International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 32, No. 4 (2018) 1850011 (17 pages)

DOI: 10.1142/S0218001418500118

Tire Defects Classification with Multi-Contrast Convolutional Neural Networks

Xuehong Cui*
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao 266061, P. R. China
cuixuehongzhe@163.com

Yun Liu
Development Planning Office
Qingdao University of Science and Technology
Qingdao 266061, P. R. China
lyun-1027@163.com

Yan Zhang
School of Electromechanical Engineering
Qingdao University of Science and Technology
Qingdao 266061, P. R. China
zy@qust.edu.cn

Chuanxu Wang
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao 266061, P. R. China
wangchuanxu_qd@163.com

Received 11 May 2017
Accepted 18 July 2017
Published 12 September 2017

The objective of this study is to improve the accuracy in tire defect classification with limited training samples under varying illuminations. We investigate an algorithm based on deep learning to achieve high accuracy with limited samples. First, image contrast normalizations and data augmentation were used to avoid overfitting problems of the network with a large number of parameters. Furthermore, multi-column CNN is proposed by combining several CNNs trained on differently preprocessed data into a multi-column CNN (MC-CNN), and then their predictions are averaged as the output of the proposed network. An average accuracy of 98.47% is achieved with the proposed CNN-based method. Experimental results show that our

*Corresponding author.

scheme receives satisfactory classification accuracy and outperforms state-of-the-art methods on the same tire defect dataset.

Keywords: Deep learning; object classification; CNN; multi-contrast; tire defects.

1. Introduction

Computer vision and pattern recognition are becoming an important trend in non-destructive detecting technologies (NDT) nowadays, which is boosting various industrial detection applications, such as in fields of steel,[18] castings,[12,8] textile,[17] TFT-LCD panel,[21] nanostructures,[32,33] titanium-coated aluminum surfaces,[30] and semiconductors[3] etc. Besides these, tire defects detection and classification also has been investigated by researchers from both academy and industry areas over the past few decades[7,10,37–40] and is considered as one of the most challenging problems in industrial information revolution era[16] due to its unique properties illustrated in our previous study,[37] these difficulties include complexity, high-variety, and high dynamic range of tire X-ray images, which are incapable to accurately model such complex data.[15]

Previous work,[11,20,42] relied on orientation sensitive bar detectors, off-center-on-surround filters, Gabor filters to describe image patches, but these filters were fixed and could not adapt to the high-varieties of defect images, could only work well in an accepted standard model for object recognition.[4,19] In contrast, Convolutional Neural Networks (CNNs),[9,27] whose filters weights are randomly initialized but learned in an unsupervised way using back-propagation, can have a strong ability to extract features and do not require people to optimize these parameters, which is called deep learning.

Deep learning networks have been used to achieve state-of-the-art results on a number of benchmark datasets and for solving difficult artificial intelligence (AI) tasks. A variety of deep learning algorithms have been proposed, e.g. Deep sparse auto encoders (Bengio),[35] Stack sparse coding algorithm,[24] Deep Belief Network (DBN) (Hinton)[5] and their extrapolations, which learn rich feature hierarchies from unlabeled data and can capture complex invariance in visual patterns. In recent ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competitions,[25] Faster R-CNN[23] has achieved the best performance in the object detection. In the most recent years, one of the most exciting advances in depth learning is the Generative Adversarial Network (GAN),[1,22,31] which makes GAN a new favorite in the field of machine learning. And Deformable Convolutional Networks[13] solve the complex geometric deformation problem of the object, etc. These deep learning methods have proved to be successful for computer vision tasks by extracting appropriate features while jointly performing discrimination. Furthermore, in Ref. 26, Shi and Zhou *et al.* proposed a stacked deep polynomial network based representation learning method for tumor classification. A discriminant deep belief network is proposed in Ref. 41 to characterize SAR image

patches in an unsupervised manner in which weak decision spaces are constructed based on the learned prototypes.

In order to explore whether the deep learning has the capacity to represent complex tire X-ray image data accurately, we propose a MC-CNN that consists of five individual CNN. A CNN, as the basic component of our proposed MC-CNN, is a hierarchical deep neural network, alternating convolutional with max-pooling layers.[14] Our whole algorithm is illustrated in Fig. 1, which is of MC-CNN training and testing stages. During the training phase, contrast normalizations were used to preprocess the tire defect images for better accuracy in the classification, and intentional data augmentation step is to enlarge the learning samples to cope with the limitation of training samples under varying illuminations. For testing phase, data processing flow consists of data preprocessing and well-trained MC-CNN decision.

Based on implementation procedure of the tire defect classification algorithm, the paper is organized as follows: Section 2 describes the data preprocessing and the creation of the training set; Section 3 describes our MC-CNN architecture and its training; Section 4 presents the experimental results and performance evaluation.

2. Data Preprocessing

The images were collected from a typical tire manufacturing enterprise in China. Source images were derived from real-world defect detection system at the end of the manufacturing line and thereafter were labeled manually by human labelers. Figure 2 shows five tire defect samples. While our MC-CNN outputs six labels, which include these five a defects and the normal sidewall category. Conceivably, X-ray images have high variation in illumination and contrast. To boost robustness to these noises, we employ contrast normalization to preprocessing these images. Moreover, in view of the fact that our database is small, while sufficient amount of data is usually needed to train deep learning network, thus data expansion techniques are introduced to enlarge our train dataset in order to achieve a pattern model with higher performance.

