Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 3, 515–523
DOI: 10.34768/amcs-2021-0035

A NESTED AUTOENCODER APPROACH TO AUTOMATED DEFECT INSPECTION ON TEXTURED SURFACES

MUHAMMED ALI NUR OZ [a,*], OZGUR TURAY KAYMAKCI [b], MUHARREM MERCIMEK [a]

[a]Department of Control and Automation Engineering
Yildiz Technical University
Cifte Havuzlar, Davutpasa Campus, 34220 Esenler, Istanbul, Turkey
e-mail: {maoz,mercimek}@yildiz.edu.tr

[b]Department of Electronics Engineering
Canakkale Onsekiz Mart University
Barbaros Mahallesi Terzioglu Campus, Prof. Dr Sevim Bulut Street, No. 20, Canakkale, Turkey
e-mail: okaymakci@comu.edu.tr

In recent years, there has been a highly competitive pressure on industrial production. To keep ahead of the competition, emerging technologies must be developed and incorporated. Automated visual inspection systems, which improve the overall mass production quantity and quality in lines, are crucial. The modifications of the inspection system involve excessive time and money costs. Therefore, these systems should be flexible in terms of fulfilling the changing requirements of high capacity production support. A coherent defect detection model as a primary application to be used in a real-time intelligent visual surface inspection system is proposed in this paper. The method utilizes a new approach consisting of nested autoencoders trained with defect-free and defect injected samples to detect defects. Making use of two nested autoencoders, the proposed approach shows great performance in eliminating defects. The first autoencoder is used essentially for feature extraction and reconstructing the image from these features. The second one is employed to identify and fix defects in the feature code. Defects are detected by thresholding the difference between decoded feature code outputs of the first and the second autoencoder. The proposed model has a 96% detection rate and a relatively good segmentation performance while being able to inspect fabrics driven at high speeds.

Keywords: autoencoders, defect detection, automatic visual inspection, deep learning.

1. Introduction

The globalization of industry is gaining momentum and its scope of influence is increasing. Renovating the machine and information technology infrastructure for a company can be a solution as long as the risk to reward ratio is remarkable. Companies rarely endorse such changes. On the other hand, expectations of target customers regarding product quality and quantity must be met to keep a competitive edge in this race. Thus, having a leading-edge product inspection maintains its importance. Automated visual inspection systems, one of the key technologies in industrial manufacturing or production, lifts the total production quantity and quality compared with those obtained with traditional inspection methods.

Visual inspection can be described as examining industrial products by acquiring image data of different modalities. Certain standards, specifications or requirements should be considered in doing so. Color, area/volume defects, texture printing offsets, thread/knit defects are some of the flaws or defects often checked during the inspection. Exemplifying the principles of visual inspection systems (VISs) is cumbersome since these systems differ widely in the type of products and defects elaborated. Recently, intelligent automated visual systems have been flourishing to be used in the product lines, and they share a general workflow to pin down the defects. This is the category of VISs examined in this paper.

Essentially the appointed goal for a VIS is to determine whether or not a product is defective and

*Corresponding author

to report defects if there are any. Da Costa *et al.* (2020) utilized deep residual neural network (ResNet50) classifiers to sort defective products. Wei *et al.* (2019) proposed the use of compressive sensing for the augmentation of the training samples, which enables training convolutional neural networks (CNNs) with a satisfying performance from a small number of samples. Zhang *et al.* (2019) proposed an on-line detection system that can capture multiple images from different angles to detect defects.

The objective of a VIS can be appointed solely as the classification of a product as either defective or not, and this can be worthwhile in some applications. On the other hand, developing an inspection technology now demands the extraction of detailed reports by getting involved with the depiction and localization of the defects. The number, location, size, and type of defects determine the quality of the product. Generally, if the quality of a product cannot be brought to a certain level, then it will be priced accordingly. Furthermore, information about defects can also be used to detect and prevent malfunctioning processes in the production line. Hence, a VIS should be able to extract comprehensive data from defects.

To segment out defects, Li *et al.* (2019b) introduced a new texture-feature description operator, the multi-directional binary pattern (MDBP). The operator was reported to be detecting defects using a threshold obtained from the similarity between the feature matrices extracted from non-defective and defective fabric images. Also, Li *et al.* (2019a) proposed separating images into defective and background regions using low-rank decomposition and efficient second-order orientation-aware descriptors. Yu *et al.* (2018) presented a coarse-to-fine model to find rail surface defects, and Wang *et al.* (2018) proposed a guidance template-based method. A fully convolutional neural network was proposed to detect and identify defects on different types of surfaces: wooden surfaces (He *et al.*, 2019; 2020), tires (Wang *et al.*, 2019b) and fabrics (Ouyang *et al.*, 2019). Although these methods perform well, they essentially suffer from the use of a limited number of samples as the input.

