Multiresolution Knowledge Distillation and Multi-level Fusion for Defect Detection

Huosheng Xie(✉) and Yan Xiao

Fuzhou University, Fuzhou, China
{xiehs,200327150}@fzu.edu.cn

Abstract. Defect detection has a wide range of applications in industry, and previous work has tended to be supervised learning, which typically requires a large number of samples. In this paper, we propose an unsupervised learning method that learns knowledge about normal images by distilling knowledge from a pre-trained expert network on ImageNet to a learner network of the same size. For a given input image, we use the differences in the features of the different layers of the expert network and learner network to detect and localize defects. We show that using comprehensive knowledge makes the differences between the two networks more apparent and that combining the differences in multi-level features can make the networks more generalizable. It's worth noting that we don't need to split the picture into patches to train, and we don't need to design the learner network additionally. Our general framework is relatively simple, yet has a good detection effect. We provide very competitive results on the MVTecAD dataset and DAGM dataset.

Keywords: Defect detection · Unsupervised learning · Knowledge distillation · Multi-level fusion

1 Introduction

During the manufacturing process of industrial products, various unavoidable defects may appear in the products, such as spots, scratches, cracks, etc. Previous defect detection methods use a supervised learning approach, which requires expensive annotation costs and a low probability of defect occurrence, which can lead to a serious imbalance in the ratio of normal to defective images in the dataset. In recent years, unsupervised defect detection methods have become increasingly popular in industry [1, 2].

Usually, in unsupervised defect detection, the defect detection problem is treated as an anomaly detection problem. During training, only normal samples (samples without defects) are used, with the aim that the network learns only the features of normal samples. In the testing phase, when the input samples contain defects, the network will output results with significant differences from the normal samples, and the identification of abnormal samples (defective samples) can be achieved by detecting the differences from the normal samples. Attention was also directed to the localization of anomaly detection, expecting pixel-level localization of defective regions in the image, which is a challenging task, but it has extraordinary significance for practical applications.

C. Yu et al. (Eds.): GPC 2022, LNCS 13744, pp. 178–191, 2023.
https://doi.org/10.1007/978-3-031-26118-3_14

Much of the existing work is mainly embodied in generative models, such as autoencoders (AE) [3–6] and generative adversarial networks (GAN) [7–10]. However, due to the powerful generalization ability of the deep autoencoder, even anomalous samples containing defects can be well reconstructed, which defeats the original purpose. The literature [6] mentions that the GAN-based approach has the following two shortcomings: non-reproducibility of the results [11, 12] and data hungriness. Recent studies [13–15] have shown that these methods do not extract the semantic features well.

Using pre-trained networks can greatly increase the training speed of the model and effectively improve the accuracy of the model [16, 17]. Salehi et al. [18] proposed MKD, they extract knowledge from multiple layers of the pre-trained source network, which can better exploit the knowledge of the source network and expand the discrepancy compared to using only the last layer of information. The loss function is the similarity of the multi-layer feature maps of the source and cloner networks, using a weighted sum of MSE and cosine similarity. Moreover, the localization uses a gradient-based interpretable method, where they consider the anomalous region to be the region that makes the sudden and large change in the value of this loss, find the back-propagation gradient of the loss, and use the gradient to find the region that causes the anomaly that increases its value. We found that this method is not effective in detecting tiny defects as well as defects in texture-based products.

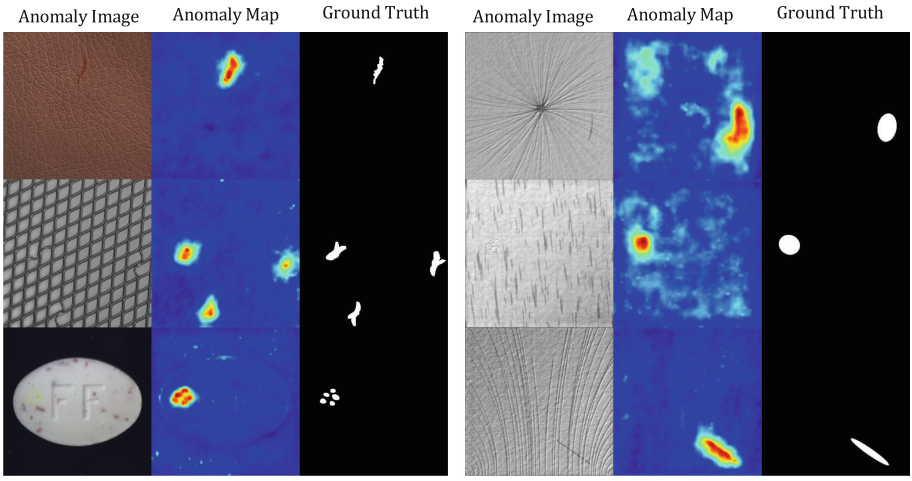

Fig. 1. Visualized results of our method on MVTecAD dataset and DAGM dataset.