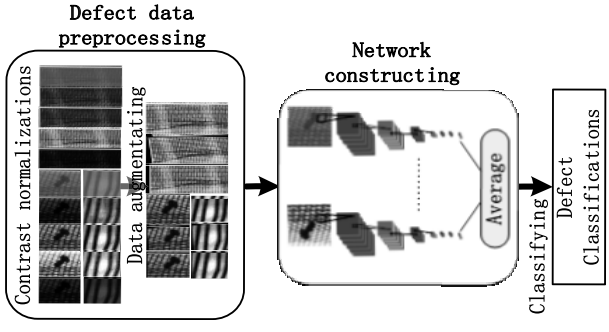


Fig. 1. Illustration of the defect classification algorithm.

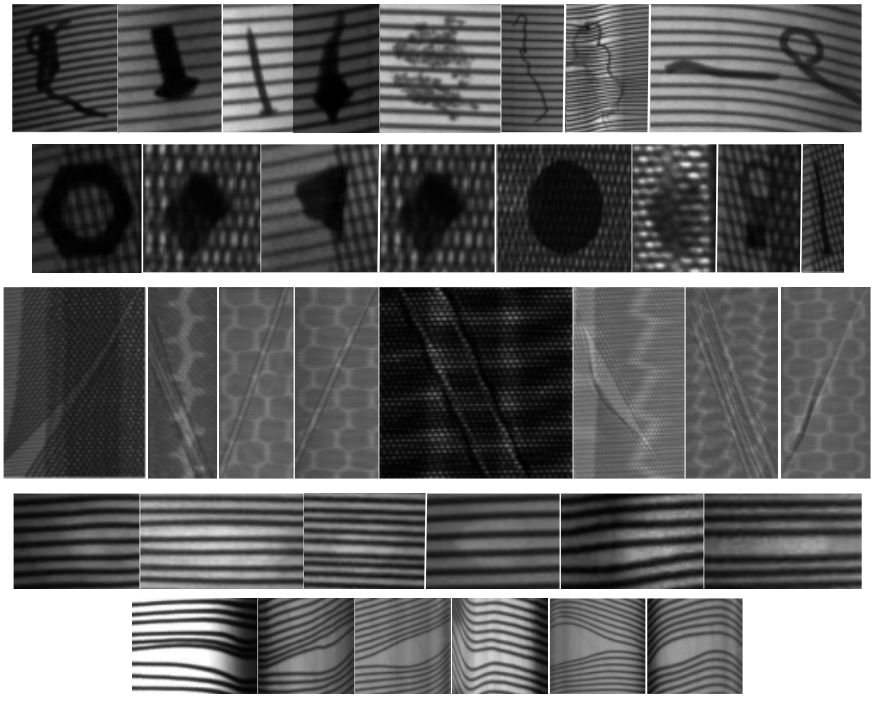


Fig. 2. Defect sample synopses of the evolving dataset. From top to bottom: Sidewall-Foreign-Matter, Belt-Foreign-Matter, Belt-Joint-Open, Bulk-Sidewall, Cords-Distance.

2.1. *Contrast normalizations*

We use the following four image preprocessing methods to normalize and enhance the original images. Therefore, each original image will have four preprocessed image results, altogether these five images will form an input to our five-channeled MC-CNN for training, as well as for testing. These four preprocessing ways are *image adjustment* , *histogram equalization*, *adaptive histogram equalization and contrast normalization*, which are briefly introduced below.

- *Image adjustment*: This to enhances image contrast by mapping its intensity values to new ones such that 1% of image pixels is saturated at low and high intensities, respectively. The method can automatically determine a threshold to increase image contrast.
- *Histogram equalization*: This to increases image contrast through transforming its intensity values into new gray levels such that the output image histogram is roughly uniform. The transformation function is given as follows:

$$S_k = T(r_k) = (L-1)\sum_{j=0}^{k} n_j \Big/ N = (L-1)\sum_{j=0}^{k} P_r(r_j), \quad 0 \le j < L \quad \text{and} \quad k = 0, 1, 2, \ldots, L-1, \tag{1}$$

where r_k is the input gray value of level k and S_k is its corresponding output, n_j is the number of occurrences of gray level j, N stands for the pixel number in image, L is the number of gray levels in the image, and $P_r(r_j)$ denotes the probability of an occurrence of a pixel of level j in the image, which is the histogram for pixel value r_j, and normalized within [0,1].

- *Adaptive histogram equalization*: This works on image tiles rather than the entire image. The image is split into nonoverlapping patches, called tiles, and the contrast of these regions are enhanced by adjusting their local histograms such that each histogram becomes roughly uniform. The formula for the local contrast enhancement algorithm is defined as follows:

$$x'_{i,j} = \begin{cases} T(x_{i,j} + k(x_{i,j} - m_{i,j})), & 0 \le x_{i,j} \le 255, \\ T(x_{i,j}), & \text{else}, \end{cases} \tag{2}$$

where $X_{i,j}$ and $X'_{i,j}$ represent the gray values of the images before and after the transformation, respectively. $m_{i,j}$ is the mean value of the pixels in the small rectangular region with $X_{i,j}$ as the center. $k(X_{i,j} - m_{i,j})$ is similar to a high-pass filter, which can enhance the edges and improve the local contrast in each region of an image, but also amplifies high-frequency noise, especially in the image's homogeneous areas. To avoid amplifying any noise, we can use the optional parameters K of adapthisteq function to limit the contrast.

- *Contrast normalization*: This enhances image edges by filtering the input image with a Difference of Gaussians (DoG filter). In this work, we use a 5×5 pixels filter. The formula is as follows:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{1}{2\pi K^2\sigma^2} e^{-\frac{x^2+y^2}{2K^2\sigma^2}}, \tag{3}$$

where σ^2 represents the Gaussian of variance.

