Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection

Zhenyu Li[1]
Ning Li[1]
Kaitao Jiang[2]
Zhiheng Ma[2]
Xing Wei[1]
Xiaopeng Hong †[3]
Yihong Gong[1]

[1] School of Software Engineering
[2] College of Artificial Intelligence
[3] School of Cyber Science and Engineering
Xi'an Jiaotong University
Xi'an, China

†Corresponding author

Abstract

Anomaly detection aims at identifying abnormal samples from the normal ones. Existing methods are usually supervised or detect anomalies at the instance level without localization. In this work, we propose an unsupervised method called Superpixel Masking And Inpainting (SMAI) to identify and locate anomalies in images. Specifically, superpixel segmentation is first performed on the images. Then an inpainting module is trained to learn the spatial and texture information of the normal samples through random superpixel masking and restoration. Therefore, the model can reconstruct the superpixel mask with normal content. At the inference stage, we mask the image using superpixels and restore them one by one. By comparing the mask areas of the original image and its reconstruction, we can identify and locate the abnormal regions. We conducted a comprehensive evaluation of SMAI on the latest MVTec anomaly detection dataset, and it shows that SMAI plays favorably against state-of-the-art methods.

1 Introduction

Abnormal detection is a critical problem, especially in industrial manufacturing in recent years. Automatic abnormal detection using machine vision techniques for industrial defect detection plays a more and more important role in modern industry since it could significantly improve production efficiency while reducing the cost. Inspired by the fact that deep learning has achieved remarkable performances on image recognition, several abnormal detection methods based on object detection [5] and semantic segmentation [8, 13] are proposed. For example, Liu et al. [15] use Faster R-CNN [19] for fabric defect detection; YOLO [18] is used for the detection of insulator defects [9] and Wang et al. [23] apply Mask R-CNN [10] to the surface defects detection of paper dish. However, such discriminative models used in object detection or semantic segmentation have the limitation that they require all the types of defects in the training data. Moreover, building a large-scale training dataset with bounding-box or pixel-level annotations is costly and labor-intensive. A few approaches, on the other side, adopt generative models for defect detection. Typically, they use AutoEncoder or Generative Adversarial Network (GAN) to generate an intermediate output (ideally, the defect-free image), which is further processed to classify or localize the defects. For example, Niu et al. [17] employ CycleGAN [6] to learn the mapping between the normal image and defect image domains via cycle consistency. These methods also need a large amount of training data, including normal images and a few abnormal images. In other words, they are weakly supervised approaches.

To address these shortcomings, we propose an unsupervised anomaly detection method called superpixel masking and inpainting (SMAI). The basic idea is that given a set of normal training samples, an image inpainting model can be trained in a self-supervised manner to restore a damaged image into a normal one. Thus we can leverage such a model to detect abnormal regions by comparing the original and restored images. With the development of deep learning, image inpainting methods have been able to restore the spatial and texture information of the image very well. Leveraging high-level semantic feature learning, image inpainting methods can generate semantically-coherent results for the missing regions. The purpose of superpixel segmentation [24] is to divide a pixel-level image into a district-level image. It can segment images based on factors such as borders, content similarity, semantic information, etc. The superpixel algorithm is widely used in computer vision. For example, Yang et al. [28] employ superpixel to object tracking [29]. They use the superpixel method to capture the structural information from the perspective of mid-level vision. Tian et al. [23] use superpixel results as the basic units instead of operating at the pixel level to extract moving objects better.

In this paper, we focus on unsupervised abnormal detection which only needs the normal training data. Specifically, we first perform superpixel segmentation on the dataset and train an image inpainting model to learn how to recover the missing region masked by a random mask on the superpixel results through a deep convolutional network. When the model has the ability to encode the spatial context information of normal data, it can fill in any damaged region with normal image content. During the inference procedure, we traverse the entire image on the superpixel result. At each location, we mask and reconstruct the image content using the well-trained inpainting model. By calculating the *SSIM* (structure similarity [26]) or L_2 value of the original image area and its corresponding reconstruction region, we can easily obtain the final abnormal map. The main contribution of this paper is we proposed an unsupervised deep method SMAI for abnormal detection. Our approach can not only give an accuracy classification but also locate the detail defect region accurately.

