Bearing Intelligent Fault Diagnosis in the Industrial Internet of Things Context: A Lightweight Convolutional Neural Network

YANXIN WANG1, JING YAN1, (Member, IEEE), QIFENG SUN2, QIJIAN JIANG1, AND YIZHI ZHOU3

1State Key Laboratory of Electrical Insulation and Power Equipment, Xi’an Jiaotong University, Xi’an 710049, China
2School of Foreign Studies, Xi’an Jiaotong University, Xi’an 710049, China
3School of Mechanical Engineering, Northwestern Polytechnical University, Xi’an 710049, China
Corresponding author: Jing Yan (yanjing@mail.xjtu.edu.cn)

ABSTRACT
The advancement of Industry 4.0 and Industrial Internet of Things (IIoT) has laid more emphasis on reducing the parameter amount and storage space of the model in addition to the automatic and accurate fault diagnosis. In this case, this paper proposes a lightweight convolutional neural network (LCNN) method for intelligent fault diagnosis of bearing, which can largely satisfy the need of less parameter amount and storage space as well as high accuracy. First, depthwise separable convolution is adopted, and a LCNN structure is constructed through an inverse residual structure and a linear bottleneck layer operation. Secondly, a novel decomposed Hierarchical Search Space is introduced to automatically explore the optimal LCNN for bearing fault diagnosis in the context of the IIoT. In the meantime, the real-time monitoring and fault diagnosis of the model are also deployed. In order to verify the validity of the designed model, Case Western Reserve University Bearing fault dataset and MFPT bearing fault dataset are adopted. The results demonstrate the great advantages of the model. The LCNN model can automatically learn and select the appropriate features, highly improving the fault diagnosis accuracy. Meanwhile, the computational and storage costs of the model are largely reduced, which contributes to its being applied to the mechanical system in the IIoT context.

INDEX TERMS Fault diagnosis, lightweight convolutional neural network, decomposed Hierarchical Search Space, Industry 4.0, the industrial Internet of Things.

I. INTRODUCTION
As the core of the whole industrial system, the role rotating machines play is of great importance in modern industry. Moreover, the safe and reliable operation of bearings to a large degree determines the operation of the mechanical systems [1], [2]. However, bearings failures are likely to occur on high-load and strong-impact conditions, leading to the aging of the entire machine and even serious performance or safety loss. With the development of Industry 4.0 and Industrial Internet of Things (IIoT) technology, continuous monitoring and real-time fault diagnosis is indispensable to detecting faults before damage as well as providing important support for maintenance [3]–[6].

Traditional mechanical fault diagnosis mainly includes three steps: i) to construct perfect characteristic parameters that can represent bearing faults by using advanced signal processing methods, such as wavelet decomposition, wavelet packet decomposition, empirical mode decomposition, variational mode decomposition, Spectral kurtosis as well as the improved algorithms of the above methods [7]–[9]; ii) to select key feature parameters via dimensionality reduction methods such as principal component analysis, and auto-encoder [10]; iii) to realize fault classification through pattern recognition methods, including support vector machine (SVM), decision tree, random forest and artificial neural network methods [11], [12]. However, traditional machine
learning methods largely rely on signal processing techniques and diagnostic experience, which makes it difficult to deal with classification or regression problems in complex situations.

With the rapid development of advanced measurement technology as well as the advancement of Industry 4.0 and the IIoT technology, numerous data are collected [13], [14]. Due to high computing complexity, however, traditional machine learning methods fail to establish decision models for these data. Therefore, deep learning has emerged. Due to high computing complexity, however, traditional machine learning methods fail to establish decision models for these data. Therefore, deep learning has emerged and has been adopted for fault diagnosis. C. L. et al. used stacked de-noising auto-encoder to identify the signal status with environmental noise and fluctuations in working conditions [15]. The proposed method proved to achieve high diagnosis accuracy and strong robustness. As for Y. L. et al, they proposed a planetary gear fault diagnosis method on the basis of power-spectral-entropy-based variational mode decomposition and deep neural network to achieve almost perfect fault diagnosis effect [16].

As one of the key branches of deep learning, convolutional neural networks (CNN), have emerged with excellent performance in bearing fault diagnosis in massive data context. S. G. et al. realized accurate, robust and general fault diagnosis of rotating machines through continuous wavelet transform and CNN method [17]. In the experiment of S. S. et al., they proposed a multi-signal fault diagnosis method via deep convolutional neural networks (DCNN) which could learn from multiple sensor signals to achieve robustness and ultimately accurate induction motor fault identification [18]. W. Y. et al. adopted broad convolutional neural network to improve the model’s diagnostic performance and incremental learning capabilities by adding newly generated additional features for self-update so as to include new abnormal samples and fault classes [19].

The aforementioned CNN methods, however, fail to take account of the computational and storage costs of the target models. Simultaneously, the CNN structures largely depend on expert experience to obtain the optimal diagnosis model. In addition, the models may be difficult for training because of gradient vanishing with the increase of the model depth due to limited samples. Recently, some fault diagnosis methods for high-speed devices that demand model deployment and real-time diagnosis have been widely used, laying the foundation for real-time and rapid diagnosis in the context of the IIoT [20], [21]. Moreover, some high-precision diagnostic methods for limited dataset have also been continuously studied [22], [23]. With aim to improve the model accuracy and effectiveness for bearing fault diagnosis as well as reduce its computation and storage costs, a lightweight convolutional neural network (LCNN) for intelligent diagnosis of bearing faults is proposed in this paper. The key of this study is how to effectively reduce the model parameters and storage space, and how to construct an optimal diagnostic model to achieve high accuracy. As for the proposed LCNN method, a novel decomposed Hierarchical Search Space is introduced to explore the optimal network for bearing fault diagnosis.

