Anomaly Detection via Reverse Distillation from One-Class Embedding - Supplementary Document

In this supplementary material, we present more details on our reverse distillation method for anomaly detection.Specifically, we describe the architectures of our proposed Reverse Distillation, especially the student decoder design, in Appendix A . Appendix B elaborates the implementation and results for one-class novelty detection. More visualizations on MVTec [2] for anomaly detection and localization are provided in Appendix C. Finally, We discuss future work that may alleviate the limitations of current work in Appendix D. Our code is available at https://github.com/hq-deng/RD4AD.

A. Model Architectures

Our reverse distillation framework consists of a teacher encoder and a student decoder. We specify our implementation details on the reverse distillation in experiments in this section.

Teacher Encoder: In our experiments, we take ResNet18 [3], ResNet50 [3] and WideResNet50 [13] as our backbones, respectively. We use the first, second, and third residual blocks in the backbone as the architecture of the teacher encoder. We use the pre-trained model from Pytorch[1] as the knowledge source of the teacher encoder.

Student Decoder: The architecture of the student decoder is symmetrical but reversed compared to the teacher encoder. For example, a down-sampling layer in the teacher encoder will be replaced by an up-sampling layer in the student decoder. In our experiments, ResNet is used as the teacher encoder. To implement the corresponding student decoder, we use the deconvolutional layer [14] with a kernel of 2 and stride of 2 as the up-sampling layer. For each decoder block, the up-sampling is performed by the first layers. Other components in the original residual layers [3, 13] are untouched. The detailed architectures are shown in Tab. 1.

B. Details for One-Class Novelty Detection

Following previous works on one-class novelty detection, we train the model with samples from a single class and detect out-of-distribution samples from other classes. All images in the MNIST [8], F-MNIST [12] and CIFAR-10 [6] datasets are in their original scale, which are 28×28, 28×28, and 32×32, respectively. We take ResNet18 [3] as the backbone of the teacher encoder. The student decoder is implemented as described in Tab. 1. We train 200 epochs with a batch size of 16. The model is optimized by Adam [5] optimizer with learning rate of 0.001.

Due to page limitation, we only compare the quantitative averages of once-class novelty detection in the main paper.Here, Tab. 2 presents the detailed numerical results of novelty detection with training on each class of samples.

C. Visualizations of Anomaly Detection and Localization on MVTec

Positive samples: We visualize the normalized anomaly score plot as a heat map. Figs. 1 to 5 show the various anomaly localization results for the texture images. Figs. 6 to 15 show the various anomaly localization results for the object images.

Negative samples: We also enumerate all the negative samples in this document. Negative samples are those in which the abnormal region is not significantly distinguished from the normal region. We show all the negative samples of the textures in Fig. 16. The negative samples of the objects are shown in Fig. 17. Note that since the heat maps are normalized, the occurrence of high heats in non-abnormal regions does not mean that they are predicted to be abnormal. Negative samples occur because abnormal areas are not significantly detected or are influenced by random factors, such as spots on the pill. All the types of defects present in the negative samples also have positive samples that are successfully detected. We discuss further improvement of future work in Appendix D.