2 Related Work

In this section, we review the supervised and unsupervised methods which have been proposed for anomaly detection.

2.1 Supervised Methods

Supervised methods for anomaly detection usually use object detection or semantic segmentation algorithms. These methods use a large amount of labeled data to train the model. For example, Faster R-CNN and YOLO are used for fabric defect detection[15] and the detection of insulator defects [9]. [25] applies Mask R-CNN to the surface defects detection of the paper dish. They mark different anomalous areas as different categories to be detected and then use the supervised method to detect them.

This type of method requires a lot of well-labeled data, and it can only detect defects that have occurred in the training data, which means that it has poor generalization ability and requires a lot of labor costs.

2.2 Unsupervised Methods

Unsupervised methods require only normal samples during training. These methods are usually based on reconstruction, such as AutoEncoder or Generative Adversarial Network (GAN). They reconstruct an abnormal image to obtain a normal image and perform anomaly detection by comparing before and after reconstruction. Such methods can capture any type of abnormal area.

Generative adversarial network (GAN) consists of two neural networks, one is called a generator and the other is called a discriminator. The generator attempts to capture the data distribution, and the discriminator estimates the probability that the samples come from training data rather than a generator. During training, the generator tries to fool the discriminator better and the discriminator tries to catch fakes generated by the generator, so the training process is called adversarial training. Therefore, GAN can be used to obtain the distribution of the normal samples. For example, in [21], the GAN called AnoGAN is used to learn the manifold of normal anatomical variability and perform anomaly detection based on these. [22] greatly speeds up anomaly detection by introducing encoder into AnoGAN. The GAN called MAD-GAN [14] is used to do the multivariate anomaly detection for time series data. In [7], ADGAN is proposed for anomaly detection by searching for a good representation of the normal samples.

AutoEncoder is also a network that used for reconstructing. It reconstructs the input image through the encoder and decoder. The AutoEncoder can be trained on the normal samples. Then anomaly detection can be performed by comparing before and after reconstruction of abnormal samples. For instance, [2] use the reconstruction probability from the variational AutoEncoder [12] to perform anomaly detection. Sakurada and Yairi [20] use AutoEncoder with nonlinear dimensionality reduction in the anomaly detection task.

Based on the unsupervised learning paradigm, we only need normal samples during training, which greatly reduces the labor costs. At the same time, based on these methods, we can detect any kind of anomalies without being limited to the training set.

3 The Proposed Method

We propose the SMAI to perform anomaly detection and localization, and only normal images are required during training. We will introduce the three parts of SMAI. They are the image inpainting module, the training module, and the test module.

3.1 Image Inpainting Module

We employ the image inpainting method called PEN-Net [30]. This algorithm can restore the input masked image to the normal image. The algorithm uses the U-Net network as the backbone structure and uses a pyramid context to improve the effectiveness of encoding. It applies the rich semantic information of the high-level features to guide the inpainting of the low-level features layer by layer through the attention mechanism. The reconstructed image is semantically reasonable and the restoration content has clear and rich texture details.

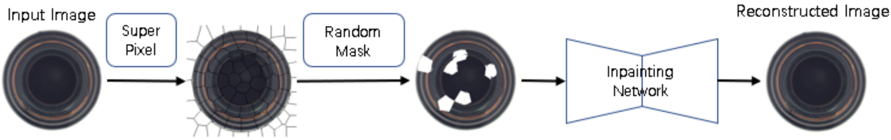


Figure 1: The training flowchart of SMAI. We randomly mask the superpixel results and then reconstruct it to train the image inpainting module.

3.2 Training Module

We assume that the distribution of the anomaly area is inconsistent with other areas in the image. If we mask the normal area, the restoration content should be highly similar to the corresponding area in the input image. If the abnormal area is masked, on the other hand, the restoration content should be generated based on the normal sample distribution and has a low similarity with the corresponding area.

