This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

Preprocessing-Free Gear Fault Diagnosis Using Small Datasets with Deep Convolutional Neural Network-Based Transfer Learning

Pei Cao, Shengli Zhang, and J. Tang, *Member, IEEE*

***Abstract*—Early diagnosis of gear transmission has been a significant challenge, because gear faults occur primarily at microstructure or even material level but their effects can only be observed indirectly at a system level. The performance of a gear fault diagnosis system depends significantly on the features extracted and the classifier subsequently applied. Traditionally, fault-related features are extracted and identified based on domain expertise through data preprocessing which are system-specific and may not be easily generalized. On the other hand, although recently the deep neural networks based approaches featuring adaptive feature extractions and inherent classifications have attracted attention, they usually require a substantial set of training data. Aiming at tackling these issues, this paper presents a deep convolutional neural network-based transfer learning approach. The proposed transfer learning architecture consists of two parts; the first part is constructed with a pre-trained deep neural network that serves to extract the features automatically from the input, and the second part is a fully connected stage to classify the features that needs to be trained using gear fault experimental data. Case analyses using experimental data from a benchmark gear system indicate that the proposed approach not only entertains preprocessing free adaptive feature extractions, but also requires only a small set of training data.**

***Index Terms*—alexnet, deep convolutional neural network, gear fault diagnosis, transfer learning**

I. Introduction

Condition monitoring and fault diagnosis play essential role in ensuring the safe and sustainable operations of modern machinery systems. Gearbox, as one common component used in those systems, is prone to fault condition or even failure, because of the severe working condition with high mechanical loading and typically long operational time. Currently, vibration signals are most widely used to infer the health condition of gear system, because they contain rich information and can be easily measured using off-the-shelf, low-cost sensors. Indeed, gear vibration signals contain three components: periodic meshing frequencies, their harmonics, and random noise. For a healthy gear system, the meshing frequencies and their harmonics dominate the vibration response. Fault conditions cause additional dynamic effects.

The practice of fault diagnosis of gear system using vibration signals has proved to be a very challenging subject. The mainstream of gear condition monitoring is built upon various feature extraction methods that are manual and empirical in nature [1-3]. Generally, a certain signal processing technique is applied to vibration signals to identify fault-related features that are selected based on engineering judgment. Subsequently, a classifier is developed and applied to new signals to predict fault occurrence in terms of type and severity. There have been extensive and diverse attempts in manually and empirically identifying and extracting useful features from gear vibration signals, which fall into three main categories: time-domain analysis [4][5], frequency domain-analysis [6-8] and time-frequency analysis [9-13]. Time-domain statistical approaches can capture the changes in amplitude and phase modulation caused by faults [5][14]. In comparison, spectrum analysis may extract the features more easily to detect distributed faults with clear sidebands [6][8][15]. To deal with noise and at the same time utilize the transient components in vibration signals, many efforts have focused on joint time-frequency domain analysis utilizing Wigner-Ville distribution [9][16], short time Fourier transform [10][17], and various wavelet transforms [11][18]. The time-frequency distribution in such analysis can in theory lead to rich analysis results regarding the time- and frequency-related events in signals.

Although the manual and empirical methods of feature extraction have seen various levels of successes, obviously their effectiveness is hinged upon the specific features adopted in the diagnostic analysis. It is worth emphasizing that the choices of features as well as the often-applied signal preprocessing techniques are generally based on domain expertise and subjective decisions on a specific gear system. For example, while wavelet transforms have been popular and it is well known that each wavelet coefficient can be interpreted as the energy concentration at a specific time-frequency point, it is evident from large amount of literature

This research is supported by the National Science Foundation under Grant IIS-1741171.

Pei Cao is with Department of Mechanical Engineering, University of Connecticut, 191 Auditorium Road, Unit 3139, Storrs, CT 06269, USA (email: pei.cao@uconn.edu).

Shengli Zhang is with Stanley Black & Decker, Global Tool & Storage Headquarter, Towson, MD 21286, USA (e-mail: beyonglee@gmail.com).

Jiong Tang is with Department of Mechanical Engineering, University of Connecticut, 191 Auditorium Road, Unit 3139, Storrs, CT 06269, USA (email: jiong.tang@uconn.edu).

.

that there does not seem to be a consensus on what kind of wavelet to use for gear fault diagnosis. This should not come as a surprise. On one hand gear faults occur primarily at microstructure or even material level but their effects can only be observed indirectly at a system level; consequently there exists a many-to-many relationship between actual faults and the observable quantifies (i.e., features) for a given gear system [19]. On the other hand, different gear systems have different designs which lead to very different dynamic characteristics. As such, the result on features manually selected and, to a large extent, the methodology employed to extract these features for one gear system design may not be easily extrapolated to a different gear system design.

Fundamentally, condition monitoring and fault diagnosis of gear systems belongs to the general field of pattern recognition. The advancements in related algorithms along with the rapid enhancement of computational power have trigged the wide spread of machine learning techniques to various applications. Most recently, deep neural network-based methods are progressively being investigated. When the parameters of a deep neural network are properly trained by available data, representative features can be extracted in a hierarchy of conceptual abstractions, which are free of human interference compared to manual selection of features. Some recent studies have adopted such type of approaches in gear fault diagnosis, aiming at identifying features implicitly and adaptively and then classifying damage/fault in an automated manner with minimal tuning. For example, Zhang et al [20] developed a deep learning network for degradation pattern classification and demonstrated the efficacy using turbofan engine dataset. Li et al [21] proposed a deep random forest fusion technique for gearbox fault diagnosis which achieves 97.68% classification accuracy. Weimer et al [22] examined the usage of deep convolutional neural network for industrial inspection and demonstrated excellent defect detection results. Ince et al [23] developed a fast motor condition monitoring system using a 1-D convolutional neural network with a classification accuracy of 97.4%. Abdeljaber et al [24] performed real-time damage detection using convolutional neural network and showcased satisfactory efficiency.

Deep neural network is undoubtedly a powerful tool in pattern recognition and data mining. As an end-to-end hierarchical system, it inherently blends the two essential elements in condition monitoring, feature extraction and classification, into a single adaptive learning frame. It should be noted that the amount of training data required for satisfactory results depends on many aspects of the specific problem being tackled, such as the correctness of training samples, the number of pattern classes to be classified, and the degree of separation between different classes. In most machinery diagnosis investigations, the lack of labeled training samples, i.e., experiment data of known failure patterns, is a common issue, because it is impractical to collect experimental data of each failure type and especially severity for a machinery system. To improve the performance given limited training data, some recent studies have attempted to combine preprocessing and data augmentation techniques, e.g., discrete wavelet transform [25], antialiasing/decimation filter [23], and wavelet packet transform [21], with neural networks for fault diagnosis. Nevertheless, the preprocessing techniques employed, which are subjected to selection based on domain expertise, may negatively impact the objective nature of neural networks and to some extent undermines the usage of such tools.

In this research, aiming at advancing the state-of-the-art, we present a deep neural network-based transfer learning approach utilizing limited time-domain data for gearbox fault diagnosis. One-dimensional time-domain data of vibration responses related to gear fault patterns are converted into graphical images as input. The approach inherits the non-biased nature of neural networks that can avoid the manual selection of features. Meanwhile, the issue of limited data is overcome by formulating a new neural network architecture that consists of two parts. Massive image data (1.2 million) from ImageNet (http://www.image-net.org/challenges/LSVRC/2010/) are used first to train an original deep neural network model, denoted as neural network A. The parameters of neural network A are transferred (copied) to the new architecture as the first part. The second part of the architecture, an untrained neural network B, accommodates the gear fault diagnosis task and is further trained using experimentally generated gear fault data. Unlike traditional neural networks, the training set of transfer learning do not necessarily subordinate to the same category or from the same physical background [26]. As to be demonstrated later, with this new architecture, highly accurate gear fault diagnosis can be achieved using limited time-domain data directly without involving any subjective preprocessing techniques to assist feature extraction. The rest of this paper is organized as follows. In Section II, building upon convolutional neural network and transfer learning, we develop the specific architecture for gear fault diagnosis. In Section III, experimental data are analyzed using the proposed approach with uncertainties and noise; comparisons with respect to different approaches are conducted as well. Concluding remarks are summarized in Section IV.

