Hindawi
Journal of Sensors
Volume 2021, Article ID 5554920, 17 pages
https://doi.org/10.1155/2021/5554920

Research Article

A Convolutional Neural Network-Based Classification and Decision-Making Model for Visible Defect Identification of High-Speed Train Images

Zhixue Wang,[1] **Jianping Peng**,[1] **Wenwei Song**,[1] **Xiaorong Gao**,[1] **Yu Zhang**,[1] **Xiang Zhang**,[1] **Longfei Xiao**,[2] **and Li Ma**[2]

[1]*School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China*
[2]*Chengdu Lead Science & Technology Co. Ltd., Chengdu 610091, China*

Correspondence should be addressed to Jianping Peng; adams.peng@swjtu.edu.cn

Received 11 January 2021; Revised 5 February 2021; Accepted 8 March 2021; Published 28 March 2021

Academic Editor: Bin Gao

In high-speed train safety inspection, two changed images which are derived from corresponding parts of the same train and photographed at different times are needed to identify whether they are defects. The critical challenge of this change classification task is how to make a correct decision by using bitemporal images. In this paper, two convolutional neural networks are presented to perform this task. Distinct from traditional classification tasks which simply group each image into different categories, the two presented networks are capable of inherently detecting differences between two images and further identifying changes by using a pair of images. In doing so, even in the case that abnormal samples of specific components are unavailable in training, our networks remain capable to make inference as to whether they become abnormal using change information. This proposed method can be used for recognition or verification applications where decisions cannot be made with only one image (state). Equipped with deep learning, this method can address many challenging tasks of high-speed train safety inspection, in which conventional methods cannot work well. To further improve performance, a novel multishape training method is introduced. Extensive experiments demonstrate that the proposed methods perform well.

1. Introduction

Traditional classification tasks using supervised learning methods, such as neural networks and support vector machines, generally require that all categories are available and the number of samples is sufficient. However, in the area of high-speed train safety inspection, abnormal targets that indicate there are underlying dangers while the train is running are scarce. Thus, we do not have a sufficient number of samples to implement deep learning to detect the abnormal targets. Instead, we devised a method based on the structural similarity method (SSIM) in the previous work [1]. In this method, the historical train images without malfunction are taken as baselines. When the current images are obtained and compared against the baselines, the changes occurring to the current trains are detected. As trains are exposed to the open air, there are various complex factors that cause the train surface to change. Therefore, most of the changes are not abnormal targets (correct alarm) but safety changes (false alarm) such as stains and marks, as shown in Figure 1(a) (row 1 and row 2). Besides, in order to obtain superior imaging quality, we photograph the train with supplementary lighting that usually leads to luminance difference. The luminance changes are usually mistakenly detected as abnormal targets, as shown in Figure 1(a) (row 3).

Although the previous work [1] saves plenty of manpower, there remains a need for inspectors to spend time classifying which changes are dangerous. To further reduce labor cost, this work is aimed at an automatic identification of the correct alarms with deep learning. There are various challenges in this task. The correct alarms are the components that are either loose (movement or rotation) or lost

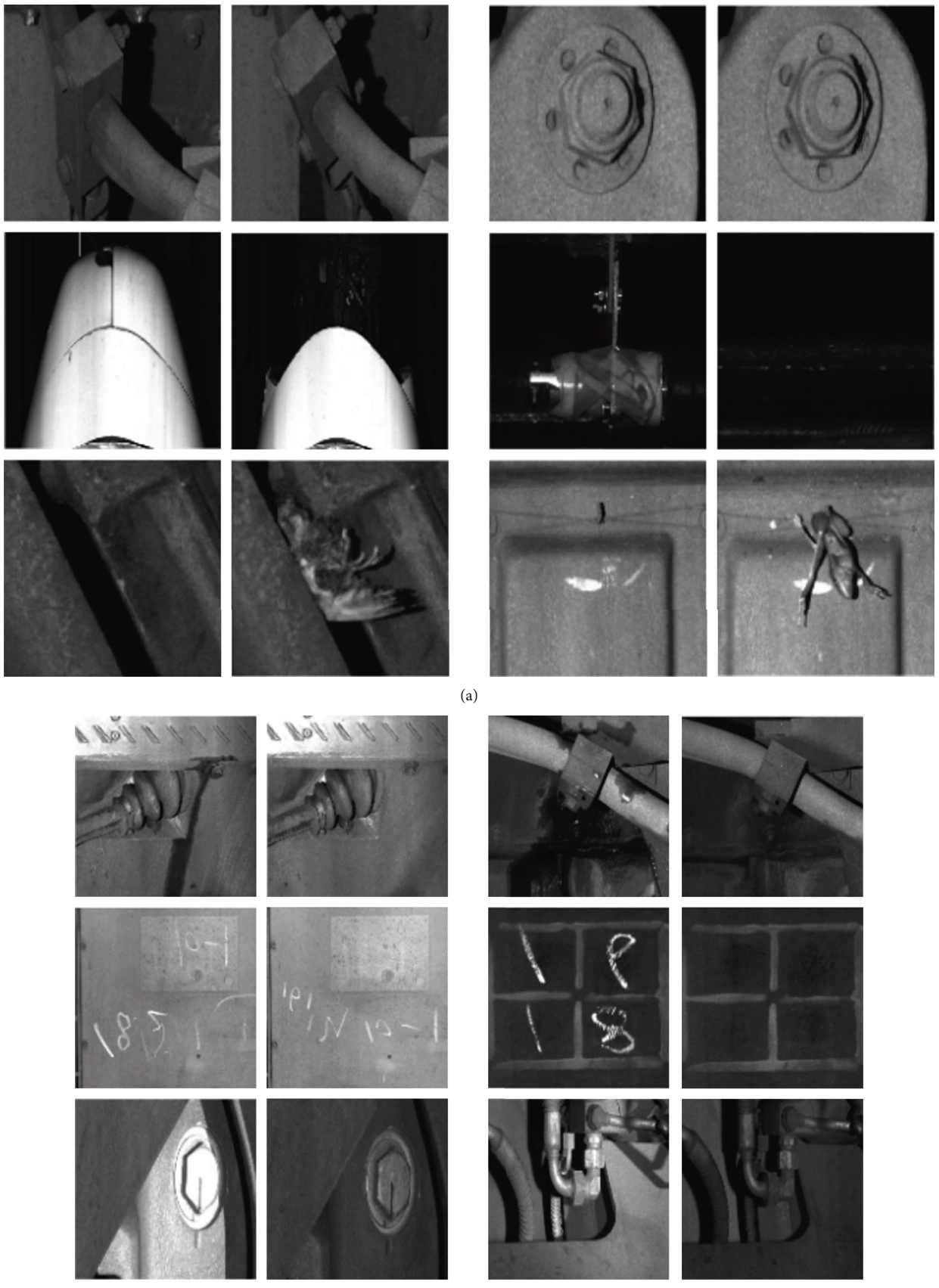


Figure 1: Examples of changes: (a) correct alarms; (b) false alarms.

and foreign bodies that appear on power installations such as the pantograph, as shown in Figure 1(b). These abnormal targets can lead the train to stop or even cause the train to turn over. Therefore, the abnormal target detection is extremely significant. However, correct alarms are incapable to be recognized by traditional classification methods, because their input is only one image (state). According to one state, algorithms cannot analyse whether the components are loose or lost. As for the false alarms, the algorithm should not just judge whether there are stains or luminance differences, because these existing signs do not indicate that there is no looseness, loss, etc. Therefore, the change information between two stages is required to assist the decision-making.

