.01	84.19	93.96	95.55	100.00
ICNN	60.58	88.40	90.63	91.33	91.11
MS-CNN	63.86	71.60	82.06	89.33	89.63
EVGG	92.86	96.66	95.39	98.88	100.00
GNPENet	100.00	100.00	100.00	100.00	100.00

methods obtain 100% classification accuracy only at 70%. The results further confirm the generalization ability of the proposed GNPENet as it achieves such a high classification accuracy with the filters learned from another same-type dataset. It is worth noting that ICNN and MS-CNN using frequency-domain information perform considerably worse than the other methods. This may be due to the fact that dataset B2 is a bit less similar to B1 in the frequency domain. As a result, after extracting the frequency-domain information, the

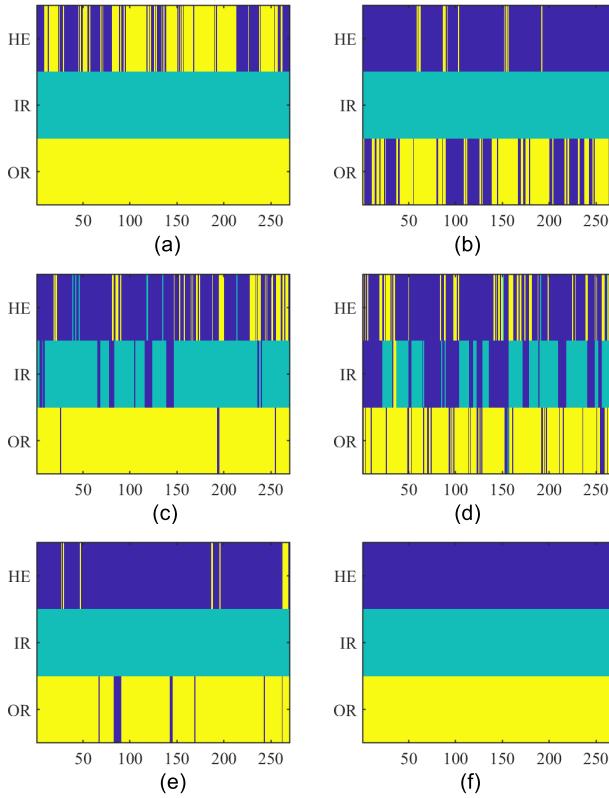


Fig. 12. Colormap on dataset B2. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

classification accuracies all lag behind the other time-domain methods.

According to colormaps in Fig. 12, with a training sample size of 10%, GNPENet achieves 100% classification accuracy. However, all other methods misclassify some samples. From the point of view of types, IR is the easiest to recognize, so all methods except ICNN and MS-CNN achieved 100% accuracy, whereas the other two classes are more similar due to their similarity, and hence, other methods, such as PCANet, ResNet, and EVGG, misclassified some of their samples.

As can be seen from the visualization graphs in Fig. 13, the distribution of data points is consistent with the above colormaps well. In detail, for PCANet, the samples of faults HE and CR are close to some extent, which leads to misclassifications. For ResNet and EVGG, the IR samples, which are all correctly classified, are located in an independent area. For ICNN and MS-CNN, they perform poorly because the sample points of different classes are mixed together and cannot be classified very accurately. However, the sample distribution of the visualization graph of GNPENet is excellent since it separates samples of different classes far enough while clustering same-class samples close together.

H. Experiment on Dataset B3

Dataset B3 contains five fault types: outer race 1, inner race 1, ball, inner race 2, and outer race 2, which are abbreviated as OR1, IR1, BA, IR2, and OR2, respectively. Since dataset B3 is collected from real industrial machines, which is meaningful to test the real performance of the algorithms,

