Defect detection of laptop appearance based on improved multi-scale normalizing flows

1st Jie Zhang
AHU-IAI AI Joint Laboratory,
Anhui University
Institute of Artificial Intelligence,
Hefei Comprehensive National Science Center
Hefei, China
wa21301041@stu.ahu.edu.cn

2nd Zerui Li
Institute of Artificial Intelligence,
Hefei Comprehensive National Science Center
Hefei, China
lizr@iai.ustc.edu.cn

3th Yunbo Zhao
Department of Automation, USTC
Institute of Advanced Technology, USTC
Institute of Artificial Intelligence,
Hefei Comprehensive National Science Center
Hefei, China
ybzhao@ustc.edu.cn

***Abstract*—In the laptop production process, timely detection of appearance defects is essential to ensure product quality. At present, there are many shortcomings in the manual visual inspection-based method on the laptops production line. In addition, due to the wide variety of laptop appearance defects and extreme differences in defect scales, existing defect detection algorithms perform poorly in the field of laptop appearance inspection. In response to the above problems, this paper proposes a defect detection algorithm based on improved multi-scale normalizing flows. First, the multi-level features extracted from the backbone network are fused by using the pyramid feature fusion module to obtain multi-scale features with rich semantic and spatial information. Then, the effective density estimation of the multi-scale features is achieved by fusing the normalizing flows of attention mechanisms. Finally, the defects are detected and localized based on the output likelihood values. The experimental results demonstrate the effectiveness of the proposed method in detecting and locating appearance defects.**

***Index Terms*—Laptop appearance defect detection, Pyramid feature fusion, Attention mechanism, Normalizing flows.**

I. Introduction

With the continuous development of information technology, more and more information technology is integrated into the industrial manufacturing field, which has prompted the vigorous development of theories and technologies related to intelligent manufacturing. Industrial surface defect detection is one of the key issues in the field of intelligent manufacturing, specifically in the laptop manufacturing industry, timely detection of various appearance defects in the production process is necessary to ensure production safety and product quality. The existing detection process is mainly achieved by manual, which is inefficient and limited by the human eye's inability to accurately identify small defects.

During the production of laptops, appearance defects are generated randomly. There are defects such as fingerprints and water stains generated by human factors, or scratches caused by automated equipment. Based on actual production line surveys, we summarize the main challenges of laptop appearance defect detection: 1) the probability of defects generated during normal production is extremely small, which makes defect samples difficult to collect. 2) some defect types may have never appeared before, requiring detection algorithms with better generalization. 3) the scale of different types or even the same type defects varies greatly, and the difficulty of detecting small-scale defect types is often greater than that of large-scale defect types.

To summarize the defect detection methods based on traditional digital image processing and machine learning [1], most of the methods are limited by specific scenarios or rely on expert knowledge, and the speed and accuracy of detection are not high, and are prone to missed inspection and false inspection. In recent years, high-precision detection methods based on deep learning have received more attention, but the large demand for defective samples in the training process limits their application in industry.

In response to the above problems, motivated by [1], an improved defect detection algorithm with multi-scale normalizing flows is proposed in this paper. Which can effectively extract the multi-scale features of the image and estimate its density for defect detection, the framework of the algorithm is shown in Figure 1, and the details are explained in Section 3.

This work was supported by the National Natural Science Foundation of China (No. 62173317) and the Key Research and Development Program of Anhui (No. 202104a05020064).(Corresponding authors: Yunbo Zhao)

Compared with [2], the novelty of this paper lies in the use of pyramidal feature fusion to extract multi-scale features more effectively. Also, the shuffle attention mechanism is introduced in the normalizing flows network [3], which enables the normalizing flows to better fit the data distribution. In addition, the cross-scale fully convolutional module is further enhanced to make full use of multi-scale feature information. We evaluate our method with data collected on a laptop production line and achieve an AUROC score of 99.2% for defect detection at the image-level.