To be able to better detect small defects as well as defects in textured products, we offer an alternative strategy. First, we follow the framework of knowledge distillation, distilling knowledge from one network to another. Our expert network is a pre-trained VGG16 network model [19] on the ImageNet dataset [20], and the learner network is the same size as the expert network, but the learner network is not pre-trained. In the training phase, the normal images without any defects are sent to the expert network and the learner network respectively, and the learner network will acquire different semantic information at multiple levels in the expert network, and it should not be neglected

that the learner network only learns the manifold of normal data sufficiently. When an image with defects is input, the learner network and the expert network will diverge, and the greater the difference between the features of the defective and normal images, the greater their divergence will be, and the two networks will show different divergences at different layers. We only use MSE Loss to distill knowledge during the training period. In the testing phase, we use cosine similarity to obtain anomaly maps between the two networks at different levels. The value of each pixel on the anomaly maps represents the degree to which the expert network diverges from the learner network, and the more pronounced this divergence is, the more likely it is that a defect exists. By fusing multiple levels of anomaly maps, we can have excellent detection and localization effects on different types of defects (see Fig. 1). Compared with MKD [18], our method can effectively detect and locate different types of defects, especially in textured products, and has a significant improvement in the accuracy of detection. In addition to using the MVTecAD dataset, we also tested our experimental results on the DAGM dataset containing various types of texture patterns, and the data showed that our method can have excellent results in detecting defects in texture-based products.

2 Related Work

2.1 Image Reconstruction

A typical reconstruction-based approach uses an autoencoder to compress the input image. During training, the model only reconstructs the normal samples for learning, and the defective regions cannot be reconstructed well, and the presence of defects is determined based on the reconstruction error between the input data and the reconstructed data. Bergmann et al. [3] introduced structural similarity SSIM to a general autoencoder, integrating luminance, contrast and structural information to compensate for the visual inconsistency caused by reconstruction errors using the Euclidean distance metric alone. Some other methods [7–9, 21] use Generative Adversarial Network to constrain the data distribution.

Some approaches use self-supervised learning to force the model to learn semantic features of the image itself from restricted information. Golan et al. [22] subjected the normal samples to multiple geometric transformations. Similarly, Fei et al. [14] proposed that ARNet adds an attribute erasure module to the autoencoder framework to erase the color information and perform geometric transformations. Puzzle-AE [6] introduces another common self-supervised learning task, puzzle decryption. RIAD [23] based on image restoration for anomaly detection.

2.2 Feature Modeling

Yi et al. [24] improved on SVDD [25] and Deep SVDD [26] by dividing the whole image into several patches. Shi et al. [27] develop an effective feature reconstruction mechanism for anomaly detection. Cohen et al. [28] combined the idea of KNN and extraction of multilevel features to achieve good results in pixel-level localization. Wang et al. [29] achieved superior results in localization accuracy by extracting features from the ResNet

intermediate layer and using a step-by-step phase multiplication method. CAVGA [30] makes clever use of the attention mechanism and expects the model to focus on the normal regions of the image. Bergmann et al. [15] first applied the knowledge distillation method to anomaly detection, since the training is based on patches, the training cost is too high and heavily depends on the size of patches.

