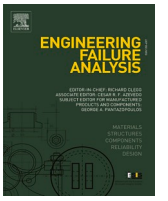
Engineering Failure Analysis 122 (2021) 105268

Contents lists available at ScienceDirect

Engineering Failure Analysis

journal homepage: www.elsevier.com/locate/engfailanal



Semantic segmentation of ferrography images for automatic wear particle analysis

Xinliang Liu [1], Jingqiu Wang [1], Kang Sun, Liang Cheng, Ming Wu, Xiaolei Wang [*]

National Key Laboratory of Science and Technology on Helicopter Transmission, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ARTICLE INFO

Keywords:
Wear particle analysis
Ferrography
Convolutional neural network
Semantic segmentation

ABSTRACT

Automatic wear particle detection and classification has remained a high priority research area for wear condition monitoring and failure analysis. In this study, a deep convolutional neural network (DCNN) with three modules, namely, an encoder, atrous spatial pyramid pooling (ASPP), and a decoder, is constructed. Instead of using handcrafted features, the DCNN can automatically learn features through a layer-wise representation and realize semantic segmentation, i.e., segmentation and identification concurrently, of five types of wear particles in ferrograph images using end-to-end processing. Experimental results show that the DCNN achieves 82.5% accuracy. This proposed method unifies the segmentation, classification, and edge location of the wear particles into a single model, avoids the accumulation and transmission of errors caused by numerous steps applied in a traditional linear process, and improves the efficiency and accuracy of the wear particle analysis.

1. Introduction

Wear particles in the lubricant used in machine equipment contain rich information that can be used to evaluate the wear condition and determine the wear mechanisms occurring [1].

Different types of wear particles, including normal rubbing, spherical, cutting, fatigue, and oxides particles etc., are generated through the wear process by different wear mechanisms. For example, the spherical particles indicate there might be cracks on the components of rolling bearing, the cutting particles are the evidence that abrasive wear happened in machine equipment, and their amount shows the severity of the wear. Ferrography is a methodology which separates wear particles from lubrication oil by a high gradient magnetic field, deposits them on a glass slide, and provides a microscopic examination and analysis qualitatively and quantitatively on wear particles. So that the wear mechanisms and wear state (normal, severe, or failure) are determined according to corresponding criteria of the equipment. Ferrography has been successfully used to monitor the conditions of aircraft engines, gearboxes, and transmissions since 1970s [2,3].

The automatic analysis of a ferrograph image remains a challenge despite numerous studies in this area. A computer-aided ferrograph image analysis process usually employs a unidirectional linear procedure, i.e., image preprocessing → wear particle segmentation → feature extraction → feature simplification → wear particle classification. There are two problems with this type of procedure. First, the uncertainties and errors of the wear particle segmentation will have an inevitable impact on the subsequent

* Corresponding author.
E-mail address: wxl@nuaa.edu.cn (X. Wang).
[1] These authors contributed equally in this work.

https://doi.org/10.1016/j.engfailanal.2021.105268
Received 31 May 2020; Received in revised form 9 January 2021; Accepted 26 January 2021
Available online 2 February 2021

analysis. Image segmentation algorithms such as Otsu [4] and watershed [5] have been used to separate wear particles in ferrograph images [6,7]. However, because some wear particles are deposited as chains, some being adjacent or even overlapping with each other, there is no unique or universal segmentation algorithms can be applied to various images. Second, all features such as the shape [8–15], color [16,17], and texture [18] of the wear particles are designed according to specific recognition tasks. They are called handcrafted or manual design features, which may have information redundancy or lead to the neglection of important features. Thus, the error accumulation between steps, and the use of handcrafted feature-based methods, limit the accuracy and efficiency of wear particle analysis.

Some recognition methods, including support vector machine (SVM) [19] and artificial neural network (ANN) [20], have been applied in the classification of wear particles [21]. However, shallow neural network models usually require input features, and thus the effectiveness of these methods depends on manual feature extraction to a certain extent. The linear flow of computer-aided ferrograph image analysis makes it difficult to tune the algorithms as a whole, which limits their application in automatic wear particle analysis.

When experienced analysts observed a ferrograph image, they would not be bothered by wear particle segmentation. Although wear particles have blurred, or partially overlapping edges, they can be immediately identified by a glimpse, mainly because the human brain can process a whole image directly and recognize all wear particles after pre-training and learning. Researchers can make use of comprehensive information of wear particle edges and surfaces, which make the segmentation and recognition a mutual