[1] https://pytorch.org/

block name	output size	De-ResNet-18	De-ResNet-50	De-WideResNet-50
deconv_3	16×16	$\begin{bmatrix} 2 \times 2, 256 \\ 3 \times 3, 256 \end{bmatrix}$ $\begin{bmatrix} 2 \times 2, 256 \\ 3 \times 3, 256 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 256 \\ 2 \times 2, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 1$ $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 5$	$\begin{bmatrix} 1 \times 1, 256 \times 2 \\ 2 \times 2, 256 \times 2 \\ 1 \times 1, 1024 \times 2 \end{bmatrix} \times 1$ $\begin{bmatrix} 1 \times 1, 256 \times 2 \\ 2 \times 2, 256 \times 2 \\ 1 \times 1, 1024 \times 2 \end{bmatrix} \times 5$
deconv_2	32×32	$\begin{bmatrix} 2 \times 2, 128 \\ 3 \times 3, 128 \end{bmatrix}$ $\begin{bmatrix} 2 \times 2, 128 \\ 3 \times 3, 128 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 128 \\ 2 \times 2, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 1$ $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \times 2 \\ 2 \times 2, 128 \times 2 \\ 1 \times 1, 512 \times 2 \end{bmatrix} \times 1$ $\begin{bmatrix} 1 \times 1, 128 \times 2 \\ 2 \times 2, 128 \times 2 \\ 1 \times 1, 512 \times 2 \end{bmatrix} \times 3$
deconv_1	64×64	$\begin{bmatrix} 2 \times 2, 64 \\ 3 \times 3, 64 \end{bmatrix}$ $\begin{bmatrix} 2 \times 2, 64 \\ 3 \times 3, 64 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 64 \\ 2 \times 2, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 1$ $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \times 2 \\ 2 \times 2, 64 \times 2 \\ 1 \times 1, 256 \times 2 \end{bmatrix} \times 1$ $\begin{bmatrix} 1 \times 1, 64 \times 2 \\ 2 \times 2, 64 \times 2 \\ 1 \times 1, 256 \times 2 \end{bmatrix} \times 2$

Table 1. Architectures for student decoder on MVTec [2].

Dataset	MNIST				F-MNIST				CIFAR-10			
Method	LSA	OCGAN	HRN	Ours	LSA	OCGAN	HRN	Ours	LSA	OCGAN	HRN	Ours
0	99.3	99.8	99.5	99.9	91.6	85.5	92.7	93.4	73.5	75.7	77.3	89.3
1	99.9	99.9	99.9	99.9	98.3	93.4	98.5	99.6	58.0	53.1	69.9	91.7
2	95.9	94.2	96.5	98.9	87.8	85.0	88.5	92.5	69.0	64.0	60.6	78.1
3	96.6	96.3	97.4	98.9	92.3	88.1	93.1	95.5	54.2	62.0	64.4	71.1
4	95.6	97.5	97.2	98.1	89.7	85.8	92.1	93.9	76.1	72.3	71.5	86.5
5	96.5	98.0	97.2	99.1	90.7	88.5	91.3	96.5	54.6	62.0	67.4	83.9
6	99.4	99.1	99.2	99.9	84.1	77.5	79.8	83.0	75.1	72.3	77.4	91.6
7	98.0	98.1	97.6	99.1	97.7	93.9	99.0	99.5	53.5	57.5	64.9	90.2
8	95.3	93.9	94.3	99.0	91.0	82.7	94.6	97.3	71.7	82.0	82.5	92.4
9	98.1	98.1	97.1	99.5	98.4	97.8	98.8	98.7	54.8	55.4	77.3	90.1
Average	*97.5*	*97.5*	*97.6*	***99.2***	*92.2*	*87.8*	*92.8*	***95.0***	*64.1*	*65.7*	*71.3*	***86.5***

Table 2. AUROC(%) results for One-Class Novelty Detection compared with LSA [1], OCGAN [10], and HRN [4].

D. Future Work

We used models pre-trained on ImageNet [7] as teacher encoders in our experiments. We believe that studying a self-supervised trained model is a feasible direction. Since the parameters of the teacher encoder need to be fixed, fine-tuning [11] or self-supervised [9] training on anomaly-free samples would help to produce better representations.

In addition, we observe that the anomaly detection and localization results are still affected by noise. The anomaly-free samples in the MVTec [2] are limited, but traditional data augmentation strategies might make the anomaly-free samples out of the distribution. Developing reasonable data augmentation strategies in training will improve the overall AD performance.

In conclusion, we believe that the future work mentioned above combined with reverse distillation will improve the anomaly detection performance even further.