There are many components and equipment fitted on the train, especially at the bottom of the trains. That makes the image information of high-speed trains too complex to extract satisfactory edge information. Furthermore, without stable features, it is difficult to describe stains, luminance, foreign bodies, and component state changes. Actually, we cannot describe all kinds of shapes of the stains, but we can describe an abnormal condition of a component, even if describing all is unwise. Therefore, the manually designed descriptors, such as SIFT [2], may be not considered as an optimal method due to the factors mentioned above. On the other hand, it is effortless to obtain a large dataset that contains corresponding image pairs from the algorithm designed according to the previous work [1]. Therefore, deep learning [3] is adopted.

The paper is organized as follows. Section 2 explores other works that are somewhat like ours. Section 3 describes the two proposed convolutional neural networks (CNNs) for change classification in detail. Then, the experimental results and analyses are presented in Section 4. Finally, Section 5 gives the conclusions drawn from performing this study.

2. Related Works

2.1. Convolutional Neural Networks. CNNs, a family of algorithms especially suited to image analysis, have been applied in different ways, including image classification, object detection, and semantic image segmentation. Due to its strong ability of automatically learning high-level feature representations of images, CNNs can extract enough features for image classification [4–7] and perform better than traditional algorithms such as SIFT, HOG, and SURF. Moreover, it has the unique characteristic of preserving local image relations while performing dimensionality reduction. This makes it easy for CNNs to capture important feature relationships in an image and reduce the number of parameters the algorithm has to compute. CNNs are able to take as inputs and process both 2-dimensional images and 3-dimensional images; Ref. [8] proposed a 3DCNN to classify computed tomography (CT) brain scans which are 3-dimensional volumes. Based on the above, CNNs are the most popular machine learning in image recognition tasks. In object detection, there are also some excellent models such as Faster R-CNN [9], YOLO [10], and SSD [11]. Inspired by the successes of CNNs in above computer vision tasks, many researchers [12–15] make their efforts in different fields by using CNNs and achieve the state-of-the-art.

2.2. Change Classification. CNNs have been applied in different contexts for the comparison of image pairs [11–15]. Despite achieving state-of-the-art results in their tasks with two images, CNNs have yet to be applied to classifying change to the best of our knowledge. In Refs. [16], [17], and [18], CNNs are performed for change detection in the area of earth observation image analysis. By using a pair of coregistered aerial images taken at different times, the networks can infer the change map. It can be used to analyse the evolution of land use, urban coverage, deforestation, etc. In Refs. [19] and [20], the networks are trained to determine if two images correspond to each other by learning their similarity metric. They are widely used in image retrieval and face verification. By leveraging the convolution neural network, a variety of different challenges, for instance, changes in viewpoint, illumination problem, shading, and camera setting difference, are circumvented.

In brief, the first work is to detect where the changes occur, and the second one is to compute how similar they are. Our task is to recognize what kinds of changes happen or judge if these changes are dangerous. It needs to be emphasized that despite our use of change classification to identify abnormal targets for high-speed trains, it can be used for recognition or verification applications where decisions should be made over change information. In addition, although the inference process of the networks is comprised of one stage, they inherently divide the task into two parts, learning change information and classifying the change. We will show it in Sections 4 and 5.

3. Proposed Method

In this paper, two convolutional neural networks with reference to the residual network (ResNet) are presented [21, 22] to perform the change classification. There are two major differences between the change classification task and the traditional ones. First, the input of the networks should be two images instead of one that is required for the traditional classification task. Second, besides extracting image features, the proposed networks should learn to compare the image pairs to detect the change information.

The most straightforward way of improving the performance of the neural networks is to increase the depth [21–26], for which our networks are designed to have 32 layers. By extensive experiments, it is demonstrated that, as for our task, networks going deeper and wider (more units at each layer) cannot bring higher test accuracy but overfitting. The depth and width of our ultimate networks are optimal. The networks are trained end-to-end with 16k image pairs, and the number of samples is enough for a sufficient convergence with 80k iterations. Pretraining with other datasets is not utilized due to the differences between the conventional classification tasks and ours, and the considerable type differences between our dataset and the publicly available datasets such as ImageNet and MS COCO. In addition, according to Ref. [27], if the dataset is large enough (>10k), pretraining only

helps accelerate convergence but does not improve test accuracy or reduce overfitting. Thus, our designs are deemed reasonable. Moreover, a multishape training method is introduced to improve the performance.

3.1. Architectures. As mentioned above, the input to the CNNs is a pair of images. In this case, the main problem is how to integrate the two-image information to feed into the networks. The images we use are 1-channel grayscale images. The first idea is cascading the two images to be a "two-channel image." Although the "two-channel image" does not exist, it is convenient to process in the CNNs using two-channel convolution kernels. This architecture is called a cascaded model as shown in Figure 2(a). In the cascaded model, the two-channel image is processed by convolution layers to obtain the feature maps that contain change information. In order to reduce the overfitting, the global average pooling [16] is used for these feature maps to derive the final feature vector. Finally, the change category is outputted by the fully connected layer (FC).

The second architecture is inspired by Zhan et al. [28], in which two parallel networks are used to learn the pixel domain and wavelet domain information, and are cascaded by a fusion layer to implement image deblocking. Distinct from their work, the two parallel networks are involved to extract feature maps of the historical image (baseline) and current image as shown in Figure 2(b). Identical to the cascaded model, each branch applies a series of convolution layers and global average pooling. Then, the two branch outputs are concatenated and given to the top network that consists of FC. The two branches can be viewed as two feature extractors and the top network as a classifier. Consistent with Siamese and pseudo-Siamese networks [16–20, 26, 27], according to whether the weights of the two branches are shared, this architecture can be categorized into two types. Their performance is shown in Section 4.2.

3.2. Network Details. At present, ResNet [21, 22] and inception network [29–32] are accepted as excellent architectures. Therefore, while designing our networks, we refer to both of them. Owing to efficient convergence performance and concise structure, the residual module is primarily utilized in our networks. We adopt the bottleneck block that consists of two 1×1 and one 3×3 conv kernels [33]. The two 1×1 kernels are involved in dimensionality reduction and increment [21, 22, 29, 31, 32] to reduce the computation workload. The reason why we choose 3×3 conv kernels is that it has been demonstrated that multiple 3×3 conv kernels have the same receptive field as the larger one and have better nonlinear expressiveness due to activation function being used multiple times [24, 31]. Figure 3 presents the details of the block, in which batch normalization (BN) [30] is used as preactivation to improve the regularization of our models. The block can be expressed as

$$x_{i+1} = x_i + F(x_i, W_i), \tag{1}$$

where F indicates a series of BN, ReLU, and convolution operation; x_i and x_{i+1} are the input and output of the block; and W_i is the parameter which the model needs to learn. Recursively, Equation (1) is transformed into

$$x_m = x_n + \sum_{i=n}^{m} F(x_i, W_i). \tag{2}$$

Thus, the feature x_m of any deeper layer can be denoted as the feature x_n of any shallower layer n plus a residual function. Moreover, Equation (2) contributes to nice backward propagation properties. Denoting the loss function as l, we can obtain

$$\frac{\partial l}{\partial x_n} = \frac{\partial l}{\partial x_m}\left(1 + \frac{\partial l}{\partial x_n}\sum_{i=n}^{m} F(x_i, W_i)\right), \tag{3}$$

so that the loss can be directly propagated back to any shallower layer and the gradient of a layer cannot vanish [22]. At the end of the networks, the SoftMax layer is used to generate the pseudoprobability distribution, and by computing the cross-entropy, the loss is obtained to train the networks.

