An Unsupervised-Learning-Based Approach for Automated Defect Inspection on Textured Surfaces

Shuang Mei, Hua Yang, *Member, IEEE,* and Zhouping Yin, *Member, IEEE*

***Abstract*—Automated defect inspection has long been a challenging task especially in industrial applications, where collecting and labeling large amounts of defective samples are usually harsh and impracticable. In this paper, we propose an approach to detect and localize defects with only defect-free samples for model training. This approach is carried out by reconstructing image patches with convolutional denoising autoencoder networks at different Gaussian pyramid levels, and synthesizing detection results from these different resolution channels. Reconstruction residuals of the training patches are used as the indicator for direct pixelwise defect prediction, and the reconstruction residual map generated in each channel is combined to generate the final inspection result. This novel method has two prominent characteristics, which benefit the implementation of automatic defect inspection in practice. First, it is absolutely unsupervised that no human intervention is needed throughout the inspection process. Second, multimodal strategy is utilized in this method to synthesize results from multiple pyramid levels. This strategy is capable of improving the robustness and accuracy of the method. To evaluate this approach, experiments on convergence, noise immunity, and defect inspection accuracy are conducted. Furthermore, comparative tests with some excellent algorithms on actual and simulated data sets are performed. Experimental results demonstrated the effectiveness and superiority of the proposed method on homogeneous and nonregular textured surfaces.**

***Index Terms*—Convolutional denoising autoencoder (CDAE), defect inspection, texture analysis, unsupervised learning, automatic optical inspection.**

I. Introduction

TEXTURED surface inspection aims to identify regions that exhibit dissimilar properties with defect-free areas according to certain criteria such as texture patterns or structures. It commonly occurs in many industrial applications, such as textile inspection [1], wood inspection [2], and ceramic tile inspection [3]. Surface defects are generally inspected by experienced inspectors traditionally. Similar manual approaches are usually time consuming and highly subjective. Therefore, they are still insufficient to meet the needs of modern industrial production. Automated optical inspection uses machine vision techniques for industrial defect inspection and has drawn much attention in recent years, especially for textured surface inspection [4]–[6].

Researches on texture characteristics have important theoretical and practical value, and various methods of texture analysis emerge in an endless stream continuously. Commonly used texture analysis methods can be categorized into four categories according to the utilized image processing techniques. These categories are as follows:

1) statistical methods;
2) structural methods;
3) filter-based methods;
4) model-based methods [7].

Statistical methods usually utilize the grayscale distributions of image regions to describe texture characteristics, e.g., the heterogeneity and directionality. They are effective especially for natural textures, such as ceramic tiles and woods. Gray-level co-occurrence matrix-based [8] and local descriptor-based [9] methods are typical statistical texture analysis means. Structural methods focus on finding texture primitives of texture images and they are especially suitable for textures with obvious structural attributes. Characteristics of texture primitives, such as the density, directionality, or scale size, directly affect the texture pattern and structural visual effect. This approach is commonly applied to textures with repetitive patterns such as fabrics [10] and bricks [11]. By using texture-primitive features, Sharma *et al.* [12] proposed a semisupervised approach for segmentation and classification of medical images and achieved superior performance. Filter-based methods aim to describe textures in a transformed domain using spatial transformations, filters, or filter banks. They are the most widely used approaches for texture analysis, description and inspection. In general, these methods can be divided into spatial domain [13], frequency domain [14], and spatial-frequency domain [15]. Fourier analysis [14], wavelet transform [16], and Gabor transform [17] are commonly used skills in filter-based texture analysis. However, these methods are usually used together with other approaches to specific issues [16], [18], [19]. Schneider and Merhof [18] presented an automatic method for plain and twill fabric detection by combining Fourier analysis, template matching, and fuzzy clustering. Finally, model-based algorithms describe texture patterns by modeling special distributions or other attributes with certain models [20], [21]. Xie and Mirmehdi [20] presented an approach to detect and localize defects in color textures with a Gaussian mixed model and it achieved better performance

Manuscript received July 13, 2017; revised October 16, 2017; accepted December 01, 2017. Date of publication February 5, 2018; date of current version May 10, 2018. This work was supported in part by the National Science Foundation of China under Grant 51327801 and Grant 51475193 and in part by the Major Project Foundation of Hubei Province under Grant 2016AAA009. The Associate Editor coordinating the review process was Dr. Weiwen Liu. *(Corresponding author: Hua Yang.)*

The authors are with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: huayang@hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2018.2795178

0018-9456 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

when compared with traditional Gabor filter bank-based method. Gao *et al.* [21] developed a spatial-frequency feature extraction algorithm for defect detection using smooth Itakura–Saito nonnegative matrix factorization. This algorithm exhibited superior performance. Similar approaches are not limited to specific models and tend to have higher flexibility.

Although good effects may be achieved with these methods on the description of texture features and the detection of texture defects, most of them are application dependent and are mainly suitable for homogeneous textures. Recently, with the development of deep learning technology, methods that use deep neural networks are gradually rising in the industrial defect inspection field. Ren *et al.* [22] proposed a supervised convolutional neural network (CNN) architecture for image patch classification with transfer learning. Then pixelwise prediction is conducted with the trained classifier over input images. This model exhibits good performance for wood defect inspection. Weimer *et al.* [23] and Park *et al.* [24] also proposed novel deep CNN architectures, which obviously improve the accuracy of automated defect detection. Li *et al.* [25] proposed a Fisher criterion-based stacked denoising autoencoder (AE) model for deformable patterned fabric defect detection. Discriminative representations can be achieved by training with defective and defect-less samples. Though effective, all of these methods share a common issue that labeled training data are required for model training even with the fine-tuning strategy [22]. This requirement may be harsh and even impossible in many industrial occasions where some defect types are unpredictable and occur only during production. Furthermore, due to the limited resources in a factory environment, collecting a wide range of defective samples is usually difficult. Therefore, the scalability of these methods may be limited in industrial practice.

To deal with these challenges, in this paper, we propose a novel approach, which we call multiscale convolutional denoising AE (MSCDAE), to detect and localize defects with only defect-free samples for model training. This approach employs an inverted pyramid architecture that textural image patches at different resolution scales can be reconstructed with a convolutional denoising AE (CDAE) network in each pyramid layer. Reconstruction residuals of patches are served as the criteria for direct pixelwise prediction. Prediction result of each pyramid layer is then synthesized to form the final inspection representation for a candidate sample.

The contributions of this paper include the following.

1) An unsupervised novel defect inspection method which can be trained with only defect-free samples is proposed. This method is capable of dealing with defects, which have not occurred before in production without any human intervention.
2) This method is implemented in a multimodal form. (Pyramid layers with various resolutions contain different textural information and can be viewed as multiple modalities.) This strategy is able to enhance robustness and accuracy of the method.
3) Attributes such as the convergence, noise immunity, and inspection accuracy of the proposed method are analyzed in detail in this paper. Extensive comparative tests are also conducted to evaluate this method quantitatively and qualitatively, and the inspection performance is demonstrated to be more robust and accurate.

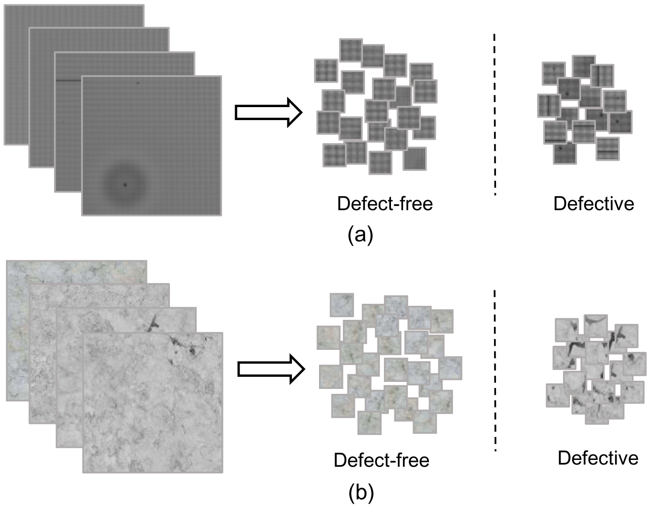

Fig. 1. Distributions of defect-free and defective patches in the patch domain for (a) homogeneous and (b) nonregular textural samples. Defect-free patches in sample (b) are different but visually consistent.

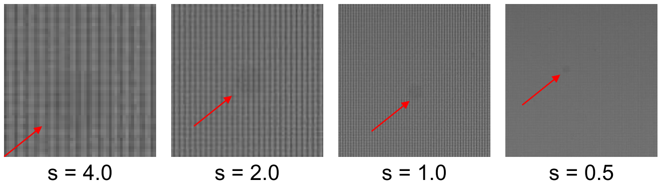

Fig. 2. Region mura defect in a LCD panel sample at different scales.