In the production line, products in focus can be replaced by structurally different new ones. Hence, the VIS should be applicable, rapidly and conveniently adjustable to these new products. Lizarraga-Morales *et al.* (2019) proposed rough-set-based rules involving the extraction of binary features from both defective and defect-free images and in this wise detection of defects on textiles with certain patterns. For defect detection, Li *et al.* (2016) presented Fisher criterion based stacked denoising autoencoders trained with limited defect-free samples. Luo *et al.* (2019) proposed the selectively dominant local binary patterns framework that could be applied to a variety of manufacturing industries. Wang *et al.* (2019a) presented an entity sparsity pursuit to identify rare defects that violate the low-rank structure of the image. Alipour and Harris (2020) proposed improving the robustness of material-specific deep learning models for crack detection to be used for various materials. The most significant shortcoming of these methods is that a large data set is always required to train or adjust these models.

The systems in industrial production are designed to minimize defects, thus obtaining sufficiently defective images of the products for training becomes a time-consuming task. Recently, researchers have proposed systems that do not need defective data to develop all-purpose intelligent VISs. Lian *et al.* (2019) proposed an adversarial network to generate numerous exaggerated samples on the defects to improve the performance of the classifiers. Sun *et al.* (2019) used transfer learning with adaptive multiscale image collection. To detect defects without negative samples, Jia *et al.* (2020) proposed splitting the image into lattices and measuring image similarity with respect to the template statistics learned from defect-free samples. Similarly, Kang and Zhang (2019) presented the Elo-rating algorithm and utilized it on the integral image to detect defects in fabrics. Similar efforts for the same objective have been undertaken in several other research areas involving unsupervised deep learning methods such as the stacked de-noising convolutional auto-encoder (Xie *et al.*, 2019) or the deep convolutional generative adversarial network (Hu *et al.*, 2019).

The methods mentioned above deserve appreciation, yet a significant amount of human intervention to keep the system running remains a necessity. Yang *et al.* (2019) addressed this problem and proposed a fast and accurate multiscale feature-clustering-based convolutional autoencoder method using only defect-free samples; the detection of defects with high accuracy with very little human intervention was achieved. Similarly, Bergmann *et al.* (2019) advised the use of unsupervised-learning-based convolutional denoising autoencoder networks consisting of different Gaussian pyramid levels. The capabilities of these methods are the state of the art, yet training times, the size of the presented networks and the data sets that were used to test performance were kept fairly simple.

The main thrust of this paper is establishing a defect detection model to be used in a real-time intelligent visual surface inspection system. The system can detect defects that are scattered on repetitive textures patterns with high accuracy, and the method requires very low training time as well as low memory. The proposed approach does not require defective samples for the training phase, and it can be used in a variety of different materials.

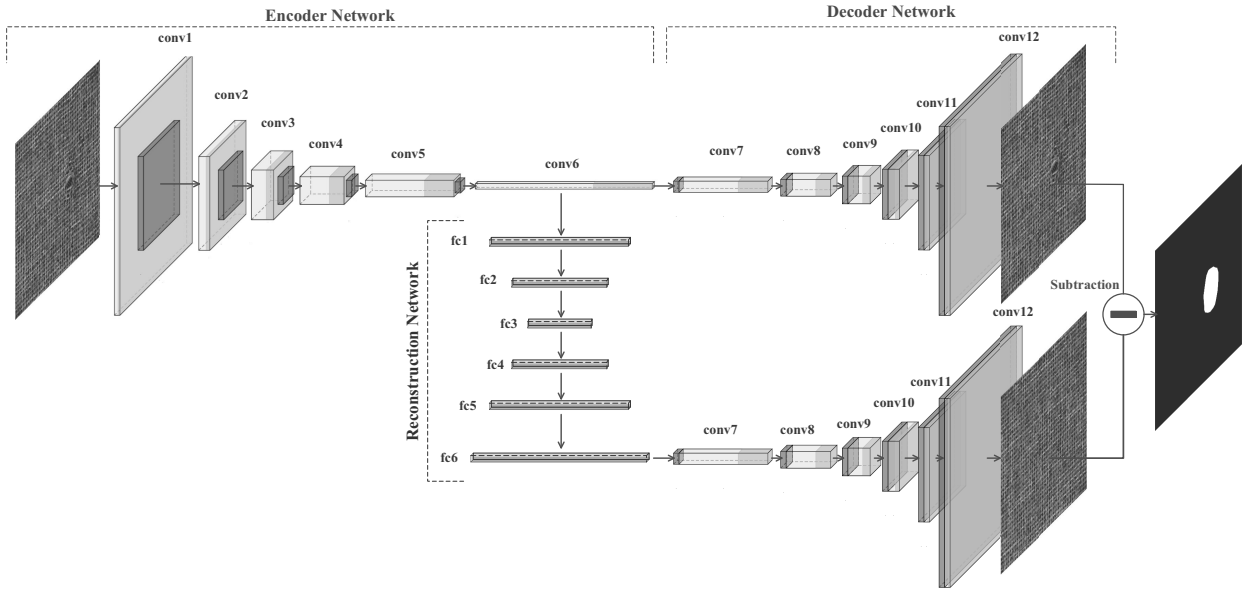

Fig. 1. Overall architecture of the proposed nested AE model.

