Glancing at the Patch: Anomaly Localization with Global and Local Feature Comparison Supplementary Material

Shenzhi Wang[1] Liwei Wu[1] Lei Cui[1] Yujun Shen[2]
[1] SenseTime Research [2] The Chinese University of Hong Kong
{wangshenzhi, wuliwei, cuilei}@sensetime.com sy116@ie.cuhk.edu.hk

A. Overview

This supplementary material is organized as follows:

- We describe the architecture of Local-Net, Global-Net and DAD-head in Sec. B.
- We introduce the implementation details on MVTec AD [3] and CIFAR-10 [11] dataset in Sec. C.
- We provide more experiment results on MVTec AD dataset [3] in Sec. D, including results in image-level anomaly detection task (Sec. D.1) and more visualizations in pixel-level anomaly localization task (Sec. D.2).

B. Model Architecture

In this part, we provide the architecture of the models, *i.e.,* Local-Net, Global-Net and DAD-head.

Local-Net. Local-Net accepts image patches, whose shapes are $33 \times 33 \times 3$, and outputs local features, whose shapes are $1 \times 1 \times 128$. Tab. 1 illustrates the architecture of the Local-Net, where the negative slopes in all leaky ReLU layers are 5×10^{-3}.

Table 1. **Architecture of Local-Net.**

Layer	Output Size	Parameters		
		Kernel	Stride	Padding
Input	$33 \times 33 \times 3$			
Conv1	$31 \times 31 \times 128$	3×3	1	0
Leaky ReLU	$31 \times 31 \times 128$			
MaxPool	$15 \times 15 \times 128$	2×2	2	0
Conv2	$11 \times 11 \times 256$	5×5	1	0
Leaky ReLU	$11 \times 11 \times 256$			
MaxPool	$5 \times 5 \times 256$	2×2	2	0
Conv3	$4 \times 4 \times 256$	2×2	1	0
Leaky ReLU	$4 \times 4 \times 256$			
Conv4	$1 \times 1 \times 128$	4×4	1	0
Leaky ReLU	$1 \times 1 \times 128$			

Global-Net. Global-Net is fed with images and binary masks, both of which are with shape $256 \times 256 \times 3$, and outputs global features in shape of $1 \times 1 \times 128$. The detailed architecture of the Global-Net is shown in Tab. 2, where PartialConv stands for the partial convolution [12].

Table 2. **Architecture of Global-Net.**

Layer	Output Size	Parameters		
		Kernel	Stride	Padding
Input	$256 \times 256 \times 3$			
PartialConv0	$125 \times 125 \times 16$	7×7	2	0
ReLU	$125 \times 125 \times 16$			
MaxPool	$62 \times 62 \times 16$	3×3	2	0
PartialConv1_1	$30 \times 30 \times 32$	3×3	2	0
ReLU	$30 \times 30 \times 32$			
PartialConv1_2	$28 \times 28 \times 32$	3×3	1	0
ReLU	$28 \times 28 \times 32$			
PartialConv2_1	$13 \times 13 \times 64$	3×3	2	0
ReLU	$13 \times 13 \times 64$			
PartialConv2_2	$11 \times 11 \times 64$	3×3	1	0
ReLU	$11 \times 11 \times 64$			
PartialConv3_1	$5 \times 5 \times 128$	3×3	2	0
ReLU	$5 \times 5 \times 128$			
PartialConv3_2	$3 \times 3 \times 128$	3×3	1	0
ReLU	$3 \times 3 \times 128$			
PartialConv4	$1 \times 1 \times 128$	3×3	1	0

DAD-head. Given a global and local feature, the Distortion Anomaly Detection (DAD) head predicts the probability of whether they match or not. Both features are $128d$ vectors and they are concatenated as a $256d$ vector before fed into the DAD-head. The architecture of the DAD-head is shown in Tab. 3, where the negative slope in all leaky ReLU layers are 1×10^{-2}.

Table 3. **Architecture of DAD-head.**

Layer	Input	FC0	Leaky ReLU	FC1	Leaky ReLU	Softmax
Output Size	256	128	128	2	2	2

C. Implementation Details

In this section, we describe the implementation details on MVTec AD [3] (Sec. C.1) and CIFAR-10 [11] (Sec. C.2) datasets.

C.1. Implementation Details on MVTec AD

Our approach consists of two steps, *i.e.*, pre-training Local-Net, and training Global-Net and DAD-head with the Local-Net fixed. Images are resized to 256×256 resolution and the cropped patches are with 33×33 pixels.

Pre-training Local-Net. We distill Local-Net from pre-trained ResNet-18 [8][1]. The distillation is first performed on ImageNet [5] for $50K$ iterations with batch size 64. Then we fine-tune the Local-Net on some certain category of MVTec AD [3] for another $50K$ iterations with batch size 16. In both processes, we set loss weights as $\lambda_k = \lambda_c = 1.0$. Adam optimizer [10] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used and the learning rate is set as 2.0×10^{-4}.

Training Global-Net and DAD-head. After pre-training, the parameters of Local-Net are fixed during the training of Global-Net and DAD-head. We randomly crop patch $\mathbf{p}$ from the image, and add some random stains on the patch to produce $\mathbf{p}^-$. The loss weight λ_t is set as 0.01. Both Global-Net and DAD-head employ an Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and they are updated with learning rates 10^{-4} and 10^{-5} respectively. The training is performed for $200K$ iterations with batch size 64.

Inference. At the inference stage, for every image, patches are cropped one after another in a roster-scan order, with 20 patches on each side (*i.e.*, totally 400 patches for an image), where patches are uniformly distributed and overlap is allowed between two adjacent patches. The anomaly score is estimated with $\lambda_s = 0.8$. After calculating anomaly scores for all patches in an image, we fuse these scores into a score map using the inverse distance weighted (IDW) interpolation [2] with $p = 5$.

C.2. Implementation Details on CIFAR-10

In this part, we introduce the implementation details of the one-class classification experiment on CIFAR-10 [11].

Pre-training Local-Net. The distillation on ImageNet [5] is the same as that in Sec. C.1. During the fine-tuning on each category of CIFAR-10, for every image $\mathbf{I}$, we resize $\mathbf{I}$ into the patch size (*i.e.*, 33×33) as $\mathbf{I_L}$, which functions as the image patch in Sec. C.1. Other settings remain the same as in Sec. C.1, except that the learning rate is set as 10^{-5}.

Training Global-Net and DAD-head. During training Global-Net and DAD-head, for every image $\mathbf{I}$, we resize $\mathbf{I}$ into the patch size (*i.e.*, 33×33) and the image size (*i.e.*, 256×256), denoted as $\mathbf{I_L}$ and $\mathbf{I_G}$ respectively. $\mathbf{I_L}$ and $\mathbf{I_G}$ can be perceived as the patch and image in Sec. C.1. Here, we use a random different image resized into the patch size as the negative patch $\mathbf{p}^-$. Other settings remain the same as in Sec. C.1, except that the learning rates for Global-Net and DAD-head are both 10^{-5} and the number of training iterations is $300K$.

[1]We use the ResNet-18 checkpoint in https://download.pytorch.org/models/resnet18-5c106cde.pth.

D. More Experiment Results

We further provide more experiment details, including image-level anomaly detection results (Sec. D.1) and more anomaly localization visualizations (Sec. D.2) on MVTec AD dataset [3].

D.1. Image-level Anomaly Detection on MVTec AD

Despite anomaly localization task, we also test our method's anomaly classification capability on the image-level anomaly detection task. Specifically, for each category in MVTec AD, we train a model to separate abnormal images from anomaly-free images.