References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019. 2

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019. 1, 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[4] Wenpeng Hu, Mengyu Wang, Qi Qin, Jinwen Ma, and Bing Liu. Hrn: A holistic approach to one class learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19111–19124. Curran Associates, Inc., 2020. 2

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[6] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. 1

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 2

[8] Yann LeCun. The mnist database of handwritten digits, 1998. 1

[9] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9664–9674, June 2021. 2

[10] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019. 2

[11] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2806–2814, June 2021. 2

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 1

[13] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1

[14] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, 2010. 1

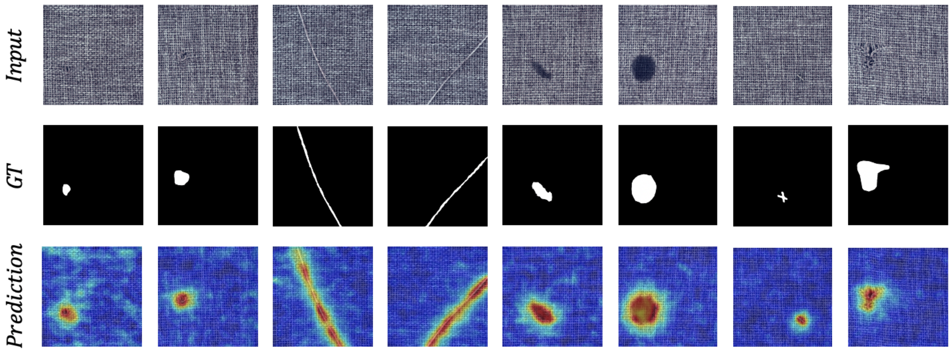

Figure 1. Positive samples of the carpet.

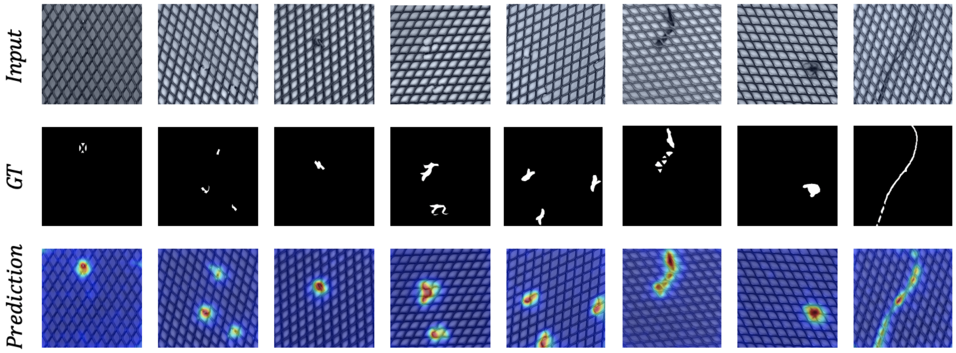

Figure 2. Positive samples of the grid.

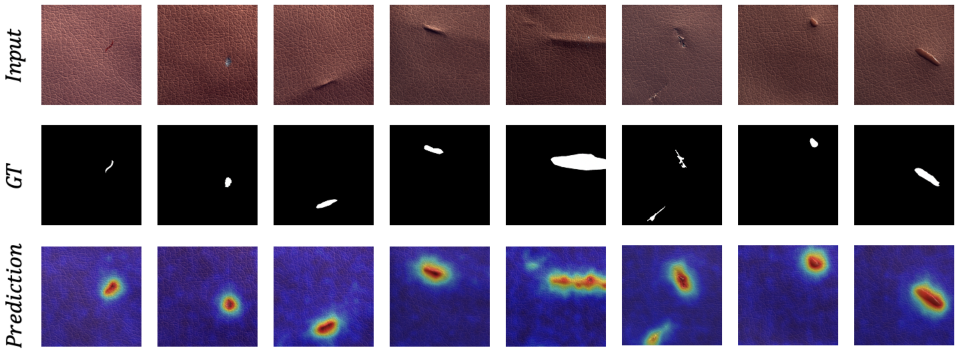

Figure 3. Positive samples of the leather.