First, we use the superpixel methods [1, 27] to divide the image into multiple superpixel blocks and randomly mask these superpixel blocks on normal images when training. Suppose the normal image set is $X = \{x_1, x_2, \cdots, x_n\}$. Divide the image into m superpixel blocks and mask k superpixel blocks randomly at a time. The masked image set is recorded as X_{mask} and $X_{\text{mask}} = \{x^1_{\text{mask}}, x^2_{\text{mask}}, \ldots, x^n_{\text{mask}}\}$. We input the masked image set X_{mask} into the Image Inpainting Module. The purpose of the inpainting module is to restore the masked area and generate an image that is similar to the normal image X. That is, make the reconstruction of the masked image approach to X. We use the structural similarity index for measurement. The training objective $\mathcal{J}$ can be described as Equation 1.

$$\mathcal{J} = \min \left(SSIM \left(f_{ip} \left(X_{\text{mask}} \right), X \right) \right), \tag{1}$$

where $f_{ip}(\cdot)$ denotes the image inpainting module, X and X_{mask} represent the original and masked image set, respectively. During the training process, the ability of the inpainting module that restores the masked area to a normal area is trained. The training process is shown in Figure 1.

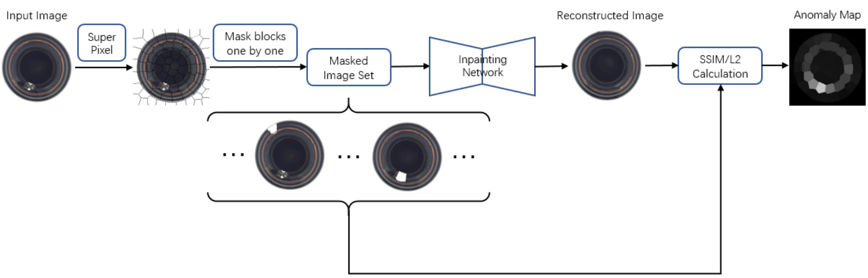
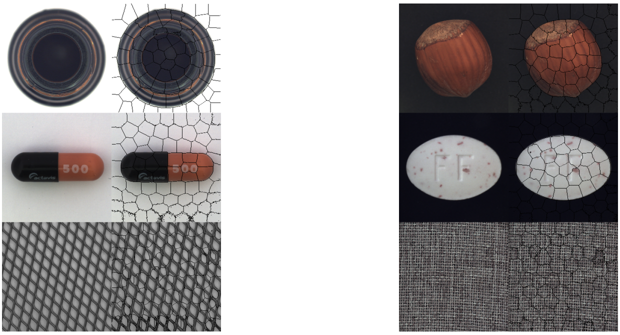


Figure 2: The testing flowchart of SMAI. We mask the superpixel blocks one by one on the superpixel results and reconstruct them, then obtain the anomaly map according to the *SSIM* or L_2 value before and after reconstruction of the mask area.

3.3 Test Module

During the testing, the superpixel algorithm is first applied to the images as in training. Suppose the test image is X. Divide the image into m superpixel blocks. Then we mask the superpixel blocks one by one. The mask result of each block will generate a masked image, so each test image will generate m masked images. Record it as $X_{\text{mask}} = \left\{x_{\text{mask}}^1, x_{\text{mask}}^2, \ldots, x_{\text{mask}}^m\right\}$. It should be noted that this X_{mask} is different from it in the training module. A masked image set is generated by only one test image.



(a) Bottle, Capsule and Grid (b) Hazelnut, Pill and Carpet

Figure 3: Examples for superpixel segmentation results, where the left is the original image and the right is superpixels.

We input the masked image set X_{mask} into the inpainting module and get the reconstructed output image set. Record it as $X_{\text{Inpainting}} = \left\{x_{\text{Inpainting}}^1, x_{\text{Inpainting}}^2, \ldots, x_{\text{Inpainting}}^m\right\}$. The $SSIM$ or L_2 calculation is performed on the corresponding mask block between the test image X and each restoration image in the $x_{\text{Inpainting}}$. When using $SSIM$, the $SSIM$ value shows the similarity between before and after reconstruction of the image in this mask block. On the contrary, $1 - SSIM$ shows the difference. The $1 - SSIM$ is accumulated by mask blocks and an anomaly map is generated based on the similarity value. Brightness shows the degree of difference. When using L_2, we directly use the L_2 value to produce abnormal graphs. Therefore the highlighted area is the defective is. The calculation method is shown in Equation 2 and 3.

