Supplementary materials for "Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation"

Jihun Yi[0000-0001-5762-6643] and Sungroh Yoon[0000-0002-2367-197X]

Data Science and Artificial Intelligence Laboratory
Electrical and Computer Engineering
Seoul National University, Seoul, South Korea
{t080205,sryoon}@snu.ac.kr

S1 Pseudo code

Algorithm 1 Patch SVDD (train)

Input normal images $\{\mathbf{x}\}$, hyperparameter λ, encoder f_θ, and classifier C_ϕ
for patch $\mathbf{p}$ in $\{\mathbf{x}\}$ **do** ▷ Train the encoder
    $\mathbf{p}_1 \leftarrow$ `RandomJitter`$(\mathbf{p})$
    $\mathcal{L}_{\text{SVDD'}} \leftarrow \|f_\theta(\mathbf{p}) - f_\theta(\mathbf{p}_1)\|_2$
    $\mathbf{p}_2, y \leftarrow$ `RandomNeighborhood`$(\mathbf{p})$ ▷ A neighborhood in the 3×3 grid
    $\mathcal{L}_{\text{SSL}} \leftarrow$ `Cross-entropy`$\left(y, C_\phi\left(f_\theta(\mathbf{p}), f_\theta(\mathbf{p}_2)\right)\right)$
    Backprop $\mathcal{L}_{\text{Patch SVDD}} \leftarrow \lambda\mathcal{L}_{\text{SVDD'}} + \mathcal{L}_{\text{SSL}}$
end for
$f_{\text{big}}, f_{\text{small}} \leftarrow f_\theta$ ▷ Split encoders
$S_{\text{big}}, S_{\text{small}} \leftarrow \varnothing$ ▷ Sets of normal features
for patch $\mathbf{p}$ in $\{\mathbf{x}\}$ **do** ▷ Patch size K with stride S
    $S_{\text{big}} \leftarrow S_{\text{big}} \cup \{f_{\text{big}}(\mathbf{p})\}$
end for
for patch $\mathbf{p}$ in $\{\mathbf{x}\}$ **do** ▷ Patch size K with stride S
    $S_{\text{small}} \leftarrow S_{\text{small}} \cup \{f_{\text{small}}(\mathbf{p})\}$
end for
return $(S_{\text{big}}, S_{\text{small}})$, $(f_{\text{big}}, f_{\text{small}})$ ▷ Normal features and trained encoders

Algorithm 1 trains a hierarchical encoder using $\mathcal{L}_{\text{Patch SVDD}}$. After the training, sets of features of normal patches are extracted using the trained multi-scale encoders. The outputs of Algorithm 1 are the sets of normal features and trained encoders. Algorithm 2 performs inspection on a query image and outputs the anomaly map and anomaly score.

Algorithm 2 Patch SVDD (test)

Input query image $\mathbf{x}$, normal feature sets $(S_{\text{big}}, S_{\text{small}})$, and encoders $(f_{\text{big}}, f_{\text{small}})$
Initialize $\mathcal{M}_{\text{big}}$ and $\mathcal{M}_{\text{small}}$
for patch $\mathbf{p}$ in $\mathbf{x}$ **do** ▷ Patch size K with stride S
  $d \leftarrow \min_{h \in S_{\text{big}}} \|f_{\text{big}}(\mathbf{p}) - h\|_2$ ▷ Anomaly score of a patch
  Distribute d to $\mathcal{M}_{\text{big}}$ of each pixel in $\mathbf{p}$
end for
for patch $\mathbf{p}$ in $\mathbf{x}$ **do** ▷ Patch size K with stride S
  $d \leftarrow \min_{h \in S_{\text{small}}} \|f_{\text{small}}(\mathbf{p}) - h\|_2$ ▷ Anomaly score of a patch
  Distribute d to $\mathcal{M}_{\text{small}}$ of each pixel in $\mathbf{p}$
end for
$\mathcal{M}_{\text{multi}} \leftarrow \mathcal{M}_{\text{small}} \odot \mathcal{M}_{\text{big}}$ ▷ Element-wise multiplication
$a \leftarrow \max \mathcal{M}_{\text{multi}}$ ▷ Anomaly score
return $\mathcal{M}_{\text{multi}}$, a ▷ Anomaly map and anomaly score

S2 Results

S2.1 Numerical results

Table S1: **Anomaly detection (`Det.`) and segmentation performances (`Seg.`) of proposed Patch SVDD on MVTec AD [1] dataset.** The inspection performances for each class are given in AUROC, and the average values are also reported in Table 1 of the main paper.