II. Transfer Learning for Gear Fault Diagnosis

The proposed transfer learning approach is built upon deep convolutional neural network. Deep neural networks have enjoyed great success but require a substantial amount of training instances for satisfactory performance. In this section, for the sake of completeness in presentation we start from the essential formulations of convolutional neural network and transfer learning, followed by the specific architecture developed for gear fault diagnosis with limited training data.

A. Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of biologically inspired neural networks featuring one or multiple convolutional layers that simulate human visual system [27]. In recent years, due to the enhancement in computational

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

power and the dramatic increase in the amount of data available in various applications, CNNs-based methods have shown significant improvements in performance and thus have become the most popular class of approaches for pattern recognition tasks such as image classification [28], natural language processing [29], recommending systems [30] and fault detection [23]. CNNs learn how to extract and recognize characteristics of the target task by combining and stacking convolutional layers, pooling layers and fully connected layers in its architecture. Figure 1 illustrates a simple CNN with an input layer to accept input images, a convolutional layer to extract features, a ReLU layer to augment features through non-linear transformation, a max pooling layer to reduce data size, and a fully connected layer combined with a softmax layer to classify the input to pre-defined labels. The parameters are trained through a training dataset and updated using back propagation algorithm to reflect the features of the task that may not be recognized otherwise. The basic mechanism of layers in CNNs is outlined as follows.

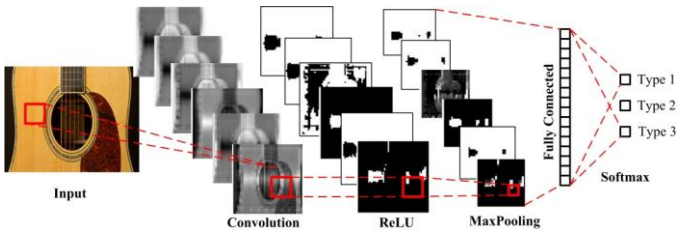


Figure 1 An example of convolutional neural network.

Convolutional layer Each feature map in the convolutional layer shown in Figure 1 is generated by a convolution filter. Generally, the input and convolution filters are tensors of size $m \times n$ and $p \times q \times K$ (K is the number of filter used), respectively. Stride (i.e., step size of the filter sliding over input) is set to 1 and padding (i.e., the number of rows and columns to insert around the original input) is set to 0. The convolution operation can be expressed as,

$$y_{d_1,d_2,k} = \sum_{i=0}^{p}\sum_{j=0}^{q} x_{d_1 i, d_2 j} \times f_{i,j,k} \tag{1}$$

where y, x and f denote the element in feature map, input and convolution filter, respectively. $f_{i,j,k}$ represents the element on the i-th column and j-th row for filter k. $y_{d_1,d_2,k}$ is the element on the d_1-th column and d_2-th row of feature map k. And $x_{d_1 i, d_2 j}$ refers to the input element on the i-th column and j-th row of the stride window specified by d_1 and d_2. Equation (1) gives a concise representation of the convolution operation when the input is 2-demensional, and stride and padding are 1 and 0. Higher dimension convolution operations can be conducted in a similar manner. To be more evocative, suppose the input image can be represented by a 4×7 matrix and the convolution kernel is a 3×3 identity matrix. As we take kernel and stride it over the image matrix, dot products are taken in each step and recorded in a feature map matrix (Figure 2). Such operation is called convolution. In CNNs, multiple convolution filters are used in a convolutional layer, each acquiring a feature piece in its own perspective from the input image specified by the filter parameters. Regardless of what and where a feature appears in the input, the convolutional layer will try to characterize it from various perspectives that have been tuned automatically by the training dataset.

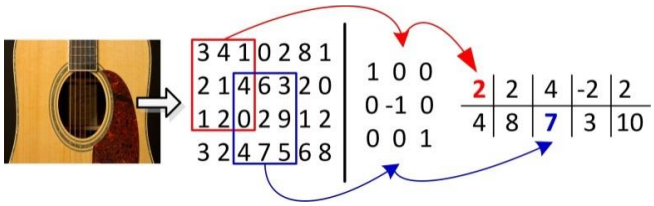


Figure 2 Illustration of convolution operation.

ReLU layer In CNNs, ReLU (rectified linear units) layers are commonly used after convolutional layers. In most cases, the relationship between the input and output is not linear. While the convolution operation is linear, the ReLU layer is designed to take non-linear relationship into account, as shown in the equation below,

$$\bar{y} = \max(0, y) \tag{2}$$

The ReLU operation is applied to each feature map and returns an activation map (Figure 3). The depth of the ReLU layer equals to that of the convolutional layer.

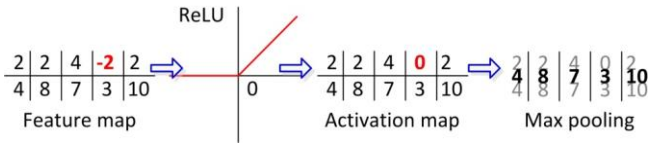


Figure 3 Illustration of ReLU and max pooling.

Max pooling layer Max pooling down-samples a sub-region of the activation map to its maximum value,

$$\hat{y} = \max_{L_1 \le i \le U_1,\, L_2 \le i \le U_2} \bar{y}_{i,j} \tag{3}$$

where $L_1 \le i \le U_1$ and $L_2 \le j \le U_2$ define the sub-region. The max pooling layer not only makes the network less sensitive to location changes of a feature but also reduces the size of parameters, thus alleviates computational burden and controls overfitting.

B. *Transfer learning*

CNNs are powerful tools, and the performance can generally be improved by up-scaling the CNN equipped. The scale of a CNN concurs with the scale of the training dataset. Naturally, the deeper the CNN, the more parameters need to be trained, which requires a substantial amount of valid training samples. Nevertheless, in gear fault diagnosis, the training data is not as sufficient as that of data-rich tasks such as natural image classification. In fact, it is impractical to collect physical data from each failure type and especially severity since the severity level is continuous in nature and there are infinitely many possible fault profiles.

Figure 4 illustrates a representative relationship between data size and performance for different learning methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

While the performance of a large-scale CNN has the potential to top other methods, it is also profoundly correlated with the size of training data. Transfer learning, on the other hand, is capable of achieving prominent performance commensurate with large scale CNNs using only a small set of training date [31][32]. By applying knowledge and skills (in the form of parameters) learned and accumulated in previous tasks that have sufficient training data, transfer learning provides a possible solution to improve the performance of a neural network when applied to a novel task with small training dataset. Classic transfer learning approaches transfer (copy) the first n layers of a well-trained network to the target network of layer $m > n$. Initially, the last $(m-n)$ layers of the target network are left untrained. They are trained subsequently using the training data from the novel task. Let the training datasets from the previous task $\mathbf{D}_{\text{pre}}$ and the novel task $\mathbf{D}_{\text{nov}}$ be represented as

$$\mathbf{D}_{\text{pre}} = \{\mathbf{X}_{\text{pre}}, \mathbf{L}_{\text{pre}}\},\ \mathbf{D}_{\text{nov}} = \{\mathbf{X}_{\text{nov}}, \mathbf{L}_{\text{nov}}\} \quad (4a, b)$$

where $\mathbf{X}$ is the input and $\mathbf{L}$ is the output label. The CNNs for both tasks can then be regarded as, $\hat{\mathbf{L}}_{\text{pre}} = \text{CNN}_{\text{pre}}(\mathbf{X}_{\text{pre}}, \boldsymbol{\theta}_{\text{pre}})$,

$$\hat{\mathbf{L}}_{\text{nov}} = \text{CNN}_{\text{nov}}(\mathbf{X}_{\text{nov}}, \boldsymbol{\theta}_{\text{nov}}) \quad (5a, b)$$

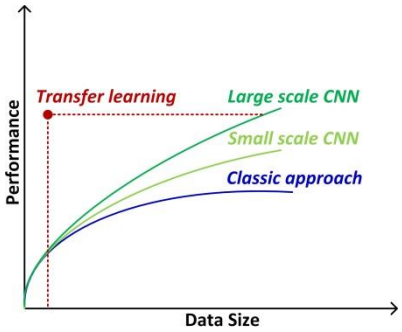


Figure 4 Learning methods: data size vs. performance.