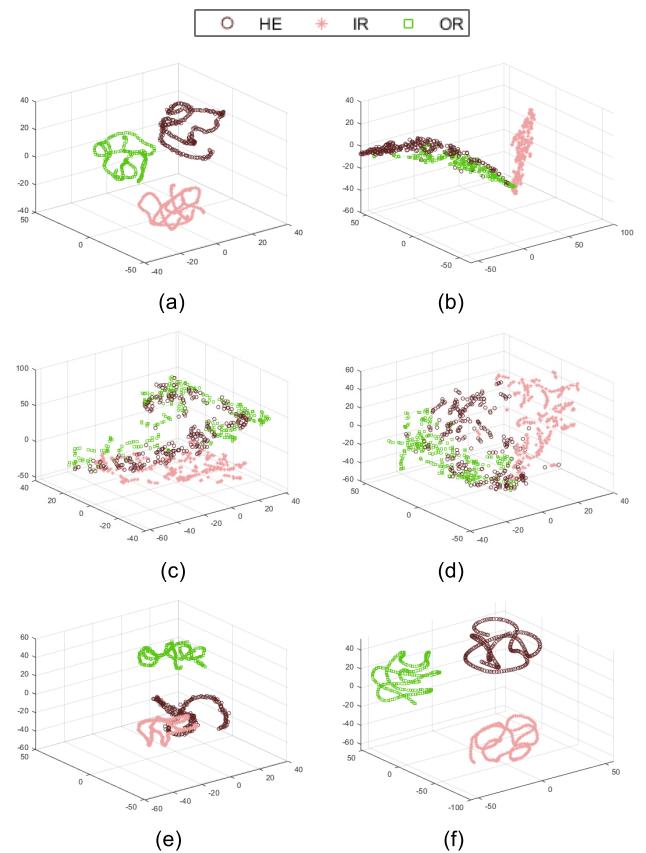


Fig. 13. Feature visualization on dataset B2. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

TABLE VI
CLASSIFICATION ACCURACY OF DIFFERENT α VALUES
(IN PERCENT) ON DATASET B3

Method	5	10	30	50	70
PCANet	73.47	75.56	83.43	96.00	94.67
ResNet	61.89	59.11	85.43	88.00	87.33
ICNN	77.26	79.11	83.14	96.40	98.00
MS-CNN	57.68	67.11	88.86	96.80	98.67
EVGG	61.47	60.00	78.57	81.60	83.33
GNPENet	80.42	82.89	91.43	98.40	100.00

we conduct the experiments from two aspects, that is, the filters learned from dataset B1 are first applied to dataset B3 to test the generalization ability of the methods from the standard dataset to the real dataset. Then, the experiments using dataset B3 alone for training and testing were also performed.

It can be concluded from Table VI that the performance of various methods in real industrial scenarios is somewhat degraded compared to the standard dataset. However, with the gradual increase of the training sample size, the performance of the methods reaches a high level. The performance of GNPENet is basically the same as in the standard dataset, i.e., the performance is substantially ahead of other methods, especially when the training sample size is relatively small. For example, at a training sample size of 5%, GNPENet achieves an accuracy of 80.42%, while most of the other traditional CNN methods are around 50%–60%. It is worth

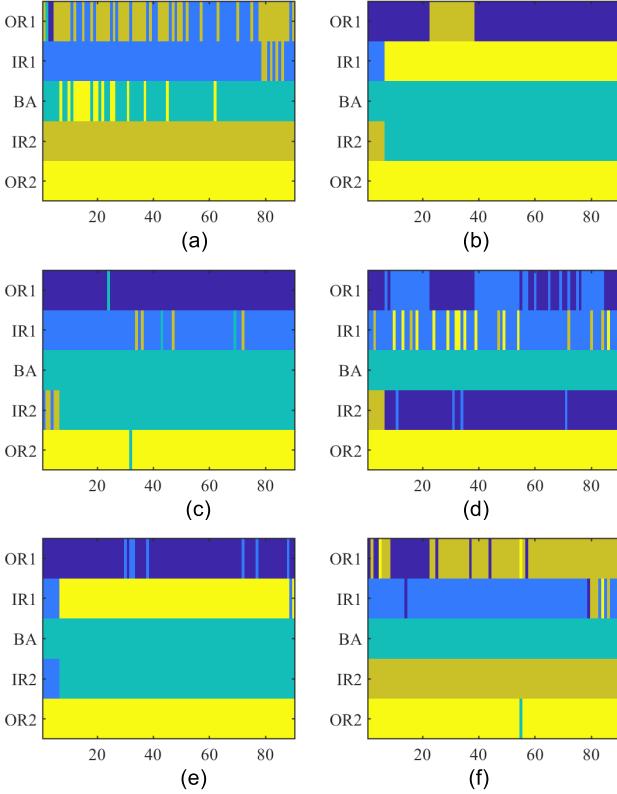


Fig. 14. Colormap on dataset B3. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

noting that both the convolutional filters and the model here are learned from B1, thus also demonstrating the generalization performance of GNPENet.

