# A Surface Defect Detection Based on Convolutional Neural Network

Xiaojun Wu[1,2], Kai Cao[1], Xiaodong Gu[1]

[1] Harbin Institute of Technology, Shenzhen, Guangdong 518055, China
[2] Shenzhen Key Laboratory for Advanced MotionControl and Modern Automation Equipment
Shenzhen, Guangdong, China
wuxj@hit.edu.cn, 949038575@qq.com, 1290205602@qq.com

**Abstract.** Surface defect detection is a common task in industry production. Generally, designer has to find out a suitable feature to separate defects in the image. The hand-designed feature always changes with different surface properties which lead to weak ability in other datasets. In this paper, we firstly present a general detecting method based on convolutional neural network (CNN) to overcome the common shortcoming. CNN is used to complete image patch classification. And features are automatically exacted in this part. Then, we build a voting mechanism to do a final classification and location. The good performances obtained in both arbitrary textured images and special structure images prove that our algorithm is better than traditional case-by-case detection one. Subsequently, we accelerate algorithm in order to achieve real-time requirements. Finally, multiple scale detection is proposed to get a more detailed locating boundary and a higher accuracy.

**Keywords:** CNN, Defect Inspection.

## 1 Introduction

Visual analysis for product surface is a common computer vision application. Current detection algorithm relies on human-designed features, which are always special and not comprehensive. So it is very difficult to have good portability and often limited by designer's experience. In some high structural texture image and special structure image, it seems difficult to distinguish background and target region. Although feature becomes more and more complex, detection effect is not significantly improved. On the other hand, Deep learning method becomes more and more popular, due to its strong ability in exacting feature. Defect detection is a specific application in industrial detection. In this paper, we introduce an easy surface detection method based on convolutional neutral network (CNN). In the second part of this paper, we briefly introduce the related work. In the third part, we introduce algorithm process, including the single scale detection algorithm, algorithm acceleration and the realization of multi-scale detection algorithm. In the fourth part, we verify the effectiveness of our algorithm on five different data sets, and give the corresponding analysis.

## 2 Related Work

Generally, texture defect detection method can be divided into four main types: statistical, structural, filter based and model based. These algorithms choose hand-designed features as the core. Statistical methods include such as well-known techniques based on histogram [1] and co-occurrence matrices [2], structural approach [3] include such as texture elements extracted method. Filter based approach [4] include such as spatial domain and frequency domain filtering design. And there are also some model based approaches [5-7]. Although feature becomes more and more complex, designer's subjective experience and cognition still impact the effect of algorithm. A more obvious drawback is that almost all methods are only suitable for similar datasets and get bad results when little change occurred in datasets. In recent few years, deep learning has led to very good performance in several areas, such as visual recognition, speech recognition and natural language processing. Le Cun et al establishing the modern framework of CNN called LeNet-5[8]. Since 2006, many methods have been developed. AlexNet[9] is similar to LeNet-5 but with a deeper structure. Then several works are proposed to improve its performance such as ZFNet[10], VGGNet[11], GoogleNet[12] and ResNet[13]. All these networks have been proved to receive a decent object detection results on ILSVRC challenge. One of the most famous objects proposal based CNN detector is region-based CNN(R-CNN) [14]. CNN is proved more effective for different image sets than traditional method.

In this paper, we introduce an easy surface detection method based on CNN. Image features are automatic extracted. In the training stage, all we need to do is to prepare training data and labels. CNN extracts feature according to input characteristics and the labels. After that a voting mechanism is introduced for location. Then, we accelerate algorithm by using sliding window on feature map. When running time is cut down, we propose a multi-scale detection method by using two networks to obtain a more elaborate boundary and a higher accuracy. The follow-up experiment proves our algorithm has good detection performance on some texture image and some special structural image such as metallic gasket and screw image.