CNN operator denotes the mapping of a convolutional neural network given parameters $\boldsymbol{\theta}$ from input to predicted output $\hat{\mathbf{L}}$. The parameters of the previous task is trained through

$$\boldsymbol{\theta}_{\text{pre}}' = \arg\min_{\boldsymbol{\theta}_{\text{pre}}}(\mathbf{L}_{\text{pre}} - \hat{\mathbf{L}}_{\text{pre}}) = \arg\min_{\boldsymbol{\theta}_{\text{pre}}}(\mathbf{L}_{\text{pre}} - \text{CNN}_{\text{pre}}(\mathbf{X}_{\text{pre}}, \boldsymbol{\theta}_{\text{pre}})) \quad (6)$$

where $\boldsymbol{\theta}_{\text{pre}}'$ stands for the parameters after training. Thereupon, the trained parameters of the first n layers can be transferred to the new task as,

$$\boldsymbol{\theta}_{\text{nov}}(1:n)' := \boldsymbol{\theta}_{\text{pre}}(1:n)' \quad (7)$$

The rest of the parameter can be trained using training samples from the novel task,

$$\boldsymbol{\theta}_{\text{nov}}(1:m)' = [\boldsymbol{\theta}_{\text{nov}}(1:n)'', \boldsymbol{\theta}_{\text{nov}}(n:m)'] = \arg\min_{\boldsymbol{\theta}_{\text{nov}}(1:m)}(\mathbf{L}_{\text{nov}} - \text{CNN}_{\text{nov}}(\mathbf{X}_{\text{nov}}, [\boldsymbol{\theta}_{\text{nov}}(1:n)', \boldsymbol{\theta}_{\text{nov}}(n:m)])) \quad (8)$$

In Equation (8), by setting differential learning rates, the parameters in the first n layers are fine-tuned as $\boldsymbol{\theta}_{\text{nov}}(1:n)''$ using a smaller learning rate, and the parameters in the last $(m-n)$ layers are trained from scratch as $\boldsymbol{\theta}_{\text{nov}}(n:m)'$. The phrase "differential learning rates" refers to different learning rates for different parts of the network during our training. In general, the transferred layers (i.e., the first n layers) are pre-trained to detect and extract generic features of inputs which are less sensitivity to the domain of application. Therefore, the learning rate for the transferred layers is usually very small. In an extreme case where the learning rate for the transferred layers is zero, the parameters in the first n layers transferred are left frozen.

Therefore, the CNN used for the novel task for future fault classification and diagnosis can be represented as,

$$\text{CNN}_{\text{nov}}(\mathbf{X}_{\text{nov}}, [\boldsymbol{\theta}_{\text{nov}}(1:n)'', \boldsymbol{\theta}_{nov}(n:m)']) \quad (9)$$

where the parameters in the first n layers are first transferred from a previous task. Meanwhile, as the last $(m-n)$ layers are trained using the training dataset of the novel task, the first n layers are fine-tuned for better results.

$$\boldsymbol{\theta}_{\text{nov}}' = [\boldsymbol{\theta}_{\text{nov}}(1:n)'', \boldsymbol{\theta}_{\text{nov}}(n:m)'] \quad (10)$$

Transfer learning becomes possible and promising because, as has been discovered by recent studies, the layers at the convolutional stages (convolutional layers, ReLU layers and pooling layers) of the convolutional neural network trained on large dataset indeed extract general features of inputs, while the layers of fully connected stages (fully connected layers, softmax layers, classification layers) are more specific to task [33][34]. Therefore, the n layers transferred to the new task as a whole can be regarded as a well-trained feature extraction tool towards similar tasks and the last few layers serve as a classifier to be trained. Even with substantial training data, initializing with transferred parameters can improve the performance in general [35].

In this research, transfer learning is implemented to gearbox fault diagnosis. The CNN is well-trained in terms of pulling characteristics from images. As illustrated in Figure 5, the parameters in the convolutional stage, i.e., the parameters used in the convolution filter, the ReLU operator and the max pooling operator are transferred to the fault diagnosis task. The parameters used in the fully connected layer and the softmax layers are trained subsequently using a small amount of training data generated from gear fault experiments.

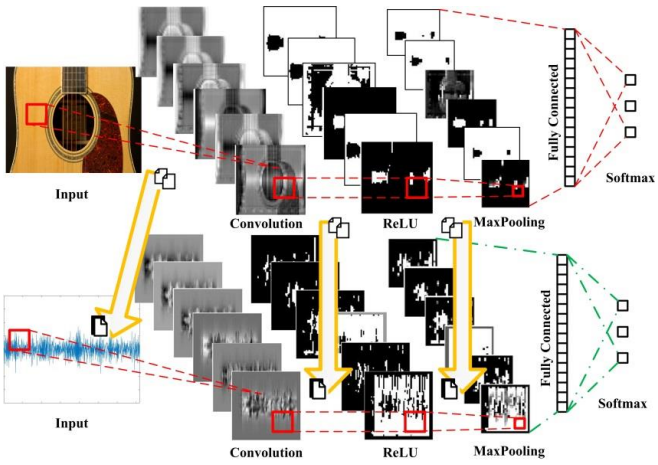


Figure 5 Illustration of transfer learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

C. Proposed architecture

In this sub-section we present the proposed architecture. In gear fault diagnosis, vibration responses are recorded using accelerometers during gearbox operation. The time-domain vibration signals can then be represented directly by 2D grey-scale/true-color images (as shown in Figure 5) which serve as inputs of the deep CNN. More details on image representation of time-domain data will be provided in Section III.A. The deep CNN adopted as the base architecture in this study was originally proposed by Krizhevsky et al [28] which is essentially composed of five convolutional stages and three fully connected stages (Figure 6). This base architecture showed its extraordinary performance in Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) and has since been repurposed for other learning tasks [31].

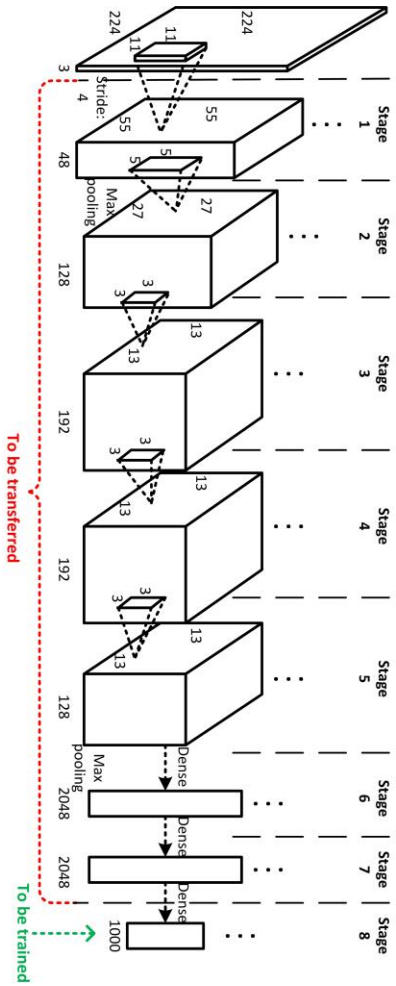


Figure 6 Illustration of the transfer learning architecture

In the base architecture, the parameters are trained using approximately 1.2 million human/software labeled 3D true-color nature images from ImageNet Large Scale Visual Recognition Challenge 2010 (http://www.image-net.org/challenges/LSVRC/2010/). The trained parameters in the first five stages are well-polished in characterizing high-level abstractions of the input image and thus have the potential to be used for other tasks with image inputs. Meanwhile, the last three stages are trained to nonlinearly combine the high-level features. Although the images of vibration signals may look different from the images used to train the original CNN, useful features can be extracted in a similar manner as long as the CNN adopted is capable of identifying high-level abstractions [35]. Stage 8 of the original architecture is configured for 1000 classes in the previous image classification task. Therefore, the first seven stages of the base architecture can be possibly transferred to facilitate gear fault diagnosis. As discussed in Section II.B, the first seven stages indeed serve as a general well-trained tool for automatic feature extraction. The more stages and layers used, the higher level of features can be obtained. The final stage is left to be trained as a classifier using the experimental data specific to the fault diagnosis task. As specified in Table 1, a total of 24 layers are used in the proposed architecture; the parameters and specifications used in the first 21 layers can be transferred from the base architecture.

