Wei Hou[1, 2], Xian Tao[2], Wenzhi Ma[2], De Xu[1, 2]

[1] University of Chinese Academy of Sciences, School of Artificial Intelligence, Beijing, China

[2] Chinese Academy of Sciences, Institute of Automation, Research Center of Precision Sensing and Control, Beijing, China,
e-mail: houwei2018@ia.ac.cn,
e-mail: taoxian2013@ia.ac.cn,
e-mail: mwzdove@sina.com,
e-mail: de.xu@ia.ac.cn

SACNN: Spatial Adversarial Convolutional Neural Network for Textile Defect Detection

DOI: 10.5604/01.3001.0014.3808

Abstract
Constructing textile defect detection systems is significant for quality control in industrial production, but it is costly and laborious to label sufficient detailed samples. This paper proposes a model called 'spatial adversarial convolutional neural network' which tries to solve the problem above by only using the image-level label. It consists of two parts: a feature extractor and feature competition. Firstly, a string of convolutional blocks is used as a feature extractor. After feature extraction, a maximum greedy feature competition is taken among features in the feature layer. The feature competition mechanism can lead the network to converge to the defect location. To evaluate this mechanism, experiments were carried on two datasets. As the training time increases, the model can spontaneously focus on the actual defective location, and is robust towards an unbalanced sample. The classification accuracy of the two datasets can reach more than 98%, and is comparable with the method of labelling samples in detail. Detection results show that defect location from the model is more compact and accurate than in the Grad-CAM method. Experiments show that our model has potential usage in defect detection in an industrial environment.

Key words: *textile defect detection, feature extraction, feature competition, CNN.*

Introduction

Textile defect detection is crucial to quality control in industrial production. Traditionally, the main approach for inspectors to find defects in textile is with their eyes, which is usually harmful to inspectors' eyesight under strong illumination. Thus, intelligent visual systems have been used in many production lines to perform automatic defect detection. These automatic defect detection technologies not only allow people to get rid of laborious and repetitive work but also greatly improve the efficiency of quality control.

Nowadays, existing defect detection methods can be mainly classified into two categories: classical approaches and the recently popular CNN (convolutional neural network) models. Classical approaches mainly rely on human-engineered features and traditional classification methods according to specific application scenarios. Li et al. proposed the Gabor-HOG method to extract direction-aware descriptors, and combined with low-rank matrix decomposition, locate defects on fabric [1]. Huang et al. introduced an unsupervised method using a low-rank representation based on texture prior to the detection of defects on natural surfaces and formulated the detection process as a novel weighted low-rank reconstruction model [2]. Pedro et al. used entropy-based automatic selection of the wavelet decomposition level and wavelet reconstruction to detect defects in textures [3]. Carrera et al. used sparse representations to detect defects in nanoproducts [4]. In [5] a framework based on a template-matching strategy was constructed to detect defects. Although these human-engineered features perform well in specific tasks, feature extraction algorithms often need to be redesigned when extended to other tasks.

Recently, CNN has achieved breakthroughs in many computer vision tasks. Many CNN based methods have also achieved considerable success in defect detection for several application scenarios. In [6] an end-to-end CNN classifier was trained to distinguish defective or defect-free image patches, and it used a sliding window to move over the whole image to classify every patch. Ali et al. extracted features using discrete cosine transform (DCT) and then trained a neural network to classify [7]. Paul et al. used an autoencoder with a perceptual loss function based on structural similarity to segment defective and defect-free pixels directly from images of nanofibrous materials [8]. Lin et al. proposed a class activation mapping (CAM) technique to localise defect regions without using region-level human annotation [9]. Tao et al. proposed a novel deep CNN (DCNN) cascading architecture for performing localisation and detecting defects in insulators [10]. Qin et al. fused the convolutional features in both the encoder and decoder networks and constructed a new DeepCrack network for crack detection [11]. Chen et al. cascaded three DCNN-based detection stages in a coarse-to-fine manner and achieved a high detection rate with good adaptation and robustness in complex environments [12]. In [13], Yu et al. presented a novel 2-stage FCN (fully convolutional networks) framework for surface defect inspection in industrial environments and combined the segmentation and detection tasks.

In the aforementioned CNN-based methods, the defect detection process is constructed as an object detection task or an instance segmentation task. In the object detection task, a tight bounding box needs to be marked around the defect. In the image segmentation task, it is necessary to mark each defective pixel in the image. Examples of defect label are shown in ***Figure 1***.

Since CNN is a data-driven approach, a large number of correctly labeled samples are needed to train a deep CNN model [14]. However, in actual industrial scenarios, it is costly and laborious to acquire sufficient labeled images in industrial environments. Moreover, experienced inspectors rarely label samples, and there are also differences in labelling results among different inspectors. Thus, there is the question: can defects be detected and located only by means of the image-level label (defective or defect-free) without complex label information. In order to deal with the question above, we propose a new model called 'spatial adversarial convolutional neural network' (SACNN). In this model, based on two properties of CNN, we firstly use some stacked convolutional layers as a feature extractor. Features in the higher layer not only have a larger visual field