To better analyze the result from the perspective of the digit, the colormaps of the results are also obtained to show which specific samples in each type were misclassified. The colormaps are shown in Fig. 14.

From the colormaps in Fig. 14, it can be concluded that OR2 samples are easily recognized by all the algorithms. However, accurate classification of IR2 samples seems a hard task for all traditional CNNs, i.e., ResNet, ICNN, and EVGG misclassified almost all IR2 samples as BA, and MS-CNN misclassified the IR2 samples as OR1 samples. However, GNPENet recognized almost all OR2 samples except one. Overall, GNPENet only made errors on the OR1 sample, while almost all types of samples were accurately classified. Other methods, on the other hand, made a large number of errors on samples of multiple classes, resulting in lower classification accuracies.

Then, the visualization graphs are also shown in Fig. 15. According to the visualization graphs in Fig. 15, the visualization graph also reflects the results of the identification of specific fault types in the colormaps figures. For example, ResNet and EVGG both identify the OR1 samples accurately while failing to recognize the IR1 and IR2 samples. Thus, in the visualization graph, the IR1 and OR2 samples are quite close to each other, the BA and IR2 samples are likewise very close to each other, and the OR1 samples are in a more distant region.

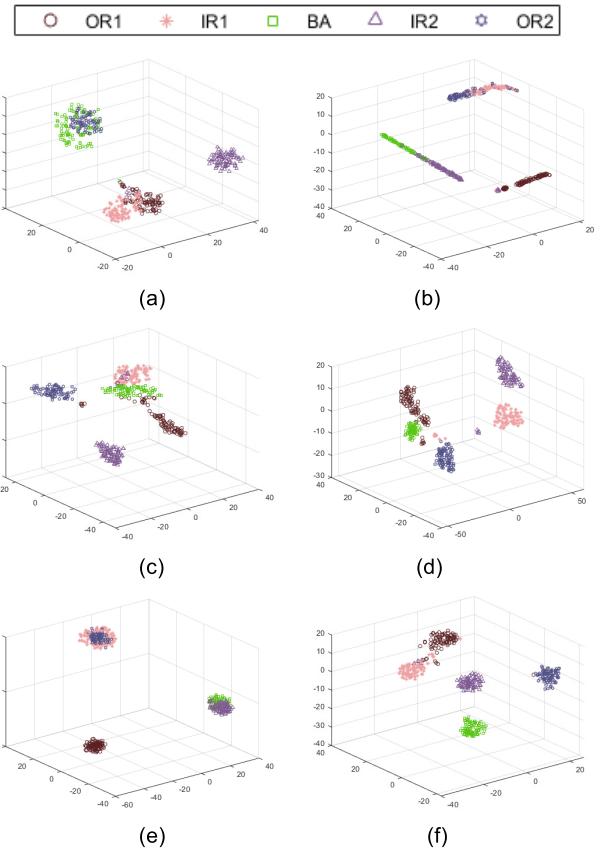


Fig. 15. Feature visualization on dataset B3. (a) PCANet. (b) ResNet. (c) ICNN. (d) MS-CNN. (e) EVGG. (f) GNPENet.

TABLE VII
CLASSIFICATION ACCURACY OF DIFFERENT α VALUES (IN PERCENT)
ON DATASET B3 WITH B3 FILTERS

Method	5	10	30	50	70
PCANet	77.68	75.60	80.27	91.15	92.17
ResNet	60.42	63.56	96.88	100.00	100.00
ICNN	69.89	69.11	81.71	99.20	100.00
MS-CNN	47.58	63.11	88.00	100.00	100.00
EVGG	49.89	57.78	82.00	84.00	94.00
GNPENet	82.53	89.78	97.33	100.00	100.00

As we described above, since this dataset is collected from real industrial environments, it can reflect the performance of the method in real environments. Therefore, we also tested directly on this dataset and the experimental results are listed in Table VII.