$$\mathrm{M} = \sum_{i=1}^{m} \left(1 - SSIM_{\text{mask}}\left(x_{\text{Inpainting}}^i, X\right)\right), \tag{2}$$

$$\mathrm{M} = \sum_{i=1}^{m} \left(L2_{\text{mask}}\left(x_{\text{Inpainting}}^i, X\right)\right), \tag{3}$$

where M, $SSIM_{\text{mask}}$, $L2_{\text{mask}}$, m, $x_{\text{Inpainting}}^i$, X indicates the defected anomaly map, the $SSIM$ calculation on the mask area corresponding to the masked image x_{mask}^i, the $L2$ calculation on the mask area corresponding to the masked image x_{mask}^i, the number of divided superpixel blocks, the output of image inpainting module and the original image. Figure 2 shows the specific process of the test module, and Figure 3 shows some superpixel samples.

Category	**Our SMAI** *SSIM*	**Our SMAI** l_2	AE *SSIM*	AE l_2	AnoGAN	CNN-Feature Dictionary
Carpet	0.57 0.72	0.93 0.4	0.43 0.90	0.57 0.42	0.82 0.16	0.89 0.36
Grid	0.91 0.93	0.81 0.98	0.38 1.00	0.57 0.98	0.90 0.12	0.57 0.33
Leather	0.00 1.00	0.82 0.67	0.00 0.92	0.06 0.82	0.91 0.12	0.63 0.71
Wood	0.10 1.00	0.79 0.95	0.84 0.82	1.00 0.47	0.89 0.47	0.79 0.88
Bottle	1.00 0.91	1 0.71	0.85 0.90	0.70 0.89	0.95 0.43	1.00 0.06
Cable	0.72 0.55	0.74 0.51	0.74 0.48	0.93 0.18	0.98 0.07	0.97 0.24
Capsule	0.44 0.70	0.57 0.73	0.78 0.43	1.00 0.24	0.96 0.20	0.78 0.03
Hazelnut	1.00 0.53	0.93 0.74	1.00 0.07	0.93 0.84	0.83 0.16	0.90 0.07
Metal Nut	0.41 0.70	0.96 0.28	1.00 0.08	0.68 0.77	0.86 0.13	0.55 0.74
Pill	0.96 0.34	0.85 0.46	0.92 0.28	1.00 0.23	1.00 0.24	0.85 0.06
Screw	0.73 0.56	0.76 0.91	0.95 0.06	0.98 0.39	0.41 0.28	0.73 0.13
Tile	0.00 1.00	0.94 0.43	1.00 0.04	1.00 0.54	0.97 0.05	0.97 0.44
Toothbrush	0.92 0.67	0.83 0.93	0.75 0.73	1.00 0.97	1.00 0.13	1.00 0.03
Transistor	0.88 0.73	0.77 0.53	1.00 0.03	0.97 0.45	0.98 0.35	1.00 0.15
Zipper	0.75 0.80	0.84 0.97	1.00 0.60	0.97 0.63	0.78 0.40	0.78 0.29
Mean	0.63 **0.74**	0.84 0.68	0.78 0.49	0.82 0.59	**0.88** 0.22	0.83 0.30
Mean Acc	0.685	**0.76**	0.635	0.705	0.55	0.565

Table 1: Results of the evaluated methods when applied to the classification of anomalous images. For each dataset category, the ratio of correctly classified samples of anomaly-free (top row) and anomalous images (bottom row) is given. The last row(Mean Acc) is the average between anomalous and anomaly-free accuracies.

4 Experimental Results

To demonstrate the effectiveness of our approach, an extensive evaluation of the specific abnormal detection datasets MVTec [4] is performed. We measure the performance of our unsupervised abnormal detection framework against existing pipelines. Details of experimental settings are introduced in Section 4.1. The experiment results and the analysis of the effectiveness of our model are described in Section 4.2.

4.1 Experimental Settings

We experimented on the MVTec Abnormal Detection dataset, using four Titan Xp for training and one for testing. We have compared with several baseline methods on multiple indicators, including the ratio of correctly classified samples of anomaly-free and anomalous images and the relative per-region overlap, which is the same evaluation indicators as in [4].