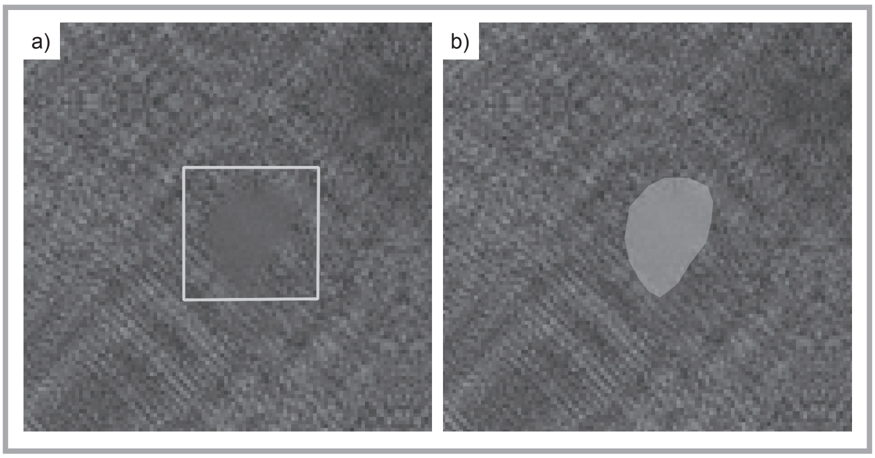

Figure 1. *Examples of defect label: a) bounding box, b) pixel-wise mark.*

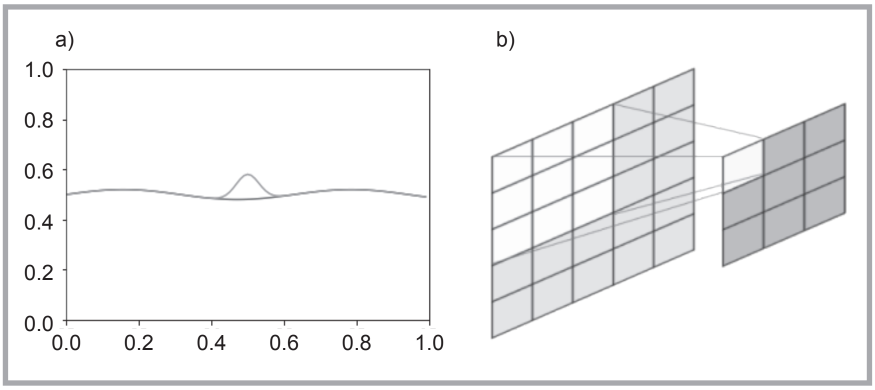

Figure 2. *Two properties of CNN: a) after parameter revision in one step; only a small neighborhood of the output is affected; b) every feature in the higher layer of CNN has its own visual field in the input image.*

that can cover the entire defect, but they also keep the original spatial relation among one another. Then, the last layer of the feature extractor is classified in every position. After that, a maximum greedy competition is taken among these classification results inside a sample, and the model can spontaneously focus on the actual defective position. Experiments on two datasets show that our model has potential usage in defect detection.

Contributions of this work are as follows:

- A spatial adversarial convolutional neural network is proposed for textile defect detection. The model only uses the image-level label (defective or defect-free) and can spontaneously focus on the position of the defect through spatial competition;
- Experiments on two datasets show that our model is robust towards unbalanced samples between the defect-free and the defective, which is faced in real industrial conditions; thus, our model has potential for usage in defect detection.

The remainder of this paper is organised in three sections: the following section introduces the intuitions behind SACNN and describes the structure of our model in detail; in the next section experiments were carried out on two datasets to verify the feasibility of our model and exhaustive discussion is held, and finally conclusions are drawn in last section.

Spatial Adversarial Convolutional Neural Network (SACNN)

In recent years, CNN has achieved fantastic success in various computer vision tasks. While it achieves superior performance, it is hard to interpret because of a lack of decomposability into intuitive and understandable components [15]. The mechanism of CNN has drawn much research attention, and some researchers have shown some insights into CNN. In this section, we will discuss two properties of CNN which are close to our model, and then based on them we will build our SACNN model for defect detection.

Two properties of CNN

CNN is a kind of neural network. Therefore, in theory, CNN can fit any kind of function, and it has a local property. When training a network using the gradient descent method, every step to revise the parameter mainly affects a small neighborhood of the output [16]. Another property is that the neurons in CNN are not fully connected, with every neuron in hidden layers having its own visual field in the input image, which can be considered as a feature extractor [17, 18]. Some object detection models detect objects by sliding bounding boxes over the feature map, then they make a classification of features in the bounding box, as illustrated in ***Figure 2***, and reiterate these two properties as follows:

- Using the gradient descent method to train a CNN, every step to revise the parameter mainly affects a small neighborhood of the output, while other regions change little;
- CNN can be seen as a generic feature extractor, with every feature having its corresponding visual field in the input image.

Defect detection faces the problem that the defect region usually takes a small part of the entire image, and this small part can make the whole image a defect class. Hence, every part of the image should be considered to decide whether it is defective or defect-free. In addition, samples collected are far from balanced. Usually, the number of defect-free samples is much larger. Using the two properties of CNN above, we can conduct a trial to solve the problem of defect detection by proposing spatial adversarial convolutional neural networks.