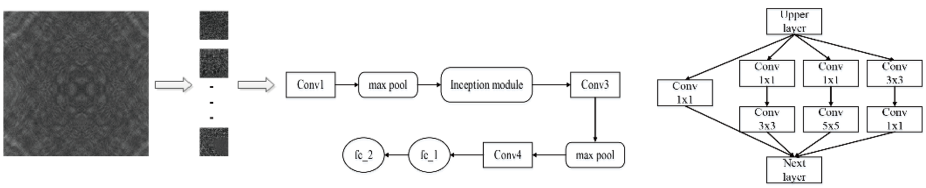
## 3 Methodology

This section presents all the works we have done, including three parts. Firstly, we explain the basic detecting scheme including how to build a convolution neural network (CNN) for image block classification and then how to locate the defect region. The algorithm can be divided into two parts: off-line training and on-line detection. An obvious drawback of CNN is that it needs a lot of time to compute. So referring to R-CNN method, we introduce a method to speed up the algorithm. And in the third part, we use two different networks to refine the results.

### 3.1 Detection Scheme

We use image blocks as input and design a CNN to classify image blocks. After this step, most background regions are excluded. Then, we use voting mechanism to exclude the interference of the background image blocks which are difficult to separate, realizing classification and location.

**Network Structure Design.** In this part, we design a network to extract feature and classify each image block. Fig.1 illustrates our CNN structure, it takes image blocks as input (the block size is 64× 64). Firstly, network processes the block with several convolution and pooling layers to automatically extracted features. Then, inception layer raised from GoogLeNet[12] is used to rich network performance. Inception layer is composed by four branches, three of them are convolution kernels with size 1 × 1、3× 3、5× 5 and the rest branch is a 3× 3 pooling kernel (shown in Fig.1 b). Each branch extracts features from the upper layer. Then features are concatenated together for the next layer. Compared with single layer, four branches make feature information richer for classification. Our specific parameters of CNN are shown in Tab1.




(a) Base structure (b) Inception module

**Fig. 1.** CNN structure

**Table 1.** CNN parameters of layers

| Layer ID | Layer Type | Output (C× W× H) | Parameter (#kernel number× output size # stride # padding) |
|---|---|---|---|
| 0 | Input | 3× 64× 64 | |
| 1 | Convolution | 16× 32× 32 | 16× 5× 5 #2 #2 |
| 2 | Pooling | 16× 16× 16 | 2× 2 MAX |
| 3 | Inception | 64× 16× 16 | |
| 4 | Convolution | 128× 8× 8 | 128× 5× 5 #2 #2 |
| 5 | Pooling | 128× 4× 4 | 2× 2 AVE |
| 6 | Convolution | 256× 2× 2 | 256× 3× 3 #1 #2 |
| 7 | Full Connection | 384 | |
| 8 | Full Connection (softmax) | 2 | |

**Voting Mechanism.** During the stage of patch-classification by CNN, we excluded most of the background regions. As whole image can often be divided into hundreds of patches, false detection is inevitable. In the detection process, an image block misclassification will lead to misclassification and location error in the whole image, as Fig2 shows. Here, we propose a voting mechanism to solve this problem. In the process of image segmentation, the stride of sliding window is equal to half of the size of image block, so that two adjacent image blocks have 50% overlapped area and each pixel in the image is included in four image blocks. Four image blocks generate four CNN voting results. In the positioning process, when the defect votes exceed a certain threshold, the pixel is regarded as defect pixel, just as Fig3 shows.

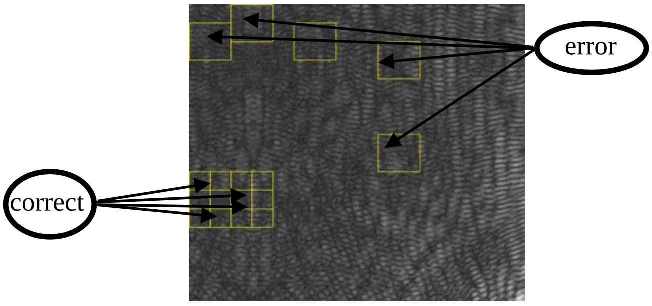



**Fig. 2.** Misclassification after CNN

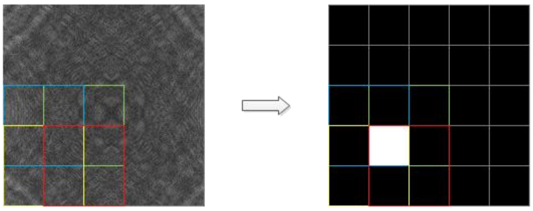


**Fig. 3.** voting process

### 3.2 Algorithm Acceleration

Through the classification and voting mechanism of CNN, the algorithm can effectively realize the defect detection. In the other hand, a large number of data often means slow speed. Large computation is a disadvantage of convolutional neural network. Here, we take some acceleration method to improve the training speed and detection speed. Training speed and detecting speed is about 10 times faster through improvement.

