Neurocomputing 380 (2020) 212–224

Contents lists available at ScienceDirect

Neurocomputing

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

A two-stage attention aware method for train bearing shed oil inspection based on convolutional neural networks

Xiao Fu[a], Kenli Li[a,*], Jing Liu[b], Keqin Li[c], Zeng Zeng[d], Cen Chen[a]

[a] College of Information Science and Engineering, Hunan University, China
[b] College of Computer Science and Technology, Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA
[d] Institute for Infocomm Research, A*STAR, Singapore

ARTICLE INFO

Article history:
Received 1 June 2019
Revised 28 October 2019
Accepted 4 November 2019
Available online 9 November 2019

Communicated by Dr. Nianyin Zeng

MSC:
00-01
99-00

Keywords:
Attention mechanism
Image segmentation
Object detection
Railway inspection
Transfer learning

ABSTRACT

As an important component of trains, rolling bearing is always faced with the defection of shed oil, which inevitably threatens the train safety. Therefore, it is of great significance to conduct defection inspection on bearing shed oil. Due to the complex structure of rolling bearings, traditional signal analysis approaches cannot detect the defections of bearing shed oil with high-efficiency and low cost. In recent years, deep learning has achieved remarkable growth and been successfully applied to various computer-vision tasks. Motivated by this fact, we propose a two-stage attention aware method to recognize defections of bearing shed oil. The proposed method is based on convolutional neural networks, can automatically learn bearing defect features, and does not need manual feature design and extraction like traditional methods. The two-stage method cascades a bearing localization stage and a defection segmentation stage, to recognize the defect areas in a coarse-to-fine manner. The localization stage extracts the foremost bearing region and removes the useless part of images, so as to focus the attention of segmentation stage only on the target region. In segmentation stage, we propose a novel attention aware network APP-UNet16, to segment defect areas from extracted bearing region. APP-UNet16 stacks attention gates to enable the attention-aware features change adaptively, and thus can learn to focus on target defect areas automatically. We also utilize transfer learning in constructing the encoder of APP-UNet16, and introduce spatial pyramid pooling to connect the encoder and decoder, to improve traditional UNet. A series of comparative experiments are conducted, to compare our two-stage method with one-stage method which directly perform segmentation on original train images. The results indicate that the proposed two-stage inspection method achieves higher robustness and accuracy in recognizing defect areas with small oil spot. And the experimental results on proposed APP-UNet16 also demonstrate that a better segmentation performance is achieved, compared to traditional UNet and related state-of-art approaches. We will release the source code as well as the trained models to facilitate more research work.

1. Introduction

1.1. Background and motivation

Rail transport remains the mainstream transportation modality of modern times, hence train operation safety continuous to attract substantial attention. If hidden faults for an operating train cannot be discovered in a timely manner, such faults may cause immeasurable losses to life and property. Usually, train fault defection has relied on manual inspection by technicians in factories or around the train when it stops temporally at a station, this is extremely inefficient and is strongly affected by bad weather. To improve the efficiency and reduce unnecessary labour costs, research on automatic train fault detection for moving vehicles is of great interest.

Many studies have made advances in auto diagnosis for train faults in last decades. Yao et al. proposed a wavelet envelope method to diagnose railway bearing faults [40]. Raveendran et al. proposed a method based on dynamic models to evaluate the degree of damage of mechanical equipment [30]. Li proposed a fault detection model for rolling bearings based on the SOM neural network [21]. In addition, in [28], the characteristics of fault signals are represented by wavelet entropy values, then SVM is used as the fault recognition model. However, traditional signal processing

* Corresponding author.
E-mail addresses: fxiao@hnu.edu.cn (X. Fu), lkl@hnu.edu.cn (K. Li), luijing_cs@wust.edu.cn (J. Liu), lik@newpaltz.edu (K. Li), zengz@i2r.a-star.edu.sg (Z. Zeng), chencen@hnu.edu.cn (C. Chen).

https://doi.org/10.1016/j.neucom.2019.11.002
0925-2312/

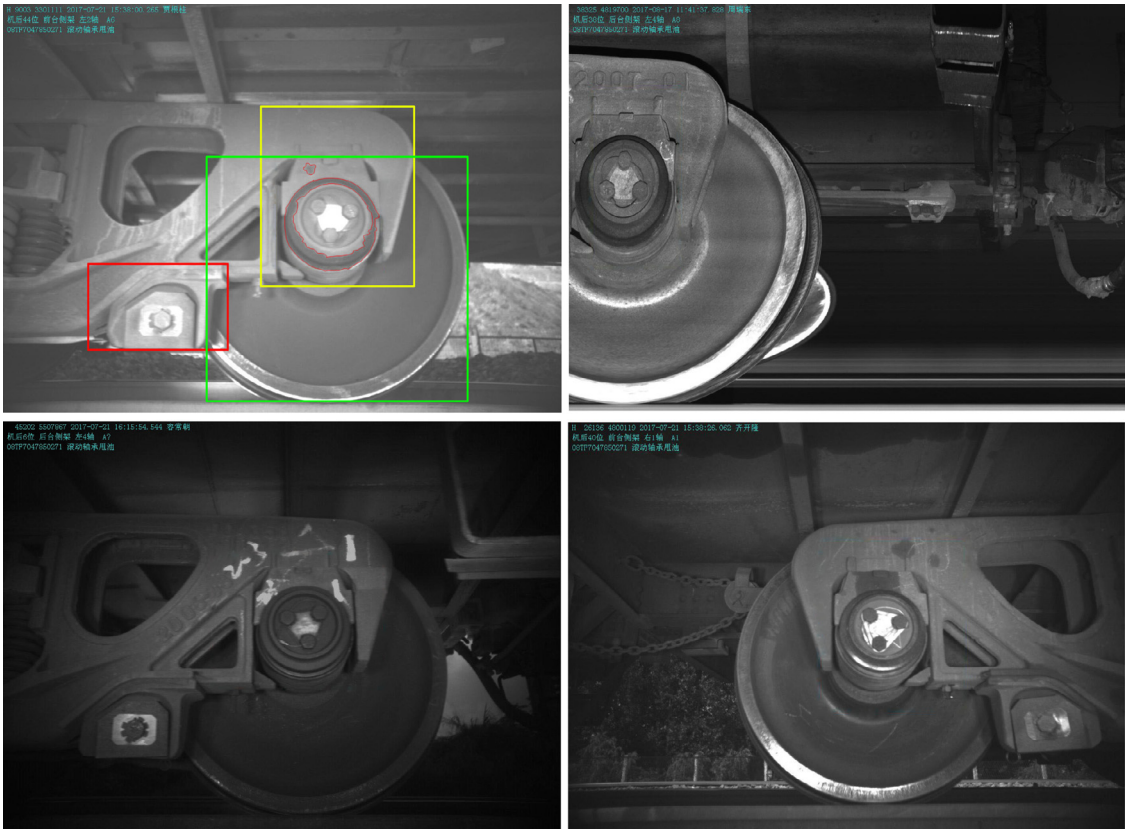

Fig. 1. The original images of train bearings captured by high-speed cameras. The yellow boxed area in the first picture denotes a region of the train bearing covered with oil stains, and the red curve depicts the area covered with spots of oil shed by the train bearing. Moreover, there are many other train components, such as the wheelset, labelled with the green box, and the locking plate, labelled with the red box, in the pictures.

methods are not effective enough, and depend too much on technicians' rich experience of fault detection as well as knowledge of railway systems.

A few years ago, Trouble of Moving Freight Car Detection Systems (TFDS) [35] and similar systems came into the use of the Chinese railway system, which use high-speed line scan cameras to capture images of critical components of moving trains. These images are then transmitted to centralized monitoring offices for further technical analysis. Although productivity has been increased in this manner, technicians remain faced with thousands of images to check every workday, which consumes excessive human resources and is likely to miss many faults. Therefore, automated train fault inspection based on computer vision is becoming a critical area of development. Recently, given their rapid development, deep learning methods are especially successfully applied in a large variety of imaging tasks, such as face recognition [9], object detection and semantic segmentation. And applying deep neural networks in industrial surface defection inspection is becoming more widespread as well. Deep learning methods can learn defection features automatically, while traditional signal processing methods rely on manual feature design and extraction. Based on these factors, we propose an automatic inspection method for train bearing shed oil based on convolutional neural networks, which aims to improve inspection efficiency.

1.2. Challenges

As the main component of a train, rolling bearings are constantly faced with hazards resulting from shed oil. And this defection can lead to overheated or cut axles and can seriously affect the safety of train operations. The major characteristic of this type of fault is the numerous small dark oil spots distributed on and around the bearing, as shown in Fig. 1. Obviously, train bearing inspection requires precise segmentation of defect areas for deeper diagnosis of the extent of the defect. Nevertheless, the inspection of bearing shed oil via deep learning methods of image segmentation must overcome many challenges.

The defect images are grayscale, and captured by high-speed cameras from different angles and different lighting conditions. Therefore, it is easy to confuse oil spots with shadow and difficult to identify defect areas. Moreover, there are two main challenges that must be overcome. First, there are many other components of a train in addition to the bearings, such as wheelsets, locking plates, and angle cocks, in each picture of our dataset, as shown in Fig. 1. Therefore, the train bearing only occupies around 10% of the image, while the oil spots are much smaller than the train bearing. Undoubtedly, it is a typical small object recognition problem. It is hard to directly obtain defection information on bearings from a full image, as there is an excessive amount of useless information that hinders the extraction and analysis of features. Second, there are few defect pictures available to train a CNN. Collecting data from real-world applications is expensive, because of the low probability of fault occurrence and time-consuming nature of labelling ground-truth data. And achieving high performance with limited data is extremely difficult.

