A Novel Method of Bearing Fault Diagnosis in Time-Frequency Graphs Using InceptionResnet and Deformable Convolution Networks

SHAOBO LI[1,2], **WANLI YANG**[1], **ANSI ZHANG**[2], **HUIBIN LIU**[2], **JINYUAN HUANG**[1], **CHUANJIANG LI**[2], **AND JIANJUN HU**[1,3]

[1]School of Mechanical Engineering, Guizhou University, Guiyang 550025, China
[2]Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, Guiyang 550025, China
[3]Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA

Corresponding authors: Wanli Yang (yang_wanli1993@163.com) and Jianjun Hu (jianjunh@cse.sc.edu)

ABSTRACT Bearing fault diagnosis has attracted increasing attention due to its importance in the health status of rotating machinery. The data-driven models based on deep learning (DL) have become more and more intelligent in the field of fault diagnosis, and among them convolutional neural network (CNN) has been widely used in recent researches. However, traditional CNN is not easy to capture right fault features due to their fixed geometric structures, especially under complex working conditions in fault diagnosis. To address these challenges, we propose a novel model by combining InceptionResnetV2 with Deformable Convolution Networks, named DeIN. We replace the basic form of convolution with deformable convolution in specific layers, and a main classifier and an auxiliary classifier are designed to output the classification result of our proposed model, to adapt to the non-rigid characters and larger receptive field in time-frequency graph (TFG). Experimentally, the one-dimensional signals are transformed into TFGs and as input of the proposed model, and this aims to find useful features during the training process. To verify the generalization ability of the proposed model, we apply a set of cross-over tests based on two popular datasets, and our model achieved 99.87% and 94.52% highest-precision fault classification results comparing with other state-of-the-art CNN models.

INDEX TERMS Rolling element bearing, fault diagnosis, InceptionResnetV2, deformable convolution, time-frequency graph.

I. INTRODUCTION

As an indispensable component of rotating machinery, rolling bearings always determine the state of machines. Due to complex working environments and conditions, rolling bearing failures are inevitable and unpredictable. These failures severely impair the stability and performance of machines. Therefore, many bearing fault diagnosis methods are proposed. These fault diagnosis methods can be divided into model-driven methods, and data-driven methods [1]. In big data era, data-driven models have become more and more attractive. Deep learning is such a kind of powerful data-driven method that has been successfully adopted in

The associate editor coordinating the review of this manuscript and approving it for publication was Lance Fiondella.

various areas such as computer vision, automatic speech recognition, natural language processing, audio recognition, remaining useful life prediction and bioinformatics [2]–[7].

Deep learning attempts to model hierarchical representations behind data and classify(predict) patterns via stacking multiple layers of information processing modules in hierarchical architectures. Convolutional neural networks (CNNs) proposed by LeCun [8] is characterized by two key properties: spatially shared weights and spatial pooling. CNNs can effectively extract features from massive data. Therefore, in current intelligent fault diagnosis field, CNNs has been widely adopted. Zhang *et al.* [9] proposed WDCNN (wide convolution kernel deep convolutional neural network) based on time domain fault signal, the first layer of large convolution kernel was trained by the optimization algorithm that

could automatically remove the useless noise for diagnosis, and WDCNN allowed the wide convolution kernel to have greater domain adaptability to the fluctuation of the raw signal, which has the advantage of automatically learning for diagnosis features on the time domain. WDCNN also added batch standardization (AdaBN) and ensemble learning of TICNN with training Interference [10] to improve its stability. Due to the convolution kernels of the same size have the same range of receptive fields then as the input to a fixed-size pooling layer for pooling operation, this method makes the feature data often suffered background noise influence. Atrous convolution [11]–[13] adjusts the expansion ratio to change the offset of the convolution kernel, when the dilation ratio [14] equal 1, the convolution characteristics are equivalent to the standard convolution, and when the expansion ratio is more than or equal to 2, a filter 'with holes', which can avoid part of the background noise. Chen *et al.* [15] proposed a deep inception net with atrous convolution (ACDIN) model, and it has excellent performance when tested on the time domain data from natural damaged bearings and trained on the artificial conditions, and we also proposed a small sample learning method of few-shot learning [16] which used Siamese networks base on limited data.

However, for raw fault signals, it is often difficult to extract enough useful signal features for fault classification, especially under varied working conditions. Thus, the method of transforming vibration signal to time-frequency graphs (TFGs) has been proposed, which using short-time Fourier transform(STFT) and TFGs can reflect the characteristic information of bearings [17]–[19]. Zhu *et al.* [20] used the capsule network (ICN) to analyze the TFGs which effectively improved the fault classification accuracy. Zhang *et al.* [21] proposed a 1-D convolutional adversarial network (A2CNN) model to training dataset for transfer learning, the coefficients of the fast Fourier transform(FFT) were trained as the input data of A2CNN, Domain adaptation (DA) was used between different operating conditions to the original labeled data and the target domain information, Zhao *et al.* [22] applied wavelet transform to original signals and combined analysis with other neural network architectures. Yuan *et al.* [23] proposed the Wavelet FCNN method for dealing with the faults of wind turbine blade icing. The discrete wavelet coefficients processing were used in the FCNN model to improve recognition accuracy [24].

Most of the existing studies have been able to achieve a higher accuracy rate, mostly because it depends on the assume – all the data must be in the same working condition and have the same distribution and feature space. While in real world, bearing conditions vary from different working conditions. Meanwhile, classic convolutional neural network is difficult to capture right fault features due to the fixed geometric structures in their building modules [25], [26]. So, deformable convolution and deformable RoI pooling were proposed [26], [27]. These new modules can enhance the transformation modeling capability of CNNs. The experiment in this paper shows that

learning dense spatial transformation in deep CNNs is effective for sophisticated vision tasks such as object detection and semantic segmentation [28]–[30]. Furthermore, in convolutional network, such InceptionResNet, a series of convolution kernel combining different sizes in convolution feature extraction can improve recognition accuracy [31], [32].

In this paper, in order to achieve higher accuracy and better performance in bearing fault diagnosis under varying working conditions, we propose a novel DeIN model, which uses deformed convolution in low-dimensional features to extract useful signal features, and apply InceptionResNetV2 to process the output of deformed convolution. In experiments, DeIN can better adapt to the data of different working conditions with signal features and recognition ability, which method is similar to the attention mechanism and can find features that are valuable to the model during the network training process, and DeIN will be used in a wide range of PHM problems and solve fault diagnosis problems in more fields. In experiments we compare the deep convolution models: ACDIN, WDCNN, AlexNet, ResNet, ICN, IRv2-lite and DeIN. By cross-testing between different conditions, we obtain the generalization ability of each deep convolution model. In particular, we performed multiple experimental comparisons between DeIN and IRv2-lite without deformed convolution and found better generalization of DelN. Due to the similarity in the data distribution of the same type of faults in different working conditions, DeIN is better able to fit the fault type features than conventional convolutional neural networks. The main contributions of this article lie in the following aspects:

(1) The novel DeIN method filters the background noise of signal data automatically by using deformable convolution kernels, we purposefully modify DeIN structure and redefine the loss function. so that it performs better when in different working condition data.

(2) By using comparatively fewer parameters, experimental results showed that DeIN is effective for fault diagnosis both on sufficient and limited data and comparing convolutional neural network models such as WDCNN when the sample is relatively small (5% of the total sample), DeIN has a higher accuracy of 88.54% and WDCNN is 78.32%.

(3) The proposed DeIN, which reaches a high accuracy up to 99.87% and 94.52% on average under different working conditions and equipment.