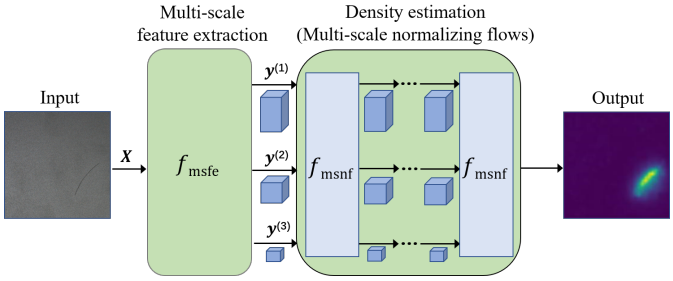

Fig. 1: The general framework of the proposed method. It mainly consists of two steps: multi-scale feature extraction and density estimation.

II. Related Work

In simple terms, defect detection at the image level is similar to image anomaly detection, i.e. the algorithm is expected to be able to distinguish between normal and abnormal images. Specifically in the field of industrial surface defect detection, it is often also necessary to locate the location of the defect. In the following, a brief review of various surface defect detection methods similar to our task is presented, which we classify into generative based methods and feature representation based methods and other deep learning methods.

A. Generative based methods

The basic idea of defect detection methods based on generative models is that the models are able to reconstruct images of normal samples. Typical methods such as AE [4]and VAE [5]are trained with the aim of reconstructing normal images, and then the difference between before and after the reconstructed images is used as the anomaly score in the inference stage to determine whether the samples are defective. Based on the idea of GAN [6], zhao et al. [7] proposed a reconstruction network that aims to fix normal samples with added artificial defects in the training stage. The defects are detected in the test stage by comparing the original image with the repaired image through local binary pattern algorithm.Akcay et al. [8]combined the ideas of AE and GAN and proposed a novel adversarial autoencoder within an encoder-decoder-encoder pipeline. The reconstruction encoder is first trained to learn the potential representation vectors of the reconstructed image, and then detects image by comparing the differences between the embedding vectors of the original image and the reconstructed image in the testing stage. Due to the strong generalization ability of neural networks, this generative model also reconstructs the anomalous samples well and is prone to miss detection, resulting in low detection accuracy.

In contrast to AE and GAN, generative models based on normalizing flows [9] can explicitly estimate the data distribution density.Rudolph et al. [10] proposed a feature density estimation method capable of embedding normal images into a standard Gaussian distribution and determining defects by probability estimation in the inference stage. Based on [10], a cross-scale fully convolutional normalizing flows density estimation method was proposed by Rudolph et al [2]. which is able to utilize multi-scale feature information and retains the spatial structure for visualization. Although the above algorithm is effective in image level detection, it ignores the importance of spatial information in low-level features by adjusting the input image size to extract multi-scale features, resulting in failure to locate defect locations well.

B. Feature representation based methods

For the feature representation based methods, the basic idea is to obtain a feature extraction network by pre-training to make the distance between the feature vectors of normal images as small as possible in the training stage. In the testing stage, defect are determined by calculating the distance between the feature vectors of the test samples and the normal samples.Cohen et al. [11] store a pool of features of normal samples in the training stage. In the test stage, the nearest K features in the feature pool are found to calculate the anomaly score to determine the anomaly, and then the defect is localized by feature pyramids. However, the number of features to be stored during training is linearly related to the number of normal samples, which leads to high complexity during testing. Based on [11], Defard et al. [12] estimated the distribution of each location in the normal image by multi-level features. In the test stage, the difference between each position and the corresponding distribution is calculated as the anomaly score for that position, and finally the maximum value is selected as the image anomaly score. Due to the simple alignment of image positions, this method does not work well for detecting objects with large changes in position.

C. Other methods

Motivated by the excellent algorithms in the field of object detection, such as the two-stage algorithm Faster RCNN [13], and the one-stage algorithm YOLO [14]. They were improved and introduced to industry for the detection of various types of surface defects, such as fabrics [15], steel [16], etc. Such supervised learning algorithms can achieve excellent detection results under the condition that the number of defect samples is sufficient. However, in many industrial application scenarios, it is difficult to collect a sufficient number of defect samples for algorithm training, and it is also time-consuming and laborious to manually label the defect samples. Moreover, there may be unseen defect types in industrial production, and supervised learning-based algorithms are not well suited to detect defect

types beyond the training set. All these reasons limit the application of target detection algorithms in industrial defect detection.