The performance of the various approaches varies, as shown in Table VII. In the small sample scenario, the classic CNN algorithms perform worse than the transfer case, whereas, with large samples, the accuracy improves quite a bit. This can be attributed to the fact that the model in the transfer case is pretrained with abundant samples of B1 datasets and hence works better. GNPENet, on the other hand, has some degree of improvement in both small and large training sample sizes, which can be attributed to the fact that GNPENet has a strong feature extraction ability in small samples. Hence, GNPENet performs better even when a small amount of data is used and no transfer is performed.

TABLE VIII

CLASSIFICATION ACCURACY OF DIFFERENT α VALUES (IN PERCENT) ON DATASET B1 WITH DIFFERENT COMPONENTS

MR	SPP	5	10	30	50
Y	Y	81.12	88.01	92.68	99.52
		90.94	94.74	97.21	99.76
Y	Y	94.26	95.55	99.48	100.00
		96.68	97.70	99.65	100.00

I. Ablation Study

In order to gain more insight into the role of each part of the proposed model, we perform an ablation study in this section. We test the role played by MR and SPP on the whole network. In the GNPENet, MR is added before the process of learning the filters. This operation allows the convolutional filters learned by subsequent GNPE algorithms to be more representative and freer from background noise. SPP, which possesses multiscale feature extraction capability, theoretically has a significant role.

This experiment was performed on dataset B1 with different training sample sizes. The results of the experiment are displayed in Table VIII.

From Table VIII, it can be easily seen that both MR and SPP components boost the original network to some extent, and this boost is more pronounced when the training sample number α is low. Specifically, MR improves by 9.82%, while SPP improves by 13.14% when the training sample size is 5%. The combination of the two in turn leads to a certain degree of improvement in the network accuracy. Thus, we can conclude that these two parts are improving the overall performance of the network.

J. Analysis

From the present study, it can be concluded as follows.

1) GNPENet performs better than other algorithms (i.e., the original PCANet and the traditional CNNs, ResNet, ICNN, MS-CNN, and EVGG). For example, GNPENet outperforms the above algorithms by 5%, 37.5%, 21.25%, 22.5%, and 40% on dataset R.

2) The classification accuracy of GNPENet is maintained at a high level when the number and size of convolutional filters are varied, demonstrating its high robustness.

3) GNPENet exhibits a much better generalization ability than other algorithms, as shown by a comparison with experiments on the datasets B1–B3. In detail, compared with other algorithms, the accuracy of GNPENet on datasets B2 and B3 is 3.34%–28.4% and 3.78%–22.78% higher than other algorithms with 10% training sample size.

4) The proposed algorithm is applicable to feature extraction of various industrial data. For example, the three classical types of datasets in the standard laboratory or real-world industrial environments, rotor, gear, and bearing, are applied to prove the effectiveness of the GNPENet.

V. CONCLUSION

In this article, a novel CNN with local structure called GNPENet is proposed for fault diagnosis. The main idea

of GNPENet is to use a novel GNPE method to learn the convolutional filters of CNN. The proposed GNPENet has some superiority as follows.

1) GNPE can extract the local structure information of data compared with PCA.

2) GNPE is based on the L_p -norm, which is insensitive to noise and outliers compared with the conventional L_2 -norm. Due to the flexibility of the p -value in the L_p -norm, GNPENet is applicable to different scenarios by adjusting the p -value.

3) The SGD and BP algorithms of traditional convolutional networks consume much computation and time, and the GNPE algorithm of GNPENet makes the learning of convolutional filters simpler and more efficient.

We conduct fault diagnosis experiments on five different datasets of three types and comprehensively evaluate the proposed GNPENet method. In the experiments, GNPENet is compared with many fault diagnosis methods, including PCANet and the other four state-of-the-art CNN methods: ResNet, ICNN, MS-CNN, and EVGG. The experimental results demonstrate that GNPENet has superior feature extraction and generalization capabilities.

There are still some works about the algorithm that can be further explored. For example, other manifold algorithms (i.e., locality preserving projections (LPP) [45], isometric projection (IsoP) [46], and linear local tangent space alignment (LLTSA) [47]) can also be introduced to this model to improve the performance. Furthermore, other convolution-based models (i.e., convolutional vision transformer [48]) are also supposed to be combined with the manifold algorithms.