Spatial Adversarial Convolutional Neural Network (SACNN)

Structure of SACNN is shown in ***Figure 3***. It contains two parts: a feature extractor and feature competition. It uses convolutional layers to extract features, and the spatial information is preserved. Every feature is mapped to a specific region of the input image. In defect-free samples, all features extracted from the input image represent normal. In a defective sample, only a small portion of features represents defect (***Figure 4***). In the training phase, let these features compete with each other, and make the network aware of the defect location. Unlike other methods in the training phase, they not only give a sample's class but also addi-

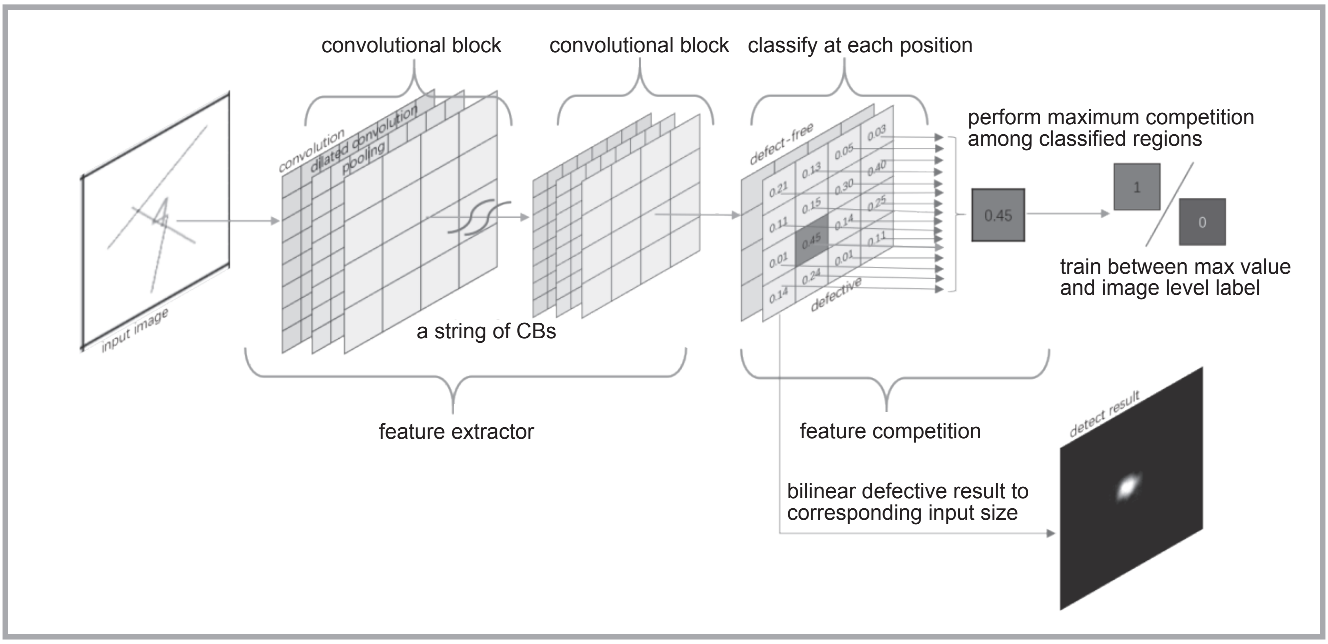

Figure 3. *Structure of SACNN.*

tional supervision information (bounding box or pixel-wise marked) to provide the defect location. The SACNN only needs an image-level label (defect-free or defective) of samples, not complex label information.

Feature extractor

This consists of a string of CBs (convolutional blocks). Every block consists of a convolutional layer, dilated convolutional layer and maximum pooling layer, followed by a ReLU (Rectified Linear Unit) activation function. When features pass through a CB, the size of them will be halved because of pooling. For convenience of expression, a feature pixel in the feature map of the output of a CB is denoted as a pixel. The number of CBs is decided by the visual field of a pixel. And the visual field of a pixel in the last CB's feature map should be approximately equal to the size of the actual defect. Therefore, the number of CBs is decided by the size of defects. This part can be trained from scratch in an end-to-end way. Also, a pretrained network can be used as a feature extractor. This part can be expressed as ***Equation (1)***.

$$F_{(s,t)}=C\odot(C\odot(C\odot(...\ (C\odot(IM_{(i,j)}))...\ ))) \quad (1)$$

where, $IM_{(i,j)}$ denotes the input image with a height of i pixels and width of j pixels, C a CB, $\odot$ the operation between the CB and input, and $F_{(s,t)}$ denotes the output of the feature extractor with a height of s pixels and width of t pixels.

Feature competition

After extracting features, feature competition takes place. The main process is as follows: 1) Classify the output of the feature extractor's every pixel into two classes (0 denotes a defect-free pixel and 1 a defective pixel) using softmax. The reason for using softmax other than the sigmoid function is that it is easier to expand to classify different defective types in the future. After classification, every pixel gets a probability of being defective and of being defect-free, which means their sum is equal to 1. And two layers are obtained, one representing the probability of being defect-free, and the other of being defective. As the sum of two layers at every pixel is equal to 1, either of them provides sufficient information. 2) In the defective layer, let all pixels compete with each other, and the pixel that has the maximum probability wins out. This pixel's probability is considered as the input image-level label. This process is called spatial adversarial, because the pixel chosen is the maximum pixel in the whole defective layer of the input image and can be mapped to a specific area in the input image. 3) Train the network between the pixel's value and the image level label using cross-entropy loss. This process can be expressed as ***Equations (2)***, ***(3)*** and ***(4)***,

$$l^{k*} = max(softmax(F^k_{(s,t)})) \quad (2)$$

$$loss^k = -(l^k \times \log(l^{k*}) + (1 - l^k) \times \log(1 - l^{k*})) \quad (3)$$

$$tloss = \Sigma\ loss^k \quad k = 1, 2, 3,..., n \quad (4)$$

where $tloss$ denotes the total loss of a batch with n samples, $loss^k$ – the weighted cross-entropy loss of a sample, l^k – the true label of sample k, and l^{k*} denotes the network's label produced.