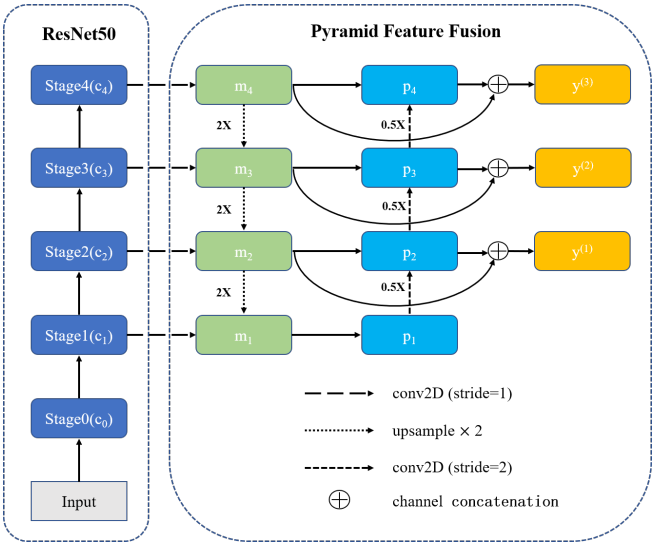

Fig. 2: The framework of multi-scale feature extraction module.

III. Method

As shown in Fig. 1, our method is divided into two steps,multi-scale feature extraction ($X \; ! \; Y$) and density estimation ($Y \; ! \; Z$). First, in the training phase, we combine the pre-trained backbone network ResNet [17] and the pyramid feature fusion module as a feature extraction network to extract multi-scale features ($y \; 2 \; Y$) from normal samples ($x \; 2 \; X$).Then they are fed into the multi-scale normalizing flows incorporates the shuffle attention, which is then subjected to density estimation. The density estimation can map the unknown distribution p_Y in the feature space to a multivariate standard normal distribution p_Z. i.e.

$$f_{\text{msnf}} \left(\begin{matrix} h \\ y^{(1)};:::;y^{(s)} \end{matrix} \right) = \left(\begin{matrix} i \\ z^{(1)};:::;z^{(s)} \end{matrix} \right) = z \; 2 \; Z \tag{1}$$

Where s indicates the number of feature scales, Z has the same dimensions as Y. Based on the distribution Z we can obtain the likelihood $p_Z(z)$ of the input data x.

The model can fit the data distribution of normal samples well after training. In the inference phase, likelihood $p_Z(z)$ is obtained by density estimation for the feature Y of the input image x, and determine whether the input sample is a defective sample based on the $p_Z(z)$. Since only normal samples are utilized in the training phase and various unknown defects are effectively detected in the testing phase. Therefore, the proposed method is an unsupervised defect detection algorithm.

A. Pyramid Feature Fusion

Motivated by [18], this paper proposes a multi-scale feature extraction module based on pyramid feature fusion as shown in Fig. 2. The left side of which is the pre-trained ResNet50 backbone network (with the final pooling layer and fully connected layer removed), and the multi-scale features $[c_1; c_2; c_3; c_4]$ are obtained using the backbone network. On the right side are two opposite feature fusion paths. In the top-down path, except for the top-level feature m_4, which is obtained by changing the number of channels by 1*1 convolution of c_4 (the number of output channels is fixed to 256), the rest of the feature maps $m_i (i = 1; 2; 3)$ are obtained by fusing a shallow feature map c_i with a deeper feature map m_{i+1}. In the bottom-up path, the feature maps $p_i (i = 2; 3; 4)$ are obtained by fusing a shallow feature $p_{i \;\; 1}$ and a deeper feature m_i, except for the bottom feature p_1 which is directly copied from m_1. The formula is described as (2). Finally, the high-level features $[m_2; m_3; m_4]$ and $[p_2; p_3; p_4]$ are concatenated along the channels to obtain the multi-scale features $y^{(1)}; y^{(2)}; y^{(3)}$.