APPENDIX OPTIMIZATION OF GENERALIZED NEIGHBORHOOD PRESERVING EMBEDDING

Considering the convexity of the constraint set, which depends on the p -value, the optimization problem is divided into two cases, similar to GPCA [43] and 2D-GPCA [36].

1) *Case 1* ($p \geq 1$): As the constraint is convex when $p \geq 1$, thus, the optimization objective can be expressed as follows:

$$\max_{\mathbf{a}} \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s, \quad s.t. \|\mathbf{a}\|_p^p = 1 \quad (21)$$

where $s \geq 1$, $p \geq 1$, $\mathbf{a} \in \mathbb{R}^a$. With the MM framework [36], [37], the optimal solution can be obtained iteratively. In detail, let k be the iteration number and \mathbf{a}^k be the projection vector of the k th step. For \mathbf{a} , \mathbf{a}^k can be regarded as a constant value. Thus, (21) can be linearized as the following equation:

$$\begin{aligned} \sum_i^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s &\geq s \sum_{i=1}^N \left[|\tilde{\mathbf{X}}_i \mathbf{a}^k|^{s-1} \circ \text{sign}(\tilde{\mathbf{X}}_i \mathbf{a}^k) \right]^T \tilde{\mathbf{X}}_i \mathbf{a} \\ &+ (1-s) \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}^k\|_s^s. \end{aligned} \quad (22)$$

For a more convenient representation given in the following, we denote the objective function as $f(\mathbf{a})$ and the constraints as $g(\mathbf{a}|\mathbf{a}^k)$, i.e.,

$$f(\mathbf{a}) = \sum_i^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s \quad (23)$$

and

$$\begin{aligned} g(\mathbf{a}|\mathbf{a}^k) &= s \sum_{i=1}^N \left[|\tilde{\mathbf{X}}_i \mathbf{a}^k|^{s-1} \circ \text{sign}(\tilde{\mathbf{X}}_i \mathbf{a}^k) \right]^T \tilde{\mathbf{X}}_i \mathbf{a} \\ &\quad + (1-s) \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}^k\|_s^s. \end{aligned} \quad (24)$$

With the MM framework, $g(\mathbf{a}|\mathbf{a}^k)$ can be regarded as the surrogate function of $f(x)$. Then, by dropping the term irrelevant to \mathbf{a} , the optimal solution to the problem can be converted as follows:

$$\mathbf{a}^{k+1} = \arg \max_{\mathbf{a}} (\mathbf{v}^k)^T \mathbf{a}, \quad \text{s.t. } \|\mathbf{a}\|_p^p = 1 \quad (25)$$

where $\mathbf{v}^k = \sum_{i=1}^N \tilde{\mathbf{X}}_i^T [|\tilde{\mathbf{X}}_i \mathbf{a}^k|^{s-1} \circ \text{sign}(\tilde{\mathbf{X}}_i \mathbf{a}^k)]$, which has removed the item that it is not related to \mathbf{a} .

Then, the solution to the problem can be expressed as

$$\mathbf{a}^{k+1} = \frac{|\mathbf{v}^k|^{q-1} \circ \text{sign}(\mathbf{v}^k)}{\|\mathbf{v}^k\|_q^{q-1}} \quad (26)$$

where $1/p + 1/q = 1$. Then, (26) can be split into two stages

$$\mathbf{u}^k = |\mathbf{v}^k|^{q-1} \circ \text{sign}(\mathbf{v}^k) \quad (27)$$

$$\mathbf{a}^{k+1} = \frac{\mathbf{u}^k}{\|\mathbf{u}^k\|_p}. \quad (28)$$

Furthermore, two special cases, $p = 1$ and $p = \infty$, should be treated separately, as it is easy to conduct that $q = \infty$ when $p = 1$ with the constraint $1/p + 1/q = 1$. Thus, the solution can be rewritten as

$$a_i^{k+1} = \begin{cases} \text{sign}(v_j^k), & i = j \\ 0, & i \neq j \end{cases} \quad (29)$$

where $j = \arg \max |v_i^k|$ and $i \in [1, w]$, i.e., $|v_j^k|$ is the largest value of $|v_i^k|$.