**Training Acceleration.** Training time of CNN is affected by single iteration time and the number of iterations. When network structure is relatively fixed, single iteration time is relatively fixed. So we accelerate the convergence speed by adding a network branch. The structure of the branch is shown in Fig 4. This branch is only used in training to avoid impact on detection results. After adding second branch (loss2 branch), the time required for a single iteration is slightly increased. While the total number of iterations drops rapidly, which lead to a nearly 8 times acceleration.

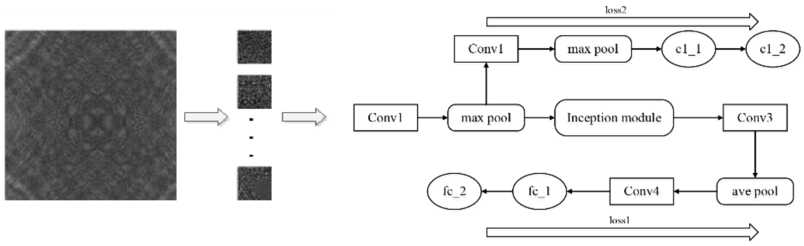



**Fig. 4.** CNN training mode

**Detection Acceleration.** Detection speed is important for a defect detection algorithm, however, it is a weakness of CNN. Inspired by fast R-CNN, we adjust the algorithm in order to obtain faster speed without affecting detection results. Whole image is used to replace image block when detecting. When the network spread to the last convolution layer, we get a larger feature map. Each $2\times 2$ window on feature map corresponds to each original image block. So we make sliding windows on the feature map, the window size is $2\times 2$. Then image blocks are predicted based on the judgment of each sliding window. With this method, we exclude redundant computation (shown in Fig.5) and get an about 17 times acceleration. It takes almost 0.5 seconds with the input size $512\times 512$ on GPU GT640M.

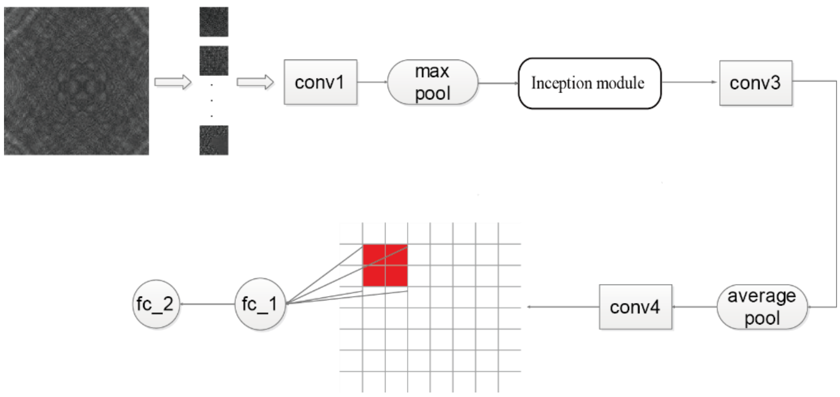



**Fig. 5.** Detecting process with acceleration

### 3.3 Multiple Scale Detection

We add more content for the algorithm to achieve a better detection results. Single block size is difficult to fully extract all defect features. So we put two different networks to get a new prediction. We choose the size of input image $64\times 64$ and $32\times 32$ to train two detection network separately. Score coming from the CNN trained with $64\times 64$ images is multiplied by 0.6 and the score of CNN trained with $32\times 32$ is multiplied by 0.4. With the weight score of the two CNN, we achieve better results on some images. At the same time, by means of weighted score we integrate two detection results to obtain a more precise boundary than before.