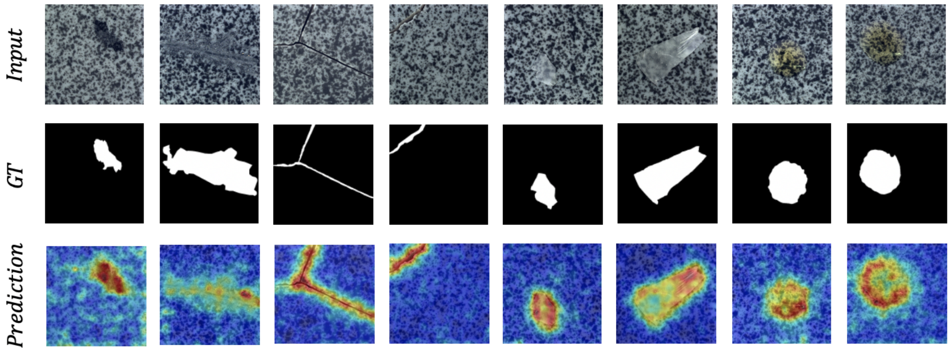

Figure 4. Positive samples of the tile.

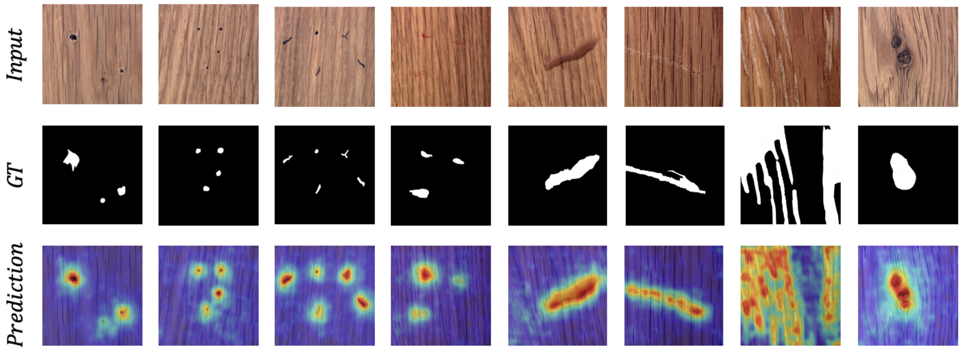

Figure 5. Positive samples of the wood.

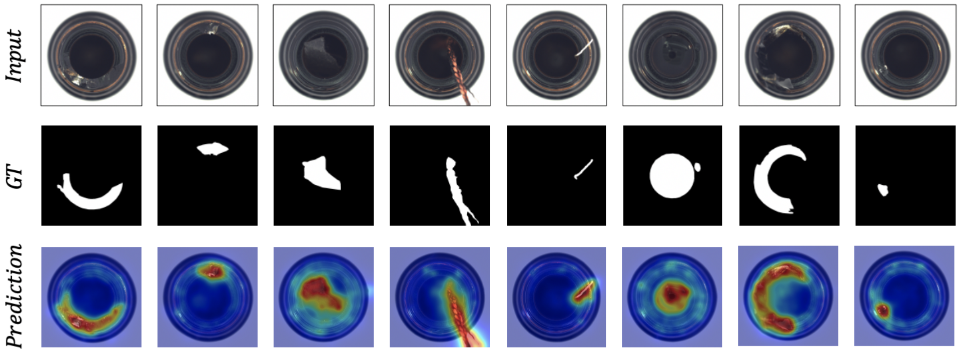

Figure 6. Positive samples of the bottle.