Note that the above normalizations are performed in gray space. Figure 3 illustrates the effect of the four different normalization results. Later on, we can say in Table 2 of Sec. 4.2.3, when these five channels are tested individually, the four channels trained with preprocessed data have higher recognition accuracy averagely than the one trained with original images.

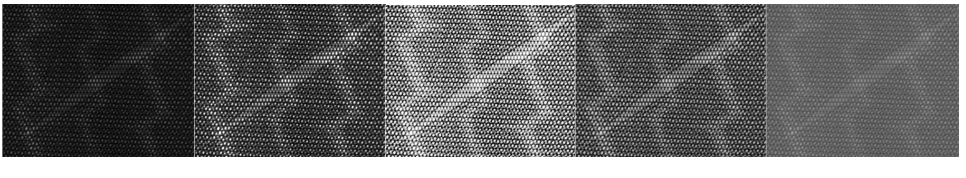


Fig. 3. Normalization results (From left to right: original, imadjust, Histeq, Adapthisteq, Conorm).

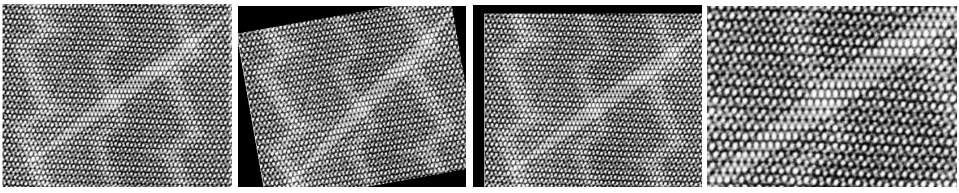


Fig. 4. Examples of transformation image (From left to right: original, rotation, translation and scaling).

2.2. *Data augmentation*

In the task of tire defect classification, CNNs are rarely used. One of the main reasons is that there is no large labeled dataset. A small amount of data will lead to overfitting of the model, making training error as very small but test error as particularly large. One way to solve this problem is to expand the training data. Usually, image geometric transformation, such as rotation, translation, scaling, or even noising are recommended to be good options to augment training set. Here we use them in a stochastic way, that is, these parameters come from a uniform distribution with specified ranges: $0.9 - 1.1$ for image size scaling; $\pm 10\%$ for translation and $\pm 5^\circ$ for rotation. We chose these proper modifications rather than extreme distortions just to prevent the emergence of new categories which will disorder the original defect patterns. Some examples of the data augmentations are shown in Fig. 4, and their positive impacts on recognition accuracy are later shown in Table 3.

3. Multi-Contrast Convolutional Neural Networks

Stimulated by microcolumns of neurons in the cerebral cortex,[2,6] we combine five individual CNN streams to construct a Multi-contrast CNN(MC-CNN); furthermore, in Ref. 14, handwritten digits experiment has verified that simply averaging the outputs of many CNN can produce better performance than a linear combination of all the CNNs. Therefore, we construct the MC-CNN by simply averaging the outputs of each CNN. The following three parts can give an overall description for our MC-CNN.

3.1. *Single CNN architecture and its training tips*

Because our MC-CNN consists of five individual CNNs, we first give brief introduction about its features and training tips. As a milestone of CNN, AlexNet has a significant architecture shown in Fig. 5(a), its unusual features go as follows. (A) ReLU Nonlinearity, which enhances training speed several times faster than their equivalents with tanh units. (B) Training on Multiple GPUs, which can greatly improve the speed of network training and make it possible to train large networks. (C) Local Response Normalization, which helps to prevent ReLUs from saturating and thus reduces network error rates. (D) Overlapping Max-Pooling, which can

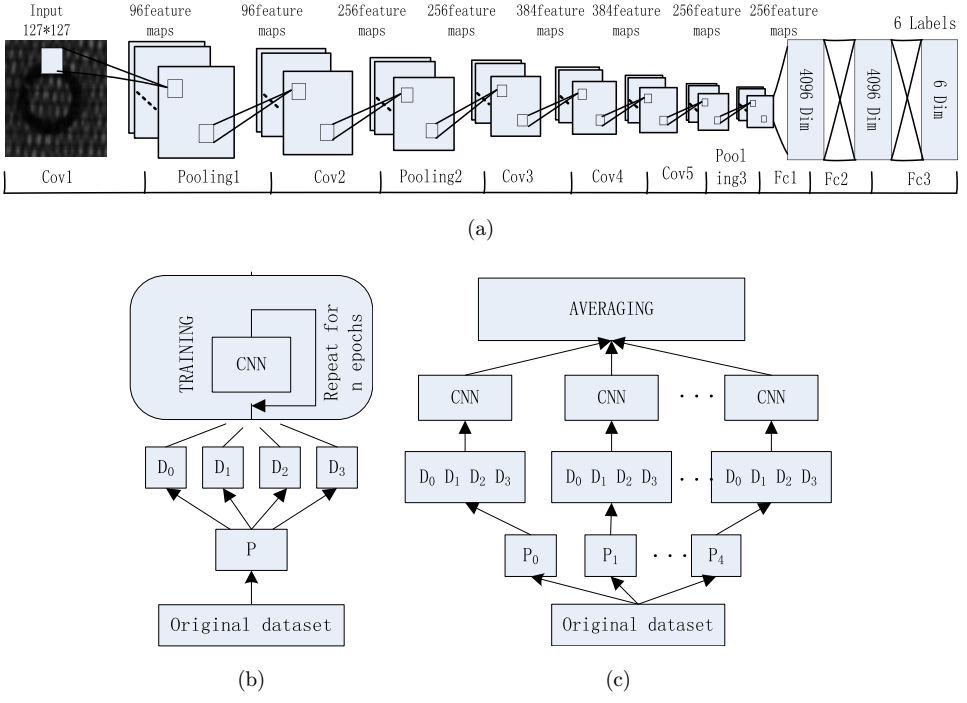


Fig. 5. (a) A single CNN architecture, (b) Training a CNN, (c) MC-CNN architecture.

slightly reduce overfitting. With the consideration of the above advantages, we chose AlexNet as our basic channel of MC-CNN.

