An Improved PixelHop Framework and Its Application in Rolling Bearing Fault Diagnosis

LANJUN WAN1,2, ZHENG ZHOU1,2, KUN GONG1,2, GEN ZHANG1,2, YUANYUAN LI1,2, CHANGYUN LI1,2

1School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China
2Hunan Key Laboratory of Intelligent Information Perception and Processing Technology, Hunan University of Technology, Zhuzhou 412007, China

Corresponding author: Lanjun Wan (wanlanjun@hut.edu.cn)

This work was supported in part by the Natural Science Foundation of Hunan Province, China under Grant 2019JJ60048, in part by the National Natural Science Foundation for Young Scientists of China under Grant 61702177, in part by the National Key Research and Development Project under Grant 2018YFB1700204, Grant 2018YFB1003401, and Grant 2019YFD1101305, and in part by the Key Research and Development Project of Hunan Province under Grant 2019GR2133.

ABSTRACT The PixelHop framework based on successive subspace learning (SSL) has been widely used in signal processing and computer vision, which can effectively improve the classification accuracy in high spatial resolution scenes through successive subspace growth. To solve the problems of insufficient feature extraction and dependence on prior knowledge in the PixelHop framework, an improved PixelHop (I-PixelHop) framework is proposed. On the basis of PixelHop framework, I-PixelHop has made the following improvements. 1) I-PixelHop fully extracts the continuous features in one-dimensional sequence data through the improved neighborhood expansion, which can provide a richer feature set. 2) The improved label-assisted regression (ILAG) unit uses the Bi-K-Means clustering algorithm to enable more correct clustering of similar samples, and it adopts the cross-entropy threshold method to alleviate the negative effects caused by the improper setting of the number of pseudo-classes. 3) The high-dimensional features are fully reused by adopting the pseudo dense connection structure to obtain a better feature set. Moreover, the proposed I-PixelHop framework is applied to the rolling bearing fault diagnosis. A series of experiments are carried out to verify the effectiveness of the proposed I-PixelHop. The experimental results show that the fault diagnosis accuracy of I-PixelHop can reach 98.91% and 98.74% on the two different rolling bearing fault datasets, and it also has satisfactory anti-noise ability, faster training speed, and smaller model size.

INDEX TERMS Fault diagnosis, machine learning, object classification, rolling bearing, successive subspace learning.

I. INTRODUCTION

Subspace learning is a classic approach to dimensionality reduction, which has been widely used in signal processing [1] and computer vision [2]. At present, many researchers have focused on the single-stage subspace learning. Bessaoudi et al. [3] proposed a linear multi-perspective subspace learning method for face kinship verification in the wild. Yang et al. [4] proposed a transfer subspace learning method by preserving the image structure, which is used to analyze encrypted images. Qin et al. [5] developed a structured subspace learning method that induces symmetric non-negative matrix factorization to learn similar subspaces and latent subspaces. Liao et al. [6] designed a classification algorithm based on supervised subspace learning and non-local representation for human body posture change recognition. The above research can find the most representative subspace through mathematical operations. However, it is difficult to find the optimal subspace through the single-stage subspace learning in high spatial resolution scenes.

SSL [7] is a machine learning method based on feed-forward design [8], which can effectively improve the classification accuracy in high spatial resolution scenes. Chen et al. [9] developed the PixelHop framework based on SSL, which is better than the classic convolutional neural network model of similar model complexity in terms of classification accuracy and training complexity. On the basis of PixelHop, FaceHop [10] is proposed to efficiently classify the low-resolution face gender, VoxelHop [11] is developed to ac-
Aiming at the problem of insufficient feature extraction, DefakeHop [12] is put forward to detect image forgery, and R-PointHop [13] is proposed to provide an accurate point cloud registration. The above research demonstrate that SSL has been successfully adopted in several aspects of the field of computer vision, but it has not been applied in rolling bearing fault diagnosis.

Recently, various machine learning methods have been widely used in rolling bearing fault diagnosis. Shao et al. [14] combined support vector machine (SVM) and multi-scale time-shifting discrete entropy to diagnose rolling bearing faults. Toma et al. [15] used genetic algorithm to select effective features and adopted k-nearest neighbor (KNN) algorithm to classify rolling bearing faults. Lin et al. [16] used back propagation neural network (BPNN) to perform fault state detection of rolling bearing. Shih et al. [17] adopted local mean decomposition and permutation entropy to extract fault features of rolling bearing vibration signals and utilized K-Means clustering algorithm to identify rolling bearing faults. Although the above methods have achieved good fault diagnosis accuracy, the problem of manual feature extraction is still very prominent.

In recent years, deep learning has been introduced into the field of fault diagnosis [18]. Deep learning does not require complex feature engineering, which can automatically extract fault features from vibration signals. In [19], [20], VGG-19 and ResNet-50 were applied in rolling bearing fault diagnosis, respectively. Liu et al. [21] improved the lightweight neural network ShuffleNet and applied it to rolling bearing fault diagnosis. Wan et al. [22] proposed the improved 2D LeNet-5 network for rolling bearing fault diagnosis. The above deep learning models provide a better fault diagnosis accuracy, but they have a higher training complexity.

Compared with traditional machine learning models, the PixelHop framework has higher classification accuracy. Compared with modern deep learning models, the PixelHop framework has more advantages in training speed and training complexity. Therefore, in this paper, the PixelHop framework is adopted in rolling bearing fault diagnosis. However, the PixelHop framework has problems such as insufficient feature extraction and dependence on prior knowledge. To solve these problems, an improved PixelHop framework named I-PixelHop is proposed, it can make full use of features and reduce the dependence on prior knowledge, which makes it more suitable for rolling bearing fault diagnosis.

The main contributions of this paper are as follows.

- An improved PixelHop framework is proposed to solve the problems of insufficient feature extraction and dependence on prior knowledge in the PixelHop framework.
- Aiming at the problem of insufficient feature extraction, I-PixelHop fully extracts the continuous features in one-dimensional sequence data through the improved neighborhood expansion, and the high-dimensional features are fully reused by adopting the pseudo dense connection structure, which can provide a richer and better feature set.
- Aiming at the problem of dependence on prior knowledge, the ILAG unit is developed. It uses the Bi-K-Means clustering algorithm to enable more correct clustering of similar samples, and adopts the cross-entropy threshold method to alleviate the negative effects caused by improper setting of the number of pseudo-classes.
- The proposed I-PixelHop framework is applied to the rolling bearing fault diagnosis, and a series of experiments are conducted to evaluate its effectiveness. The experimental results show that I-PixelHop not only obtains better fault diagnosis accuracy, but also has satisfactory anti-noise ability, faster training speed, and smaller model size.

The rest of the paper is organized as follows. Section II introduces the PixelHop framework based on SSL. Section III describes the proposed I-PixelHop framework. Section IV discusses the application of I-PixelHop in rolling bearing fault diagnosis. Section V presents the experimental results and analysis. Section VI concludes the paper.