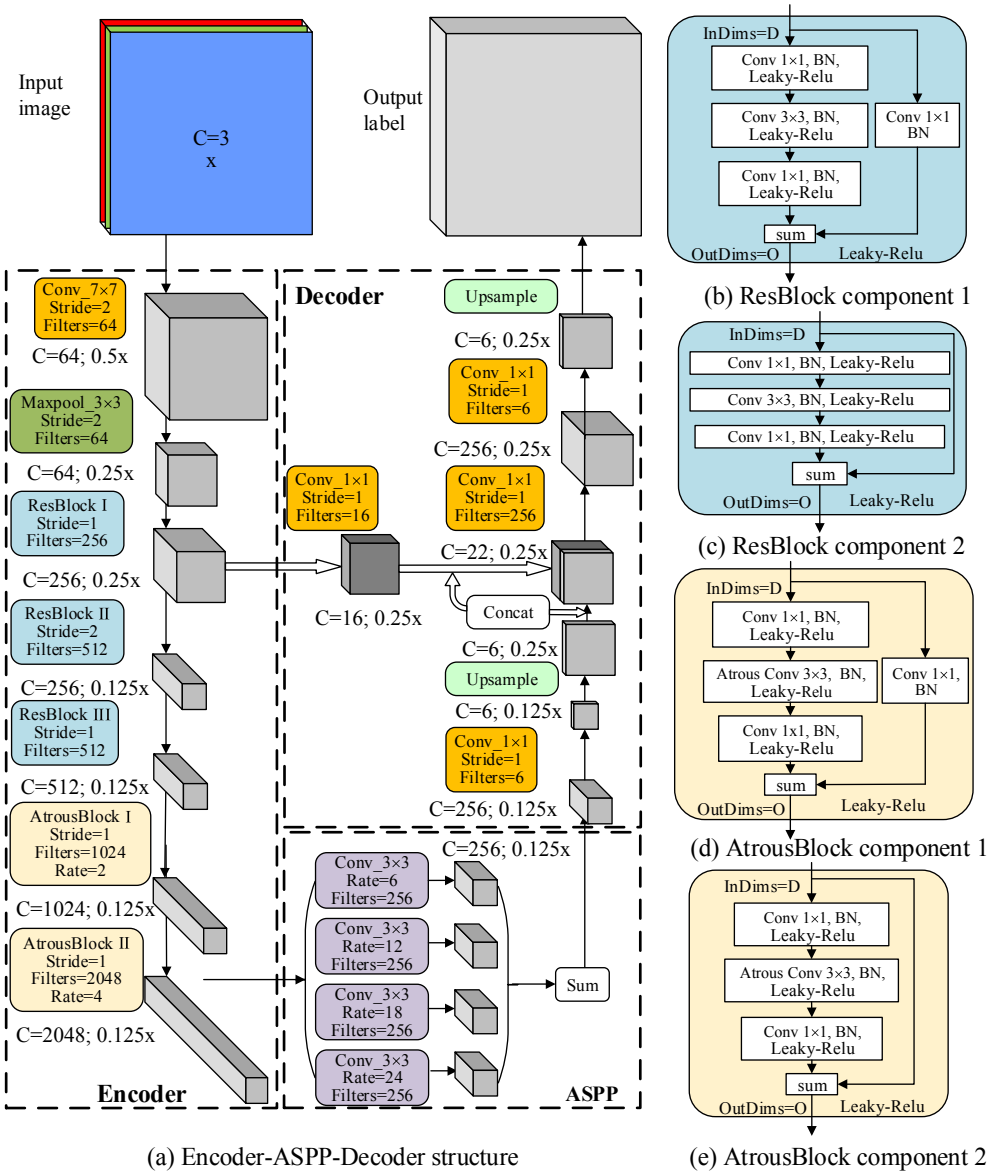


Fig. 1. Architecture of the proposed DCNN.

interaction and unified process.

In 2006, Hinton proposed the theory of Deep Learning (DL) [22]. Convolutional neural network (CNN) is a popular research area in the field of DL. It uses convolution operations to extract features of images and inherently take the significant information, such as surface textures, shapes, colors, into consideration. CNN has certain advantages such as the self-learning and transmission of features, as well as an end-to-end processing, which significantly improves the accuracy of the classification or recognition over traditional methods.

There are many applications of CNNs in image classification [23,24] and object detection [25,26]. Recently, Wang and Peng used CNN models [27–30] for ferrograph image classification and wear particle detection. This shows the great potential of CNN models for wear particle analysis.

Image classification CNNs are typically used to analyze an entire image and judge its category. Object detection CNNs are used to find a bounding box of the objects in an image. Conceptually, semantic segmentation is also called pixel-level classification, and associates a label or category with every pixel in an image [31]. Therefore, the semantic segmentation of wear particles is equivalent to the segmentation and classification of wear particles in ferrograph images simultaneously by a CNN model. Comparatively, for the wear particle analysis, semantic segmentation might obtain better results than image classification followed by object detection because it combines these two steps together. By implementing a semantic segmentation, the wear particle analysis process can be simplified, and errors caused by an inaccurate segmentation of the wear particles can be avoided, which is conducive to the overall optimization of the algorithm.

In this paper, a deep convolutional neural network (DCNN) model for the semantic segmentation of wear particles is proposed. The DCNN model provides an end-to-end processing for automatic wear particle recognition, and realizes the semantic segmentation of five types of wear particles, namely, chain, cutting, block, spherical, and oxide particles, so that qualitative and quantitative analysis of wear particles can be conducted then.

2. Proposed DCNN for semantic segmentation of wear particles

Compared with image classification, the semantic segmentation of an image is a more challenging task, and therefore, in addition to basic operations including convolution, pooling, and activation, the DCNN for semantic segmentation requires some other operations including up-sampling, atrous convolution [32], feature fusion, a residual block [33], and batch normalization (BN) [34], please refer to the corresponding paper for details.

2.1. Architecture of DCNN

Semantic segmentation can be considered a pixel-level classification task aiming at giving each pixel a class label. Differing from image-level classification, which applies only a forward process to extract features of an entire image and determine its category, semantic segmentation requires not only a forward process (an encoder process) but also a reverse process (a decoder process) to up-sample back to its original size and fulfill the labeling of each pixel. Therefore, the proposed DCNN is composed of three modules, namely, an encoder, ASPP, and a decoder, as shown in Fig. 1(a). In what followings, the three modules are described in detail.

2.2. Encoder module

The input of the encoder module is the original color image. The encoder module acts as a feature extractor because it extracts high-level features after a series of convolution, activation, and pooling operations.

As shown in Fig. 1(a), the encoder module has a convolutional layer, a max-pooling layer, three ResBlocks, and two AtrousBlocks.

In a conventional CNN, several pooling layers are used to reduce redundant information and remain invariant to spatial transformations [32], which results in the last feature maps being small and the occurrence of incorrect classification problems when the last feature map is resampled back to the original size. To solve this problem, in the proposed DCNN, ResNet [33] is taken as the extractor backbone in the encoder module, and the ResBlock and AtrousBlock components are designed and constructed, as shown in Fig. 1(b)–(e).

ResBlock I consists of ResBlock component I (Fig. 1(b)) followed by two ResBlock components II (Fig. 1(c)). ResBlock II consists of one ResBlock component II. ResBlock III consists of three ResBlock components II. AtrousBlock I has one AtrousBlock component I (Fig. 1(d)) followed by 22 AtrousBlock components II (Fig. 1(e)). AtrousBlock II has three AtrousBlock components II.

Both ResBlock components I and II include three convolution layers, which adopt a BN and leaky-ReLU activation [35]. As the main difference between them, component I has a 1×1 convolution layer for a shortcut connection, which is used to change the channel number. As shown in Fig. 1(a), ResBlock I has 64 and 256 input and output channels, respectively, and the 1×1 convolution layer confirms that the shortcut connection obtains the same number of channels as the output.

The AtrousBlock component is similar to the ResBlock component, but adopts atrous convolution in the second layer, as shown in Fig. 1(d) and (e).

Two improvements have been made to the encoder module in the proposed DCNN. First, a leaky-ReLU activation function has been adopted because it ensures that all nodes are activated. Second, an atrous convolution is applied to increase the receptive field of the convolution kernel. The original ResNet uses five pooling layers to make the feature maps small and invariant to spatial transformations, whereas our DCNN expects larger feature maps and makes a pixel-level prediction. Therefore, atrous convolution layers are applied to replace the last two convolutional and pooling layers, which increases the resolution by 4 compared with the original

extractor.