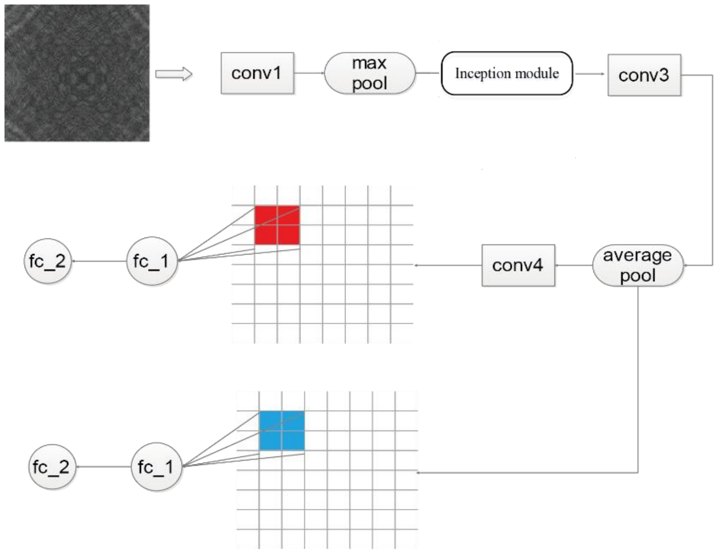



**Fig. 6.** Multiple scale detection process

## 4 Experiment and Analysis

In this section, we show relevant experimental results with different dataset. Experimental process can be roughly divided into three parts: The first part, we present the detection results with single network in some image sets to prove the validity of our algorithm. A part of these sets are provided by the German Association for Pattern Recognition (GAPR), others are collected by us. We compared results between ours and ViDi (software for industrial inspection). In the second part, we describe the results of algorithm acceleration. Finally, we show the effect of multiple scale detection on improving detection accuracy and refining region boundaries.

### 4.1 Detection with Single Network

In this part, we show the detection effect based on single network. We test on 3 different defect types. We train and test our framework on the dataset of DAGM 2007 [16], which representing regional defects, linear defects and point defects and the rest two image sets are collected by us which represent the defects of special structure. We train by SGD with momentum. We use a minibatch size of 128 patches which size is 64× 64 and fixed learning rates of 10-4, the ratio of training defect patches and non-defect patches is 1:1. We use momentum 0.9, weight decay of 10-5. Detailed results are shown in Fig7 and Tab2. It is proved that our single network algorithm has strong detection capability and high applicability on all these image sets.

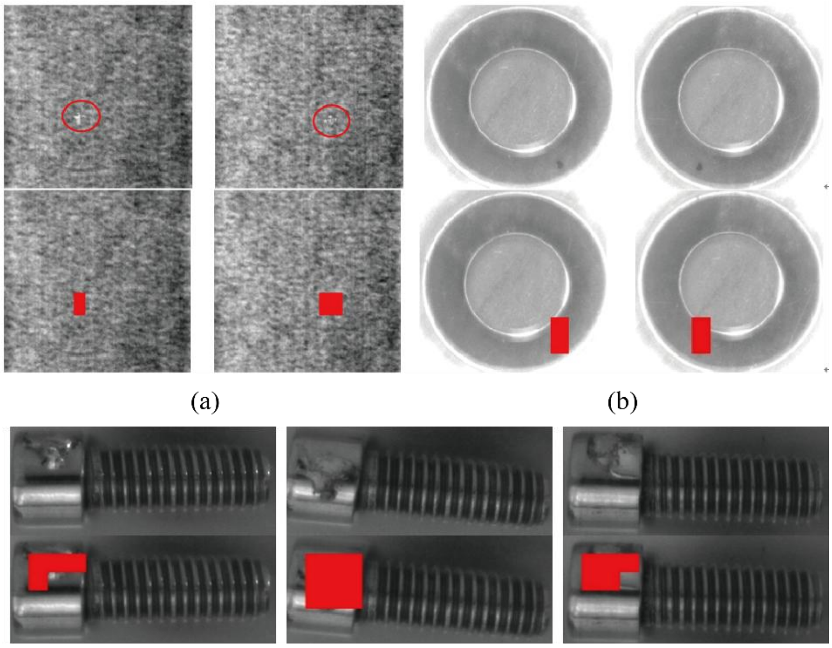



**Fig. 7.** Result images (a) DAGM defects (b) metallic gasket (c) screw image

**Table 2.** Detection result in different image sets

| Data sets | Truth positive | Truth negative | False positive | False negative | Accuracy |
|---|---|---|---|---|---|
| regional defects | 985 | 150 | 0 | 15 | 98.6% |
| linear defects | 949 | 150 | 0 | 51 | 95.6% |
| point defects | 996 | 133 | 17 | 4 | 98.2% |
| metallic gasket | 13 | 24 | 2 | 2 | 89.1% |
| screw image | 18 | 27 | 3 | 2 | 90.0% |