Category	**Our SMAI** *SSIM*	**Our SMAI** l_2	AE *SSIM*	AE l_2	AnoGAN	CNN-Feature Dictionary
Carpet	0.28 0.87	0.27 0.88	0.69 0.87	0.38 0.59	0.34 0.54	0.20 0.72
Grid	0.60 0.96	0.84 0.97	0.88 0.94	0.83 0.90	0.04 0.58	0.02 0.59
Leather	0.98 0.51	0.44 0.86	0.71 0.78	0.67 0.75	0.34 0.64	0.74 0.87
Wood	0.34 0.62	0.54 0.80	0.36 0.73	0.29 0.73	0.14 0.62	0.47 0.91
Bottle	0.48 0.91	0.26 0.86	0.15 0.93	0.22 0.86	0.05 0.86	0.07 0.78
Cable	0.09 0.82	0.04 0.92	0.01 0.82	0.05 0.86	0.01 0.78	0.13 0.79
Capsule	0.20 0.81	0.39 0.93	0.09 0.94	0.11 0.88	0.04 0.84	0.00 0.84
Hazelnut	0.29 0.96	0.52 0.97	0.00 0.97	0.41 0.95	0.02 0.87	0.00 0.72
Metal Nut	0.18 0.90	0.07 0.92	0.01 0.89	0.26 0.86	0.00 0.76	0.13 0.82
Pill	0.13 0.93	0.24 0.92	0.07 0.91	0.25 0.85	0.17 0.87	0.00 0.68
Screw	0.25 0.94	0.57 0.96	0.03 0.96	0.34 0.96	0.01 0.80	0.00 0.87
Tile	0.98 0.60	0.14 0.62	0.04 0.59	0.23 0.51	0.08 0.50	0.14 0.93
Toothbrush	0.39 0.96	0.61 0.96	0.08 0.92	0.51 0.93	0.07 0.90	0.00 0.77
Transistor	0.20 0.82	0.06 0.85	0.01 0.90	0.22 0.86	0.08 0.80	0.03 0.66
Zipper	0.17 0.74	0.46 0.9	0.10 0.88	0.13 0.77	0.01 0.78	0.00 0.76
Mean	**0.37** 0.82	0.36 **0.89**	0.22 0.87	0.32 0.81	0.09 0.74	0.13 0.78

Table 2: Results of the evaluated methods when applied to the segmentation of anomalous regions. For each dataset category, the relative per-region overlap (top row) and the ROC AUC (bottom row) are given. The best performing method is highlighted in boldface.

4.1.1 Datasets

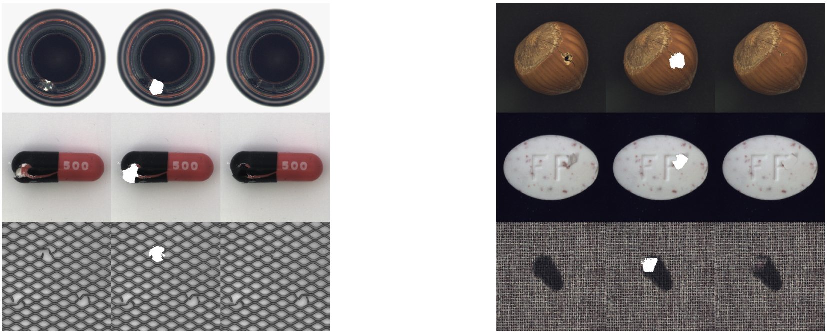
The MVTec Abnormal Detection dataset comprises 15 categories with 3629 images for training and validation and 1725 images for testing. The training set contains only normal images without defects. The test set contains images containing various kinds of defects and defect-free images. Five categories cover different types of regular (carpet, grid) or random (leather, tile, wood) textures, while the remaining ten categories represent various types of objects. In total, 73 different defect types are present, on average five per category. All image resolutions are in the range between 700 $\times$ 700 and 1024 $\times$ 1024 pixels. Pixel-precise ground truth labels for each defective image region is provided. In total, the dataset contains almost 1900 manually annotated regions.

