

Research Article

FWDNet: A Novel Recognition Network for Ferrography Wear Debris Image Analysis

Fengguang Jia ^{1,2} and Haijun Wei¹

¹*School of Merchant Marine, Shanghai Maritime University, Shanghai 201306, China*

²*School of Naval Architecture and Port Engineering, Shandong Jiaotong University, Weihai 264209, China*

Correspondence should be addressed to Fengguang Jia; dz_jfg@163.com

Received 2 December 2021; Revised 15 December 2021; Accepted 24 December 2021; Published 18 January 2022

Academic Editor: Deepak Kumar Jain

Copyright © 2022 Fengguang Jia and Haijun Wei. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ferrography wear debris in lubricating oil contains abundant worthy information about the state of the machinery and equipment. In order to develop an online monitoring system based on condition maintenance and fault diagnosis, wear debris needs to be identified automatically. Through various tribological experiments, a dataset of seven kinds of wear debris was established. In this study, DenseNet121 was used as the base network to construct a DCNN model (FWDNet) using the transfer learning method. FWDNet obtained an accuracy of 90.15% through a 10-fold crossvalidation test. The results indicate that FWDNet and DCNN mode is suitable for the identification of wear debris and can be used in actual condition monitoring systems in the future.

1. Introduction

One important trend for mechanical equipment maintenance is the application of condition monitoring techniques. The idea is to analyze real-time data to monitor the health of the equipment and predict the occurrence of faults [1, 2]. Currently, techniques that get focused on include vibration analysis, acoustic emission with ultrasound, oil analysis, and wear debris analysis (WDA). Each one has its own advantages and constraints. It is widely accepted that a single technique cannot meet the requirements for all situations; still, wear debris analysis is considered as one of the most effective approaches by many users of condition-based maintenance techniques for the following reasons.

Firstly, wear debris is usually produced in the friction pairs that move with each other. The relevant parameters of wear debris, such as size, shape, and surface morphology can reflect the wear mechanism and judge the position and severity of wear generation [3, 4].

Secondly, the patterns of quantity and type of wear debris changes over time have a strong correlation with the state of the machine. It is possible to predict potential fail-

ures or component's deterioration from a very early stage to avoid catastrophic accidents [5, 6].

Although WDA can provide a lot of information for problem detection and fault diagnosis, the technique has not been widely used in the industry for that the classification of wear debris, and the equipment fault diagnosis technology relies heavily on the experience of operators. These limitations lead to the need for automation of this technology from which the industry will undoubtedly facilitate rapid diagnosis of equipment and reduce the need for human resources [7].

Nowadays, the advancement and availability of the instrument make it possible to wear debris classification and equipment fault diagnosis [8]. Therefore, the method of observing and analyzing the morphological characteristics of wear debris is a promising online solution [9]. Until now, researchers have done a lot of work in the establishment of an automatic classification system of wear debris. It is reported that various machine learning algorithms can be used to distinguish wear debris using geometric parameters, such as area, perimeter, and elongation parameters [10, 11]. The results show that these morphological characteristics

can distinguish certain kinds of wear debris effectively. However, there are several kinds of abnormal debris larger than $20\ \mu\text{m}$ cannot be accurately identified. In the study by Stachowiak et. al., they used scanning electron microscopy to observe and analyze the morphological characteristics of abrasive, fatigue, and sliding debris, such as shape, color, and texture [12–14]. Studies have shown that these simple features can distinguish between spherical, cutting, severe sliding, and nonferrous metal debris. Wu et al. had done a lot of research on the real-time online detection of wear debris [15]. Based on a watershed algorithm for gray image segmentation, the team developed a set of online visual ferrography (OLVF). It made some progress in wear debris extraction and overlapping particle separation. However, the device did not obtain the morphological characteristics of the debris; so, it could not accurately reflect the wear pattern and position. In conclusion, the intelligent identification of ferrography wear debris images had been studied extensively, and the recognition of simple wear debris, such as cutting and colored debris, had been basically realized. However, there are still many problems in the realization of complex wear debris and classification system, such as the identification of severe sliding, lamellar fatigue, and spalling fatigue debris, which are still the technical bottlenecks of intelligent identification [16, 17].

Deep convolutional neural network (DCNN) is a branch of deep learning, which includes a feedforward neural network with convolution computation and deep structure [18]. With the development of deep learning theory and the improvement of numerical computing equipment, DCNN has been developed rapidly and applied to computer vision, natural language processing, and other fields [19, 20]. DCNN has broad prospects for wear debris' detection; yet, relevant research is nearly blank so far.

The research objective is to establish a deep convolutional neural network for debris' identification from scratch, solving practical problems such as generating different kinds of wear particles, DCNN model selection, and transfer learning testing [21]. Using this network to complete the task of accurately identifying severe sliding, lamellar fatigue, spalling fatigue, cutting, normal, spherical, and nonferrous metal debris provides technical support for the online application of ferrography. The main reason why we use the above seven types of wear debris is that they are common in the process of wear and can reflect the wear mechanism and machine state.

2. Methodology

2.1. Dataset. In this study, according to the generation mechanism of severe sliding, fatigue laminar, chunky spalling wear debris, and under the condition of strict control of temperature, load, and speed, the experiment was carried out by Bruker Universal Mechanical Tester shown in Figure 1(a), which had a multipurpose foundation that can be fitted with a series of driving modules that simulate rotation, linear or oscillatory motion, and an upper bracket with a force transducer. Simulating different wear patterns by a series of tribo-

logical experiments had been performed: pin-disk test, reciprocating sliding test, and four-ball test.

Here, Figure 1(b) showed the pin-disk module. The upper sample was a fixed pin made of AISI 420 stainless steel, and the lower sample was a rotating disk made of AISI E52100 steel. 10 ml of Great Wall L-CKT220 lubricating oil was added to the friction pair. The load of the experiment was set to 28 kg (274.4 N) with a rotational speed of 900 r/min and duration of 24 hours. When the friction pairs slid relative to each other, the material was sheared off into wear debris due to the adhesion effect. In the process of the pin-disk test, slight and severe adhesive wear debris was generated.

The reciprocating sliding test was used to generate severe sliding wear debris. The upper sample was a reciprocating pin made of HT250 cast iron, and the lower sample was a fixed disk made of GCr15 steel. 12 ml of Great Wall L-CKT220 lubricating oil was added to the friction pair. The load of the experiment was set to 45 kg (441 N) with a reciprocating frequency of 5 Hz on the stroke of 15 mm and duration of 12 hours. Due to the excessive load and/or the high speed, local adhesion and severe surface plastic flow on the surface of the material, resulting in some severe sliding wear debris.