2. Proposed approach

2.1. Network architecture of the nested autoencoder. Traditionally, neural networks are trained to detect and segment defective areas of images using labeled images of both defective and normal samples. However, creating a balanced training set that includes numerous defect types is a challenging task as defects are usually very rare. Taking advantage of unsupervised learning approaches, which do not require labeled data, neural networks can be trained to segment defects without defective samples.

Autoencoders (AEs) are the most widely used architectures for unsupervised learning tasks. They can reconstruct an image without the manipulation of a user or the need for label. An AE forces the data through a bottleneck and then reconstructs, the original data from the bottleneck representation which is also referred to as code. Standard fully connected AEs perform poorly on images and the size of the neural network for acceptable performance has to be quite large. This results in a poor and tedious training phase.

Convolutional autoencoders (CAEs), on the other hand, can capture local spatial features by a series of filters called kernels and, with a large number of layers, extract high-level features from an image. An encoder, which houses the generation of the bottleneck effect and produces a feature representation as the result, can be designed using the convolution and pooling layer. The decoder, which reconstructs the image from the feature representation, can be built using the upsampling and convolution layers.

It is assumed that, by training convolutional autoencoders on defect-free samples, the CAE gains the ability to extract only features related to defect-free samples. Therefore, when a defect is present on the tested image, the CAE will not be able to regenerated the same defect at its output. Taking advantage of this property, defects can be detected and segmented using the residuals of the input and output image. However, a major problem with this assumption is that AEs tend to "peek" at the input image, thus generating an image that is more similar to the defective input one than its defect-free equivalent (Bhattad *et al*., 2018). This peeking behavior, which is a result of features that AEs learn during training, might also be able to represent defective areas of input images.

A simple way to overcome this problem is the use of a narrower bottleneck. This provides better reconstruction performance, yet presents fewer reconstruction details. Another method is to train CAEs to exploit randomly applied masks similar to denoising AEs. However, this method is computationally expensive in application.

In this paper, we propose a nested AE architecture for real-time defect detection which is capable of turning defective images into their defect-free counterparts and segmenting out the texture defects seen in inspected images. It also provides a low run time and training time as well as low complexity. The proposed architecture suggests that two nested AEs can overcome the cheating habits of AEs. The first one is composed of CNNs and extracts useful features for the reconstruction of the image, and it will be referred to as the feature extraction AE.

The encoding module of the feature extraction AE, E, first transforms the defective input image $\tilde{X}$ to