In addition to its superior architecture, some strategies are adopted to prevent overfitting learning. (A) Data Augmentation: Artificially enlarging image dataset is the easiest and most common method to reduce overfitting, which we have talked it in last section. (B) Dropout: This later refers to the output of each hidden neuron with probability of less than or equal to 0.5, these neurons are abandoned. That is, they neither contribute to the forward pass nor participate in backpropagation. Specially, this AlexNet CNN output is a fully-connected layer which is fed to softmax and produces a probability distribution for six tire defect labels.

3.2. *MC-CNN design and its training*

Our MC-CNN is composed of five individual CNN, whose structure is shown in Fig. 5(c). It has several layers, first of all, the original image is preprocessed by different preprocessors $P_0, \ldots, P_4$, while P_0 indicates that no preprocessing operations are performed, P_1 to P_4 refer to four image normalization of *Adjustment, Histogram Equalization, Adaptive Histogram Equalization and Contrast* Normalization, respectively. Next, all original or preprocessed images are arbitrary distorted in a specified range to enlarge the dataset, which are marked as D_i. Here, D_0

indicates that no distorting operations are performed, while D_1–D_3 refer to size scaling, translation and rotation. Among them, the structure of a single channel CNN is explicitly illustrated in Fig. 5(b). At last, for a given input image, its predictions of five streams are averaged as final classification results, by the formula below:

$$y^i_{\text{MC-CNN}} = \frac{1}{5} \sum_{j}^{\text{streams}} y^i_{\text{CNN}_j}, \tag{4}$$

where i refers to the ith defect class and j is the jth stream for five CNN streams, respectively.

4. Experiment

Our experiments contain six parts. The first part is the initializations for dataset and single CNN configuration; the second one is to check optimal structure of MC-CNN, which is to judge the proper number of individual CNN in our MC-CNN to achieve best recognition effectiveness; thirdly, we make recognition accuracy comparisons between our MC-CNN and each of five individual CNNs; later on, we evaluate the effect of data augmentation; and then, comparisons to the state-of-the-art are carried on; finally, we analyze recognition results for each of six defect types. Additionally, our test platforms are based on 3.60 GHz 4-core CPUs and 16 GB RAM, and an *Ubuntu* 16.04, *Caffe* and *Python* 2.7.

4.1. *Experiment initializations*

4.1.1. *Initialization of our dataset*

We collected 1582 original tire-defect and normal sidewall X-ray images containing five typical defect categories and normal sidewall category, namely, Belt-Foreign-Matter (BFM), Sidewall-Foreign-Matter (SFM), Belt-Joint-Open (BJO), Cords-Distance (CD), Bulk-Sidewall (BS) and Normal-Cords (NC). They were derived from a manufacturing line in China and their defect patches are extracted and labeled manually. For testing dataset preparation, its acquisition process is the same as that of the training set, except no manual labels.

The initialization is shown in Fig. 6. According to statistics on the tire defect dataset, it consists of variable-resolution images ranging between 50×50 and 200×500 pixels due to the uncertainty of tire defect occurrences in the production line. Before training, all original images are preprocessed by four contrast normalizations (details in Sec. 2.1) as shown in Fig. 7. Therefore, we can create four additional datasets, and each dataset is distorted by rotation, translation and scaling (details in Sec. 2.2). Our MC-CNN implementation requires all training images to be of equal size, so the datasets were down-sampled or up-sampled to a fixed resolution of 127×127.

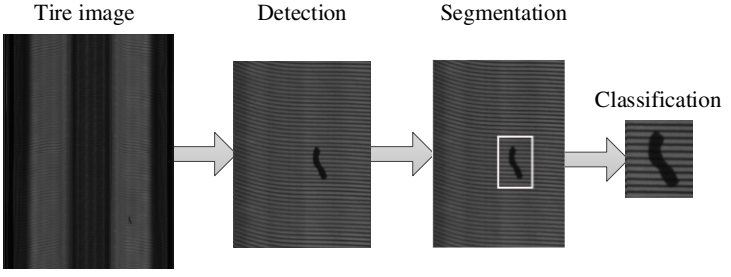


Fig. 6. Process of dataset initialization.

4.1.2. *Initialization of single CNN configuration*

With the consideration of the above advantages in Sec. 3.1, we chose AlexNet as our basic channel of MC-CNN. The architecture of AlexNet with 12 layers and parameters of each layer are displayed in Table 1. The output layer has six neurons corresponding to six tire defect types in our dataset. Before training, all of filter weights are randomly initialized by a uniform distribution in the range $[-0.05, 0.05]$. Our network uses Rectified Linear Units (ReLU) nonlinearity as each neuron's