Setup. We use the same training and inference pipelines and hyper-parameters as those in Sec. C.1 yet the performance is evaluated from the image-level instead of the pixel-level. Here, for each image, we assume the maximum value of anomaly score map as anomaly score.

Baselines. The baselines include GeoTrans [6], GANomaly [1], AE [7], ARNet [9] and AESc+Stain [4]. The results of GeoTrans, GANomaly and AE are borrowed from [9], and the results of AESc+Stain and ARNet are reported by their original papers.

Quantitative Results in Image-level AUROC. The image-level AUROC results are shown in Tab. 4. Our method considerably exceeds the existing alternatives ($\sim 2\%$).

D.2. More Quantitative Results on MVTec AD

We provide more qualitative anomaly localization results by using our algorithm on all the categories in MVTec AD dataset [3]. The categories include: carpet (Fig. 1), grid (Fig. 2), leather (Fig. 3), tile (Fig. 4), wood (Fig. 5), bottle (Fig. 6), cable (Fig. 7), capsule (Fig. 8), hazelnut (Fig. 9), metal nut (Fig. 10), pill (Fig. 11), screw (Fig. 12), toothbrush (Fig. 13), transistor (Fig. 14), and zipper (Fig. 15). We observe that our approach performs steadily in all these categories consisting of various anomaly types, demonstrating its generalization ability and robustness.

In particular, when dealing with some anomalies which seem ordinary in each single patch, such as cable swap (the second example in Fig. 7), faulty imprint (the last example in Fig. 8), bent lead (the first example in Fig. 14), cut lead (the second example in Fig. 14), and misplacement (the fourth and fifth example in Fig. 14), our algorithm adequately locates the abnormal areas, benefiting from the

Table 4. Comparison results among different anomaly detection methods in the **image-level anomaly detection task on MVTec AD dataset** [3]. Competitors include GeoTrans [6], GANomaly [1], AE [7], ARNet [9] and AESc+Strain [4]. The results of GeoTrans and GANomaly are borrowed from [9], and the results of ARNet and AESc+Stain are originally reported in their papers. **Image-level AUROC** is used as the evaluation metric.

Category	GeoTrans	GANomaly	AE	ARNet	AESc+Stain	Ours
Carpet	0.44	0.70	0.64	0.71	0.89	**0.92**
Grid	0.62	0.71	0.83	0.88	**0.91**	0.67
Leather	0.84	0.84	0.80	0.86	**0.89**	0.83
Tile	0.42	0.80	0.74	0.74	**0.99**	0.97
Wood	0.61	0.83	0.97	0.92	0.95	**1.00**
Bottle	0.74	0.89	0.65	0.94	0.98	**0.99**
Cable	0.78	0.76	0.64	0.83	0.89	**0.98**
Capsule	0.67	0.73	0.62	0.68	0.74	**0.79**
Hazelnut	0.36	0.79	0.73	0.86	0.94	**0.99**
Metal Nut	0.81	0.70	0.64	0.67	0.73	**0.85**
Pill	0.63	0.74	0.77	0.79	**0.84**	0.82
Screw	0.50	0.75	**1.00**	**1.00**	0.74	0.87
Toothbrush	0.97	0.65	0.77	**1.00**	**1.00**	0.92
Transistor	0.87	0.79	0.65	0.84	0.91	**0.97**
Zipper	0.82	0.75	0.87	0.88	0.94	**1.00**
Mean	0.67	0.76	0.75	0.84	0.89	**0.91**

comparison between the local and the global information. Furthermore, it is noticeable that, in some categories where the objects might rotate, *e.g.*, hazelnut (Fig. 9) and screw (Fig. 12), our method is still able to detect the abnormal areas accurately, suggesting that our algorithm remains robust even in rotating situations.

References

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conf. Comput. Vis.*, 2018. 2, 3

[2] Patrick M Bartier and C Peter Keller. Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (idw). *Computers & Geosciences*, 1996. 2

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 2, 3

[4] Anne-Sophie Collin and Christophe De Vleeschouwer. Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise. *arXiv preprint arXiv:2008.12977*, 2020. 2, 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 2

[6] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Adv. Neural Inform. Process. Syst.*, 2018. 2, 3

[7] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2006. 2, 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[9] Chaoqin Huang, Fei Ye, Jinkun Cao, Maosen Li, Ya Zhang, and Cewu Lu. Attribute restoration framework for anomaly detection. *arXiv preprint arXiv:1911.10676*, 2019. 2, 3

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 2

[12] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Eur. Conf. Comput. Vis.*, 2018. 1