3.3. Multishape Training. The image pairs are directly provided by the previous work [1], and the shape is arbitrary. Namely, the height-width ratio is uncertain, for which our network should adapt to different shapes. To address the issue of different image shapes in training, we utilize three shapes: 180×360, 256×256, and 360×180. Roughly the same total pixels of the three shapes ensure that the computation is approximate in training. Thanks to the global average pooling, before being fed into FC, three shapes can be converted to the same length vector. In training, image pairs are alternately reshaped to the three shapes. While testing, image pairs are reshaped to the closest one.

In doing so, the dataset is augmented to some extent, and the benefit is twofold. In addition to improving the test accuracy due to a larger amount of data, it is conducive to change learning. However, in many previous works, such as R-CNN [34] and SPP-net [34], warping is not recommended due to the veridicality change. In contrast, our task is to recognize not what it is but the changes. Thus, after deformation, the change learning remains unaffected. In particular, stains, luminance changes, and foreign bodies have no stable feature. As a result, if the shape is changed, we could be unable to realize that. The examples are shown in Figure 4.

From Figure 4, it can be seen that after warping, the new stains, luminance changes, and foreign bodies are generated, and they all look natural. Certainly, the backgrounds may be anamorphic. However, they are desirable and make the networks more capable of learning changes instead of background category. For instance, as for the looseness such as what is shown in Figure 1(d), after training, the network may not learn what changes occur but learn that these components are usually in trouble, that is, even though the components in Figure 1(d) do not rotate, it can be recognized as a correct alarm. It is confirmed that this situation does not occur in Section 4.5.

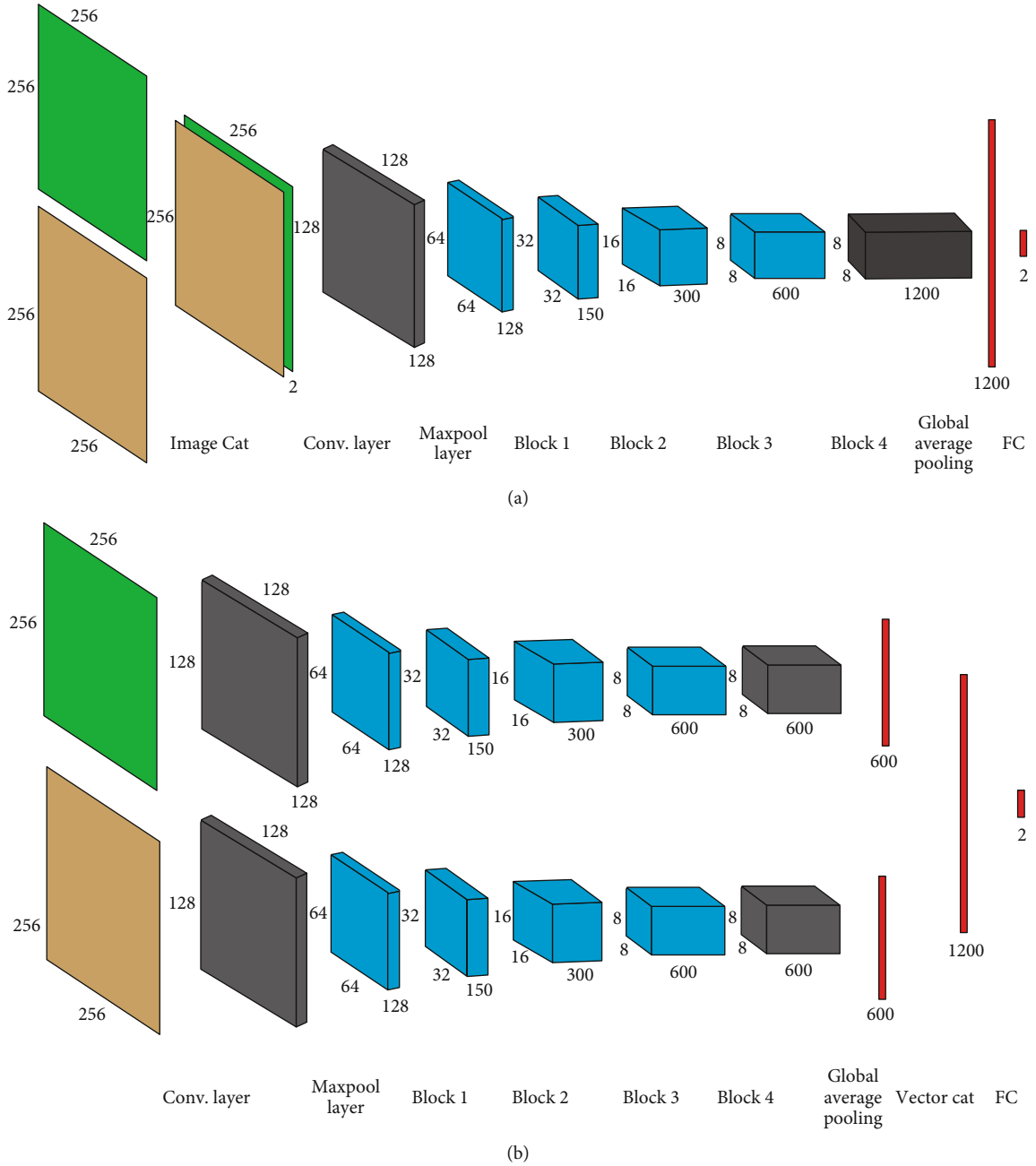


Figure 2: Two architectures: (a) cascaded model; (b) parallel model.

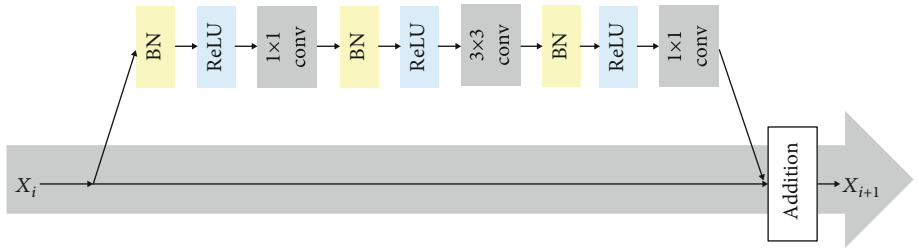


Figure 3: Bottleneck block.