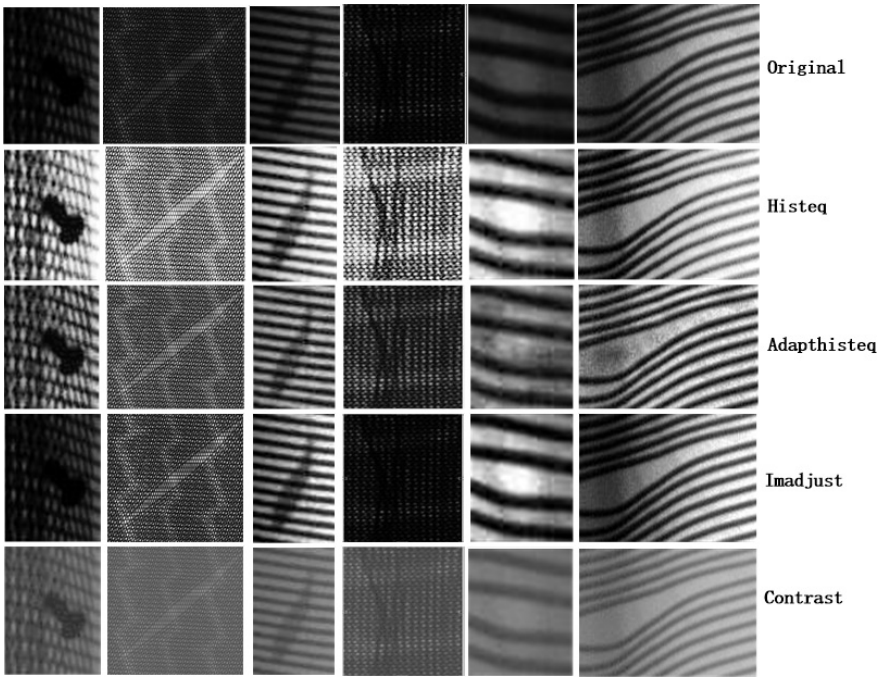


Fig. 7. Tire defects from the training set before and after normalization.

Table 1. 12-layered CNN architecture.

Layer	Type	Maps & Neurons	Kernel	Stride
0	Input	3 maps of 127×127 neurons		
1	Convolutional	96 maps of 30×30 neurons	11×11	4
2	Max pooling	96 maps of 15×15 neurons	3×3	2
3	Convolutional	256 maps of 15×15 neurons	5×5	1
4	Max pooling	256 maps of 7×7 neurons	3×3	2
5	Convolutional	384 maps of 7×7 neurons	3×3	1
6	Convolutional	384 maps of 7×7 neurons	3×3	1
7	Convolutional	256 maps of 7×7 neurons	3×3	1
8	Max pooling	256 maps of 3×3 neurons	3×3	2
9	Fully connected	4096 neurons	1×1	1
10	Fully connected	4096 neurons	1×1	1
11	Fully connected	6 neurons	1×1	1

activation function in order to learn blob, edge and other shape features in input images.

Parameter initializations are as follows: Considering the small quantities of validation dataset, in order to prevent error descending too fast, we set the initial learning rate *base-lr* as 0.001. For test dataset, we set test *batch volume batch* as 246, *test batch test-iter* as 4, and *test interval test-interval* as 200, namely, test every 200 iterations and display classification accuracy at a time. Unlike AlexNet in which two GPUs are used, in this work we set the *solver_mode* as CPU. The remaining parameters of the deep architecture were the same as the default parameters in the Caffe net optimization model.

4.2. *Optimal channel number testing for MC-CNN structure*

In order to explore how many single CNNs can be used to obtain the highest recognition rate when constructing MC-CNN, this section gives the performance comparison of MC-CNN with different CNN channels, which are single net, three nets, five nets and nine nets, respectively, as shown in Fig. 8. (Note that in order to show the comparison of experimental results more clearly, we set X scale of X-Axis to a *log form* when drawing).

As can be seen from Fig. 8, the recognition rate of MC-CNN with nine nets is the highest up to 99.26%. Followed by MC-CNN of five nets, its recognition rate is 99.12%. The recognition rate of MC-CNN of three nets is 98.87%. The network with the lowest recognition rate is MC-CNN with single net, whose recognition rate is 97.87%. Both experimental and theoretical results show that the MC-CNN with more single nets can improve the rate of classification accuracy. But increasing channels will aggravate computational cost and hardware cost. Moreover, it can also be seen from Fig. 8, the nine-channeled MC-CNN works (99.26%) similar to the 5-channeled MC-CNN (99.12%), which proves that increasing more channels does not further improve accuracy noticeably. To make a tradeoff, we adopt MC-CNN with five nets.

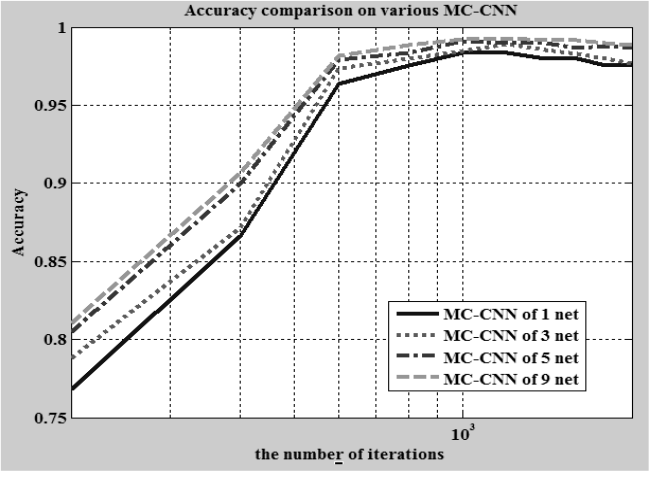


Fig. 8. Recognition rate comparison on various MC-CNN.

4.3. *Accuracy comparisons between MC-CNN and each of the five individual CNNs*

Here, our MC-CNN architecture contains five single nets, they are trained on original images and four different preprocessing images, respectively. Their recognition rates are summarized in Table 2. Each column shows the recognition rates of single CNN trained on the original and four different preprocessing images, respectively. At the same time, we also report the recognition rate of MC-CNN. As can be seen, the resulting MC-CNN with 5 CNN streams achieves a test recognition rate of 98.47% which is a drastic improvement with respect to any of the individual CNN. Besides, recognition rates of the latter four individual CNNs are better than the first column. That is, the recognition rate of individual networks where training samples have been preprocessed by four contrast normalizations is higher than that of the unprocessed. Finally, it is worth noting that none of single CNN recognition rates is outstanding, but combining them into a MC-CNN enhances robustness to different kinds of noise and achieves a higher recognition rate than any single CNN.