The rest of this paper is organized as follows. In Section II, we illustrate the motivation why we use the CDAE network for defect inspection and review the foundations of CDAE network briefly. Then, in Section III, we describe the procedures of the proposed MSCDAE model in detail, steps to train the model and test candidate images are summarized. In order to analyze the model and compare the overall performance, experimental results are exhibited in Section IV. Then the implementation details of the experiments are summerized in Section V. The applications and limitations of the proposed model are illustrated in Section VI. Finally, we conclude this paper in Section VII.

II. Motivation and Foundations

As stated above, the CDAE network and Gaussian pyramid structure are applied to the proposed MSCDAE model. The former is used to reconstruct the input image patches and generate reconstruction residual map for prediction. Respectively, the latter is utilized to analyze and synthesize the inspection result at different spatial resolutions. In this section, we will illustrate the motivation of using these two strategies first. Then the development of convolutional AE (CAE) and CDAE networks are briefly reviewed.

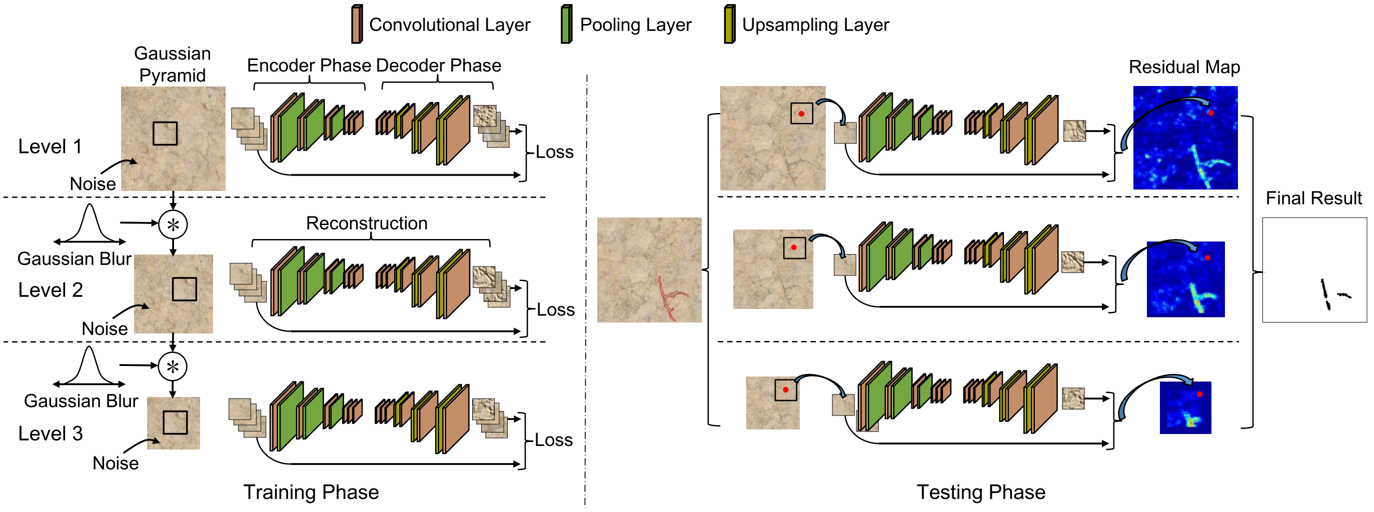

Fig. 3. Overall architecture of the proposed MSCDAE model in the training and testing phases. The CDAE module in each layer is optimized separately, and the reconstruction residual maps are resized to the size of the original image and integrated to form the final result.

A. Motivation Illustration

1) Why CDAE: The CAE network is commonly used for unsupervised learning of efficient codings [26]. The aim of this module is to learn a representation for a set of data and discover their common ingredients. For homogeneous textures in industrial, e.g., the LCD panel in Fig. 1(a), image patches, which do not contain defective areas, are highly similar in general. Therefore, it can be quite suitable to model the distribution of these defect-free image patches in the patch domain by training a CAE network to reconstruct the inputs. After training, filters in the CAE model will be sensitive to similar patches and show high responses to them. While for patches, which contain defective areas, the appearance and distribution in the patch domain are usually quite different. Therefore, the learned model may be less sensitive to them, and relatively low responses will be generated. By measuring residuals between the responses and the raw inputs, the direct pixelwise prediction can be easily conducted. For nonregular textures in industry, e.g., the ceramic tiles in Fig. 1(b), the situation is slightly different that there are some differences between the defect-free patches themselves. This phenomenon may affect the convergence of the model training to a certain extent. However, thanks to the fact that although the patterns of these ceramic tile samples appear different, the visual impression of the same product line remains consistent. Therefore, the CAE model is still capable of modeling them, and this conclusion has been verified in latter experiments. The CDAE model tends to be more robust than CAE by introducing noise for model training especially for defect inspection. This statement will also be verified.

2) Why Pyramid: Different information may exhibit for an image at different scales. As shown in Fig. 2, a region mura defect is presented. This defect becomes more and more conspicuous when the resolution scale changes from 4 to 1 and nearly disappears when the resolution scale changes to 0.5. The repetitive horizontal and vertical stripes also appear in different forms with the changes of spatial resolution. This phenomenon is in accordance with that in [20] and [27]. And it shows that processing in multiscale with pyramid may ensure the capture of sufficient textural properties, which are often data dependent. With multiscale texture analysis, characterizing a pixel based on local neighborhood pixels in various resolutions tends to be more robust. Therefore, the pixelwise prediction result can be more accurate.

B. Foundations and Developments

An AE model is based on an encoder-decoder paradigm, which is usually trained in an unsupervised fashion and allows hierarchical feature extraction from unlabeled samples. CAEs differ from conventional AEs as they retain the structure information of 2-D images and their weights are shared among local locations. The architecture of a typical CAE contains an encoder part with convolutional and pooling layers, and an analogous decoder part with deconvolutional and upsampling layers. The encoder and decoder parts can be defined as transitions ϕ and ψ, such that

$$\begin{aligned}\phi &: \mathcal{X} \rightarrow \mathcal{F}\\ \psi &: \mathcal{F} \rightarrow \mathcal{X}\\ \phi, \psi &= \underset{\phi,\psi}{\operatorname{argmin}} \|\mathbf{x} - \psi\ (\phi\ (\mathbf{x}))\|^2\end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$ refers to an image patch in the $\mathcal{X}$ domain, $\mathbf{z} = \phi(\mathbf{x}) \in \mathbb{R}^p = \mathcal{F}$ refers to the corresponding hidden layer map in the $\mathcal{F}$ domain. Assume $\mathbf{x}'$ denotes the reconstruction, the encoder and decoder processes can be expanded as

$$\begin{aligned}\mathbf{z} &= \sigma\ (\mathbf{W} \circ \mathbf{x} + \boldsymbol{b})\\ \mathbf{x}' &= \sigma'\ (\mathbf{W}' \circ \mathbf{z} + \boldsymbol{b}')\end{aligned} \tag{2}$$

where "$\circ$" is the convolution process; $\mathbf{W}$ and $\mathbf{W}'$ are the weight matrices; $\boldsymbol{b}$ and $\boldsymbol{b}'$ are the bias vectors for the encoder and decoder, respectively; and σ and σ' are the nonlinear mapping processes, specifically, the convolutional, pooling, deconvolutional, and upsampling processes. Particularly, the pooling and upsampling processes are usually conducted in the form of max pooling and max unpooling [28]. The CAE model can be trained to minimize the reconstruction errors (such as mean squared errors)

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \frac{1}{2N} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathbf{x}'_i \right\|^2 + \lambda \cdot \sum_{w \in \{\mathbf{W}, \mathbf{W}'\}} \|w\|_F \qquad (3)$$

where N is the number of samples, λ is a constant that balances the relative contributions of the reconstruction and the regularization terms, and $(\|\mathbf{x}_i - \mathbf{x}'_i\|^2)^{1/2}$ is the reconstruction residual of the ith image patch.

The CDAE network is slightly different from CAE that it takes partially corrupted inputs for model training and aims to recover the original undistorted inputs. This is done by first corrupting the initial input $\mathbf{x}$ into $\tilde{\mathbf{x}}$ by means of a stochastic mapping $\tilde{\mathbf{x}} \sim q(\tilde{\mathbf{x}}|x)$ [29]. In latter experiments, the salt and pepper noise is utilized for the corruption process. Assume $\tilde{\mathbf{x}}'$ is the reconstruction of the corrupted data $\tilde{\mathbf{x}}$, loss of the CDAE model is measured as $\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}')$. The concrete form of $\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}')$ is similar to that of $\mathcal{L}(\mathbf{x}, \mathbf{x}')$ in (3). Stochastic gradient descent algorithm [30] can be easily applied for optimization of CAE and CDAE models.