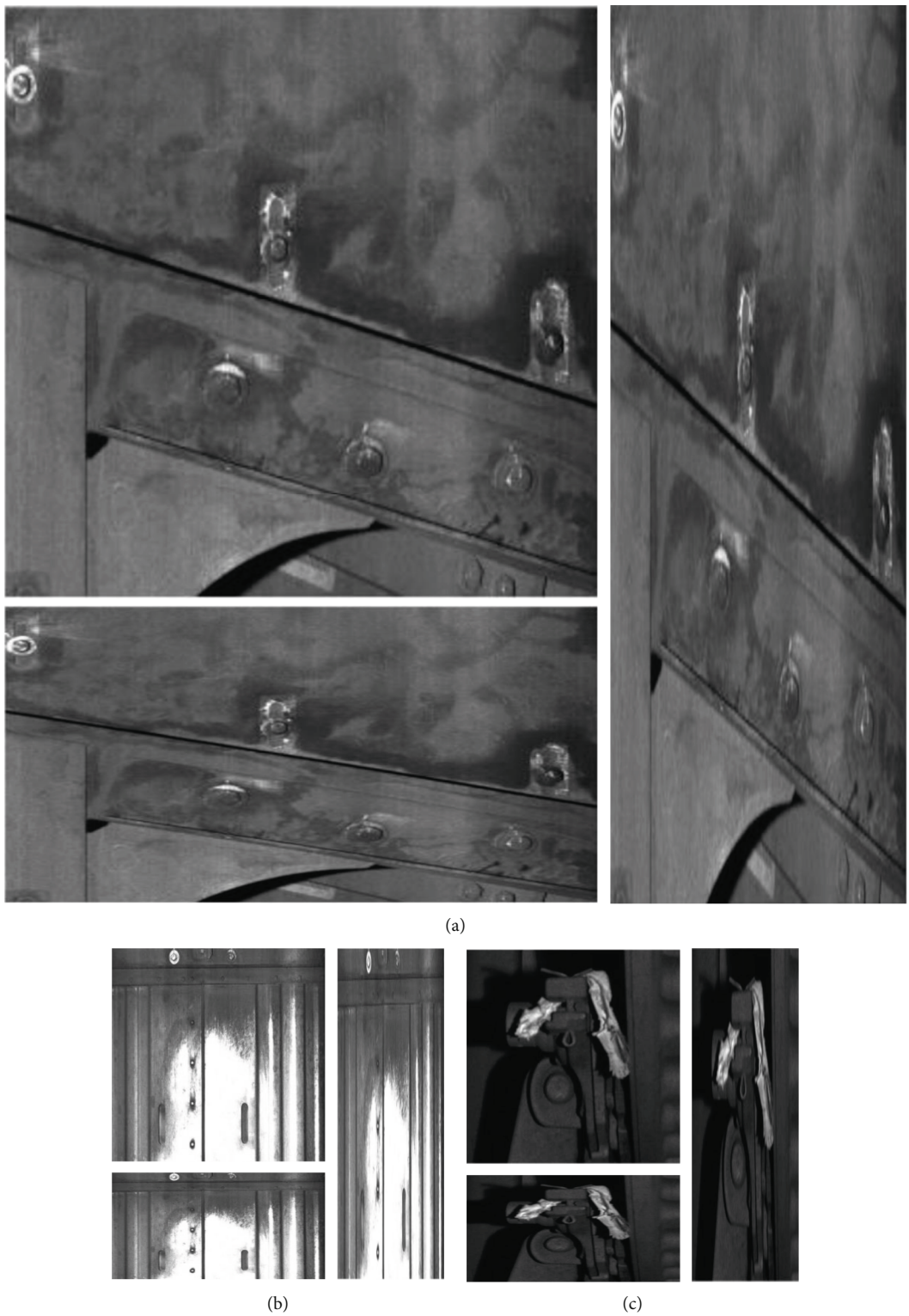


Figure 4: Examples of deformation: (a) stain; (b) luminance change; (c) foreign body.

4. Experimental Results and Analysis

As compared to tradition classification tasks, it has different input and a different goal. Thus, it is meaningless to compare our networks with state-of-the-art networks such as ResNet [21, 22] and inception network [29–32] which are all applied on conventional classification tasks. The point of our experiments is to determine the optimal configuration and to explore the reasonable pretreatment methods for change classification.

All networks are trained with Adam [35]. An exponential decay learning rate is used. The initial value is 0.01, and the

decay rate is 0.99. Except for the first conv layer, before convolution, BN and ReLU are performed in the first place, and the batch size is 96. To prevent overfitting, the L2 regularization is adopted and weights are initialized with the Xavier initializer [36]. All experiments are implemented six times using TensorFlow with an Nvidia GTX1080ti GPU and Intel i7-7700 CPU. The source code is publicly available at https://github.com/vivids/change_classification.

The experimental metrics used in our model are accuracy, precision, recall, and F1 score. The calculation method is shown as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4}$$

$$\text{presicion} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{6}$$

$$F = \frac{\left(\alpha^2 + 1\right) \times \text{accuracy} \times \text{precision}}{\alpha^2 \times (\text{accuracy} + \text{precision})}, \tag{7}$$

where TN, TP, FN, and FP are indicated in Table 1. F is a kind of comprehensive evaluation metrics. If α is equal to 1 in (7), it is the F1 score.

4.1. Data Sets and Data Processing. With the assistance of our previous work [1], about 18k image pairs that are the corresponding parts of the same high-speed train were collected at different times. These images are all taken from the high-speed train's body and its key components, such as the locomotive running gear, bogie, wheel, fastening bolt, and pipeline. Due to the different sizes of different key components, the acquired images have different resolution sizes, ranging from tens to thousands. The most defects contained in this dataset are the forebody, which is brought by a tree branch, the body of birds and other animals, plastic bag and other light garbage, and so on. In movement and rotation, one of the defects is a loose fastening component loose and a loose bolt, respectively.

While labeling the image pairs, it was found out that the category of many pairs is ambiguous as a result of cooccurrences of multiple circumstances such as the first image pair in Figure 1(a), where a stain and a luminance change appear simultaneously, so that the exacting correct multiclassification dataset is not available. Considering that we are not concerned about what kinds of changes occur, but in terms of whether they are dangerous, the change classification can be regarded as a binary classification task, a correct alarm or a false alarm. As the correct alarms are all structure changes whereas the false ones are nonstructure changes, the binary classification is feasible. Our experiments are primarily aimed at binary classification. Multiclassification experiments (with unsatisfactory multiclassification dataset) are also executed to demonstrate that the network can recognize different kinds of changes and assess the detail performance of the networks.

Table 1: Confusion matrix.

		Predicted label	
		Positive	Negative
True label	True	TP	FN
	False	FP	TN

As for the multiclassification task, the dataset is split into six categories, stain, luminance, mark, rotation, movement, and foreign body. Concerning binary classification, the first three categories are merged as false alarms while the rest as correct ones. There are more false alarms than correct ones, for which we discard some false ones for equity purpose. In both multiclassification and binary classification, we select about 10% data to test the networks, and the details are shown in Table 2.

In a traditional classification task, before training, the images are usually standardized to the same distribution where the mean is 0 and the variance is 1. However, it can eliminate the brightness difference between an image pair, so it hinders the networks from learning luminance changes. Instead, we merely normalize all image pixel values to [0, 1]. The images are resized to (256×256) for single-shape training and are alternately resized to (180×360), (256×256), and (360×180) for multishape training.

4.2. Two Architectures. The depth and width of neural networks are hyperparameters. To explore the optimal settings, we conduct many experiments. In Table 3, taking the cascaded model as an example, some typical settings are listed. In this section, we design two architectures based on the slim model (see Table 3 for details) to compare their performance and will demonstrate that the slim model can perform well in both speed and accuracy in Section 4.3. The two architectures are shown in Figure 2.

From Table 2, we can clearly see that cascaded model outperforms the parallel ones by a large margin in all metrics, which is attributed to the independent feature extraction of both branches, a result of which the parallel models cannot learn change information well. To further validate that the independent feature extraction hampers the learning for changes, we construct a hybrid model as displayed in Figure 5. In the front part of the network, two images are processed independently. In this way, the network can better extract the fundamental information, such as edges, of the two images. The rest is the same as the cascaded model, so this network has enough layers to detect and process the change information. However, from Table 4, it can be seen that independent feature extraction indeed does a disservice.

