Roll contact fatigue defect recognition using computer vision and Deep Convolutional Neural Networks with transfer learning

Baoling LIU[1,2], John C. Brigham[2*], Jun HE[3], Xiaocui Yuan[1], Huiling HU[1]

1. Nanchang Institute of Technology, Nanchang, 330099, China
2. Department of Engineering, Durham University, Durham, DH1 3LE, United Kingdom
3. State Grid Jiangxi Electric Power Research Institute, 330096, China

Abstract: An end-to-end machine learning approach for classifying rolling contact fatigue (RCF) defects utilizing defect images is presented and evaluated. The core component of this approach is the use of a fine-tuned AlexNet architecture (FT-AlexNet), which is a well-known pre-trained deep Convolutional Neural Network (DCNN). Through comparing the FT-AlexNet method with two classical two-step classification methods that include a feature extraction step and then train a classifier, it was found that the FT-AlexNet could not only avoid the need of additional steps and variability involved in selection of feature extraction methods and classification strategies and parameters, but also obtain the comparatively better classification accuracy and generalization ability. In addition, the 'black box' working principle of FT-AlexNet was analyzed through visualization, which displayed its robustness to noise and background interference to some degree. However, it was also found that the FT-AlexNet architecture, although improved compared to the more traditional methods, was not as accurate for the identification of micro defects for cases with substantial variation in the image background.

Keywords: RCF defect classification, Pre-trained deep Convolutional Neural Networks (DCNN), AlexNet, transfer learning, visualization technology

1. Introduction

At present, rolling contact fatigue (RCF) defects, occurring on rail surfaces and near surfaces, has become one of the main causes of derailment accidents. RCF defects mainly manifest as three types: (1) head check, (2) shelling, and (3) squat [1]. As these different types of RCF defects have different degrees of risk and significance of outcomes, accurate classification and identification is critical for establishing reasonable maintenance methods and ensuring safe operation of trains [2].

Existing rail defect detection technologies include ultrasonic testing, magnetic particle testing, electromagnetic testing and Computer vision inspection (CVI) [3-7]. Among them, due to its advantages of non-contact, high speed and high precision，CVI is becoming a more prominent tool for recognizing the rail surface defects with the improvements of computer performance and signal processing technology [7]. CVI-based rail defect recognition/classification processes usually consist of two steps: feature extraction and classifier training, as shown in Figure 1. In general, one major challenge with the two-step approach is that it is often difficult to know *a priori* about what features would be reliable predictors and no model will be a successful classifier without the appropriate features to work with [8-9]. Moreover, the feature extraction often is a costly and cumbersome trial and error process. [10-11]

In principle, an ideal method for defect classification would be end-to-end, rather than two distinct steps, in hopes that potential information could be directly extracted from the images and used to guide the classification process. Recently, deep learning (DL) neural network architectures have been developed and applied to achieve end-to-end systems that can automatically learn and apply features from raw inputs [12]. The traditional two-step method is contrasted with an end-to-end method in Figure 1.

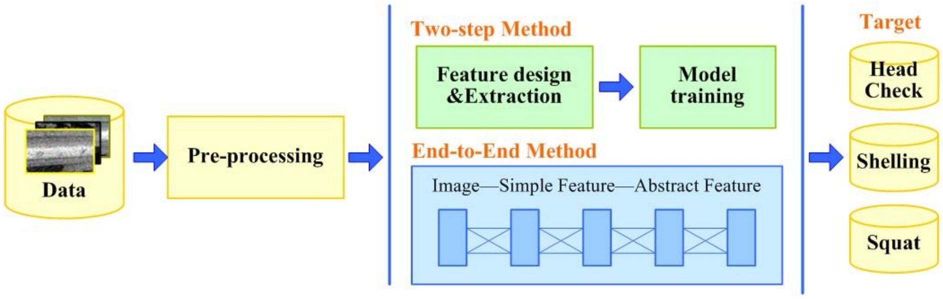


Figure 1. The two-step method for defect classification compared to the end-to-end method

In recent years, DL has been adopted in various application areas for machine learning, such as automatic natural language processing, acoustic modeling and bioinformatics, due to its end-to-end convenience and promising results in terms of accurate system representation [13-15]. However, DL has been rarely applied in the engineering field for defect recognition thus far. One challenge is that defect datasets are normally too small to train DL networks. Normally, a relatively large amount of data is required for DL methods to avoid generalization issues from overfitting. However, collecting defect images from rails is difficult because of the low occurrence of these defects. The second challenge is that the training process for DL can be excessively time consuming. Moreover, DL networks (and similar tools) are generally regarded as black box tools, because it is difficult to interpret the model due to the large number of neurons, complicated structures, and non-linear transfer functions [16].

Transfer learning (TL) methods based on deep convolutional neural network (DCNN) have been shown to be a potential solution to the problems associated with relatively small datasets, inaccuracy/overfitting, and the computational cost of training [17-18]. TL is an approach to DL network training, in which a model that has already been trained for a relatively similar task is reused for the desired task. The network only needs to be "fine-tuned" according to the desired task, which often leads to both accurate representation and improved generalization ability, while significantly reducing the training cost [19]. Additionally, visualization analysis methods for DCNN can provide an effective tool for exploring the black box processes of DCNN and lead to the ability to rationally design the DCNN structure [20-21].

This paper presents an end-to-end approach for RCF defect recognition utilizing DCNN. In particular, the strategy presented applies a pre-trained DCNN based on image data from a different domain, taking advantage of the TL capability. This approach for end-to-end RCF defect recognition realized by DCNN with TL is briefly discussed in the following Section 2. In Section 3, the proposed method is verified and compared with alternate two-step classification strategies in terms of recognition accuracy and generalization capability. The ability of DCNN to extract significant features relating to RCF defects is also evaluated using visualization techniques. Lastly, conclusions and potential further research directions are provided in Section 4.

2. Methods

2.1 Deep Convolutional Neural Networks (DCNN)

DCNN are a type of feed-forward neural network widely applied for image-related analysis with end-to-end learning, that is, learning/classifying from the original data without any *a priori* feature selection. DCNN mainly consist of at least three types of layers: 1) convolutional layers; 2) pooling layers; and 3) fully connected layers. [18] .