**II. PIXELHOP FRAMEWORK BASED ON SSL**

The PixelHop framework [9] based on multi-stage successive subspace learning mainly includes a sequence of PixelHop and LAG units in cascade.

**A. PIXELHOP UNIT**

The PixelHop units are used to capture attributes of near-to-far neighborhoods of selected pixels. Each PixelHop unit consists of neighborhood construction and Saab (subspace approximation via adjusted bias) transform.

1) **Neighborhood Construction**

In the PixelHop framework, the input subspace is not fixed but grows from one stage to another. In the first stage, an input subspace can be formed by taking the union of a pixel and its eight nearest neighbors. In the second stage, each neighborhood pixel is taken as the central pixel, and the union of the new central pixel and its eight nearest neighbors again, and so on. Thus, the former input subspace is a proper subset of the latter one. The process of union is called neighborhood construction.

2) **Saab Transform**

Saab transform [8] is a variant of principal component analysis and a linear subspace learning algorithm. The nonlinearity of the activation function is eliminated by adding bias vectors. There are two parts included in Saab transform: anchor vector selection and bias vector selection. Through Saab transform, the signal space can be decomposed into two subspaces, namely DC subspace and AC subspace.

**B. LAG UNIT**

After performing the unsupervised dimensionality reduction of training samples by Saab transform, the features extracted by Saab transform are not the most representative features.
In order to obtain more representative features, a supervised dimensionality reduction method named LAG is used, which is described as follows.

Firstly, after feature extraction in the PixelHop unit, the K-Means clustering algorithm is performed on the same class of training samples to generate \( M = b \times k \) clusters, where \( b \) is the number of real labels and \( k \) is the number of pseudo-classes corresponding to a real label.

Secondly, the output vectors are changed from one-hot vectors to probability vectors. The \( n \)-dimensional attribute vector of the \( \alpha \)-th object class is \( X_\alpha = (x_{\alpha,1}, x_{\alpha,2}, \ldots, x_{\alpha,n}) \), and its corresponding \( k \) clustering centers are denoted by \( C_{\alpha,1}, C_{\alpha,2}, \ldots, C_{\alpha,k} \), where \( C_{\alpha,i} = (c_{\alpha,i,1}, c_{\alpha,i,2}, \ldots, c_{\alpha,i,n}) \) and \( 1 \leq \alpha \leq b \). Therefore, the probability that the sample \( X_\alpha \) belongs to the clustering center \( C_{\alpha,i} \) is calculated by

\[
prob(X_\alpha, C_{\alpha,i}) = \frac{\exp(-rd(X_\alpha, C_{\alpha,i}))}{\sum_{i=1}^{k} \exp(-rd(X_\alpha, C_{\alpha,i}))},
\]

where \( d(X_\alpha, C_{\alpha,i}) \) is the simple Euclidean distance between the sample \( X_\alpha \) and the clustering center \( C_{\alpha,i} \), and \( r \) is the parameter used to determine the relationship between the Euclidean distance and the likelihood of samples belonging to a cluster. The larger \( r \) is, the faster the probability decays with the Euclidean distance. The smaller the Euclidean distance is, the greater the possibility of correct clustering is. The probability vector of the sample \( X_\alpha \) can be defined as \( p_\alpha(X_\alpha) = (prob(X_\alpha, C_{\alpha,1}), prob(X_\alpha, C_{\alpha,2}), \ldots, prob(X_\alpha, C_{\alpha,k}))^T \).

Finally, a linear least squares regression (LSR) equation group is established to correlate the input attribute vectors and the output probability vectors. The solution of the LSR equation group is a LSR matrix and is also named as the ensemble label classifier.

III. THE PROPOSED I-PXELHOP FRAMEWORK

A. OVERVIEW OF THE I-PXELHOP FRAMEWORK

To solve the problems of insufficient feature extraction and dependence on prior knowledge in the PixelHop framework, an improved PixelHop framework named I-PixelHop is proposed.

The flowchart of the I-PixelHop framework is shown in Fig. 1. Specifically, firstly, the two-dimensional grayscale or color images are sent to a series of cascaded PixelHop units, and the high-dimensional feature space in the shallow PixelHop unit is re-extracted using the pseudo dense connection structure, and the non-overlapping spatial pooling operation for dimensionality reduction is performed to get the attribute vectors. Secondly, the attribute vectors are fed to the ILAG unit while training and optimizing the LSR matrix used for supervised dimensionality reduction. Thirdly, the \( M \)-dimensional probability vectors output by all ILAG units are concatenated to form a feature vector set. After removing the mean and normalizing the variance, the feature vector set conforming to the standard normal distribution is obtained. Finally, SVM is used to classify the feature vectors, and the classification results are obtained.

B. IMPROVED NEIGHBORHOOD EXPANSION

Most one-dimensional time-domain signals have the continuous features, and the state of the current point of the time-domain sequence is correlated with the state of the former point [23]. The PixelHop framework uses fewer pixels for neighborhood construction, and the continuous features in the time-domain sequences cannot be extracted, which leads to the generated feature set cannot correctly express the feature information contained in the time-domain signals.

To solve the difficulty of extracting the continuous features in the PixelHop framework, the neighborhood space is further expanded to obtain a larger receptive field. In the

---

**FIGURE 1.** Flowchart of the I-PixelHop framework.
improved neighborhood expansion, the dimension of neighbor-
hood in the first PixelHop unit can be specified as $d^2$, and
the dimension of neighborhood in the $i$-th PixelHop unit can
be specified as $d^2 \times 9^{i-1}$, where $d \geq 4$ and $i \geq 2$.

![Figure 2](image)

**FIGURE 2.** Diagram of different neighborhood expansion amplitudes. (a) 8-neighborhood expansion, (b) 24-neighborhood expansion, (c) 48-neighborhood expansion.

Fig. 2(a) shows the neighborhood expansion amplitude adopted in the PixelHop framework, Fig. 2(b) and Fig. 2(c) present the two different initial neighborhood expansion amplitudes adopted in the I-PixelHop framework. It can be seen from Fig. 2 that when the neighborhood expansion amplitude becomes larger, each pixel will have a larger neighborhood, which can provide a feature set containing more continuous features.

### C. ILAG UNIT

1) Overall Design of ILAG Unit

To reduce the dependence on prior knowledge in the Pixel-
Hop framework, the ILAG unit is developed. It uses the Bi-K-
Means clustering algorithm to enable more correct clustering of similar samples, and it adopts the cross-entropy threshold method to alleviate the negative effects caused by improper hyperparameter settings.