Table 1 Specifications of the proposed architecture

Stage	Layer	Name	Specifications
1 (transferred)	1	Convolutional	11×11×96
	2	ReLU	N/A
	3	Normalization	5 channels/element
	4	Max pooling	3×3
2 (transferred)	5	Convolutional	5×5×256
	6	ReLU	N/A
	7	Normalization	5 channels/element
	8	Max pooling	3×3
3 (transferred)	9	Convolutional	3×3×384
	10	ReLU	N/A
4 (transferred)	11	Convolutional	3×3×384
	12	ReLU	N/A
5 (transferred)	13	Convolutional	3×3×256
	14	ReLU	N/A
	15	Max pooling	3×3
6 (transferred)	16	Fully connected	4096
	17	ReLU	N/A
	18	Dropout	50%
7 (transferred)	19	Fully connected	4096
	20	ReLU	N/A
	21	Dropout	50%
8 (to be trained)	22	Fully connected	9
	23	Softmax	N/A
	24	Classification	Cross entropy

We observe Table 1. Overfitting of the learning model is essentially controlled by the max pooling layers in Stages 1, 2, and 5, and the dropout layers in Stages 6 and 7. As explained in Section II.A, a max pooling layer not only makes the network less sensitive to location changes of a feature but also reduces the size of parameters. Therefore, max pooling can reduce computational burden and control overfitting. In our architecture, dropout layers are employed after the ReLU layers in Stages 6 and 7. Because a fully connected layer possesses a large number of parameters, it is prone to overfitting. A simple and effective way to prevent from overfitting is dropout [36]. In our study, individual nodes are "dropped out of" (temporarily removed from) the net with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

probability 50% as suggested in [36]. Dropout can be interpreted as a stochastic regularization technique which not only decreases overfitting by avoiding training all nodes, but also significantly improves training efficiency.

The loss function used is the cross-entropy function given as follows,

$$E(\boldsymbol{\theta}) = -\hat{\mathbf{L}} \ln\left(CNN(\mathbf{X},\boldsymbol{\theta})\right) + \gamma \left\|\boldsymbol{\theta}\right\|_2 = -\hat{\mathbf{L}} \ln \mathbf{L} + \gamma \left\|\boldsymbol{\theta}\right\|_2 \quad (11)$$

where $\left\|\boldsymbol{\theta}\right\|_2$ is a l_2 normalization term which also contributes to preventing the network from overfitting. Equation (11) quantifies the difference between correct output labels and predicted labels. And the loss is then back-propagated to update the parameters using the stochastic gradient descent (SGD) method [37] given as,

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \nabla E(\boldsymbol{\theta}_i) + \beta(\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}) \quad (12)$$

where α is the learning rate, i is the number of iteration, and β stands for the contribution of previous gradient step. While classical SGD and momentum SGD are frequently adopted in training CNNs for their simplicity and efficiency, other techniques, such as AdaGrad, AdaDelta or Adam [38] can also be applied to carry out optimization of Equation (11). The transferability of the base architecture and the performance of the proposed architecture for gear fault diagnosis will be investigated in the next section.

III. Gear Fault Diagnosis Implementation and Demonstration

A. Data acquisition

Many types of faults and failure modes can occur to gear transmission in various machinery systems. Vibration signals collected from such a system are usually used to reveal its health condition. In this research, experimental data are collected from a benchmark two-stage gearbox with replaceable gears as shown in Figure 7. The gear speed is controlled by a motor. The torque is supplied by a magnetic brake which can be adjusted by changing its input voltage. A 32-tooth pinion and an 80-tooth gear are installed on the first stage input shaft. The second stage consists of a 48-tooth pinion and 64-tooth gear. The input shaft speed is measured by a tachometer, and gear vibration signals are measured by an accelerometer. The signals are recorded through a dSPACE system (DS1006 processor board, dSPACE Inc.) with sampling frequency of 20 KHz. As shown in Figure 8, nine different gear conditions are introduced to the pinion on the input shaft, including healthy condition, missing tooth, root crack, spalling, and chipping tip with five different levels of severity. The dynamic responses of a system involving gear mechanism are angle-periodic. In reality, while gearbox system is recorded in a fixed sampling rate, the time-domain responses are generally not time-periodic due to speed variations under load disturbance, geometric tolerance, and motor control error etc [13]. In order to solve the non-stationary issue and eliminate the uncertainty caused by speed varying, here we apply the time synchronous averaging (TSA) approach, where the time-even signals are resampled based on the shaft speed measured by the tachometer and averaged in angular domain. As TSA converts the signals from the time-even representation to the angle-even representation, it can significantly reduce the non-coherent components in the system response. It is worth mentioning that TSA is a standard, non-biased technique that can facilitate effective pattern recognition of various datasets [13].

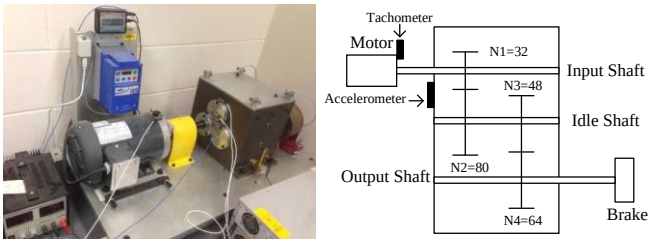


Figure 7 Gearbox system employed in experimental study.

Figure 8 Nine pinions with different health conditions (five levels of severity for chipping tip).

To proceed, in this research we adopt a preprocessing-free approach to transform the vibration signals to images in order to discover the 2D features of raw signals. As time domain vibration signals have been cast into angle-even domain for consistency as sample points (Figure 9(a)), the adjacent data points are then connected in chronological sequence to generate a polyline. Figure 9(b) shows an example of such polyline represented in an 875×656 image generated by MATLAB plot function. The original matrix or image representation of the vibration signal is then resized to a 227×227 gray scale image using Bicubic interpolation [39] as shown in Figure 9(c). There are 51,529 pixels per image. Figure 10 showcases some example images generated from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

angle-even vibration signals. For each gear condition, 104 signals are collected using the experimental gearbox system. For each signal, 3,600 angle-even samples are recorded in the course of four gear revolutions first for the case study in Section III.C, and then down-sampled to 900 angle-even points for the case study in Section III.D. Figure 10 shows 20 example signals of each type of gear condition where the vertical axis is the acceleration of the gear (rad/s^2) and the horizontal axis corresponds to the 3,600 angel-even sampling points. All the data used in this study is made public at https://doi.org/10.6084/m9.figshare.6127874.v1.

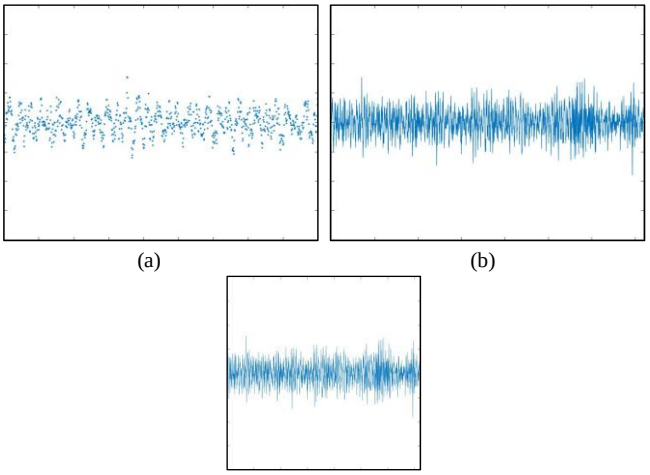
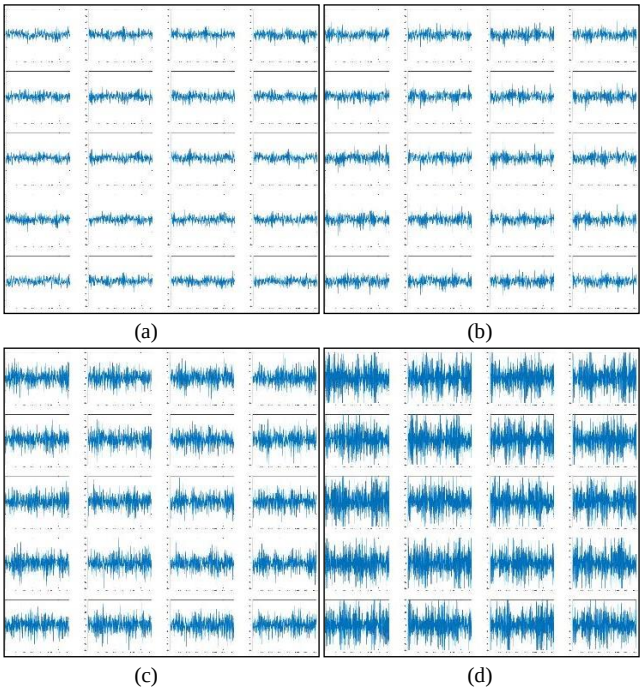


Figure 9 Construction of input for transfer leaning. (a) 875*656 image representation of 3600 samples, (b) 875*656 image representation of the samples connected, (c) 227*227 image representation of the samples connected.