4.1.2 Comparison with Other Methods

We compare with the following baselines for their performance.

- l_2 AutoEncoder which uses the CAE (Convolutional AutoEncoders) [11] architecture to reconstruct defect-free training samples through a bottleneck (latent space). Anomalies are detected by a per-pixel l_2 loss of the input with its reconstruction.

- *SSIM* (structural similarity) AutoEncoder [3] similar to l_2 AutoEncoder employing a



(a) Bottle, Capsule and Grid (b) Hazelnut, Pill and Carpet

Figure 4: Examples for reconstruction results, where the left is the test image, the middle is the masked image, and the right is the restoration image.

loss based on the structural similarity ($SSIM$).

- AnoGAN [21], a generative model, which can obtain anomaly maps by a per-pixel l_2-comparison of the input image with the generated output.
- CNN-Feature Dictionary [16], which perform Principal Component Analysis (PCA) on extracted features from the 512-dimensional avgpool layer of a ResNet-18 pre-trained on ImageNet.

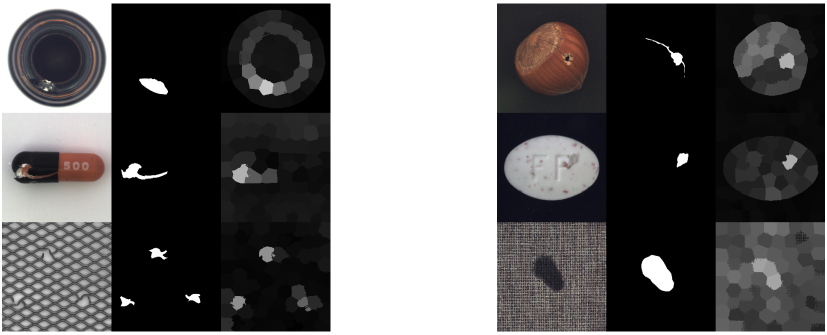
4.1.3 Implementation Details

During training, we resize the image to 256×256 and then divide it into 77 superpixel blocks. Masking 10 blocks each iteration randomly. We train the inpainting module (PEN-NET) with a learning rate of 1e-4 with a batch size of 8 for 50000 iterations. During the test, we resize the image to 256×256 as in training and mask the superpixel blocks one by one, then calculate the $SSIM$ or L_2 values before and after the image inpainting module on the masked area. Then we use the $1 - SSIM$ or L_2 values to generate the anomaly map.

4.2 Results and Analyses

We make a comprehensive comparison with several baseline methods mentioned in Sec. 4.1.2 and visualize the experimental results. Evaluation results for the classification of anomalous images and segmentation of anomalous regions are given for SMAI and dataset categories in Tables 1 and 2, respectively. Figure 4 shows some reconstruction results. We mask some anomaly areas in the abnormal image and then restore them. By comparing the content before and after the reconstruction of this area, we can obtain the anomaly map. In Figure 5 we show some test results. Among them, the right is the anomaly map and the bright areas are the defected anomaly regions.

It can be seen from Table 1 that SMAI($SSIM$) reduces the ratio of correctly classified samples of anomaly-free images but greatly improves it of anomalous images. SMAI(L_2) exceeds the baseline methods in the ratio of correctly classified samples of both anomaly-free and anomalous images. Table 2 shows that SMAI($SSIM$) greatly improves the positioning



(a) Bottle, Capsule and Grid (b) Hazelnut, Pill and Carpet

Figure 5: Examples for test results, where the left is the test image, the middle is the ground truth and the right is our result. The brighter the difference.

accuracy of abnormal regions. SMAI(L_2) also surpasses the baseline methods in both overlap and AUC score.

Specifically, in Table 1, we improve the ratio of correctly classified samples of anomaly-free and anomalous images, which means that SMAI is better than the previous methods at the instance level on anomaly detection. In Table 2, SMAI achieves the highest relative per-region overlap and AUC score compared with previous methods, which means that SMAI has made great progress in pixel-level anomaly localization. SMAI (*SSIM*) achieved the best performance on Abnormal Acc and overlap while SMAI(L_2) performed best on AUC and Mean Acc. Figure 4 shows that when we mask the abnormal area, the content before and after restoration is very different, which means it is easy to judge whether it is an anomaly area. In Figure 5, the right image is the anomaly detection map, and the brightest area represents the anomaly area. It can be seen that SMAI can accurately detect abnormal regions. To be summarized, SMAI can not only identify abnormal images more accurately but also locate abnormalities more exactly.