The convolutional layers consist of a series of fixed-size filters, which perform convolution on the image data to highlight some patterns, such as edges and position, used to characterize images. A non-linear excitation function, defined as rectified linear unit (ReLU) is often imposed after convolutional layer to facilitate faster training of a DCNN. The output of the convolutional layer can be calculated as follows:

$$V_j^r = \varphi\left(\sum_{i=1}^{S} V_i^{r-1} * \left(W_{ij}^r + b_j^r\right)\right) \tag{1}$$

where, V_j^r represents the jth output feature map on the rth convolution layer. S is the filter size. $*$ denotes an operator of the convolution. V_i^{r-1} is the ith input feature map on the convolution layer $r-1$. W_{ij}^r represents the convolutional kernel represents the ith band of the jth filter. b_j^r denotes as the bias of the jth feature map. φ is an activation function applied to the result, which is also defined as rectified linear units (ReLU). ReLU can be defined as:

$$Re\,LU = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases} \tag{2}$$

The pooling layers are normally applied after the convolutional layers to reduce the feature dimensions and to avoid overfitting problems. Pooling layers perform down-sampling operations by sliding windows across the feature maps and apply linear or nonlinear operations, such as calculating the local mean or maximum values. Max pooling is most commonly used in CNN and is written as:

$$P_j^m = \max_{k=1}^{p}\left(V_j^{(m-1)\times n+k}\right) \tag{3}$$

where, P_j^m represents the jth output feature map of mth band pooling layer. p and n are the pooling size and sub-sampling factor.

Fully connected (FC) layers are used to interpret patterns generated by the previous layers. For classification problem, the *softmax* function is the common last layer. The cross-entropy between the estimated softmax output probability distribution and the target class probability distribution is selected as the loss function.

2.2 Fine-tuned AlexNet Architecture

The AlexNet architecture was utilized and fine-tuned (taking advantage of transfer learning) for the present study on RCF defect recognition. AlexNet is a pretrained DCNN whose architecture and parameters have been trained using a large-scale annotated natural image dataset. The architecture of AlexNet includes 5 convolutional layers, 3 pooling layers and 3 FC layers, as shown in Figure 2. The last FC layer is a classifier including 1000 nodes connecting to 1000 classes and the rest of the architecture can be considered as a feature extractor. The input data used for AlexNet is RGB images with size of 227 × 227 ×3pixels [22].

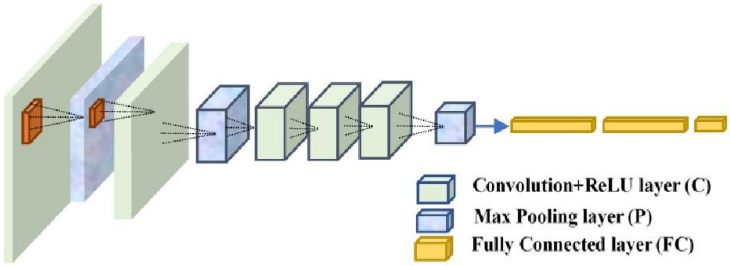


Figure 2. The architecture of the DCNN AlexNet

The fine-tuning for the desired application can then be realized by replacing the last FC layer and retraining the parameters of each layer, producing a fine-tuned AlexNet (FT-AlexNet). The process of producing the FT-AlexNet is shown in Figure 3.

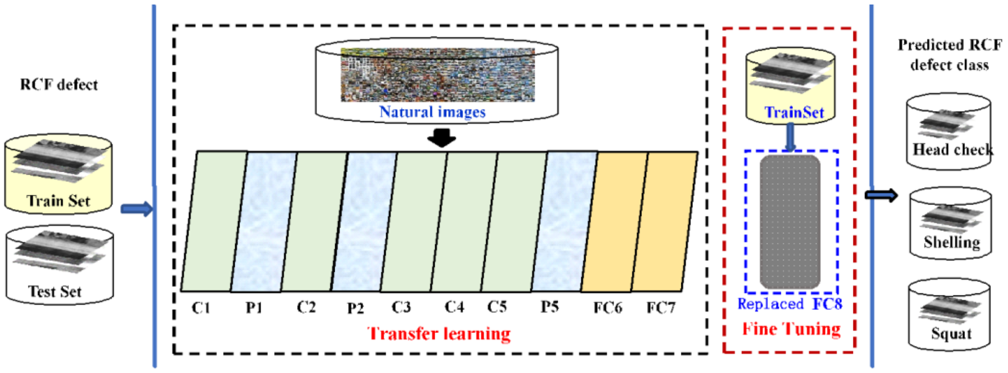


Figure 3. The principle of transfer learning and fine-tuning using AlexNet

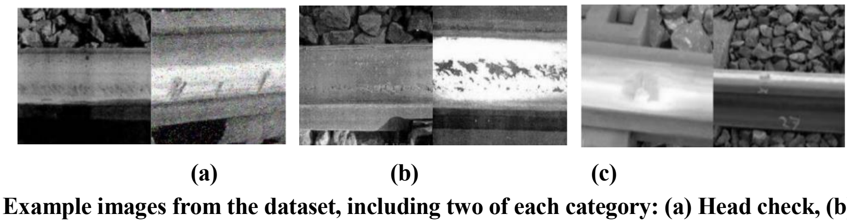
It can be seen that the last FC layer (named as FC8 in Figure 3) is replaced by a FC layer with 3 neurons, corresponding to the three RCF defect types, which takes the extracted features from previous layers and maps them into the three classes of RCF defects. The fine-tuned network retains the same architecture as the pre-trained network, except for the replaced layer. The network will learn the new mapping as well as refine the feature extraction by fine-tuning the parameters to be slightly more specific to the application after training with the new training set. There are a number of popular DCNN frameworks. In this work, the Matlab Deep Learning Toolbox was used to implement a FT-AlexNet [23].