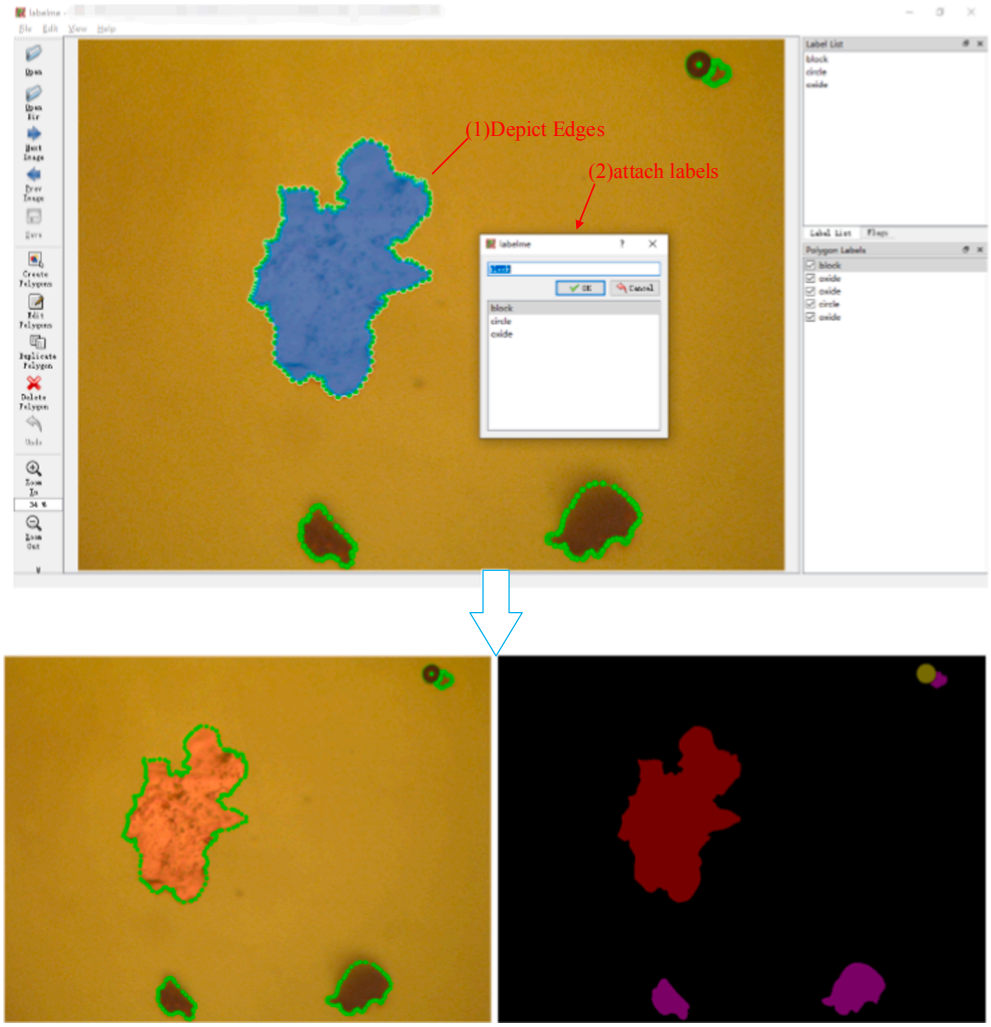
The output of the encoder module is low-resolution feature maps. It is necessary to match such feature maps for pixel-wise predictions in a semantic segmentation, hence, the other two modules, the ASPP and decoder module are designed in the proposed DCNN.

2.3. ASPP module

The ASPP module exploits multi-scale features by employing multiple parallel atrous convolutions with different atrous rates, which takes its idea from spatial pyramid pooling [36]. As shown in Fig. 1(a), the ASPP module contains four parallel atrous convolutions, the atrous rates of which are 6, 8, 18, and 24, respectively. Finally, four outputs from four atrous convolutions are summed to obtain the final feature map.

The ASPP module is important for obtaining a better understand of the details of the entire image, because it uses different atrous rates to obtain different receptive fields, and then summarizes the feature information. For example, wear particles of different sizes are better segmented and classified using the proposed DCNN, as compared with models without an ASPP module, which often mislabels large particles into several small individual areas and misses some of the small particles.

This indicates that the encoder module in a DCNN is effective at extracting the fine features to distinguish between different categories, and the ASPP module is applied to obtain a feature representation at different scales.



(a) Alabeled image (b) GT:black, background; red, block particle; purple, oxide particle; yellow, spherical particle

Fig. 2. Examples from ferrograph image dataset.

2.4. Decoder module

The decoder module is used to restore the last feature map to its original image size while labeling each pixel. The feature maps output from earlier layers may be accurate in terms of localization, but they will not capture the semantics. To take advantage of both, we borrowed ideas from a fully convolutional network (FCN) [31] and U-Net[37] to fuse low-level features with high-level features. We first apply a convolution operation to the feature map output from the ASPP module, and then double its size using up-sampling, and concatenate it with the feature map of ResBlock I to fuse the features, as shown in Fig. 1(a). Next, we apply two convolution operations to integrate the information. Finally, an up-sampling operation is applied to restore the feature map to its original size using a bilinear interpolation.

3. Experiments

3.1. Dataset

The proposed DCNN is trained and validated on a ferrograph image dataset, all images of which were taken from mining and petrochemical equipment. The original size of each image is 768 × 576.

The training of a DCNN model normally requires numerous labelled images to achieve a better performance. However, owing to the difficulty in creating per-pixel labeled segmentation datasets, their scale is not as large as the size of the classification datasets. For this reason, transfer learning (TL) [38], particularly fine-tuning from a pre-trained classification network, is a common approach in a segmentation network [39]. It has also been proved that transferring features even from distant tasks can be better than using a random initialization [40]. Therefore, in our experiments, the encoder module is based on a pre-trained ResNet, i.e., parameters of convolution and BN layers in ResBlocks (Fig. 1(a)) are transferred from the corresponding ResNet blocks, and AtrousBlocks also dilates convolution kernels of ResNet blocks. Then the ASPP and decoder modules are added to it, and is fine-tuned on the ferrograph image dataset using TL. Using TL, in our ferrograph image dataset, 690 are labelled images in the training set, 17 are in the validation set, and 25 are in the test set. Each image contains a certain number and different types of wear particles

In the ferrograph image dataset, wear particles are divided into five classes: chain, sphere, cutting, block, and oxide particles. All obvious wear particles in each image are carefully labelled manually using Labelme.