This rest of the paper is organized as follows: Section II describes the deformable convolution and inception-resnet based fault diagnosis algorithm. Section III presents the experiments and different working condition, results and discussion. Section IV concludes the paper.

II. PROPOSED DeIN ARCHITECTURE
In bearing fault diagnosis, our proposed DeIN architecture is shown in Figure 1. In this model, combining experimental data with real fault data as input, and cross-over testing is used
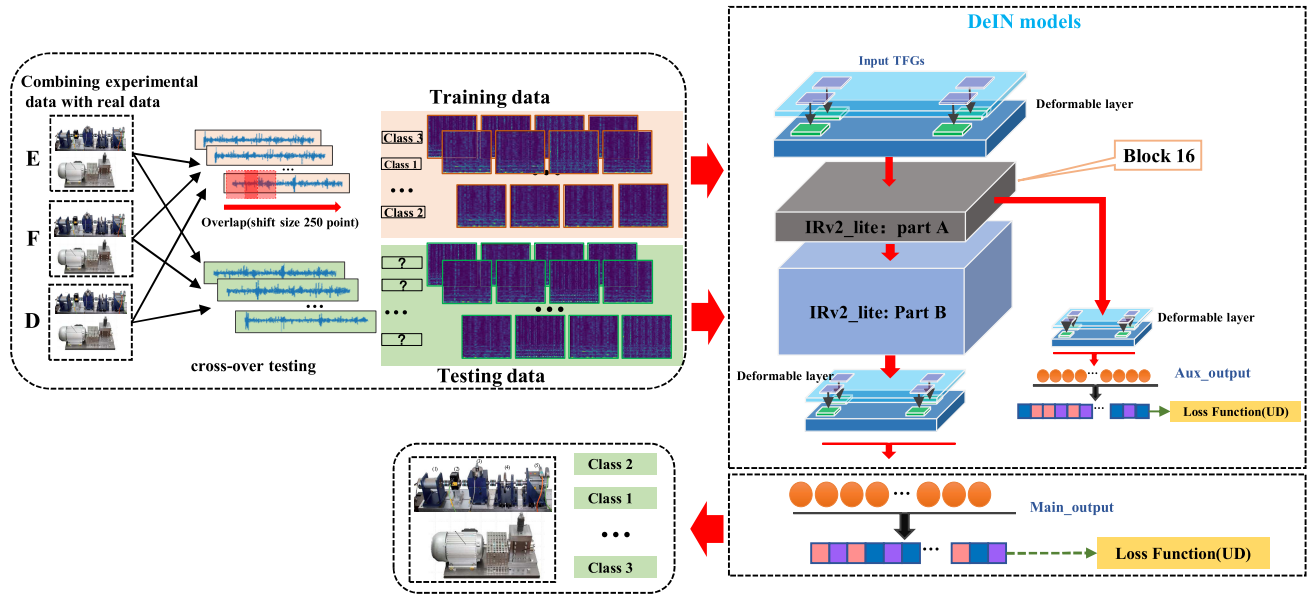
FIGURE 1. The proposed structure of DeIN.

to diagnose bearing faults under different working conditions. DeIN's efficient feature extraction capability is used to fit multiple fault data and improve model's generalization.

A. GENERAL CNN LAYER

The input data is convolved with filters kernel in convolutional layer, the convolution process is described as Equation(1), each filter uses the same kernel size to extract the partial area features of the input data, which is commonly referred to weight distribution $\mathbf{K}_i^l$ and b_i^l denotes bias in $i-th$ filter kernel with layer l, and use $\mathbf{x}^l(j)$ to denote the $j-th$ partial area in the layer l,

$$\mathbf{y}_i^{l+1}(j) = \mathbf{K}_i^l * \mathbf{x}^l(j) + b_i^l \tag{1}$$

the above denotes the input of $j-th$ neuron in frame i of layer $l+1$, the notation * computes the dot product of the kernel and the partial area.

Pooling layers are usually added after the convolutional layer in the CNN architecture. The max-pooling layer is most commonly used, it reduces the parameters and obtains the function of positional invariance. The max-pooling layer described as Equation (2),

$$\mathbf{P}_i^{l+1}(j) = \underset{(j-1)W+1 \le t \le jW}{MAX} \left\{ \mathbf{q}_i^l(t) \right\} \tag{2}$$

where $\mathbf{q}_i^l(t)$ denotes the value of $t-th$ neuron in $i-th$ frame of layer l, and $t \in [(j-1)W+1, jW]$, W denotes pooling area, $\mathbf{P}_i^{l+1}(j)$ denotes pooling operation in layer $l+1$.

After the convolution operation, the Rectified Linear Unit (ReLU), which is widely used as an activation function in deep learning. $\mathbf{y}_i^{l+1}(j)$ denotes the output of convolution operation, $\mathbf{a}_i^{l+1}(j)$ denotes the result of $\mathbf{y}_i^{l+1}(j)$ access the activation

function f, the activation layer is described as Equation (3)

$$\mathbf{a}_i^{l+1}(j) = f\left(\mathbf{y}_i^{l+1}(j)\right) = MAX\left\{0, \mathbf{y}_i^{l+1}(j)\right\} \tag{3}$$

The role of the auxiliary classifier overcomes the disappearance of gradient and provides regularization. An auxiliary classifier can achieve higher level of accuracy and stability than a network without it [15], [33]. According to the data features and training objectives, the model obtains a series of weight values for ground truth with the optimization of the parameters, expressed as Equation (4),

$$\begin{cases} P(W_M) \equiv \left\| w^{(out)} \right\|^2 + \Gamma\left(W_{main}, w^{(out)}\right) \\ Q(W_m) \equiv \alpha\left[\left\| w^{(m)} \right\|^2 + \Gamma\left(W_{aux}, w^{(m)}\right) \right] \\ F(W) \equiv P(W_M) + Q(W_m) \end{cases} \tag{4}$$

where $W_M = \left(W^{(1)}, \ldots, W^{(M)}\right)$ denotes the combining all layers of weights, $W_m = \left(W^{(1)}, \ldots, W^{(m)}\right)$ denotes the corresponding weights of an auxiliary classifier with the mth layer, the classifier weights for the ith layer are denoted as $w^{(i)}$, $P(W_M)$ is the main output objective, and $Q(W_m)$ is the auxiliary output objective.

B. PRINCIPLE OF DEFORMABLE CONVOLUTION

In the convolution operation mentioned above, the standard convolution kernel in Figure 2(a) and the standard convolution kernels are usually a series of fixed shape for data calculation operations. If the size changed, such as Atrous kernel, shown in Figure 2(b), which reduces background effects by increasing the range of receptive field. In addition, if the receptive field could be deformed when the object changes, which makes models more accessible to extract useful feature, shown in Figure 2(c).
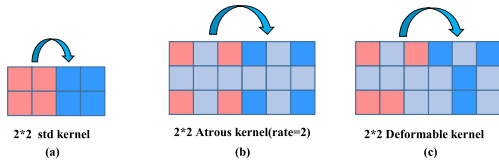
FIGURE 2. Different kinds of convolution kernel, (a) standard kernel. (b) atrous kernel. (c) deformable kernel.
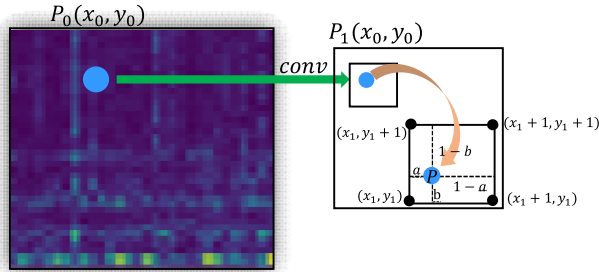

FIGURE 3. Bilinear interpolation.