The fatigue wear debris was generated by a four-ball test machine, and Figure 1(c) shows the four-ball module. The fatigue wear debris was generated by running for 30 hours with a maximum load of 150 kg (1470 N) and a velocity of 300 r/min. The microvolume of the friction surface material was repeatedly deformed by the cyclic contact stress, resulting in cracks and the separation of wear debris.

In addition, we produced cutting debris by adding particles to the system and embedding them into the soft surface and replaced the friction pair with aluminum alloy or copper alloy to produce nonferrous wear debris. Spherical debris is obtained from the lubricating oil of the rolling bearing.

SPECTRO-T2FM500 Ferrography analyzer was shown in Figure 1(d), which can separate wear debris from lubricating oil by magnetic force. Images were photographed by the optical microscope Olympus BX51 with a color charge couple device (CCD) Camera, as shown in Figure 1(e). Most of the pictures were taken under $\times 200$ magnification because it produces a sharp and clear image of the debris of size $20\ \mu\text{m}$ to $200\ \mu\text{m}$, which are of particular interest in this study. According to the size, shape, and surface texture of the wear debris, different types of wear debris images were strictly selected from the original images to make the dataset. The advantage of an optical microscope is that sample preparation is relatively easy, and it is possible to be transformed to real-time on-line image capturing. Besides the pictures were taken in the experiments described above, parts of the images in Wear Particle Atlas were also included in the dataset for diversity. Figure 2 shows some images of wear debris produced in experiments.

To better assess the validity and applicability of the model, the dataset was randomly divided into three parts, namely, the training set, the validation set, and the test set. The training set was applied to build the model. The validation set was applied to determine the network structure or

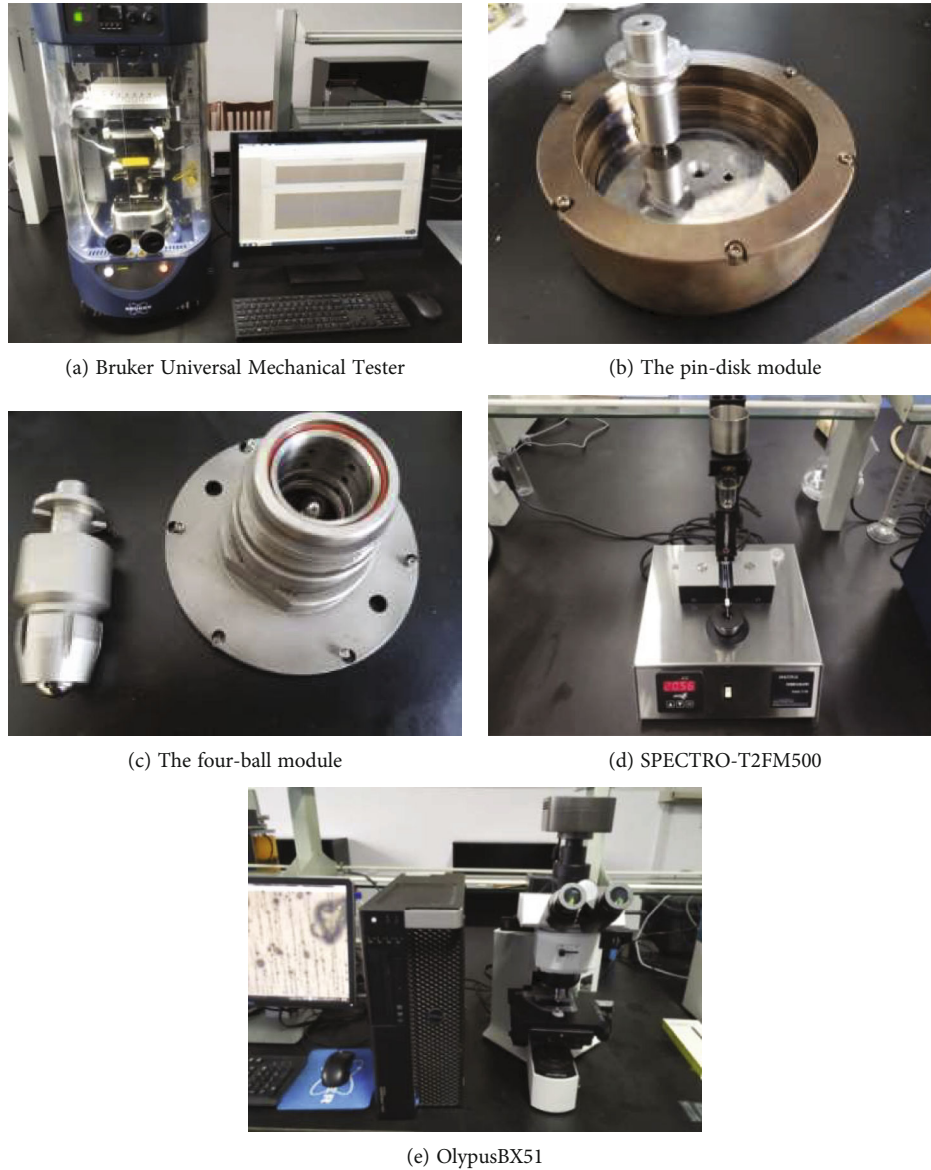


FIGURE 1: Instruments and equipment.

set the hyperparameters of the model. The test set was applied to evaluate the model finally. To prevent overfitting, an important principle was that the test set cannot be analyzed or used in any way until the final model was obtained. In our image dataset, wear debris was split into seven classes: severe sliding, lamellar fatigue, spalling fatigue, cutting, normal, spherical, and nonferrous metal debris. There are a total of 1400 images, 200 for every kind of debris, among which 40 were labeled images in the test set, 40 were in the validation set, and 120 were in the training set. An example of a data splitting strategy was shown in Figure 3.

2.2. Deep Convolutional Neural Network. DEEP convolutional neural network is a common form for image recognition tasks. Krizhevsky used the CNN with extended depth to win the first place in the Image Net competition with the best classification accuracy in 2012 [20]. It is like opening a

window, and DCNN models have been widely used in the field of image identification and classification. DCNN used the local receptive field to form the final global feature by combining the local learned features. When the same convolution kernel operated on different local receptive fields, weight sharing was used, which significantly reduced the amount of parameter calculation in the process of network operation.