By introducing deep separable convolution in place of traditional convolution, and introducing inverse residual structure and linear bottleneck layer, the computational and storage costs of the model are reduced while its accuracy is improved. The convolution operation is performed on the image data of the fault samples to capture the non-linear structure and fault trend, and the physical interpretation of the extracted features as well as the model recognition results is provided through visualization. The main contributions of this paper are as follows:

1. A LCNN model is constructed via lightweight convolution blocks rather than the traditional convolution operations, which highly improves the fault diagnosis accuracy, and largely reduces the calculation amount and model storage. Moreover, with significantly high accuracy, it effectively solves the problem of serious overfitting and vanishing gradient caused by the model deepening in limited samples.

2. A novel decomposed Hierarchical Search Space is adopted for model optimization to balance accuracy and parameters. Through decomposed Hierarchical Search Space for automatic search on the dataset for the optimal model for bearing fault diagnosis, the constructed LCNN greatly reduces the dependence on experience in the model construction process.

3. Since the CNNs are “black box”, Tensorborad is used to visualize the feature extraction results of each convolutional layer and distribution stochastic neighbor embedding (t-SNE) is applied to visualize the learned features in hidden fully connected layer. In this sense, the visualization of the entire model is achieved.

II. LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORK

A. CONVOLUTIONAL NEURAL NETWORK

As one of important branches of deep learning, CNNs excel in the field of pattern recognition as a result of their excellent feature capture capabilities [24]. A basic CNN includes an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. As a matter of fact, its essence is to construct multiple filters to convolve and pool the input data layer by layer, and extract their features layer by layer [25]. Its unique network structure can effectively reduce the number of training parameters, thus reducing the complexity of the network. Meanwhile, it guarantees invariance of translation, rotation and scaling to a certain degree.

The convolution layer consists of multiple convolution kernel filters. The kernel filter convolves with the child nodes of the input layer and then outputs the results. Each kernel filter repeatedly acts on its entire neuronal receptive field, performs a convolution operation on the pre-processed input feature map, and then uses the activation function to output the convolution result to form a feature map for local features extraction of the input feature map. Each convolution kernel filter consisting of weights \( w^{(i,j)} \) in the convolution layer does the convolution operation with the local data \( a^{(i,j)} \) in the \( l \) layer, and then outputs \( z^{(i,j)} \) through the activation function.
The convolution kernel filter convolves all the input data one by one according to the step set by the model. The mathematical expression of CNN’s convolution operation is as follows:

\[ x_{i}^{out} = f_{cov}(\sum_{i \in M} x_{i}^{input}, k_{ij} + b_{j}) \]  

(1)

where \( x_{i}^{input} \) is the \( i \)-th input feature map in the convolution layer, \( k_{ij} \) is the \( j \)-th weight matrix of the corresponding convolution layer, \( b_{j} \) is the \( j \)-th bias term of the convolution layer, \( M \) is the collection of input feature maps, \( f_{cov} \) is the activation function, and \( x_{i}^{out} \) is the output feature map.

The pooling layer which is also composed of a convolution kernel filter, is usually set after the convolution layer. The calculation in the pooling layer kernel filter is not a weighted sum of neuron nodes, but a maximum or average operation. The purpose of the pooling layer is to perform the function of secondary feature extraction. By reducing the length, width, and depth of the feature map matrix, the preset parameters are reduced, thus increasing the calculation rate. Different from the convolution operation of the convolution layer, that of the single convolution kernel filter of the pooling layer is exclusively for nodes at one depth, so the convolution operation of the sampling layer not only takes samples in the length and width directions of the input data, but also in the depth direction. During the dimensionality reduction, the convolution operation of the sampling layer is equivalent to the secondary feature extraction of the input data. The sampling mathematical expression of pooling layer convolution filter is as follows:

\[ x_{il}^{out} = f(x_{ip}^{input}, x_{i(p+1)}^{input}, \ldots) \]  

(2)

where \( x_{ip}^{input} \) is the node value of the \( p \)-th neuron of the \( i \)-th input feature map in the pooling layer, \( x_{i(p+1)}^{input} \) is the node value of the \( (p+1) \)-th neuron of the \( (i+1) \)-th input feature map in the pooling layer \( f() \) is the maximum value function or the average function, and \( x_{il}^{out} \) is the node value of the \( l \)-th neuron of the \( i \)-th output feature in the pooling layer.