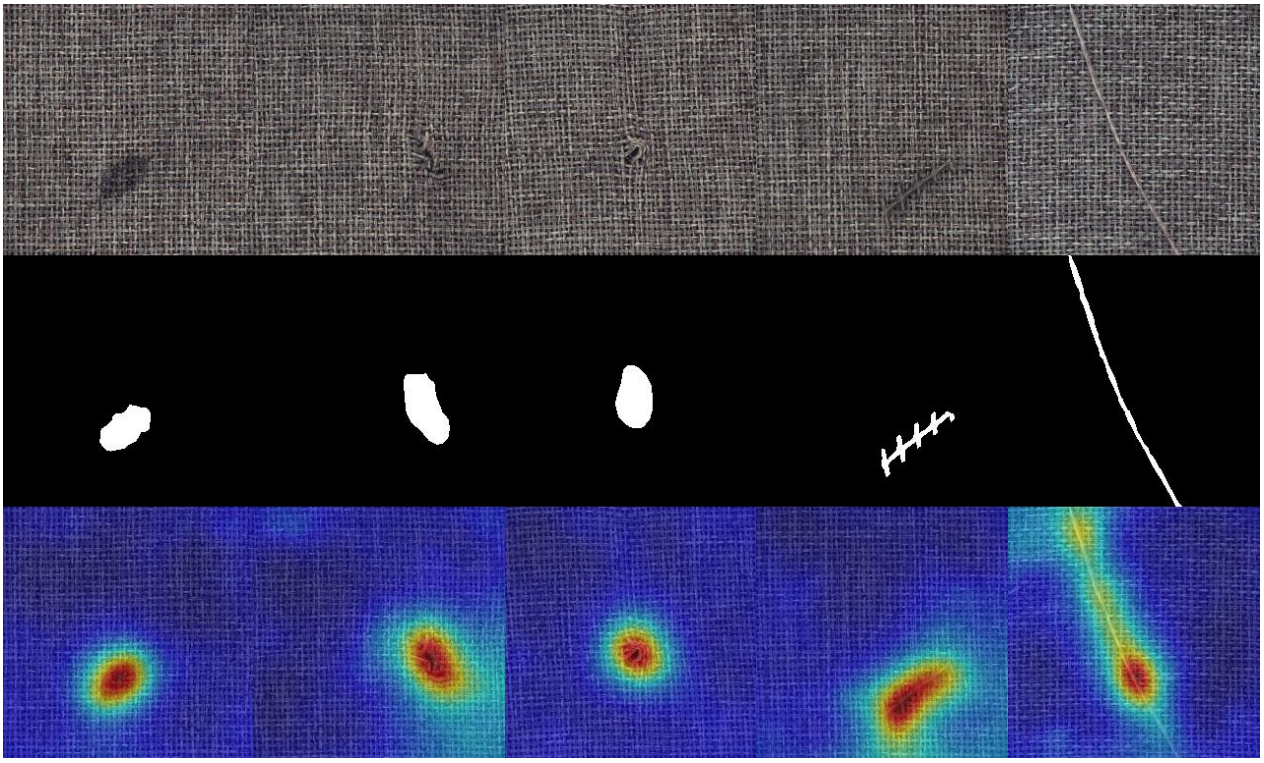

Figure 1. Carpet anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

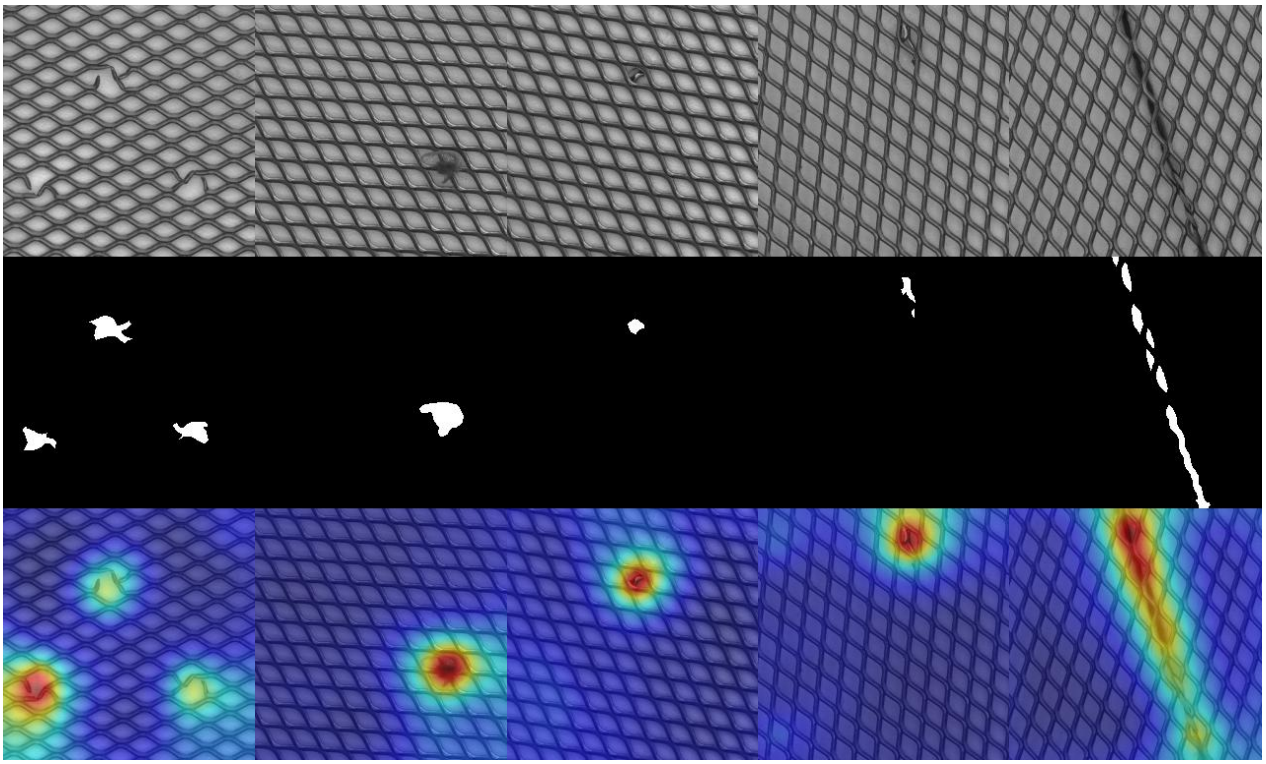

Figure 2. Grid anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

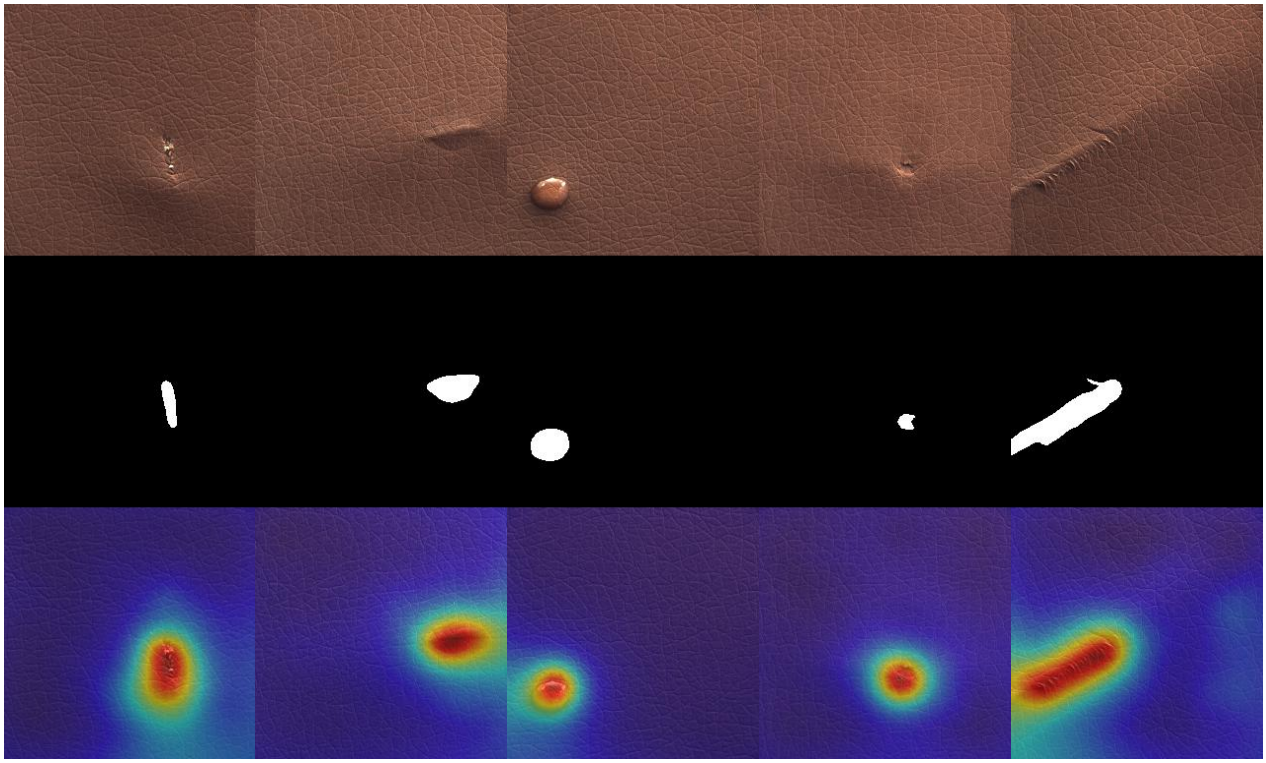