Through feature competition, SACNN can focus on the defective region automatically. This phenomenon can be explained by the properties of CNN discussed above. Property 2 makes every pixel have a visual field approximately the same size as the defect's . This property is used in many detection tasks and is easy to understand. Property 1 makes feature competition lead the network to the true defective region. This is the core idea of this paper, which can be explained as follows. In the above, feature competition takes place among the defective layer. In a defect-free sample, since every pixel is normal, the max-valued pixel should be suppressed during training. And decreasing this pixel can lead the model converging to 0 probability for all pixels in the defect-free sample. In a defective sample, the situation is much complex because the max-valued pixel may not belong to the actual defect region. The max-valued pixel is considered as a defective probability of the whole sample image, but it may not be the real defective pixel. This false-positive pixel may lead to a training error. But in other samples there exist more regions similar to this false positive pixel that are correctly classified as defect-free. Thus, although the model is revised by a false positive, it will be corrected by other

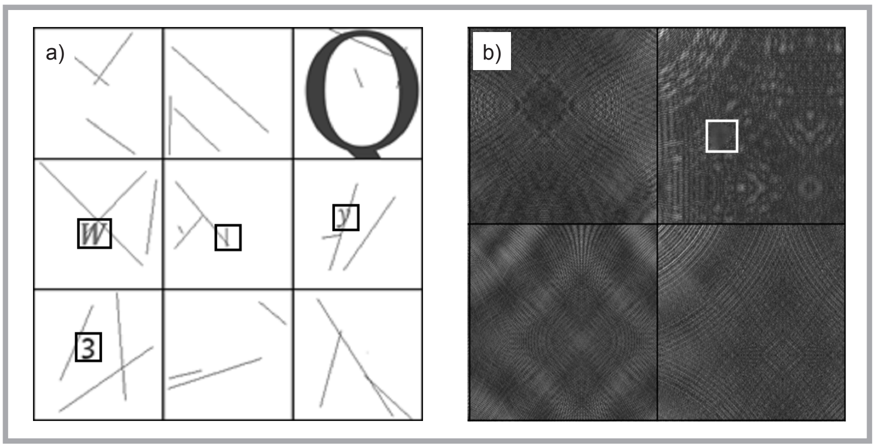

Figure 4. Samples of two datasets: a) character detection dataset, b) textile dataset. Bounding boxes in a) & b) show the position of the real defect, but this bounding box information is not used in training SACNN.

more similar regions. This is where property 1 takes effect: this wrong training step is revised by the normal samples, because a similar region maybe exists in defect-free samples, and the number of defect-free samples is much larger than the number of defective samples. Thus, there is competition among the pixels in the feature map, which will lead to the network fluctuating and slowly converging.

In the testing phase, testing images pass through the feature extractor, and every pixel in the output of the feature extractor is classified as defect-free or defective. Because the sum of the probability of being defective or defect-free is 1, any layer holds the whole information. In consideration of the defect detection task, a defective layer is used, which is extracted and resized to the corresponding test image's size using bilinear interpolation. Then we get the probability of there being a defect at each pixel of the test image. A simple threshold of 0.5 can be used to distinguish between the defect-free and the defective, or some more sophisticated methods such as Otsu [15] and non-maximum suppression [16] can be used to locate the defect position. With the help of the visual method in the experiment section, the core idea that feature competition can lead the network to converge towards the defective region will be shown, and results will support our spatial adversarial idea. Experiments in the next section show that our SACNN model works for two datasets.

Experiments and discussion

In this section, experiments were conducted to show that SACNN can spontaneously detect a defective region, which supports our spatial adversarial idea. First, two datasets are introduced, and then details of the implementation are given. Finally, the experiment results and comparisons are presented.

Datasets

The model is tested on two datasets. One is a fabricated character detection dataset, and the other is a textile dataset. ***Figure 4*** shows two samples of the datasets. The character detection dataset is used to prove the correction of the spatial adversarial mechanism. Each sample is an image with a size of 136 × 136 pixels. It has two classes: having a character or without any character. We consider character a defect, which can locate any position of the image. In order to increase the variety of samples, we add noise to them, like some line segments and a much larger character. The font size of a character is 32 × 32, and its font type is randomly chosen from those in the operating system (about 40 font types). The textile dataset is from one of the DAGM 2007 contest datasets. Each sample is a 512 × 512 pixel image. The dataset has 575 training samples and 575 test samples. The dataset also provides the mask of the defective region used for training, but we do not use it during training. And the sample number in each class in the dataset is far from balanced; the number of defect-free samples is almost five times that of defective samples.

Implementation details

The PyTorch framework is used to implement our model. The code is shared at https://github.com/yjphhw/SACNN. In training the character detection dataset, 3 convolutional blocks cascade together as a feature extractor. Every epoch has 100,000 samples, where the batch size is 64; the optimiser is a stochastic gradient descent (SGD), and the learning rate is 0.01. In training the textile dataset, five convolutional blocks cascade together as a feature extractor, where Batch size is