Furthermore, the principle of deformable convolution can be explained as follows: the bilinear interpolation method makes deformable convolution obtain the offset $\Delta\mathbf{p}_n$ in the training by backpropagation. The offset $\Delta\mathbf{p}_n$ is obtained by bilinear interpolation on the 2-D TFG, shown in Figure 3, and Equation(5)(6).

$$G_1 = \begin{cases} r_1 = (1-a)(1-b)^* p_1\,(x_1+1, y_1+1) \\ r_2 = a(1-b)^* p_1\,(x_1+1, y_1) \\ r_3 = b(1-a)^* p_1\,(x_1, y_1+1) \\ r_4 = ab^* p_1\,(x_1, y_1) \end{cases} \quad (5)$$

$$\mathbf{p} = w_1^* r_1 + w_2^* r_2 + w_3^* r_3 + w_4^* r_4 \quad (6)$$

Through Bilinear interpolation, we get the values of the four integers (r_1, r_2, r_3, r_4) pixel closest to the point P, the weights (w_1, w_2, w_3, w_4) of each integer pixel values are obtained by backpropagation (BP), shown as Equation(7).

$$\mathbf{p} = \mathbf{p}_n + \Delta\mathbf{p}_n \quad (7)$$

The critical part of the deformable convolution method in this research is to obtain the offset $\Delta\mathbf{p}_n$ by training model, and $\Delta\mathbf{p}_n$ denotes as shown in Equation(8)(9).

$$\mathcal{R} = \{(-1,-1), (-1,0), \ldots, (0,1), (1,1)\} \quad (8)$$

$$\{\Delta\mathbf{p}_n | n = 1, \ldots, N\}, \quad \text{where } N = |\mathcal{R}| \quad (9)$$

where grid $\mathcal{R}$ defines the receptive field size in conventional convolution.

For each location $\mathbf{p}_0$ on the output feature map $\mathbf{y}$, the convolution method is expressed as Equation(10).

$$\mathbf{y}\,(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}\,(\mathbf{p}_n) \cdot \mathbf{x}\,(\mathbf{p}_0 + \mathbf{p}_n) + \mathbf{b}_n \quad (10)$$

where $\mathbf{p}_0$ is initial location, $\mathbf{p}_n$ enumerates the fixed locations in $\mathcal{R}$.
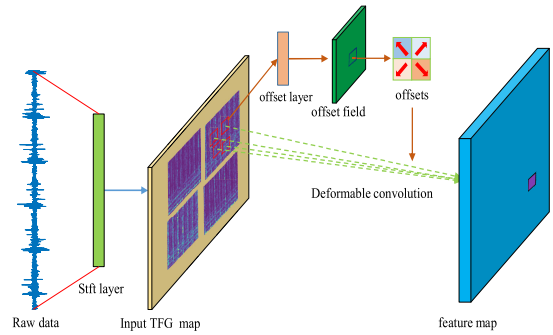

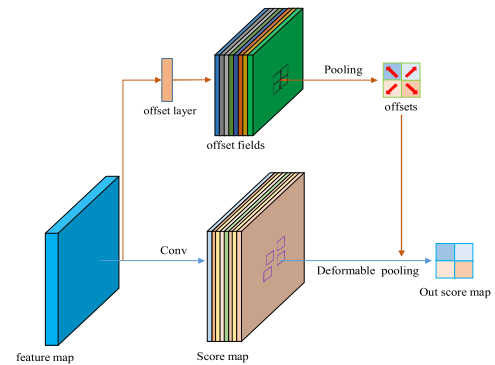**FIGURE 4.** Principle of deformable convolution.

FIGURE 5. Principle of deformable pooling.

In the bearing fault dataset, TFGs are the result of the STFT layer, at the same time, the offset field layers calculate the offset $\Delta\mathbf{p}_n$ of the convolution kernel, which makes the convolution kernel obtains deformable receptive field, shown in Figure 4, and deformable convolution method expression of the convolution kernel by increasing $\Delta\mathbf{p}_n$ could be expressed as Equation (11).

$$\mathbf{y}\,(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}\,(\mathbf{p}_n) \cdot \mathbf{x}\,(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n) + \mathbf{b}_n \quad (11)$$

In convolutional network, the pooling operation used for feature dimensionality reduction, and the number of parameters is reduced to avoid over-fitting while improving the fault tolerance. The pooling layer size is usually fixed in the standard convolution so that the convolved feature map in the pooling operation will result in unavoidable loss of feature. The method of deformable pooling layer is similar to the method of deformable convolution layer, and the shape of the pooling layer is changed by increasing the offset $\Delta\mathbf{p}_{ij}$, shown in Figure 5.

In this paper, we use max-pooling leads to faster convergence rate by selecting superior invariant features which improve generalization performance. Conventional max-pooling method is expressed as Equation (12).

$$\mathbf{y}(i, j) = \underset{\mathbf{p} \in \text{bin}(i,j)}{MAX} \left(\mathbf{x}_{ij}\,(\mathbf{p}_0 + \mathbf{p}) \right) \quad (12)$$
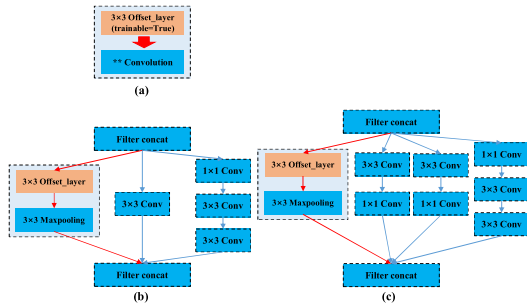
FIGURE 6. Deformable convolution and deformable pooling, (a) is deformable convolution, (b) is inception-Resnet: A layer, (c) is inception-Resnet: B layer.

FIGURE 7. IRv2-lite from inceptionResNetv2, (b) The proposed structure of DeIN.

TABLE 1. Detail parameters of DeIN model.

Layer name	Kernel size/stride	Output shape	F_1	F_2	F_3	F_4	Scale
Convolution(BN)	2×2/2/	32×32×64					
Offset_low	3×3	32×32×64					
Inception-Resnet16	-	32×32×64	32	32	48	64	0.3
Reduction-A	-	8×8×832	256	384	288	128	
Offset_pooling	3×3	16×16×64					
Max_pooling	3×3/1	-					
Convolution(BN)	1×1/1	6×6×128	128				
Dropout	-	-					0.5
Offset_top	3×3	6×6×128					
GlobalMax_pooling	-	128					
Softmax(Aux)	-	3					
Inception-Resnet8	-	8×8×832	192	128	160	192	0.2
Reduction-B	-	4×4×832	256	384	288	32	
Inception-Resnet4	-	4×4×832	192	192	64	256	0.2
Dropout	-	-					0.3
Convolution(BN)	2×2/1	4×4×128					
Offset_top	3×3	4×4×128					
GlobalMax_pooling	-	256					
Softmax(Main)	-	3					

where n_{ij} is the number of pixels in the bins, the $(i, j) - th$ bin denote two vectors of TFGs.