The DCNN network mainly consists of an input layer, convolution (CONV) layer, pooling layer, fully connected (FC) layer, and an output layer and is shown in Figure 4. The purpose of the convolution operation is to extract different features of the input. The first CONV layer may only extract some low-level features such as edges, lines, and angles. The network with more layers can extract more complex features iteratively from low-level features [22]. The FC layer usually appears in the last few layers, making weighted

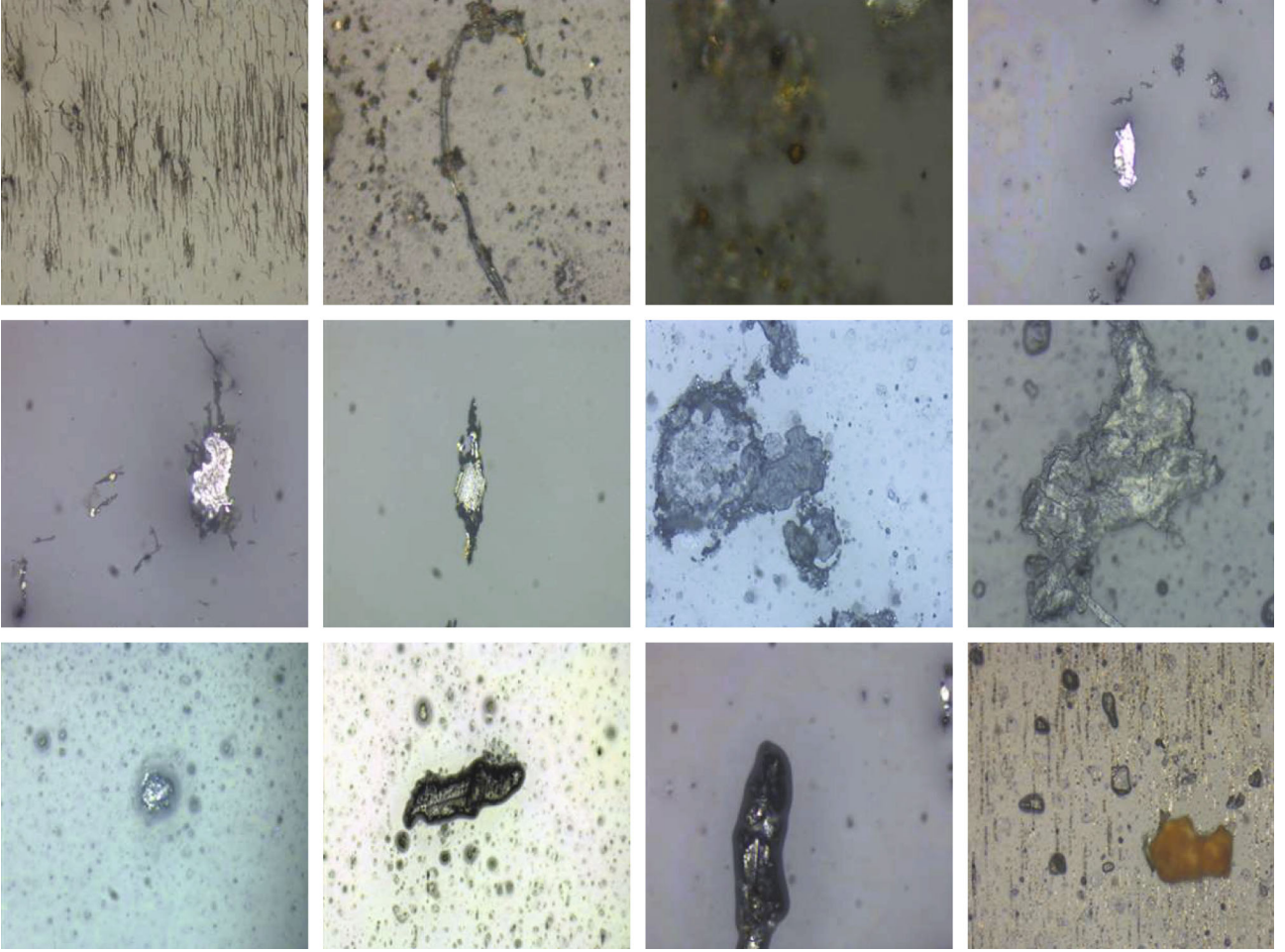


FIGURE 2: Images of wear debris produced in experiments.

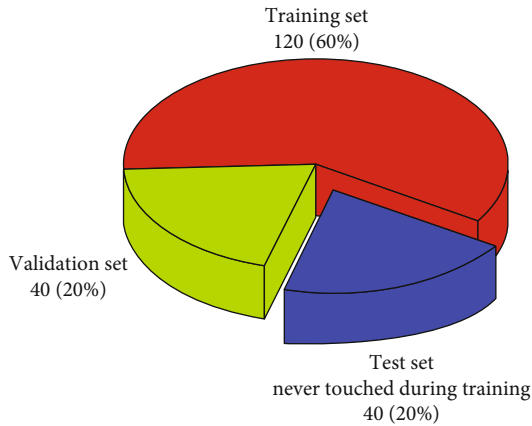


FIGURE 3: Data splitting strategy.

summation for the preceding features. Nonlinearity refers to an activation function that is usually applied after convolution or fully connected layer. The introduction of nonlinear elements into the network enables the network to solve nonlinear problems, such as classification problems. The most representative nonlinearity is ReLU, which can solve the problem of gradient disappearance to a great extent [23].

From the demonstration figure, one can see that there are also pooling and normalization layers. After feature extraction in the CONV layer, the output feature map will be transferred to the pooling layer for feature selection and information filtering. The function of the pooling layer is to replace the result of a single point in the feature map with the statistic of the feature map of its adjacent area. Normalization is an operation to scale data into a small specific interval, which can also solve the problem of gradient disappearance and explosion to some extent [24].

The CONV and FC can be calculated as follows:

$$X_j^l = f \left(\sum_{i \in M_j} X_i^{l-1} \times W_{ij}^l + b_j^l \right), \quad (1)$$

where X_j^l , X_i^{l-1} , W_{ij}^l , and b_j^l are the feature matrices of the output, the input, filters, and biases, i and j represent the components of the input and output matrices, and $f()$ is the activation function.