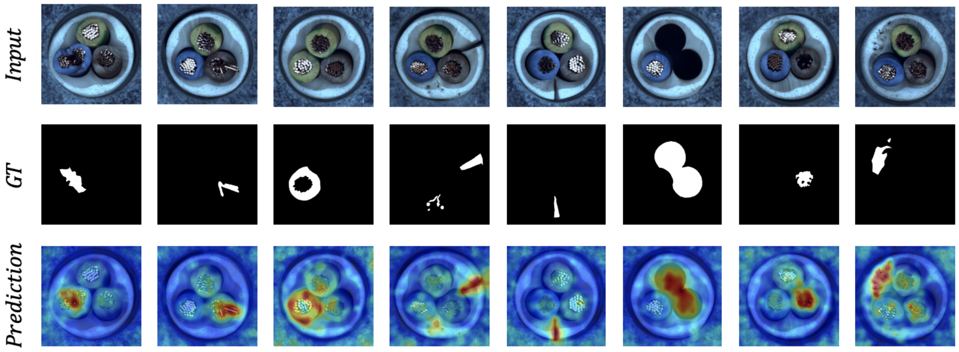

Figure 7. Positive samples of the cable.

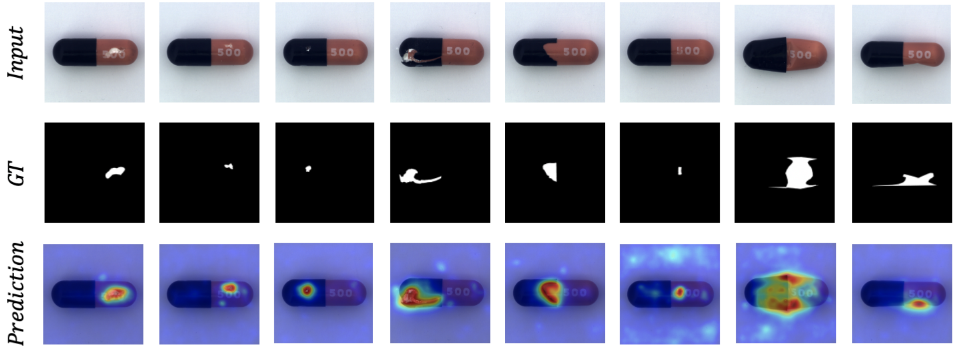

Figure 8. Positive samples of the capsule.

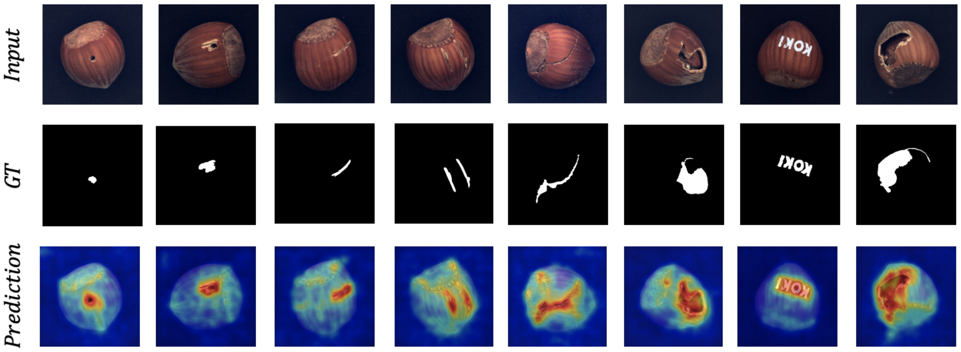

Figure 9. Positive samples of the hazelnut.