In the LAG unit, the probability vector of a sample is constructed using (1), and this process is based on simple Euclidean distance. Although the simple Euclidean distance is very practical, it neglects the differences in different attributes between samples. Therefore, the standardized Euclidean distance is used to construct the probability vector, the probability that the sample $X_\alpha$ belongs to the clustering center $C_{\alpha,i}$ is calculated by

$$
prob(X_\alpha, C_{\alpha,i}) = \frac{\exp(-rS(X_\alpha, C_{\alpha,i}))}{\sum_{i=1}^{k} \exp(-rS(X_\alpha, C_{\alpha,i}))}.
$$

(2)

In (2), the standardized Euclidean distance between the sample $X_\alpha$ and the clustering center $C_{\alpha,i}$ is calculated by

$$
S(X_\alpha, C_{\alpha,i}) = \sqrt{\sum_{j=1}^{n} \left( \frac{x_{\alpha,j} - c_{\alpha,i,j}}{sd_j} \right)^2},
$$

(3)

where $sd_j$ is the standard deviation of the $j$-th element of the $X_\alpha$ and $C_{\alpha,i}$.

The flowchart of the ILAG unit in the training stage is shown in Fig. 3(a), which is described as follows.

**Step 1:** Perform the Bi-K-Means clustering algorithm on all the same class of attribute vectors to generate $M$ clusters.

**Step 2:** Calculate the probability that the sample $X_\alpha$ belongs to the clustering center $C_{\alpha,i}$ by (2) to obtain the probability vector $p_\alpha(X_\alpha)$, where $1 \leq \alpha \leq b$ and $1 \leq i \leq k$.

**Step 3:** Adjust the probability vectors using the proposed cross-entropy threshold method, which can increase the probability of the correct classification.

**Step 4:** Use the input attribute vectors and output probability vectors to establish an equation group containing $M$ LSR equations, which is solved to obtain a LSR matrix. The LSR matrix is optimized through the feeding of training samples.

The flowchart of the ILAG unit in the test stage is shown in Fig. 3(b). The dot-product operations are performed on the LSR matrix and test samples to obtain $M$-dimensional probability vectors.

![Flowchart](image)

**FIGURE 3.** Flowchart of the ILAG unit. (a) Training stage, (b) Test stage.

2) Use of Bi-K-Means Clustering Algorithm

In order to enable more correct clustering of similar samples, the K-Means clustering algorithm adopted in the PixelHop framework is replaced by the binary K-Means (Bi-K-Means) clustering algorithm, which can make the input attribute vectors and the output probability vectors to be better soft-associated. In order to let the generated LSR equation group better fit the real distribution of data, the standardized Euclidean distance is used in the Bi-K-Means clustering algorithm. As shown in Fig. 3(a), the clustering analysis is carried out on the same class of attribute vectors using the Bi-K-Means clustering algorithm to generate $k$ clusters, and all the input attribute vectors with $b$ real labels are divided into $M = b \times k$ clusters.

Table 1 presents the fault diagnosis accuracies obtained with different clustering algorithms on the Case Western Reserve University (CWRU) rolling bearing fault dataset [24]. It can be seen from Table 1 that the fault diagnosis accuracy of Bi-K-Means clustering algorithm is 0.67% higher than...
that of K-Means clustering algorithm. In the ILAG unit, the clustering effect of the same class of attribute vectors would affect the soft-association between the input attribute vectors and the output probability vectors. Therefore, the Bi-K-Means clustering algorithm with higher clustering accuracy is adopted in the ILAG unit.

### 3) Cross-Entropy Threshold Method

In the LAG unit, the number of pseudo-classes corresponding to a real label needs to be specified. If the number of the pseudo-classes is too small, the generated LSR matrix cannot correctly fit the real distribution of data. If the number of pseudo-classes is too large, the process of solving the LSR equation group has a high time complexity, and the overfitting phenomenon also occurs, which causes the model performance cannot be evaluated correctly. It is important to reasonably specify the number of pseudo-classes, but it depends on a prior knowledge. In the ILAG unit, the number of pseudo-classes also needs to be specified, in order to alleviate the negative effects caused by the improper setting of the number of pseudo-classes, the cross-entropy threshold method is proposed.

The cross-entropy loss function is one of the most widely used loss functions, and it is often used to measure the difference information between two probability distributions. The cross-entropy function $H(p, q)$ is defined as

$$H(p, q) = -\sum_{i=1}^{z} p(X_i) \log(q(X_i)),$$

where $p(X_i)$ represents the true probability distribution of sample $X_i$, $q(X_i)$ represents the predicted probability distribution of sample $X_i$, and $z$ is the total number of samples. The proposed cross-entropy threshold method includes the following several steps.

Step 1: Define $k$ one-hot vectors $V_1, V_2, \ldots, V_k$. For any $V_j = (v_1, v_2, v_3, \ldots, v_k)$, if $i = j$, then $v_i = 1$; otherwise, $v_i = 0$, where $1 \leq i \leq k$ and $1 \leq j \leq k$.

Step 2: Calculate the cross-entropy values $H_{\alpha, 1}, H_{\alpha, 2}, \ldots, H_{\alpha, k}$ of the probability vector $p_\alpha(X_\alpha)$ and the $k$ one-hot vectors $V_1, V_2, \ldots, V_k$ by (4), where $1 \leq \alpha \leq b$. The greater the cross-entropy value, the greater the difference between the true probability distribution and the predicted probability distribution.

Step 3: Filter interference items by the threshold $TS$, which is an adjustable hyperparameter. If $H_{\alpha, i}$ is larger than $TS$, at first the value of $prob(X_\alpha, C_{\alpha, i})$ is assigned to the remaining $k - 1$ elements excluding the $i$-th element of $p_\alpha(X_\alpha)$ according to their respective weights, and then the value of $prob(X_\alpha, C_{\alpha, i})$ is set to 0, where $1 \leq \alpha \leq b$ and $1 \leq i \leq k$.

The weight $w_j$ ($1 \leq j \leq k$ and $j \neq i$) is calculated by

$$w_j = \frac{prob(X_\alpha, C_{\alpha, j})}{k} + \lambda [\frac{prob(X_\alpha, C_{\alpha, j}) - m}{s}]$$

where $\lambda$ is a regularization coefficient, $m$ and $s$ represent the mean and standard deviation of $p_\alpha(X_\alpha)$, respectively.

Step 4: Adjust the probability vector $p_\alpha(X_\alpha)$ after filtering the interference items, where $1 \leq \alpha \leq b$. The value of $prob(X_\alpha, C_{\alpha, j})$ in the adjusted probability vector $p_\alpha'(X_\alpha)$ is updated by

$$prob'(X_\alpha, C_{\alpha, j}) = w_j \cdot prob(X_\alpha, C_{\alpha, i}) + prob(X_\alpha, C_{\alpha, j})$$

where $1 \leq j \leq k$ and $j \neq i$. The updated probability vector is used to construct the LSR equation group.

### D. Pseudo Dense Connected Structure

In the PixelHop framework, the current PixelHop unit cannot sufficiently extract the high-dimensional features extracted in the previous PixelHop units, which will affect the classification accuracy. Therefore, in order to fully reuse the high-dimensional features, the pseudo dense connection structure is designed.