As shown in Fig 8, we compare detection results between our algorithm and ViDi software. ViDi is the first industrial image analysis software based on deep learning. Using the same dataset, thanks to the voting mechanism, when ViDi algorithm fails to detect, we get a good result. Overall, our algorithm has an accuracy of 98.6% better than ViDi (94%).

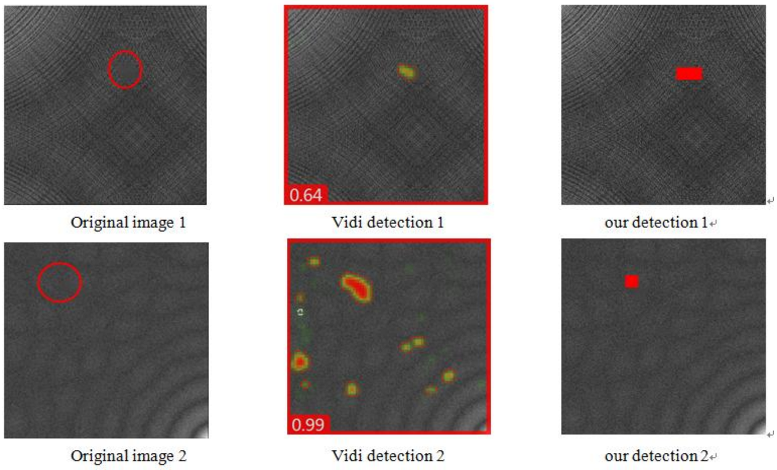
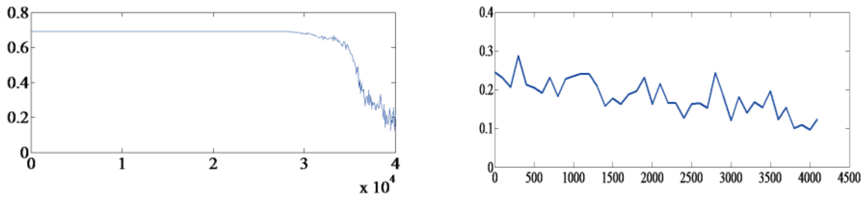



**Fig. 8.** Detection results compared with ViDi

## 4.2 Results of algorithm acceleration

Fig 9 shows acceleration in training process. From DAGM datasets, it is the iterations curve in training process. Before acceleration, CNN takes about 40000 iterations to make loss equal to 0.1. However, CNN only need 4000 iterations after adding loss2 to do the same work, almost 10 times faster. The weight of loss 1 and loss 2 are equal and result do not degrade.




(a) Single loss (b) Double loss

**Fig. 9.** The number of iterations with single loss structure

Tab 3 shows (GPU GT640M) acceleration in detection process. We use sliding window on feature map to replace cutting image. Finally, we get a speed up about 17 times. Data in Tab 3 is counted from the first image set.

**Table 3.** Time consumption comparison

| Method | time-consuming per image (s) |
|---|---|
| cutting image | 10.21 |
| feature map sliding method | 0.5856 |
| Speedup ratio | 17.435 |

### 4.3 Detection with Multiple Network

After detection acceleration, we use two networks to enhance the detection effect. As shown in Fig10, the merged detection gets a better detection result. When both networks hit defect region, bagging CNN shows a more detailed region boundary. And if one or two of the networks fails, merged detection can still get good locating effect. A region detected by two CNNs at the same time is an effective measure to reduce the risk of error detection and for a more detailed boundary. Just like Fig 10 shown.