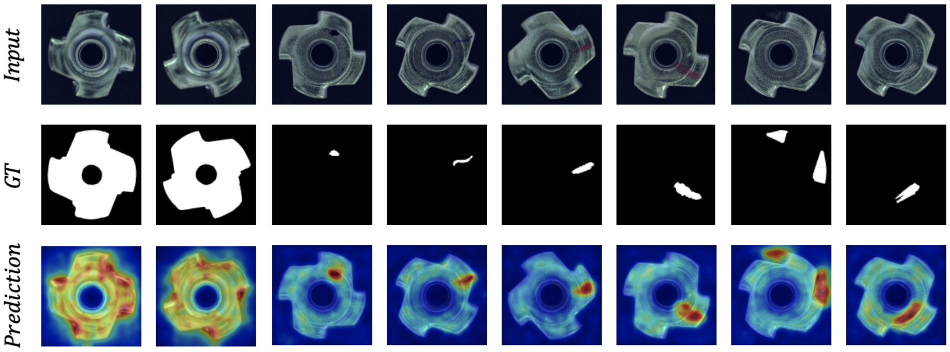

Figure 10. Positive samples of the metal nut.

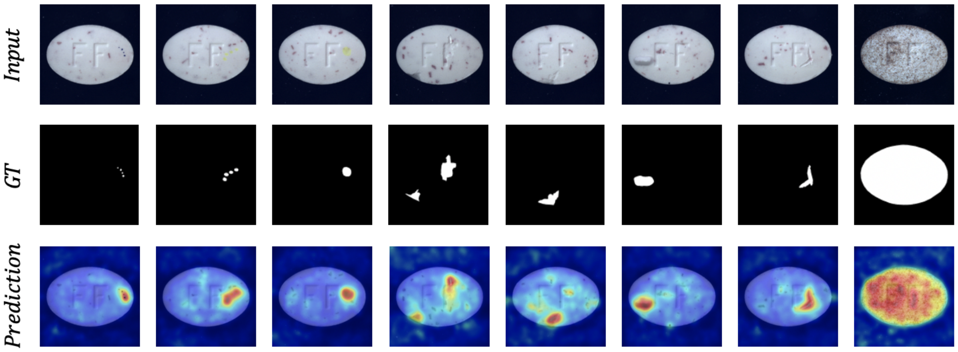

Figure 11. Positive samples of the pill.

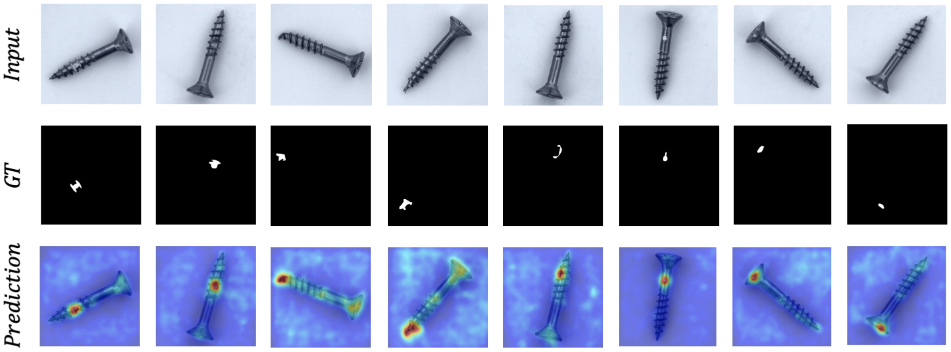

Figure 12. Positive samples of the screw.

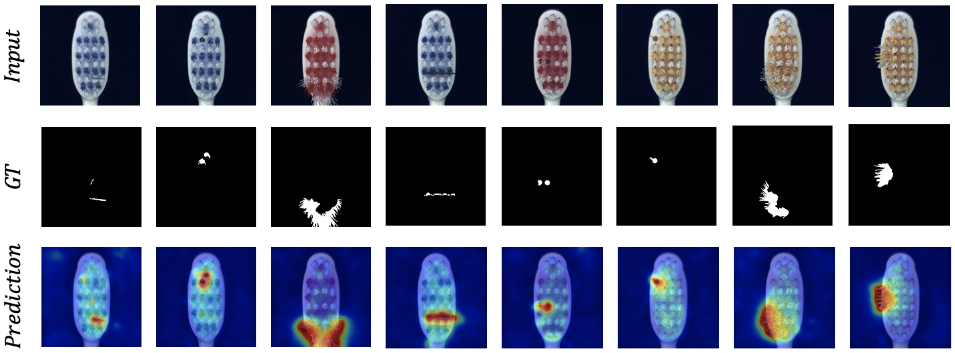

Figure 13. Positive samples of the toothbrush.