The pooling can be calculated as follows:

$$X_j^{l+1} = \text{down}(X_j^l), \quad (2)$$

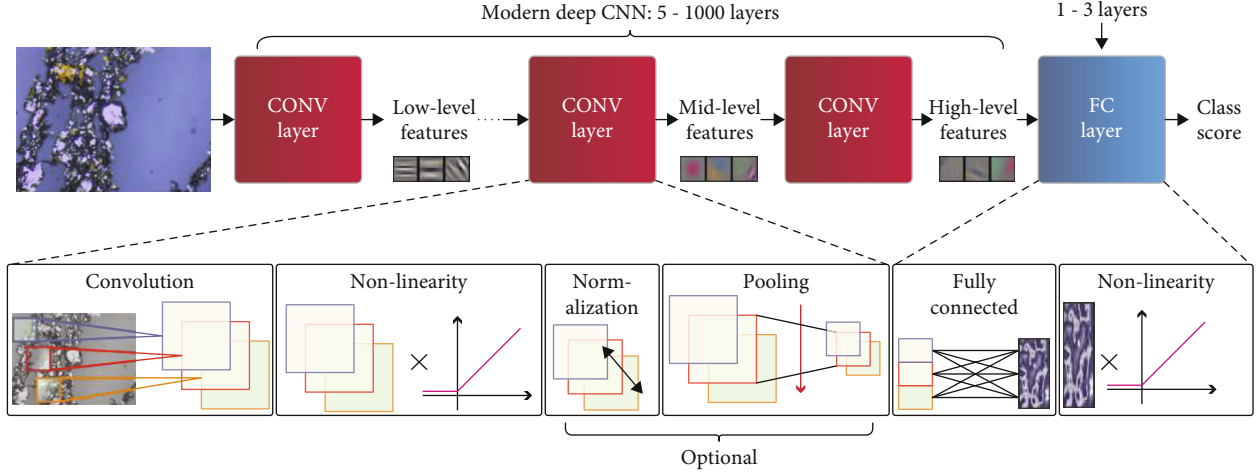


FIGURE 4: The framework of DCNN.

where X_j^{l+1} and X_j^l are the feature matrices of the output and the input, and $\text{down}()$ is the mean or maximum operation of the input feature matrices calculated by a filter with the specified size.

2.3. The Basic Architecture of DCNN. In the past decade, many DCNN models have been developed for image classification, such as AlexNet, VGGNet, GoogLeNet, ResNet, and DenseNet [20, 22, 25–27]. These DCNN models have proved their own effectiveness according to their respective datasets. However, these models have not classified images of wear debris; so, it is difficult to compare their classification performance directly.

We firstly determined a basic architecture of DCNN, which was selected from AlexNet, VGG16, InceptionV3, ResNet50, and DenseNet121, by transfer learning with our image dataset of wear debris. Here, transfer learning refers to fine tuning of the model parameters with the dataset of wear debris based on a model which has been trained on a huge dataset such as ImageNet, so as to achieve an excellent performance for our dataset quickly. Studies have shown that transfer learning is almost always better used on a new dataset.

As shown in Table 1, the architecture of 5 DCNN models has completely different frameworks in terms of the number of layers, convolution kernels, and connection modes. The input size indicates the pixel size of the input image. The data in the CONV block is expressed as “Receptive field, Number of channels.” The data in the dense net is expressed as “Connection type, Number of nodes.”

AlexNet consists of 5 CONV layers and 3 FC layers with a total of 8 layers. In training, we changed the nodes of the output layer from 1000 to 7 for the classification of our wear debris.

VGG16 contains 16 weight layers, including 13 CONV layers and 3 FC layers. This model adopts the strategy of convolution layer stacking, which makes several continuous convolution layers into a convolution layer group instead of pooling immediately after convolution. At the same time, it

unifies the convolution kernel and the pooling size, that is, 3×3 convolution and the maximum pooling.

InceptionV3 has a depth of up to 22 layers. This model is built in the form of network stacking and each subnetwork as an inception module. This model consists of three types of inception modules, named inception A, inception B, and inception C, respectively, and their detailed structure is shown in Figure 5. The inception module has two main functions: one is to filter the input feature map on the convolution kernel of 3×3 and 5×5 , respectively, which improves the diversity of the learned features and enhances the robustness of the network to different scales, and the other is to combine the filtering results of different channels with high correlation through 1×1 convolution, so as to accelerate the convergence.

ResNet50 goes even deeper to 50 layers. The model considers that it is impossible to learn a better network by simply stacking layers. The reason is that, on the one hand, gradient disappearance and explosion always exist; although, technologies such as ReLU and Batch Normalization can be alleviated to some extent; on the other hand, with the deepening of the network, the accuracy reaches saturation first and then degenerates quickly [28]. A bee-line module containing identity mapping and residual mapping was proposed to solve this problem. Instead of learning the loss function, the bee-line module learns the residual function $F(x)$, as shown in Figure 6(a). After continuous training and optimization, the residual mapping will be pushed to 0, and the error is only left identity mapping. At this time, the network is always in the optimal state. Therefore, the network will not degrade with the increase of depth. Each block of ResNet50 is composed of some residual blocks. The structure of the residual block is shown in Figure 6(b).

DenseNet121 goes even deeper to 121 layers. In addition to the basic operations of input layer and output layer, the model has four dense blocks and transition layer between each dense block. The transition layer is mainly used for dimension reduction. Each dense block contains 5 layers, each of which is directly connected to all the preceding

TABLE 1: The framework of 5 DCNN models.

	AlexNet (8 layers)	VGG16 (16 layers)	InceptionV3 (22 layers)	ResNet50 (50 layers)	DenseNet121 (121 layers)
Input size	227×227	224×224	299×299	224×224	224×224
CONV block 1	$11 \times 11, 96$	$3 \times 3, 64$ $3 \times 3, 64$ Max. pool, 64	$3 \times 3, 32$ $3 \times 3, 32$ $3 \times 3, 64$ Max. pool, 64	$7 \times 7, 64$ Max. pool, 64	$7 \times 7, 64$ Max. pool, 64
CONV block 2	$5 \times 5, 256$	$3 \times 3, 128$ $3 \times 3, 128$ Max. pool, 128	$3 \times 3, 64$ $3 \times 3, 80$ $3 \times 3, 192$	$3 \times \begin{pmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{pmatrix}$	$6 \times \begin{pmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \end{pmatrix}$ Avg. pool, 128
CONV block 3	$3 \times 3, 384$	$3 \times 3, 256$ $3 \times 3, 256$ Max. pool, 256	Inception A, 288 Inception A, 288	$4 \times \begin{pmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{pmatrix}$	$12 \times \begin{pmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \end{pmatrix}$ Avg. pool, 256
CONV block 4	$3 \times 3, 384$	$3 \times 3, 512$ $3 \times 3, 512$ $3 \times 3, 512$ Max. pool, 512	Inception B, 768 Inception B, 768 Inception B, 768 Inception B, 768 Inception B, 768	$6 \times \begin{pmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{pmatrix}$	$24 \times \begin{pmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \end{pmatrix}$ Avg. pool, 512
CONV block 5	$3 \times 3, 256$	$3 \times 3, 512$ $3 \times 3, 512$ $3 \times 3, 512$ Max. pool, 512	Inception C, 1280 Inception C, 1280	$4 \times \begin{pmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{pmatrix}$	$16 \times \begin{pmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \end{pmatrix}$
Dense net	Fc, 4096 Fc, 4096 Fc, 1000	Fc, 4096 Fc, 4096 Fc, 1000	Avg. pool, 2048 Fc, 1000	Avg. pool, 2048 Fc, 1000	Avg. pool, 1024 Fc, 1000