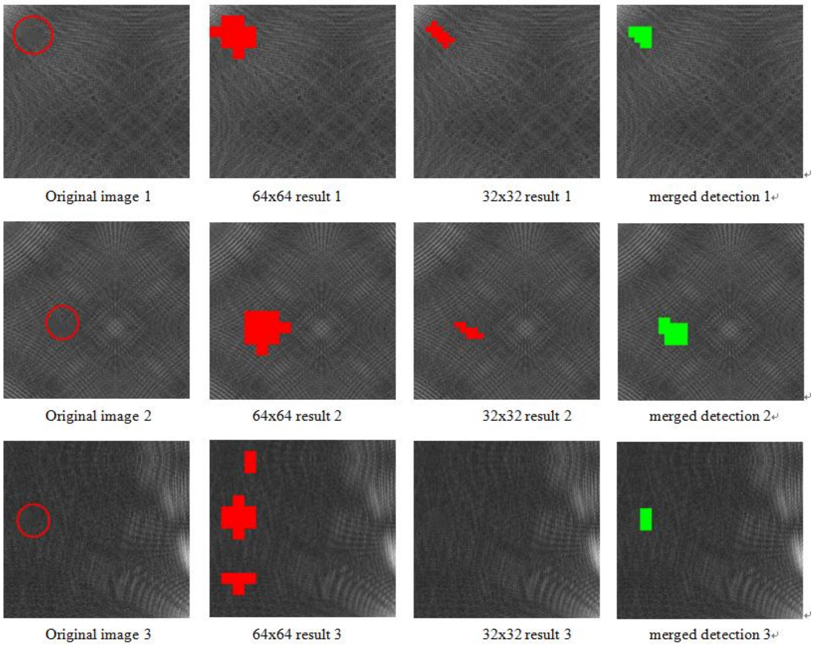



**Fig. 10.** Merged Detections with Multiple Network

## 5 Conclusions

In this paper, we design a feature extraction method by using convolutional neural networks. Solve the problem of weak adaptability caused by artificial interference in the traditional method. At the same time, we use voting mechanism to avoid false detection in the image and realize defect localization. And we get good test results in five different type image sets. By adding a new branch, the training process is about 8 times faster. Then we improve the detection speed almost 17 times by sliding window on the feature map. In order to solve detecting failure in single network, we using two different networks to get higher detection accuracy and more detailed regional boundaries. As to solve rough boundary contour in our algorithm, more researches is needed in the future.

## References


1. Swain, Michael J., and Dana H. Ballard.: "Indexing via color histograms."Active Perception and Robot Vision. Springer Berlin Heidelberg, pp.261-273 (1992).
2. Conners, Richard W., et al.: "Identifying and locating surface defects in wood: Part of an automated lumber processing system." IEEE Transactions on Pattern Analysis and Machine Intelligence 6, pp. 573-583 (1983).
3. Vilnrotter, Felicia M., Ramakant Nevatia, and Keith E. Price.: "Structural analysis of natural textures." IEEE Transactions on Pattern Analysis and Machine Intelligence 1, pp. 76-89 (1986).
4. Jolliffe, Ian.: Principal component analysis. John Wiley & Sons, Ltd (2002).
5. Mandelbrot, Benoit B.: The fractal geometry of nature. Vol. 173. Macmillan (1983).
6. Mao, Jianchang, and Anil K. Jain.: "Texture classification and segmentation using multiresolution simultaneous autoregressive models." *Pattern recognition* 25.2, pp.173-188 (1992).
7. Comer, Mary L., and Edward J. Delp.: "Segmentation of textured images using a multiresolution Gaussian autoregressive model." *IEEE Transactions on Image Processing* 8.3, pp. 408-420 (1999).
8. Le Cun, B. Boser, et al.: "Handwritten digit recognition with a back-propagation network." Advances in neural information processing systems. (1990).
9. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton.: "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. (2012).
10. Zeiler, Matthew D., and Rob Fergus.: "Visualizing and understanding convolutional networks." European Conference on Computer Vision. Springer International Publishing, (2014).
11. Simonyan, Karen, and Andrew Zisserman.: "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv,pp.1409.1556 (2014).
12. Szegedy, Christian, et al.: "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015).
13. He, Kaiming, et al.: "Deep residual learning for image recognition." *arXiv preprint* arXiv:1512.03385 (2015).
14. Girshick, Ross, et al.: "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. (2014).
15. https://hci.iwr.uni-heidelberg.de/node/3616, last accessed 2017/4/10