Fig. 2 shows a labeled image example and its ground truth (GT) image.

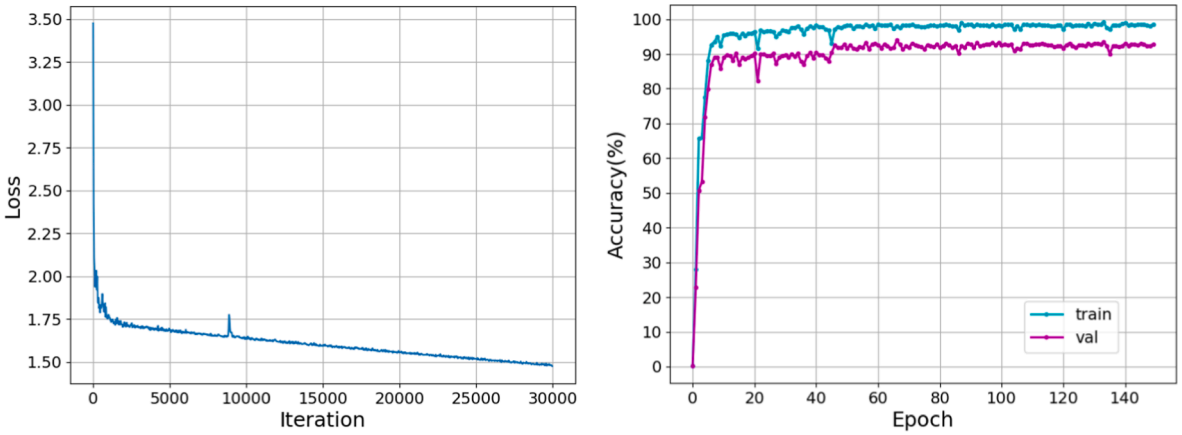
3.2. Experiment settings

The proposed DCNN model is implemented based on the TensorFlow [41] framework using a PC platform with an i7-8700 k CPU, 16 GB of RAM, and an RTX 2070-8G GPU.

During the training process, the hyperparameters are set as follows: The BN parameter is fixed, and the batch size is set to 1 in the pre-trained model. The learning rates are set to 0.0005 for the encoder module and 0.001 for the other modules, and an exponential decay is applied to the learning rate. Pixel-wise cross-entropy loss is adopted in the model, the formula is as follows:

$$Loss = -\sum_{i=1}^{M\times N} y^{(i)} \cdot log\widehat{y}^{(i)} \tag{1}$$

where M and N are the length and width of image respectively, y is the ground truth of each pixel, $\widehat{y}$is the predicted value of each pixel.



(a) Loss curve during training process (b) Validation result every 200 steps

Fig. 3. Training curve of DCNN.

A momentum optimizer with a momentum parameter of 0.9 is used to optimize the pixel-wise cross-entropy loss, and L2 regularization with a weight decay parameter of 0.0005 is applied to prevent an overfitting and obtain better results.

Fig. 3(a) shows the loss curve, which indicates that the DCNN model is convergent. Fig. 3(b) shows the accuracy of the training and validation. Both increase rapidly at the beginning and remain stable at a high percentage. This shows that the DCNN does not suffer from overfitting or underfitting problems.

3.3. Experimental results

During the experiments, we mainly adopt three evaluation metrics, namely, the pixel accuracy, mIOU, and speed.

The pixel accuracy [31] is traditionally described as follows:

$$\mathrm{Acc} = N_c / N_{all} \tag{2}$$

where N_c is the number of correctly predicted pixels and N_{all} is the number of all pixels in an image. However, the pixels of the background normally occupy a large proportion in a ferrograph image, and it is difficult to distinguish good segmentation results from poor results using the above formula. Therefore, another formula is applied in this study to show a more obvious difference:

$$\mathrm{Acc}_w = \left| (N_{pc} - N_{binc}) / N_p \right| \tag{3}$$

where N_{pc} is the number of wear particle pixels that are correctly predicted, N_{binc} is the number of background pixels that are incorrectly predicted, and N_p is the total number of wear particle pixels in the GT image.

The mean intersection over union (mIOU) is as follows:

$$\mathrm{mIOU} = \frac{1}{k+1} \sum_{i=0}^{k} \frac{N_{ii}}{N_{GT}^i + N_{pred}^i - N_{ii}} \tag{4}$$

where k is the number of classes, N_{ii} is the number of pixels of class i predicted to belong to class i, N_{GT}^i is the number of pixels belonging to class i in the GT image, and N_{pred}^i is the number of prediction pixels belonging to class i.

The speed is the processing time of each image.

Experiments were conducted on the test set of the ferrograph image dataset, some image examples of which are shown in Fig. 4. The test images contain different types of wear particles, including chain, sphere, block, cutting, and oxide particles. The corresponding semantic segmentation results are shown in Fig. 5. It can be seen that the DCNN performs well in general and can semantically segment wear particles with different sizes, colors, and categories. For example, (1), (6), and (15) in Fig. 5 show the results of semantic segmentation of an image with one type of particle. In addition, (3) (7), (8), (13), and (16) in Fig. 5 show the image results for multiple