III. Proposed Methods

In this section, procedures of the proposed MSCDAE model are discussed in detail. As shown in Fig. 3, the overall architecture of this model in the training and testing phases is exhibited. Procedures in the training phase mainly aim to learn the CDAE network at each pyramid level and calculate the optimal threshold for defect segmentation, yet those in the testing phase illustrate the processes to inspect a candidate defective image. Specific illustrations are as follows.

A. MSCDAE Model Training

Training phase of the MSCDAE model mainly includes the image preprocessing, patch extraction, model training, and threshold determination procedures.

1) Image Preprocessing: All image preparations prior to the training procedure are included in this part. They are the illumination normalization, Gaussian pyramid downsampling, and noise corruption steps.

a) Illumination normalization: Most of the existing methods are highly sensitive to illumination variations during the defect inspection process. In order to reduce the rate of false detection, an illumination normalization process based on Weber's law [31] is conducted first of all. According to this law, stimuli are perceived not in absolute terms, but in relative terms: their fold changes in magnitude relative to the background level of stimulus. Assume $\boldsymbol{I}$ is an image to be inspected, the illumination normalization process can be implemented as

$$\boldsymbol{I}' = \mathrm{WLD}(\boldsymbol{I}) \qquad (4)$$

where $\mathrm{WLD}(\cdot)$ refers to a Weber local descriptor [32], which can be expressed as

$$\mathrm{WLD}(\boldsymbol{I}) = \arctan\left(\sum_{\triangle x \in A} \sum_{\triangle y \in A} \frac{\boldsymbol{I}(x, y) - \boldsymbol{I}(x - \triangle x, y - \triangle y)}{\boldsymbol{I}(x, y)} \right) \qquad (5)$$

where $A = \{-1, 0, 1\}$, $\{(x - \triangle x, y - \triangle y) | \triangle x \in A, \triangle y \in A\}$ refers to the eight-neighborhood region of point (x, y). As shown in Fig. 4, the performance of this step with two fabric textural samples is exhibited. It can be seen that the uneven illumination conditions can be well alleviated with this illumination normalization process. (Please note that this process is redundant on occasion where the changes in brightness are supposed to be defective.)

b) Gaussian pyramid downsampling: In a Gaussian pyramid subsequent images are weighted using a Gaussian blur and scaled down by resampling, as shown in Fig. 3. Each pixel in a pyramid level contains a local average that corresponds to a pixel neighborhood in a lower level of the pyramid. That is,

$$\boldsymbol{I}^{(n+1)} = S^{\downarrow} G(\boldsymbol{I}^{(n)}), n = 1, 2, \cdots n_l; \quad \boldsymbol{I}^{(1)} = \boldsymbol{I}' \qquad (6)$$

where $S^{\downarrow}$ refers to the downsampling process, $G(\cdot)$ denotes the Gaussian convolution [33], n_l is the number of pyramid levels, and $\boldsymbol{I}'$ is the image after illumination normalization.

c) Noise corruption: As stated, the salt and pepper noise [34] is utilized for data corruption in the MSCDAE model. Let $\tilde{\boldsymbol{I}}^{(n)}$ denote the corrupted image at level n, $\tilde{g}_{i,j}$ and $g_{i,j}$ refer to the gray levels of pixels at position (i, j) in the corrupted and original clean images. The corrupted data is given by

$$\tilde{g}_{i,j} = \begin{cases} g_{i,j}, & \text{with probability } (1 - p) \\ s, & \text{with probability} p \end{cases} \qquad (7)$$

where

$$s = \begin{cases} 0, & \text{with probability } 0.5 \\ 255, & \text{with probability } 0.5. \end{cases} \qquad (8)$$

The probability p directly affects the degree of data contamination, and its impact to the model will be verified in latter experiments.

2) Patch Extraction: After image preprocessing, patches will be collected to train the CDAE network at each pyramid layer. For pixelwise prediction, characterizing a pixel based on local neighborhood information can be more robust than using only a single pixel. Nevertheless, the size of local neighborhoods is usually data dependent. Thanks to the multiscale pyramid structure, fixed size patch extraction can be conducted in multiscale levels instead of generating patches with various sizes, which are computationally expensive [20]. Suppose the patch size is $w \times h$ and the stride interval is s, dimension of the patch set generated from an image with size $W \times H$ can be expressed as $[N_p, N_c, w, h]$, where $N_p = \lceil [(W - w)/s + 1] \rceil \times \lceil [(H - h)/s + 1] \rceil$ refers to the number of patch samples, $N_c \in \{1, 3\}$ refers to the number of image channels. It should be noted that only defect-free images are utilized for model training and the patch set at each pyramid layer should not be confused with each other.

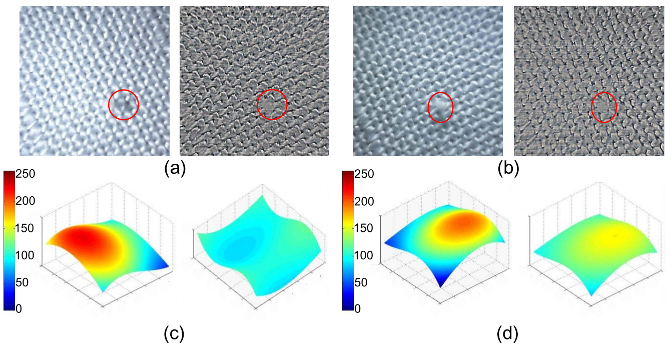

Fig. 4. Illustration of two fabric texture samples with (a) central spot and (b) side illuminations. Changes in (c) illumination before and (d) after the illumination normalization procedure. It can be seen that the interference of uneven illuminations can be weakened with the illumination normalization procedure (best viewed in color).

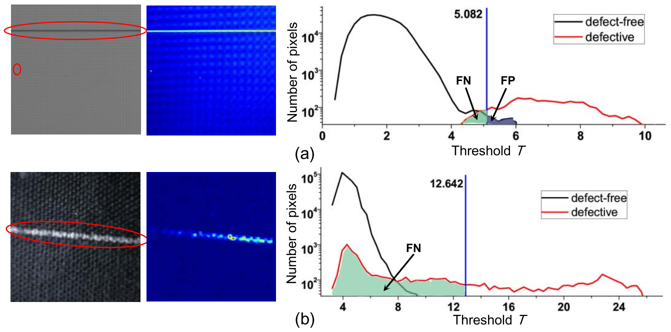

Fig. 5. Illustration of the influence of threshold for defect segmentation. First column: original textural samples. Second column: reconstruction residual maps in the first pyramid layer. Third column: histogram of pixels in the residual maps, the vertical "blue" lines refer to the segmentation thresholds (best viewed in color). (a) Defective LCD sample and (b) Defective textile sample.

3) Model Training: As stated in Section II-A, the CDAE network tends to have good reconstruction ability for industrial defect-free textural image patches after model training. Therefore, it is natural to use the reconstruction residual as the criterion for pixelwise prediction. Training of the proposed MSCDAE approach aims to model the distribution of defect-free image patches by minimizing the total reconstruction loss of all patches in each image pyramid layer. In this paper, the optimization process is conducted by applying batch gradient descent algorithm in an error back propagation fashion. We illustrate the procedures of one epoch iteration with m batches as follows:

Algorithm 1

```
Step 1: $\forall\, l$, set $\triangle \boldsymbol{W}^{(l)} = 0$, $\triangle \boldsymbol{b}^{(l)} = 0$;
Step 2: for $i = 1$ to $m$,
    a. Calculate the partial derivatives $\triangledown_{\boldsymbol{W}^{(l)}} \mathcal{L}(\mathbf{x}, \mathbf{x}')$ and
       $\triangledown_{\boldsymbol{b}^{(l)}} \mathcal{L}\left(\mathbf{x}, \mathbf{x}'\right)$;
    b. Partial differential superposition:
       $\triangle \boldsymbol{W}^{(l)} = \triangle \boldsymbol{W}^{(l)} + \triangledown_{\boldsymbol{W}^{(l)}} \mathcal{L}\left(\mathbf{x}, \mathbf{x}'\right)$,
       $\triangle \boldsymbol{b}^{(l)} = \triangle \boldsymbol{b}^{(l)} + \triangledown_{\boldsymbol{b}^{(l)}} \mathcal{L}\left(\mathbf{x}, \mathbf{x}'\right)$;
Step 3: Update weight parameters:
    a. Renew $\boldsymbol{W}^{(l)} = \boldsymbol{W}^{(l)} - \alpha \cdot \triangle \boldsymbol{W}^{(l)}$, $\boldsymbol{b}^{(l)} = \boldsymbol{b}^{(l)} -$
       $\alpha \cdot \triangle \boldsymbol{b}^{(l)}$;
    b. Disrupt the order of patches in the data set and
       finish the current epoch iteration.
```

where l denotes the lth layer of the deep networks, and α is the learning rate, which will be discussed in latter experiments. Procedures to optimize the parameters $\boldsymbol{W}'$ and $\boldsymbol{b}'$ are similar and will not specified here repetitively.