The fully connected layer is a classification module of the model. It can map the distributed features extracted by the convolution layer and the pooling layer to the target space, namely transforming from a high-dimensional space to a low-dimensional space. And it is fully connected with the previous layer, so there are more layer parameters than other layers. The feature map of the previous layer of the fully connected layer is “rolled out” in turn through the convolution operation, and then activated by the activation function Softmax for classification output. The mathematical expression of the forward propagation of the fully connected layer is as follows:

\[ a_{i}^{out} = f_{fc}(\sum_{i \in M} a_{i}^{in}, w_{ij} + b_{j}) \]  

(3)

where \( a_{i}^{in} \) is the output feature map of the layer previous to the fully connected layer, \( a_{i}^{out} \) is the output of the fully connected layer, \( w_{ij} \) is the weight between the neurons of different layers, \( M \) is the collection of input feature maps, \( b_{j} \) is the bias term, and \( f_{fc} \) is the activation function of the fully connected layer.

The fully connected layer adopts the Softmax activation function for mapping to achieve multi-classification of data. The Softmax mathematical expression is as follows:

\[ f(z_{i}) = \frac{\exp(z_{i})}{\sum_{i=1}^{I} \exp(z_{i})} \]  

(4)

where \( z_{i} = \sum_{i \in M} a_{i}^{in} \cdot w_{ij} + b_{j} \) is the output logits value of the neurons in the fully connected layer.

B. LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORK

Traditional CNNs usually increase the model depth and complexity in pursuit of high accuracy. But such large complicated models can never be applied to the real scenarios such as mobile or embedded devices. As the improved version of the CNN, LCNN operates with lightweight convolution instead of the traditional convolution. While maintaining the performance of the model, the model size is largely reduced and the model speed is increased. Therefore, it effectively reduces the calculation and storage load of the model, which is more conducive to the application of the mobile terminal and the Internet of Things context [26], [27].

1) DEPTHWISE SEPARABLE CONVOLUTION

Depthwise separable convolution is one of typical representatives of lightweight convolution. After different convolution kernels are conducted on different input channels, experiments performed by F. C. et al in Xception proved that depthwise separable convolution can be applied to DCNN on a large scale [28]; In MobileNet, due to the large-scale use of depthwise separable convolution, a large number of parameters and calculations are reduced, thus accelerating the inference speed of the model [29]. Moreover, under the same conditions, the accuracy loss of the model can be ignored.

![FIGURE 1. The schematic diagram of traditional convolution and depthwise separable convolution: (a) traditional convolution; (b) depthwise separable convolution.](image-url)
convolution. In depthwise convolution, the same number of convolution kernels with the input channels are used for layer-by-layer convolution along the depth of the feature map, but the convolution results are not aggregated. Thus this is an extreme grouping convolution process. Pointwise convolution uses the same number of $1 \times 1$ convolutions with the output convolution kernels to fuse multiple convolution results of the first step.

### TABLE 1. Parameter and calculation amount of traditional convolution and depthwise separable convolution.

|                | Traditional convolution | Deep separable convolution |
|----------------|-------------------------|----------------------------|
| Parameter amount | $D_k \times D_k \times M \times N$ | $D_k \times D_k \times M + 1 \times 1 \times M \times N$ |
| Calculation amount | $D_k \times D_k \times M \times N \times F \times D_f$ | $D_k \times D_k \times M \times D_f \times D_f + M \times N \times D_f \times D_f$ |

The calculation of the parameters and calculation amount of traditional convolution and depthwise separable convolution are shown in Table 1. Suppose that $N \times D_k \times D_k$ feature views and a $1 \times 1$ convolution sliding step are used to convolve a feature map with dimensions $D_F \times D_F \times M$, and the output feature map dimensions are $D_F \times D_F \times N$.

As can be seen from the Table 1, the ratio of the two operations is:

$$
\frac{D_k \times D_k \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_k \times D_k \times M \times N \times D_F \times D_F}
$$

(5)

2) DEPTHWISE CONVOLUTION

Depthwise convolution is a special grouping convolution. Grouping convolution was first proposed by Hinton et al. [30]. Limited by hardware resources, the networks that needed to be trained could not be processed in the same GPU. Therefore, A. K. divided the convolution operation with a large amount of computation in the convolutional layer into two groups respectively. The calculations were done in two pieces of hardware, and the results were then fused in subsequent layers. In the first step of depthwise separable convolution, the extreme grouping convolution is used. The schematic diagram of grouping convolution is shown in Fig. 2. Use the parameters in (1), and assume the number of groups is $G$, the respective parameter amount and calculation amount of the group convolution are $\frac{1}{G} \times D_k \times D_k \times M \times N$ and $\frac{1}{G} \times D_k \times D_k \times M \times N \times D_F \times D_F$. It can be seen that the parameter amount and calculation amount of the grouping convolution are $\frac{1}{G}$ of the traditional convolution. If $G$ is set as the number of channels of the input feature map, then the grouping convolution is depthwise convolution.

3) POINTWISE CONVOLUTION

When the size of the convolution kernel in the convolution layer is set as $1$, it is pointwise convolution. The operation of a $1 \times 1$ convolution kernel on a single feature map is multiplied by a coefficient, which is equivalent to a scaling of the input, so the main feature of pointwise convolution is to achieve a linear combination of multiple feature maps as well as achieve cross-channel information integration. Pointwise convolution is usually used to control the number of channels in the convolution layer. Before performing a large convolution kernel, pointwise convolution is adopted to reduce the dimensionality of the data. After using the large convolution kernel, pointwise convolution is used for data expansion, which can reduce the amount of calculations and parameters while increasing the depth of the network. The specific comparison can be seen in Table 2. Suppose a $k \times k$ convolution kernel is used to extract features from the input feature maps of $N, H, W, \text{ and } C$. The convolution step is $1$, and the size of the output feature map is also $N, H, W, \text{ and } C$. Halve the number of channels to $\frac{N}{2}$ when using pointwise convolution. The words “compress” and “enlarge” in the Table 2 indicate that the pointwise convolution is used to compress or expand the number of channels. In the above task, pointwise convolution is used to compress or expand the number of channels.