When $p = \infty$, the solution can be written as follows:

$$\mathbf{a}^{k+1} = \text{sign}(\mathbf{v}^k). \quad (30)$$

2) *Case 2 ($0 < p < 1$):* The constraint set is nonconvex in such a case. Thus, the optimization problem can be converted as follows by applying the Lagrange multiplier method:

$$\max_{\mathbf{a}} \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s - \lambda(\|\mathbf{a}\|_p^p - 1) \quad (31)$$

where $s \geq 1$, $0 < p < 1$, $\lambda > 0$, and $\mathbf{a} \in \mathbb{R}^a$. By performing such a conversion, problems that were not easily solved can be solved by thinking along the same lines as case 1. As before, \mathbf{a}^k is defined as the projection vector of the k th step. The elements of \mathbf{a}^k are replaced with ε if they are 0, where ε is a random value close to 0. According to (31), it can be concluded that

$$\begin{aligned} &\sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s - \lambda(\|\mathbf{a}\|_p^p - 1) \\ &\geq s(\mathbf{v}^k)^T \mathbf{a} + (1-s) \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s \end{aligned}$$

$$- \lambda \frac{p}{2} \mathbf{a}^T \text{diag}(|\mathbf{a}|^{p-2}) \mathbf{a} - \lambda(1 - \frac{p}{2}) \|\mathbf{a}\|_p^p + \lambda. \quad (32)$$

Similarly, the objective function and the constraint are defined as $f(\mathbf{a})$ and $g(\mathbf{a}|\mathbf{a}^k)$, i.e.,

$$f(\mathbf{a}) = \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s - \lambda(\|\mathbf{a}\|_p^p - 1) \quad (33)$$

and

$$\begin{aligned} g(\mathbf{a}|\mathbf{a}^k) &= s(\mathbf{v}^k)^T \mathbf{a} + (1-s) \sum_{i=1}^N \|\tilde{\mathbf{X}}_i \mathbf{a}\|_s^s \\ &\quad - \lambda \frac{p}{2} \mathbf{a}^T \text{diag}(|\mathbf{a}|^{p-2}) \mathbf{a} - \lambda(1 - \frac{p}{2}) \|\mathbf{a}\|_p^p + \lambda. \end{aligned} \quad (34)$$

The optimal solution to the problem can be obtained by solving an iteratively maximizing problem, which can be denoted as follows:

$$\mathbf{a}^{k+1} = \arg \max_{\mathbf{a}} s(\mathbf{v}^k)^T \mathbf{a} - \lambda \frac{p}{2} \mathbf{a}^T \text{diag}(|\mathbf{a}|^{p-2}) \mathbf{a}. \quad (35)$$

It is easy to derive its results as

$$\mathbf{a}^{k+1} = \frac{s}{\lambda p} |\mathbf{a}^k|^{2-p} \circ \mathbf{v}^k. \quad (36)$$

Based on the above solution, the final result can be split into two stages

$$\mathbf{u}^k = |\mathbf{a}^k|^{2-p} \circ \mathbf{v}^k \quad (37)$$

$$\mathbf{a}^{k+1} = \frac{\mathbf{u}^k}{\|\mathbf{u}^k\|_p}. \quad (38)$$

REFERENCES

- [1] Z. Chen and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1693–1702, Jul. 2017.
- [2] Z. Yang, B. Xu, W. Luo, and F. Chen, "Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review," *Measurement*, vol. 189, Feb. 2022, Art. no. 110460.
- [3] W. Deng, Z. Li, X. Li, H. Chen, and H. Zhao, "Compound fault diagnosis using optimized MCKD and sparse representation for rolling bearings," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–9, 2022.
- [4] X. Wang, D. Mao, and X. Li, "Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network," *Measurement*, vol. 173, Mar. 2021, Art. no. 108518.
- [5] Y. Li, X. Zhang, Z. Chen, Y. Yang, C. Geng, and M. J. Zuo, "Time-frequency ridge estimation: An effective tool for gear and bearing fault diagnosis at time-varying speeds," *Mech. Syst. Signal Process.*, vol. 189, Apr. 2023, Art. no. 110108.
- [6] H. Zhao, H. Liu, Y. Jin, X. Dang, and W. Deng, "Feature extraction for data-driven remaining useful life prediction of rolling bearings," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.
- [7] D. Zhao et al., "Enhanced data-driven fault diagnosis for machines with small and unbalanced data based on variational auto-encoder," *Meas. Sci. Technol.*, vol. 31, no. 3, Mar. 2020, Art. no. 035004.
- [8] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, Jan. 1991.
- [9] M. H. R. A. Syafi'i, E. Prasetyono, M. K. Khafidli, D. O. Anggriawan, and A. Tjahjono, "Real time series DC arc fault detection based on fast Fourier transform," in *Proc. Int. Electron. Symp. Eng. Technol. Appl. (IES-ETA)*, Oct. 2018, pp. 25–30.
- [10] H. Ali, A. S. Maulud, H. Zabiri, M. Nawaz, H. Suleman, and S. A. A. Taqvi, "Multiscale principal component analysis-signed directed graph based process monitoring and fault diagnosis," *ACS Omega*, vol. 7, no. 11, pp. 9496–9512, Mar. 2022.