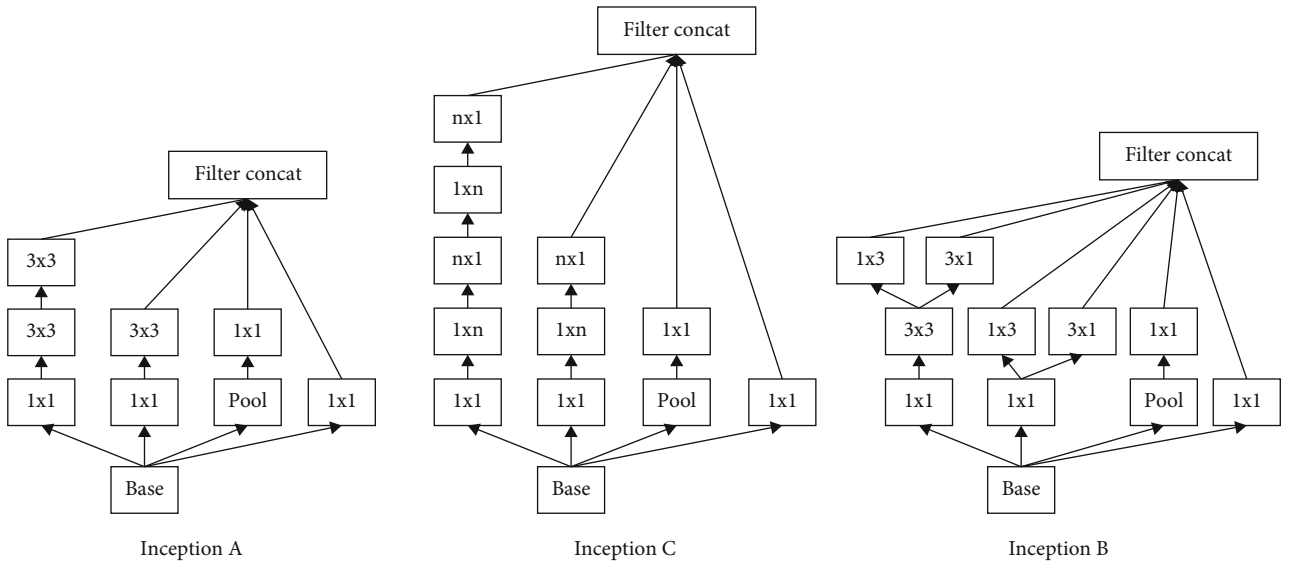


FIGURE 5: The inception module.

layers to achieve feature reutilization. Based on the design concept of dense block, the network becomes narrower and less parameter. The basic structure of dense block is shown in Figure 7. The connection mode of DenseNet121 is equivalent to that each layer directly connects input and loss, which makes the transfer of features and gradients more effective, and the network can also be deeper.

3. Transfer Learning Implementation

In the actual implementation process, researchers seldom train an entire DCNN from scratch, because it is relatively rare to have a dataset of sufficient size. What is commonly done is to reuse a pretrained model on a huge dataset for another task.

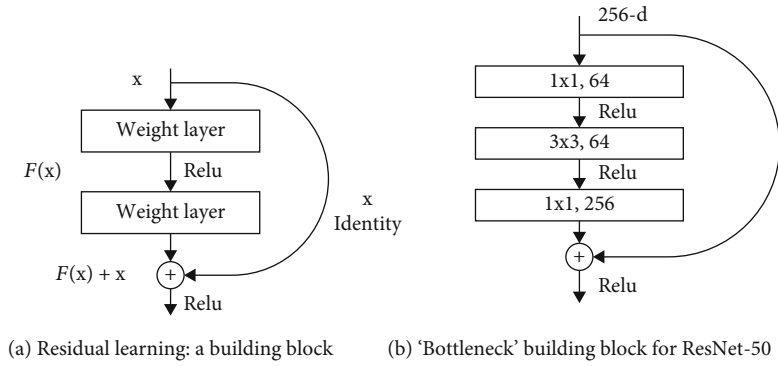


FIGURE 6: Residual block.

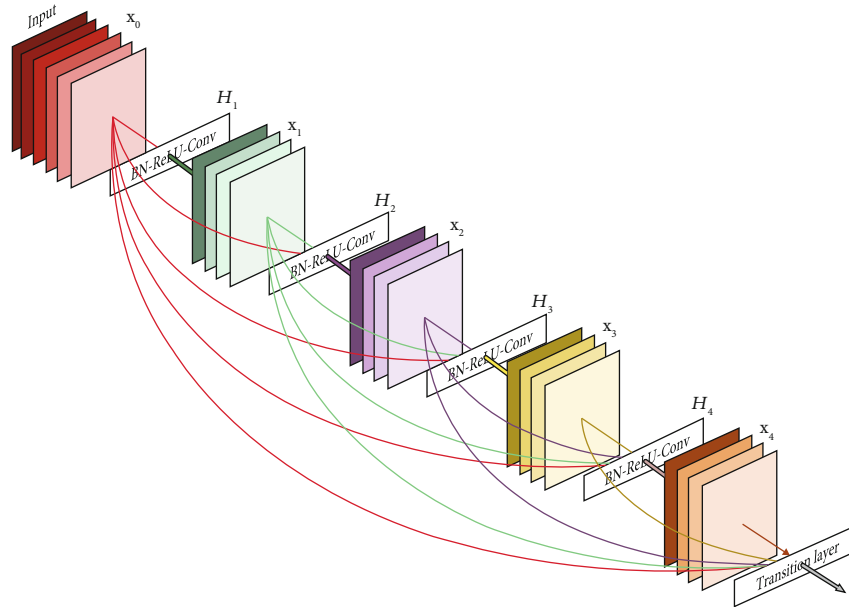


FIGURE 7: The basic structure of dense block.