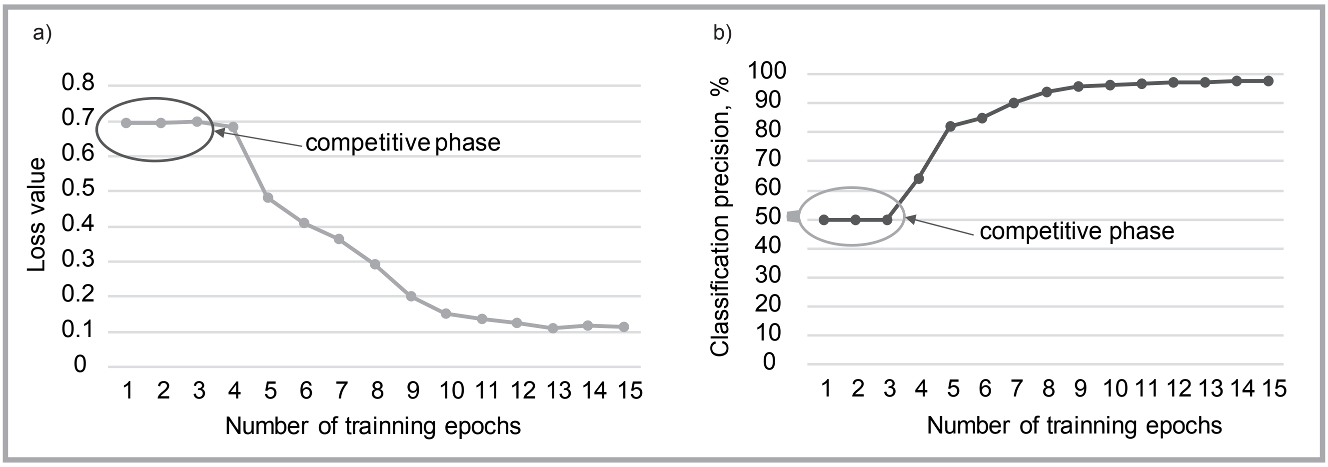

Figure 5. Competitive phase when training SACNN for the character detection dataset (textile dataset also shows similar behaviour). At the beginning of the training, both the loss value a) and classification precision b) fluctuate because of competition among features.

16; the optimiser is SGD, and the learning rate is 0.01. In the initial phase of training, the network fluctuates and is hard to converge. The reason is that the network may choose the wrong defective region in the beginning and needs time to allow the true defective region to win out. This situation is well explained and predicted by our model in prior sections and is shown in ***Figure 5***. In ***Figure 5.a***, the training loss does not decrease in the first three epochs, because feature competition takes place. In a defective sample, the max-valued pixel may not be the truly defective one, hence the loss of the model will fluctuate. After the first three epochs, the model learns some truly defective pixels, and the loss of the model starts to decrease. This phenomenon is supported by ***Figure 7***. The classification precision also takes a similar action to the loss in ***Figure 5.b***. For the textile dataset, the model undergoes a similar competitive phase during training.

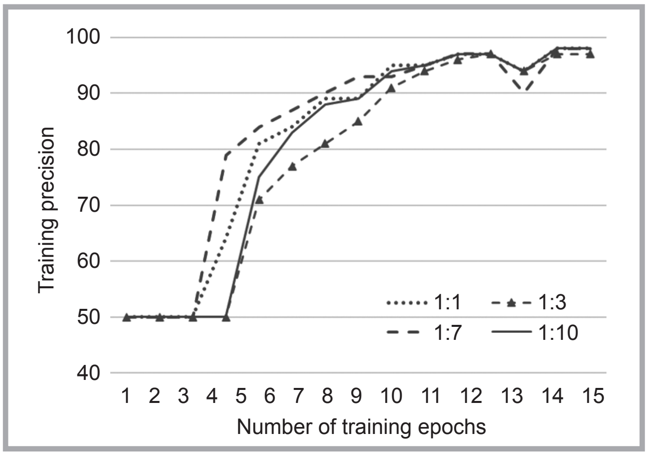

Figure 6. *Precision under different sample ratios during training SACNN for the character detection dataset.*

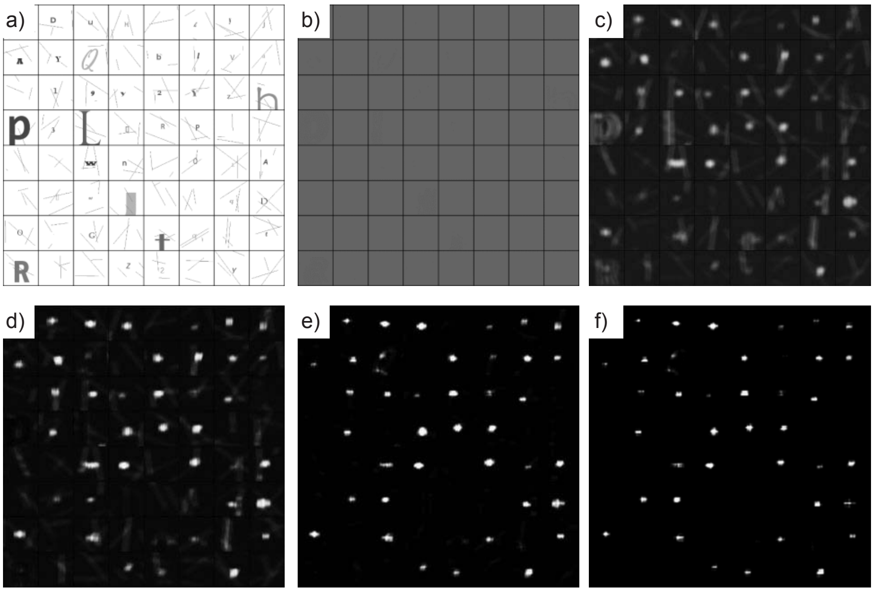

Figure 7. *SACNN convergence to the location of character: a) a batch of test samples in the character detection dataset, b) after epoch 1, locations on the feature map are in fierce competition, c) after epoch 4, some locations gain an advantage from competition, d-f) location of character wins out through competition.*