3 Result and discussion

3.1 Dataset

Imaging data available for detecting RCF defects in actual practice is often very different than those from laboratory tests due to significant uncontrollable variations, including the illumination, potential tread reflections and other unpredictable interferences. So, in order to ensure the results herein have practical significance, an initial set of 70 images of RCF defects published in [24], comprised of 27 head check, 15 shelling and 28 squat defects, that were acquired from field images of actual rails, with natural variations in lighting and background was utilized. In order to further simulate the influence of noise and increase the amount of data, the final dataset was formed by replicating the original 70 images 8 times and adding different levels of noise to each replicated image. Specifically, randomly generated “Salt & Pepper” noise was used with intensity of 0.05. The dataset was reviewed manually to ensure each image could be discerned by a human operator and too much noise was not present (i.e., to ensure the images would be acceptable in practice). 24 images were deemed to be indiscernible and were removed. Therefore, 606 images with RCF defects were used to evaluate the performance of the FT-AlexNet in this study.

It should be noted that this was not a particularly large dataset for an application of a DCNN for classification, which further emphasized the need for the transfer learning property utilized. Figure 4 shows two examples each for the three defect types from the image dataset. Of note was that even within the same defect category there were significant differences in appearance, in large part due to the variations in lighting and background in the dataset.



(a) (b) (c)

Figure 4. Example images from the dataset, including two of each category: (a) Head check, (b) Shelling, and (c) Squat.

3.2 RCF Defect Classifiers

As discussed in Section 2, the FT-AlexNet was applied to create an end-to-end classifier to recognize RCF defect. To compare the capabilities of the end-to-end DCNN approach to more traditional methods, common two-step methods (following the approach shown in Figure 1) were also applied to classify the RCF defects using the same datasets. The Histogram of oriented gradients (HOG) approach was used to extract features, and two different types of classifiers were trained using these HOG features: (1) random forest classifiers (RFs), (2) support vector machine classifiers (SVMs). For the RFs, the numbers of trees were varied from 10 to 3000. The kernel functions used for the SVMs were ‘linear’, ‘polynomial’, and ‘Gaussian’, and the orders of the ‘polynomial’ considered were 2, 3, and 5. Considering the SVM's sensitivity to parameters, the optimal parameters of the SVMs were set through heuristic search [25].

Besides the above-mentioned two-step methods, several well-known pretrained DCNN were applied to recognize the RCF defects and compared with FT-AlexNet in terms of accuracy and time cost.

3.3 Classification Results

3*.3.1 Initial Comparison of Classification Methods*

The dataset of 606 images was first broken arbitrarily into two mutually exclusive datasets: a training set of 389 images and a testing set of 217 images, and Table 1 shows the distribution of each defect category in these image sets. As normal, the network was trained using the training set, and then the generalization capability was evaluated by applying the network to predict the classes for the unseen testing set.

Table 1. Distribution of each defect category in the RCF defect image dataset and division into training and testing sets for the initial test.

	Head check	Shelling	Squat	Total
Total	237	133	236	606
Training Set	159	82	148	389
Testing Set	78	51	88	217

The FT-AlexNet approach and each of RF and SVM methods with HOG features were applied as detailed previously. In addition, to examine the features that the FT-AlexNet approach extracts internally and how they develop through the various layers the outputs of the six internal layers prior to the classification layer (FC8) were individually used as features and combined with the RF and

SVM classification strategies to predict the RCF defect types. Specifically, in the order that they occur in the FT-AlexNet architecture, the outputs from the first Pooling layer (Pool1), the second Pooling layer (Pool2), the third ReLU layer (ReLU3), the fourth ReLU layer (ReLU4), the fifth Pooling layer (Pool5) and the seventh FC layer (FC7) were considered, in turn, as features. The classification accuracies were calculated as the ratio between the number of correct predictions and the total number of samples in the testing set. Table 2 shows the accuracy for each combination of features and classifier, as well as the end-to-end FT-AlexNet.

Table 2. The accuracies of each classification strategy with respect to the combination of feature vectors and classifier parameters for the initial test.

	Parameter setting	HOG	Pool1	Pool2	ReLU3	ReLU4	Pool5	FC7
RF	10	52.15	79.75	80.36	76.56	76.93	77.91	76.68
	50	55.82	85.52	86.25	84.41	83.19	84.41	82.21
	300	55.82	84.17	83.55	85.40	83.31	85.52	79.75
	1000	58. 90	83.43	82.41	84.29	84.66	86.09	83.84
	3000	57.96	83.43	82.20	85.40	83.31	86.09	83.64
SVM	linear	77.30	71.78	87.12	78.52	91.41	88.34	90.80
	Gaussian	76.07	38.65	38.65	71.60	72.39	87.11	92.02
	polynomial(2)	77.91	87.73	88.95	91.41	90.79	87.73	90.79
	polynomial(3)	76.07	60.74	89.57	90.79	90.79	88.95	91.41
	polynomial(5)	77.30	71.78	88.34	88.34	84.05	85.27	91.00
FT-AlexNet	—	—	—	—	—	—	—	94.47

The FT-AlexNet achieved the highest recognition accuracy compared with all of the variations of the two-step approach considered. The accuracy of the SVM classifiers was generally higher than the RF classifiers, but still about 2.5% less accurate for the best case compared to the FT-AlexNet accuracy, and almost 20% less accurate than the FT-AlexNet when using the HOG features. There was some variation depending on the RF and SVM classifier parameters, particularly the SVM cases when using the FT-AlexNet layer output as features. The analysis using the FT-AlexNet internal layer outputs as features highlights a likely major reason for the high accuracy of the FT-AlexNet, which is that the FT-AlexNet can extract natural features that were more easily related to the defect class than the HOG features. This was particularly clear for the RF classifiers, which all increased in accuracy by more than 20% when switching from HOG to the FT-AlexNet internal layer outputs as features. The accuracy of the RF classifiers did not increase significantly as the information was propagated deeper into the network, whereas the SVM classifiers produced substantially more accurate classification results when using the latter FT-AlexNet layer outputs as features.

Figure 5 shows the progression of the accuracy and loss of the FT-AlexNet during training for one of the experiments. Both accuracy and loss with respect to the training set and the unseen testing set are shown. Of particular importance, the testing accuracy reached approximately 95% in this case, which indicates overfitting had not occurred.