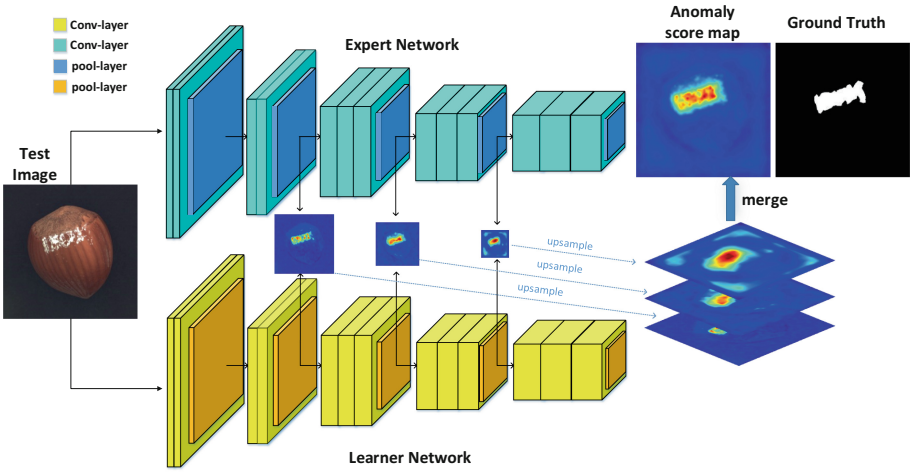

Fig. 2. Overview of our framework. During the training phase, the learner network learns the manifold of the normal data from the expert network. During the testing phase, detect and locate defects by fusing anomaly maps from multiple layers of both networks.

3 Method

This section describes in detail our proposed method for defect detection. Given a training dataset $D_{train} = \{I_1, I_2, \ldots, I_n\}$ without any defective images, we will use a pre-trained expert network E to distill knowledge to the learner network L, that detects defects in the test set, D_{test}. Once L learns the manifold of the normal data, it can assign a score to each pixel indicating how much it deviates from the training data manifold. Therefore, it has to try to learn the complete knowledge of E. The previous work related to knowledge distillation simply taught the final output information. The knowledge of the middle layer of the expert network is sometimes even better than the knowledge of the last layer [31]. For this, we encourage L to learn multi-level of knowledge, which will enable it to fully learn the normal data in the manifold. As we all know, in deep neural networks, different levels of features represent different meanings. For example, the output of the first convolutional layer is just some very simple line information, followed by possibly some shape-related information. The deeper the hierarchy goes, the easier it is to get some semantically relevant information.

Figure 2 illustrates our proposed framework.

3.1 Knowledge Distillation

In this section, we focus on how L learns the manifold of the normal data from E.

The network we use is VGG16, and the features extracted by the VGG network have demonstrated superior performance in many application directions in the field of computer vision. We call the layer where knowledge needs to be distilled the reserve layer, and define the i-th reserve layer as R_i. The features output by E at the reserve layer are called $a_E^{R_i} \in \mathbb{R}^{w \times h \times d}$, where w, h, and d denote the width, height and channel number of the feature, the features output by L at the reserve layer are called $a_L^{R_i} \in \mathbb{R}^{w \times h \times d}$, we define the distillation loss l_i of the i-th layer as

$$l_i = \frac{1}{w \times h}(a_E^{R_i} - a_L^{R_i})^2. \tag{1}$$

In order to distill multiple levels of knowledge, then the total distillation loss can be defined as

$$l = \sum_{i=1}^{N_R} \lambda_i (\frac{1}{w \times h}(a_E^{R_i} - a_L^{R_i})^2), \tag{2}$$

where N_R represents the number of reserve layers, and λ_i indicates the impact of the i-th feature scale on anomaly detection. We set all the weights by default to $\lambda_1 = \lambda_2 = \ldots = \lambda_{N_R} = 1$

To prevent some undesirable effects and additional interference factors caused by inconsistent network structures, such as inconsistent network structures leading to some differences in the output of the middle layer itself, etc. The structure of L we use is identical to that of E, the only difference being that E is pre-trained, whereas L is not. During training, we input only normal image samples and no abnormal image samples containing defects, and we keep the parameters of E unchanged while updating only the parameters of L.

The framework used for the training process is shown in Fig. 3.

3.2 Anomaly Detection

To detect possible defects contained in the images and where they are located, we feed each image into both E and L. For a normal image input without defects, the two have almost the same view of the normal image because L is well equipped with the knowledge about the normal image instilled by E. Therefore the features output by the two networks are almost identical. But for an image that contains defects, since E is pre-trained on ImageNet, and L only learns the knowledge about normal images taught by E. So the features output by the two networks for the defective image may not be consistent, and the inconsistent area is the area where the defect is located. Based on this feature, we can discern whether an image is a defective image or not, and to find the location where the defect is located.