1.3. Our contributions

To address the aforementioned challenges and improve the limitations of traditional methods, we propose a two-stage attention

aware method based on convolutional neural networks, to solve the particular problem of bearing shed oil detection. And to the best of our knowledge, this is the first work that utilizing attention mechanism in rolling bearing shed oil inspection. According to our experiments, the hierarchical approach we proposed achieves Dice similarity metric of 0.85 and Jaccard of 0.76. It represents that our method significantly improving the recognition ability of shed train bearing oil and the segmentation performance over related methods. The main contributions of our work are summarized as follows:

- We propose a two-stage attention aware method based on convolutional networks for bearing shed oil inspection, which is composed of a bearing localization stage and a segmentation stage. In localization stage, we reference the attention mechanism of human visual system, extracting the foremost bearing region from a large train image. Thus segmentation stage can be guided to concentrate attention on target regions directly and disregard overmuch background noise. Our two-stage method can better facilitate the recognition of small defect areas with oil spots, compared with one-stage method which simply segmenting original large image. And in the segmentation stage, we propose an attention aware segmentation network to segment defect areas, named APP-UNet16, achieving a better segmentation performance over traditional UNet.
- We design our attention aware segmentation network APP-UNet16 based on a symmetrical architecture of U-Net [33] which is known for working well on small dataset. Moreover, we initialize the encoder of APP-UNet16 with weights from the VGG16 model which is pre-trained on ImageNet [6] by transfer learning, therefore improving the problem that limited training samples cannot satisfy the need of training a deep convolutional neural network.
- We stack attention gates in APP-UNet16 to concatenate the downsampling features and corresponding upsampling features in multi-scales. Attention-aware features are able to change adaptively, so that network can focus on target defect areas automatically without additional supervision.
- We creatively utilize spatial pyramid pooling to connect the encoder and decoder of APP-UNet16, improving the aggregation of global features and the capture of contextual information, thus achieving a better shape prediction of target defection areas.

The rest of our work is organized as follows: In Section 2, we review the related works of object detection and image segmentation methods, etc. Then we present our proposed two-stage attention method in Section 3. And in Section 4, we describe the detail training details of our proposed method. Section 5 discusses the results of experiments we conducted. Lastly, the conclusion of our contributions and the discussion of future work are presented in Section 6.

2. Related work

2.1. Object detection

Object detection is one of the most important research areas in the computer vision field. In recent years, significant advances have been achieved on object detection tasks, based on the important improvements in convolutional neural networks. Compared with traditional feature extraction algorithms based on handcrafted features, such as Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP) and Scale-Invariant Feature Transform (SIFT), deep learning methods achieve substantially better performances. And due to the rapid developments of graphics processing units (GPUs) in recent years, many researches [4,5,20] proposed novel computing architectures based on GPUs, to promote machine learning and deep learning methods to be accelerated by GPUs, like the studies of [23,24,26,29,32]. Based on high performance and effective computing speed, deep learning methods become an inevitable trend in various fields. For example, deep learning methods have achieved successes in time series forecasting like [3], and medical diagnosis like [27] which proposes a diagnosis model based on SVM for Alzheimer's disease. And the computer vision is most widely applied field of deep learning.

The R-CNN [12] is a foundational aspect of object detection algorithms based on region proposal. As a heuristic search algorithm, Selective Search [39] is used to generate candidate regions, which may contain target objects. Region proposals are sent to a CNN model to extract features and predict the classification score of each proposal by SVM classifier. Then, the bounding box is fine-tuned by regression. The R-CNN achieves a very high performance on the VOC2007 dataset [8], but it takes too long to extract features and requires substantial amounts of hard disk space to store the features. SPPNet [14], which creatively combines a Spatial Pyramid Pooling layer between a convolutional layer and a fully connected layer, effectively solves the problem of redundant computation in region proposals, was proposed in 2014. However, it is a multistage process, and fine tuning has a significant effect on accuracy. Girshick et al. proposed a Fast R-CNN [10] based on R-CNN and SPPNet, therein reducing the time consumed for extracting feature vectors from region proposals. Their method combines RoI (Region of Interest) Pooling and Selective Search to extract features and also achieves the synchronous training of object classification and bounding box regression. However, this mehod is not an end-to-end framework. Faster R-CNN [31], proposed in 2017, constructs an RPN (Region Proposal Network) to generate region proposals directly. RPN in Faster R-CNN shares convolutional features with the detector network, thereby achieving end-to-end and real-time object detection, as well as greatly increasing the detection accuracy and speed.

2.2. Image segmentation

Image segmentation is an important aspect of machine vision algorithms for understanding images at the pixel level. With the rapid development of deep learning, image segmentation has entered a new stage of development. Convolutional Neural Networks (CNNs) have become the main trend in this field.

Traditionally, thresholding methods, clustering-based segmentation methods and graph partitioning segmentation methods are utilized for image segmentation broadly. The most popular segmentation methods based on Graph partitioning are Normalized Cut [37] which was proposed by Jianbo Shi et al, and Grab Cut [34] which was proposed by Microsoft Research Cambridge in 2004.

However, concerning deep learning networks in image segmentation, Krizhevsky et al. [19] proposed AlexNet based on CNNs and won the ImageNet 2012 competition for image segmentation and classification tasks. In [11], deep belief network (DBN) is first utilized for analysis of GICS, to segment the control lines. And Shelhamer et al. [36] from Berkeley introduced fully convolutional networks (FCNs) in 2014. This pioneering work in image segmentation, is an end-to-end segmentation work at the pixel level and replaces fully connected layers with convolutional layers. However, the pooling operations result in a loss of significant amounts of information such that the precision of the segmentation masks generated by the FCN is far from satisfactory. To solve this problem, the encoder-decoder network architecture evolves and builds connections between the downsampling and upsampling paths. U-Net [33] for biomedical image segmentation, proposed by Ronneberger et al. in 2015, is a very popular encoder-decoder

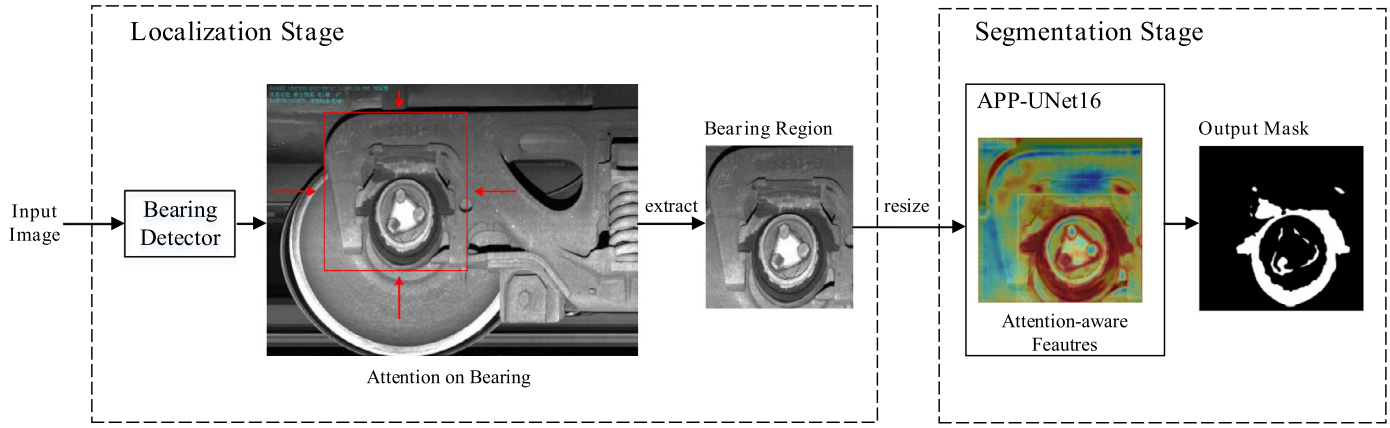

Fig. 2. A sample demonstration of employing the proposed two-stage method on an original train image, for the inspection of shed train bearing oil. The bearing localization stage locates the bearing region and extracts the region. The extracted region is then cropped and resized to 512×512 as input of the segmentation stage. The segmentation models are trained by our proposed APP-UNet16 to learn attention-aware features in segmentation stage, to recognize the oil spot areas from extracted region. Finally, a white-and-black mask obtained as the output.

network and is widely used in many applications of various fields. It is suitable for small training datasets and requires only a short training time and also achieving robust segmentation performance.

The researches in image segmentation mainly direct at two aspects. One direction is multi-scale feature ensembling, like [36] and [50], which acquire deeper semantic information from high-level features in deep networks. The other is based on the post-processing with structure prediction [51]. utilized conditional random field (CRF) to refine the edges of prediction results, thus improves the accuracy. Both directions benefits the prediction ability of image segmentation, however there is still much space to exploit significant information.

2.3. Transfer learning

Though deep convolutional neural networks is proven show high performance on various computer vision tasks, training deep CNNs from scratch is not efficient, due to the demand of large amount of training data while in general there are no sufficient data in real world tasks. In that case, the theory of Transfer Learning was proved can achieve faster and more accurate training of CNNs [25]. Such as training network using ImageNet dataset firstly, and then re-train network using the pre-trained weights as initialization weights. As introduced in [1], transfer learning is able to share and transfer knowledge among different tasks. Therefore, transfer learning can improve performance to some extent and reduce the great demand for training data.