The idea of the pseudo dense connection structure comes from the dense connection structure of DenseNet [25]. DenseNet, a convolutional neural network with a number of layers, adopts a dense connection mechanism: all layers are connected to each other, and each layer accepts the outputs all the previous layers as its additional inputs. The simple dense connection structure is shown in Fig. 4, which has the following advantages: 1) greatly reducing the number of training parameters; 2) enhancing the reuse of features using the bypass technology; 3) alleviating the problems of gradient explosion and model overfitting.

In the PixelHop framework, the input data is processed by a series of cascaded PixelHop units, and the output of the current PixelHop unit is the input of the next one, which is defined as

$$P_{i+1} = \text{PixelHop}(P_i),$$

where $P_i$ represents the input of the $i$-th PixelHop unit, $P_{i+1}$ is the output of the $i$-th PixelHop unit, and $\text{PixelHop}$ denotes the PixelHop operation. In the I-PixelHop framework, the pseudo dense connection structure can be expressed as

$$P_{i+1} = \text{PixelHop}(P_1 + P_2 + \ldots + P_i).$$

The pseudo dense connection structure is shown in Fig. 5. The features are firstly extracted through a PixelHop unit, and then the pooling is used to perform non-overlapping downsampling operation. The pseudo dense connection structure is designed.
not feeds the output of the current PixelHop unit to the next one but feeds the output of the current PixelHop unit to each PixelHop unit behind it, which can fully reuse features to obtain a better feature set. Furthermore, the pseudo dense connection structure can effectively suppress overfitting.

E. COMPARISON OF PIXELHOP AND I-PIXELHOP

The I-PixelHop framework is an improved version of the PixelHop framework [9], and both PixelHop and I-PixelHop are based on SSL. The similarities between PixelHop and I-PixelHop are as follows: i) both of them obtain the attribute vectors through neighborhood construction and Saab transform; ii) both of them process the attribute vectors through a kind of functional unit to obtain the feature vectors; iii) both of them use a SVM classifier to classify the feature vectors. However, there are the following differences between the framework structure of PixelHop and that of I-PixelHop.

- PixelHop uses a smaller neighborhood expansion amplitude, while I-PixelHop uses a larger neighborhood expansion amplitude.
- PixelHop adopts the LAG units to process the attribute vectors, while I-PixelHop adopts the ILAG units to process the attribute vectors.
- In the PixelHop framework, the output of the current PixelHop unit is only fed to the next PixelHop unit. However, in the PixelHop framework, the output of the current PixelHop unit is fed to each PixelHop unit behind it through the pseudo dense connection structure.

Due to PixelHop and I-PixelHop have different framework structures, they have different characteristics in some aspects, as listed in Table 2. The detail analysis of different characteristics between PixelHop and I-PixelHop are shown as follows.

1) Receptive field

PixelHop uses the central pixel and its eight nearest neighbors for neighborhood construction. However, I-PixelHop uses more neighbors for neighborhood construction to obtain a larger receptive field, which provides a richer feature set.

2) Feature reuse

I-PixelHop adopts the pseudo dense connection structure to fully reuse the high-dimensional features, which provides a better feature set than PixelHop.

3) Feature representation

PixelHop uses K-Means clustering algorithm to cluster the same class of samples to optimize feature representation. However, I-PixelHop uses Bi-K-Means clustering algorithm with better clustering performance to support more correct clustering of similar samples, and it uses the cross-entropy threshold method to filter the interference items of invalid pseudo-classes, which obtains a better feature representation.

4) Prior knowledge dependence

In the LAG unit of PixelHop, the number of pseudo-classes needs to be specified reasonably using a prior knowledge. In the ILAG unit of I-PixelHop, the cross-entropy threshold method is adopted to alleviate the negative effects caused by the improper setting of the number of pseudo-classes, which reduces the depen-
5) Training complexity

Due to the improved neighborhood expansion and pseudodense connection structure adopted in the I-PixelHop framework, the feature set processed by the Saab transform kernel is larger, which makes the training time of I-PixelHop model longer than that of PixelHop model.

IV. APPLICATION OF I-PIXELHOP IN ROLLING BEARING FAULT DIAGNOSIS

A. FAULT DIAGNOSIS PROCESS

The proposed I-PixelHop framework is applied to the rolling bearing fault diagnosis. The flowchart of the rolling bearing fault diagnosis based on the I-PixelHop framework is shown in Fig. 6. Firstly, the one-dimensional raw vibration signals are converted into two-dimensional grayscale images. Secondly, the grayscale image dataset is divided into a training set and a test set. Thirdly, the rolling bearing fault diagnosis model based on the I-PixelHop framework is trained with the training set. Finally, the fault diagnosis model is used to diagnose the test set, and the fault diagnosis results are obtained.

![Flowchart of the rolling bearing fault diagnosis.](Image)

B. DATA PREPROCESSING METHOD

The CWRU rolling bearing fault dataset consists of a large number of one-dimensional time-domain signals. In the rolling bearing fault diagnosis based on I-PixelHop, the one-dimensional time-domain signals need to be converted into two-dimensional grayscale images. In general, the simple signal-to-image method (STIM) [26] is used to convert the one-dimensional time-domain signals into two-dimensional grayscale images, which is difficult to extract the continuous features in one-dimensional time-domain signals. Gram angle difference field (GADF) and Gram angle sum field (GASF) [27] can easily perform angular perspective on the one-dimensional time-domain signals, thereby time correlations in different time intervals are identified. Therefore, GADF and GASF are introduced to replace STIM.

Figs. 7 and 8 show the feature images generated by GADF and GASF, respectively. It can be seen from Fig. 7 that the texture characteristics of the normal data are very obvious, the texture characteristics of the inner race fault and the outer race fault are equally clear, while the texture characteristics of the ball fault have some undulations. The feature representation is dense in normal data, but it is very sparse in fault data. As shown in Fig. 7(a), the gray-value distribution range of a single pixel and its surrounding spatial neighbors is large in the feature image generated by GADF for the normal data. As shown in Figs. 7(b)–7(j), the gray-value distribution range of a single pixel and its surrounding spatial neighbors is small in the feature images generated by GADF for the fault data. It can be seen from Fig. 8 that the feature images generated by GASF do not seem to have obvious texture characteristics. The reason is that GASF is the inverse function of GADF, if the feature image generated by GADF has obvious texture characteristics, the texture characteristics of the feature image generated by GASF are not obvious.

C. FAULT DIAGNOSIS MODEL TRAINING

The process of training the fault diagnosis model based on the I-PixelHop framework is described as follows.

Step 1: The one-dimensional original vibration signals are converted into two-dimensional grayscale images of 64 × 64, which are input into each PixelHop unit.

Step 2: The two-dimensional grayscale images are processed though the PixelHop and pooling operations to obtain the attribute vectors, which are fed to the ILAG unit and the next PixelHop unit. After the attribute vectors are processed by the ILAG unit, the probability vectors are obtained.

Step 3: Repeat Step 2 until the last PixelHop unit. Note that θ-neighborhood expansion is adopted in the first PixelHop unit and 8-neighborhood expansion is adopted in all the subsequent PixelHop units, and there is no pooling operation after the last PixelHop operation, where θ can be set to 24, 48, and so on.