### III. PROPOSED INTELLIGENT FAULT DIAGNOSIS METHOD

A. DESIGN OF THE LCNN FOR BEARING FAULT DIAGNOSIS

LCNN is usually constructed by the depthwise separable convolution, but this kind of model cannot perfectly meet the requirement posed by the fault diagnosis under IIoT context. Therefore, the residual structure with residual block is introduced to the optimal LCNN model construction [31], since it can effectively solve the problem of vanishing gradient of the network as the number of layers increases, and in the meantime improve the accuracy of the network.
The process of the standard residual block is presented in Fig. 3 (a). The input image first undergoes a $1 \times 1$ convolution with a dimensionality reduction of 0.25 times, and after a standard convolution, uses a $1 \times 1$ convolution to increase the dimension. The execution process of the inverted residual block is shown in Fig. 3 (b). After a $1 \times 1$ convolution is performed to increase the dimension by 6 times, the convolution is deeply decomposed and then the dimension is reduced. Depth-wise convolution is usually expected to be performed on high-dimensional inputs during feature extraction, thus improving the expression ability of the model. In a word, the role of the inverted residual module is to transform the input data to high dimensions and then extract features through deep decomposition.

Since depthwise convolution cannot change the input data channel, the dimension where it extracts features depends on the data input in the previous layer. To solve this problem, before the depthwise convolution, a $1 \times 1$ convolution is added to change the input data dimension, which makes it possible for depthwise convolution to extract features in high-dimensional space. It is generally believed that the activation function can effectively increase non-linearity in high-dimensional space, while destroying non-linearity in low-dimensional space. The second $1 \times 1$ convolution after deep convolution is used for dimensionality reduction. The method manifested in the linear bottleneck [10] is to remove the second activation function after $1 \times 1$ convolution and use a linear activation function to keep the features extracted by the convolution [32].

A basic LCNN module can be constructed via the depthwise separable convolution, inverse residual structure, and linear bottleneck layer. Through the network stacking, the basic LCNN is completed. The basic convolution block and basic LCNN structure are shown in Fig. 4.

### B. NETWORK OPTIMIZATION

In order to construct an optimal LCNN model suitable for bearing fault diagnosis, a novel decomposable Hierarchical Search Space is introduced. In other words, the above model is decomposed into different blocks and then the respective operations and the block-block connection relations are searched for each block. In this sense, the various layer structures are allowed for various blocks [33]. For the widely used depthwise separable convolution, we input the feature map size as $(H, W, M)$, and use the convolution kernel with the dimension $(K, K, M)$ for depthwise convolution to output feature map size $(H, W, M)$. After that, we use the convolution kernel with dimension $(1,1, M, N)$ to conduct pointwise convolution. In this case, $(H, W)$ is the input resolution, and $\frac{M}{N}$ is the number of convolution kernels. The total number of multiply-add operations is:

$$H \times W \times M \times (K \times K + N) \quad (6)$$

When the overall computing resources are limited, we need to carefully balance the kernel size $K$ and the number of filters $N$. For example, to increase the receptive field, you must reduce the number of convolution kernels $N$ while increasing the kernel size $K$ in the convolution layer, or reducing the calculation amount of other layers.

Fig. 5 demonstrates the baseline structure of the search space. The LCNN model is divided into a set of predefined blocks. The input resolution is gradually reduced and the number of convolution kernels is increased. Each block has a column of the same layer. The operations and connections are determined by the sub-search space of each block. Specifically, the sub-search space for a block $i$ consists of the following choices:

- Convolution operation: regular convolution, depthwise separable convolution;
- Convolution kernel size;
- Squeeze-and-Excitation ratio(SE);
- Skip operation: pooling, residual block, no skip;
- Output filter size $F_i$ and the number of layers $N_i$;

For instance, in Fig. 5, each layer of block 4 has an inverted $5 \times 5$ bottleneck convolutional layer, and identical residuals. And such a layer is repeated $N_4$ times. For the bearing fault data set, the decomposable Hierarchical Search Space is used to automatically perform network connection.
Y. Wang et al.: Bearing Intelligent Fault Diagnosis in the IIoT Context

C. PROPOSED DIAGNOSIS FRAMEWORK

To sum up, a novel model for bearing fault diagnosis under the IIoT context is proposed in this paper. A LCNN-based fault diagnosis framework for bearing fault diagnosis is summarized in the following three steps.

1) Network structure construction. A basic lightweight convolution block is constructed by basic elements such as deep separable convolution and inverse residual block. During the LCNN model construction process, the basic modules constructed above are used for stitching.

2) Network optimization. The novel decomposed Hierarchical Search Space is used to decompose the above model into different blocks and then search for respective operations and the block-block connection relations for each block, which allows the various layer structures for various blocks. An optimal LCNN model for bearing fault diagnosis is constructed through network optimization.