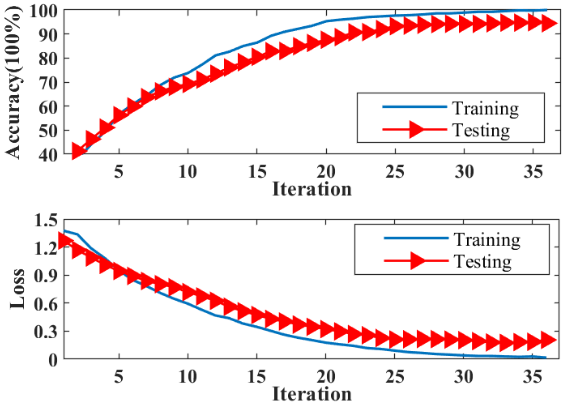
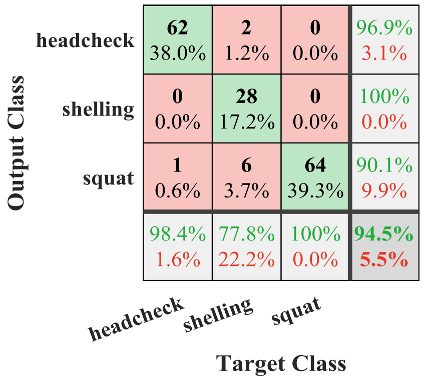


Figure 5. Accuracy and loss with respect to the training and testing sets as FT-AlexNet was trained for the initial test

The confusion matrix for the classification results is shown as Figure 6. The FT-AlexNet recognition accuracies for each type of defect were over 90%, and the accuracy in identifying squat defects was 100%, despite the effects of light, background and noise. Thus, the end-to-end FT-AlexNet method can circumvent challenges of more traditional two-step approaches in terms of feature design and classifier parameter selection, while potentially significantly improving classification accuracy as well.


Output Class
Target Class

	headcheck	shelling	squat	
headcheck	62 38.0%	2 1.2%	0 0.0%	96.9% 3.1%
shelling	0 0.0%	28 17.2%	0 0.0%	100% 0.0%
squat	1 0.6%	6 3.7%	64 39.3%	90.1% 9.9%
	98.4% 1.6%	77.8% 22.2%	100% 0.0%	**94.5%** **5.5%**

Figure 6 Confusion matrix for AlexNet for the initial test.

Alexnet is an early pretrained DCNN, and the subsequent pretrained DCNNs adopt a deeper or broader network architecture in order to improve the classification accuracy for specific problems. Table 3 shows the recognition results of FT-AlexNet and several well-known pretrained DCNNs on RCF defect. The same training options and parameters of the replaced FC layer were set for each DCNN. In order to ensure a faster learning speed, a larger learning rate is given for the replaced FC layer. At the same time, in order to avoid under-fitting problems due to less training set data, a small learning rate is set for other layers during the fine-tuning process. In this experiment, the learning rate of the replaced FC layer was 20, and the learning rate of the other layers was 10^{-4} .

As can be seen from Table 3, the FT-AlexNet had advantages in terms of accuracy and time cost for recognizing RCF defect in this study. The result is likely because the architecture of AlexNet is relatively straightforward and not too large. Furthermore, AlexNet is trained on more than a million natural images and has learned rich feature representations for a wide range of images. However, when choosing a DCNN architecture, it is always necessary to make compromises based on characteristics such as accuracy, speed, and size for specific problem considered.

Table 3 also shows the comparison of the FT-AlexNet with the traditional AlexNet, which is written as AlexNet (Freeze) in Table 3. Here, the so-called traditional AlexNet refers to the AlexNet architecture of only retraining the parameters of the replaced FC layer and keeping the parameters of the other layers unchanged. The traditional AlexNet had a faster speed because it does not need to readjust the parameters of others layer. FT-AlexNet is slower, but it is often more accurate due to extracting better features for the particular application. The computer used herein was a standard desktop PC with Intel Core i5-7300CPU and 8GB of RAM, and all tests utilized only one CPU.

Table 3. Comparison of several pre-trained DCNN

Network	Depth	Size	Parameters (Millions)	Image Input Size	Accaracy（%）	Time（s）
Vgg16	16	515MB	138.0	224-by-224	87.10	6500
googlenet	22	27MB	7.0	224-by-224	92.63	3084
Inceptionv3	48	89MB	23.9	299-by-299	80.18	11884
Resnet50	50	96MB	25.6	224-by-224	86.17	7857
AlexNet（Freeze）	8	227MB	61.0	227-by-227	85.23	477
FT-AlexNet	8	227MB	61.0	227-by-227	94.47	501

One important note is that there is a stochastic component to these methods, and if the processes are repeated the resulting accuracies will change. To address this issue the following section more thoroughly evaluates the consistency of each approach through repeated trials of each test and more controlled divisions of the training and testing datasets.

3.3.2 Reduced Training Set Size

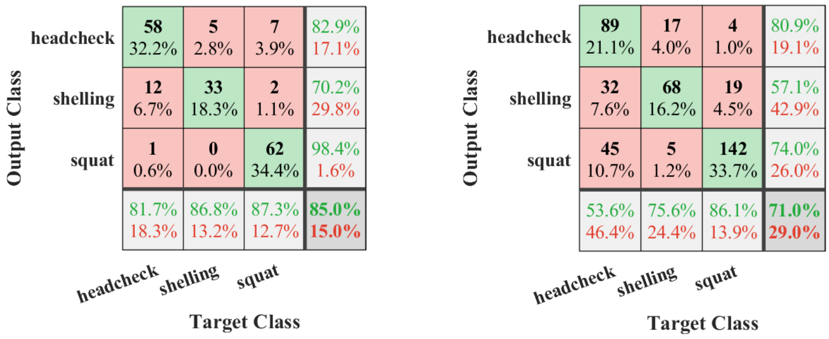
Although the FT-AlexNet approach showed the highest classification accuracy in the initial tests, one particular criticism of deep learning is the relatively large amount of training data that is often needed for accurate generalization in contrast to two-step approaches. Therefore, to evaluate the generalization, the FT-AlexNet and the two-step classifiers were again applied to build RCF defect classifiers using decreasing training data. Additionally, the dataset listed in Table 1 was reshuffled, and grouped with respect to the original 70 images, such that there was no overlap between the training set and testing set. Six tests were performed, starting with 80% of the dataset as the training set and reducing this to 30% of the dataset at 10% intervals. For these tests, only the best performing two-step classification methods using the HOG features from the initial tests were considered, SVM with the quadratic polynomial kernel function and the RF with 1000 trees. The classification accuracies of the three classifiers are shown in Table 4. The classification accuracies were the average from 10 repetitions of the experiments. Additionally, the time cost in terms of the total time to train and test a single instance of each classifier is also shown in Table 4 to intuitively compare the speed of the three classifiers.