$$\begin{aligned} m_i &= Conv_{1;3} \left(Conv_{1;1} \left(c_i \right) \quad Up_2 \left(m_{i+1} \right) \right) \\ p_i &= Conv_{1;3} \left(Conv_{2;3} \left(p_{i \;\; 1} \right) \quad m_i \right) \end{aligned} \tag{2}$$

where $Conv_{s;k}$ denotes a convolution operation with step s and kernel size of $k \quad k$; denotes element-wise sum; Up_2 denotes $2x$ bilinear interpolation upsampling operation.

The top-down feature fusion path transfers the rich global information from deeper features to shallow features, and the bottom-up feature fusion path transfers the rich local information from shallow features to deeper features. With these two feature fusion paths, the extracted multi-scale features have both rich spatial and semantic information, which improves the detection performance of small defects while reducing the computational complexity.

B. Shuffle Attention

Shuffle attention [3] is a lightweight channel-spatial attention mechanism that can be easily embedded in neural networks. Compared with similar attention mechanisms [19], the introduction of feature grouping strategy reduces the computational complexity. First, shuffle attention groups the feature vectors along the channel dimension and divides them into two parts by channel dimension. One part learns channel attention features and assigns different weights to each channel. The other part learns spatial attention features to focus feature information on important regions. Finally, the channel attention and spatial attention are effectively combined using a shuffle unit.

C. Multi-scale Normalizing Flows

The multi-scale normalizing flows is a flow model consisting of a sequence of neural networks, where each sub-network is equivalent to an affine transformation of the input. We extend the sub-network in [2] with shuffle attention mechanism, which allows the model to better utilize feature information. As shown in Fig. 3, first we input the multi-scale feature $y = y^{(1)};:::;y^{(s)}$ into the corresponding shuffle attention separately to fuse spatial attention and channel attention to

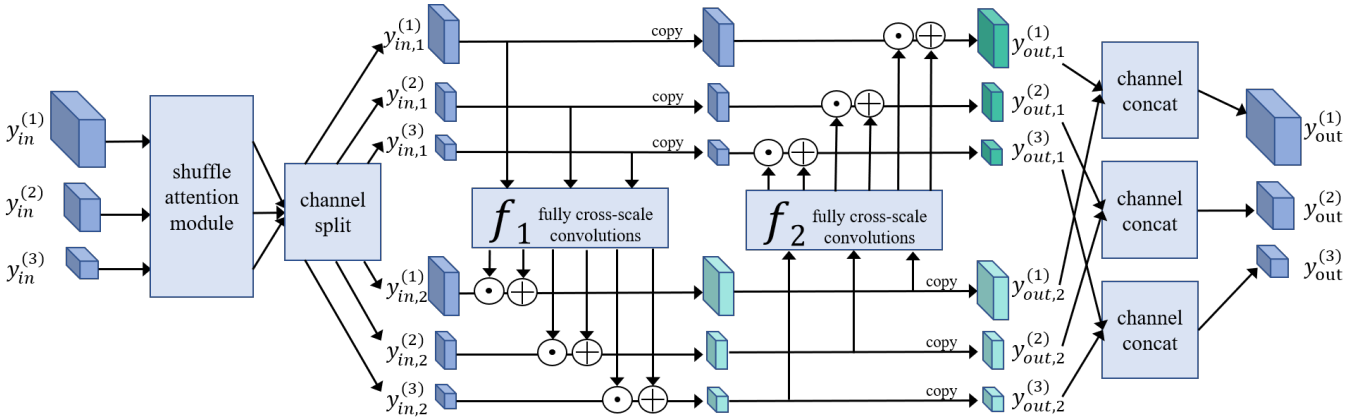

Fig. 3: Architecture of sub-networks in multi-scale normalizing flows. The multi-scale feature after fused shuffle attention are divided into two parts along the channel dimension, then input to the internal sub-network respectively to calculate the scale and shift parameters, and finally applied to the corresponding parts. The symbols and denote the element-wise multiplication and addition, respectively.