5 Conclusion

We propose the superpixel masking and inpainting method for self-supervised anomaly detection. SMAI greatly improving the ratio of correctly classified samples of anomaly-free and anomalous images, the accuracy of positioning defective areas and AUC score. We comprehensively compared SMAI with several baseline methods on the MVTec Abnormal Detection dataset. Overall, SMAI greatly surpasses the previous methods and has a huge advantage in anomaly location.

Acknowledgments

This work is sponsored by National Key R&D Program under Grant No.2019YFB1312000, National Major Project under Grant No.2017YFC0803905 and SHAANXI Province Joint Key Laboratory of Machine Learning.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 4

[2] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 2015. 3

[3] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018. 7

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6

[5] Junwen Chen, Zhigang Liu, Hongrui Wang, Alfredo Nunez, and Zhiwei Han. Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. *IEEE Transactions on Instrumentation Measurement*, 67(2): 257–269. 1

[6] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 2

[7] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–17. Springer, 2018. 3

[8] Max K Ferguson, AK Ronay, Yung-Tsun Tina Lee, and Kincho H Law. Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart and sustainable manufacturing systems*, 2, 2018. 1

[9] Fei Guo, Kun Hao, Mengqi Xia, Lu Zhao, Li Wang, and Qi Liu. Detection of insulator defects based on yolo v3. In *International Conference on Artificial Intelligence for Communications and Networks*, pages 291–299. Springer, 2019. 1, 2

[10] Kaiming He, Gkioxari Georgia, Dollar Piotr, and Girshick Ross. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 1

[11] Y. Bengio I. Goodfellow and A. Courville. In *Deep Learning*, Cambridge, MA, 2016. MIT Press. 7

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[13] Zhang Lei, Yang Fan, Daniel Zhang, and Julie Zhu Ying. Road crack detection using deep convolutional neural network. In *IEEE International Conference on Image Processing (ICIP 2016)*, 2016. 1

[14] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 703–716. Springer, 2019. 3

[15] Xianghui Liu, Zhoufeng Liu, Chunlei Li, Bicao Li, and Baorui Wang. Fabric defect detection based on faster r-cnn. In *Ninth International Conference on Graphic and Image Processing*, 2018. 1, 2

[16] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, 18(1):209, 2018. 8

[17] Shuanlong Niu, Hui Lin, Tongzhi Niu, Bin Li, and Xinggang Wang. Defectgan: Weakly-supervised defect detection using generative adversarial network. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 127–132. IEEE, 2019. 2

[18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[20] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with non-linear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014. 3

[21] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017. 3, 8

[22] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 2019. 3

[23] Zhiqiang Tian, Nanning Zheng, Jianru Xue, Xuguang Lan, Ce Li, and Gang Zhou. Video object segmentation with shape cue based on spatiotemporal superpixel neighbourhood. *IET Computer Vision*, 8(1):16–25, 2014. 2

[24] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *2011 International Conference on Computer Vision*, pages 1323–1330. IEEE, 2011. 2

[25] Xuelong Wang, Ying Gao II, Junyu Dong, Xukun Qin, Lin Qi, Hui Ma, and Jun Liu. Surface defects detection of paper dish based on mask r-cnn. In *Third International Workshop on Pattern Recognition*, volume 10828, page 108280S. International Society for Optics and Photonics, 2018. 1, 2

[26] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[27] Xing Wei, Qingxiong Yang, Yihong Gong, Narendra Ahuja, and Ming-Hsuan Yang. Superpixel hierarchy. *IEEE Transactions on Image Processing*, 27(10):4838–4849, 2018. 4

[28] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. Robust superpixel tracking. *IEEE Transactions on Image Processing*, 23(4):1639–1651, 2014. 2

[29] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006. 2

[30] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1486–1494, 2019. 3