3) Model deployment. Extract the trained LCNN to construct the fault diagnosis model. Deploy the fault diagnosis model on bearings for real time monitoring and fault diagnosis.

The LCNN-based bearing fault diagnosis process in the IIoT context is shown in Fig. 7. It mainly includes data preprocessing, data set partitioning, data enhancement, model training, model testing, and model visualization. In the device-level control part of the proposed method, data is first collected, and then the system makes decisions based on the collected data. With reference to [34], this paper intends to deal with noise in the dataset. In the process of data processing, the original vibration signal is converted into a two-dimensional picture for processing through Gra mainland Angular Field method [35]. Data preprocessing includes image normalization processing and scale transformation. In order to verify the validity of the model, Tensorboard is used as a visual tool for model training process and features.

IV. EXPERIMENTAL VERIFICATION

A. CASE STUDY ON CASE WESTERN RESERVE UNIVERSITY BEARING FAULT DATASET

1) EXPERIMENT DESCRIPTION

As one of the most widely used and representative data in mechanical fault diagnosis, Case Western Reserve University Bearing fault dataset is the best dataset to verify the excellent characteristics of the proposed method [36]. The bearing test-rig is shown in Fig. 8. The testing board consists of a...
TABLE 3. Parameter and calculation amount of pointwise convolution.

| Fault location | Inner race | Ball | Out race @6 o’clock | O |
|---------------|------------|------|---------------------|---|
| Category      | 0 1 2 3 4 5 6 7 8 9 |
| labels        | 0 0 0 0 0 0 0 0 0 0 |
| Dataset A     | 0 0 0 0 0 0 0 0 0 0 |
| Fault         | 0 1 2 0 1 2 0 1 2 0 |
| diameter(in)  | 7 4 1 7 4 1 7 4 1 7 |
| Dataset B     | 0 0 0 0 0 0 0 0 0 0 |
| Fault         | 0 1 2 0 2 2 0 1 2 0 |
| diameter(in)  | 7 4 1 7 1 8 7 4 8 8 |

2-horsepower motor (in the left), a torque sensor (in the center), a dynamometer (in the right), and control electronics (not shown). Fault diameters are set to 7 mils, 14 mils, 21 mils, 28 mils, and 40 mils. Accelerometers are placed at twelve o’clock position on the drive and fan sides of the motor housing for vibration data collection. The data of normal bearings, single-point drive end, and fan end defects are respectively collected. Data are collected at 12,000 samples / second and 48,000 samples / second drive end bearing experiments. All fan-end bearing data are collected at 12,000 samples / second.

In this paper, 12k Drive End Bearing Fault Data is selected to construct a bearing fault diagnosis dataset. The vibration data is collected by an accelerometer installed on a housing with a magnetic base. The accelerometer is installed at the 12 o’clock position of the DE bearing. The normal data, namely 7 mil, 14 mil, 21 mil, 28 mil Inner Race fault, 6 o’clock Outer Race fault and Ball fault is selected to construct 10 types of fault data for health status classification. The dataset is shown in the Table 3. Each type of health status consists of 2356 sets of fault samples, and each fault sample contains 200 original time signal points.

2) RESULTS

During the experiment, 80% of the 10 types of fault samples are used as the training set (70% for model training and 10% for cross-validation), and the remaining 20% is used as the testing set. At the same time, we trained our model with Keras (Tensorflow backend) on a machine that has a GeForce RTX 2060 GPU, an Intel i7-8700 CPU and 16 gigabytes of RAM. In order to verify the excellent performance of the LCNN model proposed in this paper, SVM and traditional CNNs, such as LeNet, AlexNet, traditional convolutional neural network with the same number of layers as LCNN in this paper (TCNN), ResNet, and ShuffleNet [37] are used for comparison. The diagnosis results are presented in Table 4.

As can be seen in Table 4, the diagnostic accuracy of the LCNN model on dataset A and B is respectively 97.386% and 100%, significantly higher than that of the other models. Specifically, the fault diagnosis accuracy of SVM on dataset A and B is 94.367% and 97.037%, LeNet’s accuracy is 13.438% and 10.349%, AlexNet’s is 10.382% and 12.157%, TCNN’s is 12.462% and 9.585%, ResNet’s is 83.285% and 89.587%, and ShuffleNet’s is 93.094% and 96.175%. Moreover, LeNet, AlexNet, TCNN all cannot be trained as the vanishing gradient, which results from the limited sample numbers. The training curves of LCNN, ShuffleNet, ResNet and the LeNet proposed in this paper are shown in Fig. 9.

It can be seen from Fig. 9 that although the training of the LCNN proposed in this paper has large fluctuations, the training and cross-validation accuracy is close to 1 when the model is stable. Although the training process of the ShuffleNet network is more stable, the cross-validation accuracy is only about 0.85, and the model learning is obviously insufficient. The training accuracy and cross-validation accuracy of the LeNet network is only about 0.1 and it does not increase with the number of iterations. As can be seen from Fig. 9 (d), as the number of iterations increases, the training accuracy of LeNet continues to improve, but its cross-validation accuracy tends to 0.1 and does not change. It is further verified that limited by the sample size, the model suffers overfitting because of gradient vanishing. Since the training process curve of AleNet is similar to that of LeNet, it will not be given here. It is further verified that the model vanishing gradient. In order to further verify the outstanding recognition results of the LCNN model in this paper, the confusion matrix of LCNN, ShuffleNet and the ResNet model are shown in Fig. 10. It can be seen that the diagnostic accuracy of the LCNN model is optimal. In addition, the difference between dataset A and B.
is mainly caused by the mixed failure of inner race fault and ball fault under the 14 mil fault size of data A.