4) Threshold Determination: Threshold is a critical parameter to distinguish the defective and defect-free points. Assum $\xi^{(i)} = \{\vartheta^{(i)}(\mathbf{x}_k) | k = 1, 2, \cdots, N_p\}$ refers to the reconstruction residual set of patches in the ith image pyramid layer, N_p is the number of total training patches, $\mathbf{x}_k$ is the kth image patch, $\vartheta^{(i)}(\mathbf{x}_k) = (\|\mathbf{x}_k - \mathbf{x}'_k\|^2)^{1/2}$ is the reconstruction residual between $\mathbf{x}_k$ and $\mathbf{x}'_k$. It can be seen from Fig. 5 that $\vartheta^{(i)}$ of the majority of defect-free pixels are mainly concentrated near the mean value and their distribution is close to the Gauss distribution. Therefore, the segmentation threshold can be defined as $T^{(i)} = \mu^{(i)} + \gamma \cdot \sigma^{(i)}$ [20], where $\mu^{(i)}, \sigma^{(i)}$ are the mean and standard deviation of set $\xi^{(i)}$. The parameter γ can be adjusted according to the segmentation sensitivity. In latter experiments, we set $\gamma = 2$ according to experimental verification, and the effect of this parameter is also discussed. Note that, the thresholds are determined by using the training patches which are all defect free. As shown in Fig. 5, two samples that are segmented by the thresholds (the vertical line with “blue” color) in the first pyramid layer are exhibited. The “green” regions, which are labeled "FN," mean the false negative points (points which are defective but judged to be defect free), and the “blue” regions, which are labeled “FP,” mean the false positive points (points which are defect free but judged to be defective). Threshold determination is to find the optimal threshold value, which achieves the tradeoff between reducing the FN and FP indicators simultaneously. Detailed procedures to select the optimal threshold will be discussed in latter experiments.

After model training, novelty detection and defect localization processes can be easily conducted for a candidate textural image. Detailed procedures are presented as follows.

B. MSCDAE Model Testing

Testing phase of the MSCDAE model is exhibited in Fig. 3. It mainly includes the image preprocessing, patch extraction, residual map construction, defect segmentation, and synthesization procedures.

1) Image Preprocessing: Steps of image preprocessing in the testing phase are slightly different from those in the training phase that the noise corruption step is not necessary for inspecting a candidate image [35]. As for the illumination normalization and Gaussian pyramid downsampling steps, the implementations are the same.

2) Patch Extraction: After preprocessing, patches will be extracted for textural images to be inspected. In the testing phase, this process should be conducted row by row or column by column strictly so as to generate the residual map conveniently [36] (in the training phase, patches can be extracted randomly). The extraction strategy and parameters (w, h, and s) applied are the same as that in the training phase.

3) Residual Map Construction: As is shown in Fig. 3, for prediction of a pixel, a local receptive field $\mathbf{x}$ with size $w \times h$ in its neighborhood will be extracted and flows forward in the trained model. The reconstruction residual $\vartheta(\mathbf{x})$ serves as the criterion and represents the extent to which the point belongs a defective area. Therefore, in each image pyramid layer, a residual map can be constructed for subsequent processing.

4) Defect Segmentation: This procedure is conducted upon the residual map in each image pyramid layer as follows:

$$\varsigma^{(i)}(\mathbf{x}_{j,k}) = \begin{cases} 0 & \text{if } \vartheta^{(i)}(\mathbf{x}_{j,k}) \le T^{(i)} \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

where $\vartheta^{(i)}(\mathbf{x}_{j,k})$ refers to the reconstruction residual at position (j,k) in the ith layer, $\varsigma^{(i)}(\mathbf{x}_{j,k})$ is the corresponding label after segmentation. Note that the defective points are labeled "1."

5) Result Synthesization: This procedure is conducted mainly to improve the robustness and accuracy of the defect inspection task. Here, we follow the strategy in [1] and [37], which combine information from multiple pyramid levels. It assumes that a defect must appear in at least two adjacent levels. Therefore, a logical AND operation can be implemented in every pair of adjacent levels to reduce false alarms. The operation is expressed as

$$\boldsymbol{\varsigma}^{(i,i+1)}(\mathbf{x}) = \boldsymbol{\varsigma}^{(i)}(\mathbf{x})\boldsymbol{\varsigma}^{(i+1)}(\mathbf{x}). \tag{10}$$

Note that each residual map is scaled up to be the same size as the original input images, "&" refers to the AND operation between pixels at the same position in the two adjacent maps. Next, the resulting maps are associated with a logical OR operation to generate the final result. That is,

$$\boldsymbol{\varsigma}(\mathbf{x}) = \boldsymbol{\varsigma}^{(1,2)}(\mathbf{x}) \mid \boldsymbol{\varsigma}^{(2,3)}(\mathbf{x}) \cdots \mid \boldsymbol{\varsigma}^{(n-1,n)}(\mathbf{x}) \tag{11}$$

where "|" refers to the OR operation between pixels at the same position, and $\boldsymbol{\varsigma}(\mathbf{x})$ is considered the final consolidated map. Note that a morphology open operation can be carried out to remove some noise interferences if necessary.

IV. Experiments and Discussion

To evaluate performance of the proposed MSCDAE model, in this section, several sets of experiments are presented. Specifically, convergence performance of the proposed MSCDAE model is evaluated first. This is very important to verify the efficiency of the MSCDAE model in practical use. Second, robustness of the model against noise in the CDAE networks is discussed. The amount of noise injection is especially important to make the trained model more stable and avoid overfitting [35]. Third, inspection performance of the model against the segmentation threshold is verified. Threshold is a very sensitive parameter which directly affects the inspection effects. Fourth, inspection performance of each pyramid level in the MSCDAE model is illustrated. The comprehensive result which serves as the ultimate inspection output is compared with results when using only single resolution level. Finally, overall inspection performance of the MSCDAE model is compared with several superior methods qualitatively and quantitatively. Corresponding analyses and discussions are made in detail. Before the introductions of these experiments, the data sets and parameters used in this paper are illustrated as follows.

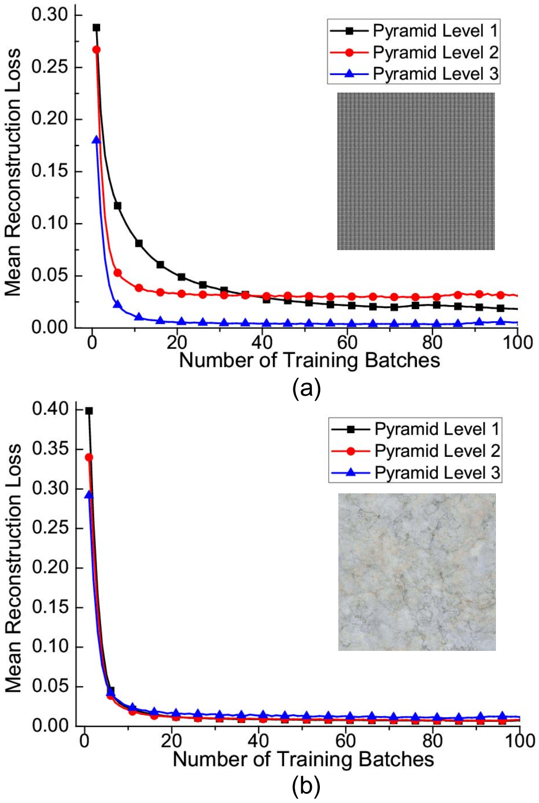

Fig. 6. Convergence tests of the proposed MSCDAE model on (a) homogeneous and (b) nonregular textures. The CDAE model in each pyramid layer is trained with only four defect-free samples (best viewed in color).

A. Data Sets and Parameters

In our experiments, textural samples which are homogeneous and nonregular are both utilized. In these data, the used fabric samples mainly come from the Fabrics [38] and the KTH-TIPS2 [39] data sets. The former consists of about 2000 samples of garments and fabrics, and some of them are captured under different illumination conditions. Respectively, the latter contains 4752 images of 11 materials, each of them is captured under varying poses, illuminations, and scales. The used LCD samples come from [36]. There are various categories of defects in this data set with different sizes and spatial scales. Other samples come from [40]. This data set is artificially generated, but similar to real world problems and has been widely used for defect inspection verification. Note that all the used images are in PNG format, and each color channel is 8 b.