Figure 3. Leather anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

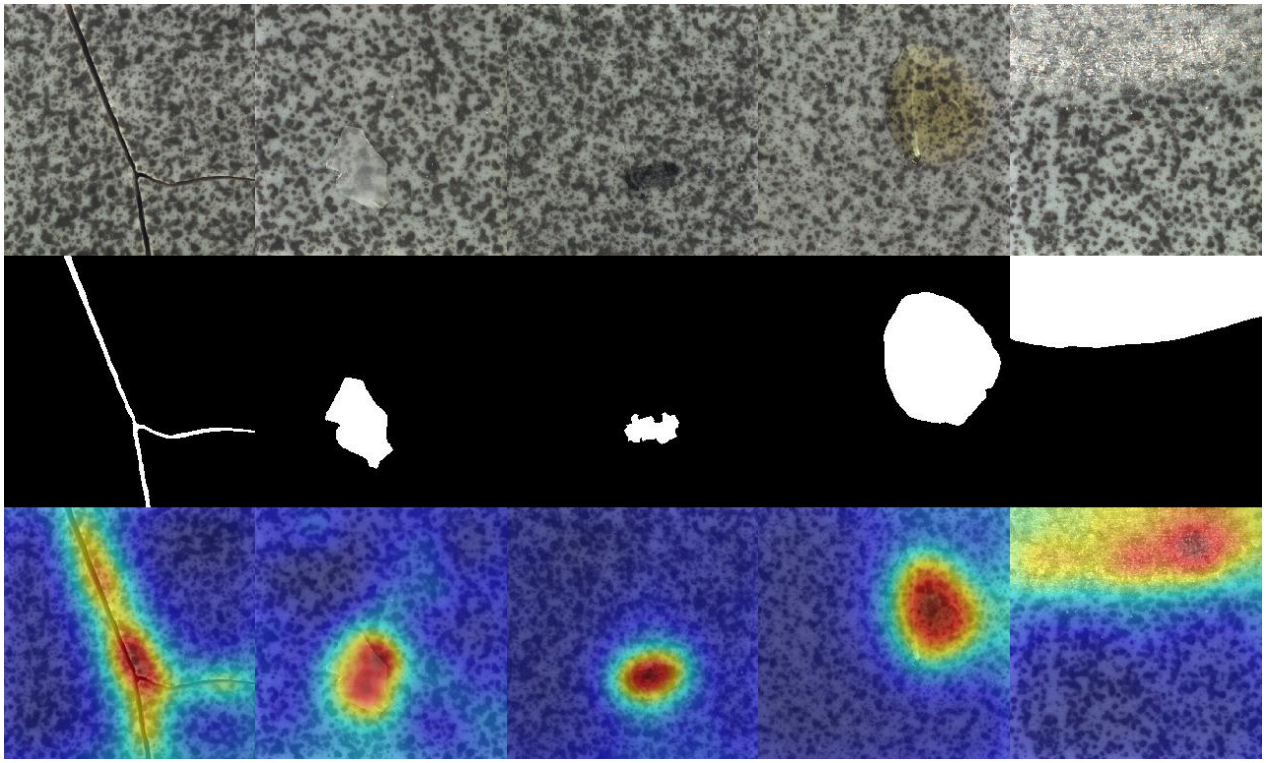

Figure 4. Tile anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

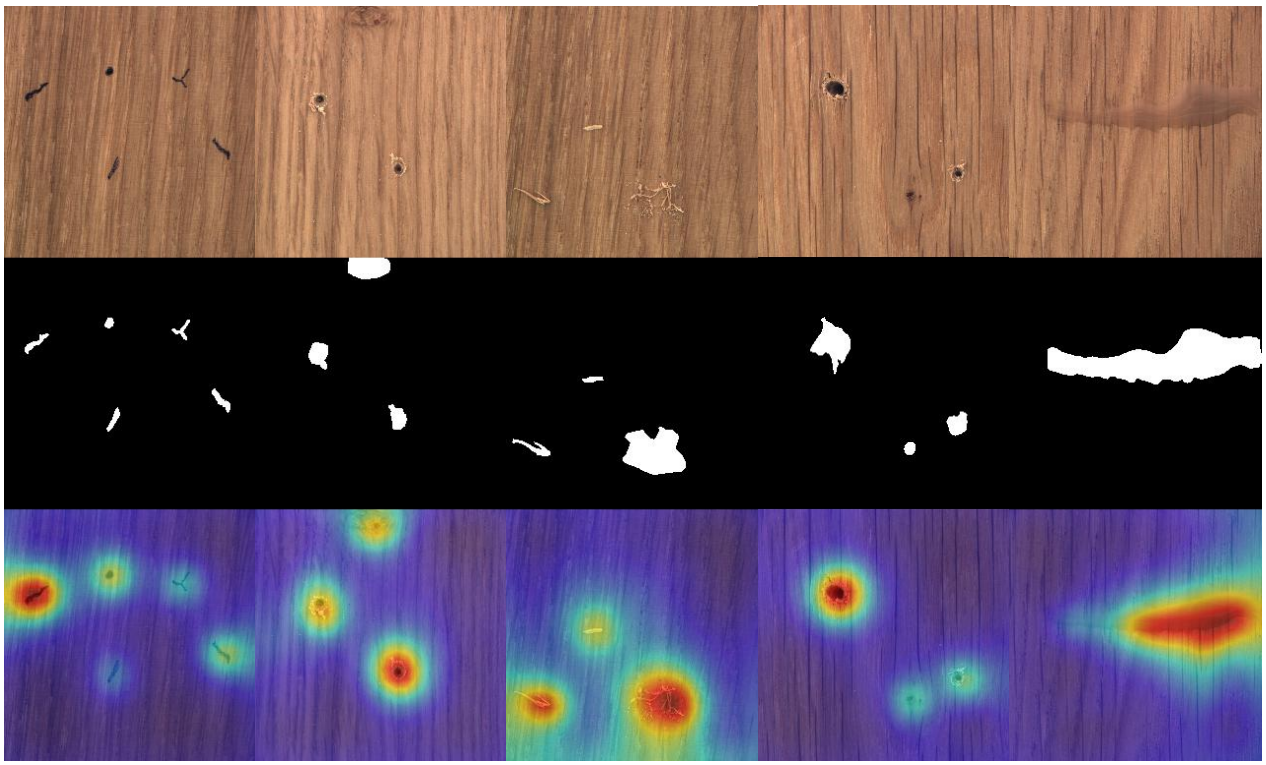

Figure 5. Wood anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

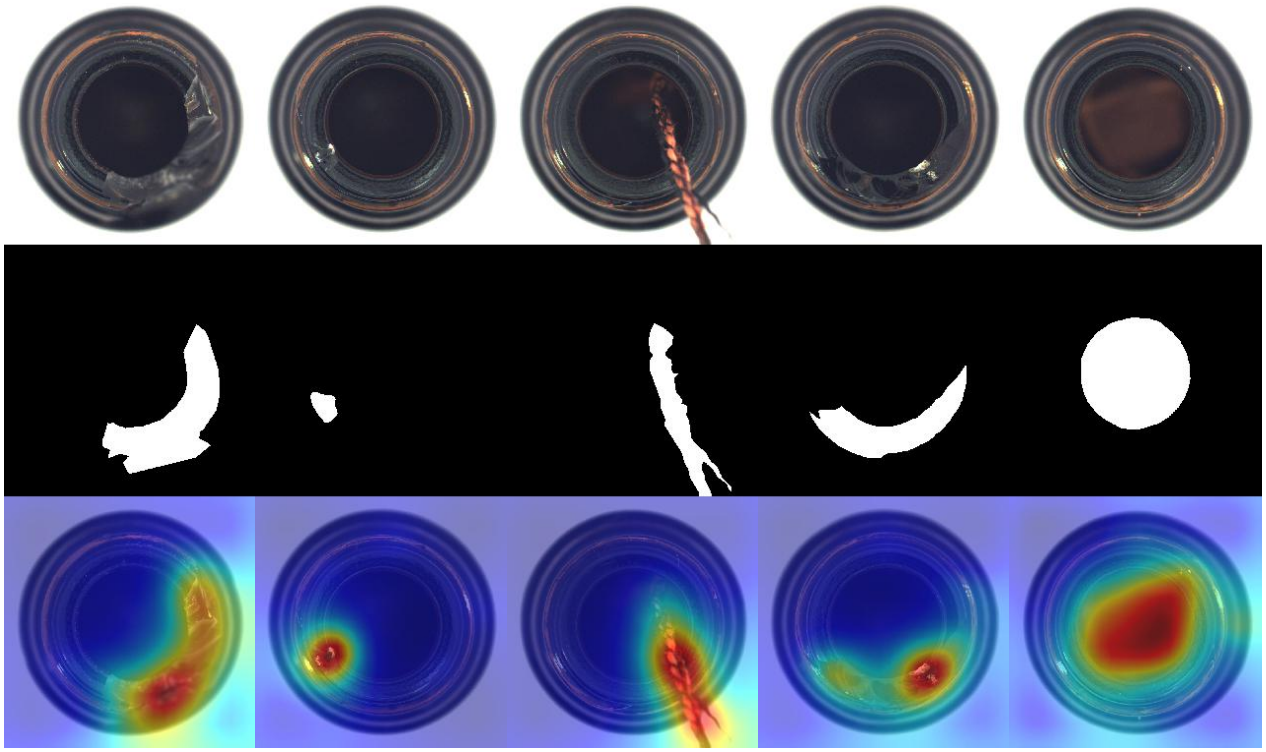