The Siamese models used in Refs. 16, 17 and 18 are similar to the parallel models, but they can perform well in change detection which is due to the difference between the two tasks. Moreover, it is also suggested that the change information is primarily extracted in the first few layers, for which these layers are significant to our proposed networks. We will demonstrate that the first convolutional layer is responsible for detecting change information in Section 4.5. Thus, we should integrate the two-image information early.

Table 2: Dataset details.

	2-cls		6-cls					
	False	Correct	Stain	Luminance	Mark	Rotation	Movement	Foreign body
Train	7857	7770	3449	3533	1824	3433	2636	1651
Test	1000	1000	350	350	350	350	350	350

4.3. Architecture Optimization. With networks going deeper, the performance is usually improved [23–26]. However, they are proven by training and testing with some very large datasets such as ImageNet and MS COCO. The reason is not only the deep networks having better nonlinear representation ability but also the shallow networks being underfit for a large dataset. In this section, we demonstrate that as for a small dataset, the deeper and wider network cannot improve the performance but can cause overfitting and bring more computation. We explore six networks of varying depth and width as shown in Table 3. For quantitative analysis of the complexity of the proposed method, we analyse the FLOPs of our networks. In our networks, the 101 layers (deepest) and 32 layers (thin) are the largest and smallest networks with FLOPs 7.5×10^9 and 6.8×10^8, respectively.

Table 5 presents the quantitative evaluation of the above six models. Each of them is tested six times to ensure objectivity. The modified ResNet-50 model [21, 22] is applied in the experiment. The modification is that the channel number of the first conv kernel is modified from 64 to 75 to be consistent with the slim model. The result reveals that the slim model is the most appropriate. Although the fat model achieves an excellent precision rate, it does poorly in the recall metric, as a result of which, the F1 rate is reduced as well. Moreover, the fat model is time-consuming. Owing to overfitting, the models that have more parameters may ignore the learning of category generality but memorize the training images. Thus, while testing, the results are not desirable. On the contrary, if the model is excessively thin or shallow, it is not qualified for the change classification task, that is, the model is underfitting.

4.4. Data Preprocessing. Preprocessing is effective in preventing the model from being affected by the irrelevant factors to some extent. For most recognition tasks, the data preprocessing can augment the dataset to improve the performance of the models. The common pretreatment methods include flipping, grayscale transformation, standardization, cropping [37], etc. However, regarding our task, the learning for different categories can be disrupted by some preprocessing methods. The methods of changing grayscale values are not suitable for luminance changes. Considering that the abnormal targets usually occupy a small part of our images, the cropping is not reasonable. In this section, we first implement training with standardization to validate that it can cause hindrance to the learning for luminance changes. Then, we train our model with image flipping horizontally and vertically to verify that it can improve accuracy. The results are listed in Table 6.

It is obvious that through flipping, the performance of the network is improved, and by means of standardization, the performance is degraded by a large margin. To further explore the influence of standardization, we implement six-category classification experiments. From Table 2, it can be seen that the number of the six categories is uneven, especially for the mark and foreign body. We first select 350 image pairs as the test data and then augment the number of marks, foreign bodies, and movements in the training set by means of grayscale transformation and cropping to equalize the dataset. As aforementioned, these augment methods are not suited to all scenarios, for which it cannot be used in training. However, we can augment the images in the disk and select the ideal ones. Because the test set is selected in advance, the experiments are considered reasonable.

Table 7 reveals that, as predicted, the recall of luminance drops sharply and the recall of stain decreases from 84.00% to 79.71%. Due to the decrease of brightness difference, the model finds it is more difficult to classify stain and luminance. According to Figure 6, we can find that, after standardization, there are more instances of luminance predicted as stains, as well as stains predicted as luminance. For instance, in Figure 6(a), 12.57% of luminance examples are predicted as stains, and 9.43% of stain examples are predicted as luminance. Based on Figure 6(b), after using standardization, the error rate increases to 18% and 12.29%. However, owing to the decrease of brightness difference, the network can learn some categories better, such as the movement and mark. For example, in the confusion matrix (Figure 6), less mark examples are classified as stain and luminance. The rate at which mark examples are classified as stain or luminance has decreased by 1.14% and 1.41%, respectively. Some categories can be targeted to use the standardization method, but it will lead the training set to have nonuniform distributions that are not beneficial for training [30]. Therefore, in our experiments, we do not adopt the standardization method.

4.5. Multishape Training. Multishape training is conducive to learning change information from different shape images. First, it can augment the dataset. Second, owing to being warped, the backgrounds are anamorphic, but the change information is almost unaffected. Certainly, we should reshape the images properly; otherwise, the change information will be harmed as well. Last, while testing, thanks to the ability to process multiple shapes, we can convert the image to the ideal shape for prediction. If high speed is not required, we can make inference with all shapes to vote which category it is. Furthermore, if the precision is pursued, only when the results predicted with all shapes are consistent will the final decision be made. Otherwise, it should be submitted to the inspector to judge.

Table 3: Configurations of cascaded models.

Layer name	Output size	32 layers (fat)	50 layers (ResNet-50)	32 layers (slim)	32 layers (thin)	23 layers (shallow)	101 layers (deepest)
conv1	128×128	7×7, 75, stride 2					
Pooling	64×64	3×3 max pooling, stride 2					
conv2_x	32×32	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 50 \\ 3 \times 3, 50 \\ 1 \times 1, 150 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 96 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 50 \\ 3 \times 3, 50 \\ 1 \times 1, 150 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 50 \\ 3 \times 3, 50 \\ 1 \times 1, 150 \end{bmatrix} \times 3$
conv3_x	16×16	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 100 \\ 3 \times 3, 100 \\ 1 \times 1, 300 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 192 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 100 \\ 3 \times 3, 100 \\ 1 \times 1, 300 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 100 \\ 3 \times 3, 100 \\ 1 \times 1, 300 \end{bmatrix} \times 4$
conv4_x	8×8	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 200 \\ 3 \times 3, 200 \\ 1 \times 1, 600 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 384 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 200 \\ 3 \times 3, 200 \\ 1 \times 1, 600 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 200 \\ 3 \times 3, 200 \\ 1 \times 1, 600 \end{bmatrix} \times 23$
conv5_x	8×8	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 400 \\ 3 \times 3, 400 \\ 1 \times 1, 1200 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 786 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 400 \\ 3 \times 3, 400 \\ 1 \times 1, 1200 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 400 \\ 3 \times 3, 400 \\ 1 \times 1, 1200 \end{bmatrix} \times 3$
Rest	1×1	Global average pooling, fc, SoftMax, cross-entropy					
FLOPs		2.4×10^9	4.1×10^9	1.5×10^9	6.8×10^8	9.7×10^8	7.5×10^9