3. Proposed two-stage method

3.1. Overall architecture

In this study, we propose a two-stage attention aware method, to address the challenges that train bearing defection inspection faces, especially the problems of small oil spots recognition and limited training samples. Fig. 2 shows an example of the inspection flow on an original train image. In the rolling bearing localization stage, we use the detector trained based on Faster R-CNN to extract the bearing region where may cover oil stains from an original input image, and remove the useless areas where not occur oil stains. Thus the attention of segmentation network is attached on extracted bearing region where may cover oil spots. The extracted region from the first stage is resized to 512×512 uniformly as input of the segmentation stage. And then in the segmentation stage, via the proposed attention aware network APP-UNet, we can obtain the final segmentation result, in the form of a white-and-black mask where white pixels represent oil spot areas and black pixels represent background.

3.2. Train bearing localization

The inspection of shed train bearing oil is a typical problem of recognizing small objects in a large resolution image. The size of the original image is 1024×1400, the image contains various train components, while the train bearing region only accounts for approximately 10% of the image area, and the oil spots are even much smaller. Therefore, given the demand for segmenting oil spot areas for train bearing defection inspection, it is necessary to guide model to suppress the irrelevant regions of the input image which hinder the information processing, and pay more attention to the region useful for our task.

From this point, we fully utilize the Faster RCNN's advantages to construct a detector and locate train bearing regions that may cover oil stains, and then crop the located region for the next segmentation stage. There are two modules in Faster R-CNN [31]: a deep fully convolutional network, called the RPN (Region Proposal Network) which generates the regions, and the Fast R-CNN detector. As a fully convolutional network, the RPN uses sliding windows to generate multiple rectangular region proposals for each position, which may contain train bearings. Each position is then parameterized to an anchor centred at each sliding window and is related to a scale and aspect ratio. The RPN provides two different outputs for each anchor: the first output is objective scores generated in the cls layer as the probability of an anchor being a target train bearing, and the second output is the bounding box generated in the reg layer to adjust the anchor for better fitting the predicted target. For the Fast R-CNN detector, proposals as RoIs (Regions Of Interest) predicted by the RPN are fed as input of this module. In addition, the final location of the bearing is generated through the bounding-box regressor layer, and the probability of classification is generated through the softmax layer.

After detecting the train bearing region from the input image by Faster R-CNN model, we only obtain four vertex coordinates of the bounding box. As the bounding boxes are irregular in size, and same-size images are required for the subsequent segmentation part, we crop the localized area according to these coordinates and then resize it to 512×512 pixels uniformly.

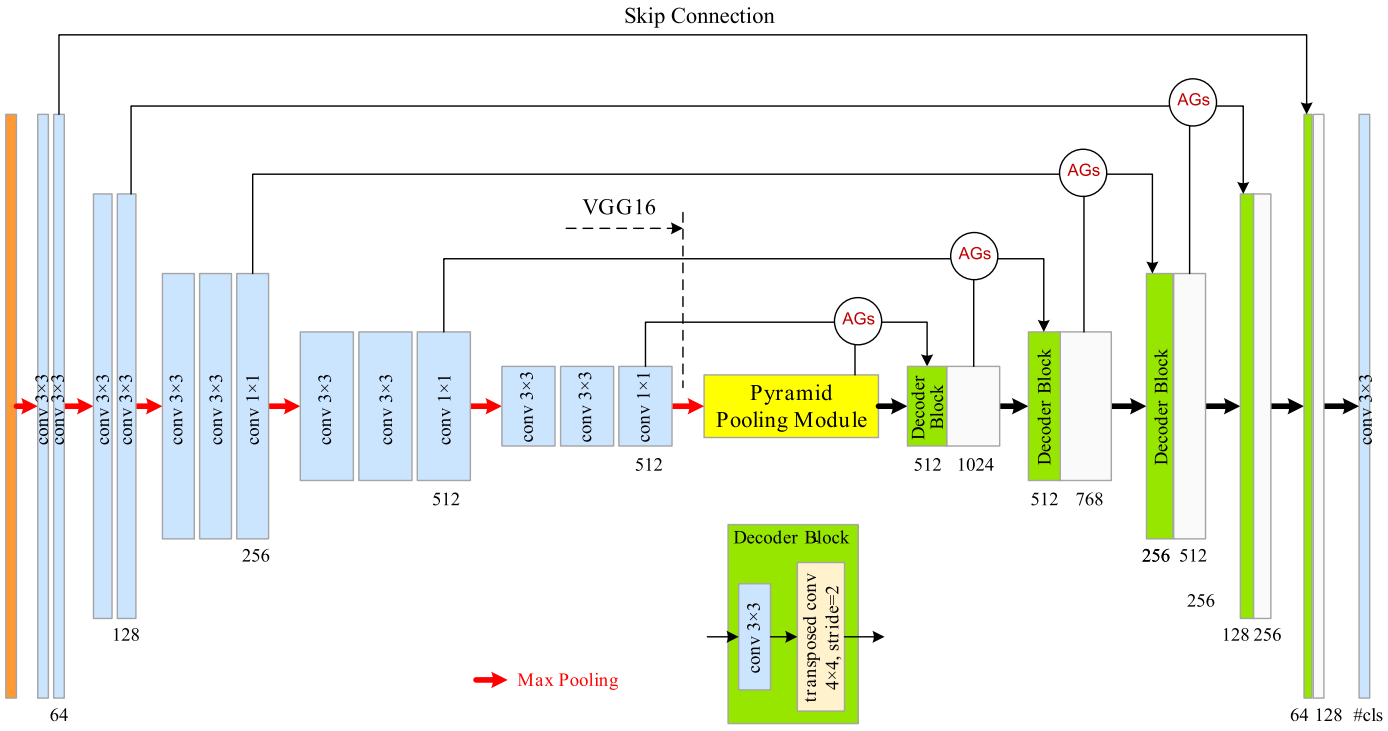

Fig. 3. The architecture of our proposed attention aware segmentation network APP-UNet16, a symmetrical structure based on U-Net and is composed of an encoder and a decoder. The changes in the blue rectangular blocks' height show the transformation of the size of the feature maps, while the width is proportional to the number of channels (the number is given under the corresponding block). The black arrows on top indicate the information transfer from each encoding layer and concatenation to a corresponding decoding layer via an attention gate (AG).

3.3. Proposed attention aware segmentation network: APP-UNet16

Train bearing defection inspection requires precise recognition of oil spots area distributed on the train bearing and around it, thus demanding a class label can be assigned to each pixel of an input image. With train bearing region already focused and extracted, we propose a novel attention aware network APP-UNet16 to perform reliable segmentation on this area. Inspired by U-Net [33] which has demonstrated excellent performance on many image segmentation tasks, especially in tasks with limited training samples, we adopt the symmetrical encoder-decoder architecture of UNet and achieve an improved segmentation network APP-UNet16. And the utilization of transfer learning in constructing the encoder of APP-UNet16 also helps to improve the robustness with limited training images and shorten training time. The proposed APP-UNet16 is capable of automatically learning to pay more attention to the relevant features which are useful for target areas, by stacking attention gates (AGs) in multi-scales. And the spatial pyramid pooling module allows our network to aggregate global contextual information. The full architecture of our attention aware segmentation network APP-UNet16 is illustrated in Fig. 3.

3.3.1. The basic structure based on UNet

U-Net is composed of a contracting path (left part) and an expansive path (right part) in a symmetrical form. The contracting path is built with a typical convolutional network structure, successively performing four times operations of sequential convolution and max pooling, for downsampling and simultaneously increasing the number of feature maps. In the expansive path, an up-convolution operation is used to halve the feature maps coming from the contracting path for upsampling, and increase the resolution of feature maps. Moreover, the skip connections as concatenation operations are performed between the contracting path and the expansive path to combine high-resolution features with upsample features.

However, the traditional U-Net method is trained with randomly initialized weights and cannot achieve the same performance as utilizing transfer learning, especially with limited training samples, as proved by Iglovikov and Shvets [16]. Therefore, we design a novel attention aware segmentation network, named APP-UNet16, based on the encoder-decoder structure of [33], and fully utilize transfer learning.

As shown in Fig. 3, we achieve a symmetrical segmentation network APP-UNet16 based on traditional UNet. In our work, VGG16 [38] is used to construct the encoder of our APP-UNet16, which consists of 13 convolutional layers, each followed by an ReLU activation operation, and 3 fully connected layers. However, we remove all fully connected layers when building the encoder, as our task is a pixel-wise classification problem. All convolutional layers in the encoder use 3×3 filters, and all max pooling layers use 2×2 kernels. Moreover, the strides of the convolutions are fixed to 1 pixel, while the strides of the max pooling operations are fixed to 2 pixels. The first convolutional layer of our encoder produces 64 channels, and the images are halved in size with network going deeper, while the number of channels doubles after each max pooling layer. Ultimately, we obtain 512 channels, and then the number of channels does not change in the subsequent layers of the encoder. For the decoder, the structure is with a stack of decoder blocks, and fully symmetric with the encoder to guarantee precise localization. Each decoder block consists of a single convolutional layer with 3×3 kernels, and a transposed convolutional layer for doubling the size of the feature maps and halving the number of channels. Moreover, we utilize a skip connection similar to that in U-Net between the contracting path and the expanding path in multi-scales. The skip connections concatenate the outputs of the transposed convolutional layers in the decoder, with the

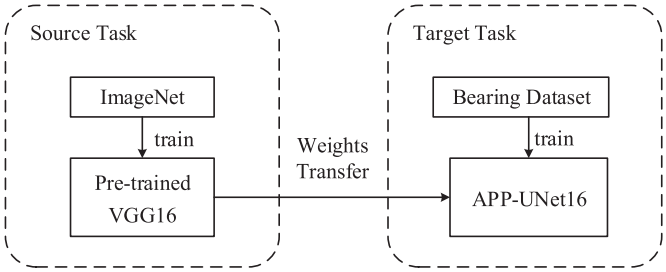

Fig. 4. The process of transfer learning in APP-UNet16. The parameters obtained by training VGG16 network on ImageNet are transferred to the encoder of APP-UNet16 when training on our bearing dataset.

outputs of the corresponding convolutional layer in the encoder, thereby keeping the number of channels same as in the symmetric part of the encoder. In addition, after 5-times upsampling procedures, we obtain 64 channels, and then, one convolutional layer is used to reduce the number of channels. Finally, a convolutional layer is employed to transform the results into a two-class classification problem to judge the classification of each pixel. And white-and-black masks with the size same as input images are obtained at last, of which white pixels present the predicted label of corresponding pixels is oil spots and black pixels present the predicted label of corresponding pixels is background.