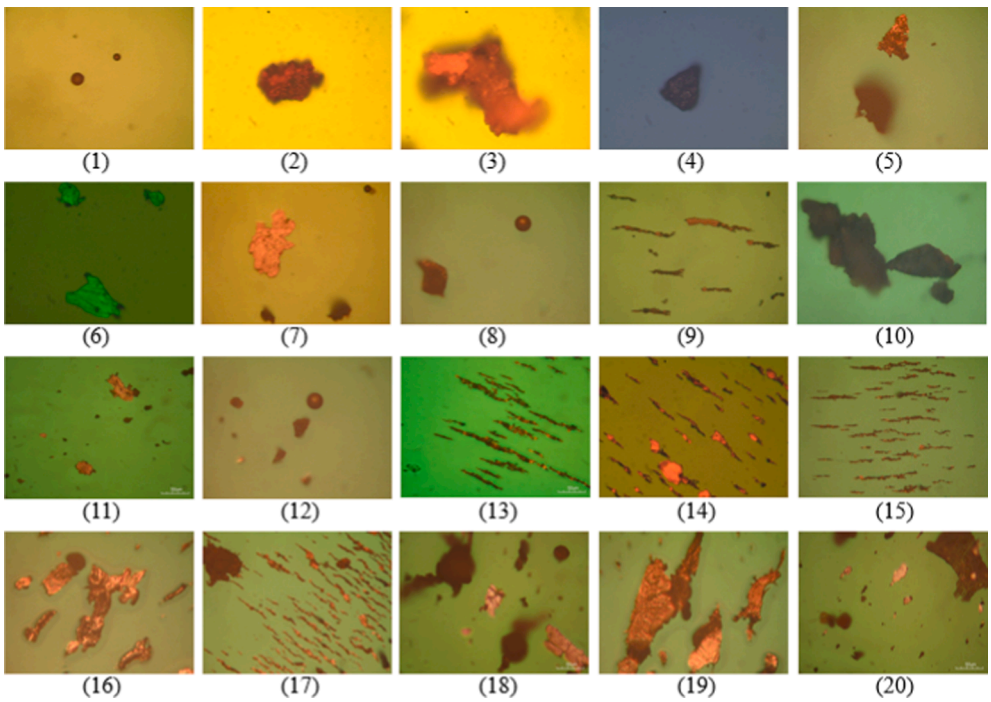


Fig. 4. Original images in test set of ferrograph image dataset.

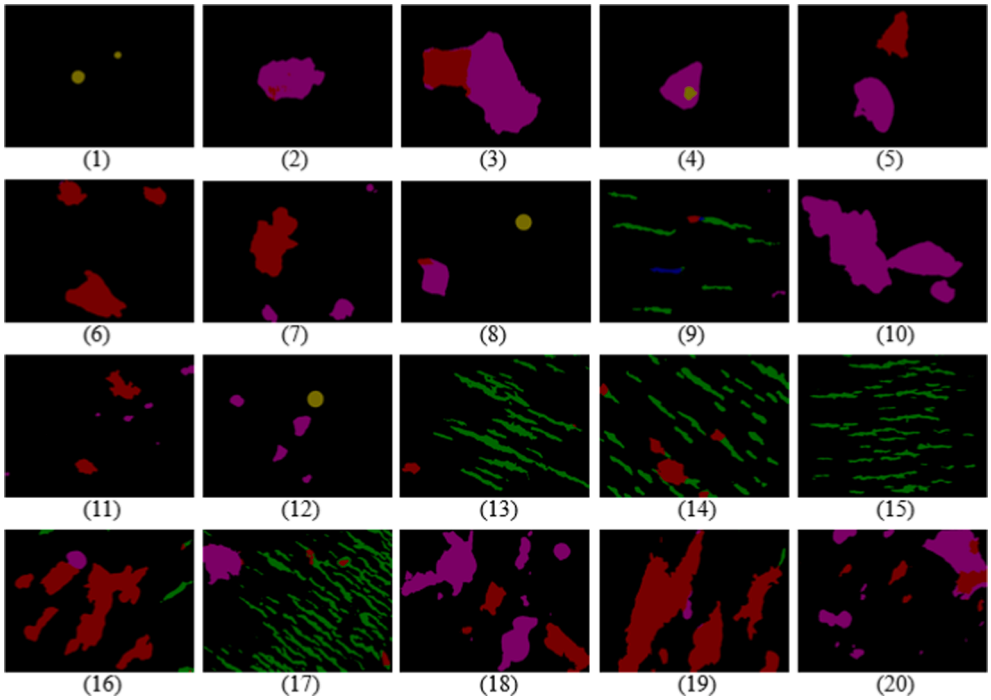


Fig. 5. Results of semantic segmentation of DCNN: black, background; red, block particle; green, chain particle; yellow, spherical particle; blue, cutting particle; purple, oxide particle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

types of wear particles, and (2), (4) (10), (16), and (19) show the results of images with large particles. Moreover, when some wear particles are blurred, the DCNN model can also obtain good results, as shown in (5) of Fig. 5.

On the test set, the accuracy of the DCNN is 82.50%, the mIOU is 84.66%, and the speed is 0.16 s per image, as shown in Table 1.

Because we used leaky-ReLU activation function in the encoder module and ASPP module in the proposed DCNN model, we compared the performance of DCNN models with different activation function and structure, the comparison results are shown in Table 1. For the proposed DCNN model, although the speed is slightly reduced due to the use leaky-ReLU and ASPP module, the accuracy of semantic segmentation results is improved to a certain extent. The results suggest that leaky-ReLU is important in extracting features of wear particles and the ASPP module is able to integrate those features to further improve the accuracy of semantic segmentation.

We also compared the performance of the proposed DCNN with some traditional ferrograph image analysis methods and current semantic segmentation models using the test set of the ferrograph image dataset.

Traditional ferrograph image analysis methods include five steps (described in Section 1), in which the segmentation and classification of wear particles are the two main steps, and thus we carried out the comparison experiments on them separately.

Four segmentation algorithms, namely, K-means, Otsu, region growing, and watershed methods, are used to segment a ferrograph image into wear particles (known as the foreground) and the background. In this case, the segmentation accuracy and mIOU of each method are shown in Table 2. The K-means method achieves an accuracy of 75.10% and an mIOU of 85.87%. However, it is necessary to provide the number of classifications in advance, and the initial cluster centers of K-means may affect the segmentation results. The Otsu method is an automatic segmentation algorithm, achieving an accuracy of 68.55% and an mIOU of 82.61%. The Otsu algorithm is sensitive to noise, and when the distribution of the background brightness is not uniform, the segmentation results are unsatisfactory. The region growing method achieves an accuracy of 58.20% and an mIOU of 77.73%, whereas the watershed method achieves an accuracy of 73.42% and an mIOU of 81.60%. These two methods require tuning of the parameters, and both are easily affected by noise, contrast, and the texture of the wear particles. It can be seen that the proposed DCNN achieves an accuracy of 87.35% and an

Table 1
Performance of the DCNN models.

	DCNN encoder(ReLU) + ASPP + decoder	DCNN encoder(leaky-ReLU) + decoder	DCNN encoder(leaky-ReLU) + ASPP + decoder
Acc_w (%)	63.99	79.70	82.50
mIOU (%)	73.59	82.22	84.66
Speed (s)	0.163	0.143	0.169

Table 2
Comparison of five segmentation methods.