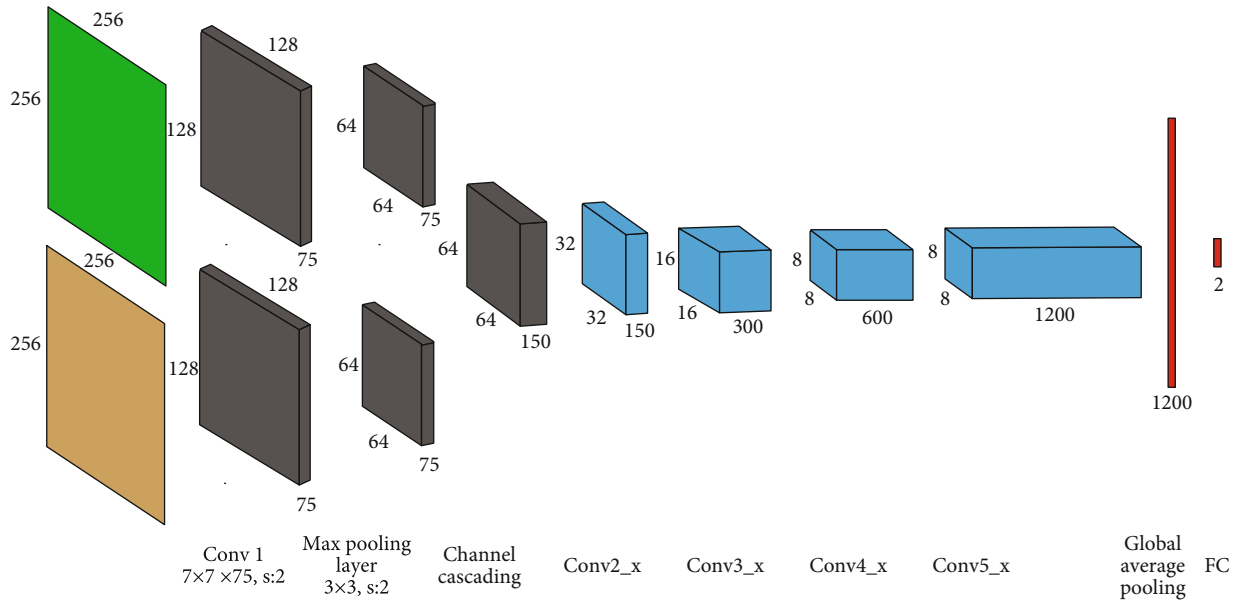


Figure 5: Hybrid model.

Table 4: Results of the three architectures. Both parallel and hybrid models have two versions according to whether the weights are shared (s) or unshared (u).

Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Training time (h)	Inference time (GPU/CPU, s)
Cascaded	92.02	94.08	89.71	91.83	7.05	0.0041/0.0461
Parallel (s)	84.06	88.67	78.15	83.06	17.85	0.0058/0.0792
Parallel (u)	83.94	86.38	80.60	83.38	19.45	0.0057/0.0798
Hybrid (s)	90.75	93.03	88.10	90.50	11.09	0.0039/0.0551
Hybrid (u)	90.90	93.84	87.55	90.58	10.73	0.0039/0.0565

Table 5: Results of cascaded models.

	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
32 layers (fat)	91.67	94.39	88.60	91.40
50 layers (ResNet)	90.82	92.95	88.33	90.58
32 layers (slim)	92.02	94.08	89.71	91.83
32 layers (thin)	90.43	92.47	88.03	90.20
23 layers (shallow)	91.16	93.98	88.00	90.88
101 layers (deepest)	90.57	93.56	87.13	90.22

Table 6: Result of different data preprocessing methods.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Slim	92.02	94.08	89.71	91.83
Slim_std	86.93	88.65	84.90	86.63
Slim_flip	93.07	95.42	90.45	92.89

To have a better understanding of multishape training, we implement the controlled experiments based on the slim model to show how it affects the network performance. Multishape training can improve the performance of our networks. Comparing the experiments 1 and 4 in Table 8 with the slim model in Table 5, it can be discovered that all scores of different metrics are increased. Similar to other data augment methods, multishape training can especially improve the performance for small datasets, the samples of which are not easy to obtain. We implement additional experiments with ideal shape inference on both binary and six-category classification datasets. We halve the data number of the binary classification dataset and carry out experiments without and with multiscale training successively. From Table 9, it can be seen that the performance is improved by a considerable margin. Besides, comparing Figures 7 and 6(a), the rate that the stain and luminance are wrongly predicted as each other goes down further. Comparing the recall of the six-category classification in Table 9 with that in Table 7, the same conclusion can be reached. Moreover, it is also revealed that multishape training is suitable for all categories according to the increase in all category recall rates.

More shapes do not indicate better performance. According to experiments 1, 3, 4, 5, and 6, although (148×442) and (442×148) shapes are included, the performance barely changed. Imagining that if we continue to add shapes such as (128×512), some images with an aspect ratio of 4 : 1 will be reshaped to the ratio 1 : 4. In this case, the change information may be damaged, thus rendering this sample useless

Table 7: Recalls of six-category classification experiments.

Method	Accuracy	Stain	Luminance	Rotation	Movement	Foreign body	Mark
Slim	86.05	84.00	82.29	83.43	86.00	87.14	93.43
Slim_std	85.52	79.71	72.86	84.86	94.00	84.29	97.43

Figure 6: Normalized confusion matrix of six categories: (a) slim method; (b) slim-std method.

Table 8: Effects of various design options on the slim model.

Options	1	2	3	4	5	6	7	8
Include (180,360), (360,180) shapes?	✓	✓	✓	✓	✓	✓	✓	✓
Include (148,442), (442,148) shapes?				✓	✓	✓		
1-shape inference?		✓						
Ideal-shape inference?	✓			✓			✓	
3-shape inference?			✓			✓		✓
5-shape inference?					✓			
Flipping?							✓	✓
Accuracy (%)	93.67	92.90	94.05	93.67	94.00	94.08	94.38	95.08
Precision (%)	94.44	93.16	95.41	95.17	95.18	95.27	94.71	95.73
Recall (%)	92.76	92.60	92.55	92.00	92.71	92.75	94.60	94.21
F1 (%)	93.59	92.88	93.96	93.56	93.92	94.00	94.35	94.96

Table 9: Additional experiments with ideal shape inference.

(a)

	Binary classification with a half dataset			
Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Slim	83.13	83.78	81.87	82.81
Slim with 3-shape training	88.86	89.37	88.17	88.76

(b)

	Six-category classification with 3-shape training						
	Accuracy	Stain	Luminance	Rotation	Movement	Foreign body	Mark
Recall (%)	89.78	87.43	83.62	85.98	92.19	91.76	97.71

for training. Therefore, it is mainly the rest shapes that contribute to the better performance of our networks.

Ideal shape inference can help improve the test score. In Table 8, we predict the change category using four strategies: only using shape (256×256) (1-shape inference); using the closest predefined shape adopted in training (ideal shape inference); using shapes (256×256), (180×360), and (360×180) (3-shape inference); and using 5-shape inference that has two additional shapes: (148×442) and (442×148). It is revealed by experiments 1 and 2 that converting the original image to the closest predefined shape is beneficial for the network to recognize changes.

Voting can give the prediction a boost. The idea is similar to multiview testing in SPP-net23. Instead of multiview images cropped from an original image, we feed multishape images reshaped from a test image to the network to predict its category. Finally, the final decision is made according to the majority. From experiments 1, 3, 4, and 5, it is obvious that voting can increase the test scores.

Currently, the best result is outputted by experiment 8 whose F1 score reaches 94.96. According to Tables 7–9, it is demonstrated that as the samples continue to be accumulated, the performance can be further improved.

4.6. Robustness. In order to verify the robustness of our model, we tested twelve pairs of images with luminance or rotation in our cascaded model. The results are shown in Figure 8.

As shown in Figure 8(a), there are 6 pairs of images with different light intensities. For example, in Row 1, the 3 pairs of images are affected by strong luminance, and almost more than half the area is covered by it. The 3 pairs of images with a little luminance in Row 2 are compared. We can see that the confidences of six-pair examples are slightly different with the highest score 99.99% and the lowest score 98.02%. They are all accurately predicted as luminance.