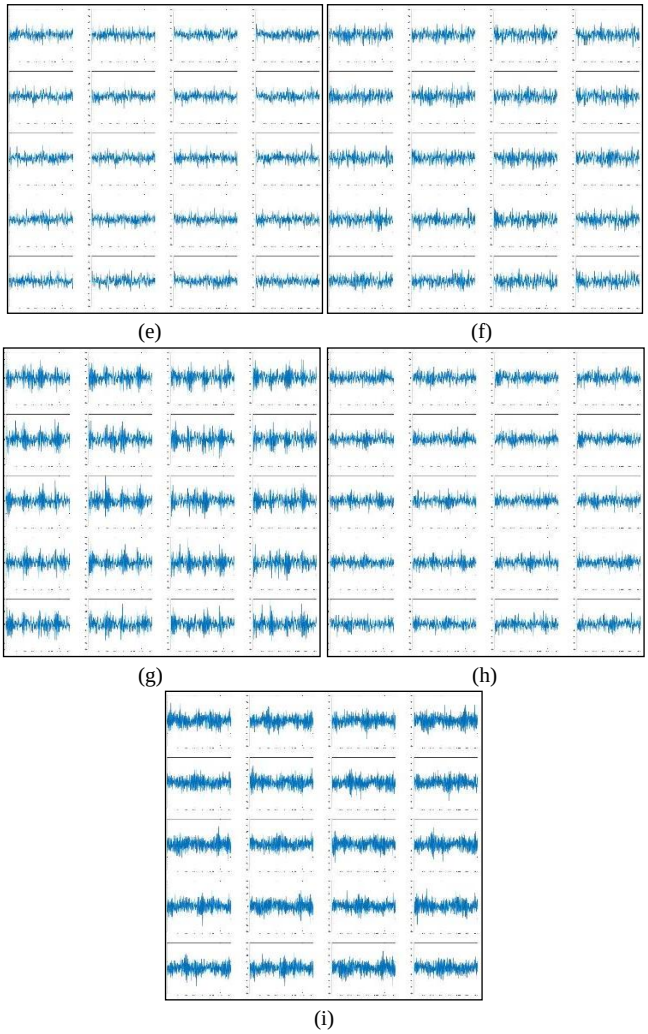


Figure 10 Vibration signal examples under different gear health conditions. (a) Healthy, (b) Missing tooth, (c) Root crack, (d) Spalling, (e) Chipping tip_5 (least severe), (f) Chipping tip_4, (g) Chipping tip_3, (h) Chipping tip_2, (i) Chipping tip_1 (most severe).

B. Setup of case illustration and comparison

In this study, in order to highlight its effectiveness, the proposed transfer learning approach is examined and compared with two contemporary approaches. As indicated, the proposed transfer learning approach does not rely on manual selection of features, and we use this approach to analyze the angle-even representation of the original time-domain signals. The first approach adopted for comparison is a three-stage (nine layers) CNN, thereafter referred to as local CNN, which consists of two convolutional stages and a fully connected stage and uses the angle-even representation of the time-domain signals as inputs. Different from the proposed approach, the local CNN will be only trained by the data generated from gearbox experiments. The specifications are the same as the stage 1, stage 2 and stage 8 given in Table 1. The other approach adopted for comparison is based upon manual identification/selection of features. In a recent investigation, it was recognized that the angle-frequency domain synchronous analysis (AFS) can enhance significantly fault-induced features in gearbox responses [13]. AFS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

resamples the time-domain signal into angle-domain based on the speed information collected from tachometer. The angle-domain signal is then sliced into a series of segments every four gear revolutions. Subsequently, angle-frequency analysis based on short time Fourier Transform is carried out on each segment of the angle-domain signal. The resultant spectrogram coefficients are then averaged to remove the noise and non-coherent components. As such, the features related to the gear health conditions are highly enhanced and a feature extraction technique, i.e. Principal Component Analysis, is employed to reduce the dimensionality. In this research, these low-dimensional data extracted by AFS are imported into support vector machine (SVM) for fault classification.

For the proposed transfer learning approach and the locally-trained CNN approach (local CNN), mini-batch size is set to 5, and 15 epochs are conducted meaning the training datasets are used to train the neural net 15 times throughout. The learning rate α is set to be $1e^{-4}$ and $1e^{-2}$ for transferred layers and non-transferred layers, respectively, following the suggestion in [28]. The momentum β in Equation (12) is set to 0.9 for transfer learning and 0.5 for local CNN. For the SVM approach based on manual feature selection, Gaussian kernel is adopted. In the next two sub-sections, the relative performance of the three approaches is highlighted as we change the sampling frequency as well as the size of the training dataset, i.e., the portion of measured gear vibration signals used for training.

Neural networks are inherently parallel algorithms. Therefore, graphical processing units (GPUs) are frequently adopted as the execution environment to take advantage of the parallelism natural of CNNs and expedite the classification process. In this research, both CNNs are trained and implemented using a single CUDA-enabled NVIDIA Quadro M2000 GPU, while AFS-SVM approach is facilitated based on an Intel Xenon E5-2640 v4 CPU.

C. Case 1 – 3,600 sampling points with varying training data size

As mentioned in Section III.A, 104 vibration signals are generated for each gear condition. In the case studies, a portion of the signals are randomly selected as training data while the rest serves as validation data. To demonstrate the performance of the proposed approach towards various data sizes, the size of the training dataset ranges from 80% (83 training data per condition, 83×9 data in total) to 2% (2 training data per condition, 2×9 data in total) of all the 104 signals for each health condition.

Table 2 shows the classification results where the mean accuracy is the average of five training attempts. The classification accuracy is the ratio of the correctly classified validation data to the total validation dataset. As illustrated in Figure 11, the proposed transfer learning approach has the best classification accuracy for all types of data size. Even when only five vibration signals per condition are selected for training, the proposed approach is able to achieve an excellent 94.90% classification accuracy, which further increases to 99%-100% when 10% and more training data are used. On the other hand, while the performance of AFS-SVM reaches the plateau (showing only minimal increments) after 20% date is used for training, the classification accuracy of local CNN gradually increases with data size from 27.99% to 97.57% and surpasses AFS-SVM eventually when 80% data is used for training, indicating the significance of the size of training data in order to properly train a neural network. Although the data size greatly affects the performance of a CNN in the general sense, the proposed transfer learning architecture exhibits very high classification accuracy because only one fully connected stage needs to be trained locally, which notably lowers the standard of the data required by a CNN in terms of achieving satisfactory outcome.