improve the feature ground semantic representation. The transformed multi-scale feature $y_{\text{in}}^{(i)}$ is then divided into two parts $y_{\text{in};1}^{(i)}$ and $y_{\text{in};2}^{(i)}$ by channel dimension, and these two parts of the feature vector are successively calculated by two internal sub-networks f_1 and f_2 to obtain the scale parameters $[s_1; s_2]$ and shift parameters $[t_1; t_2]$, which are subsequently applied to their respective corresponding parts as follows:

$$\begin{aligned} y_{\text{out};2} &= y_{\text{in};2} \quad e^{\ {}_1 s_1(y_{\text{in},1})} + \ {}_1 t_1 \quad y_{\text{in};1} \\ y_{\text{out};1} &= y_{\text{in};1} \quad e^{\ {}_1 s_2(y_{\text{out},2})} + \ {}_2 t_2 \quad y_{\text{out};2} \end{aligned} \tag{3}$$

with $y_{\text{out}}^{(i)}(i \ 2 \ f1;:::;sg)$ denotes the output of the current sub-network, and denotes element-wise product. $[\ {}_1;\ {}_2]$ are learnable parameters initialized to 0. The details of the structure of the internal sub-networks f_1 and f_2 is shown in Fig. 4, which we implement it via a fully cross-scale convolutional network, allowing it to take full advantage of the multi-scale feature information. The output is uniformly partitioned into scale and shift parameters by channel dimension.

D. Training Objective

During the training process, original feature space Y are mapped to pre-defined latent space Z via multi-scale normalizing flows f_{msnf}. The objective of training is to maximize the likelihood of the feature tensor $p_Y(y)$ by normal image. By mapping $z = f_{msnf}(y)$ and according to the variational change formula (3) , we describe the training objective as maximizing:

$$p_Y(y) = p_Z(z) \ \det \frac{@z}{@y} \tag{4}$$

By pre-defined Z as a standard Gaussian distribution, equation (3) can be simplified to be equivalent to minimizing the negative log likelihood $\log p_Y(y)$:

$$\begin{aligned} \log p_Y(y) &= \log p_Z(z) + \log \ \det \frac{@z}{@y} \\ L(y) = \quad \log p_Y(y) &= \frac{kzk_2^2}{2} \quad \log \ \det \frac{@z}{@y} \end{aligned} \tag{5}$$

with $\det \frac{@z}{@y}$ denoting the absolute value of the Jacobi determinant, which in this case refers to the Jacobi determinant of (3). Since the Jacobi of the element-wise product operator in (3) is a diagonal matrix, the Jacobi determinant can be simplified to the sum of all scale parameters $[s_1; s_2]$.

E. Detection and Localization

In the inference stage, the determination of defect based on the likelihood $p_Z(z)$ obtained from the density estimation and threshold . By calculating the mean of the output likelihood squares on all scales, which is considered as the anomaly score at the image-level.

$$A(x) = \begin{cases} 1 & \text{for } p_Z(z) < \\ 0 & \text{else} \end{cases} \tag{6}$$

where $A(x) = 1$ indicates that the input is a defective sample.

For defect localization, since proposed method is based on convolutional operations to process the feature maps, the spatial location information is preserved, which allows the method to use the output likelihood of each location $(i; j)$ of the image for defect localization. By aggregating the output $kz_{i;j}^{s}k_2^2$ along the channel dimension to obtain the anomaly score for each position of the feature map $y^{(s)}$. It is then upsampled to the resolution of the input image by bilinear interpolation. The final visualization result of defect localization is obtained.