- [11] K. Zhang, B. Tang, L. Deng, and X. Yu, "Fault detection of wind turbines by subspace reconstruction-based robust kernel principal component analysis," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [12] H. Cao, P. Sun, and L. Zhao, "PCA-SVM method with sliding window for online fault diagnosis of a small pressurized water reactor," *Ann. Nucl. Energy*, vol. 171, Jun. 2022, Art. no. 109036.
- [13] L. Yichen, T. Zhenhao, and C. Shengxian, "A multi-domain feature fusion method for wind turbine bearing fault diagnosis," in *Proc. IEEE Int. Conf. Power Sci. Technol. (ICPST)*, May 2023, pp. 86–91.
- [14] J. Burriel-Valencia, R. Puche-Panadero, J. Martinez-Roman, A. Sapena-Bano, and M. Pineda-Sanchez, "Short-frequency Fourier transform for fault diagnosis of induction machines working in transient regime," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 3, pp. 432–440, Mar. 2017.
- [15] D. Zhao, J. Li, W. Cheng, and W. Wen, "Bearing multi-fault diagnosis with iterative generalized demodulation guided by enhanced rotational frequency matching under time-varying speed conditions," *ISA Trans.*, vol. 133, pp. 518–528, Feb. 2023.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [17] P. Liang, W. Wang, X. Yuan, S. Liu, L. Zhang, and Y. Cheng, "Intelligent fault diagnosis of rolling bearing based on wavelet transform and improved ResNet under noisy labels and environment," *Eng. Appl. Artif. Intell.*, vol. 115, Oct. 2022, Art. no. 105269.
- [18] X. Wei, X. Mu, T. Jiang, W. Liu, and Z. Zeng, "Fault diagnosis method of spacecraft control systems based on PCA-ResNet," *J. Phys., Conf. Ser.*, vol. 2258, no. 1, Apr. 2022, Art. no. 012062.
- [19] J. Gu, Y. Peng, H. Lu, X. Chang, and G. Chen, "A novel fault diagnosis method of rotating machinery via VMD, CWT and improved CNN," *Measurement*, vol. 200, Aug. 2022, Art. no. 111635.
- [20] Y. Jin, C. Qin, Z. Zhang, J. Tao, and C. Liu, "A multi-scale convolutional neural network for bearing compound fault diagnosis under various noise conditions," *Sci. China Technol. Sci.*, vol. 65, no. 11, pp. 2551–2563, Nov. 2022.
- [21] H. Zhou, W. Chen, L. Cheng, D. Williams, C. W. De Silva, and M. Xia, "Reliable and intelligent fault diagnosis with evidential VGG neural networks," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.
- [22] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, 2015.
- [23] M. Qaraei, S. Abbaasi, and K. Ghiasi-Shirazi, "Randomized non-linear PCA networks," *Inf. Sci.*, vol. 545, pp. 241–253, Feb. 2021.
- [24] Y. Song and C. Chen, "MPPCANet: A feedforward learning strategy for few-shot image classification," *Pattern Recognit.*, vol. 113, May 2021, Art. no. 107792.
- [25] D. Yu and X.-J. Wu, "2DPCANet: A deep leaning network for face recognition," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 12919–12934, May 2018.
- [26] W. Yu, C. Zhao, and B. Huang, "MoniNet with concurrent analytics of temporal and spatial information for fault detection in industrial processes," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8340–8351, Aug. 2022.
- [27] H. Yang, Z. Yang, H. Yang, and Y. Xie, "Fault detection for air conditioner using PCANet," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 3363–3366.
- [28] J. Guo, J. Cheng, and Y. Yang, "Fault diagnosis method of rolling bearings based on improved multi-linear principal component analysis network," *China Mech. Eng.*, vol. 33, no. 02, p. 187, 2022.
- [29] D. Cui and K. Xia, "Dimension reduction and defect recognition of strip surface defects based on intelligent information processing," *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 6729–6736, Dec. 2018.
- [30] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1208–1213.
- [31] A. Miao, Z. Ge, Z. Song, and L. Zhou, "Time neighborhood preserving embedding model and its application for fault detection," *Ind. Eng. Chem. Res.*, vol. 52, no. 38, pp. 13717–13729, Sep. 2013.