In addition, because our training samples are limited, to guarantee the robustness of our segmentation, we utilize transfer learning when constructing our encoder. We first train the VGG16 network on ImageNet dataset for image classification as the source task to obtain the training weights. We do this because the current largest dataset, ImageNet [6], contains more than 14 million images categorized by more than 1000 object classes with approximately 1 million artificial bounding box annotations. Such a large-scale annotated image dataset is beneficial for training a pervasive model. Thus, the CNN model trained on ImageNet can be seen as a good feature extractor of low- and mid-level features such as edges and corners, while also being suitable for other vision tasks, as proven by Donahue et al. [7] and [22]. To adapt the trained CNN to new tasks in our train bearing dataset, we retain the same structure of the trained VGG-16 as our encoder for segmentation. But we remove all fully connected layers and the final classification layer, because our work is pixel-wise classification which does not require a vector of classification probabilities. We initialize our encoder with the weights trained on ImageNet and initialize the weights of our decoder randomly. Then, we finetune the network with an original learning rate of 0.0001. The transfer procedure is illustrated in Fig. 4.

3.3.2. Attention gates in APP-UNet16

In order to help our attention aware segmentation network APP-UNet16 to concentrate on the target defect areas and suppress the irrelevant areas of background, inspired by Oktay et al. [48] and [49], we stack soft attention gates (AGs) in skip connections at each scale. Via AGs to concatenate the downsampling features in the contracting path and the corresponding upsampling features in the expanding path, thereby highlighting the features which pass through skip connections. The structure of AGs utilized in our APP-UNet16 is illustrated in Fig 5.

The whole structure of an AG is used for generating attention coefficients $\alpha_i \in [0, 1]$ for each pixel i, by referencing the attention approach of [42]. Following the default setting of attention mechanisms, AGs learn a single attention value for each pixel vector x^l of layer l which comes from the encoder. Besides, to learn to focus on a subset of targets, an additional attention branch is added for gating vector g. The extracted coarse features $g^{l'}$ obtained from the corresponding layer l' of the decoder are used for gating operation, to disambiguate the effect of uncorrelated and noisy information. The gating vector $g^{l'}$ can provide contextual information, and the feature map x^l can provide local information. Both of them are fused for computing attention coefficients, enables higher accuracy, as suggested in [45].

Firstly, the convolution operation with kernel size of 2×2 and stride of 2 is performed on the input feature map x^l of layer l, while the convolution operation with kernel size of 1×1 and stride of 1 is performed on the gating vector $g^{l'}$, parameterized by W_x^T and W_g^T respectively. By doing these, achieving linear transformations to reduce computation amount. Then the processed x^l, and $g^{l'}$ which is upsampled to the same size as x^l, are added in element-wise to obtain the intermediate feature maps. Next, the ReLU activation function $\sigma_1(x) = max\{0, x\}$ is performed to non-linear transform the intermediate feature maps. The subsequent $1 \times 1 \times 1$ convolution performs linear transformation. And the sigmoid activation function $\sigma_2(x) = \frac{1}{1+exp(-x)}$ normalizes the attention coefficients by reflecting them to (0,1) range.

$$\alpha_i^l = \sigma_2(\Psi^T(\sigma_1(W_x^T x_i^l + W_g^T g_i^{l'} + b_{g^{l'}})) + b_\Psi) \tag{1}$$

where Ψ denotes the linear tranformations, and b denotes the bias.

As Eq. (1) concludes, AGs figure out attention coefficients $\alpha_i^l \in [0, 1]$ for each pixel i of layer l. And the produced attention coefficients α_i^l is utilized to scale the input feature map x^l of layer l, in the form of point multiplication, as shown in Eq. (2).

$$\widehat{x}_i^l = \alpha_i^l \cdot x_i^l \tag{2}$$

Lastly, the output feature maps $\widehat{x}^l$ of AGs are concatenated with corresponding upsampled feature maps in the expanding path. Though there are 5 times skip connections in our APP-UNet16, AG is not utilized in the skip connection of the first scale. Because the feature maps extracted at the first scale are too low-level to reflect the high dimension space, as suggested in [42]. In addition, the parameters of AGs are also passed through backward propagation. The update rule for AGs in layer $l-1$ can be formulated as follows:

$$\frac{\partial(\hat{x}_i^l)}{\partial(\Phi^{l-1})} = \frac{\partial(\alpha_i f(x_i^{l-1}; \Phi^{l-1}))}{\partial(\Phi^{l-1})} = \alpha_i \frac{\partial(f(x_i^{l-1}; \Phi^{l-1}))}{\partial(\Phi^{l-1})} + \frac{\partial(\alpha_i)}{\partial(\Phi^{l-1})} x_i^l \tag{3}$$

where Φ denotes the convolutional parameters need to be updated. And the first item is scaled with α_i.

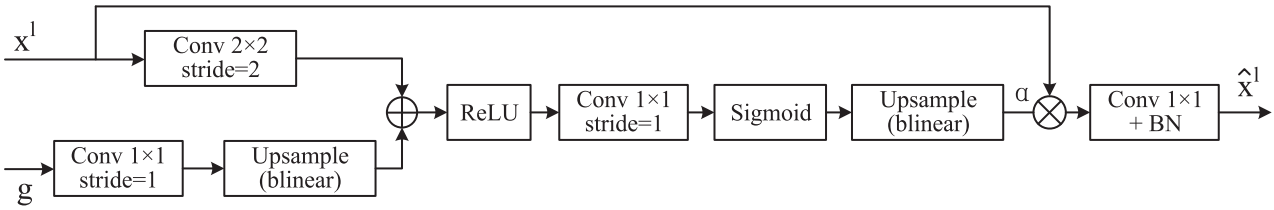

Fig. 5. The structure of attention gate (AG) in our proposed APP-UNet16. Input feature map x_l is scaled with the attention coefficient α generated by AGs. The input feature map x^l and gating vector $g^{l'}$ are utilized together for computing attention coefficients. The upsample operation of feature maps is achieved by bilinear interpolation.

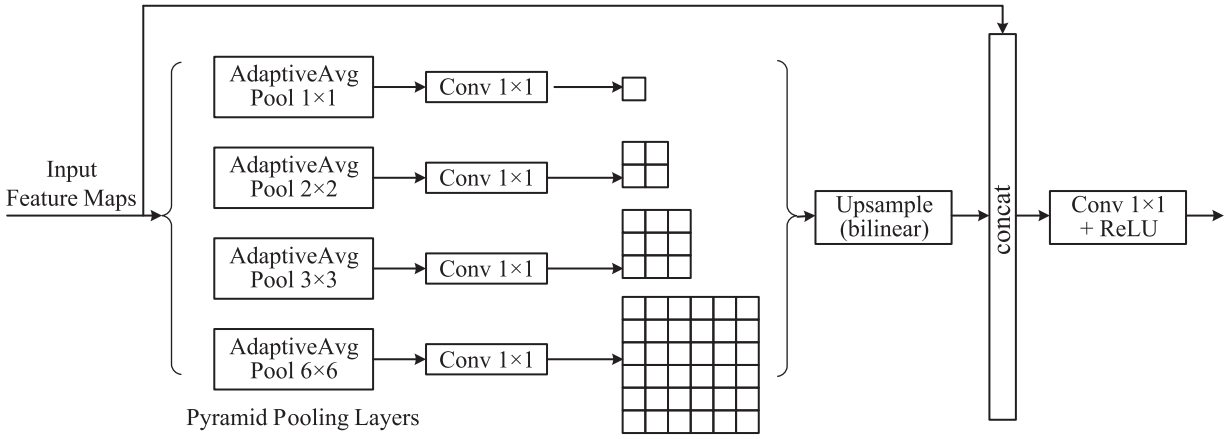

Fig. 6. The structure of the spatial pyramid pooling module in our proposed APP-UNet16. The pyramid pooling helps to capture different sub-region features with size of 1×1, 2×2, 3×3, 6×6, respectively. The following upsample and concatenation layer are used to form the final feature maps which contain both global and local contextual information.

The backward propagation of AGs allows the suppression of background regions. Stacking AGs in skip connection at multiple scales enables the aggregation of contextual information and a better segmentation performance, helping attention aware features change adaptively.

3.3.3. Spatial pyramid pooling module in APP-UNet16

In recent years, spatial pyramid pooling has been commonly applied in various computer vision tasks, such as the excellent object detection method R-CNN [12]. According to [46], the actual receptive field of CNN is much small than the theoretical receptive field. This results in many networks not sufficiently utilizing the global scene. As [14,17] proved, fusing global receptive fields with different sub-region information is a more effective global prior representation. Motivated by this, [47] utilizes pyramid pooling and successfully improves the performance of scene perception. It captures different sub-region information from different scales and then concatenated together to incorporate more global contextual information. Inspired by this, we creatively utilize spatial pyramid pooling to connect the encoder and decoder of APP-UNet16, to improve the segmentation performance. Spatial pyramid pooling helps APP-UNet16 aggregate global context, thus achieving a better prediction of the shape and boundary of the defect areas.