In convolutional neural networks, local features extracted by convolutional layers are combined by subsequent convolutional layers to form more complex features. In this learning process, a hierarchy of features emerges, where lower-level convolutional layers

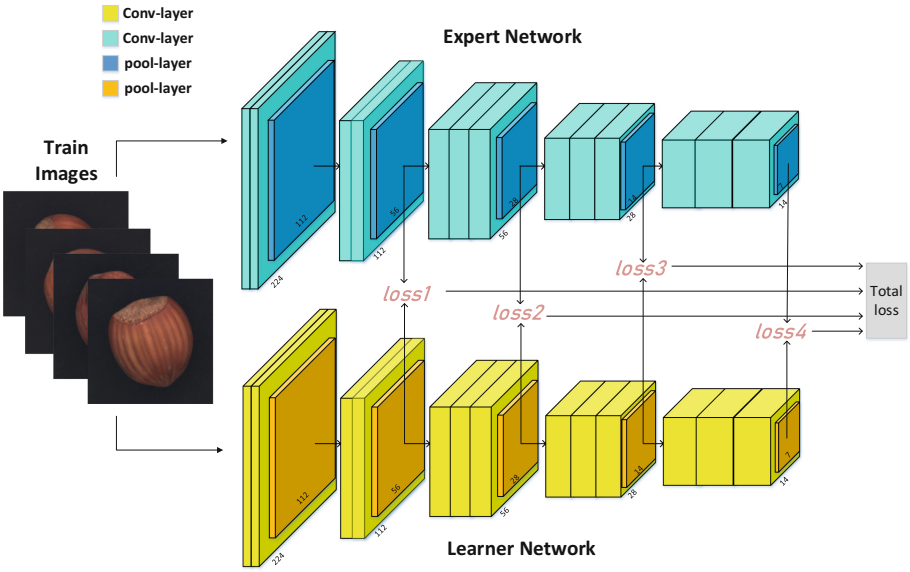

Fig. 3. The process of knowledge distillation. Knowledge from the middle and last layers of the expert network is distilled into the learner network. This knowledge is about the manifold of the normal data.

may learn lower-level features (edges, corners, etc.), while higher-level convolutional layers may learn more advanced features (dog heads, bird tails, etc.). The output of different layers of a convolutional neural network corresponds to different levels of features, and the different levels of features also represent different meanings. Intuition tells us that when an image with defects is input to the network, the features a_L^R output by each level of L will differ to a different degree from the features a_E^R output by each level of E, and we combine the differences in the features of multiple levels as a way to improve the detection accuracy of the network, and the experiments prove that our idea is correct.

In the testing phase, we use the cosine similarity to measure the difference between the features output by the two networks. The difference exhibited by two features a_L^R and a_E^R of dimension d × w × h is represented by an image of size w × h, this image is the anomaly map M_i of the current layer. M_i is formulated as

$$M_i = CosineSimilarity\left(a_E^{R_i}, a_L^{R_i}\right). \tag{3}$$

Since the size of the anomaly maps generated by each layer is inconsistent, it is necessary to upsample all the anomaly maps to a uniform size, noted as M_i^*, for subsequent fusion. M_i^* is formulated as

$$M_i^* = Upsampling\left(M_i\right). \tag{4}$$

The final generated anomaly map M is formulated as

$$M = \sum_1^N \beta_i M_i^*, \tag{5}$$

where β_i indicates the impact of the i-th anomaly map, N represents the number of anomaly maps to be fused.

4 Experiments

In this section, we investigate the performance of our model in different datasets, and in each dataset, we test the performance of the model in defect detection and the performance of the model in defect region localization, respectively. The results of the experiments show that we achieve good performance in both the anomaly detection task and the anomaly localization task.

4.1 Dataset

MVTecAD. MVTecAD is a dataset for anomaly detection. Unlike previous anomaly detection datasets, which mimic actual industrial production scenarios and are primarily used for unsupervised anomaly detection, this dataset is more focused on real-world applications. The dataset contains 5354 high-resolution color images of different objects (ten categories) and texture categories (five categories). The dataset is further divided into normal images for training and anomalous images for testing, with 73 anomaly types, such as scratches, dents, contamination and various structural changes, all of which are labeled at the pixel level.

DAGM2007. The DAGM2007 dataset [32] is a dataset for fabric defect detection that contains ten different classes of images. Since the DAGM2007 dataset was originally prepared for supervised and weakly supervised tasks, which contains some classification annotations, we need to discard this part of annotations and rearrange the DAGM2007 dataset to keep it consistent with the MVTec AD dataset so that it can successfully complete the task of unsupervised anomaly detection. Our reproduced DAGM2007 dataset contains 3858 images, and all ten categories are texture-based images with different sizes of defects in different categories.