Method	K-means	Otsu	Region Growing	Watershed	DCNN
Acc_w (%)	75.10	68.55	58.20	73.42	87.35
mIOU	85.87	82.61	77.73	81.60	93.29

mIOU of 93.29%, which is the best performance among the five methods applied.

In addition, we also compared the DCNN with SVM, which is one of the most frequently used classification methods. The inputs of the SVM are the histogram of oriented gradients (HOG) feature [42], the scale-invariant feature transform (SIFT) feature [43], and the image itself, the results of which are shown in Table 3. It should be noted that instead of using the original image, we used each wear particle as a single image to obtain the classification results for comparison, as shown in Fig. 6. Table 3 shows the accuracy of each method, of which the SIFT + SVM method achieves an accuracy of 61.05%, the HOG + SVM and image + SVM methods obtained a similar performance of 79.99% and 78.49%, respectively. However, these classification methods cannot perform well when there are numerous types of wear particles in a ferrograph image or when part of the wear particles are blurred or overlapped.

A traditional ferrograph image analysis method contains five steps through a linear process, and the accuracy of the segmentation will affect the performance of the wear particle classification. Differing from traditional methods, the proposed DCNN has an end-to-end processing, and can achieve a relatively high accuracy in the semantic segmentation of wear particles.

The DCNN is also compared with two state-of-the-art semantic segmentation CNNs, namely, FCN and DeepLab v2, the results of which are shown in Table 4. All three models are trained on the same ferrograph image dataset. It can be seen that although the DCNN is slightly slower than the other two models, it performs well in terms of accuracy and the mIOU. The FCN model achieves an accuracy of 57.55% and an mIOU of 68.70%, and the DeepLab v2 model achieves an accuracy of 56.23% and an mIOU of 72.55%, whereas the proposed DCNN achieves an accuracy of 82.50% and an mIOU of 84.66%.

Fig. 7 shows three different ferrograph images with chain, cutting, and block particles, respectively. The first through the last rows are the original images, GT images, and the semantic segmentation results of FCN, DeepLab v2, and DCNN, respectively.

FCN is a typical encoder-decoder model, and does not apply an atrous convolution or an ASPP module. As shown in Fig. 7, the segmentation results of FCN show some mistakes; for example, the chain particles are improperly labeled as block particles, and a large block particle is incorrectly divided into several small particles.

The segmentation results of DeepLab v2 and the proposed DCNN are better, as shown in Fig. 7. We believe this is because both apply atrous convolution and an ASPP module, which play an important role in the classification of wear particles. However, the DeepLab v2 model does not have a decoder module, and it directly restores the last feature map to the original image size using interpolation and does not fuse the low-level feature information. As shown in the inset of Fig. 8(a), the segmentation results of DeepLab v2, particularly the edges of the wear particles, are not accurately located as compared with the GT image. Because the proposed DCNN uses a decoder module, it can fuse different level features, achieving more accurate results, as shown in Fig. 8(b).

After the semantic segmentation of ferrograph images, the information of wear particle concentration can be obtained directly, which is helpful for condition monitoring and failure diagnosis. Fig. 9(a) is an example of ferrograph image with some particles, and Fig. 9(b) shows the semantic segmentation result. Fig. 9(c) is the concentration of different types of wear particles, which indicates there were many large block and oxide particles, and severe adhesion or fatigue wear might happen. But if it was near failure or not depends on the criteria of the equipment.

4. Conclusion

The classification of wear particles provides important clues for the identification of the wear condition and wear mechanism. Semantic segmentation is a pixel-level classification, which associates each wear particle in an image with a category. In this study, a DCNN model was constructed to achieve a semantic segmentation of wear particles of multiple types in ferrograph images.

The proposed DCNN has a cascade of three modules, namely, an encoder, ASPP, and a decoder. The encoder and decoder modules are based on the residual structure, and by designing a ResBlock component and an AtrousBlock component that adopt Leaky-ReLU and an atrous convolution, a DCNN for the semantic segmentation of wear particles can be trained. Instead of manually designing and selecting the features, the proposed DCNN can automatically learn the features through a layer-wise representation and realize a semantic segmentation of the five types of wear particles in the ferrograph images. The results of semantic segmentation can be directly used for the output of concentration information, which is conducive to the subsequent wear particle analysis. The experimental results show that the average accuracy of the proposed DCNN on the test set is approximately 82.5%, which is higher than that achieved by traditional ferrograph image classification methods.

The proposed DCNN realizes an end-to-end processing; that is, from the original image to the classification results of different types of wear particles, it unifies the segmentation, classification, and edge location of the wear particles into a single model, avoids the

Table 3
Results of three traditional classification methods.

Method	SIFT + SVM	HOG + SVM	Image + SVM
Acc_w (%)	61.05	79.99	78.49

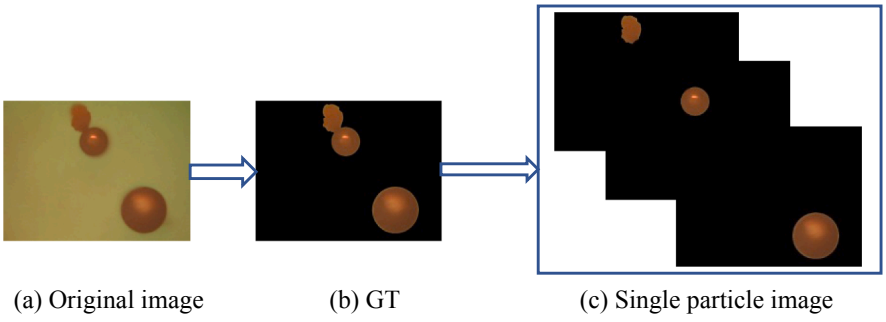


Fig. 6. An example image for comparing different classification methods.

Table 4
Performances of the three models.