**TABLE 5.** Other performance indicators.

| Model    | Parameters (Million) | Weight storage (MB) | Training time (minutes) | Testing time (seconds) |
|----------|----------------------|---------------------|-------------------------|------------------------|
| LCNN     | 3.06                 | 12.522              | 27.058                  | 37.67                  |
| ShuffleNet | 0.97                | 4.385               | 16.035                  | 36.75                  |
| ResNet   | 11.18                | 43.647              | 22.047                  | 39.62                  |
| LeNet    | 14.43                | 56.396              | -                       | 27.44                  |
| AlexNet  | 24.72                | 96.695              | -                       | 28.34                  |
| TCNN     | 107.01               | 418.171             | -                       | 41.16                  |

In the IIoT context, factors such as the calculation amount and storage space to a large degree determine the models’ availability. In Table 5, other performance indicators such as the weight storage and parameters of the above models are listed in detail. As is shown in Table 5, ShuffleNet possesses the smallest amount of parameters and least storage space, followed by the LCNN model proposed in this paper, which is 1-2 orders of magnitude smaller than other models. In spite of the longest training time, the performance of the LCNN model is not influenced due to the offline update of model. As for the testing time, LeNet takes the shortest time, followed by AlexNet, ShuffleNet, LCNN, ResNet, and TCNN.

Considering multiple indicators such as diagnostic accuracy, model parameters, and storage, it can be seen that the LCNN model proposed in this paper has state of art performance on Case Western Reserve University Bearing fault dataset.

CNN is usually called a “black box” model. In order to give an intuitive explanation of the model prediction, a t-distribution stochastic neighbor embedding (t-SNE)
method is adopted to visualize the learned features in the hidden fully connected layer. The visualization results of the LCNN, ShuffleNet, and ResNet are presented in Fig. 11. In the flat hidden fully connected layer, samples under the same fault condition are clearly collected together and even separated, indicating the good feature representation capabilities of the learned feature descriptors. After performing non-linear mapping in the classifier, the characteristics of different fault conditions are well separated in the fully connected layer hidden in the last layer, despite the slight overlapping of the single samples, which is consistent with the diagnostic accuracy in Fig. 10. Furthermore, as to the data under invisible conditions, we can observe a larger overlapping area in the ResNet model, while the features learned using LCNN have better separability.

In order to represent the features extracted by each layer of the convolution operation, as well as give a physical interpretation of each layer feature, TensorBoard (TensorFlow, USA) is adopted to visualize the features. Fig. 12 shows the feature maps of some layers. The feature maps indicate that the learned convolution filter is relatively smooth in space, suggesting that training is adequate. In the initial process, the convolution layer mainly extracts the outline of the vibration signal, but then the features gradually become more abstract.

B. CASE STUDY ON MFPT BEARING FAULT DATASET

1) EXPERIMENT DESCRIPTION

The MFPT bearing fault dataset, as another typical dataset for mechanical bearing fault diagnosis, is one of the important references for verifying the performance of the proposed method [38]. The MFPT bearing fault dataset was provided by the American Society for Mechanical Failure Prevention Technology with the data prepared by Dr. E. B., chief engineer of NRG Systems. This dataset includes data from the bearing testing stand (baseline bearing data, outer race failure under various loads, inner race failure under various loads) and three actual failures.

In order to ensure the samples' balance, we construct the bearing fault dataset via 3 baseline conditions (Nor), 7 inner race fault conditions (IR), and 7 outer race fault conditions (OR) to verify the performance of LCNN. The data consists of the following data points: N with 1,757,808 data points, IR with 1,025,388 data points, OR with 1,025,388 data points, and the 97656 data sampling rate. The total image settings generated from data are as follows: Nor 1800, IR 2100, OR 2100, and the time step in each picture is 0.01 seconds. Three types of sample are shown in Fig. 13.

2) RESULTS

In order to verify the performance of the LCNN on the MFPT bearing fault dataset, 70% of the data is randomly selected as the training set, 10% as the validation set for model training, and the remaining 20% for testing. The diagnostic results of each model are shown in Table 6. As can be seen from Table 6, the recognition rate of the LCNN is up to 99.92%. LeNet, AlexNet, and TCNN are directly trained, but their diagnostic effect is extremely poor, AlexNet and TCNN in particular.
TABLE 6. Diagnosis results of MFPT bearing fault dataset.