Experiment results

Classification result: two results can be produced from our SACNN model in ***Figure 2***. One is directly from the network output, being the class of the input image, whether the image is defective or defect-free. The character detection dataset is used to validate the effectivity of the model. ***Table 1*** shows that the accuracy of the model for the dataset is 98.4%, which means the spatial adversarial mechanism proposed is effective. Then, the model is trained for the textile dataset, and the accuracy of defect classification reaches 98.6%. Since the model is trained by the image level class, it is essentially a weakly supervised method. There are no similar methods which research datasets. Therefore, two other methods are compared: one is a supervised method from [21], and the other is an unsupervised method from [22]. In [21] Kim et al. clipped the images into patches and labelled the class of each patch. A fully supervised CNN is used for classification as defect or defect-free. The accuracy of the defect classification is 99.8%, while that of ours is only 1.2% lower. The accuracy of the model is comparable to the supervised method, and is higher than the unsupervised method [22] by 7.1%.

In addition, the number of defective and defect-free samples is unbalanced. When training the model, the ratio of defective and defect-free samples is roughly 1:5

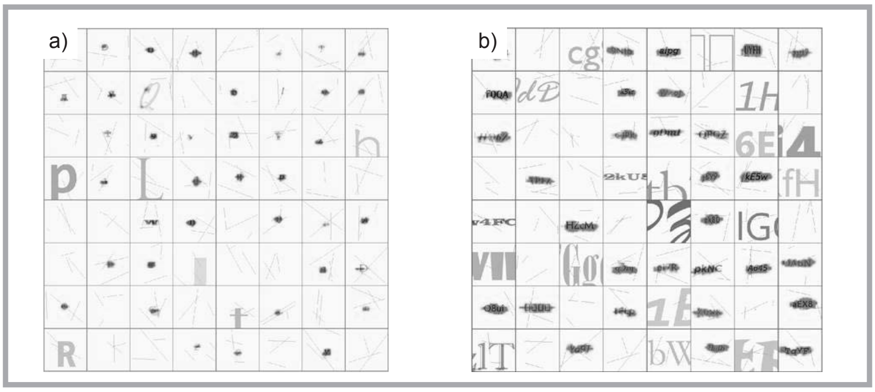

Figure 8. *Overlay of detection results on test samples: a) test samples with only one character, b) test samples with multiple characters.*

Table 1. *Classification accuracy.*

Dataset	SACNN training from scratch		Fine-tuning results in [21]		Weibull features [22]
	Training epochs	Test accuracy	Training epochs	Test accuracy	Test accuracy
Character detection dataset	20	98.4%	–	–	–
Textile dataset	1760	98.6%	3	99.8%	91.5%

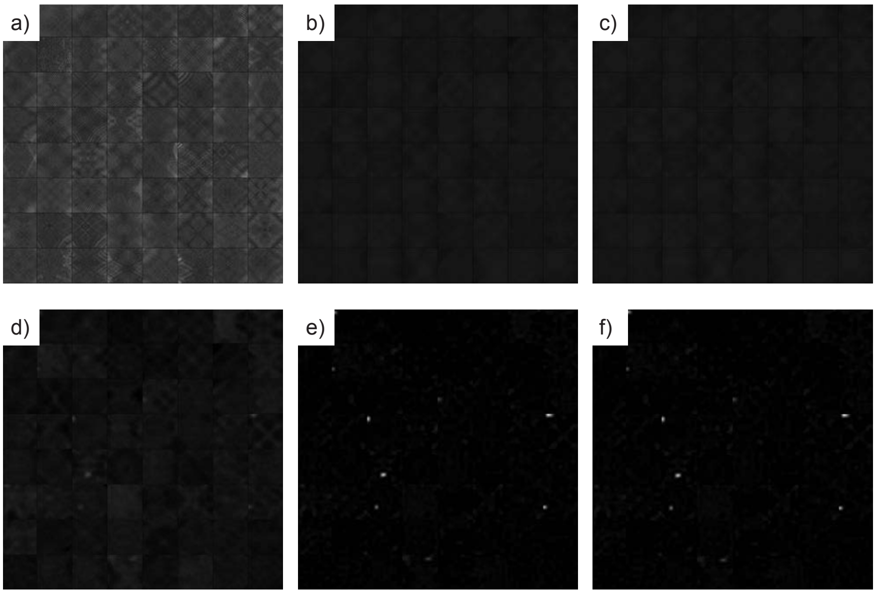

Figure 9. *SACNN convergence to the location of the textile defect: a) a batch of test samples in the textile dataset, b) after epoch 100, locations on the feature map are in fierce competition, c) after epoch 1200, some locations gain an advantage from competition, d-f) location of defect wins out.*