Parameter tuning can directly affect the effect of an algorithm. In the proposed MSCDAE model, parameters, which have significant influences, mainly include the patch size $w \times h$ in the patch extraction step, the noise proportion p in the noise corruption step, the segmentation control parameter γ in the

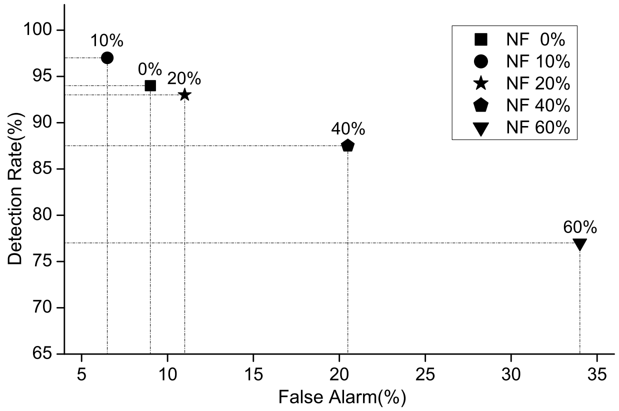

Fig. 7. Inspection performance (detection rate and false alarm) against different proportions of noise for model training.

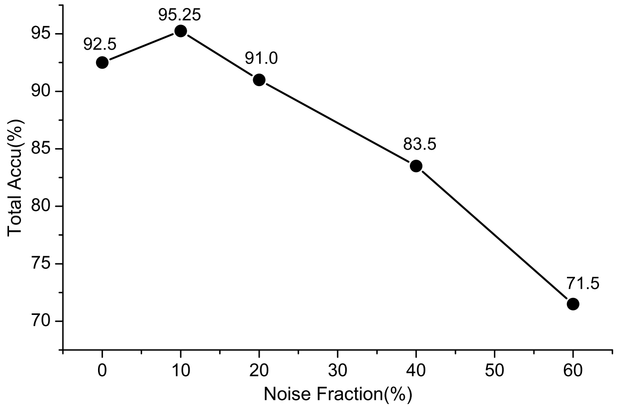

Fig. 8. Overall inspection accuracy against different proportions of noise for model training.

threshold determination step, the tradeoff constant λ in loss function in (3), and the learning rate α in the model training step. The latter two are called "hyper parameters," in which the λ can be determined by cross validation [41], $\lambda = 0.001$ is used in latter experiments after verification. As for the learning rate α, a heuristic principle is applied by choosing a value close to the largest learning rate that does not cause divergence [42]. In latter experiments, we start with $\alpha = 0.1$, and if the loss increases, this parameter will decrease automatically until no divergence is observed. As for the other three parameters, corresponding experiments will be conducted to verify their impacts, and motivations to determine their values are also discussed.

B. Convergence Evaluation

As stated before, obtaining a large number of defect samples and labeling their ground-truth regions are not feasible in many industrial applications. Therefore, whether the MSCDAE model can be applied in practice with only unlabeled samples for model training needs to be validated.

As shown in Fig. 6, relationships between the iteration number and mean reconstruction loss of MSCDAE are presented. In the two tests, the CDAE networks in each pyramid layer of MSCDAE are trained on the LCD panel samples (homogeneous texture) and ceramic tile samples (nonregular texture) separately. In each application, four defect-free textural images of 512 × 512 pixels are utilized and the patch size is 8 × 8. (Please note that the changes in patch size do not significantly affect the inspection performance because of the pyramid architecture [20]; therefore, this parameter is fixed throughout our experiments.) It can be seen from Fig. 6 that the proposed MSCDAE method is capable of going toward stability in very few iterations no matter it is trained on homogeneous textures or nonregular textures. In addition, it should be noted that the training procedure of this model is usually done offline, so it has no impact on the online defect inspection efficiency.

C. Robustness Against the Noise Fraction

As stated in Section III-A1, the salt and pepper noise is added into input data during training in terms of improving robustness of the trained MSCDAE model. (Please note that the random noise injection process is conducted only in the training phase. It should not be used in the testing phase so as to ensure fairness of performance comparison [35].) Here, we attempt to evaluate how the proportion of noise, which is called noise fraction (NF), will affects the inspection performance. In this test, 200 defective and 200 defect-free LCD panel samples of 512 × 512 pixels are utilized, and three indicators related to accuracy of the inspection are applied to verify the performance via varying NF. The indicators are

$$\begin{aligned}&\text{Detection Rate (DR): TP / (TP + FN)}\\&\text{False Alarm (FA): FP / (TN + FP)}\\&\text{Total Accu (TA): (TP + TN) / (TP + FN + TN + FP)}\end{aligned} \tag{12}$$

where TN refers to the defect-free samples which are identified defect free, FP is those which are identified defective; TP refers to the defective samples which are identified defective, FN is those which are identified defect free, respectively. With these definitions, it can be seen that DR is the ratio of correctly identified defective samples, FA is the ratio of defect-free samples, which are falsely identified defective, and TA is the ratio of all correctly identified samples. Generally, DR and FA need to be used together, and the goal of choosing the optimal NF is to increase DR, meanwhile depressing FA. Fig. 7 shows the inspection DR and FA with different NF values. It can be seen that when the NF indicator varies in range {0%, 10%, 20%, 40%, 60%}, the DR rises first, then falls rapidly, while the FA shows an opposite trend. When NF = 10%, the inspection performance is the best. Furthermore, we exhibit the overall inspection accuracy in Fig. 8. It is clear to us that the inspection results show a parabolic trend, rising first and then falling rapidly. When NF varies from 10% to 60%, TA drops from 95.25% to 71.5%, with a drop rate of 33.2%. This is mainly due to the fact that with heavily corrupted data, the CDAE model fails to learn meaningful feature representations, and thus unable to distinguish defective and defect-free patches. In latter experiments, NF = 10% is utilized in the proposed MSCDAE model.

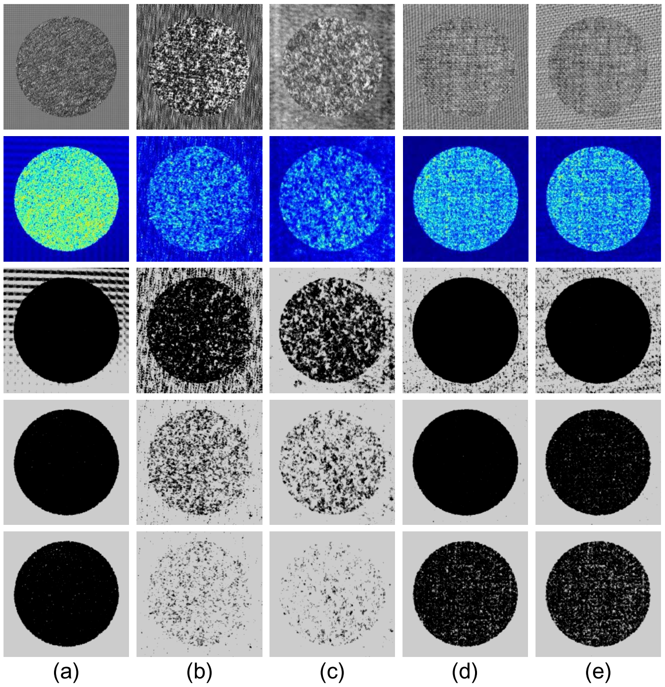

Fig. 9. Inspection performance of the MSCDAE model against the segmentation threshold. First row: original synthetic collage samples. Second row: reconstruction residual maps. Third, fourth, and fifth rows: segmentation results when $\gamma = 1, 2$, and 3 (best viewed in color).

D. Robustness Against the Segmentation Threshold

As stated, threshold is a very critical parameter, which directly affects the defect inspection performance. In the proposed MSCDAE model, segmentation threshold in the ith pyramid layer is defined as $T^{(i)} = \mu^{(i)} + \gamma \cdot \sigma^{(i)}$, where γ is the parameter, which can be adjusted according to the segmentation sensitivity. In this part, we attempt to evaluate how the parameter γ will affect the inspection performance. As shown in the first row of Fig. 9, some synthetic collages generated from textural samples are exhibited. In each case, the background is the learned texture for model training (four 512 × 512 samples are used for training), and the foreground (circular region) is treated as the novelty to be detected. Note that areas of the background and the foreground are the same, and textures selected for these two parts are particularly similar so as to increase the difficulty for segmentation. Effect of the parameter γ is verified by separating foregrounds from these collages. As shown in the second row of Fig. 9, the reconstruction residual maps in the first pyramid layer are exhibited. Corresponding segmentation results when $\gamma = 1, 2, 3$ are also presented in the third, fourth, and fifth rows. It can be seen that there are many falsely detected regions when $\gamma = 1$, and with γ increases, the defect-free areas, which are falsely identified defective will gradually decrease, and this is the same as the correctly identified defective areas.