Figure 6. Bottle anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

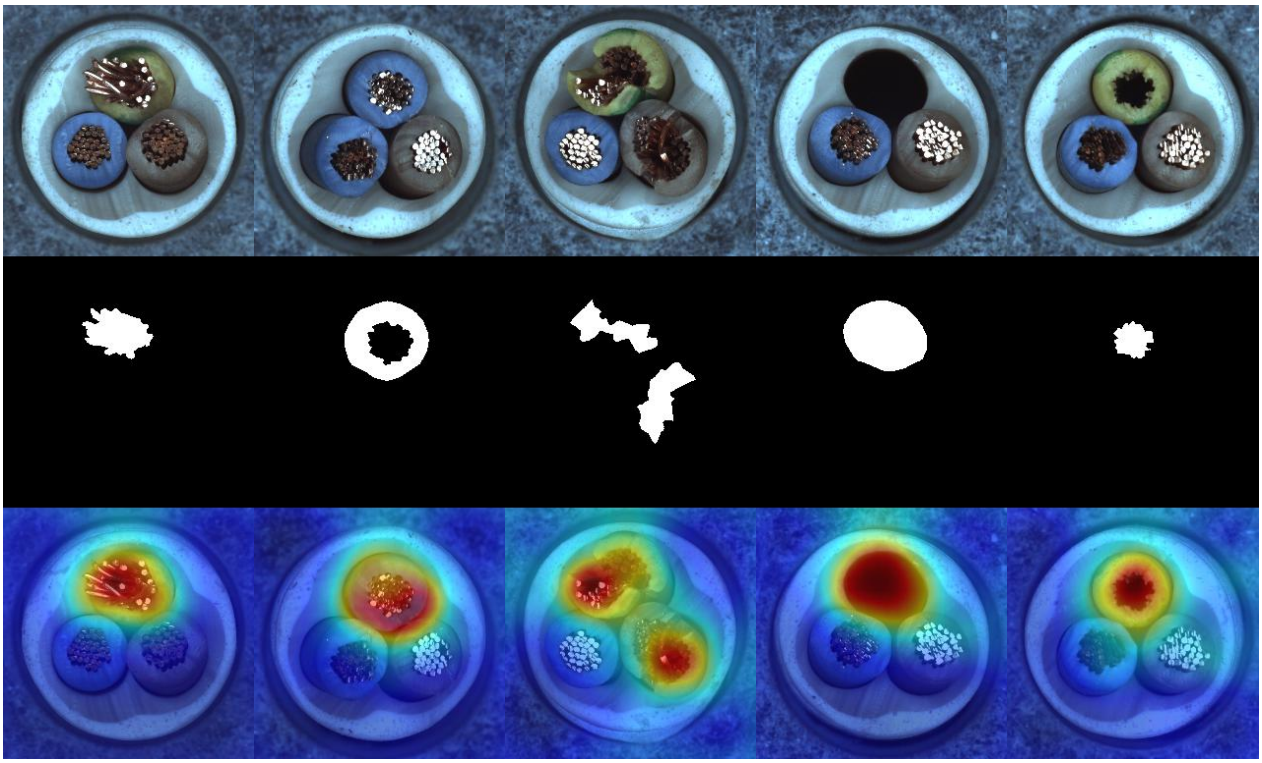

Figure 7. Cable anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

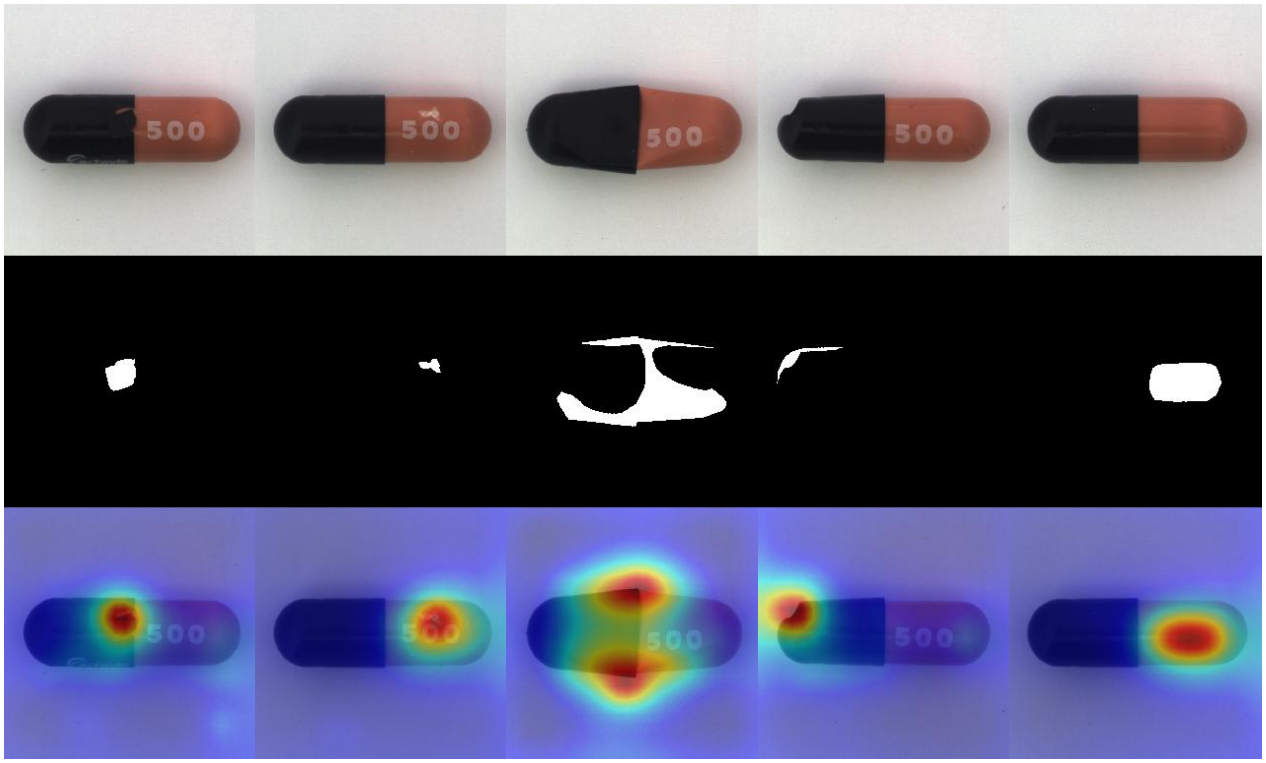

Figure 8. Capsule anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