The whole architecture of the pyramid pooling module in our APP-UNet16 is demonstrated in Fig. 6. The produced coarse feature maps with 512 channels which come from the downsampling path are as input, and then fed into 4 different pyramid scales. As an excellent global contextual prior, Global Average Pooling has been widely utilized in image classification tasks such as [15] and [44], it also achieves good results for semantic segmentation in [43]. In this case, we perform the adaptive average pooling operation on input feature maps at each pyramid scale, and the size of pooling kernel varies corresponding to each pyramid scale. In our four-level pyramid pooling module, the size of the pooling kernels at each scale are fixed to 1×1, 2×2, 3×3 and 6×6 respectively, refers to the setting in [14]. The varying-size pooling kernels can be beneficial for the abstraction of different sub-regions, thereby capture global contextual features as much as possible. In order to keep the original size of global features, we connect a upsample layer to perform bilinear interpolation after the pyramid pooling layer at each scale thus to get the same size feature maps as the input features. Lastly, the features from different pyramid scales are concatenated to the input feature maps, to form the ultimate pyramid pooling global feature. As our task acquires pixel-wise classification, we perform a 1×1 convolution to reduce the amount of channels to keep the original dimension, instead of flattening the final global features and then feeding into a fully connected layer like [14] for image classification.

3.4. Data augmentation

Because the number of train bearing image samples is very limited, especially images that contain bearing faults, we want to address the challenges posed by the limited number of samples and imbalanced dataset for image segmentation tasks. In this case, we use an additional data augmentation process to generate additional training data. By data augmentation, to improve the robustness of the network and to learn the desired invariance, which is essential in segmentation tasks when there are minimal available training data.

For bearing images, we primarily use operations such as applying rotation, shift, zoom and shear to generate more samples. The parameters of these augmentation operations are usually set to small values, to ensure that there is only a slight difference between augmented images with original images. We rotate available images by a certain angle randomly to change the orientation of the image contents, and we set the degree range for the random rotations to 0.2. Moreover, we zoom the images in and out by certain scale factors, and we set the range for the random zooming to 0.5. We also shift the images from left to right and from top to bottom, the ranges for both the horizontal and vertical shifts are set to 0.05. Furthermore, we apply a shear transformation to the training data in a certain direction to scale the directed distance from every point of an image to a line parallel to that direction, and the shear angle in the counter-clockwise direction is set to 0.05 degrees. After the above operations, the number of bearing image samples is increased to 4664.

4. Training details

In this section, we introduce the training details of our detection and segmentation models, including the initialization of all parameters. All training processes fully utilize the GPU memory on a system with CentOS Linux release 7.4.1708 (Core) and Tesla P100 PCIe 16GB GPUs.

Training Localization Model. Faster R-CNN consists of an RPN and the Fast R-CNN, where the loss function of the RPN is defined as follows:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \qquad (4)$$

where p denotes the probability of the corresponding anchor being a target object and p^* is the true label of a proposal ($p^* = 1$ when this proposal is a target; otherwise, $p^* = 0$). In addition, t denotes the coordinates of the predicted bounding box, t^* denotes the coordinates of the ground-truth bounding box, i is the index of an anchor in a mini-batch. L_{cls} is the classification loss, while L_{reg} is the regression loss. N_{cls} is the number of anchors, and N_{reg} is the number of anchor locations. The loss function of the Fast R-CNN is defined as follows:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \tag{5}$$

where p is a probability distribution over outputs, u is the class index of the proposal (we only have two classes: train bearings and background), t^u are the coordinates of the bounding box predicted by the model, and v is the bounding box of the ground truth. And the value of $[u \geq 1]$ is 1 when $u \geq 1$ and 0 otherwise. Moreover, L_{cls} is the classification loss for the true class u, L_{loc} is the regression loss for the bounding box, and λ is a hyper-parameter.

In this paper, instead of training the Faster R-CNN model by 4-step alternating training as in [31], which trained the RPN and Fast R-CNN independently, we train our bearing detection model in an end-to-end manner using approximate joint training to achieve a faster and more convenient training process. We merge the two networks, the RPN and Fast R-CNN, into one network when training, and we fine tune this network by stochastic gradient descent (SGD) [2] to optimize the features for both tasks. Four types of losses are obtained: the RPN classification loss, the RPN regression loss, the RoI classification loss and the RoI regression loss. The RPN loss is combined with the Fast R-CNN loss for the shared layers during backward propagation. For the anchors in the RPN, we use three scales, 128^2, 256^2, and 512^2 pixels, and 3 aspect ratios, 1: 1, 1: 2, and 2: 1, with $k = 9$ anchors at each sliding window, and obtain the same number of proposals, as suggested in [31]. To address highly overlapped proposals and reduce redundancy, we use non-maximum suppression (NMS) on the proposals according to their cls scores. And the intersection-over-union (IoU) threshold of the NMS is set to 0.8, as there is only a target class in our task, such setting can prevent missing recognition of target objects. After NMS, the number of proposals decreases sharply, as there is only one target object, i.e., a train bearing, in each picture of our dataset, thus we choose the top-1 ranked proposal for detection.

Training segmentation model. The image segmentation task can be considered as a pixel-wise classification problem, and our task for segmenting oil stains is a typical binary classification problem. Thus, our attention aware segmentation network APP-UNet16 uses the common loss function of binary classification, which is defined as follows:

$$L_1 = \frac{1}{n}\sum_{i=1}^{n}(-y_i \log \hat{y}_i - (1 - y_i)\log(1 - \hat{y}_i)) \tag{6}$$

where y_i is an annotated binary value of pixel i, while $\hat{y}_i$ is the predicted probability of the corresponding pixel, and n is the number of pixels. Moreover, to maximize the probabilities of obtaining the correct pixels and the intersection between the predictions and the corresponding ground-truth mask simultaneously, the loss function of APP-UNet16 is defined in the following manner:

$$L = L_1 - \log\left(\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i\hat{y}_i}{y_i + \hat{y}_i - y_i\hat{y}_i}\right)\right) \tag{7}$$

The model is trained by minimizing the loss function through the Adam optimizer [18] with the learning rate initialized to 0.0001, as small learning rate is more suitable for transfer learning. We start the training models with a batch size of 16, to obtain stronger higher extensive ability.

5. Experiments

5.1. Experimental setup

Data description. We build two datasets for training and evaluating our two-stage attention aware method for bearing shed oil inspection. All images of the operating train used in our experiments on the localization model are captured by high-speed cameras, with a high resolution of 1400×1024, therein containing both the train bearing and miscellaneous components. To build the train bearing dataset for locating train bearing regions through the object detection model we trained, we label the train bearing areas in each image by drawing a rectangle. Then, we record the four coordinates of the labelled area and annotate the labelled region as a 'bearing'. All information about the labelled rectangles and relevant classification are organized as XML files and matched with the corresponding image. There are 240 training samples and 120 testing samples in our train bearing dataset. For building our segmentation dataset to identify the oil spot regions from the pictures, we record the locations of the oil spots using curved lines and label these areas in white. And we label the remaining background in black. After data augmentation, our segmentation dataset contains 3364 images for training and 1300 images for testing.

Evaluation metrics. The Faster R-CNN model achieves excellent performance on multi-class object detection and classification tasks. However, in our research, we only trained the model to detect the location of train bearings, which is a single-class classification problem. In addition, our purpose is to obtain the location of the train bearings as precisely as possible, thus building a better foundation for next stage's segmentation. Therefore, to evaluate the object detection model that we trained for locating bearings, we do not utilize the mAP, which is commonly used in multi-class classification tasks, as our evaluation metric. Instead, we use IoU (Intersection over union) to evaluate our object detection model. IoU measures the relevance score of the predicted area generated by the model that we trained and our labelled ground truth. Assume set A as a predicted area and set B as the ground truth. IoU as a similarity measure can be defined as follows:

$$IoU(A, B) = \frac{|A \bigcap B|}{|A \bigcup B|} = \frac{|A \bigcap B|}{|A| + |B| - |A \bigcup B|} \tag{8}$$

At the segmentation stage, we obtain a white-and-black mask predicted by our segmentation method, where white areas denote the target area, i.e., the area covered by oil stains, and black areas denote background. To evaluate the pixel-wise accuracy, we use the above-mentioned IoU to measure the similarity between the predicted results and the labelled ground truth. In addition, we choose Dice's Coefficient as the similarity evaluation metric, which is defined as follows:

$$Dice(A, B) = \frac{2|A \bigcap B|}{|A| + |B|} \tag{9}$$

5.2. Experiments of our proposed method

In this section, we run a series of comparative experiments to illustrate the performance of our proposed two-stage attention aware method for bearing shed oil inspection. Our two-stage inspection method based on convolutional neural networks can accomplish the inspection of an image in average 2.432 seconds, while the manual image-based train fault detection process may cost at least 1 minute for each technicians. Undoubtedly, our method improves efficiency and saves artificial, compared with manual inspection which is labor intensive. Moreover, we compare our two-stage method with the one-stage method without the localization stage. To evaluate the performance of the 'one-stage' method, we remove the procedure of locating the region of the

Fig. 7. Binary masks, where white pixels represent areas of oil stains and black pixels represent background. Image (a) shows the segmentation results of the original image in Fig. 8 predicted by the one-stage model, and (b) shows a ground-truth mask.