In order to further quantify the effect of the parameter γ, as shown in Fig. 10, we plot the ROC curves by varying γ value with respect to these collage samples. The true positive rate (TPR) and false positive rate (FPR) are slightly different from the definitions in Section IV-C. The former refers to the proportion of the correctly segmented defective area in the foreground, while the latter refers to that of the falsely segmented defective area in the background. In Fig. 10, the "⊗," "★," and "☒" in each curve correspond to the result when γ is set as 3, 2, 1. It can be seen from the trends that with γ increases, both the TPR and FPR will decrease. Furthermore, according to the attribute of the ROC curve, the closer the curve is to the top-left corner, the more accurate the segmentation result will be [the top-left corner (0, 1) represents that both the background and foreground are correctly segmented absolutely]. That is to say, the goal of choosing the optimal threshold value is reducing the false negative rate (FNR) and FPR indicators simultaneously. Therefore, we can take the false detection rate (FDR) as an indicator to discover the optimal threshold value, FDR = FNR + FPR = (1 − TPR) + FPR. As can be seen in Table I, the indicator FDR have the minimum average value when $\gamma = 2$. That is to say, when $\gamma = 2$, the average defect inspection performance is the best. For verifications of the threshold parameters in other pyramid layers, the methods are similar. In latter experiments, $\gamma = 2$ will be utilized for defect segmentation.

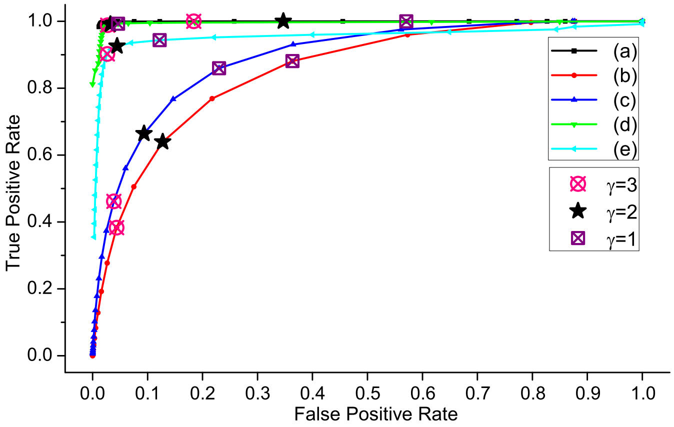

Fig. 10. ROC curves with different threshold values for the quantitative analysis on textural samples in Fig. 9 (best viewed in color).

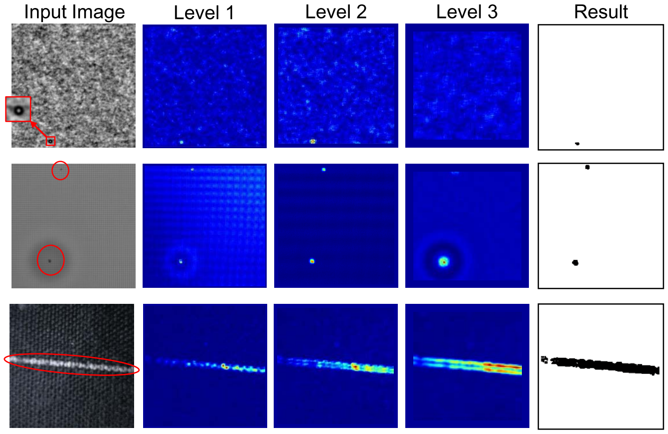

Fig. 11. Localizing texture defects with the proposed MSCDAE model on marble samples (first row), LCD panel samples (second row), and fabric samples (third row). Left to right are original defect images, residual maps from the first, second, third pyramid layers, and the final comprehensive results (best viewed in color).

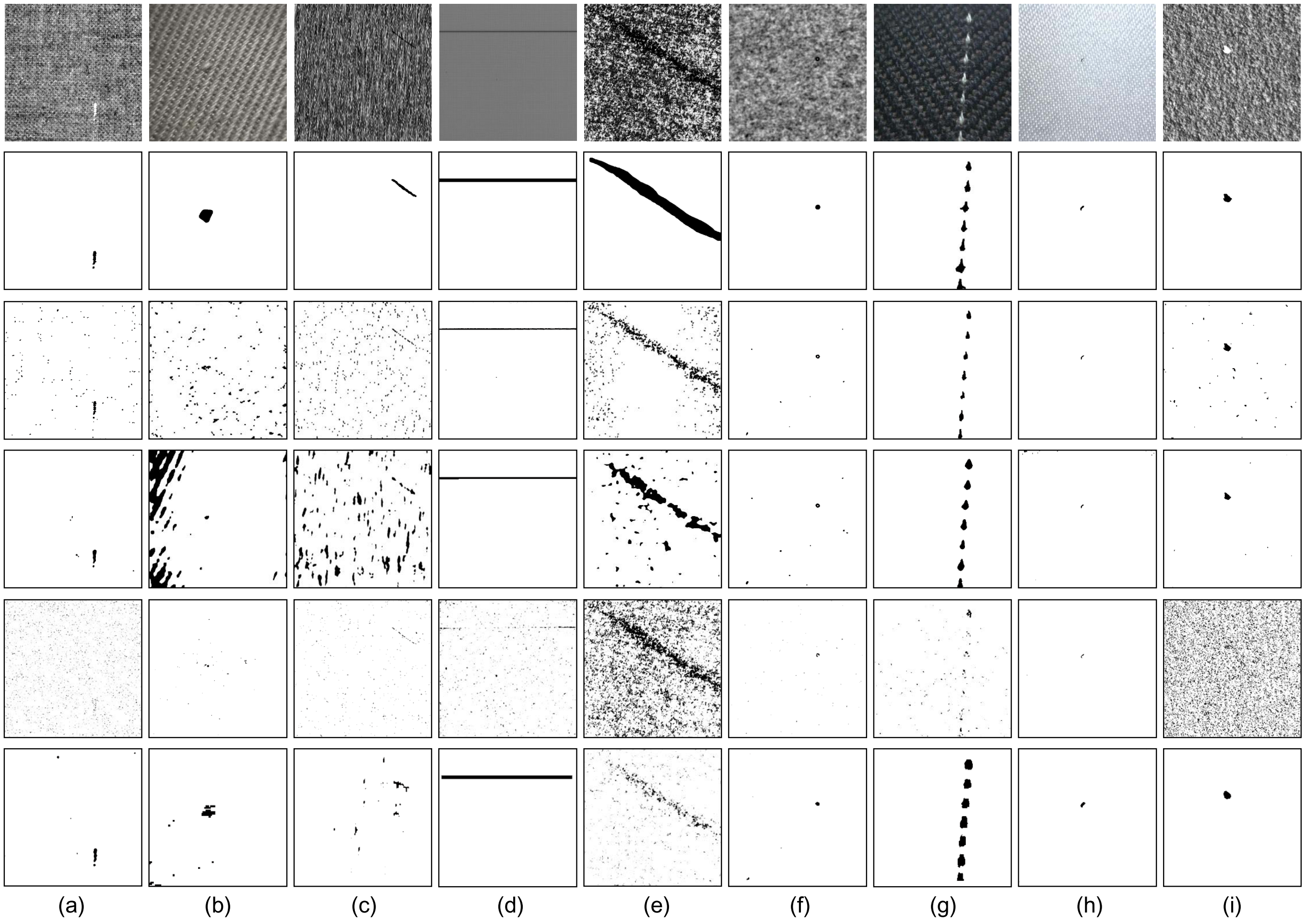

Fig. 12. Segmentation results of various textural samples with different methods. Top to bottom are the original defect samples, ground-truth regions, results obtained by the DCT, LCA, PHOT methods, and the proposed MSCDAE model.

TABLE I
AVERAGE FDR WITH DIFFERENT THRESHOLD PARAMETERS γ

Threshold value / Average FDR	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$
FDR = (1 - TPR) + FPR	0.332	**0.282**	0.321

E. Defect Inspection Evaluation

As stated before, the Gaussian pyramid architecture is utilized in the proposed MSCDAE model. And the final inspection result of this model is the synthesis of result from each pyramid layer. However, whether this fusion mechanism is beneficial to the overall inspection performance still needs to be validated. As shown in Fig. 11, the inspection results of several textural samples (column 1) with the proposed MSCDAE method are exhibited. The reconstruction residual heat maps at pyramid level l, $l = 1, 2, 3$, are also visualized (columns 2–4). It can be seen that the inspection results are very sensitive to pyramid resolutions.

As detailed textural information may be concealed or manifested in different resolutions, the CDAE networks in different levels of MSCDAE will present different activation responses even to the same object. It should be noted that in order to generate the final inspection results (column 5), residual map in each scale is interpolated to the same size as the original image. By contrastive analysis of the final results and results from independent pyramid levels, conclusion can be drawn that the used fusion mechanism is capable of enhancing robustness and eliminating false alarms, e.g., in level 3 of sample 1, no defects are detected with the model, but the final result is accurate; in level 1 of sample 2, many defective regions are falsely detected, but the final result is also satisfactory. Therefore, the proposed MSCDAE model tends to be more robust and accurate for defect inspection than models with a single resolution scale.