4.2 Experimental Setup

Both E and L use the VGG16 network model with the same structural size. During the training phase, we choose the four final layers of each convolutional block, i.e. max-pooling layers, to be the reserve layer. Unlike the training phase, in the testing phase, for an input image of 224×224, the size of the anomaly maps output through these four storage layers are 56×56, 28×28, 14×14, and 7×7, and finally we have to upsample all the anomaly maps to the same size as the input image, which is 224×224. So for too small sizes, the upsampling process will produce some errors, which we do not want to see, so in the test phase to ensure that there is not too much interference, we discarded the output of the last storage layer 7×7, i.e. only 56×56, 28×28 and 14×14 anomaly maps are used.

Table 1. Image-level detection results on MVTecAD.

	Category	L2_AE	AnoGAN	LSA	CAVGA	VAE	MKD	PatchSVDD	STFPM	OURS
Textures	Leather	46.0	52.0	70.0	71.0	71.0	95.1	90.9	–	**99.8**
	Wood	83.0	68.0	75.0	85.0	89.0	94.3	96.5	–	**99.3**
	Carpet	67.0	49.0	74.0	73.0	67.0	79.3	92.9	–	**98.5**
	Tile	52.0	51.0	70.0	70.0	81.0	91.6	97.8	–	96.9
	Grid	69.0	51.0	54.0	75.0	83.0	78.0	94.6	–	**99.3**
Objects	Bottle	88.0	69.0	86.0	89.0	86.0	99.4	98.6	–	**99.2**
	Hazelnut	54.0	50.0	80.0	84.0	74.0	98.4	92.0	–	**98.5**
	Capsule	61.0	58.0	71.0	83.0	86.0	80.5	76.7	–	**95.8**
	Metal Nut	54.0	50.0	67.0	67.0	78.0	73.6	94.0	–	**99.6**
	Pill	60.0	62.0	85.0	88.0	80.0	82.7	86.1	–	**98.4**
	Cable	61.0	53.0	61.0	63.0	56.0	89.2	90.3	–	**92.3**
	Transistor	52.0	67.0	50.0	73.0	70.0	85.6	91.5	–	**91.8**
	Toothbrush	74.0	57.0	89.0	91.0	89.0	92.2	**100**	–	88.3
	Screw	51.0	35.0	75.0	77.0	71.0	83.3	81.3	–	**93.3**
	Zipper	80.0	59.0	88.0	87.0	67.0	93.2	**97.9**	–	97.1
	Mean	63.0	55.0	73.0	78.0	77.0	87.8	92.1	95.5	**96.5**

Table 2. Pixel-level detection results on MVTecAD.

	Category	SSIM_AE	L2_AE	AnoGAN	CNN_Dict	VAE	MKD	PatchSVDD	STFPM	OURS
Textures	Leather	78.0	75.0	64.0	59.0	92.5	98.1	97.4	**99.3**	98.6
	Wood	73.0	73.0	62.0	91.0	83.8	84.8	90.8	**97.2**	94.5
	Carpet	87.0	59.0	54.0	72.0	73.5	95.6	92.6	98.8	**99.0**
	Tile	59.0	51.0	50.0	93.0	65.4	82.8	91.4	**97.4**	96.7
	Grid	94.0	90.0	58.0	59.0	96.1	91.8	96.2	**99.0**	98.9
Objects	Bottle	93.0	86.0	86.0	78.0	92.2	96.3	98.1	**98.8**	98.5
	Hazelnut	97.0	95.0	87.0	72.0	97.6	94.6	97.5	**98.5**	98.3
	Capsule	94.0	88.0	84.0	84.0	91.7	95.9	95.8	**98.3**	91.0
	Metal Nut	89.0	86.0	76.0	82.0	90.7	86.4	98.0	**97.6**	96.5
	Pill	91.0	85.0	87.0	68.0	93.0	89.6	95.1	**97.8**	97.0
	Cable	82.0	86.0	78.0	79.0	91.0	82.4	96.8	**95.5**	94.8
	Transistor	90.0	86.0	80.0	66.0	91.9	76.5	97.0	**82.5**	80.0
	Tootbrush	92.0	93.0	90.0	77.0	98.5	96.1	98.1	98.9	**99.0**
	Screw	96.0	96.0	80.0	87.0	94.5	96.0	95.7	**98.3**	96.7
	Zipper	88.0	77.0	78.0	76.0	86.9	93.9	95.1	**98.5**	97.6
	Mean	87.0	82.0	74.0	78.0	89.3	90.7	95.7	**97.0**	95.8

For all the following experiments, we will use the framework shown in Fig. 2. In which, we do a Batch Normalization operation after each convolutional layer, not only for E but also for L. Batch Normalization allows each layer of the network to learn

itself slightly more independently of the other layers. The SGD optimizer is used in the experiment, the learning rate is set to 0.3, the batch size is 32, all the input images are resized to 224 × 224, and the final output image is also 224 × 224 in size.