Table 2. Recognition rates [%] of the MCCNN and its five CNN.

Experiment	Original	Imadjust	Histeq	Adapthisteq	Conorm
1	96.84	96.81	96.92	96.52	96.73
2	96.72	96.79	96.85	96.77	96.79
3	96.62	96.83	96.82	96.85	96.81
4	96.66	96.80	96.87	96.78	96.70
Avg.	96.71	96.81	96.87	96.73	96.76
Average CNN recognition rate: 96.78					
Our MCCNN: 98.47					

Table 3. Results comparison of our MC-CNN without and with data augmentation.

Usage of Data Augmentation	None	Scaling	Translation	Rotation	Scaling and Translation	Translation and Rotation	Three Combinations
Overall Accuracy (%)	96.32	97.23	97.41	97.16	97.85	97.69	98.47

4.4. *Evaluation to the effect of data augmentation*

To prove the effectiveness of data augmentation for improving the performance, we experimented on none or one or more combination of data augmentation transform for a general comparison, shown in Table 3, which evaluates the effect of data augmentation on our MC-CNN. The results show that training with expansion defect data can yield a better result (2.15%) than without using data augmentation training. Accuracy is steadily improved (see Table 3) when more expansion data are used. The overall accuracy of the three combinations is up to 98.47%.

4.5. *Comparisons with the state-of-the-art methods*

4.5.1. *Recognition accuracy comparisons to state-of-the-art methods*

To validate the effectiveness of our scheme, we experimented the available state-of-the-art methods using the same dataset for a general comparison shown in Table 4, which includes BP neural network,[36] KNN,[36] Zhang,[36] PCA+BP neural network, ScSPM 09,[34] LLC 10,[29] KSPM-200-3,[28] KSPM-400-2[28] and AlexNet[25] methods. In BP,[36] KNN[36] and Zhang[36] methods, the tire image is represented by three characteristics: shape feature, gray feature and texture feature. And then BP neural network, KNN and nonlinear SVM classifier are used, respectively, to classify the images. But in the PCA+BP method, the result of linear combinations of all columns of image matrices forms the features of the image, then PCA is used to reduce the dimension, at last PB neural network is used to classify the images. In the KSPM-200-3 method, we set dictionary of size $N = 200$ with a 3-layered pyramid

Table 4. Comparison on state-of-the-art methods using the original dataset.

Methods	Test Accuracy %	Validation Accuracy %
BP[2]	86.2	/
KNN[2]	81.9	/
Zhang[2]	94.2	/
PCA+BP	69.44	/
ScSPM09	95.56	/
LLC10	94.85	/
KSPM-200-3	92.77	/
KSPM-400-2	92.37	/
Alexnet	96.84	98.33
Our method	98.47	99.12

structure, while in KSPM-400-2 we set $N = 400$ with pyramid structure of two layers. SIFT features were used in ScSPM09, LLC10 and KSPM methods, and linear SVM classifier was used in ScSPM09 and LLC10, while in KSPM-200-3 and KSPM-400-2 nonlinear SVM classifier was used. In AlexNet, we use the same parameters as the individual CNNs in this paper. As shown in Tables 2 and 4, all individual CNNs performance is better than any other method used for comparison on our tire defect dataset. Furthermore, our MC-CNN with 5 CNN streams achieves a recognition rate of 98.47% and has a drastic improvement with respect to any of the individual CNN and any other methods.

4.5.2. *Detailed reasons for our better results*

As can be seen from Table 4, the validation classification accuracies are better than the test classification accuracies in both of AlexNet classifier and our MC-CNN. There are two reasons for this: firstly, insufficient training samples, and secondly, parameters were not optimized. Zhang[2] received acceptable classification accuracies, mainly attributed to the extraction of exquisite features, as well as the proper choice of classifiers. However, the accuracy of the PCA+BP method is lowest due to the absence of exquisite features, even using the PCA technology for dimensionality reduction. ScSPM09 and LLC10 are two successful sparse coding-based methods that have been extensively studied and applied in various domains, both of them also exhibit better performance. But in all the methods except the AlexNet and ours, their extraction of image features and the selection of classifiers need deliberate design and optimization by researchers, which would affect the classifications dramatically.

Our proposed scheme is superior to other algorithms in classification performance. The main reasons are as follows: (1) bionic structure of the MC-CNN network: It is designed by combining several CNN columns according to the micro-columns of neurons in the cerebral cortex; (2) Each CNN channel is already sufficient to achieve new high performance due to the superior structure mentioned in Sec. 3.1; (3) In automatic deep learning, especially the end-to-end learning, which is quite unlike the traditional learning mentioned above, it need not divide the whole classification process into multiple steps or sub-problems, such as the extraction of image features and the selection of classifiers, instead, it can completely do everything from raw inputs to desired outputs. This not only makes image classification process easier, but can also adjust itself in every deep learning layer for the final task, which can reach perfect cooperation between each layer, and can finally improve the classification accuracy greatly. According to statistics, the classification accuracy of deep learning algorithm is 10–15% higher than those of traditional methods.

4.6. *Recognition analysis to each defect type*

This paper not only gives the comparison to previous algorithms, but also gives the identification results of each defect with our proposed algorithm. Recognition results for each type of defect are given in Table 5.

Table 5. Recognition result on test samples.