The expression of the deformable max-pooling layer by increasing $\Delta \mathbf{p}_{ij}$ as Equation 13.

$$\mathbf{y}(i, j) = \underset{\mathbf{p} \in \text{bin}(i,j)}{MAX} \left(\mathbf{x}_{ij} \left(\mathbf{p}_0 + \mathbf{p} + \Delta \mathbf{p}_{ij} \right) \right) \qquad (13)$$

C. Inceptionresnetv2 OF DeIN ARCHITECTURE

In inceptionResnetV2, deformable convolution simply expressed as Figure 6(a), meanwhile, inception-ResnetA and inception-ResnetB which include max-pooling layers, but in DeIN, these max-pooling layers become deformable max-pooling layer by adding offset layer, which increased the adaptability of DeIN to data, as shown in the Figure 6(b)(c),

Due to a large parameters of network layers of Inception-Resnetv2 [34], it is not suitable for TFGs rapid diagnosis, considering the fault signal appear periodically in a short time. So, in this paper, the Inception-ResnetV2 was simplified, and the computing resource consumption saved by reducing the number of network layers. IRv2_lite is a retrenched version of inceptionRestnetV2 to improve diagnostic speed.

In DeIN, Block16, Block8 and Block4 replace standard inception Resnetv2 layers, a deformable convolution and pooling layer extract features from TFGs. The deformed convolutional layer placed on the input layer as Offset-low, Aux output, Main output as Offset-top. Among them, in consideration of the fault signal appear periodically in a short time, we put the Aux output followed by block16, and a light network can be used to output the result. Overview of IRv2-lite and DeIN structure illustrated in Figure 7, such as (b), Offset is 'yes' denotes it has deformable pooling layer and detail parameters shown in Table 1,

D. LOSS FUNCTION

We encode labels by one-hot Equation(14), and choose softmax function and cross-entropy as loss function, which are used in the multiple classification task, the loss function is
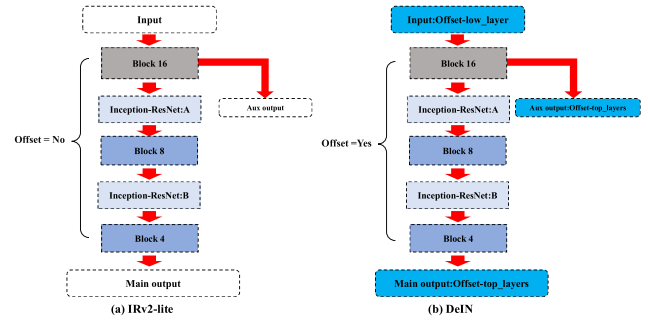
calculated by Equation (15)(16),

$$one - hot : \begin{cases} \begin{bmatrix} [1, 0, 0, ..., 0]_1 \\ [0, 1, 0, ..., 0]_2 \\ ... \\ [0, 0, 0, ..., 1]_n \end{bmatrix} \end{cases} \qquad (14)$$

$$Softmax_i = \frac{e^i}{\sum_{j=1}^{n} e^j}$$

$$loss(q|p) = - \sum_i q_i \log p_i \qquad (15)$$

$$\begin{cases} Z = \sum_{j=1}^{n} e^{x_j} \\ loss = - \log \left(e^{x_i} / Z \right), \quad i \in n \end{cases} \qquad (16)$$

where p_i is the distribution of prediction values, q_i is the distribution of true values, x_j is the output of fully connected (FC) layers, and n denotes n classes.

In the training process, the model output will tend to a specific class due to certain defect of softmax function [35], we introduce the uniform distribution(UD) loss function in the DeIN and optimize it, so that the loss function does not only fit the one-hot distribution, but also consider fitting the uniform distribution [36], [37]. The modified loss function improves the accuracy of the classification and effectively
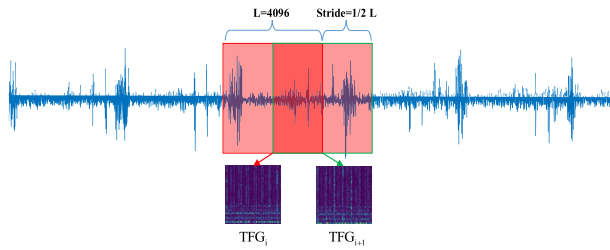
FIGURE 8. TFGs data from STFT.

(a) standard convolution in TFG   (b) DeIN in TFG

FIGURE 9. Standard convolution and DeIN architecture in TFG. (a) standard convolution on TFG. (b) deformable convolution on TFG.

prevents overfitting, which is calculated by Equation 17,

$$loss = -(1 - \varepsilon) \log \left(e^{z_i}/Z \right) - \varepsilon \sum_{i=1}^{n} \frac{1}{n} \log \left(e^{z_i}/Z \right) \quad (17)$$

where ε denotes harmonic parameter, in this research, we set $\varepsilon = 0.1$.

## III. EXPERIMENTAL SETUP AND DATA DESCRIPTION

A. DATA PREPROCESSING

STFT is widely used for time-frequency processing of time-domain signals, STFT uses the sliding window with same stride to extract the signal features. Usually, in order to avoid information redundancy, the stride size is half of the window length, the time-frequency graphs (TFGs) are output of STFT, as shown in Figure 8. The proposed DeIN focuses on the part of interesting feature-area, ignoring redundant data features and background noise. The offset is an important method for performing deformation convolution and makes the extraction area more flexible than the fixed feature extractors in TFGs, as shown in Figure 9. When the model was trained, the offset was updated by each weight-optimized, and then the model accurately extracted the valuable features for the classification task. Furthermore, the standard pooling layer becomes a deformed max-pooling layer by adding an offset, the pooling layer extract interesting feature-area from the output of deformable convolution.

In experimental preparation, the raw data are transformed to time-frequency graphs from a time-domain signal. The shape of the input TFG is 64×64. The experiments we run on a Ubuntu 18.04 operating system computer with 8 kernels Intel Xeon(R) W-2123@3.60GHz CPU, 32GB memory and a NVIDIA GTX 1080ti GPU, the CNN models created based on deep learning were implemented using the Keras 2.0 framework with TensorFlow1.9 as the backend.

B. CWRU DATASET

The test stand of CWRU bearing dataset is shown in Figure 10, the damage of the inner ring, the outer ring, and the ball of the bearing is single point damaged by EDM. Since the position of the outer ring damage is relatively fixed, the damage point is relative to the bearing load area. The different positions have a direct impact on the vibration response of the motor/bearing system. The fault location at 6 o'clock and 48k sample ratio are selected in the study,
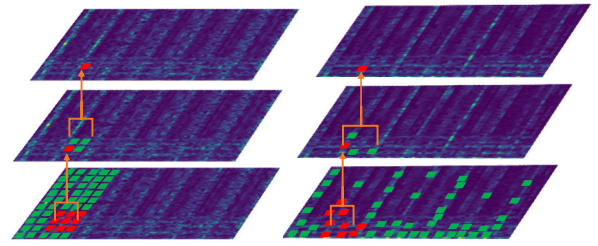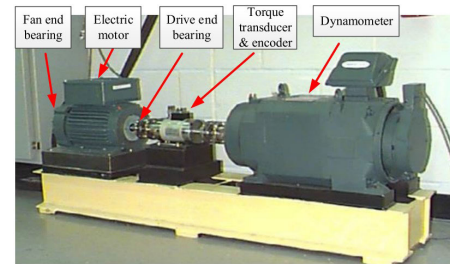
FIGURE 10. Bearing fault test rig used by CWRU.

TABLE 2. Description of rolling bearing on CWRU.