Method	FCN	DeepLab v2	DCNN
Acc_w (%)	57.55	56.23	82.50
mIOU (%)	68.70	72.55	84.66
Speed (s)	0.11	0.14	0.16

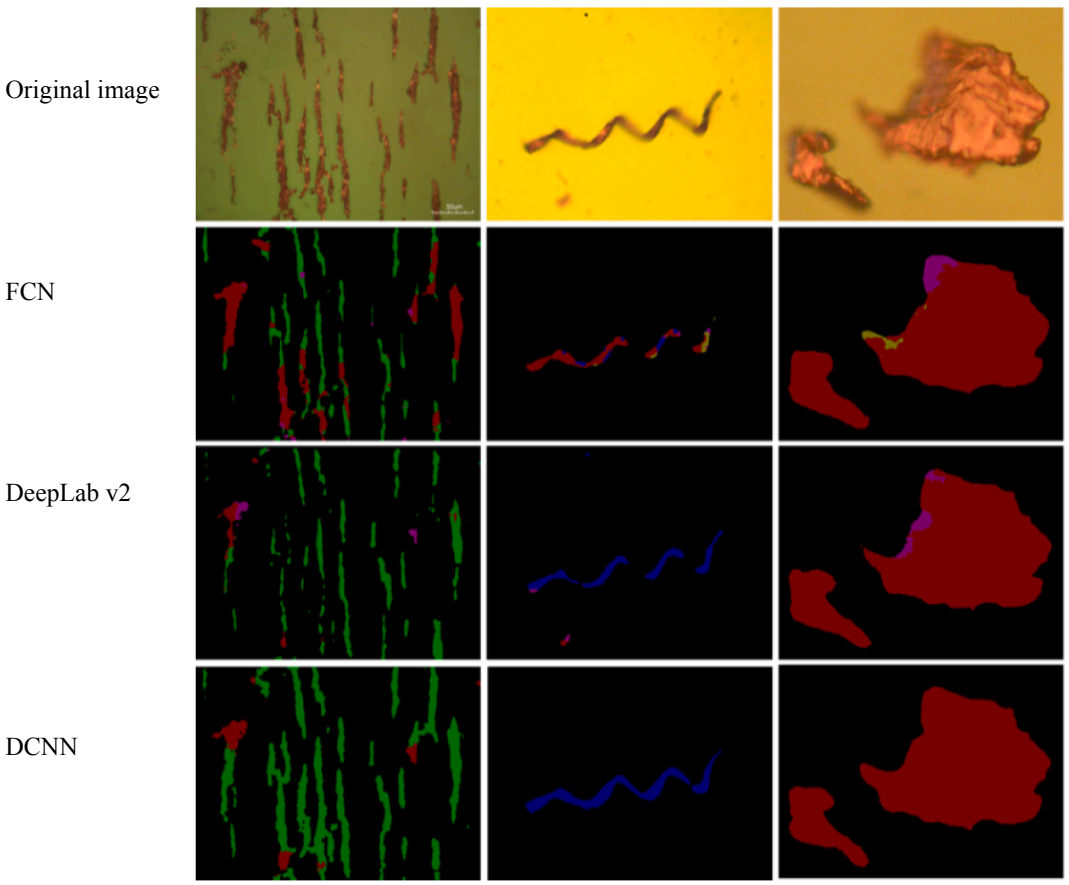


Fig. 7. Semantic segmentation results of the three models.

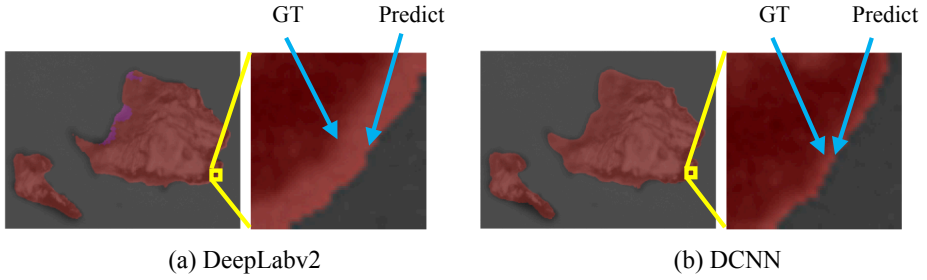


Fig. 8. Edge location of two CNN models.

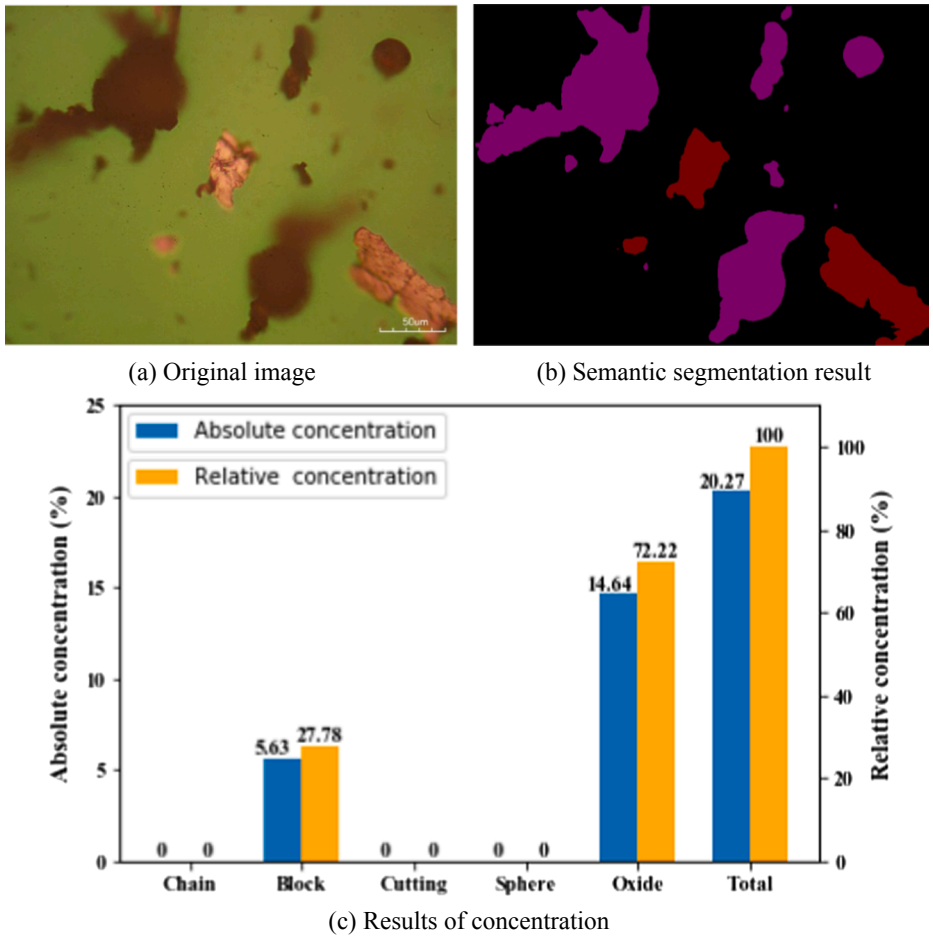


Fig. 9. An example for concentration calculation.

accumulation and transmission of errors caused by numerous steps applied in a traditional linear process, and improves the efficiency and accuracy of wear particle analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Kumar, Advancement and current status of wear debris analysis for machine condition monitoring: a review, Ind. Lubricat. Tribol. 65 (1) (2013) 3–11.