F. Overall Performance Comparison

In order to verify the effect of the proposed MSCDAE model more fully, in this section, we compared the inspection performance of this model with several superior unsupervised algorithms, e.g., the discrete cosine transform (DCT) [43], low-pass filtering with curvature analysis (LCA) [44], and phase only transform (PHOT) [45] methods. Defective regions inspected with these methods upon various textural samples are exhibited in Fig. 12. As can be seen that the DCT

TABLE II

QUANTITATIVE SEGMENTATION RESULTS OF VARIOUS TEXTURAL SAMPLES WITH DIFFERENT METHODS

Criteria / Samples	Recall				Precision				F1-Measure			
	DCT	LCA	PHOT	Ours	DCT	LCA	PHOT	Ours	DCT	LCA	PHOT	Ours
(a)	0.7179	0.4511	0.1411	**0.7215**	0.3997	**1.0**	0.0179	0.6193	0.5153	0.6218	0.0318	**0.6665**
(b)	0.1841	0.0921	0.0419	**0.4111**	0.0536	0.0071	0.3459	**0.5467**	0.0830	0.0132	0.0748	**0.4693**
(c)	0.2730	**0.4775**	0.1328	0.2034	0.0408	0.0238	0.1127	**0.1426**	0.0710	0.0453	0.1219	**0.1677**
(d)	0.3188	0.5493	0.0742	**0.8765**	0.9676	**1.0**	0.1631	0.9231	0.4796	0.7091	0.1020	**0.8992**
(e)	0.4412	0.5610	**0.6099**	0.3578	0.4701	0.6385	0.2989	**0.6418**	0.4552	**0.5972**	0.4012	0.4595
(f)	0.5556	0.6120	0.3180	**0.6322**	0.5070	0.4123	0.3673	**0.6962**	0.5302	0.4927	0.3409	**0.6627**
(g)	0.5294	0.6216	0.1720	**0.7381**	**1.0**	0.8290	0.5350	0.6866	0.6923	0.7105	0.2603	**0.7114**
(h)	0.5493	0.5070	0.4366	**0.9718**	**1.0**	0.2769	0.9394	0.5656	0.7091	0.3582	0.5962	**0.7150**
(i)	0.8157	0.6413	0.4140	**0.8808**	0.3910	0.8985	0.0060	**0.9204**	0.5287	0.7484	0.0119	**0.9002**

method (the third row) shows good performance mainly in obvious periodic textures, e.g., samples in Fig. 12(d) and (g). It performs poorly in nonregular textures, e.g., samples in Fig. 12(b), (c), and (e), and it is easy to introduce falsely detected regions. This phenomenon is primarily due to the fact that the spectrums of defective and defect-free regions in nonregular textures are easily confused in DCT. For the LCA method (the fourth row), the periodic textures are removed before inspection by retaining the low-frequency parts and eliminating the high-frequency parts. Similar methods are likely to be affected by illumination [sample in Fig. 12(b)] and perform general to defects with relatively large sizes [sample in Fig. 12(e)]. As for the PHOT method (the fifth row), defects, such as samples in Fig. 12(d) and (g), are easy to be omitted because of the phase only transform strategy. In contrast to these phenomena, the proposed MSCDAE method provides a novel way to distinguish defective and defect-free regions by directly learning from several defect-free samples. It exhibits relatively good performance on both homogenous and nonregular textures.

Furthermore, three quantitative indicators are utilized to analyze the inspection results quantitatively. The indicators utilized are defined as

$$\text{Recall} = \frac{\text{TP}}{\text{GT}} \tag{13}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{F1-Measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

where TP and FP denote the number of true positive and false positive pixels, GT denotes the total number of ground-truth pixels for the given images. The F1-Measure indicator is a comprehensive evaluator upon both recall and precision. The quantitative segmentation results of textural samples in Fig. 12 with the DCT, LCA, PHOT, and our method are exhibited in Table II. The bold data indicates better results. It can be seen that the proposed MSCDAE model has better compatibility with both the Recall and Precision indicators, and this conclusion can also be drawn from the F1-Measure results in the fourth column. From these data, we can find that the proposed MSCDAE model has better comprehensive inspection performance to all these textural samples, no matter they are homogeneous or nonregular. However, the other three methods may fail in some cases, e.g., the DCT algorithm shows poor precision performance to the samples in Fig. 12(b) and (c), it is mainly because the textures are nonregular and the defects are too small that many interferences are introduced in the segmentation results; the LCA algorithm is invalid to the samples in Fig. 12(b), (c), and (h), the reason is also related to the spatial scales of the defects in these samples that spectra of small defects are easily omitted in the frequency filtering operation. As for the PHOT algorithm, it fails in the samples in Fig. 12(a), (b), (d), and (i). The phenomenon is mainly related to the nature of the algorithm itself. This experiment further confirms the superiority and stability of the proposed MSCDAE method.

V. IMPLEMENTATION DETAILS

The proposed MSCDAE model is implemented on a desktop computer with eight cores, 32 GB memory, and GTX 980Ti Nvidia GPU. The training process of this model is time-consuming. With four 512×512 training samples, the training time can reach 742 s. But fortunately, this process is usually conducted offline and will not affect the online inspection. Computational complexity of the MSCDAE model in the online inspection phase mainly concentrates on the steps to gain and to reconstruct the patch set. This two steps in MSCDAE are the most time-consuming and nested-loop procedures, which are directly affected by the sizes of input images and network architectures of the model [46]. But the practical computational workload is more inclined to the hardware accelerators [46]. According to our experiments, for a candidate image with 512×512 pixels, the average inspection time with the proposed model (patch size: 8×8) is 674 ms and is able to meet the real-time inspection demands.

The proposed model is implemented using Python. The packages involved in the implementation mainly include Keras [47], Theano [48], Scikit-Learn [49], Numpy [50], and Scipy [51].

VI. APPLICATIONS AND LIMITATIONS

The proposed MSCDAE approach is a novel unsupervised-learning-based defect inspection method and it exhibits good

TABLE III
ABBREVIATIONS UTILIZED IN THIS PAPER

AE	Auto-Encoder
AOI	Automated Optical Inspection
CAE	Convolutional Auto-Encoder
CDAE	Convolutional Denoising Auto-Encoder
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
DR	Detaction Rate
FA	False Alarm
FDR	False Detection Rate
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GPU	Graphics Processing Unit
GT	Total Number of the Ground-Truth Pixels
LCA	Low-Pass Filtering with Curvature Analysis
LCD	Liquid Crystal Display
MSCDAE	Multi-Scale Convolutional Denoising Auto-Encoder
NF	Noise Fraction
PHOT	Phase Only Transform
TA	Total Accuracy
TN	True Negative
TP	True Positive
TPR	True Positive Rate

performance on homogeneous and nonregular textured surfaces. In this paper, experimental results have demonstrated the effectiveness and superiority of this algorithm in LCD panels, ceramic tiles, and textiles. However, they are not limited. In the future, we will verify this algorithm in more application scenarios, such as the defect inspection of wood, thin film, solar wafers, and so on.

In addition, although defect inspection efficiency of the MSCDAE model meets the needs of actual production, it can still be accelerated with more efficient data-exchanging mechanism between GPU and CPU. In the future, strategies will be adopted to further improve this approach.

VII. Conclusion

We proposed a novel method MSCDAE for unsupervised-learning-based defect inspection. This method constructs a Gaussian pyramid-based CDAE architecture to distinguish defective and defect-free regions. The obtained reconstruction residual map in each pyramid level is synthesized to form the final inspection evaluation. Analysis and experiments confirmed that by making full use of the unsupervised learning and multimodal result-fusion strategies, the defect inspection performance can be more accurate and robust. In the future, we will verify this model in more applications and further improve it.

Appendix

Abbreviations utilized in this paper are listed in Table III as follows.

References

[1] W. Kim and C. Kim, "Total variation flow-based multiscale framework for unsupervised surface defect segmentation," *Opt. Eng.*, vol. 51, no. 12, pp. 1272011–1272019, 2012.

[2] O. Silvén, M. Niskanen, and H. Kauppinen, "Wood inspection with non-supervised clustering," *Mach. Vis. Appl.*, vol. 13, no. 5, pp. 275–285, 2003.

[3] Z. Hocenski, S. Vasilic, and V. Hocenski, "Improved canny edge detector in ceramic tiles defect detection," in *Proc. 32nd Annu. Conf. IEEE Ind. Electron. (IECON)*, Nov. 2006, pp. 3328–3331.

[4] A. Armillotta, "Assessment of surface quality on textured FDM prototypes," *Rapid Prototyping J.*, vol. 12, no. 1, pp. 35–41, 2006.

[5] G.-H. Hu, "Automated defect detection in textured surfaces using optimal elliptical Gabor filters," *Optik-Int. J. Light Electron Opt.*, vol. 126, no. 14, pp. 1331–1340, 2015.