Name	Load/HP	Rotational speed	Damage size/10^{-3} in
A	1	1772[rpm]	7,14,21
B	2	1750[rpm]	7,14,21
C	3	1730[rpm]	7,14,21

and vibration acceleration signal data was recorded under the working conditions of the motor load conditions of 0, 1, 2, and 3 horsepower(HP), and 1HP, 2HP, 3HP conditions (except 0HP) named A, B, C, shown as Table 2.

Each training dataset made into 3 categories according to the type of bearing fault, label '0' denotes None (Normal bearing), label '1' denotes 3 types of ball faults (EDM manual fault size in 0.007inch, 0.014inch, 0.021inch) and label '2' denotes 3 types of inner ring faults (EDM manual fault size in 0.007inch, 0.014inch, 0.021inch), label '3' denotes 3 types outer ring fault (EDM manual fault size in 0.007inch, 0.014inch, 0.021inch), as shown in Table 3,

Before this experiment, the training dataset of CWRU bearing contained 18000 samples, 2000 samples and 3000 samples for validation and testing respectively. For AlexNet, ResNet, ICN, IRv2-lite, and DeIN, the input sample is 64×64 TFG, but the ACDIN and WDCNN models, which use the raw time-domain signals as input, and every sample consists 4096 points.

The accuracy of these deep model depicted in Figure 11, the A $\rightarrow$A, B$\rightarrow$B, C$\rightarrow$C experiments based on CWRU data have approximately reached 100% [9], this is because the training data and testing data has the same distribution, but this situation is different under real working conditions. So, in order to verify the generalization ability of DeIN, we apply

TABLE 3. Detail of rolling bearing datasets.

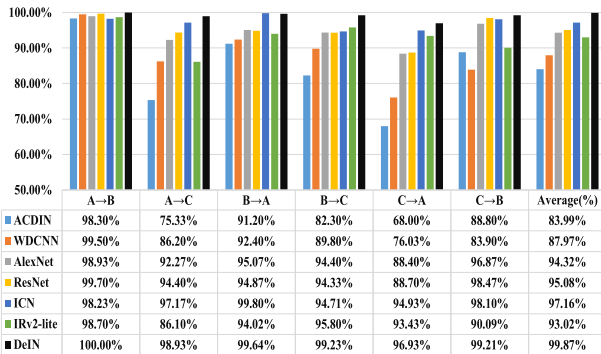|	Fault Location	None	Ball			Inner Race			Out Race		
Name	Fault Diameter	0	0.007	0.014	0.021	0.007	0.014	0.021	0.007	0.014	0.021
	Fault Labels	**0**	**1**	**1**	**1**	**2**	**2**	**2**	**3**	**3**	**3**
	Training	1800	1800	1800	1800	1800	1800	1800	1800	1800	1800
A	Validation	200	200	200	200	200	200	200	200	200	200
	Testing	300	300	300	300	300	300	300	300	300	300
	Training	1800	1800	1800	1800	1800	1800	1800	1800	1800	1800
B	Validation	200	200	200	200	200	200	200	200	200	200
	Testing	300	300	300	300	300	300	300	300	300	300
	Training	1800	1800	1800	1800	1800	1800	1800	1800	1800	1800
C	Validation	200	200	200	200	200	200	200	200	200	200
	Testing	300	300	300	300	300	300	300	300	300	300

	A→B	A→C	B→A	B→C	C→A	C→B	Average(%)
ACDIN	98.30%	75.33%	91.20%	82.30%	68.00%	88.80%	83.99%
WDCNN	99.50%	86.20%	92.40%	89.80%	76.03%	83.90%	87.97%
AlexNet	98.93%	92.27%	95.07%	94.40%	88.40%	96.87%	94.32%
ResNet	99.70%	94.40%	94.87%	94.33%	88.70%	98.47%	95.08%
ICN	98.23%	97.17%	99.80%	94.71%	94.93%	98.10%	97.16%
IRv2-lite	98.70%	86.10%	94.02%	95.80%	93.43%	90.09%	93.02%
DeIN	100.00%	98.93%	99.64%	99.23%	96.93%	99.21%	99.87%

FIGURE 11. Diagnosis result of the proposed DeIN tested at different working condition compared with other methods.

FIGURE 12. The histogram of state-of-art methods' accuracy on CWRU dataset diagnosis result.

FIGURE 13. C→A features visualization via t-SNE (a), (b) and confusion matrix(c), (d).

the current popular testing method 'cross-over testing' in the experiment. A→B denotes the model is trained under A working condition and tested under B working condition, etc.. Similarly, we conducted the experiments of A→C, B→A, B→C, C→A, C→B, to simulate the actual production status and to verify the generalization ability of DeIN.

At the same time, we compared our DeIN model with other popular deep learning models. The comparison of ACDIN, WDCNN, AlexNet, ResNet, ICN, IRv2-lite and DeIN under cross-over testing based on the CWRU bearing (contained 18000 samples, 2000 samples and 3000 samples for validation and testing respectively), shown in Figure 11. It is clear that ACDIN only has an average accuracy of 83.99 %, so it performed the worst under the cross-over testing. The WDCNN, which has an average accuracy of 87%, ranking second to last. The AlexNet, ResNet, ICN, and IRv2-lite models using TFGs as input are generally better than ACDIN, WDCNN. We review the Table 2, A load is 1 HP and rotational speed is 1772 rpm, B load is 2HP and rotational speed is 1750 rpm, C load is 3HP and rotational speed is 1730 rpm, the data distribution is different due to the rotational speed changes, the rotational speed is the most obvious between A and C, so the accuracy of all the deep learning models reduced under the two cross-over testing in A→C and C→A. In A→C, ICN has an accuracy of 97.17%, IRv2-lite has 86.1%, DeIN has 98.93% more than ICN's and IRv2-lite's, In C→A, ICN has an accuracy of 94.93%, IRv2-lite has 93.43%, DeIN has 96.93% more than ICN's and IRv2-lite's too. In total, the proposed DeIN has an average accuracy of 99.87%, which is much better than other models, it clearly shown the accuracy distribution in Figure 12.
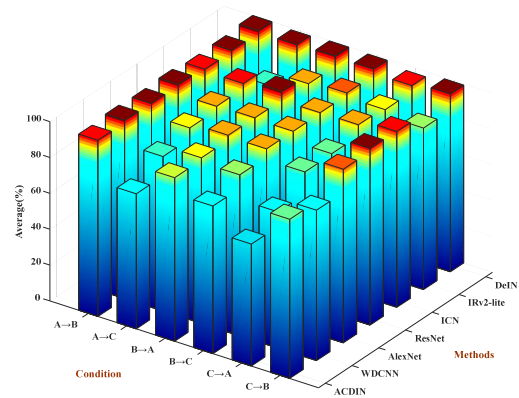
Since the low-to-high speed(C: 1730[rpm]→A: 1772[rpm]) condition gets low accuracy during the whole training process in deep learning models, in order to further understand the training process, features of the last hidden layer are visualized via t-SNE, shown in Figure 13(a)(b), and confusion matrix is further compared with the case of C→A, as shown in Figure 13(c)(d). Both of label '0'(Normal bearing) clustered well, for IRv2-lite, the result of the label '2' is not as good, it can be found that the label '2' is misclassified to the label '3' in Figure 13(a). But in the proposed model DeIN, it obviously shown that label '2' classified well in Figure 13(b) and confusion matrix results is better in Figure 13(d) than IRv2-lite that without deformable convolution. this experimental reflect that deformable convolution can improve the accuracy of classification in DeIN by extracting useful features during the network training process.