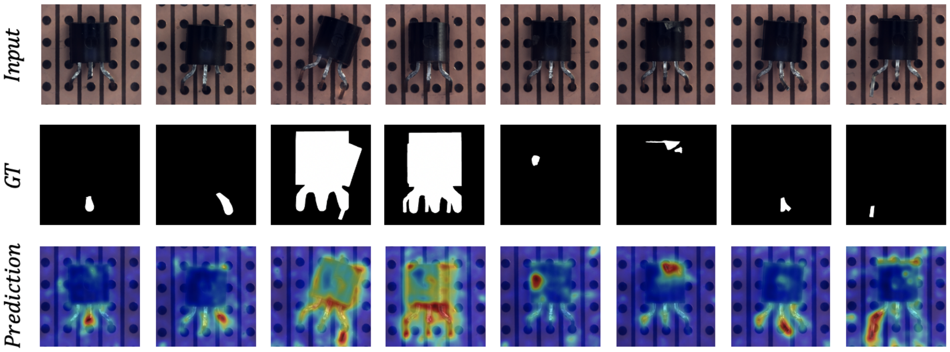

Figure 14. Positive samples of the transistor.

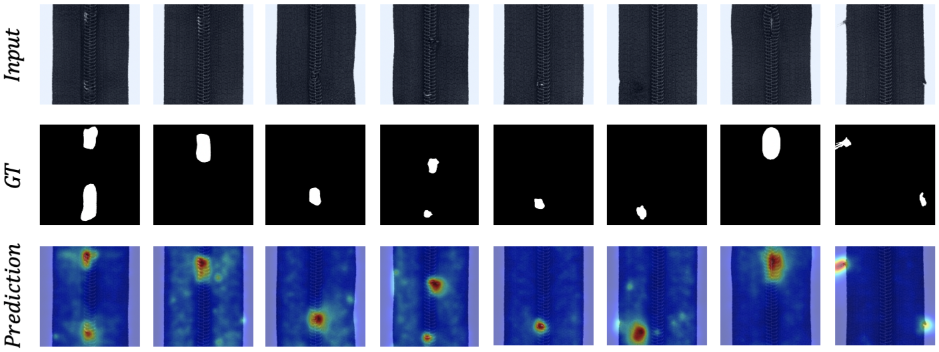

Figure 15. Positive samples of the zipper.

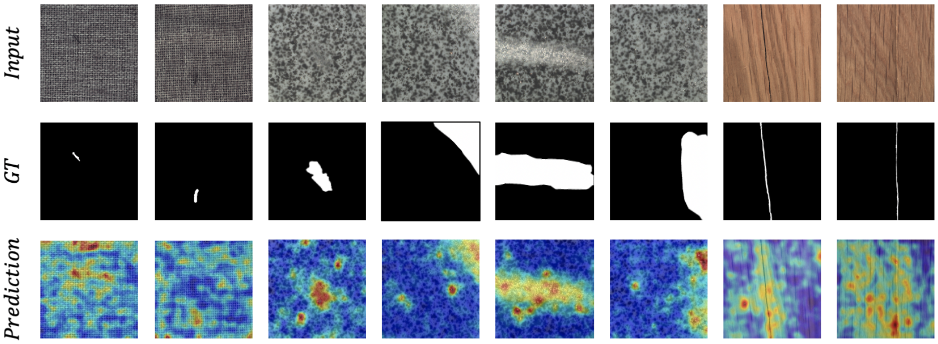

Figure 16. Negative samples of the textures.