Table 4. The accuracies of each classification strategy with respect to the percentage of the dataset used for training.

	80%		70%	60%	50%	40%	30%	
	Accuracy	Time(s)	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Time(s)
HOG+SVM(Poly2)	53.33	1183	50.00	44.38	44.33	44.81	43.94	461
HOG+RF(1000)	62.58	307	52.22	51.75	50.76	48.22	46.79	244
FT-AlexNet	92.50	779	87.78	86.58	77.73	71.69	68.64	368

As the training set decreased from 80% to 30%, the classification accuracies of all classifiers decreased in general. The FT-AlexNet again achieved the highest recognition accuracy compared with the two-step approaches considered. Moreover, the relative improvement through FT-AlexNet compared to the two-step methods was substantially higher than the improvement seen in the initial tests, with a maximum difference of more than 30% and minimum difference of more than 20%, depending on the size of the training set. A likely reason was that the FT-AlexNet classifiers were more capable of deriving the actual pattern from the physical differences in the defect types and generalized better. It can also be seen from Table 4 that the accuracy of SVM is significantly lower than that of Table 2. This is because the test set corresponding to Table 4 contained unfamiliar data caused by the original image that does not overlap the training set. However, the accuracy of RF has not decreased significantly. On the contrary, because of the higher number of training samples, the accuracy of RF shown in Table 4 is higher than that in Table 2. This is partly because that RF has the advantages of handling high-dimensional data and being insensitive to parameter settings, which makes it possible to obtain better accuracy than SVM when the testing set contains a large number of unfamiliar samples. Although the minimum training set size of 30% (181 images) is relatively small for applications of DCNN architectures, the fact that the FT-AlexNet still maintained the highest accuracy emphasizes the benefits of using pre-trained models. In terms of the computing cost (i.e., total computing time shown in Table 4), that of the FT- AlexNet approach was between the two two-step methods. Thus, the computing cost of utilizing such a pre-trained DCNN is not significantly different than more traditional approaches.

Figure 7 again shows the confusion matrices for one trial each of the FT-AlexNet classifier trained using 70% and 30% of the total training set. As the training set was reduced from 70% of the total data to 30%, the ability to correctly classify the three types of RCF defects clearly decreased. The head check defect became particularly difficult to identify as the training set size decreased. The accuracy of the head check case is somewhat surprising as there were considerably fewer shelling cases in the training set in comparison, and yet shelling was more accurately classified. This result highlights that it is not just the amount of data that affects the classification capability, and the ease with which a specific defect type can be discerned contributes significantly as well.


Output Class
Target Class
Output Class
Target Class

	headcheck	shelling	squat	
headcheck	58 32.2%	5 2.8%	7 3.9%	82.9% 17.1%
shelling	12 6.7%	33 18.3%	2 1.1%	70.2% 29.8%
squat	1 0.6%	0 0.0%	62 34.4%	98.4% 1.6%
	81.7% 18.3%	86.8% 13.2%	87.3% 12.7%	**85.0%** **15.0%**

(a)

	headcheck	shelling	squat	
headcheck	89 21.1%	17 4.0%	4 1.0%	80.9% 19.1%
shelling	32 7.6%	68 16.2%	19 4.5%	57.1% 42.9%
squat	45 10.7%	5 1.2%	142 33.7%	74.0% 26.0%
	53.6% 46.4%	75.6% 24.4%	86.1% 13.9%	**71.0%** **29.0%**

(b)

Figure 7. Confusion matrix for FT-AlexNet trained with (a) 70% and (b) 30% of the total dataset.

3.4 Network Visualization

Although initially DCNNs were generally applied as blackbox tools, more recently, work has shown that DCNN processes can be visualized using aspects such as the activations produced by each layer of a trained network [26]. Thus, this visualization approach can be used to at least in-part explain the effectiveness of the FT-AlexNet classifiers evaluated in the previous sections. For example, it has been shown that with the deepening of the architecture, the features extracted by a DCNN become more inclined to abstract information. For the case of defect or object detection applications such as the one herein, the first convolutional layer of FT-AlexNet usually extracts information such as the outline and color of the objects in the image, whereas the fifth convolutional layer can extract characteristics such as the position and texture of the object.

In AlexNet, the size of the i^{th} activated layer is written as $r_i \times r_i \times n_i$, where n_i is the channel number and each channel corresponds to a separate $r_i \times r_i$ grayscale image that contains activation information, generally known as receptive field. For convolutional layers and pooling layers, the output size of the receptive field can be determined by the following relationship:

$$r_{i+1} = \frac{\left(r_i - k_i + 2 \times p_i\right)}{s_i} + 1 \tag{4}$$

r_i and r_{i+1} indicate the width (and height) of the input and output receptive fields, respectively. k_i, s_i and p_i are the kernel size, stride and padding, respectively. The parameters of convolutional layers and pooling layers used for AlexNet are shown in Table 5 [30]. Figure 8 shows the activation-based visualization results for an arbitrarily chosen squat defect image, including the dimensions of each layer output (note that the size of input image is $227 \times 227 \times 3$).

Table 5. Parameters for each layer of AlexNet.