In order to further prove the performance of DeIN under limited data, using 5%, 10%, 15%, 25%, 50%, 70% and 100% of 18000 samples respectively, the dataset is shown in Table 4.

TABLE 4. Detail of rolling bearing datasets on different samples.

Load		5%	10%	15%	25%	50%	70%	100%
	Training	900	1800	2700	4500	9000	12600	18000
A	Validation	100	200	300	500	1000	1400	2000
	Testing	150	300	450	750	1500	2100	3000
	Training	900	1800	2700	4500	9000	12600	18000
B	Validation	100	200	300	500	1000	1400	2000
	Testing	150	300	450	750	1500	2100	3000
	Training	900	1800	2700	4500	9000	12600	18000
C	Validation	100	200	300	500	1000	1400	2000
	Testing	150	300	450	750	1500	2100	3000

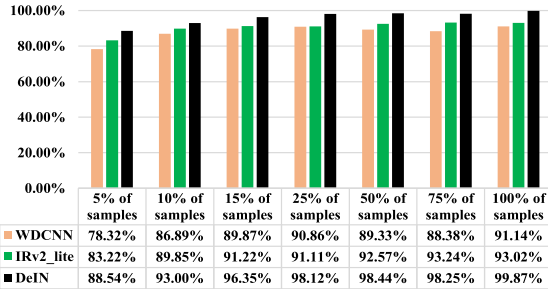|	5% of samples	10% of samples	15% of samples	25% of samples	50% of samples	75% of samples	100% of samples
WDCNN	78.32%	86.89%	89.87%	90.86%	89.33%	88.38%	91.14%
IRv2_lite	83.22%	89.85%	91.22%	91.11%	92.57%	93.24%	93.02%
DeIN	88.54%	93.00%	96.35%	98.12%	98.44%	98.25%	99.87%

FIGURE 14. State-of-art methods in limited samples on CWRU.

WDCNN, IRv2-lite and DeIN were used in this experiment, the results are shown in Figure 14. It is clear that IRv2-lite and DeIN performance are better than WDCNN. In addition, DeIN has deformable convolution which is better than IRv2-lite. In 5% of all samples, the accuracy DeIN is 88.54%, which is higher 5-10% than WDCNN's and IRv2-lite's. When the training sample is 100%, DeIN reached 99.87% more than 91.14% of WDCNN and 93.02% of IRv2-lite.

At the same time, in box diagram as shown in Figure 15, when trained in 5%, 10%, 15%, 25%, 50%, 70% and 100% samples, the model results are most divergent in smallest training samples of 5%, because shuffle training samples and local datasets are stochastic, so the results of A→C and C→A are not stable. When training samples gradually increased to 100%, the results of A→C and C→A are increasing and gradually stable. It is proved that the DeIN can effectively fit the datasets and the limited data does have specific research value for the deep learning model.

C. PADERBORN BEARING DATASET

To generate experimental data for the development of damaged bearings by using vibration signals, the test rig is a modular system that ensures flexible use of different defects in different working conditions [38], the test rig consists of several modules: (1) an electric motor, (2) a torque-measurement shaft, (3) a rolling bearing test module, (4) a flywheel and (5) a load motor, shown in Figure 16.

The ball bearings with different types of damage are mounted in the bearing test module to generate the experimental data, The other is the bearing run-to-failure test rig in Figure 17, which applied radial force is higher than in usual bearing applications to accelerate the appearance of fatigue
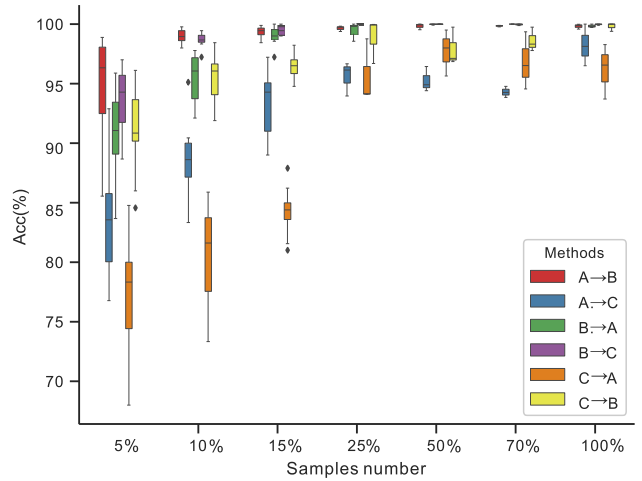

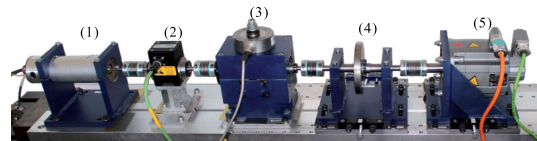**FIGURE 15.** Diagnosis result of the proposed DeIN in different sample via box diagram.

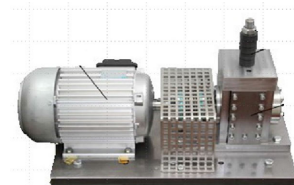**FIGURE 16.** Artificial damage bearings test rig.

FIGURE 17. Apparatus for run-to-failure accelerated life time test.

TABLE 5. Description of rolling bearing on Paderborn working conditions.

Name	Rotational speed	Load Torque	Radial force	Name of Setting
D	1500[rpm]	0.7[Nm]	1000[N]	N15_M07_F10
E	1500[rpm]	0.1[Nm]	1000[N]	N15_M01_F10
F	1500[rpm]	0.7[Nm]	400[N]	N15_M07_F04

damages, both of the artificial damage bearings test rig and run-to-failure test rig include a high sampling rate (64 kHz).

In the two test rigs, the 32 different types of bearing tests performed in 3 different working conditions: 12 bearings with artificial damage, 14 bearings with natural damage from accelerated lifetime tests and 6 healthy data. The main bearings fault location shown in Figure 18.

The basic setup (set No. 0) of the operation parameters, the test rig runs at n = 1,500 rpm with a load torque of M = 0.7 Nm and a radial force on the bearing of F = 1,000 N. Three additional settings are used by reducing the parameters one by one to n = 900 rpm, M = 0.1 Nm and F = 400N (set No.1-3), respectively. These three conditions named D, E, F, shown as Table 5.
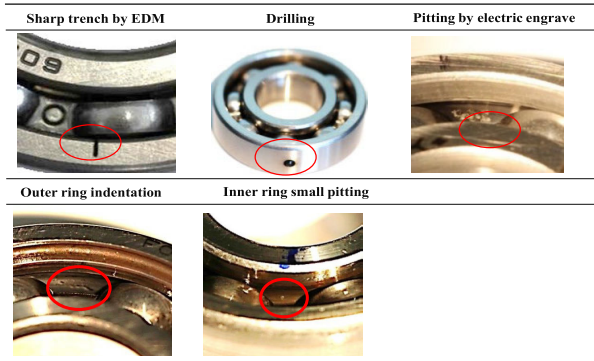
FIGURE 18. Artificial damage of bearing in top three pictures, and real damage of bearing in below two pictures.

TABLE 6. List of data sets used for training and testing the DeIN.

Fault Location Fault Labels	None 0	Out Race 1	Inner Race 2
File No.	K001 K002	Artificial damages: (KA01, KA05, KA07) Real damages: (KA04, KA15, KA16, KA22, KA30)	Artificial damages: (KI01, KI03,KI05, KI07) Real damages: (KI14, KI16, KI17, KI18, KI21)

TABLE 7. Detail of rolling bearing datasets on Paderborn.