Step 4: The probability vectors output by all ILAG units are concatenated to a feature vector set, which are used to train a SVM classifier, and finally the rolling bearing fault diagnosis model is obtained.

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

The dataset used in experiments is provided by CWRU [24], and the fault data collected on the drive-end at the sampling frequency of 12 kHz and the normal baseline data are selected. These data are divided into a training set and a test set according to the ratio of 7:3, and the description of rolling bearing data is shown in Table 3.

The hardware configurations of the experimental platform include one quad-core Intel Xeon E3-1225 v5 CPU at 3.3 GHz and 64 GB main memory. The software configurations of the experimental platform are as follows: CentOS 8.1, Python 3.7.9, Scikit-learn 0.20.3, and OpenCV 4.4.0.

In the training of the rolling bearing fault diagnosis model based on the I-PixelHop framework, the hyperparameter settings are shown in Table 4.

The parameter r is used to determine the relationship between the Euclidean distance and the likelihood of samples...
belonging to a cluster. The consequence of improper selection of $r$ is described in Section II-B.

The parameter $k$ is the number of pseudo-classes corresponding to a real label. If the value of $k$ is too small, the I-PixelHop model will be underfitting; otherwise, the I-PixelHop model will be overfitting and the model training time will be significantly increased.

The parameter $TS$ is the cross-entropy threshold set in the ILAG unit, and its value is less than 0. The value of $TS$ is determined by the values of the elements hoped to be filtered in the probability vectors.

The parameter $\lambda$ is a regularization coefficient, which is used to limit the weight adopted in the cross-entropy threshold method to a certain range, and it is adjusted according to the distribution of the dataset.

B. VALIDATION OF THE IMPROVED NEIGHBORHOOD EXPANSION

1) The Influence of the Improved Neighborhood Expansion on the Fault Diagnosis Accuracy

To verify the influence of the improved neighborhood expansion on the fault diagnosis accuracy, the I-PixelHop mod-
TABLE 3. Description of rolling bearing data.

| Fault Type       | Fault Diameter (inch) | Fault Label |
|------------------|-----------------------|-------------|
| Normal           | Null                  | 0           |
| Inner race fault | 0.007                 | 1           |
| Inner race fault | 0.014                 | 2           |
| Inner race fault | 0.021                 | 3           |
| Ball fault       | 0.007                 | 4           |
| Ball fault       | 0.014                 | 5           |
| Ball fault       | 0.021                 | 6           |
| Outer race fault | 0.007                 | 7           |
| Outer race fault | 0.014                 | 8           |
| Outer race fault | 0.021                 | 9           |

TABLE 4. Hyperparameter settings of I-PixelHop.

| Parameter Name | Parameter Value |
|----------------|-----------------|
| $r$            | 5               |
| $k$            | 8               |
| $TS$           | −1.3           |
| $\lambda$      | 0.005           |

els with different numbers of PixelHop units which have different neighborhood expansion amplitudes are trained. Considering the computational resources of the experimental platform and computational complexity of the I-PixelHop model, the upper limit of the number of cascaded PixelHop units is set to 4, and this setting is also applied to all experiments. In the data preprocessing, the one-dimensional time-domain signals are divided into several samples, each sample contains 4096 sample points, and each sample is converted into a 64×64 grayscale image by using STIM.

Fig. 9 shows the comparison of fault diagnosis accuracies obtained by the I-PixelHop models with different neighborhood expansion amplitudes. As seen in Fig. 9, as the number of PixelHop units increases from 1 to 4, the fault diagnosis accuracies obtained by the I-PixelHop models with 8-neighborhood expansion, 24-neighborhood expansion, and 48-neighborhood expansion have increased from 51.61%, 72.89%, and 78.07% to 93.15%, 94.35%, and 94.58%, respectively. The results show that increasing the number of PixelHop units can significantly improve the fault diagnosis accuracy of the I-PixelHop model.

In addition, when the number of PixelHop units is smaller, the fault diagnosis accuracies of the I-PixelHop models with 24-neighborhood expansion and 48-neighborhood expansion are significantly better than that of the model with 8-neighborhood expansion, which means that increasing the neighborhood expansion amplitude can also greatly improve the diagnosis accuracy. However, as the increase of the number of PixelHop units, the difference in fault diagnosis accuracy between the model with 24-neighborhood expansion and that with 48-neighborhood expansion gradually becomes smaller, which shows that when the PixelHop units reaches a certain number, the increase of neighborhood expansion amplitudes has little effect on the fault diagnosis accuracy.

2) The Influence of the Improved Neighborhood Expansion on the Model Size

To evaluate the influence of the improved neighborhood expansion on the model size, the I-PixelHop models with different numbers of PixelHop units which have different neighborhood expansion amplitudes are trained.

Fig. 10 shows the comparison of sizes of the I-PixelHop models with different numbers of PixelHop units which have different neighborhood expansion amplitudes. As shown in Fig. 10, when the number of PixelHop units and the neighborhood expansion amplitude are increased, the model size is also increased. The reason is that the number of PixelHop units is equal to the number of Saab transform kernels in the
I-PixelHop framework, and the increase of Saab transform kernels will lead to the increase of the model size. In addition, due to the feature set becomes larger as the increase of neighborhood expansion amplitude, the Saab transform kernel will automatically add more filters, which leads to the increase of the model size.

3) The Influence of the Improved Neighborhood Expansion on the Model Training Time

To evaluate the influence of the improved neighborhood expansion on the model training time, the I-PixelHop models with different numbers of PixelHop units which have different neighborhood expansion amplitudes are trained.

![FIGURE 11. Comparison of the training time of the I-PixelHop models with different number of PixelHop units which have different neighborhood expansion amplitudes.](image)

Fig. 11 shows the comparison of the training time of the I-PixelHop models with different numbers of PixelHop units which have different neighborhood expansion amplitudes. As seen in Fig. 11, when the number of PixelHop units and the neighborhood expansion amplitude are increased, the model training time is also increased. The reason is that the data need to be processed by more PixelHop units as increasing the number of PixelHop units. Moreover, when the neighborhood expansion amplitude is increased, the feature set will become larger, which leads to that the I-PixelHop model needs more time to process the bigger feature set.

According to the above three experiments, the neighborhood expansion amplitude is set as 24, which is helpful for constructing an I-PixelHop model with higher fault diagnosis accuracy, smaller model size, and less model training time.

C. VALIDATION OF ILAG UNIT

To verify the effectiveness of the ILAG unit, the I-PixelHop models with LAG unit and ILAG unit are trained, respectively. In addition, these I-PixelHop models have different numbers of PixelHop units.