	SFM	BFM	BJO	CD	BS	NC	Correct Number	Total Number	Accuracy %
SFM	68	0	0	0	0	0	68	68	100
BFM	0	53	0	0	0	0	53	53	100
BJO	0	0	52	0	0	0	52	52	100
CD	0	0	0	54	1	0	54	55	98.18
BS	0	0	0	0	42	3	42	45	93.33
NC	0	0	0	0	1	42	42	43	97.67
Total number							311	316	98.47

As can be seen from Table 5:

(1) The recognition rate of Bulk-Sidewall is the lowest, and it is mainly mistaken for Normal-Cords. By observing the image of the mistaken sample, it is found that the Bulk-Sidewall defect image with the weak edge is very similar to that of Normal-Cords image, which leads to misclassification.
(2) In addition to the Bulk-Sidewall defects, the recognition rates of the rest defect types are relatively higher. This is because the depth CNN is a multi-level network that can abstractly express the defect image layer by layer so that a higher level of features can be used to describe complex defect images. Moreover, the multi-stream network itself has a stronger ability of classification.

5. Conclusions

The detection and classification of universal objects or generalized automatic deployment, e.g. tire defects, is often an ambiguous and challenging task especially in real-world application. Inspired by recent successful approaches, the approach we investigated in the present work, that is, five single CNNs are combined into a MC-CNN to classify tire defects, which is an application of deep learning to real-world industrial tasks. Our MC-CNN improves the recognition rate from an average of 96.78% to 98.47%. None of preprocessing methods is superior in single CNN, but combining them into a MC-CNN enhances robustness to different kinds of noise and achieves a higher recognition rate than any other methods above. This work would provide practical usefulness to both researchers and practitioners in various industrial fields.

Acknowledgments

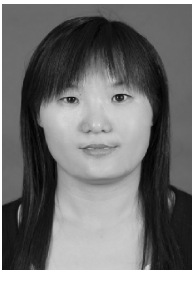
This work was supported by the National Natural Science Foundation of China No. 61472196, by the Shandong Provincial Natural Science Foundation under Grant No. ZR2014FL021, by the Applied Basic Research Project of Qingdao Grant No. 15-9-1-83-JCH and by the Doctoral Found of QUST under Grant No. 010022671.

References

1. S. Albanie, S. Ehrhardt and J. F. Henriques, Stopping GAN Violence: Generative Unadversarial Networks (2017).
2. S. V. Alekseenko, S. N. Toporova and P. Y. Shkorbatova, Neuronal connections of eye-dominance columns in the cat cerebral cortex after monocular deprivation, *Neurosci. Behav. Physiol.* **38**(7) (2008) 669.
3. X. L. Bai, Y. M. Fang, W. S. Lin, L. P. Wang and B. F. Ju, Saliency-based defect detection in industrial images by using phase spectrum industrial informatics, *IEEE Trans. Ind. Inf.* **10**(4) (2014) 2135–2145.
4. A. Bohi, D. Prandi, V. Guis *et al.*, Fourier descriptors based on the structure of the human primary visual cortex with applications to object recognition, *J. Math. Imaging Vis.* (2016) 1–17.
5. L. Brocki and K. Marasek, Deep belief neural networks and bidirectional long-short term memory hybrid for speech recognition, *Arch. Acoust.* **40**(2) (2015) 191–195.
6. D. P. Buxhoeveden and M. F. Casanova, The minicolumn hypothesis in neuroscience, *Brain* **125**(5) (2002) 935–951.
7. X. Cui, Y. Liu and C. Wang, Defect automatic detection for tire X-ray images using inverse transformation of principal component residual, *Int. Conf. Artificial Intelligence & Pattern Recognition*, 2016, pp. 1–8.
8. H. Feng, Z. G. Jiang, F. Y. Xie, P. Yang, J. Shi and L. Chen, Automatic fastener classification and defect detection in vision-based railway inspection systems, *IEEE Trans. Instrum. Meas.* **63**(4) (2014) 877–888.
9. R. Girshick, J. Donahue, T. Darrell *et al.*, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. Computer Vision and Pattern Recognition* (2014).
10. Q. Guo, C. Zhang, H. Liu *et al.*, Defect detection in tire X-Ray images using weighted texture dissimilarity, *J. Sens.* **2016** (2016) 1–12
11. G. H. Hu, Automated defect detection in textured surfaces using optimal elliptical Gabor filters, *Optik — Int. J. Light Electron Opt.* **126**(14) (2015) 1331–1340.
12. J. J. Ji and C. Ye, The automatic detection technology for the surface defects of automobile engine cylinder, *Key Eng. Mater.* **693** (2016) 1458–1465.
13. Jifeng Dai, Haozhi Qi, Yuwen Xiong *et al.*, Wei Deformable Convolutional Networks, arXiv:1703.06211, 2017.
14. A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *Proc. Neural Information Processing System*, 2012.
15. M. Längkvist, L. Karlsson and A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, *Pattern Recognit. Lett.* **42** (2014) 11–24.
16. H. V. Scheel, Recommendations for implementing the strategic initiative GERMAN INDUSTRIE 4.0 Final Report of the Industrie 4.0 Working Group, July 2014.
17. Y. Li, W. Zhao and J. Pan, Deformable patterned fabric defect detection with Fisher criterion-based deep learning, *IEEE Trans. Autom. Sci. Eng.* **14**(2) (2017) 1256–1264.
18. Q. Luo and Y. He, A cost-effective and automatic surface defect inspection system for hot-rolled flat steel, *Robot. Comput. Integr. Manuf.* **38** (2016) 16–30.
19. J. Mutch, D. G. Lowe, Object class recognition and localization using sparse features with limited receptive fields, *Int. J. Comput. Vis.* **80**(1) (2008) 45–57.
20. M. Namdar, J. Adamowski, H. Saadat *et al.*, Land-use and land-cover classification in semi-arid regions using independent component analysis (ICA) and expert classification, *Int. J. Remote Sens.* **35**(24) (2014) 8057–8073.