4.3 MVTec Anomaly Detection Dataset

As in previous work, the area under the receiver operating characteristic curve (AUROC) was used as the metric used to evaluate the experiments.

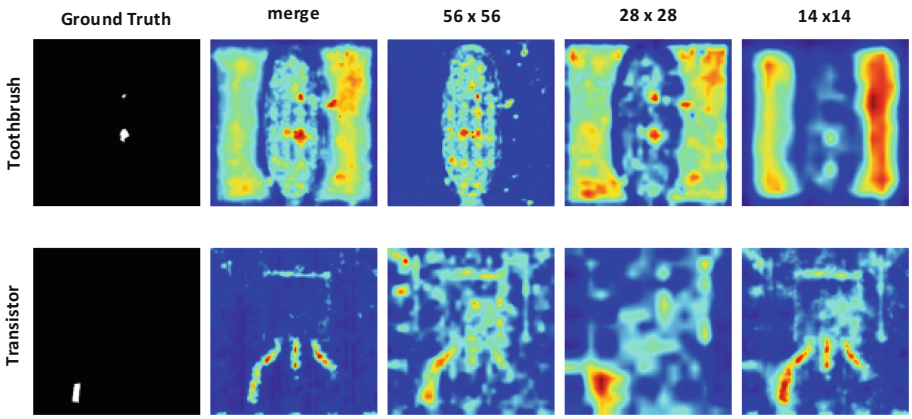

Fig. 4. Samples of bad results.

Detection. The results in Table 1 show that the multi-level feature fusion approach we used in MVTecAD has a significant improvement in detection performance compared to MKD. Especially in the texture class data, we can have good detection results regardless of the class of texture defects. However, in the Objects category, we find that the four categories of Cable, Transistor, toothbrush, and screw prevent us from going further, especially the toothbrush category, in which the detection is even worse than most of the previous methods. Observing the output anomaly maps of sizes 56, 28, and 14, as shown in Fig. 4, we found that the anomaly maps of sizes 28 and 14 judged the background region as anomalous, and after upsampling, this wrong determination was amplified, so that the accuracy of detection was greatly affected when the three were fused.

Localization. The results in Table 2 show that although our method is not optimal compared to other methods, a closer look shows that our method copes well with the various defects in the texture category. However, in the Objects category, our method does not perform well in Capsule and Transistor, especially transistor, which is less accurate than all other categories by a dozen, a very bad effect for the final average. It is easy to see that the MKD method also performs very poorly in the transistor category, so perhaps the problem arises in the VGG network itself. For the Capsule category, by observing its anomaly maps of sizes 56, 28, and 14, we find that the problem still occurs

in the two small-sized anomaly maps of 28 and 14, especially the 14-sized anomaly map, which thinks that almost all the backgrounds are anomalous, and then upsampled to further expand this wrong determination result, causing the final result to become poor.

Table 3. Image-level detection results and pixel-level detection results on DAGM.

	Detection				Localization			
Category	MKD	STFPM	OURS	OURS*	MKD	STFPM	OURS	OURS*
Class1	56.3	97.8	94.9	**98.6**	56.7	**90.9**	88.7	88.1
Class2	90.6	93.8	95.7	**99.9**	97.0	94.4	97.1	**97.4**
Class3	74.6	**90.6**	57.6	75.9	83.8	**88.0**	78.7	83.3
Class4	**100**	**100**	**100**	**100**	95.4	**98.6**	97.5	97.4
Class5	68.8	83.5	95.2	**97.2**	68.9	88.9	93.2	**94.4**
Class6	90.7	**99.8**	67.6	98.3	76.1	**88.9**	75.8	80.4
Class7	49.0	**100**	98.8	**100**	71.4	**94.6**	89.5	90.6
Class8	55.0	**97.9**	82.5	97.3	75.6	**97.1**	94.5	96.7
Class9	66.0	87.5	68.8	**100**	91.1	**97.2**	79.0	94.3
Class10	94.5	**99.0**	95.6	97.8	96.8	**98.4**	98.2	**98.4**
Mean	74.6	95.0	85.7	**96.5**	81.3	**93.7**	89.2	92.1

4.4 DAGM Dataset

For the DAGM dataset, we adopt AUROC as the evaluation index used for detection as well as localization.