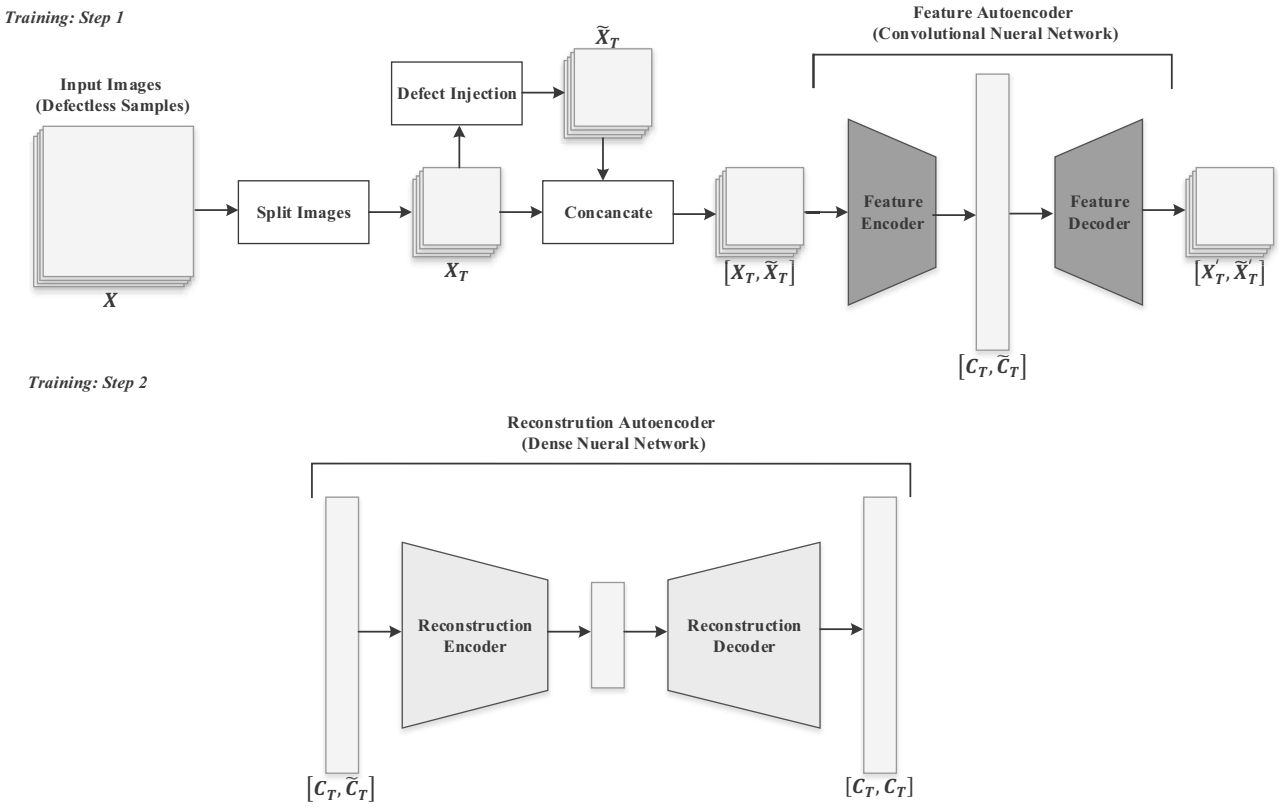

Fig. 2. Training steps of the proposed nested AE defect detection scheme.

the feature space, where the image is encoded into a one-dimensional feature vector C. The encoder module can be defined as a nonlinear mapping function

$$\begin{aligned} E : X &\rightarrow C, \\ C &= E(X), \\ C &= \varphi_E(W_E \circ X + b_E), \end{aligned} \tag{1}$$

where W_E, b_E and φ_E represent kernels, bias vectors and the convolution operation, respectively.

Instead of reconstructing a defect-free image, the second AE, which can be referred to as the prediction AE, P, predicts defect-free counterparts of the extracted feature vector. Predicting a defect-free image feature vector $C^{'}$ from the defective feature vector $\tilde{C}^{'}$ is much more effective and robust since the AE cannot distort the response using the input image or the feature vector. The reconstruction module can be defined as a nonlinear mapping function

$$\begin{aligned} C^{'} &= R(\tilde{C}^{'}), \\ C^{'} &= \varphi_R(W_R \tilde{C}^{'} + b_R), \end{aligned} \tag{2}$$

where W_R, b_R and φ_R represent kernels, bias vectors and the convolution operation, respectively.

Finally, the decoder module of the feature extraction AE generates two images from predicted defect-free $C^{'}$ and input image feature vectors $\tilde{C}^{'}$. As a result, a predicted flawless image X^c and the reconstructed image $X^{'}$ are obtained. The decoder module can be defined as a nonlinear mapping function

$$\begin{aligned} X^{'} &= D(C^{'}), \\ X^{'} &= \varphi_D(W_D \circ C^{'} + b_D), \\ \tilde{X}^{'} &= D(\tilde{C}^{'}), \\ \tilde{X}^{'} &= \varphi_D(W_D \circ \tilde{C}^{'} + b_D), \end{aligned} \tag{3}$$

where W_D, b_D and φ_D represent kernels, bias vectors and convolution operation, respectively.

The difference between the predicted defect-free image $X^{'}$ and the reconstructed input image $\tilde{X}^{'}$ is computed and then the absolute value of the difference is used to obtain an error map.

$$EM = |X^{'} - \tilde{X}^{'}|. \tag{4}$$

These two images are used instead of the original image $\tilde{X}$ in order to remove the unmodeled detail from the error map.

A segmented image S is obtained after flaws are detected and segmented out using a fixed global

threshold T

$$S(x,y) = \begin{cases} 0 & \text{if } EM(x,y) \geq T, \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

The overall architecture of the proposed model is given in Fig. 1.

2.2. Training nested AE for defect detection. The design of a real-time inspection system can be a long and arduous task. The execution time of the algorithms must be precise and very low to meet the production speed demands. A defect detection algorithm must be as simple as possible while ensuring high detection rates. Furthermore, the algorithm must be able to process multiple split images in parallel without paying an extra bit of attention to its context. In that, training input images X should be fragmented into the dimensions of the smallest repetitive pattern. In the proposed defect detection scheme, training and testing are carried out without any delays in processing the fragmented images. Our assumption is that the lighting is optimal and the image acquisition system is stable.

To train the nested AE with only a few good samples, fragmented images X_T are intentionally injected with defects. This step is similar to the corruption step of denoising the AE, but instead of random noise, the images are injected with artificial defects such as randomly generated lines, solid and translucent circles. These artificial defects have been chosen empirically to train the network, and they carry characteristics similar to the ones of the defects encountered in real-life systems.