![FIGURE 12. Comparison of the fault diagnosis accuracies obtained by I-PixelHop models with LAG unit and ILAG unit.](image)

Fig. 12 shows the comparison of the fault diagnosis accuracies obtained by I-PixelHop models with LAG unit and ILAG unit. When the number of PixelHop units is 4, the fault diagnosis accuracy obtained by the I-PixelHop model with ILAG unit is 96.14%, which is 1.79% higher than that with LAG unit. This is because the Bi-K-Means clustering algorithm has better clustering performance than the K-Means clustering algorithm, and the cross-entropy threshold method can alleviate the negative effects caused by the improper setting of the number of pseudo-classes. Therefore, the ILAG unit is necessary to be used in the I-PixelHop framework.

D. VALIDATION OF PSEUDO DENSE CONNECTED STRUCTURE

To verify the effectiveness of the pseudo dense connection structure, the I-PixelHop models with and without the pseudo dense connection structure are trained, respectively. In addition, these I-PixelHop models have different numbers of PixelHop units.

Fig. 13 shows the comparison of fault diagnosis accuracies obtained by I-PixelHop models with and without the pseudo dense connection structure. As shown in Fig. 13, as the increase of the number of PixelHop units, the fault diagnosis accuracies of the I-PixelHop model with the pseudo dense connection structure are 91.53%, 95.19%, and 97.22% respectively, which are 0.07%, 0.77%, and 1.08% higher than without the pseudo dense connection structure respectively. The results show that the I-PixelHop model with the pseudo dense connection structure can achieve better fault diagnosis accuracy than that without the pseudo dense connection structure, and the gap between them is increased with the increase of the number of PixelHop units. This is because the pseudo dense connection structure fully extracts high-dimensional features when more PixelHop units are used. Therefore, it is necessary to use the pseudo dense connection structure.
structure to utilize more PixelHop units, and the number of PixelHop units is set to 4.

E. COMPARISON OF DIFFERENT DATA PREPROCESSING METHODS

To evaluate the influence of different data preprocessing methods on the fault diagnosis accuracy of the I-PixelHop model, the I-PixelHop models with different data preprocessing methods are trained.

![Comparison of fault diagnosis accuracies obtained by I-PixelHop models with and without the pseudo dense connection structure.](image)

Fig. 13. Comparison of fault diagnosis accuracies obtained by I-PixelHop models with and without the pseudo dense connection structure.

![Comparison of fault diagnosis accuracies obtained by I-PixelHop models with different data preprocessing methods.](image)

Fig. 14. Comparison of fault diagnosis accuracies obtained by I-PixelHop models with different data preprocessing methods.

![The confusion matrix of the test samples used in the I-PixelHop model.](image)

Fig. 15. The confusion matrix of the test samples used in the I-PixelHop model.

The results show that GADF can better improve the fault diagnosis accuracy of the I-PixelHop model than GASF. Therefore, GADF is used to preprocess the rolling bearing data.

F. ANALYSIS OF THE CONFUSION MATRIX

For supervised learning algorithms, the confusion matrix is used to visualize their prediction performance. Each column of the confusion matrix represents the predicted label, and each row represents the true label in the dataset.

![The confusion matrix of the test samples used in the I-PixelHop model.](image)

G. DIAGNOSIS EFFECT ANALYSIS UNDER VARIOUS LOAD CONDITIONS

In order to further evaluate the effectiveness of the I-PixelHop model, the experiments are carried out to obtain the fault diagnosis accuracies of the I-PixelHop model under various load conditions. The fault data adopted in the experiments are collected under the motor load of 0, 1, 2, 3 horsepower (HP) with the fault diameter of 0.021 fault diameter and the inner race fault with 0.021 fault diameter. For rolling bearing fault data, the average diagnosis accuracy of the I-PixelHop model for inner race faults is the highest, which is 99.06%. The average diagnosis accuracy of the I-PixelHop model for ball faults is the lowest, which is 98.33%. The results mean that the I-PixelHop model can accurately diagnose various rolling bearing faults.
Table 5 presents the fault diagnosis accuracies obtained by the I-PixelHop model under various load conditions. As shown in Table 5, the I-PixelHop model achieves satisfactory fault diagnosis accuracies under 0 HP (1797 rpm), 1 HP (1772 rpm), 2 HP (1750 rpm), and 3 HP (1730 rpm). The average fault diagnosis accuracies obtained under 0 HP, 1 HP, 2 HP, and 3 HP can reach 99.72%, 99.07%, 98.87%, and 98.83%, respectively. It is easy to see that the I-PixelHop model obtains a higher diagnosis accuracy under 0 HP, and the diagnosis accuracies obtained under 2 HP and 3 HP are relatively low. The reason may be that the fault features of the vibration signals collected under the motor load of 0 HP are more obvious and easier to be identified correctly.

**H. ANALYSIS OF ANTI-NOISE ABILITY**

To evaluate the effectiveness of the I-PixelHop model under the noisy environment, the Gaussian white noise is added to the original vibration signals to obtain composite signals with different signal-to-noise ratios (SNRs). SNR is defined as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right),$$

where $P_{\text{signal}}$ and $P_{\text{noise}}$ represent the power of signal and the power of noise, respectively. The larger the SNR, the smaller the proportion of noise in the signal. Fig. 16 shows the original signal and the signals under different SNRs, where the SNR ranges from -4 dB to 10 dB. When SNR is 0 dB, the proportion of signal and the proportion of noise in the composite signals are the same.

The I-PixelHop model is trained with noisy data which have various SNRs. In the data preprocessing, the composite signals are divided into several samples, each sample contains 4096 sample points, and each sample is converted into a 64×64 grayscale image by using GADF. Fig. 17 shows the comparison of fault diagnosis accuracies obtained by I-PixelHop models trained by noisy data with different SNRs. As shown in Fig. 17, with the decrease of SNR, the fault diagnosis accuracies of I-PixelHop models decrease from 93.42% to 78.68%, and that of PixelHop models decrease from 91.62% to 76.21%. The results show that the I-PixelHop models achieve better diagnosis accuracy than the PixelHop models under the noisy environment with different SNRs. Therefore, the I-PixelHop model has better anti-noise ability than the PixelHop model. Furthermore, when the
SNR is larger than 0 dB, the fault diagnosis accuracies of I-PixelHop models remain above 86%. The results mean that the I-PixelHop model has good anti-noise ability. The reason is that the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.

I. COMPARISON WITH OTHER FAULT DIAGNOSIS METHODS

In order to further demonstrate the power of the I-PixelHop model, a series of comparison experiments are conducted on the CWRU dataset and KAT dataset [28]. The KAT dataset is provided by KAT data center in Paderborn University, which collects the vibration signals and current signals at a sampling frequency of 64 kHz. Both the vibration signals and current signals include the healthy data, inner race fault data, and outer race fault data. Both the inner race fault data and outer race fault data include the artificial fault data and real damage fault data, and the real damage fault data are collected through the accelerated life test. In order to simulate the real environments, the vibration signals with real damage faults are used in the experiments.