| Index | Model   | Defects type | Overall |
|-------|---------|--------------|---------|
|       |         | Nor(%) | IR(%) | OR(%) | (%) |
| 1     | LCNN    | 100.00  | 99.76 | 100.00 | 99.92 |
| 2     | ShuffleNet | 100.00  | 78.33 | 81.90 | 86.74 |
| 3     | ResNet  | 100.00  | 95.23 | 98.10 | 97.78 |
| 4     | LeNet   | 82.62   | 63.57 | 69.76 | 71.98 |
| 5     | AlexNet | 34.52   | 33.33 | 34.04 | 33.96 |
| 6     | TCNN    | 34.76   | 33.81 | 36.19 | 34.92 |

TABLE 7. Other performance indicators.

| Model   | Parameters (Million) | Weight storage (MB) | Training time (minutes) | testing time (seconds) |
|---------|----------------------|----------------------|-------------------------|------------------------|
| LCNN    | 3.99                 | 16.42                | 7.36                    | 19.42                  |
| ShuffleNet | 0.97               | 4.37                 | 5.32                    | 22.24                  |
| ResNet  | 11.18                | 43.84                | 5.91                    | 17.85                  |
| LeNet   | 14.43                | 56.38                | -                       | 14.17                  |
| AlexNet | 24.72                | 96.59                | -                       | 15.88                  |
| TCNN    | 107.01               | 418.171              | -                       | 17.57                  |

It is possible that the gradient disappearance make it difficult for models to be trained with the limited data samples.

Other performance indicators of the models are shown in Table 7. The model parameters and storage are consistent with those in IV(A). The training time and testing time are consistent with IV(A). After comprehensively considering multiple indicators such as diagnostic accuracy, model parameters, and storage, it can be seen that the LCNN proposed in this paper has state of art performance on the MFPT bearing fault dataset, which further validates the powerful advantages of the proposed model for bearing fault diagnosis.

FIGURE 14. The t-SNE results of different model.

The t-SNE results of LCNN, ShuffleNet, and ResNet model is shown in Fig.14. The three types of samples are clearly separated, and the classification boundary and region of the LCNN are more obvious, further verifying its optimal performance in recognition accuracy. The features extracted by the convolution layer are shown in Fig. 15. In the initial process of the convolution feature extraction, the convolution layer mainly extracts the outline of the vibration signal, but the subsequent features gradually become more abstract.

V. CONCLUSION

This paper proposes a novel LCNN model for intelligent bearing fault diagnosis. The proposed method can be mainly divided into three steps: first, construct LCNN on the basis of basic operations such as depthwise separable convolution, inverse residual structure, and linear bottleneck. Secondly, the novel decomposed Hierarchical Search Space decomposes the above model into different blocks and then searches for the operation and the block-block connection relationship for each block to automatically explore the optimal LCNN for bearing fault diagnosis in the IIoT context. Finally, model training is performed, and the learned deep features are input into the Softmax classifier to achieve accurate and stable diagnosis of bearing faults.

The proposed LCNN is applied to the fault diagnosis of bearing vibration experimental data. The results demonstrate that this method overcomes the dependence on artificial feature extraction of traditional deep learning models, and in the meantime tackles the traditional CNNs’ dependence on the sample size, low diagnostic accuracy, large storage and calculation costs. Therefore, it is more effective and robust than current intelligent diagnostic methods. The high accuracy and small storage and calculation cost make the proposed LCNN more suitable in the IIoT context. However, it is a problem that needs to be considered and solved in the future as to how to optimize the training and testing time of the model.

REFERENCES

[1] R. Liu, B. Yang, E. Zio, and X. Chen, “Artificial intelligence for fault diagnosis of rotating machinery: A review,” *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, Aug. 2018.

[2] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong, and M. H. Lim, “Low-speed bearing fault diagnosis based on AsrSSEA model using acoustic emission and vibration signals,” *IEEE Access*, vol. 7, pp. 46885–46897, 2019.

[3] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, “Online fault diagnosis method based on transfer convolutional neural networks,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 509–520, Feb. 2020.

[4] W. Deng, S. Zhang, H. Zhao, and X. Yang, “A novel fault diagnosis method based on integrating empirical wavelet transform and fuzzy entropy for motor bearing,” *IEEE Access*, vol. 6, pp. 35042–35056, 2018.
[5] L. Zhang, J. Lin, B. Liu, Z. Zhang, X. Yan, and M. Wei, “A review on deep learning applications in prognostics and health management,” IEEE Access, vol. 7, pp. 162415–162438, 2019.

[6] Y. Chen, Z. Zhen, H. Yu, and J. Xu, “Application of fault tree analysis and fuzzy neural networks to fault diagnosis in the Internet of Things (IoT) for aquaculture,” Sensors, vol. 17, no. 12, p. 153, Jan. 2017.

[7] Z. Wang, W. Du, J. Wang, J. Zhou, X. Han, Z. Zhang, and L. Huang, “Research and application of improved adaptive MODMEDA fault diagnosis method,” Measurement, vol. 140, pp. 63–75, Jul. 2019.

[8] H. Hu, L. Zhang, H. Yan, Y. Bai, and P. Wang, “Denoising and baseline drift removal method of MEMS hydrophone signal based on VMD and wavelet threshold processing,” IEEE Access, vol. 7, pp. 59913–59922, 2019.

[9] T. Han, C. Liu, L. Wu, S. Sarkar, and D. Jiang, “An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems,” Mech. Syst. Signal Process., vol. 117, pp. 170–187, Feb. 2019.