	Conv1	Pool1	Conv2	Pool2	Conv3	Conv4	Conv5	Pool5
k_i	11	3	5	3	3	3	3	3
s_i	4	2	1	2	1	1	1	2
p_i	0	0	2	0	1	1	1	0
n_i	96	96	256	256	384	384	256	256

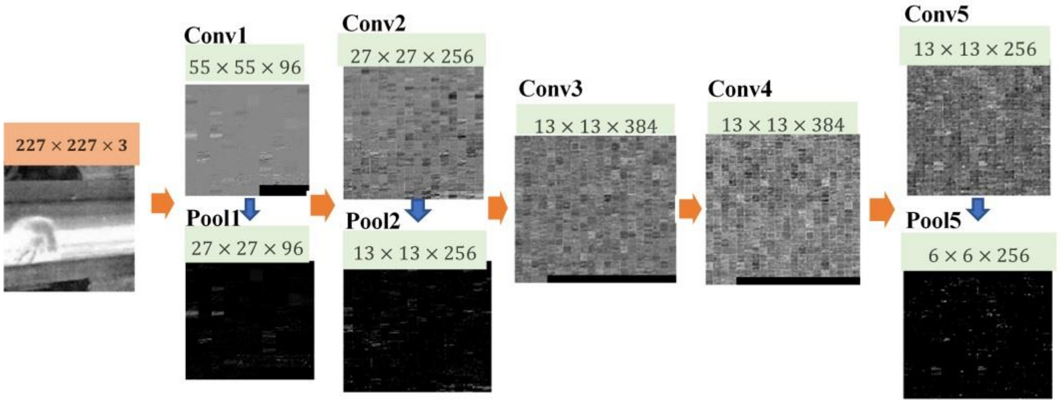
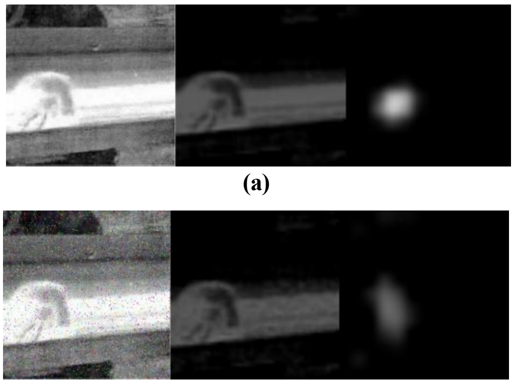


Figure 8. Visualization of every convolutional layer for AlexNet processing an arbitrarily chosen squat defect image.

In the following, the outputs of the **FT-AlexNet** classifiers for RCF defects are examined to explore the type of features that are highlighted within the different layers of the network and how these features are affected by variations in noise and other image aspects, such as background.

3*.4.1 Robustness to Noise*

Figure 9(a) shows the original squat defect image and the subgraphs from the 43rd channel of the 1st convolutional layers (Con1) and the 199th channel from the 5th convolutional layer (Con5). It can be seen that Con1 extracted the outlines of the rail head and the defect, excluding the rest of the image, while Con5 further refined the extraction down to the position and the general shape of the defect alone. Figure 9(b) shows a noisy version of the same original squat defect image and the subgraphs from same channels of the same layers. Even in the presence of noise, FT-AlexNet was still able to clearly extract the rail head and defect, with those images being nearly identical to the noise-free case, and even though there was some distortion due to the noise, the further refined position and shape of the squat defect remained consistent as well.

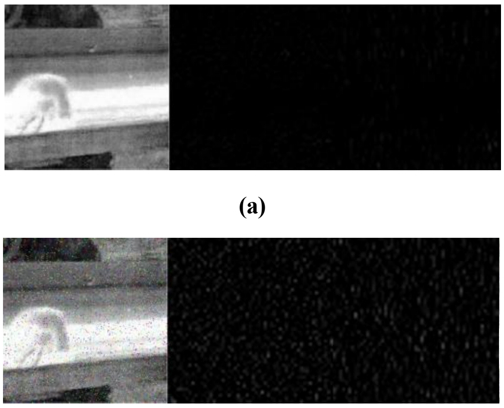


(a)

(b)

Figure 9. From left to right – the input image and the subgraphs from the 43rd channel of Con1 and the 199th channel of Con5 produced by the FT-AlexNet for an example squat defect image (a) original image (without noise) and (b) noisy image

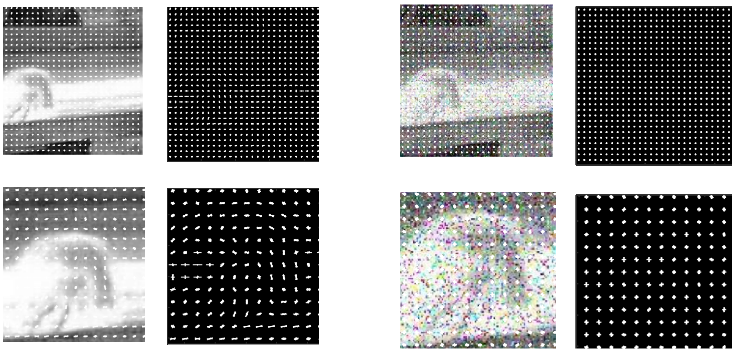
Figure 10 shows subgraphs from the 69th and 80th channels of Con1 for the same two input images shown in Figure 9. There is nothing shown in the subgraphs for the original image (i.e. input image without noise), as shown in Figure 10(a), whereas there are speckles distributed in the outputs for the noisy image, shown in Figure 10(b), which follow the same pattern as the salt and pepper noise included in the image. This result shows how FT-AlexNet is capable of extracting/differentiating noise from the images by segregating it to certain channels during the training process. In particular for this example, Con1 was able to distinguish the high frequency information of the white noise, to effectively denoise the signal. Although this denoising process did cause a certain amount of information to be lost, as shown in the output for Con5 in Figure 9(b), the important features of the defect were still able to be seen.



(b)

Figure 10. From left to right – the input image and the subgraphs from the 69th channel and the 80th channel of Con1 produced by the FT-AlexNet for an example squat defect image (a) original image and (b) noisy image.