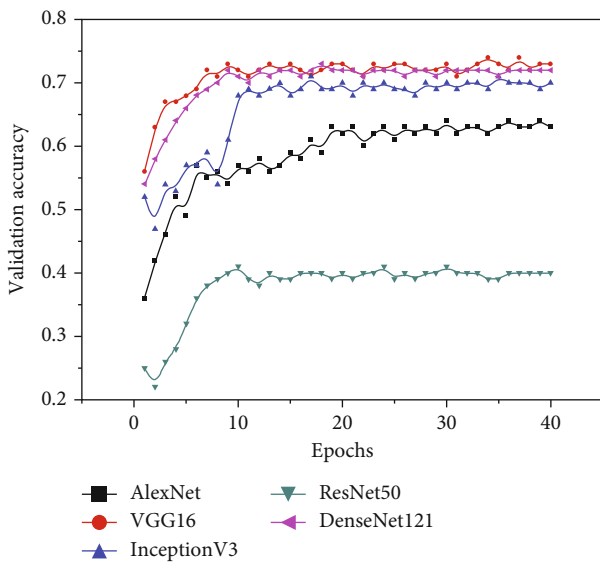


FIGURE 8: The validation accuracy curve of 5 DCNN models in the case of frozen base network.

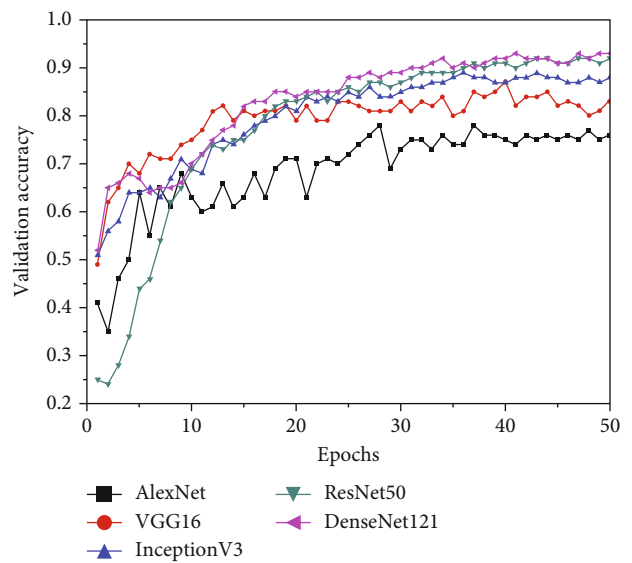


FIGURE 9: The validation accuracy curve of 5 DCNN models in the case of unfrozen the whole network.

TABLE 2: The contrast of experimental results of 5 DCNN models.

DCNN models	Fine-tuned layers	Time cost per epoch (s)	Validation accuracy (%)	Test accuracy (%)
AlexNet	The whole network	745	76.00	74.60
VGG16	Dense net	759	75.00	
	Block 5	760	81.00	
	Block 4	766	83.00	
	Block 3	759	85.00	
	Block 2	763	83.00	
	The whole network	759	84.00	83.99
InceptionV3	Dense net	770	70.00	
	Block 5	784	85.00	
	Block 4	791	88.00	
	Block 3	794	89.00	
	The whole network	798	88.00	82.19
ResNet50	Dense net	409	44.00	
	Block 5	420	89.00	
	Block 4	430	89.00	
	Block 3	437	91.00	
	The whole network	441	92.00	87.99
DenseNet121	Dense net	413	72.00	
	Block 5	437	86.00	
	Block 4	451	89.00	
	Block 3	486	93.00	
	The whole network	535	92.00	88.39

In the experiment, firstly, we build the above five DCNN models based on the TensorFlow [29], including the network structure and the training parameters and weights of the corresponding dataset and ran on 2 NVidia GeForce 1080 Titan GPU. Secondly, we changed the input layer and output layer of the pretrained model, the size of the input image was processed according to Table 1, and the output category was set to 7 according to the category of wear debris. Thirdly, we keep the base network and parameters unchanged and trained the added part. In training, we employed operations such as displacement, rotation, scaling, and flipping for the input image to expand the training dataset, so as to improve the generalization ability and robustness of the network. Finally, we verified the accuracy of the model on the validation set.

The validation accuracy curve of 5 DCNN models was shown in Figure 8. Obviously, the classification performance of DenseNet121 and VGG16 was better than that of the other three models. Resnet50 was the worst, and its accuracy remained around 40%. The convergence speed of AlexNet was the slowest, and it only converged around the 20th epochs.

The above experiments were carried out on the basis of freezing the base network and parameters, only changing the output layer nodes. The effect may be affected by the original training dataset, which cannot fully reflect the applicability of the model to the image of wear debris. Next, we unfroze each CONV blocks of the base network one by

one and fine-tuned the training parameters until the whole base network was completely unfrozen.

From Figure 9, one can see the validation accuracy curve of 5 DCNN models in the case of unfrozen the whole network. Obviously, DenseNet121 and ResNet50 achieved the best validation accuracy of more than 90%, but ResNet50 converged slower than DenseNet121. Especially in the initial stage of learning, the validation accuracy of ResNet50 had less than 30%, and the reason may be that the initial weight was not applicable to the new task, resulting in the increase of calculation cost. Although the convergence speed of AlexNet was the slowest, it needed to converge almost the 40th epoch, and the validation accuracy was also the lowest, but even more than 70%.

Table 2 shows the time cost per epoch in different blocks, the validation accuracy, and the test accuracy of 5 DCNN models. Because the number of layers of AlexNet was less and the cost of calculation was lower, the whole network was trained directly instead of fine-tuning its layers.

From the perspective of validation accuracy, with the increase of the number of unfrozen network blocks, the validation accuracy of all models was also increasing and tended to be stable. Generally, the validation accuracy of training on the whole network was the highest. From the perspective of test accuracy, the test accuracy of all models was slightly lower than the validation accuracy. DenseNet121 achieved the highest test accuracy of 88.39%. From the perspective of time cost per epoch, ResNet50 had the

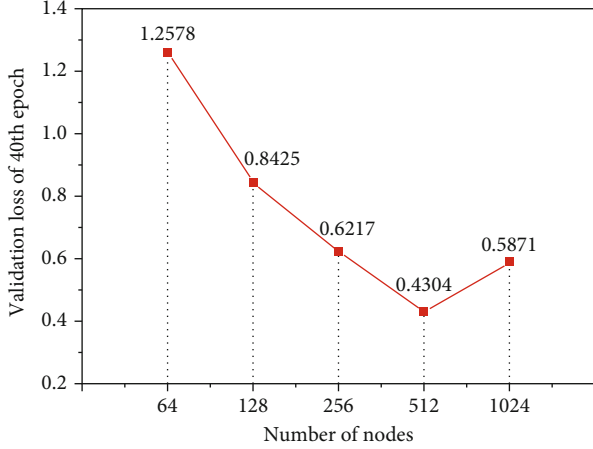


FIGURE 10: The validation loss of 40th epoch for the number of nodes in the penultimate FC layer.