- [32] X. Sha, C. Luo, and N. Diao, "A robust fault detection method based on neighborhood preserving embedding," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
- [33] Y. Zhai et al., "Robust SAR automatic target recognition based on transferred MS-CNN with L2-regularization," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–13, Nov. 2019.
- [34] L. Zhang et al., "Distance-based weight transfer for fine-tuning from near-field to far-field speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.
- [35] F. You, Y. Cao, and C. Zhang, "Deep domain adaptation with a few samples for face identification," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 178–183.
- [36] J. Wang, "Generalized 2-D principal component analysis by Lp-norm for image analysis," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 792–803, Mar. 2016.
- [37] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
- [38] D. Liu, Z. Xiao, X. Hu, C. Zhang, and O. P. Malik, "Feature extraction of rotor fault based on EEMD and curve code," *Measurement*, vol. 135, pp. 712–724, Mar. 2019.
- [39] P. Cao, S. Zhang, and J. Tang, "Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning," *IEEE Access*, vol. 6, pp. 26241–26253, 2018, doi: [10.1109/ACCESS.2018.2837621](https://doi.org/10.1109/ACCESS.2018.2837621).
- [40] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," *Mech. Syst. Signal Process.*, vol. 64, pp. 100–131, 2015.
- [41] H. Huang and N. Baddour, "Bearing vibration data collected under time-varying rotational speed conditions," *Data Brief*, vol. 21, pp. 1745–1749, Dec. 2018, doi: [10.1016/j.dib.2018.11.019](https://doi.org/10.1016/j.dib.2018.11.019).
- [42] A. Lundström and M. O'Nils, "Factory-based vibration data for bearing-fault detection," *Data*, vol. 8, no. 7, p. 115, Jun. 2023.
- [43] Z. Liang, S. Xia, Y. Zhou, L. Zhang, and Y. Li, "Feature extraction based on lp-norm generalized principal component analysis," *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 1037–1045, Jul. 2013.
- [44] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
- [45] X. He and P. Niyogi, "Locality preserving projections," in *Proc. NIPS*, vol. 16, 2003, pp. 186–197.
- [46] D. Cai, X. He, and J. Han, "Isometric projection," presented at the Conf. Artif. Intell. (AAAI), 2007.
- [47] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, nos. 7–9, pp. 1547–1553, Mar. 2007.
- [48] H. Wu et al., "CVT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.



Jinping Wang is currently pursuing the master's degree with the College of Computer and Information Science, Chongqing Normal University, Chongqing, China.

He is doing research in machine learning and computer vision.



Ruisheng Ran received the B.S. degree in mathematics from Chongqing Normal University, Chongqing, China, in 1998, and the M.S. degree in computational mathematics and the Ph.D. degree in computer application technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2004 and 2006, respectively.

He is currently a Professor with the College of Computer and Information Science, Chongqing Normal University. He is doing research in pattern recognition and computer vision.



Bin Fang (Senior Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1989, the M.S. degree in electrical engineering from Sichuan University, Chengdu, China, in 1994, and the Ph.D. degree in electrical engineering from The University of Hong Kong, Hong Kong, in 2001.

He is currently a Professor with the Department of Computer Science, Chongqing University, Chongqing, China. His research interests include computer vision, pattern recognition, medical image processing, biometrics applications, and document analysis. He has published more than 100 technical articles.

Dr. Fang is an Associate Editor of the *International Journal of Pattern Recognition and Artificial Intelligence*.