The proposed architecture is trained in two phases, and training the feature extraction AE is the first phase. This AE projects the images to be inspected to a one-dimensional invertible feature space. To this end, both good fabric X_T and the defect injected samples $\tilde{X}_T$ are concatenated and used to train the feature extraction AE. Training is considered successful when the smallest texture fibers and minimum size defects are accurately modeled in the reconstructed images. The feature extraction AE is trained according to the mean square error

$$L_{\text{out}} = \frac{1}{N} \sum_{i=1}^{N} \left\| X_{T_i} - X_{T_i}{}' \right\|^2, \tag{6}$$

where N is the batch size.

In the second phase, the prediction AE is trained to predict the corresponding defect-free feature vector from the input. After the successful training of the feature extraction AE, good fabric samples and defect injected samples are passed through the encoder of the outer AE, resulting in defective $\tilde{C}_T$ and their corresponding good feature vectors C_T. Defective feature vectors and defect-free vectors are concatenated as $C_{\text{in}} = \tilde{C}_T + C_T$ to train the prediction AE, with the target vector being their defect-free counterparts as the outputs $C_{\text{out}} = C_T + C_T$. The prediction AE is also trained using the mean square error as the metric

$$L_{\text{pr}} = \frac{1}{N} \sum_{i=1}^{N} \left\| C_{\text{in}} - C'_{\text{out}} \right\|^2, \tag{7}$$

where C'_{out} is the predicted output.

The training scheme steps are given in Fig. 2.

2.3. Testing the nested AE. Once the images to be detected are split into patches of convenient sizes, for each patch the corresponding feature vector is extracted with the help of the encoder module of the feature extraction AE. A feature vector is then fed into the prediction AE to get its counterpart defect-free feature vector. These two vectors are then converted back to the image domain by the decoder module of the feature extraction AE. Finally, two images are obtained as

$$\begin{aligned} \tilde{X}'_T &= D(E(\tilde{X}_T)), \\ X^c_T &= D(P(E(\tilde{X}_T))), \end{aligned} \tag{8}$$

where $\tilde{X}_T$ is the image to be inspected, $\tilde{X}'_T$ is the reconstructed image and X_T is the defect-free image.

An error map is generated with a simple difference operation between these two images. The error map is passed through a simple average filter for better visualization of the regions with errors. The threshold level, used to detect and segment flaws from the EM, is chosen as the maximum value of the error maps calculated from good fabric samples.

2.4. Experimental results. In this section, the proposed nested AE structure is tested on a challenging data set and the performance is presented and discussed. Since the method is designed to be used in defect detection systems, defect detection performance will be evaluated considering the rates more than the segmentation success. Other performance criteria such as the speed, size, and complexity of the algorithm will be discussed.

We tested our method on the MVTec anomaly detection data set, which is a quite complex and challenging dataset for defect inspection when compared with the ones in the literature (MVTEC AD, 2019). The data set consists of 3629 images from 15 categories for training and verification, and 1725 images from 15 categories for testing. The proposed method has been tested on the carpet image set, which is indeed a textile product, and contains 279 samples for training and 117 samples for testing. The training set consists of only defect-free images. The test set consists of flawless images coming along with various types of

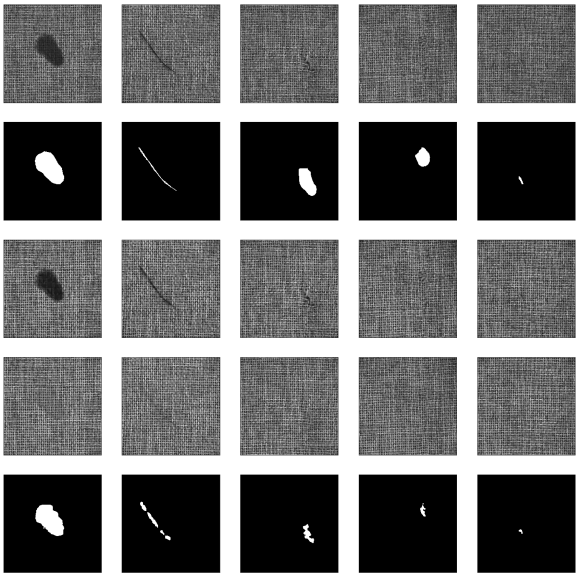

Fig. 3. Test results of the proposed nested AE model for five different defect types. Top to bottom are the original defective samples, ground-truth regions, reconstructed images and predicted defect-free images, respectively.

defects such as color, cut, hole, metal contamination, and thread. Synthetic defects similar to the abnormalities that may arise in real industrial inspection scenarios were superposed onto the test images. All images have a spatial resolution of 1024×1024 pixels. The images are reported to be acquired under optimal lighting conditions. For the derived defective image regions, there are accompanying pixel-level segmented patch images, and these can be treated as the ground truth for the defects.