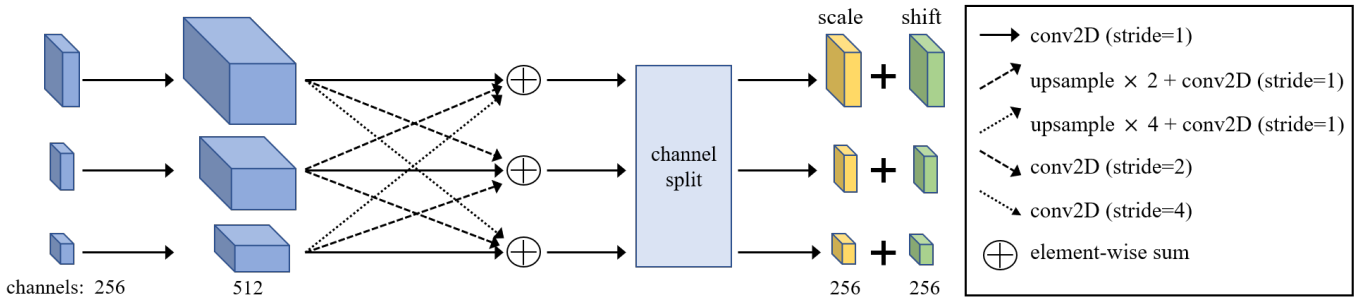

Fig. 4: Architecture of the internal sub-network inside the Fig. 3. The cross-information interaction between different scales is achieved by upsampling and strided convolution. Aggregation is implemented by element-wise sum. The output is split in the channel dimension to obtain scale parameters s and shift parameters t.

IV. Experments

In this section, we compare with other methods and perform ablation experiments to prove the performance of proposed method.

A. Dataset

The dataset is composed of images collected from a real laptop production environment. It contains four types of surface defects, i.e., fingerprints, rough scratches, tiny scratches, and water stains. The dataset contains a total of 962 images with a resolution of 512 512, which contains 706 normal samples for training and 256 test samples, and the test samples consist of 76 normal samples and 180 defective samples.

B. Experimental Detail.

We use the number of feature scales of s=3, and the input image is 512 512 resolution to obtain multi-scale feature maps of size 64 64, 32 32, and 16 16 with 512 channels each. Four serial sub-networks are used in the multi-scale normalizing flows. The training phase is set with batch size of 16 and training epoch of 120. Whole experiments are performed under a single NVIDIA GeForce RTX 3090 24G GPU.

C. Result Analysis.

To verify the performance of the proposed method in the detection of appearance defects in laptops, the area under the receiver operating characteristic curve (AUROC) at the image-level on the test set was calculated as a defect detection performance metric. The receiver operating characteristic curve (ROC) relates the true positive rate and false positive rate through the threshold in (6), which is insensitive to the ratio of abnormalities in the test set.

Experimental results are compared with several other excellent defect detection algorithms as shown in TABLE I. It can be seen that the proposed method has the highest image-level AUROC score of 99.30% among all the detection results. Compared to [2], our improved model improves by almost 1.3%, which is crucial in industrial applications.

As described in Section III-E, the results of partial defect localization visualization are obtained based on $y^{(1)}$ as shown in Fig. 5, which shows that the proposed method is able to effectively capture various types of appearance defects. Due to the restored image resolution by the upsampling operation, it leads to a slightly rough defect edge localization, but our target is not the exact segmentation of the defect location, and the visualization result by this is sufficient to help the operator locate the defect quickly.

TABLE I: Detection Performance of Different Method

Method	AUROC(%)
GANomaly [8]	84.72
PaDim [12]	95.86
DifferNet [10]	97.10
CS-Flow [2]	98.03
Ours	**99.30**

TABLE II: Results of Ablation Experiment

Experimental configuration	A	B	C	D
Multi-scale	×	√	×	√
Attention mechanism	×	×	√	√
AUROC(%)	95.56	98.32	96.68	**99.30**

D. Ablation Experiments.

To quantify the impact of the improved strategies in our work, we conducted ablation experiments by comparing the detection performance of laptop appearance defects under different configuration strategies. Four groups of ablation experiments were designed as shown in TABLE II. The detection metrics increased significantly after the introduction of the multi-scale feature extraction module alone, which indicates that the fusion of spatial and semantic information plays a significant contribution to defect detection. The small increase in metrics after adding the shuffle attention mechanism alone indicates that improving the semantic expression of features also plays a contributing role. After adding both modules simultaneously, the metrics reach the highest, which demonstrates the effectiveness of the improvements.