From human knowledge, rotation is easy to classify as movement that is an abnormal change needs to be detected. As well as luminance, in Figure 8(b), we selected six-pair rotation examples with different rotation angle. From the results, it can be discovered that the confidence of per image pairs is very close. It denotes that our cascaded model has strong robustness.

4.7. Analysis. It has been demonstrated that the features extracted by different layers are hierarchical [38]. For example, layer 1 may extract the fundamental features such as edges. Layer 2 responds to corners, and the deeper layers may capture similar textures and more class-specific variation. Usually, the function of the first few layers is uniform, so it is the common practice to freeze them when fine-tuning [9, 39]. To find out what our networks have learned,

Figure 7: Normalized confusion matrix of six-category classification with 3-shape training.

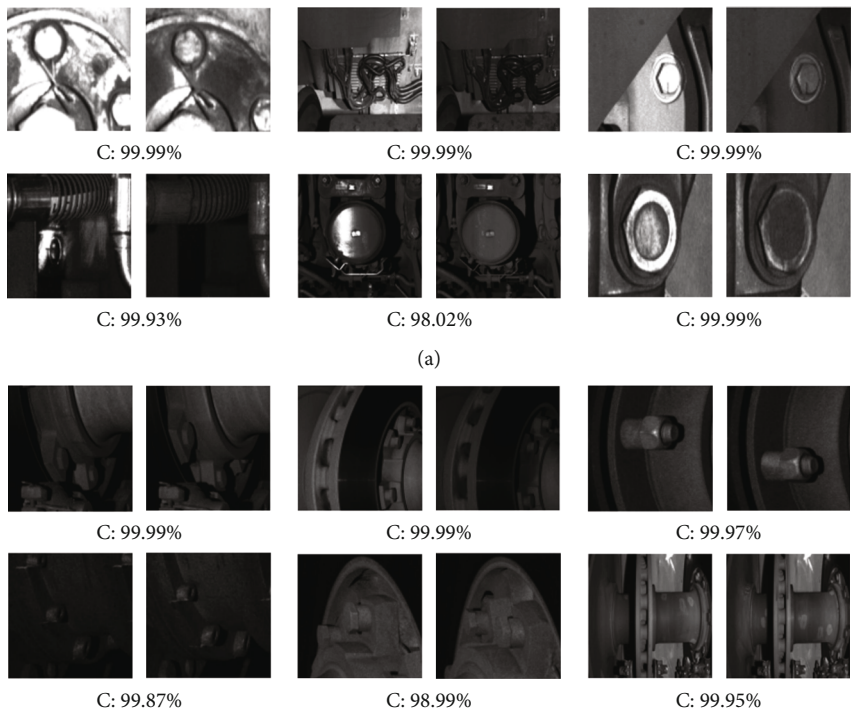


Figure 8: Twelve-pair examples for robustness verification: (a) luminance; (b) rotation. C: confidence.

we visualize all feature maps. Owing to the features extracted by the deep layers being too abstract to understand for us, we show four feature maps extracted by the first layers in Figure 9. We can find that our networks can not only extract the basic edge information but also learn to detect the changes (column 6). For example, in Figure 9(a) (R1, C6 and R3, C6), the component rotation is detected. As for (R2, C6) and (R4, C6), the change parts are segmented. In Figure 9(b) (C6), the position with stains and luminance changes is intensively responsive. Although we intuitively

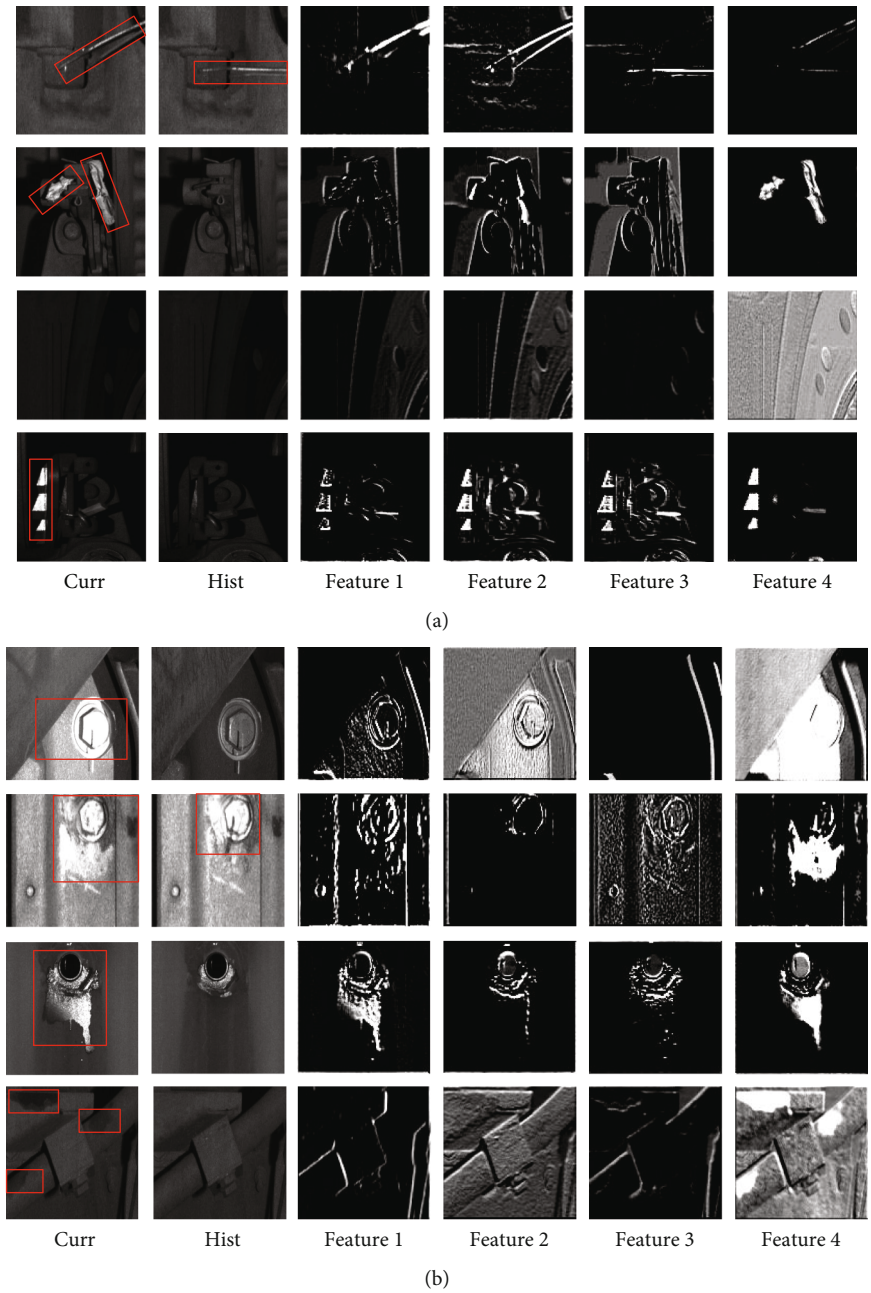


Figure 9: Feature maps extracted by the first convolutional layer for eight particular examples: (a) correct alarms; (b) false alarms. The red rectangles denote the difference of each image, such as, in (a) (R2, C1), there are forebodies; in (b) (R3, C1), the rectangle denotes the area with stain.

think that, compared with the early independent feature extraction methods, cascaded models may suffer from extracting the fundamental information of the two images, with difficulty, from Figure 9, we can see that the cascaded model can perform well in extracting the features of each image and in detecting the change information. Therefore, the cascaded model is superior.