[6] D.-M. Tsai and C.-C. Kuo, "Defect detection in inhomogeneously textured sputtered surfaces using 3D Fourier image reconstruction," *Mach. Vis. Appl.*, vol. 18, no. 6, pp. 383–400, 2007.

[7] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 7, no. 3, pp. 1–22, 2008.

[8] T. Vujasinovic *et al.*, "Gray-level co-occurrence matrix texture analysis of breast tumor images in prognosis of distant metastasis risk," *Microscopy Microanal.*, vol. 21, no. 3, pp. 646–654, 2015.

[9] T. Mäenpää and M. Pietikäinen, "Texture analysis with local binary patterns," *Handbook Pattern Recognit. Comput. Vis.*, vol. 3, pp. 197–216, May 2005.

[10] B. Mallik-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Textile Res. J.*, vol. 70, no. 9, pp. 758–762, 2000.

[11] L. Lefebvre and P. Poulin, "Analysis and synthesis of structural textures," in *Proc. Graph. Interface*, 2000, pp. 77–86.

[12] N. Sharma, A. K. Ray, S. Sharma, K. Shukla, S. Pradhan, and L. M. Aggarwal, "Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network," *J. Med. Phys./Assoc. Med. Phys. India*, vol. 33, no. 3, pp. 119–126, 2008.

[13] F. Malik and B. Baharudin, "The statistical quantized histogram texture features analysis for image retrieval based on median and Laplacian filters in the DCT domain," *Int. Arab J. Inf. Technol.*, vol. 10, no. 6, pp. 616–624, 2012.

[14] H. Ji, X. Yang, H. Ling, and Y. Xu, "Wavelet domain multifractal analysis for static and dynamic texture classification," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 286–299, Jan. 2013.

[15] J. Rakun, D. Stajnko, and D. Zazula, "Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry," *Comput. Electron. Agricult.*, vol. 76, no. 1, pp. 80–88, 2011.

[16] J. Scharcanski, "Stochastic texture analysis for measuring sheet formation variability in the industry," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1778–1785, Oct. 2006.

[17] A. G. Zuñiga, J. B. Florindo, and O. M. Bruno, "Gabor wavelets combined with volumetric fractal dimension applied to texture analysis," *Pattern Recognit. Lett.*, vol. 36, pp. 135–143, Jan. 2014.

[18] D. Schneider and D. Merhof, "Blind weave detection for woven fabrics," *Pattern Anal. Appl.*, vol. 18, no. 3, pp. 725–737, 2015.

[19] T. M. Basile, L. Caponetti, G. Castellano, and G. Sforza, "A texture-based image processing approach for the description of human oocyte cytoplasm," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 10, pp. 2591–2601, Oct. 2010.

[20] X. Xie and M. Mirmehdi, "TEXEMS: Texture exemplars for defect detection on random textured surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1454–1464, Aug. 2007.

[21] B. Gao, H. Zhang, W. L. Woo, G. Y. Tian, L. Bai, and A. Yin, "Smooth nonnegative matrix factorization for defect detection using microwave nondestructive testing and evaluation," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 4, pp. 923–934, Apr. 2014.

[22] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 99, no. 2, pp. 1–12, 2017.

[23] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Ann.-Manuf. Technol.*, vol. 65, no. 1, pp. 417–420, 2016.

[24] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine learning-based imaging system for surface defect inspection," *Int. J. Precision Eng. Manuf.-Green Technol.*, vol. 3, no. 3, pp. 303–310, 2016.

[25] Y. Li, W. Zhao, and J. Pan, "Deformable patterned fabric defect detection with fisher criterion-based deep learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1256–1264, Apr. 2017.

[26] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, Sep. 2014.

[27] Y. Xu, X. Yang, H. Ling, and H. Ji, "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 161–168.

[28] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2018–2025.

[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[30] S. Klein, J. P. W. Pluim, M. Staring, and M. A. Viergever, "Adaptive stochastic gradient descent optimisation for image registration," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 227–239, 2009.

[31] B. Wang, W. Li, W. Yang, and Q. Liao, "Illumination normalization based on Weber's law with application to face recognition," *IEEE Trans. Signal Process. Lett.*, vol. 18, no. 8, pp. 462–465, Aug. 2011.

[32] J. Chen *et al.*, "WLD: A robust local image descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1705–1720, Sep. 2010.

[33] S. Saitoh, "Approximate real inversion formulas of the Gaussian convolution," *Appl. Anal.*, vol. 83, no. 7, pp. 727–733, 2004.

[34] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479–1485, Oct. 2005.

[35] S. Yin *et al.*, "Noisy training for deep neural networks in speech recognition," *EURASIP J. Audio Speech Music Process.*, vol. 2015, no. 1, pp. 1–14, 2015. [Online]. Available: https://citations.springer.com/item?doi=10.1186/s13636-014-0047-0

[36] S. Mei, H. Yang, and Z. Yin, "Unsupervised-learning-based feature-level fusion method for mura defect recognition," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 1, pp. 105–113, Feb. 2017.

[37] J. Escofet, R. F. Navarro, M. S. M. Garcia-Verela, and J. M. Pladellorens, "Detection of local defects in textile Webs using Gabor filters," *Opt. Eng.*, vol. 37, no. 8, pp. 2297–2307, 1998.

[38] C. Kampouris, S. Zafeiriou, A. Ghosh, and S. Malassiotis, "Fine-grained material classification using micro-geometry and reflectance," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 778–792.

[39] B. C. Mario Fritz, E. Hayman, and J.-O. Eklundh. *The KTH-TIPS and KTH-TIPS2 Image Databases*. Accessed: Jun. 25, 2017. [Online]. Available: http://www.nada.kth.se/cvap/databases/kth-tips/documentation.html

[40] M. Wieler and T. Hahn. *Weakly Supervised Learning for Industrial Optical Inspection*. Accessed: Jun. 25, 2017. [Online]. Available: https://hci.iwr.uni-heidelberg.de/node/3616

[41] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of Database Systems*. Boston, MA, USA: Springer, 2009, pp. 532–538. [Online]. Available: https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_565#howtocite

[42] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade* (Lecture Notes in Computer Science), vol. 7700. Berlin, Germany: Springer, 2012, pp. 437–478.

[43] H.-D. Lin, "Tiny surface defect inspection of electronic passive components using discrete cosine transform decomposition and cumulative sum techniques," *Image Vis. Comput.*, vol. 26, no. 5, pp. 603–621, 2008.

[44] D.-M. Tsai and T.-Y. Huang, "Automated surface inspection for statistical textures," *Image Vis. Comput.*, vol. 21, no. 4, pp. 307–323, 2003.

[45] D. Aiger and H. Talbot, "The phase only transform for unsupervised surface defect detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 295–302.

[46] J. Cong and B. Xiao, "Minimizing computation in convolutional neural networks," in *Proc. Int. Conf. Artif. Neural Netw.*, vol. 8681. 2014, pp. 281–290. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-11179-7_36#citeas

[47] F. Chollet. *Keras*. Accessed: Jan. 10, 2017. [Online]. Available: https://github.com/fchollet/keras

[48] R. Al-Rfou *et al.* (2016). "Theano: A python framework for fast computation of mathematical expressions." [Online]. Available: https://arxiv.org/abs/1605.02688

[49] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[50] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011.

[51] E. Jones, T. Oliphant, and P. Peterson. (2014). *SciPy: Open Source Scientific Tools for Python*. Accessed: Sep. 1, 2016. https://github.com/fchollet/keras

Shuang Mei received the B.S. degree in mechanical electronic and information engineering from the China University of Geoscience, Wuhan, China, in 2012. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan.

His current research interests include image processing, pattern recognition, and deep learning.

Hua Yang (M'12) received the B.S. and M.S. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2008, respectively, and the Ph.D. degree from Hiroshima University, Higashihiroshima, Japan, in 2011.

He worked in Hiroshima University as a Research Associate in 2011 and then became an Assistant Professor in 2012. He is currently an Associate Professor with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology. His current research interests include high-speed vision and its applications-object recognition, detection and tracking, particle image velocity, and dynamic-based vision inspection.

Zhouping Yin (M'04) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1994, 1996, and 2000, respectively.

He was a Cheung Kong Chair Professor with the Huazhong University of Science and Technology in 2009. Since 2005, he has been the Vice Head of the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, where he is also a Professor with the School of Mechanical Science and Engineering. He has been a Principal Investigator for several projects sponsored by the General Program and Major Program of the National Science Foundation of China, the National Basic Research Project of China, and others. He has authored or co-authored two monographs, three chapters in English books, and over 30 papers in international journals, such as IEEE TRANSACTIONS, ASME Transactions, and Computer-Aided Design. He is leading a research group and conducting research in electronic manufacturing equipment and technology, including flexible electronics and electronics packaging.

Dr. Yin was a recipient of the China National Funds for Distinguished Young Scientists in 2006.