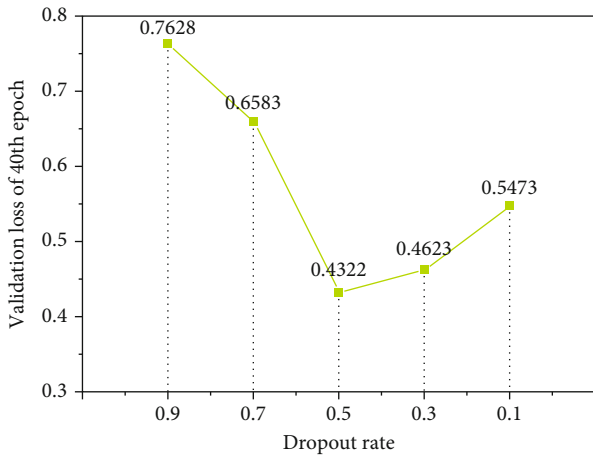


FIGURE 11: The validation loss of 40th epoch for the dropout rate.

highest efficiency, while DenseNet121 was slightly lower than it, and the time cost of training for the other three models was higher than them. This may be mainly due to the contribution of the residual block and dense block structures. But with the improvement of computing ability, the time cost of fine-tuning a DCNN model may become less important. For example, the time to fine-tune the whole InceptionV3 is about 60 minutes compared to only half the time for DenseNet121.

Through the above analysis, DenseNet121 was used as the basic structure for further optimization in this study.

4. Results and Discussion

Through transfer learning, we established a network framework, including CONV layers and FC layers. Then, we needed to adjust the network hyperparameters to make it suitable for the classification of wear debris images. In order to choose the optimal hyperparameters, we designed a series of comparative experiments. These experiments were based on the network structure of DenseNet121, which we trained

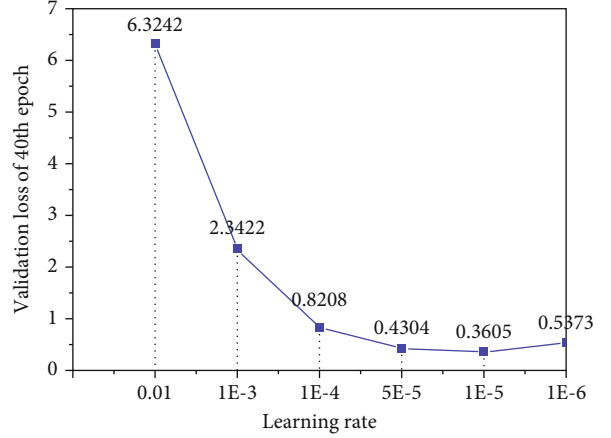


FIGURE 12: The validation loss of 40th epoch for the learning rate.

before. The optimization of one variable in the experiment was based on other variables remain unchanged.

Wear debris image recognition is a multiclassification problem; so, some properties of the model are determined. A Softmax function was used to convert the linear prediction value to the classification probability in the output layer. RMSprop was used as the method of stochastic gradient descent. The crossentropy loss function was chosen as the objective function.

$$S_i = \frac{e^i}{\sum_j e^j}, \quad (3)$$

$$s_{dw} = \beta s_{dw} + (1 - \beta) dW^2,$$

$$s_{db} = \beta s_{db} + (1 - \beta) db^2,$$

$$W = W - \alpha \frac{dW}{\sqrt{s_{dw} + \epsilon}}, \quad (4)$$

$$b = b - \alpha \frac{db}{\sqrt{s_{db} + \epsilon}},$$

$$H(P, Q) = E_{x \sim P}[-\log Q(x)], \quad (5)$$

where S_i is the probability score of the i -th element in an array containing j elements. s_{dw} and s_{db} are the gradient momentum of the loss function accumulated in the previous iteration, β is an exponent of the gradient accumulation, W and b represent weight and bias matrix, respectively, and ϵ is generally 10^{-8} . $H(P, Q)$ represents the crossentropy loss function of the probability distributions P and Q , while $E_{x \sim P}$ means that we use the probability distribution P to calculate the expectation.

Firstly, we optimized the number of nodes in the FC layers. The number of nodes in the last FC layer was determined, which was 7, which was the classification of wear debris images. The validation loss of the 40th epoch for the number of nodes in the penultimate FC layer was shown in Figure 10. The experimental results showed that the optimal number of nodes was selected to 512.

TABLE 3: Hyperparameters and test accuracy of FWDNet.

Type	Base network	Input size	Output nodes	Nodes of FC layer	Loss function	Dropout rate	Learning rate	Activation function	10-fold crossvalidation
Method/ value	DenseNet121	224, 224, 3	7	512	Softmax/ crossentropy	0.5	10^{-5}	ReLU	90.15%

Secondly, we optimized the dropout rate in FC layers. Dropout is an effective way to prevent overfitting [30]. It represents the proportion of randomly discarding several output features of the layer during training. The validation loss of the 40th epoch for the dropout rate was shown in Figure 11. The experimental results showed that 0.5 was the best choice for the dropout rate.

Finally, we optimized the learning rate. One suitable learning rate can enable the loss function to converge to the local minimum value at the appropriate time. When the learning rate was set too small, the convergence process became very slow. On the contrary, when the learning rate was set too large, the gradient may oscillate back and forth near the minimum value and may even fail to converge. The validation loss of the 40th epoch for the learning rate was shown in Figure 12. The experimental results showed that the learning rate was selected to 10^{-5} .

After the above experiments, we constructed a DCNN model for ferrography wear debris image recognition, which we named FWDNet. The final hyperparameters and test accuracy of the network model were shown in Table 3. In order to verify the fitting and generalization ability of the model, we used the method of 10-fold crossvalidation test [31]. During the crossvalidation, the wear debris dataset was randomly divided into ten subsamples. Among the ten subsamples, one subsample was reserved as the validation dataset, and the remaining nine subsamples were reserved as the training dataset. The crossvalidation process was repeated ten times, and each subsample was used as validation dataset once. Finally, the average accuracy of 10 validation resulted in evaluating the model. From Table 3, one can see that the accuracy of 10-fold crossvalidation was 90.15%.