The image content of trains is complex, which makes it unlikely to recite all situations for the networks. However, most of the abnormal targets appear in some fixed place such as the bolt, so it is reasonable to doubt whether our networks indeed learn how to recognize the changes. To verify that our networks do not memorize the components that usually go wrong but can identify the changes, we show three-pair examples that are the same components of the train in Figure 10. We can find that our networks are confident and can make the correct decision. In (R1, C2), even though there exists a luminance difference between the two bolts, the network remains capable to recognize that the bolt is loose. It is demonstrated that the networks have the sense of priority, namely, if the safety change and dangerous change simultaneously occur, the network will judge it as a correct alarm.

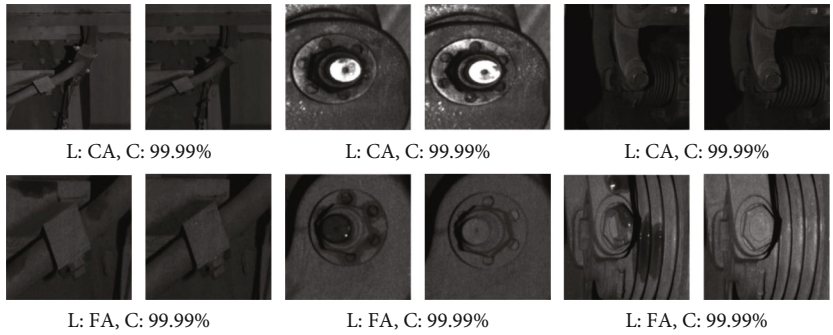
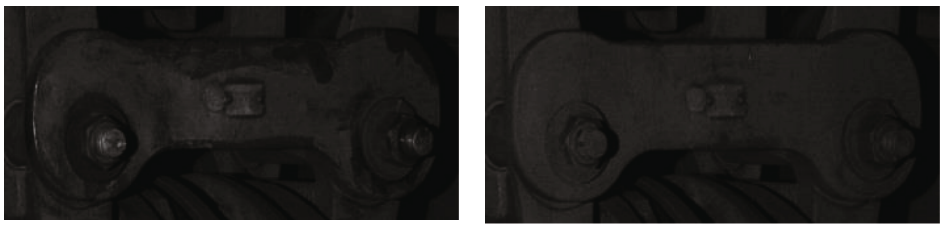


Figure 10: Three pairs of examples that are the same component of the train. L: label; CA: correct alarm; FA: false alarm; C: confidence.



(a) (b)

Figure 11: An example of false negative: (a) current state; (b) historical state.

However, if the dangerous change only occupies a very small part of the whole image while the safety change occupies the majority, such as what is shown in Figure 11, the network may not assess it as dangerous. In Figure 11, after glancing, we may treat it as a false alarm triggered by stains. However, if we look carefully, we will find one bolt loose. We will further study how to address this problem in the subsequent work.

5. Conclusions

In this paper, a cascaded model and a parallel one are presented to achieve change classification. According to the experimental results and network analysis, it is found out that the cascaded model is superior. Based on the cascaded model, extensive experiments are implemented to explore the optimal setting including the depth, width, and pretreatment methods. These experiments also demonstrate the differences among change classification task, traditional classification, and related works such as change detection. In addition, a novel training strategy is tailored to change classification, i.e., the multishape training method. It is experimentally validated that this strategy can improve the performance by a large margin and it is suitable for all categories.

Although we apply change classification to the task of high-speed train safety inspection, it is also suited to other classification scenarios where the decisions cannot be made with a single state. Change classification can also be considered as a solution to the tasks with rare positive samples. Our future direction is to explore how to address the false negative problem caused by structural changes occupying small areas in large images.

Data Availability

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

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61771409 and the Science and Technology Program of Sichuan under Grant No. 2019YJ0228.

References

[1] W. Song, X. Gao, J. Peng, J. Li, and L. Xie, "Abnormal target detection of high-speed train's roof," in *IEEE Far East Ndt New Technology & Application Forum*, C. Xu, Ed., pp. 143–148, IEEE, Xi'an, China, 2017.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[4] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE*

conference on computer vision and pattern recognition, Providence, RI, USA, 2012.

[5] X. Qiu, M. Li, L. Dong, G. Deng, and L. Zhang, "Dual-band maritime imagery ship classification based on multilayer convolutional feature fusion," *Journal of Sensors*, vol. 2020, 16 pages, 2020.

[6] D. G. Lee, Y. H. Shin, and D.-C. Lee, "Land cover classification using SegNet with slope, aspect, and multidirectional shaded relief images derived from digital surface model," *Journal of Sensors*, vol. 2020, 21 pages, 2020.

[7] B. Basnet, H. Chun, and J. Bang, "An intelligent fault detection model for fault detection in photovoltaic systems," *Journal of Sensors*, vol. 2020, 11 pages, 2020.

[8] J. Ker, S. P. Singh, Y. Bai, J. Rao, T. Lim, and L. Wang, "Image thresholding improves 3-dimensional convolutional neural network diagnosis of different acute brain hemorrhages on computed tomography scans," *Sensors*, vol. 19, no. 9, p. 2167, 2019.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016.

[11] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *European conference on computer vision*, Cham, 2016.

[12] R. Li, H. Liang, Y. Shi, F. Feng, and X. Wang, "Dual-CNN: a convolutional language decoder for paragraph image captioning," *Neurocomputing*, vol. 396, pp. 92–101, 2020.

[13] X. Cui, D. Wang, and Z. Jane Wang, "Multi-scale interpretation model for convolutional neural networks: building trust based on hierarchical interpretation," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2263–2276, 2019.

[14] R. Li, F. Feng, I. Ahmad, and X. Wang, "Retrieving real world clothing images via multi-weight deep convolutional neural networks," *Cluster Computing*, vol. 22, no. S3, pp. 7123–7134, 2019.

[15] Y. Tang and W. Xiangqian, "Salient object detection using cascaded convolutional neural networks and adversarial learning," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2237–2247, 2019.

[16] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral Earth observation using convolutional neural networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2115–2118, Valencia, Spain, 2018.

[17] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *IEEE International Conference on Image Processing*, Athens, Greece, 2018.

[18] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep Siamese convolutional network for optical aerial images," *IEEE Geoscience & Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017.

[19] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4353–4361, Boston, MA, USA, 2015.

[20] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conference on Computer Vision & Pattern Recognition (CVPR)*, vol. 1, pp. 539–546, San Diego, CA, USA, 2005.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision (ECCV)*, Cham, 2016.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[25] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *International Conference on Machine Learning, Lille*, 2015.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *NIPS*, 2015.

[27] K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet pretraining," 2018, https://arxiv.org/abs/1811.08883/.

[28] W. Zhan, X. He, S. Xiong, C. Ren, and H. Chen, "Image deblocking via joint domain learning," *Journal of Electronic Imaging*, vol. 27, no. 3, 2018.

[29] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE Computer Society*, Boston, MA, USA, 2015.

[30] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, Lille, 2015.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016.

[32] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, 2017.

[33] M. Lin, Q. Chen, and S. Yan, "Network in network," in *International Conference on Learning Representations (ICLR)*, Banff, 2014.

[34] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, pp. 580–587, Columbus, OH, USA, 2014.

[35] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, 2014.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, Italy, 2010.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[38] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, Cham, 2014.

[39] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago, 2015.