21. J. H. Oh, W. S. Kim, C. H. Han and M. H. Park, Defect detection of TFT-LCD image using adapted contrast sensitivity function and wavelet transform, *IEICE Trans. Electron.* **E90-C** (2007) 2131–2135.
22. A. Radford, L. Metz and S. Metz, Unsupervised representation learning with deep convolutional generative adversarial networks, *Comput. Sci.* (2015).
23. S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, arXiv:1506.01497.
24. P. Romeu, F. Zamora-Martínez, P. Botella-Rocamora *et al.*, Stacked denoising autoencoders for short-term time series forecasting, *Artificial Neural Networks* (Springer International Publishing, 2015), pp. 463–486.
25. O. Russakovsky, J. Deng, H. Su *et al.*, Imagenet large scale visual recognition challenge, *Int. J. Comput. Vis.* **115**(3) (2015) 211–252.
26. J. Shi, S. Zhou, X. Liu *et al.*, Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset, *Neurocomputing* **194**(19) (2016) 87–94.
27. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *Proc. Int. Conf. Learning Representation* (2015).
28. S. Lazebnik, C. Schmid and J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, *2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.
29. J. Wang, J. Yang, K. Yu *et al.*, Locality-constrained linear coding for image classification, *IEEE Computer Vision and Pattern Recognition* (2010) 3360–3367.
30. M. Win, A. R. Bushroa, M. A. Hassan, N. M. Hilman and A. Ide-Ektessabi, A contrast adjustment thresholding method for surface defect detection based on mesoscopy, *IEEE Trans. Ind. Inf.* **11**(3) (2015) 642–649.
31. J. M. Wolterink, T. Leiner, M. A. Viergever *et al.*, Generative adversarial networks for noise reduction in low-dose CT, *IEEE Trans. Med. Imag.* (2017).
32. L. Xu and Q. Huang, Modeling the interactions among neighboring nanostructures for local feature characterization and defect detection, *IEEE Trans. Autom. Sci. Eng.* **9**(4) (2012) 745–754.
33. L. Xu and Q. Huang, EM estimation of nanostructure interactions with incomplete feature measurement and its tailored space filling designs, *IEEE Trans. Autom. Sci. Eng.* **10**(3) (2013) 579–587.
34. J. Yang, K. Yu, Y. Gong *et al.*, Linear spatial pyramid matching using sparse coding for image classification, *IEEE Conf. Computer Vision Pattern Recognition FL* (2009) 1794–1801.
35. S. Yeung, A. Kannan, Y. Dauphin *et al.*, Tackling Over-pruning in Variational Autoencoders (2017).
36. Y. Zhang, Research on Nondestructive Tire Defect Detection using Computer Vision Methods, Ph.D. dissertation, Qingdao University of Science and Technology China.
37. Y. Zhang, D. Lefebvere and Q. Li, Automatic detection of defects in tire radiographic images, *IEEE Trans. Autom. Sci. Eng.*, DOI: 10.1109/TASE.2015.2469594.
38. Y. Zhang, T. Li and Q. Li, Detection of foreign bodies and bubble defects in tire radiography images based on total variation and edge detection, *Chin. Phys. Lett.* **30**(8) (2013).
39. Y. Zhang, T. Li and Q. Li, Defect detection for tire laser shearography image using curvelet transform based edge detector, *Opt. Laser Technol.* **47** (2013) 64–71.
40. Y. Zhang, Y. Sidibé, G. Maze, F. Leon, F. Druaux and D. Lefebvre, Detection of damages in underwater metal plate using acoustic inverse scattering and image processing methods, *Appl. Acoust.* **103** (2016) 110–121.

41. Z. Zhao, L. Jiao, J. Zhao, J. Gu and J. Zhao, Discriminant deep belief network for high-resolution SAR image classification, *Pattern Recognit.*, doi:10.1016/j.patcog.2016.05.028.
42. Z. D. Tsai and M. H. Perng, Defect detection in periodic patterns using a multi-band-pass filter, *Mach. Vis. Appl.* **24**(3) (2013) 551–565.



Xuehong Cui received her M.S degree in Information Engineering from Qingdao University, Qingdao, China, in 2009. She is currently working towards her Ph.D in the College of Information Science and Technology in Qingdao University of Science and Technology, Qingdao, China. Presently, she is a lecturer in the same department. Her research interests include computer vision, pattern recognition and fault detection.

Yun Liu received his M.S. degree from Tongji University in 1995 and his Ph.D from China University of Mining and Technology, Beijing, in 2001. Currently, he is a professor in the School of Information Science and Technology, Qingdao University of Science and Technology, China. He has published more than 17 papers in international journals.

Yan Zhang received his M.S. and Ph.D. degrees in Automatic Control from Qingdao University of Science and Technology, Qingdao, China, in 2009 and 2014, respectively. He is an Associate Professor of Electrical and Computer Engineering in Qingdao University of Science and Technology, Qingdao, China. His research interests include digital image processing, pattern recognition and fault detection. He has published more than ten papers in international journals.

Chuanxu Wang received his Ph.D. from Ocean University of China, Qingdao, China, in 2007. Currently, he is a Professor in the College of Information Science and Technology, in Qingdao University of Science and Technology, Qingdao, China. His current research interests cover video signal processing and pattern recognition. He has published more than 11 papers in international journals.