For a comprehensive defect detection task, both the shape and size of the defect must be accurately determined. The performance of the method is evaluated with respect to these two attributes of the defects. Accuracy, sensitivity, specificity criteria will be used to discuss defect detection performance. Test results will be labeled as true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN). If a defective image is classified as defective, then the test result is TP; however, if the defective image is classified as defect-free then the test result is FN. Similarly, if a defect-free image is classified as defective, then the test result is FP and also if a defect-free image is classified as defect-free than the test result is TN. The expressions for accuracy, sensitivity, specificity terms can be stated as follows:

$$\begin{aligned} \text{accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}, \\ \text{sensitivity} &= \frac{\text{TP}}{\text{TP}+\text{FN}}, \\ \text{specificity} &= \frac{\text{TN}}{\text{TN}+\text{FP}}. \end{aligned} \tag{9}$$

Table 1. Accuracy (Acc.), sensitivity (Sens.) and specificity (Spec.) measures calculated for each defect type.

Defect types	No. of images	No. of faults	Acc.	Sens.	Spec.
Color	19	21	0.7619	0.7619	–
Cut	17	18	1.0000	1.0000	–
Good	28	28	1.0000	–	0.0000
Hole	17	17	0.8500	1.0000	1.0000
Metal Cont.	17	18	1.0000	1.0000	–
Thread	19	20	1.0000	1.0000	–
All	**117**	**122**	**0.9660**	**0.9468**	**0.9677**

Accuracy, sensitivity and specificity measures calculated for each defect type are given in Table 1. Accuracy is the ability to distinguish defective and good samples. The proposed method is able to mark 96% of all cases correctly. It has a low accuracy of 76% in color defects, which is essentially because we experimented on monochrome images for the sake of computation time, and the network size and the color characteristics of the samples were ignored. Consequently, color defects that are originally present can be indistinguishable in monochrome image use. Sensitivity measures the ability to differentiate defective samples correctly, just like specificity measures the ability to differentiate good samples.

In order to evaluate the defective region segmentation performance of the method, the relative per-region overlap of the segmentation set S_s and ground truth set S_g is calculated using the Jaccard index, which is the intersection between two sets divided by their union,

$$L_C = \frac{|S_s \cap S_g|}{|S_s \cup S_g|}. \tag{10}$$

Defect detection and defective region segmentation performance of the proposed method is compared with the results of commonly used methods taken from the work of Bergmann *et al*. (2019) and the results are presented in Table 2. The reconstructed images are given in Fig. 3 to exemplify the performance of these methods for different defect types.

Even when tested on a challenging data set, the proposed method is able to detect and segment defective regions successfully. The segmentation success depends on the size of the defect dealt with. Segmentation of thin or tiny defects is a demanding task. One reason for this is the minimum threshold, which results in zero false positives in training images. Altering the number of pixels fit inside the anticipated minimum defect size

Table 2. Detection rates and segmentation (Seg.) performance comparison of the proposed method with other commonly used approaches such as structural similarity autoencoders, L2 autoencoders, anomaly detection with generative adversarial networks, CNN feature dictionary and texture inspection. Top to bottom are the ratio of correctly classified normal samples, correctly classified defective samples and overall segmentation performance values, respectively.

Per.	Ours	AE (SSIM)	AE (L2)	AnoGAN	CNN feature dictionary	Texture inspection
Detection	**0.97**	0.43	0.57	0.82	0.89	0.57
rates	**0.95**	0.97	0.42	0.16	0.36	0.61
Seg.	**0.78**	0.69	0.38	0.34	0.20	0.29

by increasing the resolution of the image can be still an expensive way out.

A visual comparison of standard AE based methods and the proposed method is provided in Fig. 4. Vanilla AEs and denoising AEs are trained to have the same layers as the external AE. Their reconstruction capabilities have been presented to be inefficient by merely evaluating the test results. As mentioned before, the strong reason why both the denoising AE and the vanilla AE fail to reconstruct a defect-free image is their direct connection to the input image causing AEs to cheat. The proposed method overcomes their incompetence by employing an inner AE, which also has the advantage of reconstructing much smaller data. Our method can reconstruct a defect-free sample even in the most challenging cases.

The proposed method is trained on a PC with an Nvidia RTX 2080 GPU, equipped with 32 GB of RAM, an Intel i7-9700k processor, and a Windows 10 64-bit operating system. The network can be trained under six hours. The running time of the algorithm is 88 ms for an image of a 1024×1024 resolution. Considering that 1 mm corresponds to 5 pixels and our method is suitable for parallel operation, it can be deduced that the algorithm can inspect fabric webs flowing at 2 m/s with several parallel cameras. Besides, unlike most models in the literature, this network only has a size of 300 MB and can be used in embedded systems.