As shown in Table3, what can be seen is that MKD performs poorly in datasets containing complex texture class datasets and small defects, and in many of these classes, MKD does not perform well for defect detection. In contrast, our method of fusing multiple anomaly maps at different scales performs well in the datasets of these texture classes. For the default way of assigning 1/3 weight to anomaly maps of sizes 56, 28, and 14 respectively, it improves nearly 11% over MKD in terms of defect detection effect and about 8% in terms of defect localization effect. The other way of assigning 1/2, 1/3 and 1/6 weights to 56, 28 and 14 respectively, has a substantial improvement in the detection effect of defects. The weight assignment method is discussed more in Sec. 5.1. The second way of assigning weights greatly reduces the errors generated in the process of sampling to 224 on the anomaly maps of two small sizes, 28 and 14, resulting in a significant improvement of the final average effect. The table also shows that after the weight adjustment, for class6 and class9, the defect detection effect has a qualitative leap, and the defect location effect has a considerable improvement.

5 Ablation Study

5.1 Fusing Weights of Multi-scale Anomaly Maps

The outputs of our three selected convolutional blocks are 56×56, 28×28, and 14×14, respectively, with default weights of 1/3 (β) for each of the three anomaly maps, along with a set of weights of 1/2, 1/3, and 1/6 (β*), corresponding to the layers where 56, 28, and 14 are located, respectively. Our original intention of designing the second set of weights was to worry that the size of the anomaly map output from the latter two layers was too small. In the upsampling process, because it is filling the non-existent pixel points by interpolation, it is not really detecting the presence of defects, then for these two small sizes, there is definitely an error in the upsampling process. To reduce this error, we penalize the weights of anomaly maps of small size, the smaller the anomaly map the smaller the weights assigned. In Table 4, for DAGM datasets with various complex texture classes, some defects are relatively small and some defects are not very obvious compared to normal data, the second weight assignment method avoids errors in upsampling for small anomaly maps and effectively ensures the accuracy of detection. But for the MVTecAD, many defects are relatively large, small size anomaly map in the process of upsampling by interpolation method of filling the part does not produce much error, and the integration of a variety of size anomaly map more to ensure the accuracy of the detection effect. So we use β for MVTecAD and β* for DAGM.

Table 4. Image-level detection results and pixel-level detection results on DAGM. β is the default weight of 1/3 for each of the three layers, and β* represents the weights of 1/2, 1/3, and 1/6.

	β		β*	
Dataset	Detection	Localization	Detection	Localization
MVtecAD	**96.5**	95.8	90.5	**95.9**
DAGM	85.7	89.2	**96.5**	**92.1**

5.2 Number of Layers During Training

In the exception detection phase we use the output of the first three blocks of the last four blocks instead of using the output of the last block. So we thought about a question: Since we only use the first three blocks, do we need to learn about the last block? The results in Table 5 can show that even if only three blocks are used, in the learning phase, that is, the training phase, the learner network learns the complete four blocks of knowledge of the expert network, which still has some improvement for the final overall defect detection as well as localization effect.

Table 5. Ablation studies for training layers.

	3 layers		4 layers	
Dataset	Detection	Localization	Detection	Localization
MVtecAD	96.2	95.0	**96.5**	**95.8**
DAGM	85.7	89.2	**96.5**	**92.1**

5.3 Individual Layer v.s. Multi-level Layers

The experiments performed in this section are to demonstrate the necessity of our proposed fusion of multiple scale anomaly maps, and it can be clearly seen that the three different sizes of anomaly maps, 56, 28 and 14, have different detection effects for defects in different categories. For small-sized anomaly maps, the detection accuracy may be higher, but they inevitably have errors in the upsampling process, while for large-sized anomaly maps, although the detection effect is not very good, there is not much error in the upsampling process. So taking all these factors into consideration, we fused multiple scales of anomaly maps (Table 6).

Table 6. Performance with different sizes of anomaly maps.

	56	28	14	Multilevel
AUROC	87.3	89.8	88.1	**95.8**

6 Conclusion and Discussion

We show that comprehensive knowledge propagation from a pre-trained expert network to a learner network with the same structure and combining the differences in multiple intermediate layer features of the two networks are effective in detecting defects contained in images, especially for texture-like images. Our approach avoids designing learner networks and does not require the expensive training time cost based on patch.