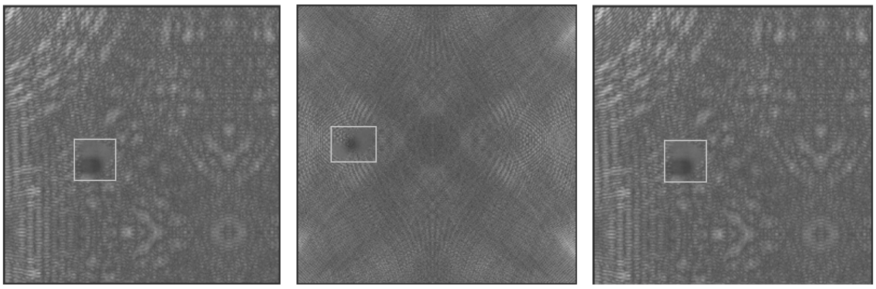

Figure 10. *Enlarged detection result for textile dataset. The detection results are overlaid on the test samples. The bounding box is the ground truth. The red area is the located result.*

in training samples. Experiments under different ratios were carried out for the character detection dataset. As can be seen from ***Figure 6***, SACNN is robust towards the unbalanced sample set. After 15 epochs of training, the accuracy is almost the same when the ratio of defective samples to defect-free samples is greater than 1:10. This implies the model can tolerate some level of sample imbalance. The classification result is from the output of the model's end-to-end training. Based on the classification result, the defect location result can be drawn from it.

Defect location result: Another result is from the output of the softmax layer, which classifies every pixel in the feature map into two classes. Every pixel in the feature map represents a region in the input image. Since the input image only changes in scale, the relative position is the not changes in the feature map. The defective layer is selected and resized to the size of the input image using bilinear interpolation, and this defective probability map could show the possible location of the defect. In ***Figure 6***, training accuracy is improved over time, and it is also true for the accuracy of defect location. SACNN can slowly converge to the location of the defect. This process is be visually explained by ***Figures 7*** and ***9***. ***Figure 7*** shows character detection accuracy changes at different training epochs. When the training starts, the network hardly selects the true position of the defect, and will fluctuate. Over time, the network can slowly focus on the real defect location through competition, which is also true for textile defect detection; similar results are shown in ***Figures 9*** and ***10***. The defect location can be detected after 1760 epochs of training, and the detection results are almost located inside the boundary of the real defect location.

In ***Figure 8***, detection results overlay test samples, showing that the model finds the right location no matter whether there is one or several characters in a sample. Besides, it comprises the same cost of detection, because the defective layer is produced without extra processing. After acquiring the defective probability map, threshold 0.5 can be selected to binarise the map to obtain the defect location result.

The defect location detection result is compared with Gradient-weighted Class Activation Mapping (Grad-CAM) [17]. It is a technique for visual explanations of decisions from CNN. It can locate which part of the sample contributes to the classification result. Like SACNN, Grad-CAM also uses an image-level label to train the CNN model, and uses the intermediate result to show the important parts which contribute to the classification result. We compare SACNN with Grad-CAM for textile datasets. VGG19 was used for fine-tuning in Grad-CAM. ***Figure 11*** gives the results of comparison for the textile dataset. ***Figure 11.a*** shows three defective samples from the textile dataset, and ground true bounding boxes are marked. ***Figure 11.b*** gives visual results from Grad-CAM, showing which part contributes to classification. ***Figure 11.c*** displays detection results from SACNN. From the comparison results, we can conclude that both SACNN and Grad-CAM can detect the correct defective location; but SACNN gets a more compact result.

Conclusions

In textile defect detection, we are faced with unbalanced and insufficiently detailed labelled data. This paper proposes a model called SACNN, designed based on the intuition of two features of CNN. It uses a string of convolutional blocks to first extract features and then classifies the image. Moreover, it can detect the location of defects through the spatial adversarial mechanism. Experiments on two datasets show that SACNN can achieve an accuracy of defect classification close to that of the supervised method, and is robust towards imbalanced datasets. Compared to Grad-CAM, SACNN can spontaneously focus on defect location through feature competition, and can obtain more compact and accurate results. In the future, we will use the model proposed in actual industrial environments and strengthen it to adapt to defects at different scales and of various types.