Condition	Fault Location	None	Inner Race			Out Race		
D	Training	1800	1800	1800	1800	1800	1800	1800
	Validation	200	200	200	200	200	200	200
	Testing	300	300	300	300	300	300	300
E	Training	1800	1800	1800	1800	1800	1800	1800
	Validation	200	200	200	200	200	200	200
	Testing	300	300	300	300	300	300	300
F	Training	1800	1800	1800	1800	1800	1800	1800
	Validation	200	200	200	200	200	200	200
	Testing	300	300	300	300	300	300	300

Each bearing in 20 measurements under a load setting measured a vibration signal of approximately 4s at a sampling rate of 64 kHz and a total of approximately 256,000 points. For this experiment, this datasets contained signal data obtained from healthy bearings, artificially damaged bearings, and real damaged bearings, all these bearings were running under three different loads and at a speed of 1500 rpm, the rolling bearing datasets filename selected is shown in Table 6. The detail of rolling bearing datasets selected in Table 7.

To better identify the DeIN's ability, the names of the working conditions are D, E, and F. D→E denotes the model is trained under D working condition and tested under E working condition. Similarly, we conducted the experiments of D→F, E→D, E→F, F→D, F→E, in the Paderborn experiment, the 7 deep learning models of ACDIN, WDCNN, AlexNet, ResNet, ICN, IRv2-lite, and DeIN were carried out, as shown in Figure 19, the average accuracy of ResNet was 77.52%, which was the worst performance. The average accuracy of the DeIN is far more than the other 6 deep learning models. Due to the rotational speed and the load torque are the same in D and F, among of the 7 deep learning models can achieve better results in D→F and F→D,
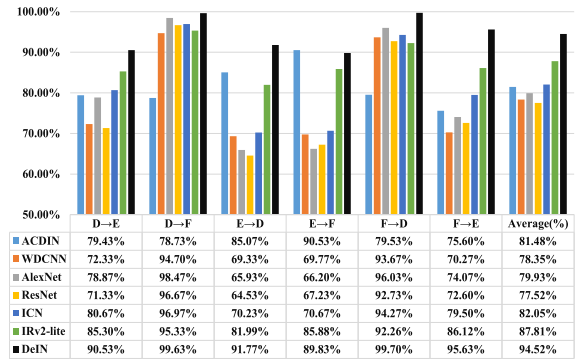

FIGURE 19. State-of-art methods on Paderborn.

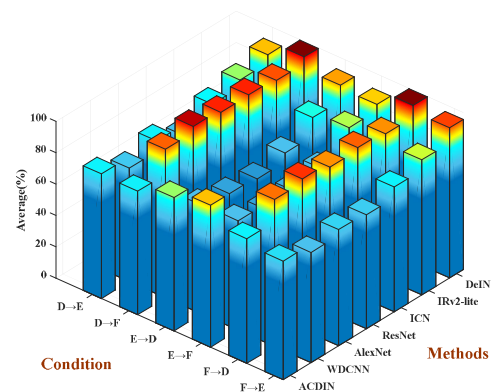|	D→E	D→F	E→D	E→F	F→D	F→E	Average(%)
ACDIN	79.43%	78.73%	85.07%	90.53%	79.53%	75.60%	81.48%
WDCNN	72.33%	94.70%	69.33%	69.77%	93.67%	70.27%	78.35%
AlexNet	78.87%	98.47%	65.93%	66.20%	96.03%	74.07%	79.93%
ResNet	71.33%	96.67%	64.53%	67.23%	92.73%	72.60%	77.52%
ICN	80.67%	96.97%	70.23%	70.67%	94.27%	79.50%	82.05%
IRv2-lite	85.30%	95.33%	81.99%	85.88%	92.26%	86.12%	87.81%
DeIN	90.53%	99.63%	91.77%	89.83%	99.70%	95.63%	94.52%

FIGURE 20. The histogram of state-of-art methods' accuracy on Paderborn dataset diagnosis result.

the accuracy of DeIN in D→F and F→D even reached 99.63% and 99.70% respectively, far exceeding the ACDIN has accuracy of 78.73% and 79.53%. At the same time, the performance of IRv2-lite is also very good overall, except in D→E and E→D performance is bad. In the CWRU experiment, ICN has accuracy of 97.16%, but in the Paderborn experiment, it only has 82.05% which is lower than the average accuracy of DeIN' s 94.52%. which is much better than other models, it clearly shown the accuracy distribution in Figure 20.

To better understand the effect of IRv2-lite and DeIN in Paderborn bearing datasets, Figure 21(a)(b) show feature visualization via t-SNE: the last hidden layer visualization. Figure 21(c)(d) show confusion matrix results. From Figure 21(a)(b), both of label '1'(out race damages) clustered well, for IRv2-lite, the resolution of the label '0'(normal bearings) is not as good, it can be found that the label '0' is misclassified to the label '2'(inner race damages) in Figure 21(a). But in the proposed model DeIN, it obviously shown that label '2' classified well in Figure 21(b) and confusion matrix results is better in Figure 21(d) than IRv2-lite that without deformable convolution. this experimental proved again that deformable convolution can improve the accuracy of classification in DeIN by extracting useful features during the network training process.
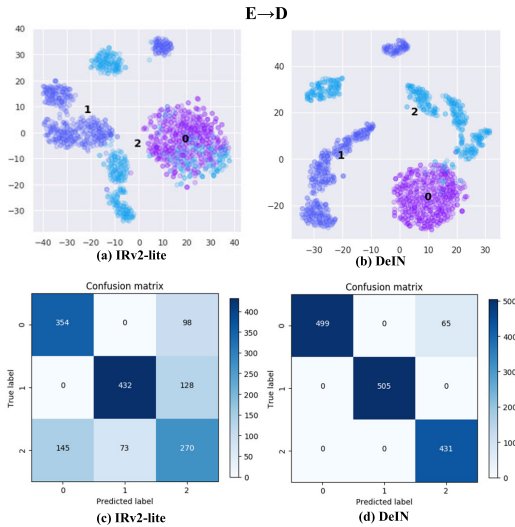
FIGURE 21. E→D features visualization via t-SNE (a), (b) and confusion matrix (c), (d).

IV. CONCLUSION

In this paper, we have proposed a powerful approach for rolling bearing fault diagnosis named DeIN, which can improve the feature extraction ability of the model in the bearing fault data by applying deformable convolution and InceptionResnetv2. Experiments with various deep learning models show that the bearing data processed to TFGs by STFT could achieve better performance in fault diagnosis. DeIN based on TFGs has excellent ability to eliminate background noise, and the output of the Auxiliary-output classifier and the Main-output classifier guarantees the training speed of the model while improving high-precision diagnostics. Comparing with the popular time-domain signal models WDCNN, ACDIN and the latest TFGs fault diagnosis models (such as ICN), DeIN reaching 99.87% and 94.52% higher accuracy than these models based on CWRU and Paderborn datasets respectively. Furthermore, to facilitate open source in the field of bearing fault diagnosis, all our models in this work and datasets are open sourced and can be downloaded at the URL (https://mekhub.cn/philbaz/Bearing_Fault_Diagnosis_in_TFGs_Using_DeIN). In the future, DeIN will be used to solve fault diagnosis problems in more fields.

REFERENCES

[1] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.

[2] S. Ren and K. He, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[3] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.

[4] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[5] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, Jun. 2014.

[6] A. Zhang, H. Wang, S. Li, Y. Cui, Z. Liu, G. Yang, and J. Hu, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Appl. Sci.*, vol. 8, no. 12, p. 2416, 2018.