Therefore, FWDNet is an excellent DCNN model for image classification of ferrography wear debris and can be used for real-time online detection of wear debris.

5. Conclusion

In this study, DenseNet121 is used as the base network to construct a DCNN model (FWDNet) using the transfer learning method for intelligent classification of ferrography wear debris images. FWDNet contains up to 121 weight layers, and the efficiency of feature extraction and gradient descent is greatly improved by adopting the dense block design concept. FWDNet obtained an accuracy of 90.15% through a 10-fold crossvalidation test.

The intelligent identification and classification of ferrography wear debris images can reflect the wear state and wear mechanism, which can provide the basis for condition maintenance and fault diagnosis of equipment. In this study, a dataset containing seven kinds of wear debris, including

severe sliding, lamellar fatigue, spalling fatigue, cutting, normal, spherical, and nonferrous metal debris, was established through a series of tribological experiments. In order to construct a real-time online wear debris intelligent recognition system, expanding the dataset should be a long-term task.

Instead of manually designing and selecting the features, FWDNet can automatically learn the features through multiple processing layers composed of nonlinear transformations to realize end-to-end processing, that is, from the original image to the identification of different kinds of wear debris, avoiding the accumulation and transmission of errors caused by numerous steps applied in a traditional linear process, and improving the efficiency and accuracy of wear debris analysis.

In future research, we will pay more attention to pixel-level segmentation algorithm to find a better ferrography wear debris recognition algorithm.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This work was supported by the Key Research and Development Plan of Shandong Province (2018GGX105002, 2019JZZY020712).

References

- [1] S. K. Yang, "A condition-based failure-prediction and processing-scheme for preventive maintenance," *IEEE Transactions on Reliability*, vol. 52, no. 3, pp. 373–383, 2003.
- [2] S. Raadnui, "Wear particle analysis-utilization of quantitative computer image analysis: a review," *Tribology International*, vol. 38, no. 10, pp. 871–878, 2005.
- [3] W. Li and H. Pham, "An inspection-maintenance model for systems with multiple competing processes," *IEEE Transactions on Reliability*, vol. 54, no. 2, pp. 318–327, 2005.
- [4] C. Kowandy, C. Richard, Y. M. Chen, and J. J. Tessier, "Correlation between the tribological behaviour and wear particle morphology— case of grey cast iron 250 versus Graphite and PTFE," *Wear*, vol. 262, no. 7-8, pp. 996–1006, 2007.

- [5] R. K. Upadhyay, "Microscopic technique to determine various wear modes of used engine oil," *Journal of Microscopy and Ultrastructure*, vol. 1, no. 3, pp. 111–114, 2013.
- [6] S. Ebersbach, Z. Peng, and N. Kessissoglou, "The investigation of the condition and faults of a spur gearbox using vibration and wear debris analysis techniques," *Wear*, vol. 260, no. 1-2, pp. 16–24, 2006.
- [7] J. Wang, L. Zhang, F. Lu, and X. Wang, "The segmentation of wear particles in ferrograph images based on an improved ant colony algorithm," *Wear*, vol. 311, no. 1-2, pp. 123–129, 2014.
- [8] B. J. Roylance, "Ferrography-then and now," *Tribology International*, vol. 38, no. 10, pp. 857–862, 2005.
- [9] T. P. Sperring and T. Nowell, "SYCLOPS—a qualitative debris classification system developed for RAF early failure detection centres," *Tribology International*, vol. 38, no. 10, pp. 898–903, 2005.
- [10] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.
- [11] M. Kumar, P. S. Mukherjee, and N. M. Misra, "Advancement and current status of wear debris analysis for machine condition monitoring: a review," *Industrial Lubrication and Tribology*, vol. 65, no. 1, pp. 3–11, 2013.
- [12] G. P. Stachowiak, P. Podsiadlo, and G. Stachowiak, "A comparison of texture feature extraction methods for machine condition monitoring and failure analysis," *Tribology Letters*, vol. 20, no. 2, pp. 133–147, 2005.
- [13] G. Stachowiak and P. Podsiadlo, "Towards the development of an automated wear particle classification system," *Tribology International*, vol. 39, no. 12, pp. 1615–1623, 2006.
- [14] G. P. Stachowiak, G. Stachowiak, and P. Podsiadlo, "Automated classification of wear particles based on their surface texture and shape features," *Tribology International*, vol. 41, no. 1, pp. 34–43, 2008.
- [15] T. Wu, H. Wu, Y. Du, N. M. Kwok, and Z. Peng, "Imaged wear debris separation for on-line monitoring using gray level and integrated morphological features," *Wear*, vol. 316, no. 1-2, pp. 19–29, 2014.
- [16] S. Wang, T. Wu, K. Wang, and T. Sarkodie-Gyan, "Ferrograph analysis with improved particle segmentation and classification methods," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 2, 2020.
- [17] Y. Peng, J. Cai, T. Wu, G. Cao, N. Kwok, and Z. Peng, "WP-DRnet: A novel wear particle detection and recognition network for automatic ferrograph image analysis," *Tribology International*, vol. 151, p. 106379, 2020.
- [18] Y. Lecun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [19] L. Deng, J. Li, J. T. Huang et al., "Recent advances in deep learning for speech research at Microsoft," in *International conference on acoustics, speech, and signal processing*, Vancouver, BC, Canada, 2013.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Image net classification with deep convolutional neural networks," *Advances In Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [21] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. P. Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," in *European Conference on Computer Vision*, Berlin, Heidelberg, 2008Springer.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on image net classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015.
- [24] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: a tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.
- [25] K. Simonyan and A. Zisserman, "Very DEEP convolutional networks for large-scale image recognition," 2014, <http://arxiv.org/abs/1409.1556>.
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.
- [27] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Hawaii, America, 2017.
- [28] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference On Machine Learning PMLR*, Lille, France, 2015.
- [29] M. Abadi, A. Agarwal, P. Barham et al., "TensorFlow: large-scale machine learning on heterogeneous distributed systems," 2015, <http://arxiv.org/abs/1603.04467>.
- [30] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [31] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of K-fold cross-validation," *Journal of Machine Learning Research*, vol. 5, pp. 1089–1105, 2004.