1) Comparison with the PixelHop Framework

To further verify the effectiveness of the rolling bearing fault diagnosis method based on the I-PixelHop framework, the I-PixelHop model is compared with the PixelHop model in the fault diagnosis accuracy, model size, and model training time. In the data preprocessing, the vibration signals are divided into several samples, each sample contains 4096 sample points, and each sample is converted into a $64 \times 64$ grayscale image by using GADF. The preprocessed data are divided into a training set and a test set according to the ratio of 7:3.

Table 6 shows the comparison of various performance of the PixelHop model and I-PixelHop model. As listed in Table 6, for the CWRU dataset, the fault diagnosis accuracy of the I-PixelHop model is higher 4.14% than that of the PixelHop model; for the KAT dataset, the fault diagnosis accuracy of the I-PixelHop model is higher 4.56% than that of the PixelHop model. The reason is that the neighborhood expansion can provide a richer feature set, the Bi-K-Means clustering algorithm with stronger clustering ability can enable more correct clustering of similar samples, and the pseudo dense connection structure can fully reuse the high-dimensional features. It is worth noting that the PixelHop model has more advantages in terms of the model training time and model size. Because when the feature set becomes richer, the Saab transform kernel will automatically add more filters, which leads to the increase of the model size. The I-PixelHop framework needs to process the richer feature set to construct the fault diagnosis model, which leads to the increase of the model training time.

2) Comparison with the Traditional Machine Learning

To further verify the effectiveness of the rolling bearing fault diagnosis method based on the I-PixelHop framework, the I-PixelHop model is compared with fault diagnosis models based on traditional machine learning algorithms, including SVM [14], KNN [15], BPNN [16], and K-Means [17]. The data preprocessing method for the traditional machine learning algorithms is as follows. Firstly, the vibration signals are divided into several samples, each sample contains 4096 sample points. Secondly, the sample is decomposed by the three-layer wavelet packet decomposition [29]. Finally, the data obtained from the third level decomposition are used to calculate the wavelet energy to get 8 time-frequency features. In addition, the hyperparameter settings of the traditional machine learning algorithms are as follows. Note that these settings are selected by the grid-search method.

- SVM: the penalty coefficient $C$ is set to 1, the radial basis function is used as kernel function, and the $\gamma$ is set to 0.25.
- KNN: the number of nearest neighbors is set to 5.
- BPNN: the number of input layer neurons is set to 8, the number of hidden layer neurons is set to 25, the number of output layer neurons is set to 10 and 3 for the CWRU dataset and the KAT dataset respectively, the learning rate is set to 0.001, and the maximum number of iterations is set to 1000.
- K-Means: the number of clusters is set to 10 and 3 for the CWRU dataset and the KAT dataset respectively, and the maximum number of iterations is set to 1000.

![Comparison of fault diagnosis accuracies obtained by I-PixelHop models trained by noisy data with different SNRs.](Image)

**FIGURE 17.** Comparison of fault diagnosis accuracies obtained by I-PixelHop models trained by noisy data with different SNRs.

**TABLE 6.** Comparison of various performance of the PixelHop model and I-PixelHop model.

| Method          | Fault Diagnosis Accuracy (%) | Training Time (s) | Model Size (MB) |
|-----------------|------------------------------|-------------------|-----------------|
|                 | CWRU | KAT | CWRU | KAT | CWRU | KAT |
| PixelHop        | 94.77 | 94.18 | 413  | 474  | 0.8  | 0.9 |
| I-PixelHop      | 98.91 | 98.74 | 489  | 637  | 1.1  | 1.3 |

- The reason is that the neighborhood expansion can provide a richer feature set, the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.

- The reason is that the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.

- The reason is that the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.

- The reason is that the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.

- The reason is that the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.

- The reason is that the Bi-K-Means clustering algorithm has better clustering ability and the cross-entropy threshold method can optimize feature representation.
Comparison of fault diagnosis accuracies obtained by diagnosis models based on different machine learning algorithms.

Fig. 18 shows the comparison of fault diagnosis accuracies obtained by diagnosis models based on different machine learning algorithms. As seen in Fig. 18, the I-PixelHop model achieves the highest fault diagnosis accuracy. When the experiments are conducted on the CWRU dataset, the fault diagnosis accuracy of the I-PixelHop model is higher 5.83%, 7.52%, 7.86%, and 7.46% than that of SVM, KNN, BPNN, and K-Means, respectively. When the experiments are conducted on the KAT dataset, the fault diagnosis accuracy of the I-PixelHop model is higher 21.51%, 24.99%, 16.37%, and 29.41% than that of SVM, KNN, BPNN, and K-Means, respectively. The main reasons are as follows. For SVM, the hyperparameter selection has a great impact on the fault diagnosis accuracy, and its parameters are difficult to be granularly tuned. For KNN, it depends on the classification results of its neighbors, if a sample is misclassified by the KNN classifier, the classification results of the following samples will be affected. For BPNN, it is easy to fall into the local optimal solution due to its weights are easily converged to the local minimum. For K-Means, it is affected by the simple Euclidean distance and over-reliance on the designation of the initial clustering centers. It also can be seen from Fig. 18 that the fault diagnosis accuracies of SVM, KNN, BPNN, and K-Means obtained on the KAT dataset are not satisfactory, but the I-PixelHop model obtains a satisfactory diagnosis accuracy on the KAT dataset. The results show that the I-PixelHop model has more advantage than the traditional machine learning algorithms in terms of the fault diagnosis accuracy.

3) Comparison with the Deep Learning

To further verify the effectiveness of the rolling bearing fault diagnosis method based on the I-PixelHop framework, the I-PixelHop model is compared with fault diagnosis models based on the deep learning algorithms, including VGG-16 [30], ResNet-18 [31], ShuffleNet V2 [21], and LeNet-5 [22]. Note that when the experiments are conducted on the CWRU dataset, the last full-connection layer of VGG-16, ResNet-18, ShuffleNet V2, and LeNet-5 respectively uses 10 neurons to classify the rolling bearing faults; when the experiments are conducted on the KAT dataset, a full-connection layer with 3 neurons is added as the last full-connection layer, and the number of neurons of the second-to-last full-connection layer is changed from 10 to 20. In the data preprocessing, the vibration signals are divided into several samples, each sample contains 4096 sample points, and each sample is converted into a 64×64 grayscale image. In addition, the hyperparameter settings of the deep learning algorithms are as follows.

- VGG-16 and ResNet-18: the neural network structure and hyperparameters are set according to [30] and [31], respectively.
- ShuffleNet V2 and LeNet-5: the neural network structure and hyperparameters are set according to [21] and [22], respectively.

Table 7 shows the comparison of fault diagnosis accuracies, training time, and model sizes of the fault diagnosis models based on different deep learning algorithms and the I-PixelHop framework. Note that for VGG-16, ResNet-18, ShuffleNet V2, and LeNet-5, the model size can be calculated according to the number of training parameters of the neural network and the memory size occupied by each training parameter.