[7] Z. Su, Y. Li, and G. Yang, "Dietary composition perception algorithm using social robot audition for mandarin Chinese," *IEEE Access*, vol. 8, pp. 8768–8782, 2020.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.

[10] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mech. Syst. Signal Process.*, vol. 100, pp. 439–453, Feb. 2018.

[11] Y. Liu, X. Zhu, X. Zhao, and Y. Cao, "Adversarial learning for constrained image splicing detection and localization based on atrous convolution," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2551–2566, Oct. 2019.

[12] X. Huang and X. Lei, "Algorithm of image classification based on Atrous-CNN," in *Proc. Chin. Control Decis. Conf. (CCDC)*, 2019, pp. 5324–5328.

[13] Z. Ren, Q. Kong, J. Han, M. D. Plumbley, and B. W. Schuller, "Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 56–60.

[14] L.-C. Chen and Y. Zhu, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[15] Y. Chen, G. Peng, C. Xie, W. Zhang, C. Li, and S. Liu, "ACDIN: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis," *Neurocomputing*, vol. 294, pp. 61–71, Jun. 2018.

[16] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong, and J. Hu, "Limited data rolling bearing fault diagnosis with few-shot learning," *IEEE Access*, vol. 7, pp. 110895–110904, 2019.

[17] J. Wang, Z. Mo, H. Zhang, and Q. Miao, "A deep learning method for bearing fault diagnosis based on time-frequency image," *IEEE Access*, vol. 7, pp. 42373–42383, 2019.

[18] J. Shi, W. Huang, C. Shen, X. Jiang, and Z. Zhu, "Dual-Guidance-Based optimal resonant frequency band selection and multiple ridge path identification for bearing fault diagnosis under time-varying speeds," *IEEE Access*, vol. 7, pp. 144995–145012, 2019.

[19] J. Huang, G. Chen, L. Shu, S. Wang, and Y. Zhang, "An experimental study of clogging fault diagnosis in heat exchangers based on vibration signals," *IEEE Access*, vol. 4, pp. 1800–1809, 2016.

[20] Z. Zhu, G. Peng, Y. Chen, and H. Gao, "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis," *Neurocomputing*, vol. 323, pp. 62–75, Jan. 2019.

[21] B. Zhang and W. Li, "Adversarial adaptive 1-D convolutional neural networks for bearing fault diagnosis under varying working condition," pp. 1–19, 2018.

[22] Y. Zhao, Y. Shen, Y. Zhu, and J. Yao, "Forecasting wavelet transformed time series with attentive neural networks," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1452–1457.

[23] B. Yuan and C. Wang, "WaveletFCNN: A deep time series classification model for wind turbine blade icing detection," Tech. Rep., 2019.

[24] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019.

[25] S. Cahyawijaya, B. Wilie, and W. Adiprawita, "IDEnet: Inception-based deep convolutional neural network for crowd counting estimation," in *Proc. 5th Int. Conf. Electr. Eng., Comput. Sci. Informat. (EECSI)*, Oct. 2018, pp. 548–553.

[26] H. Wu, Z. Xu, J. Zhang, and G. Jia, "Offset-adjustable deformable convolution and region proposal network for visual tracking," *IEEE Access*, vol. 7, pp. 85158–85168, 2019.

[27] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[28] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018.

[29] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang, "Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1874–1887, Aug. 2018.

[30] M. Sun, G. Zhang, H. Dang, X. Qi, X. Zhou, and Q. Chang, "Accurate gastric cancer segmentation in digital pathology images using deformable convolution and multi-scale embedding networks," *IEEE Access*, vol. 7, pp. 75530–75541, 2019.

[31] C. Szegedy and S. Ioffe, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[33] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017.

[34] M. Längkvist, L. Karlsson, and A. Loutfi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014.

[35] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[36] X. Li, D. Chang, T. Tian, and J. Cao, "Large-margin regularized softmax cross-entropy loss," *IEEE Access*, vol. 7, pp. 19572–19578, 2019.

[37] G. Liang, H. Hong, W. Xie, and L. Zheng, "Combining convolutional neural network with recursive neural network for blood cell image classification," *IEEE Access*, vol. 6, pp. 36188–36197, 2018.

[38] C. Lessmeier and J. K. Kimotho, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification," in *Proc. 3rd Eur. Conf. Progn. Heal. Manag. Soc.*, 2016, pp. 152–156.

SHAOBO LI received the Ph.D. degree in computer software and theory from the Chinese Academy of Sciences, China, in 2003.

From 2007 to 2015, he was the Vice Director of the Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University (GZU), China. He is currently a Professor with the School of Mechanical Engineering, GZU. He is also a part-time Doctoral Tutor with the Chinese Academy of Sciences. He has been the Dean of the School of Mechanical Engineering, GZU, since 2015. He has published more than 200 articles in major journals and international conferences. His current research interests include the big data of manufacturing and intelligent manufacturing. His research has been supported by the National Science Foundation of China and the National High-Tech Research and Development Program (863 Program). He received honors and awards from New Century Excellent Talents in the University of Ministry of Education of China, Excellent Expert, and Innovative Talent of Guizhou Province, the Group Leader of Manufacturing Informatization, and the Alliance Vice Chairman of the intelligent manufacturing industry in Guizhou.

WANLI YANG received the B.S. degree in mechanical engineering from Guizhou University, Guiyang, China, in 2016, where he is currently pursuing the M.S degree with the School of Mechanical Engineering.

His current research interests include machinery condition monitoring, intelligent fault diagnostics, and prognostics of rotating machinery.

ANSI ZHANG received the B.S. degree in mechanical engineering from Shanghai University, Shanghai, China, in 2014. He is currently pursuing the Ph.D. degree with the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, China. From 2017 to 2018, he was a Visiting Ph.D. Student with the University of South Carolina, Columbia, USA.

His current research interests include machine learning, machinery condition monitoring, and intelligent fault diagnostics.
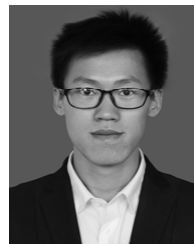
HUIBIN LIU received the B.S. degree in mechanical and electrical engineering from the China University of Mining and Technology, Xuzhou, China, in 2017. He is currently pursuing the M.S. degree with the School of Mechanical Engineering, Guizhou University, China.

His current interests include machine learning, machinery condition monitoring, and intelligent fault diagnosis.

JINYUAN HUANG received the B.S. degree from the Hunan Institute of Science and Technology of China, in 2017. She is currently pursuing the M.S. degree with the School of Mechanical Engineering, Guizhou University, China. Her research interests include machine learning, deep learning, remaining useful life estimation, and so on.

CHUANJIANG LI received the B.S. degree in engineering mechanics from the North University of China, in 2018. He is currently pursuing the M.S. degree with the School of Mechanical Engineering, Guizhou University, China.

His research interests include machine learning, deep learning, intelligent fault diagnostics, and so on.

JIANJUN HU received the B.S. and M.S. degrees in mechanical engineering from the Wuhan University of Technology, China, in 1995 and 1998, respectively, and the Ph.D. degree in computer science from Michigan State University, in 2004, with a focus on machine learning and evolutionary computation.

He was a Postdoctoral Fellow in bioinformatics with Purdue University and the University of Southern California, from 2004 to 2007. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA. He is also a Visiting Professor with the School of Mechanical Engineering, Guizhou University, Guiyang, China. His research interests include machine learning, deep learning, data mining, evolutionary computation, fault diagnosis, bioinformatics, and material informatics.

• • •