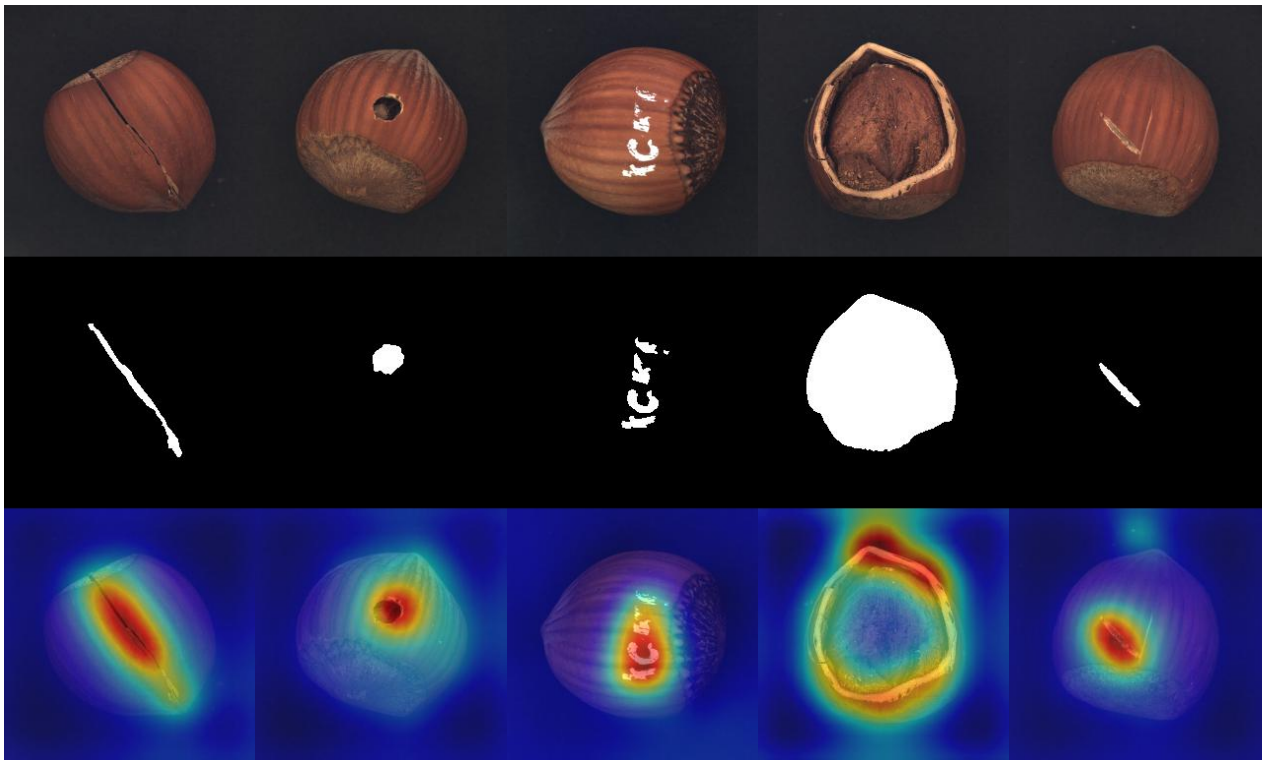

Figure 9. Hazelnut anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

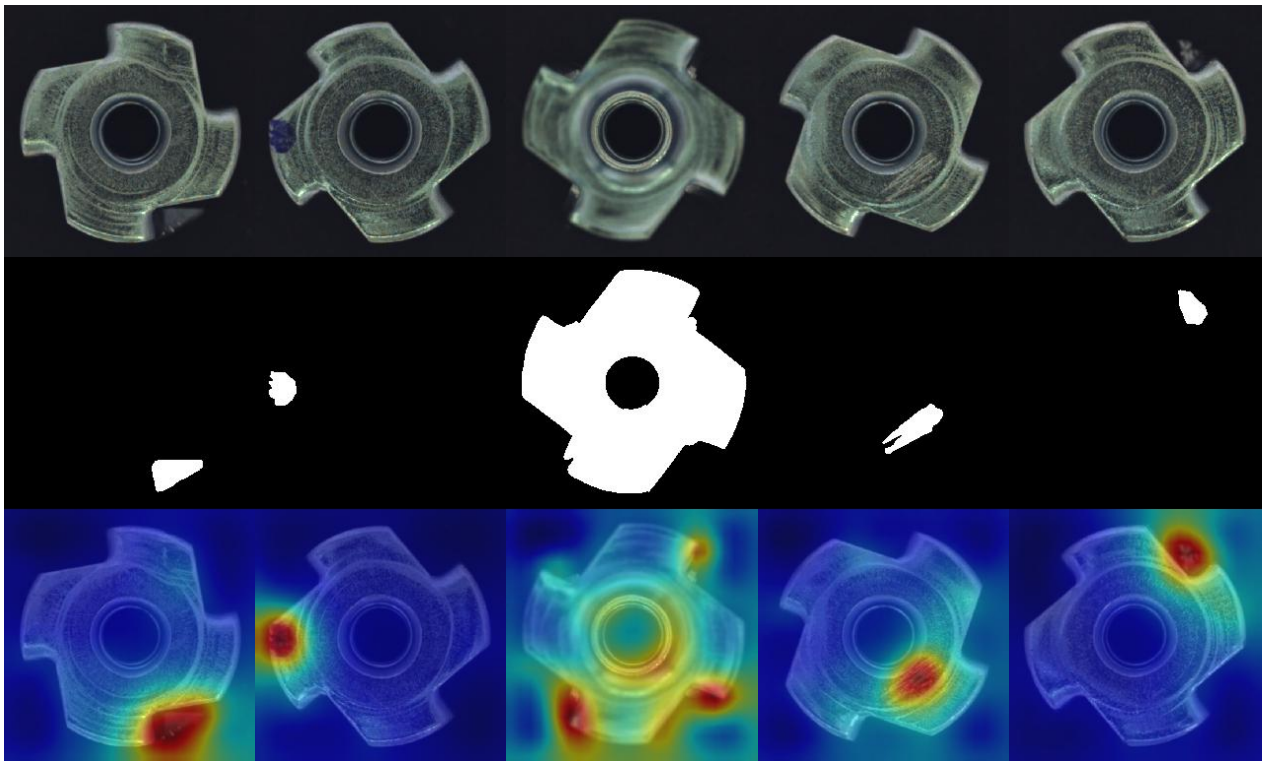

Figure 10. Metal nut anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

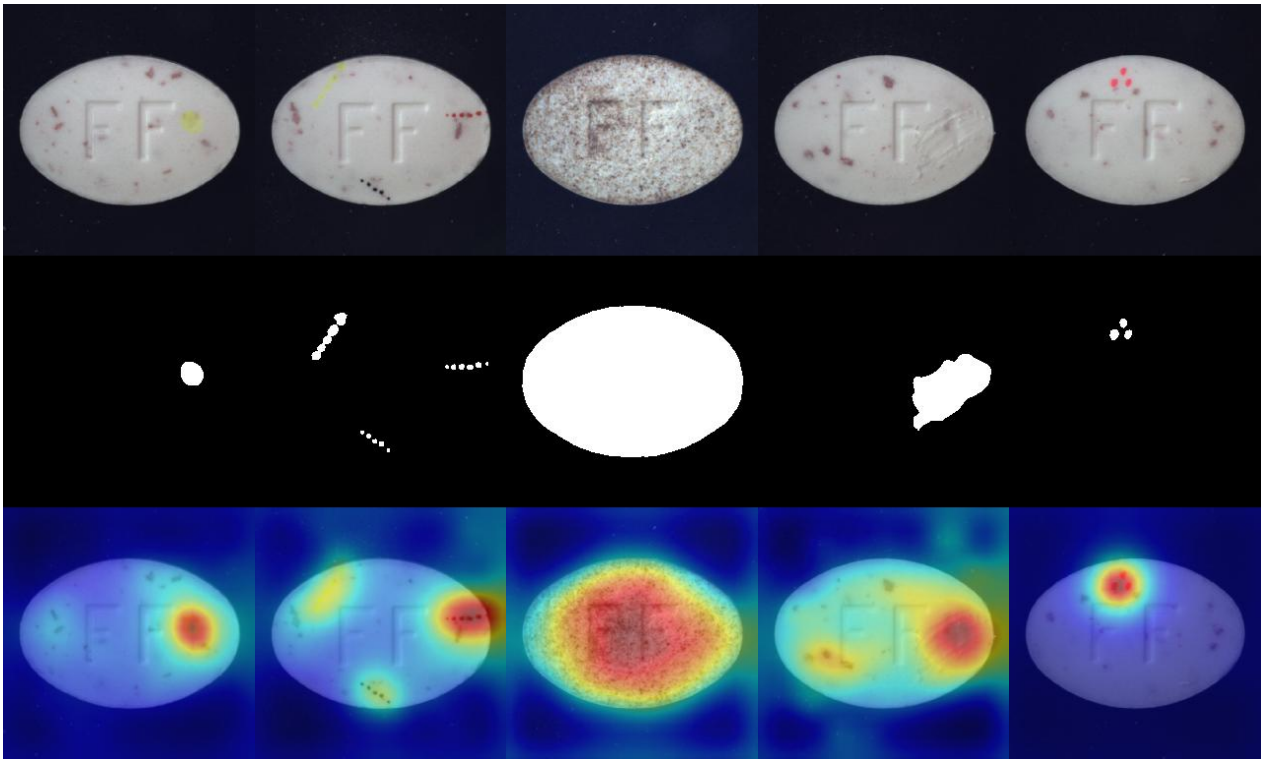