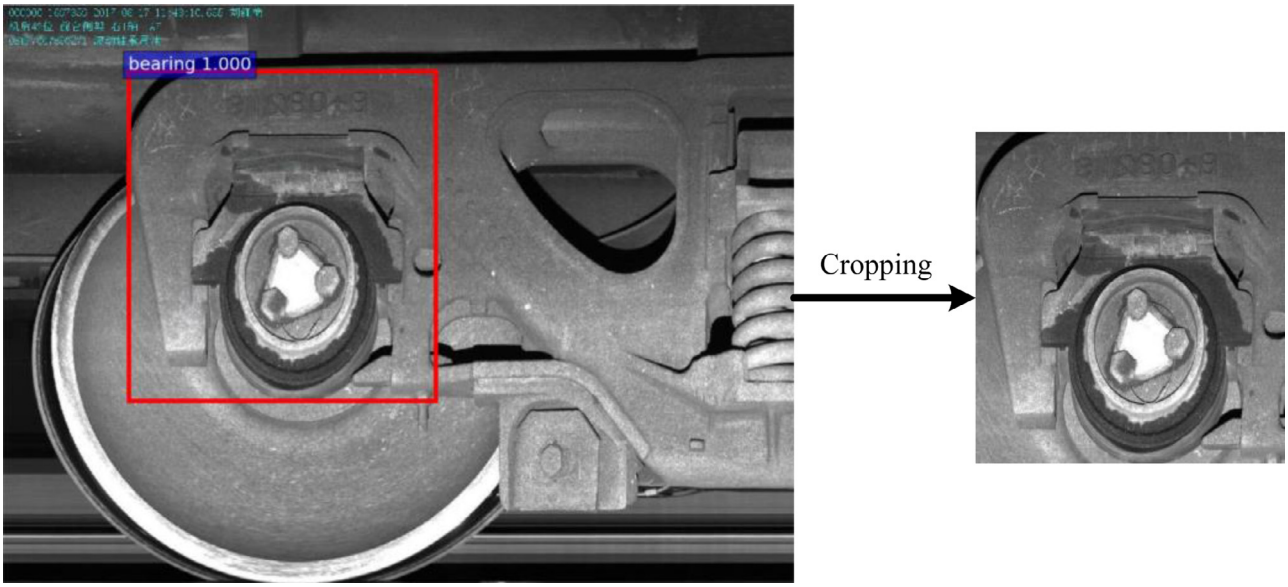

Fig. 8. The predicted train bearing region by the detection model is indicated by the red rectangle; the over label name and corresponding score denote the probability of the region being the target object. The region is cropped for the next stage.

Table 1
Accuracy comparison in different conditions of the second stage, between one-stage method and our proposed two-stage attention inspection method.

Method	Condition	Dice (mean)	Dice (std)	IoU (mean)	IoU (std)
one-stage	UNet	0.3608	0.1916	0.2370	0.1454
	UNet16	0.5705	0.1917	0.4258	0.2000
	no AGs	0.6204	0.1759	0.4733	0.1869
	no pyramid pooling	0.6283	0.667	0.4798	0.1801
	APP-UNet16	**0.6545**	**0.1496**	**0.5050**	**0.1678**
two-stage	UNet	0.7143	0.2709	0.6120	0.2717
	UNet16	0.7941	0.1467	0.6799	0.1767
	no AGs	0.8492	0.1222	0.7551	0.1620
	no pyramid pooling	0.8374	0.1221	0.7369	0.1584
	APP-UNet16	**0.8581**	**0.1177**	**0.7676**	**0.1580**

rolling bearing, and we directly use the original images captured by the high-speed cameras for segmentation. Fig. 7. illustrates the binary mask directly predicted from an original image by the 'one-stage' method and its ground truth, while the final segmentation result of our two-stage method is illustrated in Fig. 10 (E). And Table 1 demonstrates the segmentation accuracy of our proposed two-stage method, and the one-stage method in various conditions of the remaining second stage; such as with standard UNet, UNet16 (pretrained on VGG16) without AGs and without pyramid pooling respectively for segmentation. The results validate the effectiveness of our proposed two-stage method, which locates the focused bearing region. The proposed two-stage attention aware method enables the segmentation models to focus on the target area and then achieve a refine segmentation. Next, we will discuss the experimental results of our localization and segmentation parts more detail separately in Sections. 5.3 and 5.4.

5.3. Localization experiment

We perform train bearing localization using our trained Faster R-CNN model. In particular, we use the weights of the VGG-16 model trained on ImageNet to initialize the weights of the network for training, and a GPU is used to accelerate the training. The training and testing implementations in our work are from the open-source code implemented under TensorFlow 1.3.0 [13]. We obtain our Faster R-CNN model after 150,000 iterations and obtain a total loss of 0.0128, a classification loss of 0.0046, a regression loss of 0.008, and both RPN classification and RPN regression losses of 0.0001. In addition, the inputs of the training and testing in this phase are of the original size of 1400×1024. The testing results

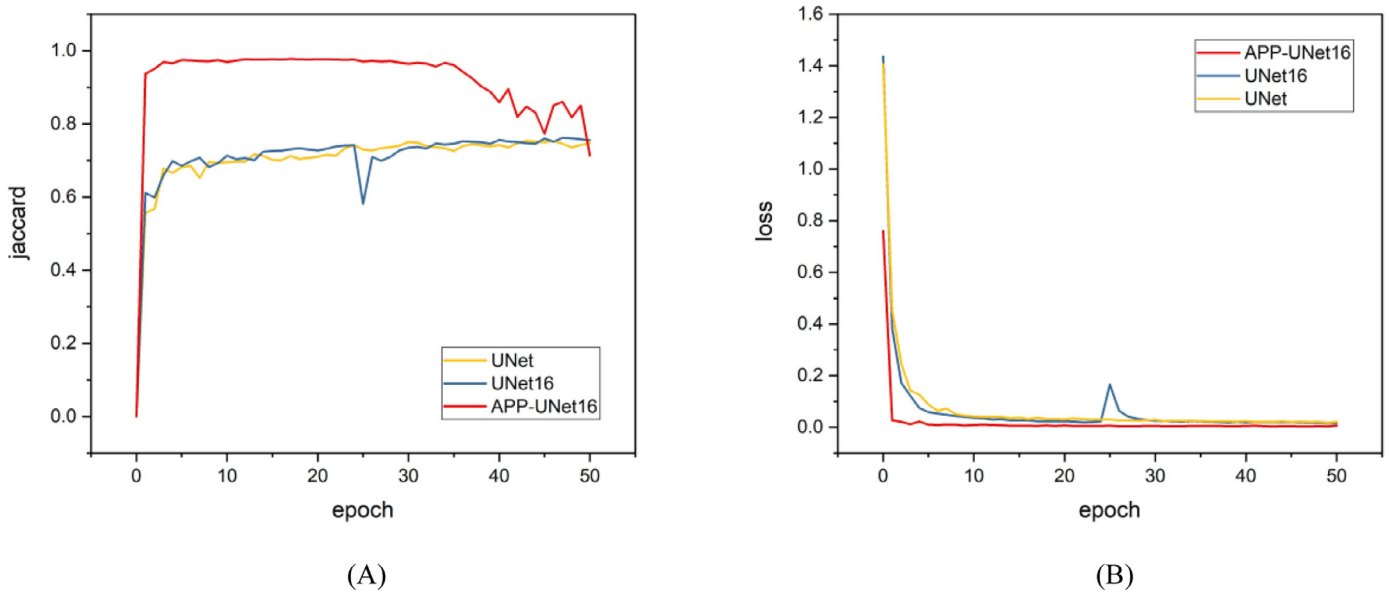

Fig. 9. The Jaccard results of a training epoch for the three segmentation models in Image (A), and the loss results in Image (B). The yellow line presents the UNet model trained from scratch, the blue line presents the UNet16 model trained with the weights from the pre-trained VGG16, and the red line presents the model trained by our proposed APP-UNet16.

Table 2
The comparisons of train bearing localization accuracy achieved by the Faster R-CNN models with different initialization.

Method	IoU
Faster R-CNN (ResNet50)	0.9058
Faster R-CNN (VGG-16)	**0.9371**

Table 3
Accuracy comparison between UNet model with randomly initialization weights; UNet16, which is initialized with the weights from VGG16; and our proposed APP-UNet16 (also initialized with the weights from VGG16).

Method	Dice (m)	Dice (std)	IoU (m)	IoU (std)
UNet	0.7143	0.2709	0.6120	0.2717
UNet16	0.7941	0.1467	0.6799	0.1767
APP-UNet16	**0.8581**	**0.1177**	**0.7676**	**0.1580**

obtained by the Faster R-CNN models with different initialization are evaluated by IoU and presented in Table 2. The results represent that utilizing pre-trained VGG-16 as the feature extractor achieves the best performance with IoU of 93.71%, and an example of the detection results is shown in Fig. 8. According to our experiment, we can conclude that the automatic extraction of bearing areas by the Faster R-CNN is effective, and is able to replace the manual work necessary for extracting the train bearing regions for further segmentation.

5.4. Segmentation experiments

Comparison with traditional UNet. To obtain increased performance with limited training samples, we adopt the basic architecture of UNet to construct our attention aware segmentation network APP-UNet16. And VGG-16 is utilized to construct the encoder and the weights of which is initialized by the VGG-16 model pre-trained on ImageNet, by transfer learning. In order to evaluate the performance improvement of our proposed APP-UNet16 over the standard UNet, we compare it with traditional UNet and UNet16 (initialized with VGG-16 weights), to show the effect of transfer learning. The comparison results, which consist of the mean value (denotes *m*) and standard deviation (denotes *std*) of Dice and IoU, are summarized in Table 3.