It is worth noting that we are pre-trained on ImageNet, which may have unexpected effects if self-supervised learning is used, and that some of the network's problems present in VGG could perhaps be improved by adding some modules; nevertheless, we provide a promising direction.

Acknowledgements. This research was partially funded by Xianyang Science and Technology Research PlanProject (2021ZDYF-NY-O014) and Xi'an Science and Technology Plan Project (2022JH-JSYF-O270). All supports and assistance are sincerely appreciated

References

1. Bergmann, P, Fauser, M, Sattlegger, D, et al.: MVTec AD--A comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9592–9600 (2019)
2. Mei, S., Wang, Y., Wen, G.: Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. Sensors **18**(4), 1064 (2018)
3. Bergmann, P., et al.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint. arXiv, 1807.02011 (2018)
4. Collin, A.S, De Vleeschouwer, C.: Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp. 7915–7922 (2021)
5. Dehaene, D, Frigo, O, Combrexelle, S, et al.: Iterative energy-based projection on a normal data manifold for anomaly localization. arXiv preprint. arXiv, 2002.03734 (2020)
6. Salehi, M., Eftekhar, A., Sadjadi N, et al.: Puzzle-ae: novelty detection in images through solving puzzles. arXiv preprint. arXiv, 2008.12959 (2020)
7. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: semi-supervised anomaly detection via adversarial training. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV 2018. LNCS, vol. 11363, pp. 622–637. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20893-6_39
8. Schlegl, T., Seeböck, P., Waldstein, S.M., et al.: f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Med. Image Anal. **54**, 30–44 (2019)
9. Perera, P., Nallapati, R., Xiang, B.: Ocgan: one-class novelty detection using gans with constrained latent representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2898–2906 (2019)
10. Sabokrou, M., Khalooei, M., Fathy, M., et al.: Adversarially learned one-class classifier for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3379–3388 (2018)
11. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint. arXiv 1701.04862 (2017)
12. Salimans, T., Goodfellow, I., Zaremba, W., et al.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
13. Wang, S., Zeng, Y., Liu, X., et al.: Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
14. Fei, Y., Huang, C., Jinkun, C., et al.: Attribute restoration framework for anomaly detection. IEEE Trans. Multimedia **24**, 116–127 (2020)
15. Bergmann, P., Fauser, M., Sattlegger, D., et al.: Uninformed students: student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4183–4192 (2020)
16. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better?. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2661–2671 (2019)
17. Sun, R., Zhu, X., Wu, C., et al.: Not all areas are equal: transfer learning for semantic segmentation via hierarchical region selection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,. pp. 4360–4369 (2019)
18. Salehi, M., Sadjadi, N., Baselizadeh, S., et al.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14902–14912 (2021)

19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv 1409.1556 (2014)
20. Deng, J., Dong, W., Socher, R., et al.: Imagenet: a large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255 (2009)
21. Carrara, F., Amato, G., Brombin, L., et al.: Combining gans and autoencoders for efficient anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp. 3939–3946 (2021)
22. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
23. Zavrtanik, V., Kristan, M., Skočaj, D.: Reconstruction by inpainting for visual anomaly detection. Pattern Recogn. **112**, 107706 (2021)
24. Yi, J., Yoon, S.: Patch svdd: patch-level svdd for anomaly detection and segmentation. In: Proceedings of the Asian Conference on Computer Vision (2020)
25. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learn. **54**(1), 45–66 (2004). https://doi.org/10.1023/B:MACH.0000008084.60811.49
26. Ruff, L., Vandermeulen, R., Goernitz, N., et al.: Deep one-class classification. In: International Conference on Machine Learning. PMLR, pp. 4393–4402 (2018)
27. Shi, Y., Yang, J., Qi, Z.: Unsupervised anomaly segmentation via deep feature reconstruction. Neurocomputing **424**, 9–22 (2021)
28. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint. arXiv 2005.02357 (2020)
29. Wang, G., Han, S., Ding, E., et al.: Student-teacher feature pyramid matching for unsupervised anomaly detection. arXiv preprint. arXiv 2103.04257 (2021)
30. Venkataramanan, S., Peng, K.-C., Singh, R.V., Mahalanobis, A.: Attention guided anomaly localization in images. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12362, pp. 485–503. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58520-4_29
31. Zhang, R., Isola, P., Efros, A.A., et al.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
32. Wieler, M., Hahn, T.: Weakly supervised learning for industrial optical inspection. In: DAGM Symposium In (2007)