Figure 11. Pill anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

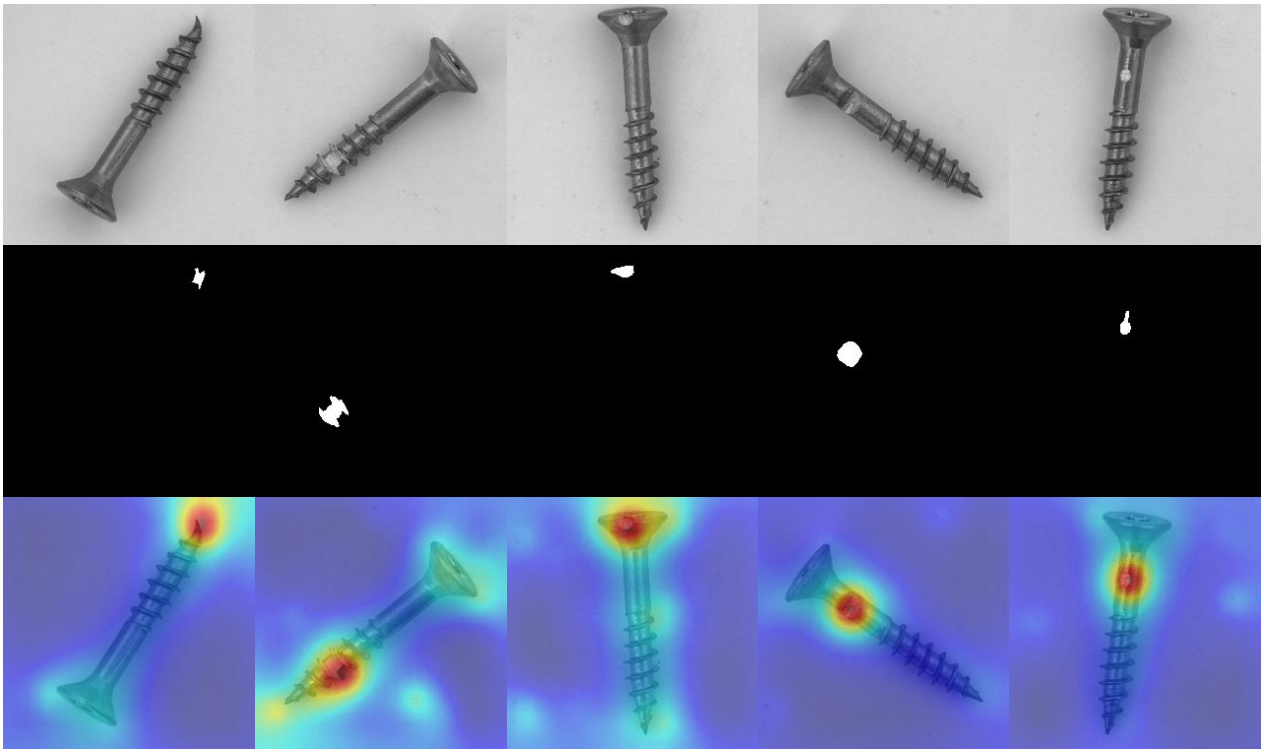

Figure 12. Screw anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

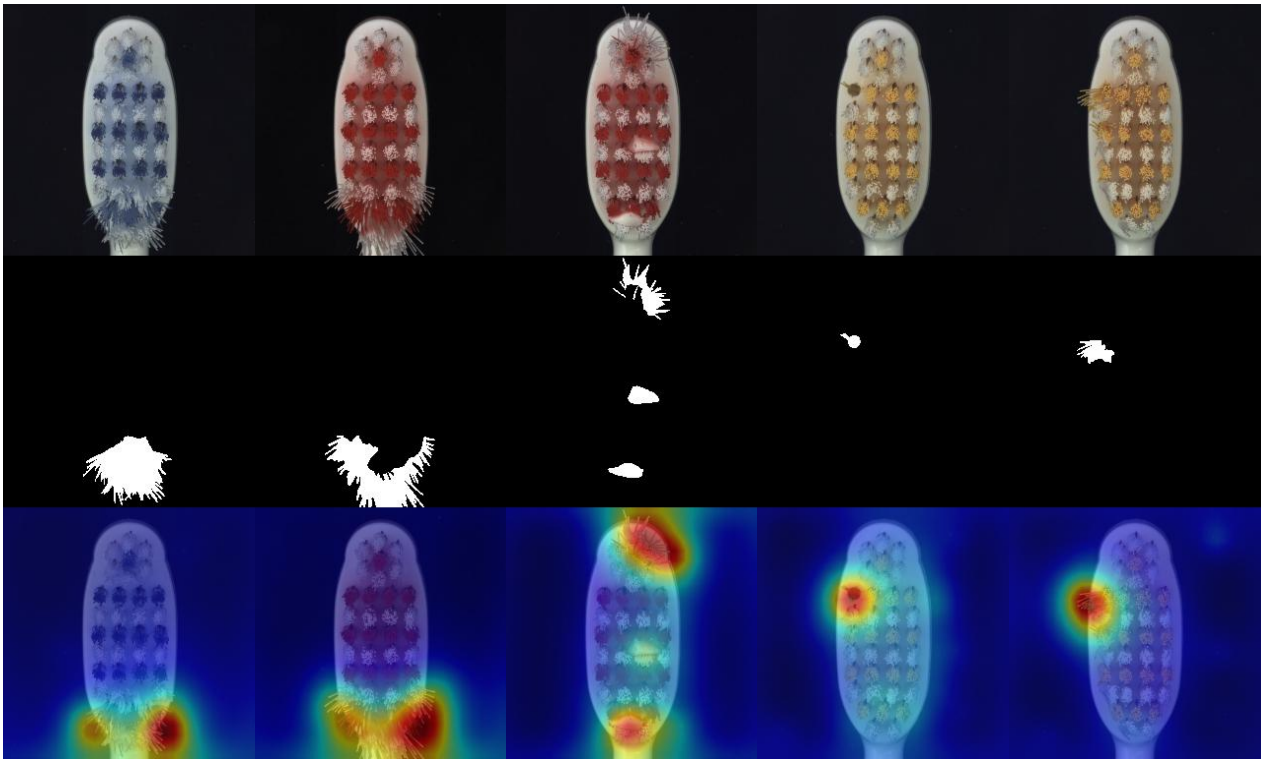

Figure 13. Toothbrush anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