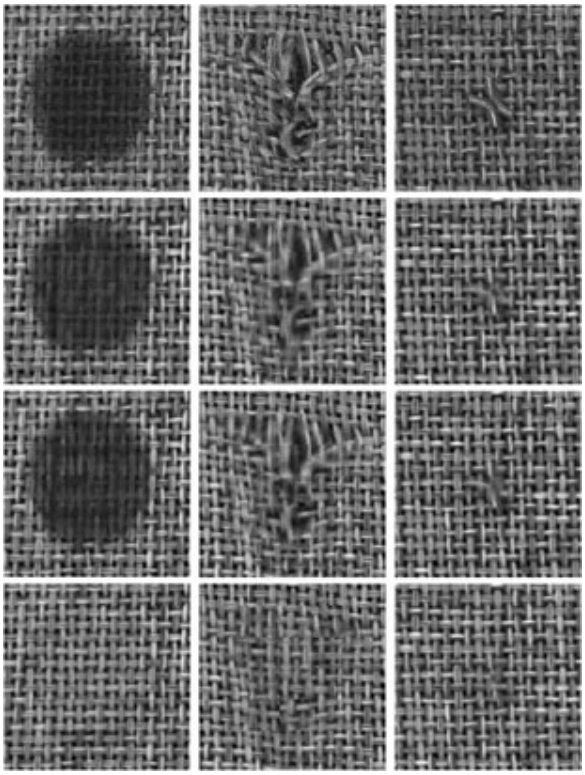

Fig. 4. Comparison of standard AE based methods with the proposed nested AE approach. Top to bottom are the original defective samples, predicted defect-free images of the vanilla convolutional AE, the denoising convolutional AE and the proposed nested AE, respectively.

3. Conclusion

With the push of the developing technology and competitive industrial environment, a great demand has emerged for intelligent visual inspection systems. An intelligent visual inspection system should be applicable to different textured and patterned surfaces and can be adjusted quickly to new products. Moreover, it should keep pace with high production speed. Nested AEs are the main thrust of this study, and the proposed method can recognize surface defects in the real world and it requires only defect-free sample images for training. The first of two nested AEs extracts the features of the image and reconstructs the image from these features. The other is used to identify and fix defects in the feature code. The nested AE is trainable under six hours for very complex patterned surfaces. Our method is suitable for a real-time system due to its small footprint and fast operation. The proposed approach is tested on a real-world complex dataset provided by MVTec with 97% defect detection and 78% pixel-level defective region segmentation accuracy. In future work, we plan

to increase pixel-level detection accuracy by comparing reconstructed defective and defect-free images with a better method instead of a simple subtraction operation.

Acknowledgment

This research is funded by the Scientific and Technological Research Council of Turkey (TUBITAK) under the project number 118E607.

References

Alipour, M. and Harris, D.K. (2020). Increasing the robustness of material-specific deep learning models for crack detection across different materials, *Engineering Structures* **206**: 110157.

Bergmann, P., Fauser, M., Sattlegger, D. and Steger, C. (2019). MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA*, pp. 9592–9600.

Bhattad, A., Rock, J. and Forsyth, D. (2018). Detecting anomalous faces with 'no peeking' autoencoders, *arXiv* 1802.05798.

da Costa, A.Z., Figueroa, H.E. and Fracarolli, J.A. (2020). Computer vision based detection of external defects on tomatoes using deep learning, *Biosystems Engineering* **190**: 131–144.

He, T., Liu, Y., Xu, C., Zhou, X., Hu, Z. and Fan, J. (2019). A fully convolutional neural network for wood defect location and identification, *IEEE Access* **7**: 123453–123462.

He, T., Liu, Y., Yu, Y., Zhao, Q. and Hu, Z. (2020). Application of deep convolutional neural network on feature extraction and detection of wood defects, *Measurement* **152**: 107357.

Hu, G., Huang, J., Wang, Q., Li, J., Xu, Z. and Huang, X. (2019). Unsupervised fabric defect detection based on a deep convolutional generative adversarial network, *Textile Research Journal* **90**(3–4): 247–270, DOI:10.1177/0040517519862880.

Jia, L., Chen, C., Xu, S. and Shen, J. (2020). Fabric defect inspection based on lattice segmentation and template statistics, *Information Sciences* **512**: 964–984.

Kang, X. and Zhang, E. (2019). A universal defect detection approach for various types of fabrics based on the Elo-rating algorithm of the integral image, *Textile Research Journal* **89**(21–22): 4766–4793.

Li, C., Gao, G., Liu, Z., Huang, D. and Xi, J. (2019a). Defect detection for patterned fabric images based on GHOG and low-rank decomposition, *IEEE Access* **7**: 83962–83973.