Table 2 Classification results (3,600 sampling points)

Method / Training data	Transfer learning Accuracy (%)		Local CNN Accuracy (%)		AFS-SVM Accuracy (%)	
80% (83 per condition)	100 100 100 100 100	Mean: 100	91.01 99.47 97.35 100 100	Mean: 97.57	86.72 88.62 87.80 87.26 86.99	Mean: 87.48
60% (62 per condition)	100 100 100 100 100	Mean: 100	90.48 97.62 58.99 88.89 67.72	Mean: 80.74	87.30 87.83 88.62 87.04 87.83	Mean: 87.72
40% (42 per condition)	100 100 100 100 100	Mean: 100	88.89 98.39 44.44 62.72 83.69	Mean: 76.63	86.74 86.38 85.84 87.99 86.38	Mean: 86.67
20% (21 per condition)	100 100 100 99.60 100	Mean: 99.92	61.31 72.56 85.41 70.41 58.77	Mean: 69.69	86.48 86.08 85.01 86.35 87.28	Mean: 86.24
10% (10 per condition)	99.88 98.23 99.88 99.29 99.76	Mean: 99.41	64.07 57.09 55.56 44.56 57.80	Mean: 55.82	80.97 86.17 78.84 86.29 86.88	Mean: 83.83
5% (5 per condition)	99.55 97.19 80.02 98.09 99.66	Mean: 94.90	65.54 37.71 31.99 28.17 57.13	Mean: 44.11	75.31 84.85 81.14 73.29 84.85	Mean: 79.89
2% (2 per condition)	76.80 73.31 69.39 73.42 68.19	Mean: 72.22	26.14 27.67 32.24 31.70 22.22	Mean: 27.99	61.87 73.97 41.72 69.72 64.92	Mean: 62.44

The average computational time consumed by each method is reported in Table 3, which contains the portions used for both training and classification. Generally speaking, deep neural networks are more time consuming in training compared to traditional approaches. The computational cost per iteration of a mini-batch back-propagation is proportional to the number of weights involved. And the overall

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

computational time is linearly proportional to the size of the training data. As shown in Table 3, when the size of training data is small (2%), the transfer learning approach not only leads in accuracy, but also in computational efficiency compared to AFS-SVM.

Table 3 Computational time comparison (average of 5 attempts)

Method / Training data	Transfer learning (sec)	Local CNN (sec)	AFS-SVM (sec)
80%	588.467	373.283	52.156
60%	453.063	284.445	50.581
40%	311.824	198.517	48.181
20%	167.406	108.909	48.046
10%	98.872	64.747	47.998
5%	66.152	42.800	47.846
2%	42.840	28.847	47.781

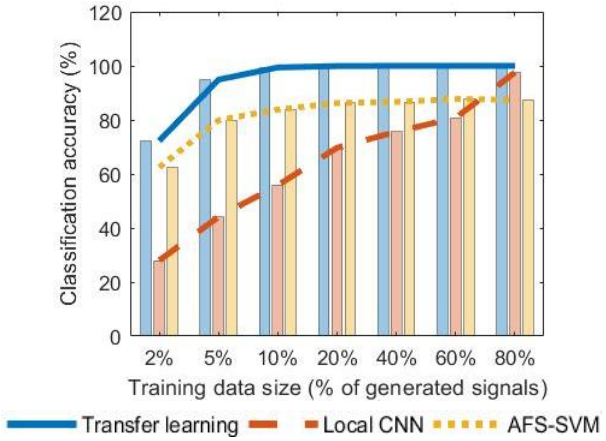


Figure 11 Comparison of classification results when training data size varies.

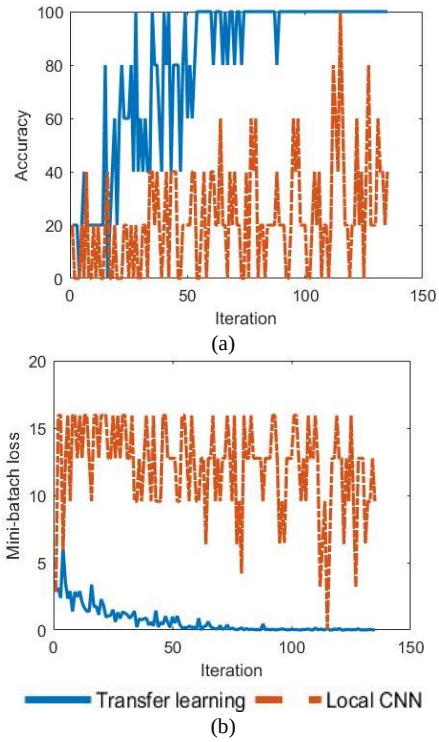


Figure 12 Convergent histories of transfer learning and local CNN for 5% training data. (a) Accuracy, (b) Mini-batch loss.

Figure 12 shows the convergent histories (mini-batch accuracy and mini-batch loss) of the proposed approach and local CNN when 5% data is used for training. As can be seen from the comparisons, transfer learning gradually converges in terms of both accuracy and loss as the training iterates while local CNN inclines to 'random walk' due to insufficient data. Compared with AFS-SVM, the proposed approach not only excels in performance, but also requires no preprocessing effort, which makes it more unbiased in feature extraction and readily applicable to other fault diagnosis practices. The proposed approach also shows satisfactory outcomes in the regard of robustness. As demonstrated in Figure 13, it has the smallest variance among all cases. On the other hand, the performance of the under-trained local CNN oscillates the most.

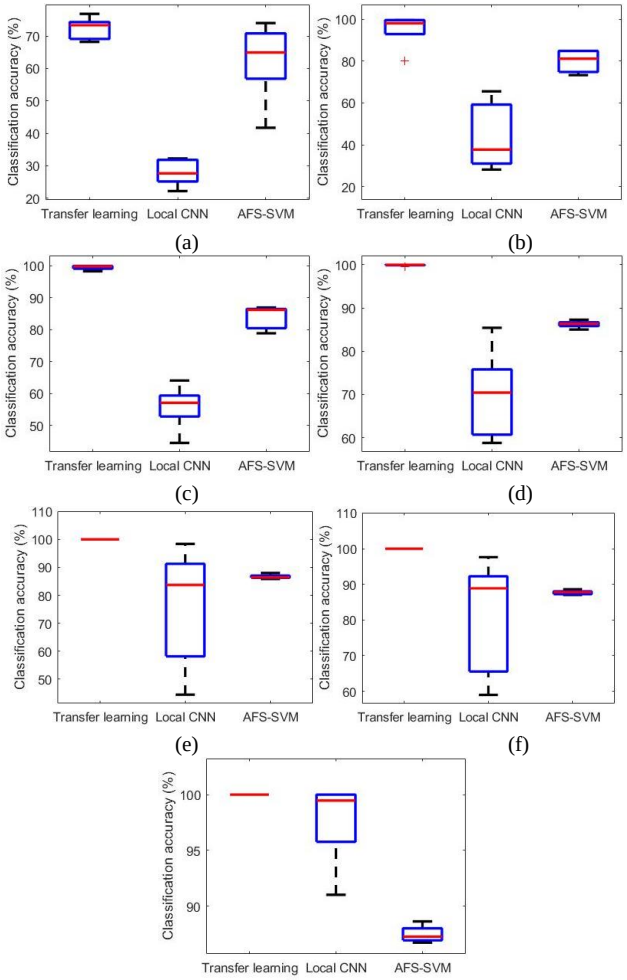


Figure 13 Comparison of box plots of classification results when training data size varies.
(a) 2%, (b) 5%, (c) 10%, (d) 20%, (e) 40%, (f) 60%, (g) 80%

As mentioned in Section II.C, the parameters in the first five convolutional stages of the original CNN are well-trained in characterizing high-level abstractions while the last three fully connected stages are trained to nonlinearly combine the high-level features. Hence, it is recommended to repurpose Stages 1 to 5 for novel tasks as to adaptively extract image

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

features. Whether to transfer Stages 6 and 7 remains optional depending on the training data size. In our previous comparisons, only Stage 8 is reconstructed (from 1000 classes to 9 classes) and trained using local dataset. Here, we also compare the accuracy of the transfer learning approach when different aggregates are transferred. As shown in Table 4, transferring Stages 1 to 7 and transferring Stages 1 to 6 yield similar performances, which are better than transferring merely Stages 1 to 5 especially when data size is small. Recall Table 1. Stage 6 contains 4096 more weighting parameters, which apparently requires more training data to fine-tune even though the feature extraction passage is well-established. Moreover, transferring more layers may indeed prevent the model from overfitting because the layers transferred are already extensively trained so the generalization error is naturally reduced when the model is repurposed and only a small portion is trained by a different set of data.