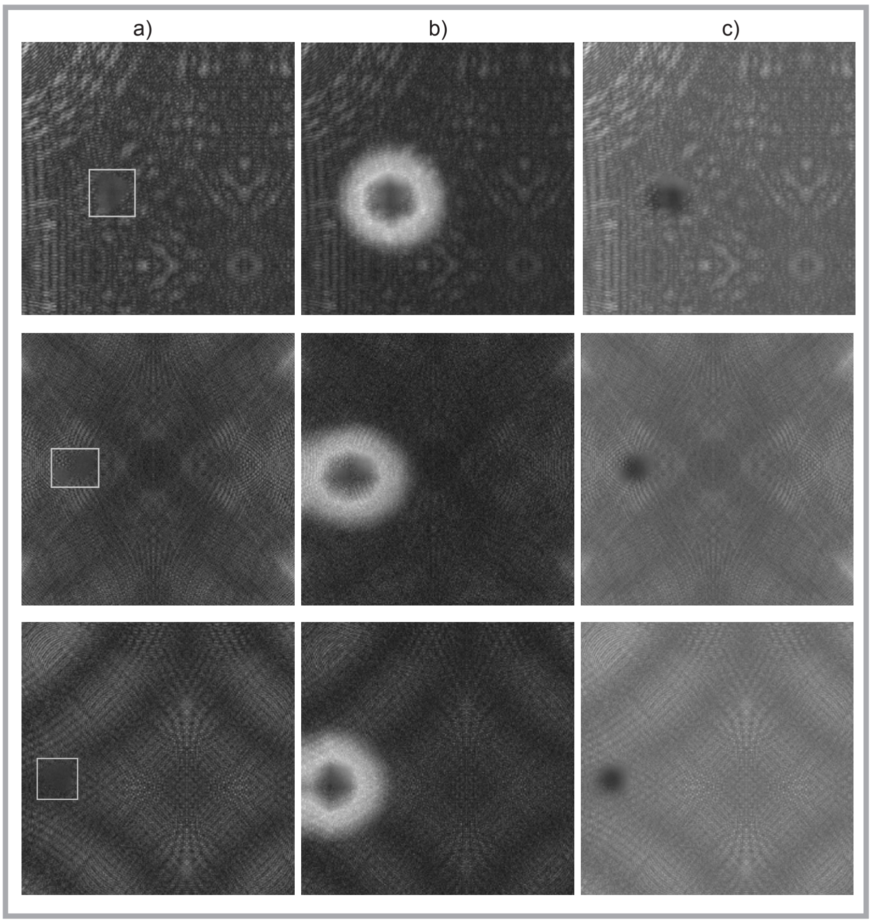

Figure 11. Comparison of SACNN and Grad-CAM for the textile dataset: a) test images with ground truth bounding box, b) visual results from GAD-CAM method, c) detection results from SACNN.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61703399 and 62066004.

References

1. Chunlei Li, Guangshuai Gao, Zhoufeng Liu, Di Huang, Sheng Liu, Miao Yu. Defect Detection for Patterned Fabric Images Based on GHOG and Low-Rank Decomposition. 2017, [Online]. Available: https://arxiv.org/abs/1702.05555.
2. Qizi Huang, Hong Zhang, Xiangrong Zeng, Wenwei Huang. Automatic Visual Defect Detection Using Texture Prior and Low-Rank Representation. *IEEE Access* 2018;6: 37965-37976.
3. Navarro Pedro J, Isla Carlos Fernández, Alcover Pedro María, Suardíaz Juan. Defect Detection in Textures through the Use of Entropy as a Means for Automatically Selecting the Wavelet Decomposition Level. Sensors (Basel, Switzerland) 2016; 16, 8: 1178.
4. Carrera Diego, Manganini Fabio, Boracchi Giacomo, Lanzarone Ettore. Defect Detection in SEM Images of Nanofibrous Materials. *IEEE Transactions on Industrial Informatics* 2017; 13, 2: 551-561.
5. Hui Kong, Jian Yang, Zhihua Chen. Accurate and Efficient Inspection of Speckle and Scratch Defects on Surfaces of Planar Products. *IEEE Transactions on Industrial Informatics* 2017; 13, 4:1855-1865.
6. Haiyong Chen, Yue Pang, Qidi Hu, Kun Liu. Solar Cell Surface Defect Inspection Based on Multispectral Convolutional Neural Network. *Journal of Intelligent Manufacturing* 2018; 1-16.
7. Ali Rebhi, Issam Benmhammed, Sabeur Abid, Farhat Fnaiech. Fabric Defect Detection Using Local Homogeneity Analysis and Neural Network. *Journal of Photonics* 2015; 2015: 1-9.
8. Bergmann P, Lowe S, Fauser M, Sattlegger David, Carsten Steger. Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. 2018, [Online]. Available: http://arxiv.org/abs/1807.02011.
9. Hui Lin, Bin Li, Xinggang Wang, Yufeng Shu, Shuanglong Niu. Automated Defect Inspection of LED Chip Using Deep Convolutional Neural Network. *Journal of Intelligent Manufacturing* 2018. DOI: 10.1007/s10845-018-1415-x.
10. Xian Tao, Dapeng Zhang, Zihao Wang, Xilong Liu, Hongyan Zhang, De Xu. Detection of Power Line Insulator Defects Using Aerial Images Analyzed with Convolutional Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-13, Oct 2018.
11. Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, Song Wang. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Transactions on Image Processing* 2019; 28, 3: 1498-1512.
12. Junwen Chen, Zhigang Liu, Hongrui Wang, Alfredo Núñez, Zhiwei Han. Automatic Defect Detection of Fasteners on the Catenary Support Device Using Deep Convolutional Neural Network. *IEEE Transactions on Instrumentation and Measurement* 2018; 67, 2: 257-269.
13. Zhiyang Yu, Xiaojun Wu, Xiaodong Gu. Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment. *International Conference on Computer Vision Systems* 2017: Oct, 417-426.
14. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
15. Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst., Man, Cybern*. 1979; SMC-9(1): 62-66.
16. Alexander Neubeck, Luc J. Van Gool. Efficient Non-Maximum Suppression. *18th International Conference on Pattern Recognition* (ICPR 2006) 2006; 20-24 Aug, Hong Kong, China.
17. Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. 2017 *IEEE International Conference on Computer Vision* (ICCV), Venice, Italy, 2018: 618-626.
18. Gardner WA. Learning Characteristics of Stochastic-Gradient-Descent Algorithms: A General Study, Analysis, and Critique. *Signal Processing* 1984; 6, 2: 113-133.
19. Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems, 2012.
20. Hertel Lars, Barth Erhardt, Käster Thomas, Martinetz Thomas. Deep Convolutional Neural Networks as Generic Feature Extractors. *International Symposium on Neural Networks* 2015; pp. 1-4,.
21. Seunghyeon Kim, Wooyoung Kim, Yung-Kyun Noh, Frank C. Park. Transfer Learning for Automated Optical Inspection. *International Joint Conference on Neural Networks IEEE*, 2017.
22. Timm Fabian, Barth Erhardt. Non-Parametric Texture Defect Detection Using Weibull Features. Proc. SPIE 7877, Image Processing: Machine Vision Applications IV, 2011.

Received 07.04.2020 **Reviewed 22.04.2020**