[2] B.J. Roylance, Ferrography—then and now, Tribol. Int. 38 (10) (2005) 857–862.
[3] N. Eliaz, R.M. Latanision, Preventative maintenance and failure analysis of aircraft components, Corros. Rev. 25 (1–2) (2007) 107–144.
[4] N. Otsu, A threshold selection method from gray level histograms, IEEE Trans. Syst. Man Cybernetics 9 (1) (1979) 62–69.
[5] L. Vincent, P. Soille, Watershed in digital spaces: an efficient algorithm based immersion simulations, IEEE Trans. Pattern Anal. Mach. Intell. 13 (6) (1991) 583–598.
[6] J. Wang, L. Zhang, X. Wang, Combining k-means clustering and watershed algorithm for the segmentation of color ferrograph image, J. China Univ. Min. Technol. 42 (5) (2013) 866–872.
[7] H. Wu, T. Wu, Y. Peng, et al., Watershed-based morphological separation of wear debris chains for on-line ferrograph analysis, Tribol. Lett. 53 (2) (2014) 411–420.
[8] W. Yuan, K.S. Chin, M. Hua, et al., Shape classification of wear particles by image boundary analysis using machine learning algorithms, Mech. Syst. Sig. Process. 72–73 (2016) 346–358.
[9] M. Wolski, P. Podsiadlo, G.W. Stachowiak, Characterization of surface topography from small images, Tribol. Lett. 61 (1) (2015) 1–14.
[10] H. Liu, H. Wei, L. Wei, et al., An experiment on wear particle's texture analysis and identification by using deterministic tourist walk algorithm, Ind. Lubricat. Tribol. 67 (6) (2015) 582–593.
[11] G.P. Stachowiak, G.W. Stachowiak, P. Podsiadlo, Automated classification of wear particles based on their surface texture and shape features, Tribol. Int. 41 (1) (2008) 34–43.
[12] N.K. Myshkin, A.Y. Grigoriev, Morphology: Texture, shape, and color of friction surfaces and wear debris in tribodiagnostics problems, J. Frict. Wear 29 (3) (2008) 192–199.
[13] M.h. Laghari, F. Ahmed, Wear Particle Profile Analysis, in: 2009 International Conference on Signal Processing Systems, Singapore, 2009, pp. 546–550.
[14] S. Raadnui, Wear particle analysis—utilization of quantitative computer image analysis: a review, Tribol. Int. 38 (10) (2005) 871–878.
[15] T. Wu, H. Wu, Y. Du, et al., Imaged wear debris separation for on-line monitoring using gray level and integrated morphological features, Wear 316 (1–2) (2014) 19–29.
[16] L. Jiang, G. Chen, A quantitative analysis method in ferrography based on color image processing, in: 1st International Conference on Modelling and Simulation, Nanjing, China, 2008, pp. 512–515.
[17] N.K. Myshkin, H. Kong, A.Y. Grigoriev, et al., The use of color in wear debris analysis, Wear 251 (2001) 1218–1226.
[18] J. Wang, X. Wang, A wear particle identification method by combining principal component analysis and grey relational analysis, Wear 304 (2013) 96–102.
[19] D. Gu, L. Zhou, J. Wang, Ferrography wear particle pattern recognition based on support vector machine, China Mech. Eng. 17 (2006) 1391–1394.
[20] V.D. Gonçalves, L.F.d. Almeida, M.H. Mathias, Wear particle classifier system based on an artificial neural network, J. Mech. Eng. (56) (2010) 284–289.
[21] Q. Li, T. Zhao, L. Zhang, et al., Ferrography wear particles image recognition based on extreme learning machine, J. Electr. Comput. Eng. 2017 (2017) 1–6.
[22] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[23] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems. Lake Tahoe, 2012, pp. 1097–1105.
[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large scale image recognition, ICLR 2015 (2015) 1–14.
[25] R. Girshick, J. Donahue, T. Darrell, et al., Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, USA, 2014, pp. 580–587.
[26] R. Girshick, Fast R-CNN[EB/OL], 2015. http://arxiv.org/abs/1504.08083.
[27] S. Wang, T.H. Wu, T. Shao, et al., Integrated model of BP neural network and CNN algorithm for automatic wear debris classification, Wear 426–427 (2019) 1761–1770.
[28] Y. Peng, J. Cai, T. Wu, et al., A hybrid convolutional neural network for intelligent wear particle classification, 2019.
[29] P. Peng, J. Wang, Wear particle classification considering particle overlapping, Wear 422–423 (2019) 119–127.
[30] P. Peng, J. Wang, FECNN: a promising model for wear particle recognition, Wear 432–433 (2019) 202968.
[31] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651.
[32] L.C. Chen, G. Papandreou, I. Kokkinos, et al., DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2018) 834–848.
[33] K. He, X. Zhang, S. Ren, et al., Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[34] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015. arXiv:1502.03167 [cs.LG].
[35] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL 2013). pp.
[36] K. He, X. Zhang, S. Ren, et al., Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1919.
[37] O. Ronneberger, P. Fischer, T. Brox, "U-Net: convolutional networks for biomedical image segmentation, MICCAI 2015, Part III, LNCS 9351 (2015) 234–241.
[38] A. Ahmed, K. Yu, W. Xu, et al., Training Hierarchical Feed-Forward Visual Recognition Models Using Transfer Learning from Pseudo-Tasks, Springer, Berlin Heidelberg Berlin, Heidelberg, 2008, pp. 69–82.
[39] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, et al., A survey on deep learning techniques for image and video semantic segmentation, Appl. Soft Comput. 70 (2018) 41–65.
[40] J. Yosinski, J. Clune, Y. Bengio, et al., How transferable are features in deep neural networks? Adv. Neural Inform. Process. Syst. Colorado, USA 4 (January) (2014) 3320–3328.
[41] M. Abadi, A. Agarwal, P. Barham, et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, ,016. arXiv:1603.04467.
[42] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.
[43] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–100.