Table 4 Classification results of transfer learnings (average of 5 attempts)

Method / Training data	Transfer learning (Stages 1-7) Accuracy (%)	Transfer learning (Stages 1-6) Accuracy (%)	Transfer learning (Stages 1-5) Accuracy (%)
80%	100	100	100
60%	100	100	100
40%	100	100	99.97
20%	99.92	99.87	99.06
10%	99.41	99.28	72.50
5%	94.90	96.30	64.54
2%	72.22	72.98	48.91

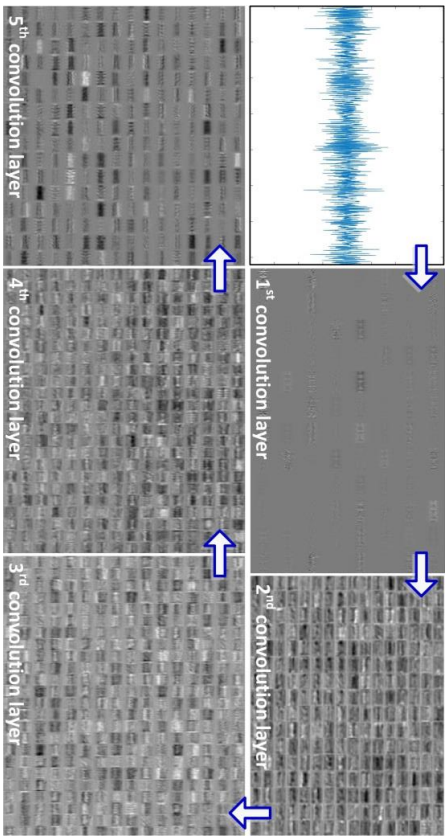
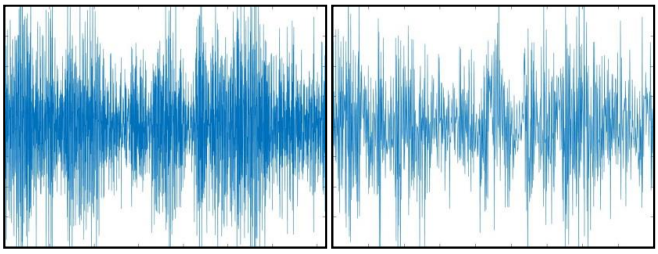


Figure 14 Feature maps extracted by 5 convolution layers of the proposed transfer learning approach.

As discussed in Section II.B and Section II.C, the transferred stages of the proposed architecture tend to extract the high-level abstract features of the input that cannot be recognized otherwise, even if the input is different from that of the previous task. Figure 14 gives an example of such procedure by showing the feature maps generated in each convolutional layer by the proposed architecture when it is used to classify a gearbox vibration signal. It is seen that the abstraction level of the input image continuously escalates from the 1st feature map to the 5th feature map. In general, the number of convolutional stages equipped is correlated with the level of abstraction the features can be represented in CNNs. As demonstrated in this case study, the base architecture is indeed transferable towards gear fault diagnosis tasks and the proposed approach performs well with raw image signal inputs, which indicates the transferred layers constructed in this study are generally applicable to represent useful features of an input image in high-level abstraction.

D. Case 2 – 900 sampling points with varying training data size

In Case 1, each vibration signal is composed of 3,600 angel-even data points in the course of 4 gear revolutions. In some practical fault diagnosis systems, however, the sampling rate may be lower, which means that some features could have been lost. To take this factor into consideration and further examine the approach, we now down-sample the original vibration signals to 900 angel-even data points (Figure 15) and apply the same three methods for classification.



(a) (b)

Figure 15 Vibration signal of a spalling gear.
(a) 3,600 sampling points, (b) 900 sampling points

Table 5 lists the comparison of the classification results of the three methods with different training data sizes. Similar to Case 1, the proposed transfer learning approach is the best performer. Figure 16 illustrates the classification results before and after down-sampling. While lowering the sampling rate deteriorates the overall performance of all approaches, each method exhibits the similar trend as seen in Section III.C. For transfer learning, it starts with 60.11% classification accuracy and reaches 95.88% when only 20% of data is used as training data whilst the accuracies of local CNN and AFS-SVM are 43.56% and 70.07%. Local CNN performs better than AFS-SVM when 80% data is used for training. Unlike AFS-SVM, the performance of local CNN can be largely improved if significantly more training data is incorporated because the parameters of lower stages can be learned from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837621, IEEE Access

scratch. Eventually, the performance of local CNN could reach that of the transfer learning approach. Nevertheless, for cases with limited data, the proposed transfer learning approach has an extensive performance margin compared to local CNN or other preprocessing-based shallow learning methods such as AFS-SVM. Even with ample training data, initializing with transferred parameters can improve the classification accuracy in general. Moreover, the proposed approach requires no preprocessing. Similar to Case 1 in Section III.C, the proposed approach is very robust especially when 40% or more data is used for training (Figure 17).

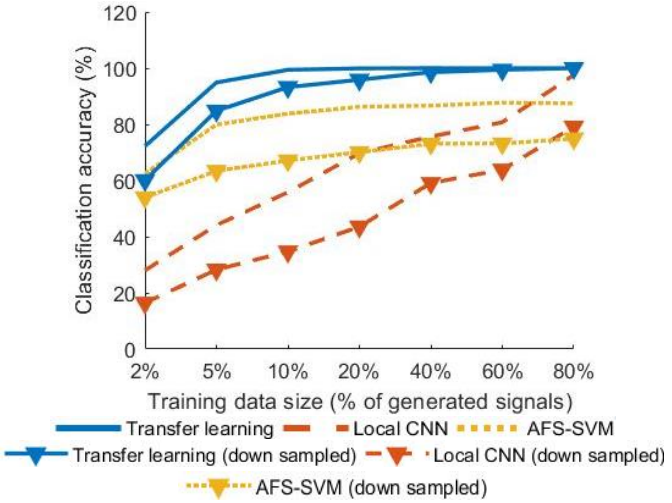


Figure 16 Classification results of the three methods after down sampling.

Table 5 Classification results (900 sampling points)

Training data \ Method	Transfer learning Accuracy (%)		Local CNN Accuracy (%)		AFS-SVM Accuracy (%)	
80% (83 per condition)	100 100 100 100 100	Mean: 100	85.26 65.66 71.32 80.89 92.53	Mean: 79.13	74.07 74.60 75.13 76.72 74.07	Mean: 74.92
60% (62 per condition)	100 99.21 99.74 98.68 99.47	Mean: 99.42	77.25 57.67 63.76 72.22 48.41	Mean: 63.86	75.40 74.34 71.16 74.07 70.90	Mean: 73.17
40% (42 per condition)	99.10 99.10 98.92 98.92 96.77	Mean: 98.56	62.90 74.91 56.63 38.35 62.72	Mean: 59.10	74.19 72.94 72.58 73.66 72.22	Mean: 73.12
20% (21 per condition)	94.91 95.72 92.77 98.80 97.19	Mean: 95.88	34.27 40.56 44.44 44.71 53.82	Mean: 43.56	70.15 72.69 68.41 69.21 69.88	Mean: 70.07
10% (10 per condition)	94.68 93.38 90.07 92.08 95.98	Mean: 93.24	27.78 39.83 46.57 17.97 41.37	Mean: 34.70	68.20 68.68 65.96 66.78 65.96	Mean: 67.12
5% (5 per condition)	70.73 86.65 89.12 90.24	Mean: 84.83	24.88 15.80 23.20 33.40	Mean: 28.27	64.42 62.96 63.86 65.66	Mean: 63.43
2% (2 per condition)	87.43 55.99 59.91 58.06 61.11 65.47	Mean: 60.11	44.05 21.90 17.43 22.22 9.59 11.11	Mean: 16.45	60.27 47.93 57.30 58.06 51.53 55.12	Mean: 53.99