For comparison with the FT-AlexNet results, Figure 11 shows the features extracted from the same two squat defect images (with and without noise) using the HOG method. The HOG features are sensitive to the presence of the defect for the image without noise, but when noise is included, it is no longer possible to see any clear pattern in the effect of the defect location on the HOG features. Although the HOG method has been shown to have some robustness against light changes, it is clearly limited in the presence of substantial noise.



(a) (b)

Figure 11. Input image with the HOG features (top left), HOG features alone (top right), close-up view of the defect region in the image with the HOG features (bottom left) and close-up view of the HOG features in the region of the defect (bottom right) for an example squat defect image (a) original image and (b) noisy image.

3.*4.2 Robustness to Background Changes*

There was significant variation in the dataset used herein with respect to how much of the image was part of the rail being inspected and how much was background. In general, the outputs of the convolutional layers of images with a larger portion of background (i.e., non-rail portions of the image) were substantially more complex. However, the classification results showed that the pre-trained network was robust to the interference from these variations in the amount of background, as shown through the high classification accuracy. To explore an example with a significant portion of the image as background, Figure 12 shows the 133rd channel of Con1 and its subsequent ReLU layer (ReLU1) for an example squat defect image (different than that used in the previous section) without noise. The 133rd channel of Con1 still shows much of the original image, including both rail and background. However, the ReLU1 output clearly highlights the rail tread and the squat defect, separating these from the background, even though it is a relatively smaller portion of the total image.

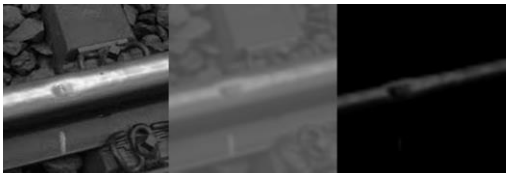


Figure12. From left to right – the input image and the subgraphs from the 133rd channel of Con1 and ReLU1 produced by the FT-AlexNet for an example squat defect image.

To examine the latter layers of the network, Figure 13 shows the output of 6 different channels of the 5th convolutional layer (Con5) for the squat defect example. These outputs show that there are several different channels that identify different localized features relating to the position of the rail and the different components of the rail image, such as the defect, roadbed, rail web, and fasteners. However, the output of Con5 is substantially more abstract than that shown for Con1, which is expected as the information propagates further into the network. Note that although Con5 is the last convolutional layer, the features would be further refined for the final classification by the following three fully connected layers, but this process is even more abstract than that for the convolutional layers and visualization is not tractable.

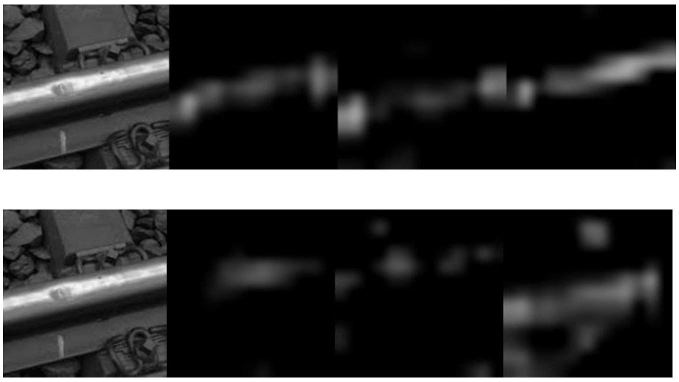


Figure 13. From left to right – (top) the input image and the subgraphs from the 14th, 50th and 168th channels of Con5 and (bottom) the input image and the subgraphs from the 33rd, 62nd and 133rd channels of Con5 produced by the FT-AlexNet for an example squat defect image.

The examples visualized so far have all been squat defect cases, but the various convolutional layer operations (i.e., filters) have different sensitivities to different features, which will ideally be dependent upon the type of defect in a given image. To examine a different defect type, Figures 14 and 15 show the outputs of various channels of Con1 and ReLU1 for two example head check defect images, one with minimal non-rail background and one with significant background. In Figure 14, the 43rd channel is shown to extract the rail tread profile, while the 85th channel is shown to extract the profile of the head check alone. In Figure 15, the 70th channel is shown to discern the dense head checks on the tread surface, while the 90th channel recognizes the contours of the head checks also, but also clearly includes the shape of the hand in the background. However, as could be seen from Figure 15, since the head check is small and the background (hand) is a large proportion in the image, it is not surprising that portions of the defect information extracted by Con1 are contaminated (i.e., include information unrelated to the defect to be classified). The result of such contamination can be seen in the confusion matrices (Figures 6 and 7), in which the head check cases were found to be the least accurately classified overall. Thus, background variations can have a significantly negative influence on the classification capability of such a DCNN approach, particularly if the defect to be identified is relatively small, as is the case for head checks.

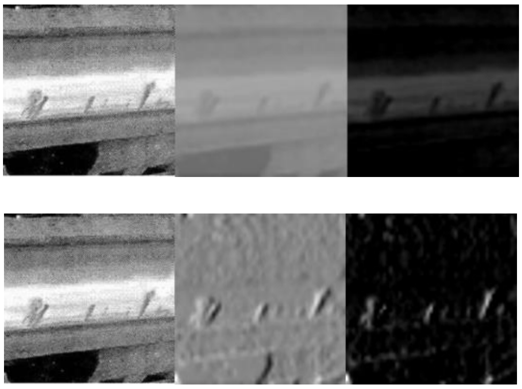


Figure 14. From left to right – (top) the input image and the subgraphs from the 43rd channel of Con1 and ReLU1 and (bottom) the input image and the subgraphs from the 85th channel of Con1 and ReLU1 produced by the t FT-AlexNet for an example head check defect image and minimal non-rail background.