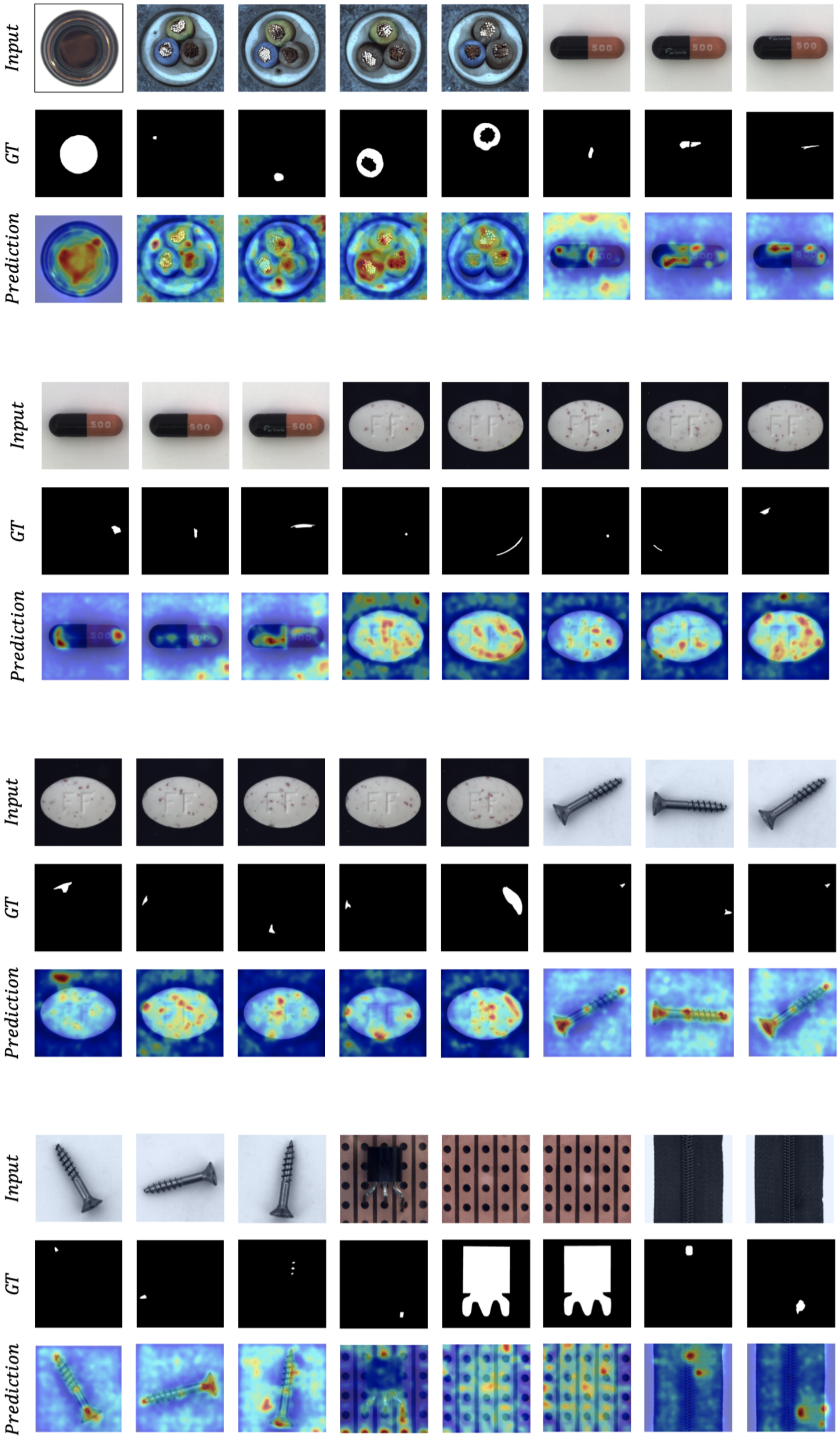

Figure 17. Negative samples of the objects.