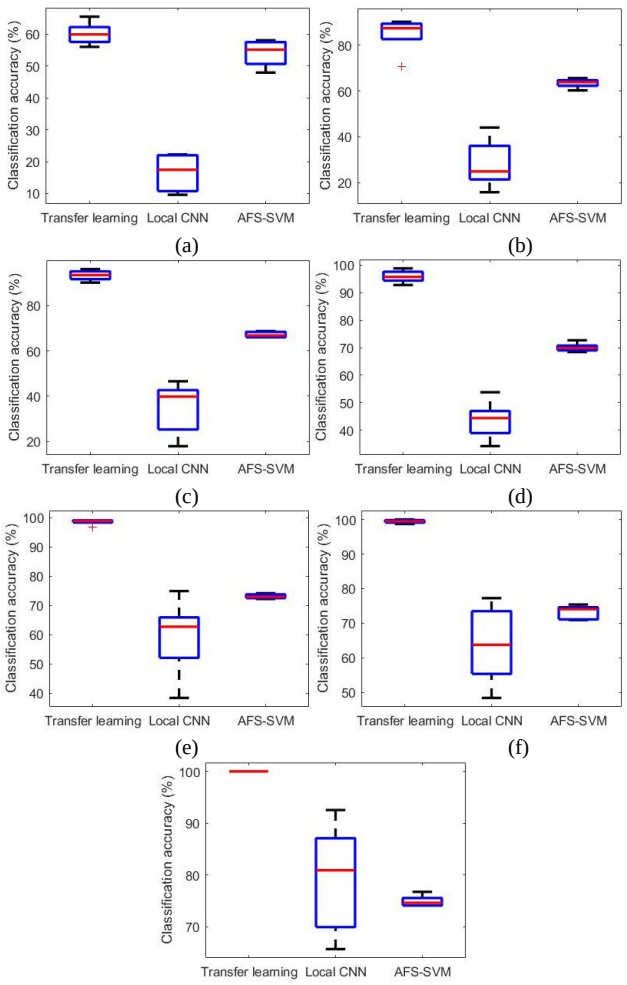


Figure 17 Box plots of classification results of the three methods after down sampling.
(a) 2%, (b) 5%, (c) 10%, (d) 20%, (e) 40%, (f) 60%, (g) 80%

IV. Concluding Remarks

In this research, a deep convolutional neural network-based transfer learning approach is developed for deep feature extraction and applied to gear fault diagnosis. This proposed approach does not require manual feature extraction, and can be effective even with a small set of training data. Experimental studies are conducted using preprocessing free raw vibration data towards gear fault diagnose. The performance of the proposed approach is highlighted through varying the size of training data. The classification accuracies of the proposed approach outperform those of other methods such as locally trained convolutional neural network and angle-frequency analysis-based support vector machine by as much as 50%. The achieved accuracy indicates that the proposed approach is not only viable and robust, but also has

the potential to be applied to fault diagnosis of other systems.

REFERENCES

[1] Kang, D., Xiaoyong, Z. and Yahua, C., 2001. The vibration characteristics of typical gearbox faults and its diagnosis plan. *Journal of vibration and shock*, *20*(3), pp.7-12.

[2] R.B. Randall, Vibration-based condition monitoring: industrial, aerospace and automotive applications, John Wiley & Sons, West Sussex, United Kingdom, 2011.

[3] F.P.G. Márquez, A.M. Tobias, J.M.P. Pérez, M. Papaelias, Condition monitoring of wind turbines: Techniques and methods, Renewable Energy, 46 (2012) 169-178.

[4] Zhou, W., Habetler, T.G. and Harley, R.G., 2008. Bearing fault detection via stator current noise cancellation and statistical control. *IEEE Transactions on Industrial Electronics*, *55*(12), pp.4260-4269.

[5] Parey, A. and Pachori, R.B., 2012. Variable cosine windowing of intrinsic mode functions: Application to gear fault diagnosis. *Measurement*, *45*(3), pp.415-426.

[6] T. Fakhfakh, F. Chaari, M. Haddar, Numerical and experimental analysis of a gear system with teeth defects, International Journal of Advanced Manufacturing Technology, 25 (2005) 542-550.

[7] Li, D.Z., Wang, W. and Ismail, F., 2015. An enhanced bispectrum technique with auxiliary frequency injection for induction motor health condition monitoring. *IEEE Transactions on Instrumentation and Measurement*, *64*(10), pp.2679-2687.

[8] W. Wen, Z. Fan, D. Karg, W. Cheng, Rolling element bearing fault diagnosis based on multiscale general fractal features, Shock and Vibration, 2015 (2015).

[9] B. Tang, W. Liu, T. Song, Wind turbine fault diagnosis based on Morlet wavelet transformation and Wigner-Ville distribution, Renewable Energy, 35 (2010) 2862-2866.

[10] Chaari, F., Bartelmus, W., Zimroz, R., Fakhfakh, T. and Haddar, M., 2012. Gearbox vibration signal amplitude and frequency modulation. *Shock and Vibration*, *19*(4), pp.635-652.

[11] R. Yan, R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: a review with applications, Signal Processing, 96 (2014) 1-15.

[12] Chen, X. and Feng, Z., 2017. Time-Frequency Analysis of Torsional Vibration Signals in Resonance Region for Planetary Gearbox Fault Diagnosis Under Variable Speed Conditions. *IEEE Access*, *5*, pp.21918-21926.

[13] Zhang, S. and Tang, J., 2018. Integrating angle-frequency domain synchronous averaging technique with feature extraction for gear fault diagnosis. *Mechanical Systems and Signal Processing*, *99*, pp.711-729.

[14] Pachaud, C., Salvetat, R. and Fray, C., 1997. Crest factor and kurtosis contributions to identify defects inducing periodical impulsive forces. *Mechanical systems and signal processing*, *11*(6), pp.903-916.

[15] Qian, S. and Chen, D., 1999. Joint time-frequency analysis. *IEEE Signal Processing Magazine*, *16*(2), pp.52-67.

[16] Baydar, N. and Ball, A., 2001. A comparative study of acoustic and vibration signals in detection of gear failures using Wigner–Ville distribution. *Mechanical systems and signal processing*, *15*(6), pp.1091-1107.

[17] Bartelmus, W. and Zimroz, R., 2009. Vibration condition monitoring of planetary gearbox under varying external load. *Mechanical Systems and Signal Processing*, *23*(1), pp.246-257.

[18] Lin, J. and Zuo, M.J., 2003. Gearbox fault diagnosis using adaptive wavelet filter. *Mechanical systems and signal processing*, *17*(6), pp.1259-1269.

[19] Lu, Y., Tang, J. and Luo, H., 2012. Wind turbine gearbox fault detection using multiple sensors with features level data fusion. *Journal of Engineering for Gas Turbines and Power*, *134*(4), p.042501.

[20] Zhang, C., Sun, J.H. and Tan, K.C., 2015, October. Deep belief networks ensemble with multi-objective optimization for failure diagnosis. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on* (pp. 32-37). IEEE.

[21] Li, C., Sanchez, R.V., Zurita, G., Cerrada, M., Cabrera, D. and Vásquez, R.E., 2016. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing*, *76*, pp.283-293.

[22] Weimer, D., Scholz-Reiter, B. and Shpitalni, M., 2016. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals-Manufacturing Technology*, *65*(1), pp.417-420.

[23] Ince, T., Kiranyaz, S., Eren, L., Askar, M. and Gabbouj, M., 2016. Real-time motor fault detection by 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics*, *63*(11), pp.7067-7075.

[24] Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M. and Inman, D.J., 2017. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, pp.154-170.

[25] Saravanan, N. and Ramachandran, K.I., 2010. Incipient gear box fault diagnosis using discrete wavelet transform (DWT) for feature extraction and classification using artificial neural network (ANN). *Expert Systems with Applications*, *37*(6), pp.4168-4181.

[26] Yang, J., Li, S. and Xu, W., 2018. Active Learning for Visual Image Classification Method Based on Transfer Learning. *IEEE Access*, *6*, pp.187-198.

[27] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396-404).

[28] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[29] Kim, Y., 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[30] Van den Oord, A., Dieleman, S. and Schrauwen, B., 2013. Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).

[31] Shie, C.K., Chuang, C.H., Chou, C.N., Wu, M.H. and Chang, E.Y., 2015, August. Transfer representation learning for medical image analysis. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE* (pp. 711-714). IEEE.

[32] Zhang, R., Tao, H., Wu, L. and Guan, Y., 2017. Transfer Learning with Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions. *IEEE Access*, *5*, pp.14347-14357.

[33] Zeiler, M.D. and Fergus, R., 2013. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.

[34] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

[35] Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems* (pp. 3320-3328).

[36] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15(1)*, pp.1929-1958.

[37] Sutskever, I., Martens, J., Dahl, G. and Hinton, G., 2013, February. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*(pp. 1139-1147).

[38] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[39] Prashanth, H.S., Shashidhara, H.L. and KN, B.M., 2009, December. Image scaling comparison using universal image quality index. In *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on* (pp. 859-863). IEEE.