The results demonstrate the advantages of transfer learning as UNet16 outperforms than traditional UNet, with Dice improvement of 0.08 and IoU improvement of 0.06. In this case, our proposed APP-UNet16 achieves the best performance, with an Dice of 85.81% and IoU of 76.76%, markedly improves the segmentation accuracy over the traditional UNet. In addition, the learning curves and the loss curves in Fig. 9 also demonstrate the advantages of our APP-UNet16 architecture. First, the APP-UNet16 model, rapidly achieves a higher accuracy after the first epoch. Second, our model more quickly converges to a steady value. In addition, the obtained stable value of our model is much higher. Finally, the differences between the masks predicted by traditional UNet, UNet16, our APP-UNet16 are shown in Fig. 10.

Table 4
Accuracy results comparison between methods with different AGs distributions. Our method adds attention modules in scale 2, 3, 4 and 5 respectively.

Method	Dice (m)	Dice (std)	IoU (m)	IoU (std)
APP-UNet (no AGs)	0.8113	0.1685	0.7109	0.2010
APP-UNet (scale 2)	0.8430	0.1046	0.7421	0.1489
APP-UNet (scale 2+3)	0.8440	0.1338	0.7500	0.1727
APP-UNet (scale 2+3+4)	0.8492	0.1222	0.7551	0.1620
APP-UNet16 (scale 2+3+4+5)	**0.8581**	**0.1177**	**0.7676**	**0.1580**
APP-UNet16 (scale 1+2+3+4+5)	0.8422	0.0763	0.7344	0.1065

Attention analysis. In order to enable our proposed attention aware segmentation network APP-UNet16 to learn the most significant regions automatically, we stack attention modules (AGs) at four-scales. Note there is no AG integrated in the first scale as the first scale are too low-level to reflect the high dimension space. And to evaluate the efficiency of our practice, we remove one, two, three and all AGs respectively, to compare with our method, the comparison results are shown in Table 4. Obviously, our method achieves the best performance with Dice of 0.8581 and IoU of 0.7676. And the predicted masks by APP-UNet16 (no AGs) and APP-UNet16 are shown in Fig 10.

Moreover, the attention heatmaps from the test image are visualized in Fig. 11, respectively from low levels to high levels of

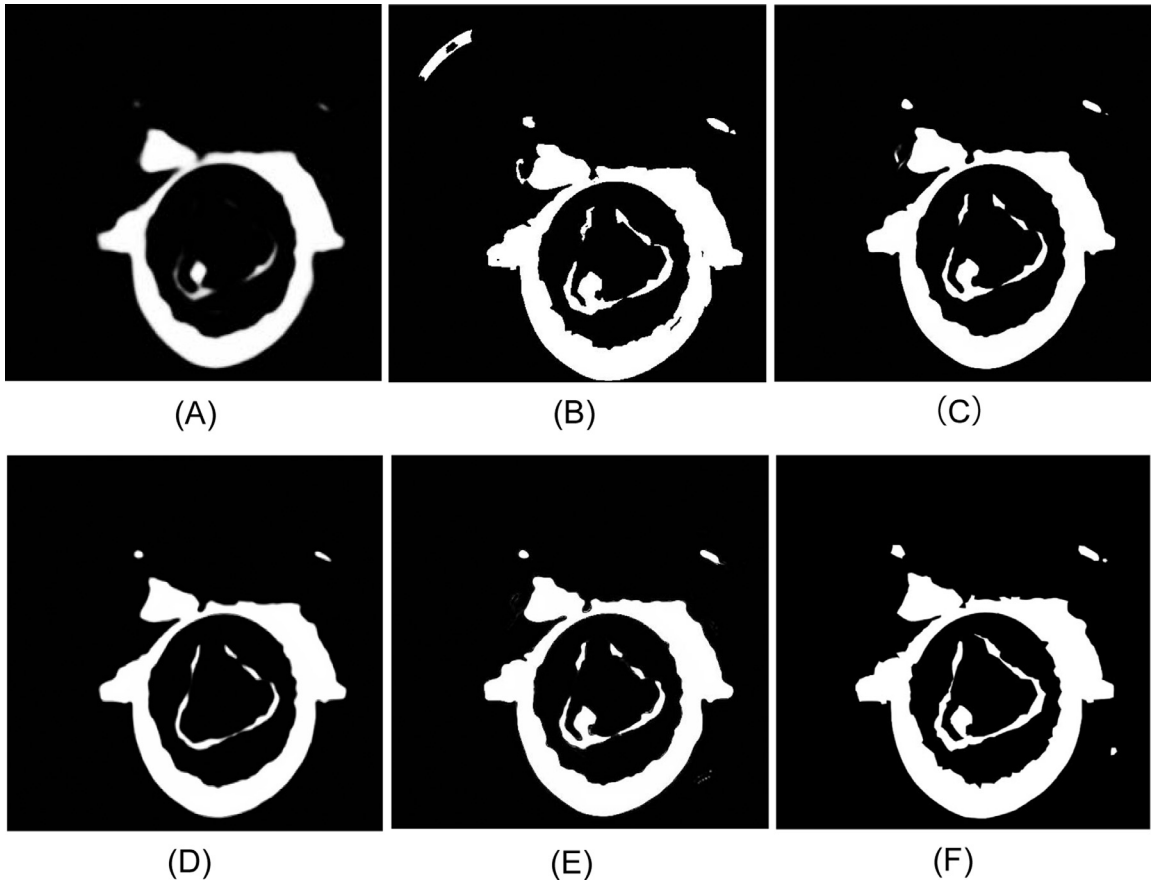

Fig. 10. Binary masks, where white pixels represent areas containing oil stains and black pixels represent background. Image (A), (B), (C), (D) show the prediction results obtained by the segmentation models trained respectively, by UNet, UNet16, APP-UNet (no AGs) and APP-UNet (no pyramid pooling), and (E) by our proposed APP-UNet16. (F) shows the ground-truth mask.

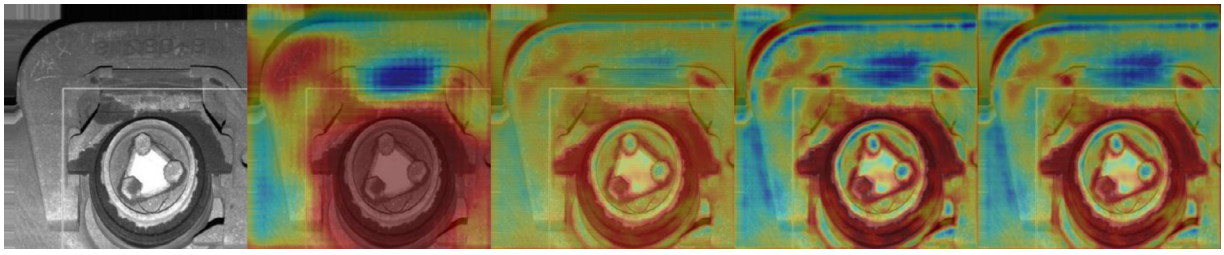

Fig. 11. From left to right: an extracted rolling bearing region; and four heatmaps visualized from the outputs of multiple AGs, from coarser scales to finer scales.

AGs. We can observe that AGs in skip connections from coarse scales only provide a rough outline of defect areas, and sequentially refined in finer scales. It is represented that attention modules of our proposed APP-UNet16 helps activating target image regions and pruning unrelated feature responses.

Pyramid pooling analysis. To capture the global contextual information, the pyramid pooling is utilized to connect the encoder and decoder of APP-UNet16. In our central pyramid pooling module, there are multi-level adaptive average pooling layers to capture the sub-regions with size of 1×1, 2×2, 3×3 and 6×6 respectively. To prove that multi-level pooling improves segmentation accuracy, we compare our method with no pyramid pooling and single-level pooling, the results are shown in Table 5. The results fully demonstrate the superiorities of utilizing pyramid pooling.

Comparison with state-of-art methods. Except for comparing with standard UNet model, we also conduct a series comparative experiments on baseline segmentation models, such as FCN [36] and Segnet [41], the corresponding results are given in Table 6. It demonstrates that our proposed APP-UNet16 shows the state-of-art segmentation performance than other methods, with an Dice of 85.81% and IoU of 76.76%, so that has high capacity to segment defect areas (oil spot areas) thus achieve a high-quality inspection for rolling bearing shed oil.

Table 5
Accuracy results comparison between no pyramid pooling method, single pooling method with different size; and our multi-level pooling with size 1×1, 2×2, 3×3, and 6×6.

Method	Dice (m)	Dice (std)	IoU (m)	IoU (std)
APP-UNet (no pooling)	0.8374	0.1221	0.7369	0.1584
APP-UNet (single pooling with 1×1)	0.7806	0.1080	0.6522	0.1366
APP-UNet (single pooling with 2×2)	0.8263	0.1127	0.7178	0.1431
APP-UNet (single pooling with 3×3)	0.8521	0.0913	0.7523	0.1256
APP-UNet (single pooling with 6×6)	0.8106	0.1380	0.7008	0.1678
APP-UNet16 (multi-level pooling)	**0.8581**	**0.1177**	**0.7676**	**0.1580**

Table 6
Accuracy comparison of state-of-are segmentation methods, FCN8s, FCN32s, and Segnet, respectively.

Method	IoU (m)	IoU (std)	Dice (m)	Dice (std)
FCN8s	0.5967	0.1101	0.7413	0.0881
FCN32s	0.7210	0.1653	0.8246	0.1391
Segnet	0.6524	0.1717	0.7731	0.1610
APP-UNet16	**0.8581**	**0.1177**	**0.7676**	**0.1580**

6. Conclusion and future work

In this work, we propose a two-stage attention aware method for bearing shed oil inspection, based on convolutional neural network. The localization stage guides the segmentation network to zoom in on the train bearing region to see better. In this stage, we use a detector based on Faster R-CNN to extract the region of interest for subsequent segmentation. And to improve the robustness and accuracy of our inspection method, we develop an attention aware U-shaped segmentation method APP-UNet16, in the segmentation stage. We stack attention modules in multi-scales, and introduce pyramid pooling into APP-UNet16. Additionally, in order to address the problem of limited training samples, we implement the encoder of APP-UNet16 by transfer learning, and utilize data augmentation to enlarge our dataset. The effectivenesses of our proposed two-stage inspection method and APP-UNet16 are demonstrated, by evaluating them on our train bearing dataset and comparing with related approaches. For future work, we will explore expanding our inspection method to other similar faults encountered by operating trains. In addition, we will further attempt to achieve higher segmentation accuracies and faster parallel training speed like [52] on our segmentation network.