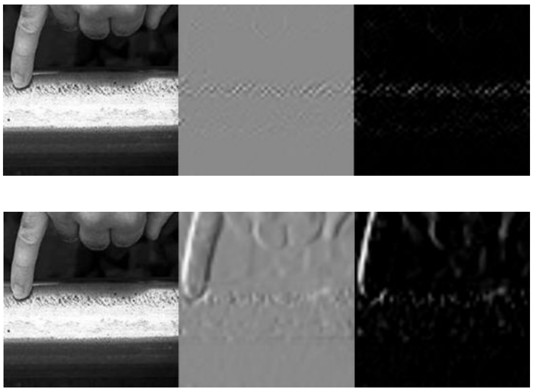


Figure 15. From left to right – (top) the input image and the subgraphs from the 70th channel of Con1 and ReLU1 and (bottom) the input image and the subgraphs from the 90th channel of Con1 and ReLU1 produced by the FT-AlexNet for an example head check defect image and substantial non-rail background.

4 Conclusions

This paper presented and evaluated a novel end-to-end method for classification of RCF defects on rail surfaces based on FT-AlexNet pretrained DCNN. Using the concept of transfer learning, it was only necessary to replace the last FC layer and to fine tune the weight parameters of each layer by using the RCF defect data. Through transfer learning, a large amount of training time was saved, while maintaining the classification accuracy. This method was compared with two classical two-step methods that rely on HOG features and several well-known pretrained DCNN as well. FT-AlxNet not only had better accuracy and generalization capabilities than these two-step methods, but also had advantages in terms of accuracy and time in comparison with other pretrained networks. The classification process of the FT-AlexNet was analyzed using visualization of layer outputs, which showed how the network naturally extracted features relating to the defects and how these features are robust with respect to noise and background interference to some extent. Further work will focus on how to improve the architecture and parameters of pretrained DCNN in order to increase the recognition accuracy of micro-defect with a large amount of background interference.

Acknowledgement

The authors would like to thank the Jiangxi Province Education Department (Grant No.GJJ161104), Jiangxi provincial department of science and technology (Grant No. 20171BAB206037) and National Natural Science Foundation of China (Grant No. 61903176)for their support.

Reference

[1] Ph Papaelias M, Roberts C, Davis C L. A review on non-destructive evaluation of rails: state-of-the-art and future development[J]. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and rapid transit, 2008, 222(4): 367-384.

[2] Jessop C, Ahlström J, Hammar L, et al. 3D characterization of rolling contact fatigue crack networks[J]. Wear, 2016, 366: 392-400.

[3] Namboodiri G N, Balasubramaniam K, Balasubramanian T, et al. Rail Weld Inspection Using Phased Array Ultrasonics[C]. Review of Progress in Quantitative Nondestructive Evaluation. AIP Publishing, 2013:887-894.

[4] Liu B, Huang P, Zeng X, et al. Hidden defect recognition based on the improved ensemble empirical decomposition method and pulsed eddy current testing[J]. Ndt & E International, 2016, 86:175-185.

[5] Antipov A G, Markov A . 3D simulation and experiment on high speed rail MFL inspection[J]. NDT & E International, 2018, 98: 177-185.

[6] Rowshandel H, Nicholson G L, Shen J L, et al. Characterisation of clustered cracks using an ACFM sensor and application of an artificial neural network[J]. NDT & E International, 2018, 98: 80-88.

[7] Karakose M, Yaman O, Murat K, et al. A New Approach for Condition Monitoring and Detection of Rail Components and Rail Track in Railway[J]. International Journal of Computational Intelligence Systems, 2018, 11(1): 830-845.

[8] Liu B, Hou D, Huang P, et al. An improved PSO-SVM model for online recognition defects in eddy current testing[J]. Nondestructive Testing and Evaluation, 2013, 28(4): 367-385.

[9] Liaw A, Wiener M. Classification and regression by random Forest[J]. R news, 2002, 2(3): 18-22.

[10] Yuan X, Wu L, Peng Q. An improved Otsu method using the weighted object variance for defect detection[J]. Applied Surface Science, 2015, 349: 472-484.

[11] Liu B, Wu H, Su W, et al. Sector-ring HOG for rotation-invariant human detection[J]. Signal Processing: Image Communication, 2017, 54: 1-10.

[12] Cha Y J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks[J]. Computer-Aided Civil and Infrastructure Engineering, 2017, 32(5): 361-378.

[13] Vuddagiri R K, Vydana H K, Vuppala A K. Curriculum Learning Based Approach for Noise Robust Language Identification using DNN With Attention[J]. Expert Systems with Applications, 2018.

[14] Kim Y, Kim M, Goo J, et al. Learning Self-Informed Feature Contribution for Deep Learning-Based Acoustic Modeling[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, 26(11): 2204-2214.

[15] Gharehbaghi A, Lindén M. A Deep Machine Learning Method for Classifying Cyclic Time Series of Biological

Signals Using Time-Growing Neural Network[J]. IEEE transactions on neural networks and learning systems, 2018, 29(9): 4102-4115.

[16] Karlaftis M G, Vlahogianni E I. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights[J]. Transportation Research Part C: Emerging Technologies, 2011, 19(3): 387-399.

[17] Fu Y, Aldrich C. Froth image analysis by use of transfer learning and convolutional neural networks[J]. Minerals Engineering, 2018, 115: 68-78.

[18] Gopalakrishnan K, Khaitan S K, Choudhary A, et al. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection[J]. Construction and Building Materials, 2017, 157: 322-330.

[19] Goodfellow I, Bengio Y, Courville A, et al. Deep learning[M]. Cambridge: MIT press, 2016. p526

[20] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]. European conference on computer vision. Springer, Cham, 2014: 818-833.

[21] Ren R, Hung T, Tan K C. A generic deep-learning-based approach for automated surface inspection[J]. IEEE transactions on cybernetics, 2018, 48(3): 929-940.

[22] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C] Advances in neural information processing systems. 2012: 1097-1105.

[23] https://matlabacademy.mathworks.com/cn

[24] Lixian Xing. Research on defect characteristics and classification of higher speed rails[D]. China academy of railway science. 2008

[25] Liu B, Hou D, Huang P, et al. An improved PSO-SVM model for online recognition defects in eddy current testing[J]. Nondestructive Testing and Evaluation, 2013, 28(4): 367-385.

[26] Yosinski J, Clune J, Nguyen A, et al. Understanding neural networks through deep visualization[J]. arXiv preprint arXiv:1506.06579, 2015.