[10] Y. Wang, J. Xiang, R. Markert, and M. Liang, “Spectral Kurtosis for Fault Detection, Diagnosis and Prognostics of Rotating Machines: A Review with Applications,” Mech. Syst. Signal Process., vols. 66–67, pp. 679–698, Jan., 2016.

[11] T. Wang, Q. Han, F. Chu, and Z. Feng, “Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review,” Mech. Syst. Signal Process., vol. 126, pp. 662–685, Jul. 2019.

[12] J. Lei, C. Liu, and D. Jiang, “Fault diagnosis of wind turbine based on long short-term memory networks,” Renew. Energy, vol. 133, pp. 422–432, Apr. 2019.

[13] M. Wollschaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0,” IEEE Ind. Electron. Mag., vol. 11, no. 1, pp. 17–27, Mar. 2017.

[14] Q. Qi and F. Tao, “Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison,” IEEE Access, vol. 6, pp. 3585–3593, 2018.

[15] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, “Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification,” Signal Process., vol. 130, pp. 377–388, Jan. 2017.

[16] Y. Li, G. Cheng, C. Liu, and X. Chen, “Study on planetary gear fault diagnosis based on variational mode decomposition and deep neural networks,” Measurement, vol. 130, pp. 94–104, Dec. 2018.

[17] S. Guo, T. Yang, W. Gao, and C. Zhang, “A novel fault diagnosis method for rotating machinery based on a convolutional neural network,” Sensors, vol. 18, no. 5, p. 1429, May 2018.

[18] S. Shao, R. Yan, Y. Lu, P. Wang, and R. Gao, “DCNN-based multi-signal induction motor fault diagnosis,” IEEE Trans. Instrum. Meas., early access, Jul. 1, 2019, doi: 10.1109/TIM.2019.2925247.

[19] W. Yu and C. Zhao, “Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability,” IEEE Trans. Ind. Electron., vol. 67, no. 6, pp. 5081–5091, Jun. 2020.

[20] Y. Wu, B. Jiang, and N. Lu, “A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 49, no. 10, pp. 2108–2118, Oct. 2019.

[21] Y. Wu, B. Jiang, and Y. Wang, “Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains,” ISA Trans., vol. 99, pp. 488–495, Apr. 2020.

[22] L. Song, C. Wang, L. Zhang, B. Du, Q. Zhang, C. Huang, and X. Wang, “Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains,” Pattern Recognit., vol. 102, pp. 107173-1–107173-11, 2020.

[23] Z. Wang, B. Du, and Y. Guo, “Domain adaptation with neural embedding matching,” IEEE Trans. Neural Netw. Learn. Syst., early access, Sep. 13, 2019, doi: 10.1109/TNNLS.2019.2935608.

[24] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, “Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,” Mech. Syst. Signal Process., vols. 72–73, pp. 303–315, May 2016.

[25] F. Jia, Y. Lei, N. Lu, and S. Xing, “Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization,” Mech. Syst. Signal Process., vol. 110, pp. 349–367, Sep. 2018.

[26] T.-W. Hui, C. C. Loy, and X. Tang, “Depth map super-resolution by deep multi-scale guidance,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 353–369.

[27] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. R. Faughnan, “Real-time human detection as an edge service enabled by a lightweight CNN,” in Proc. IEEE Int. Conf. Edge Comput. (EDGE), Jul. 2018, pp. 125–129.

[28] F. chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1251–1258.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861

[30] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012, arXiv:1207.0580. [Online]. Available: http://arxiv.org/abs/1207.0580

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, and Z. Wojna, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 4278–4284.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4510–4520.

[33] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, “MnasNet: Platform-aware neural architecture search for mobile,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2820–2828.

[34] F. Mohammadi and C. Zheng, “A precise SVM classification model for predictions with missing data,” in Proc. 4th Nat. Conf. Appl. Res. Elect. Mech. Comput. IT Eng., 2018.

[35] Z. Wang and T. Oates, “Encoding time series as images for visual inspection and classification using tiled convolutional neural networks,” in Proc. Workshops 29th AAAI Conf. Artif. Intell., vol. 1, 2015, pp. 40–46.

[36] K. A. Loparo, (2012), Case Western Reserve University Bearing Data Center, [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home

[37] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6848–6856.

[38] (2012), Condition Based Maintenance Fault Database for Testing of Diagnostic and Prognostics Algorithms. [Online]. Available: https://nftp.org/fault-data-sets/
QIFENG SUN was born in Shaoxing, Zhejiang, China, in 1996. He received the B.A. degree in english language and literature from Chongqing University, in 2018. He is currently pursuing the M.A. degree with Xi’an Jiaotong University, Xi’an, China. His current research interests include american literature and english for electrical engineering.

QUIAN JIANG was born in Chuzhou, Anhui, China, in 1995. He received the B.S. degree from Anhui University, in 2017. He is currently pursuing the M.S. degree in electrical engineering with Xi’an Jiaotong University, Xi’an, China. His current research interests include partial discharge detection and fault diagnosis for electrical equipment.

YIZHI ZHOU was born in Xi’an, Shaanxi, China, in 2000. He is currently pursuing the B.S. degree in mechatronic engineering with Northwestern Polytechnical University, Xi’an. His research interests include online monitoring and fault diagnosis for rotating machinery.