Declaration of Competing Interest

The authors declared that they have no conflicts of interest to this work.

Acknowledgments

The research was partially funded by the National Key R&D Program of China (Grant No.2018YFB1003401), the National Outstanding Youth Science Program of National Natural Science Foundation of China (Grant No. 61625202), the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China (Grant No. 61661146006, 61860206011), the National Natural Science Foundation of China (Grant No. 61602350), the Singapore-China NRF-NSFC Grant (Grant No. NRF2016NRF-NSFC001-111), the Key R&D Program of Hunan Province (Grant No. 2018GK2051), the Leading Talent Program of Science and Technology of Hunan Province (Grant No. 2017RS3025).

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[2] L. Bottou, F.E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning, SIAM Rev. 60 (2) (2018) 223–311.
[3] C. Chen, K. Li, S.G. Teo, G. Chen, X. Zou, X. Yang, R.C. Vijay, J. Feng, Z. Zeng, Exploiting spatio-temporal correlations with multiple 3d convolutional neural networks for citywide vehicle flow prediction, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 893–898.
[4] C. Cen, K. Li, A. Ouyang, T. Zhuo, K. Li, Gflink: an in-memory computing architecture on heterogeneous CPU-GPU clusters for big data, IEEE Trans. Actions Parallel Distrib. Syst. 29 (6) (2018) 1275–1288. 2018.
[5] J. Chen, K. Li, K. Bilal, X. Zhou, K. Li, P. Yu, A bi-layered parallel training architecture for large-scale convolutional neural networks, IEEE Trans. Parall. Distrib. Syst. (1) (2018). 1–1.
[6] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F.F. Li, Imagenet: A Large-scale Hierarchical Image Database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. 2009.
[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, in: Proceedings of the International conference on machine learning, 50, 2013, pp. I–647.
[8] M. Everingham, The Pascal Visual Object Classes Challenge 2007 (Voc2007) Results, in: Lecture Notes in Computer Science, 111 (1), 2005, pp. 98–136.
[9] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, A.M. Dobaie, Facial expression recognition via learning deep sparse autoencoders, Neurocomputing 273 (2018) 643–649.
[10] R. Girshick, Fast R-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[11] N. Zeng, Z. Wang, H. Zhang, W. Liu, F.E. Alsaadi, Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cognit Comput 8 (4) (2016) 684–692.
[12] R. Girshick, D. Tea, J. Donahue, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[13] Google, Tensorflow: An open source machine learning framework, 2015, https://tensorflow.google.cn/.
[14] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2014). 1904–16.
[15] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 770–778.
[16] V. Iglovikov, A. Shvets, Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, arXiv:180105746.
[17] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, IEEE, 2006, pp. 2169–2178.
[18] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: the 3rd International Conference for Learning Representations, 2015, arXiv:14126980.
[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, in: Proceedings of the International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.
[20] C. Cen, K. Li, A. Ouyang, K. Li, Flinkcl: An Opencl-based In-memory Computing Architecture on Heterogeneous CPU-GPU Clusters for Big Data, IEEE Trans Act. Comput. 67 (2018) 1765–1779. 2018.
[21] Z.C. Li, A simple som neural network based fault detection model for fault diagnosis of rolling bearings, Appl. Mech. Mater. 397–400 (2013) 1321–1325.
[22] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks, in: Proceedings of the Computer Vision and Pattern Recognition, 2014, pp. 1717–1724.
[23] C. Cen, K. Li, A. Ouyang, T. Zhuo, K. Li, Gpu-accelerated Parallel Hierarchical Extreme Learning Machine on Flink for Big Data, in: Proceedings of the IEEE Transactions on Systems Man & Cybernetics Systems, 99, 2017, pp. 1–14.
[24] M. Duan, K. Li, X. Liao, K. Li, A parallel multiclassification algorithm for big data using an extreme learning machine, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2337–2351.
[25] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[26] W. Yang, K. Li, K. Li, A parallel solving method for block-tridiagonal equations on CPU–GPU heterogeneous computing systems, J. Supercomput.. 73 (5) (2017) 1760–1781.
[27] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, Y. Li, A new switching-delayed-pso-based optimized SVM algorithm for diagnosis of Alzheimers disease, Neurocomputing 320 (2018) 195–202.
[28] N. Qin, W.D. Jin, J. Huang, P. Jiang, Z.M. Li, High speed train bogie fault signal analysis based on wavelet entropy feature, Adv. Mat. Res. 753–755 (12) (2013) 2286–2289.
[29] X. Zhu, K. Li, A. Salah, L. Shi, K. Li, Parallel implementation of mafft on cuda-enabled graphics hardware, IEEE/ACM Trans. Comput. Biol. Bioinf. 12 (1) (2015) 205–218.
[30] R.K.S. Raveendran, M.H. Azariana, N.H. Kimb, M. Pechta, Effect of multiple faults and fault severity on gearbox fault detection in a wind turbine using electrical current signals, CHEMICAL ENGINEERING 33 (2013).
[31] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[32] W. Yang, K. Li, K. Li, A hybrid computing method of SPMV on CPU–GPU heterogeneous computing systems, J. Parallel Distrib. Comput. 104 (2017) 49–60.
[33] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional Networks for Biomedical Image Segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[34] C. Rother, V. Kolmogorov, A. Blake, "Grabcut": Interactive Foreground Extraction Using Iterated Graph Cuts, in: Proceedings of the ACM SIGGRAPH, 2004, pp. 309–314.
[35] R. Liu, Y. Wang, Principle and application of tfds, Chinese Railways 5 (2005) 26–27.
[36] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2014) 640–651.

[37] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transpattern Analmachintell 22 (8) (2000) 888–905.
[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, arXiv:14091556.
[39] J.R.R. Uijlings, V.D.K.E.A. Sande, Gevers, A.W.M. Smeulders, Selective search for object recognition. Int. J. Comput. Vis. 104, (2) (2013) 154–171.
[40] D. Yao, L. Jia, M. Li, Y. Qin, W. Peng, G. Liu, S. Pang, Harmonic wavelet envelope method applied in railway bearing fault diagnosis, J. Eng. Sci. Technol. Rev. 6 (2) (2013) 24–28.
[41] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.
[42] S. Jetley, N.A. Lord, N. Lee, P.H. Torr, Learn to pay attention, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, arXiv:180402391.
[43] W. Liu, A. Rabinovich, A.C. Berg, Parsenet: Looking wider to see better, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, arXiv:150604579.
[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[45] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual Attention Network for Image Classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164.
[46] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene CNNS, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, arXiv:1412.6856.
[47] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid Scene Parsing Network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.
[48] O. Oktay, J. Schlemper, L.L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N.Y. Hammerla, B. Kainz, et al., Attention u-net: learning where to look for the pancreas, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, arXiv:180403999.
[49] S. Lin, P. Hui, Where's your focus: Personalized attention, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, arXiv:180207931.
[50] L.C. Chen, Y. Yang, J. Wang, W. Xu, A.L. Yuille, Attention to Scale: Scale-aware Semantic Image Segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 640–3649.
[51] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected CRFS, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, arXiv:14127062.
[52] J. Chen, K. Li, Q. Deng, K. Li, PS. Yu, Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing, IEEE Transactions on Industrial Informatics (2019) arXiv:190406400.

Xiao Fu received the bachelor degree of engineering from Central China Normal University, China, in 2013. She is currently a M.D. candidate in Computer Science, Hunan University, China. Her research interest includes deep learning, computer vision.

Kenli Li received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2003. He is currently a Full Professor of Computer Science and Technology with Hunan University, Changsha, China. He has published more than 200 research papers in international conferences and journals such as IEEE-TC, IEEE-TPDS, and ICPP. His current research interests include parallel computing, cloud computing, and big data computing. He is an outstanding member of CCF. He serves on the editorial board of the IEEE-TC.

Jing Liu received her B.S. degree from the college of mathematics and econometrics and Ph.D. degree from the college of information science and engineering, Hunan University, Changsha, China, in 2009 and 2015, respectively. She is a lecturer in the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. Her current research interests include scheduling, fault tolerance, embedded systems, real-time systems, edge computing.

Keqin Li is a SUNY Distinguished Professor of computer science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multi-core computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber physical systems. He has published over 440 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, Journal of Parallel and Distributed Computing. He is an IEEE Fellow.

Zeng Zeng received the PhD degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2005, and the BS and MS degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1997 and 2000, respectively. Currently, he works as a scientist III in Data Analytics Department, I2R, A*STAR, Singapore. From 2011 to 2014, he worked as a senior research fellow with the National University of Singapore. From 2005 to 2011, he worked as an associate professor in Computer and Communication School, Hunan University, China. His research interests include distributed/parallel computing systems, data stream analysis, deep learning, multimedia storage systems, wireless sensor networks, and onboard fault diagnosis.

Cen Chen is currently a Ph.D. candidate in Computer Science, Hunan University, China. His research interest includes parallel and distributed computing systems, machine learning on big data. He has published several research articles in international conference and journals of machine learning algorithms and parallel computing.