Li, F., Yuan, L., Zhang, K. and Li, W. (2019b). A defect detection method for unpatterned fabric based on multidirectional binary patterns and the gray-level co-occurrence matrix, *Textile Research Journal* **90**(7–8): 776–796, DOI: 10.1177/0040517519879904.

Li, Y., Zhao, W. and Pan, J. (2016). Deformable patterned fabric defect detection with fisher criterion-based deep learning, *IEEE Transactions on Automation Science and Engineering* **14**(2): 1256–1264.

Lian, J., Jia, W., Zareapoor, M., Zheng, Y., Luo, R., Jain, D.K. and Kumar, N. (2019). Deep learning based small surface defect detection via exaggerated local variation-based generative adversarial network, *IEEE Transactions on Industrial Informatics* **16**(2): 1343–1351.

Lizarraga-Morales, R.A., Correa-Tome, F.E., Sanchez-Yanez, R.E. and Cepeda-Negrete, J. (2019). On the use of binary features in a rule-based approach for defect detection on patterned textiles, *IEEE Access* **7**: 18042–18049.

Luo, Q., Fang, X., Sun, Y., Liu, L., Ai, J., Yang, C. and Simpson, O. (2019). Surface defect classification for hot-rolled steel strips by selectively dominant local binary patterns, *IEEE Access* **7**: 23488–23499.

MVTEC AD (2019). *MVTEC Anomaly Detection Dataset*, MVTec Software GmbH, Munich, `https://www.mvtec.com/company/research/datasets/mvtec-ad`.

Ouyang, W., Xu, B., Hou, J. and Yuan, X. (2019). Fabric defect detection using activation layer embedded convolutional neural network, *IEEE Access* **7**: 70130–70140.

Sun, J., Wang, P., Luo, Y.-K. and Li, W. (2019). Surface defects detection based on adaptive multiscale image collection and convolutional neural networks, *IEEE Transactions on Instrumentation and Measurement* **68**(12): 4787–4797.

Wang, H., Zhang, J., Tian, Y., Chen, H., Sun, H. and Liu, K. (2018). A simple guidance template-based defect detection method for strip steel surfaces, *IEEE Transactions on Industrial Informatics* **15**(5): 2798–2809.

Wang, J., Li, Q., Gan, J., Yu, H. and Yang, X. (2019a). Surface defect detection via entity sparsity pursuit with intrinsic priors, *IEEE Transactions on Industrial Informatics* **16**(1): 141–150.

Wang, R., Guo, Q., Lu, S. and Zhang, C. (2019b). Tire defect detection using fully convolutional network, *IEEE Access* **7**: 43502–43510.

Wei, B., Hao, K., Tang, X.-s. and Ding, Y. (2019). A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes, *Textile Research Journal* **89**(17): 3539–3555.

Xie, H., Zhang, Y. and Wu, Z. (2019). Fabric defect detection method combing image pyramid and direction template, *IEEE Access* **7**: 182320–182334.

Yang, H., Chen, Y., Song, K. and Yin, Z. (2019). Multiscale feature-clustering-based fully convolutional autoencoder for fast accurate visual inspection of texture surface defects, *IEEE Transactions on Automation Science and Engineering* **16**(3): 1450–1467.

Yu, H., Li, Q., Tan, Y., Gan, J., Wang, J., Geng, Y.-a. and Jia, L. (2018). A coarse-to-fine model for rail surface defect detection, *IEEE Transactions on Instrumentation and Measurement* **68**(3): 656–666.

Zhang, Z., Wen, G. and Chen, S. (2019). Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding, *Journal of Manufacturing Processes* **45**: 208–216.

Muhammed Ali Nur Oz received his BS degree from Sakarya University, Sakarya, Turkey, in 2012. He obtained his MS degree in 2015 from Yildiz Technical University, Istanbul, Turkey, which is where he is currently pursuing his PhD degree. He is now a research assistant in control and automation engineering, Yildiz Technical University. His research interests include image processing, pattern recognition, and machine learning.

Ozgur Turay Kaymakci received his PhD degree in control engineering from Istanbul Technical University, Turkey, in 2007. He is currently an associate professor at the Electronics Engineering Department of Canakkale 18 Mart University. His research interests include discrete event systems, industrial automation systems, functional safety and process control.

Muharrem Mercimek received his BS and MS degrees from Yildiz Technical University, Istanbul, Turkey. While pursuing a doctorate he was a research assistant in the Imaging, Robotics, and Intelligent Systems Lab, University of Tennessee, Knoxville, USA, and he was awarded a PhD degree in 2013. Since 2014 he has been a faculty member at the Control and Automation Department, Yildiz Technical University.

Received: 6 March 2021
Revised: 8 May 2021
Accepted: 10 June 2021