As shown in Table 7, when the experiments are conducted on both the CWRU dataset and KAT dataset, the fault diagnosis accuracies of VGG-16, ResNet-18, ShuffleNet V2, and LeNet-5 are all slightly better than that of the I-PixelHop model. However, when the experiments are conducted on the CWRU dataset, the training time of VGG-16, ResNet-18, ShuffleNet V2, and LeNet-5 are 2.55×, 3.37×, 3.78×, and 1.48× that of the I-PixelHop model, respectively; when the experiments are conducted on the KAT dataset, the training time of VGG-16, ResNet-18, ShuffleNet V2, and LeNet-5 are 2.26×, 2.91×, 1.57×, and 1.27× that of the I-PixelHop model, respectively.

As can also be seen from Table 7, when the experiments are conducted on the CWRU dataset, the model sizes of VGG-16, ResNet-18, and ShuffleNet V2 are much larger than that of I-PixelHop. Although the model size of I-PixelHop is 8.33% smaller than that of LeNet-5, the training time of the I-PixelHop model is 32.46% shorter than that of LeNet-5. When the experiments are conducted on the KAT dataset,
the I-PixelHop model still has advantages in terms of both model training time and model size. The results show that the I-PixelHop model is a lightweight model with faster training speed, smaller model size, and satisfactory fault diagnosis accuracy. Furthermore, the I-PixelHop model has the interpretability that the deep learning models do not have.

VI. CONCLUSION
In this paper, an improved PixelHop framework named I-PixelHop is proposed, which solves the problems of insufficient feature extraction and dependence on prior knowledge in the PixelHop framework. I-PixelHop provides a richer and better feature set through the improved neighborhood expansion and the pseudo dense connection structure, gives a better feature representation by the Bi-K-Means clustering algorithm, and reduces the dependence on prior knowledge using the cross-entropy threshold method. Furthermore, this paper explores the application of I-PixelHop in rolling bearing fault diagnosis. Compared with SVM, KNN, BPNN, K-Means, and PixelHop, the proposed I-PixelHop provides higher fault diagnosis accuracy and whose diagnosis accuracy reaches 98.91% and 98.74% on the two different rolling bearing fault datasets, respectively. Compared with the widely used deep learning models, the proposed I-PixelHop has shorter model training time and smaller model size.

In the modern industry and other application scenarios, the collected data are growing rapidly. It is still a challenge to improve the training efficiency of the I-PixelHop model in the big data environment. Therefore, in the future, the distributed parallelization of I-PixelHop based on Spark platform will be explored to efficiently process the big data.

REFERENCES
[1] H. O. Ahmed and A. K. Nandi, “Three-stage hybrid fault diagnosis for rolling bearings with comprehensively sampled data and subspace learning techniques,” IEEE Trans. Ind. Electron., vol. 66, no. 7, pp. 5516–5524, 2019.
[2] J. Yang, S. Yan, and T. S. Huang, “Ubiquitously supervised subspace learning,” IEEE Trans. Image Process., vol. 18, no. 2, pp. 241–249, 2009.
[3] M. Bessaoudi, A. Chouchane, A. Ouamane, and E. Boutella, “Multilinear subspace learning using handcrafted and deep features for face kinship verification in the wild,” Appl. Artif. Intell., no. 4, pp. 3534–3547, 2020.
[4] L. Yang, M. Men, Y. Xue, J. Wen, and P. Zhong, “Transfer subspace learning based on structure preservation for JPEG image mismatched steganalysis,” Signal Process. Image Commun., vol. 90, pp. 116052, 2021.
[5] Q. Qin, H. Wu, and G. Feng, “Structured subspace learning-induced symmetric nonnegative matrix factorization,” Signal Process., vol. 186, pp. 108–115, 2021.
[6] M. Liao, C. Wang, and X. Gu, “Algorithm using supervised subspace learning and non-local representation for pose variation recognition,” IET Comput. Vis., vol. 14, no. 7, pp. 528–537, 2020.
[7] M. Rouhsedaghat, M. Monajatipoor, Z. Azizi, and C.-C. J. Kuo, “Successive subspace learning: An overview,” arXiv preprint arXiv:2103.00121, 2021.
[8] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, “Interpretable convolutional neural networks via feedforward design,” J. Vis. Commun. Image Represent., vol. 60, pp. 346–359, 2019.
[9] Y. Chen and C.-C. J. Kuo, “PixelHop: A successive subspace learning (SSL) method for object recognition,” J. Vis. Commun. Image Represent., vol. 70, p. 102749, 2020.
[10] M. Rouhsedaghat, Y. Wang, X. Ge, S. Hu, S. You, and C.-C. J. Kuo, “FaceHop: A light-weight low-resolution face gender classification method,” arXiv preprint arXiv:2007.09510, 2020.
LANJUN WAN was born in Hunan, China, in 1982. He received the B.S. and M.S. degrees in computer science and technology from Hunan University of Technology, Zhuzhou, China, in 2005 and 2009, respectively, and the Ph.D. degree in circuits and systems from Hunan University, Changsha, China, in 2016. He is currently an Assistant Professor with the School of Computer Science, Hunan University of Technology, Zhuzhou, China. His research interests include industrial big data analysis, industrial equipment fault diagnosis, high-performance computing, and parallel computing. He has published many research articles in international conferences and journals, such as JPDC, CCPE, ParCo, and Sensors. He serves as a Reviewer for the JPDC, CCPE, Sensors, and IEEE Access.

ZHENG ZHOU was born in Hunan, China, in 1997. He received the B.S. degree in computer science and technology from Jiangxi Science and Technology Normal University, Nanchang, China, in 2019. He is currently pursuing his M.S. degree in computer science and technology at Hunan University of Technology, Zhuzhou, China. His research interests include industrial big data analysis and industrial equipment fault diagnosis.

KUN GONG was born in Hunan, China, in 1996. He received the B.S. degree in mechanical engineering from Hunan University of Technology, Zhuzhou, China, in 2019. He is currently pursuing his M.S. degree in computer science and technology at Hunan University of Technology, Zhuzhou, China. His research interests include industrial big data analysis and industrial equipment fault diagnosis.

GEN ZHANG was born in Anhui, China, in 1995. He received the B.S. degree in network engineering from West Anhui University, Luan, China, in 2019. He is currently pursuing the M.S. degree in computer science and technology at Hunan University of Technology, Zhuzhou, China. His research interests include industrial big data analysis and industrial equipment fault diagnosis.

YUANYUAN LI was born in Henan, China, in 1996. She received the B.S. degree in computer science and technology from Xinlian College of Henan Normal University, Zhengzhou, China, in 2019. She is currently pursuing her M.S. degree in computer science and technology at Hunan University of Technology, Zhuzhou, China. Her research interests include industrial big data analysis and industrial equipment fault diagnosis.

CHANGYUN LI was born in Hunan, China, in 1972. He received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2007. He is currently a Full Professor of Computer Science and dean of the Graduate School of Hunan University of Technology, Zhuzhou, China. His major research interests include industrial big data analysis, industrial equipment fault diagnosis, intelligent information perception and processing technology, internet of things, and software methodology. He has published many research articles in international conferences and journals, such as JICT, JSW, and JCP.

***