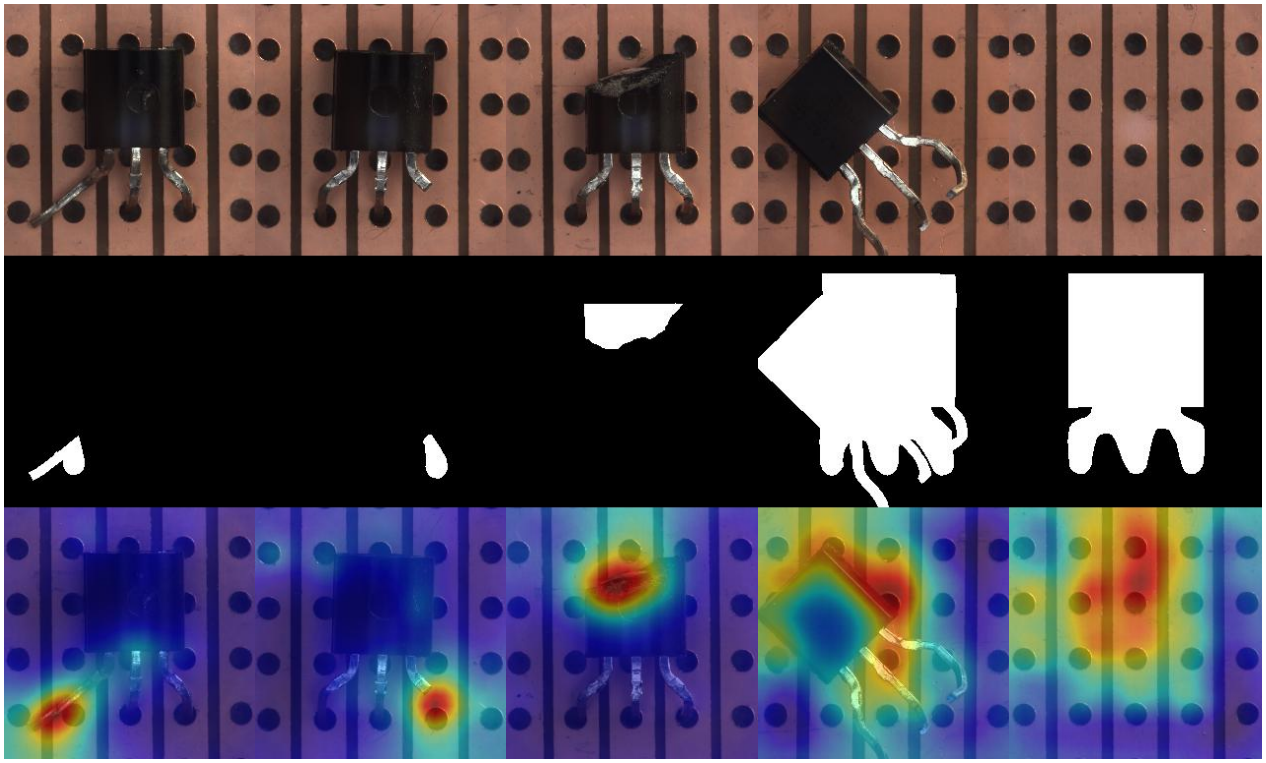

Figure 14. Transistor anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.

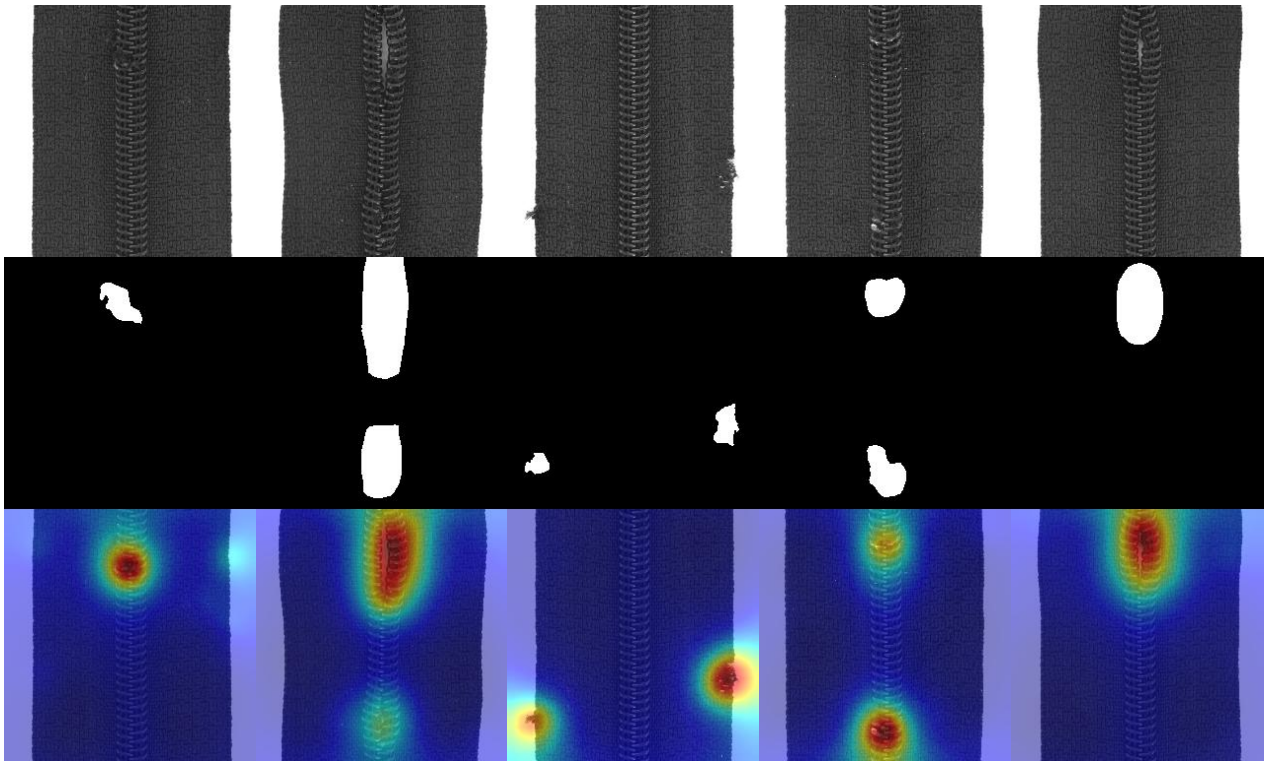

Figure 15. Zipper anomaly localization results. From top to bottom: abnormal samples, ground-truth, and the anomaly score maps produced by our algorithm.