	Patch SVDD	
Classes	`Det.`	`Seg.`
bottle	0.986	0.981
cable	0.903	0.968
capsule	0.767	0.958
carpet	0.929	0.926
grid	0.946	0.962
hazelnut	0.920	0.975
leather	0.909	0.974
metal_nut	0.940	0.980
pill	0.861	0.951
screw	0.813	0.957
tile	0.978	0.914
toothbrush	1.000	0.981
transistor	0.915	0.970
wood	0.965	0.908
zipper	0.979	0.951
Average	**0.921**	**0.957**

Table S2: **The effect of hierarchical encoding.** Aggregating the results from multi-scale inspection boosts the performance, and adopting hierarchical structure to the encoder is helpful as well. The plot of the data is provided in Fig. 12 of the main paper.

Hierarchical	K	`Det.`	`Seg.`
✗	64	0.810	0.879
✓	64	0.894	0.932
✓	32	0.902	0.957
✓	Agg. (64 & 32)	**0.921**	**0.957**

S2.2 Anomaly maps

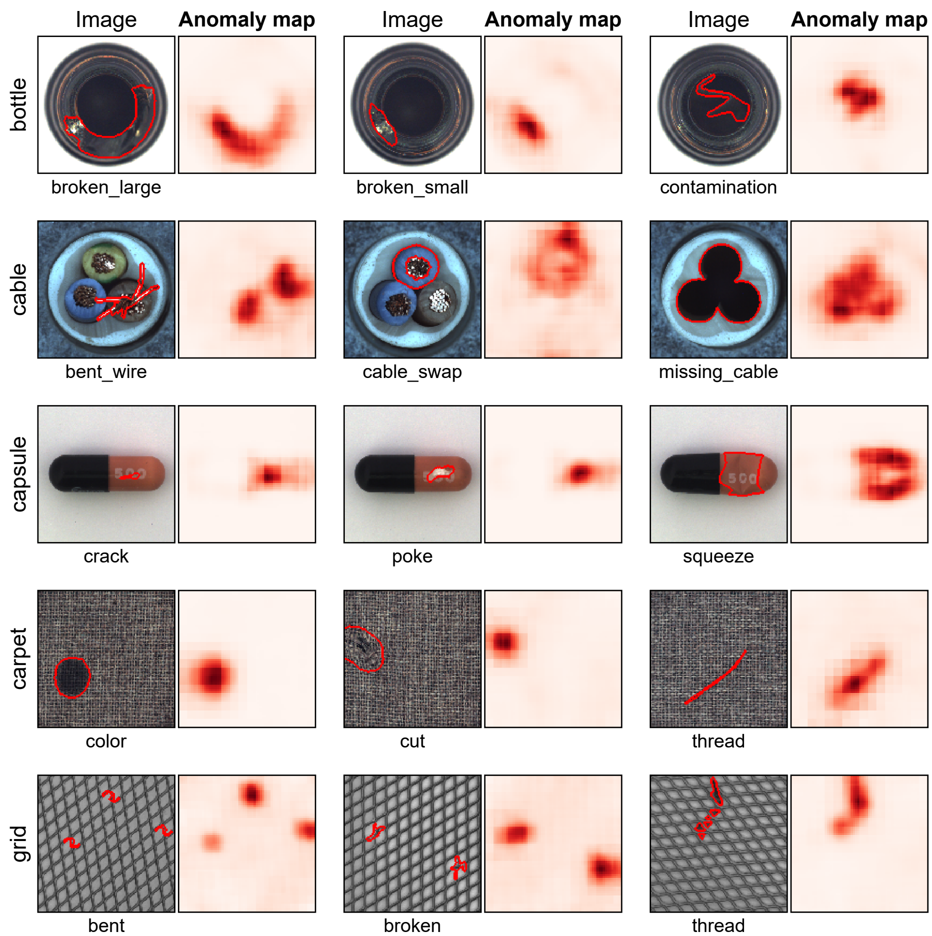

Fig. S1: **Anomaly maps generated by the proposed method.** Patch SVDD generates anomaly maps of the images in each class of MVTec AD [1] dataset. The ground truth defect annotations are depicted as red contours in the image, and the darker heatmap indicates higher anomaly scores. The name of the class is provided at the left of the image, and the type of the defect is indicated below the image.

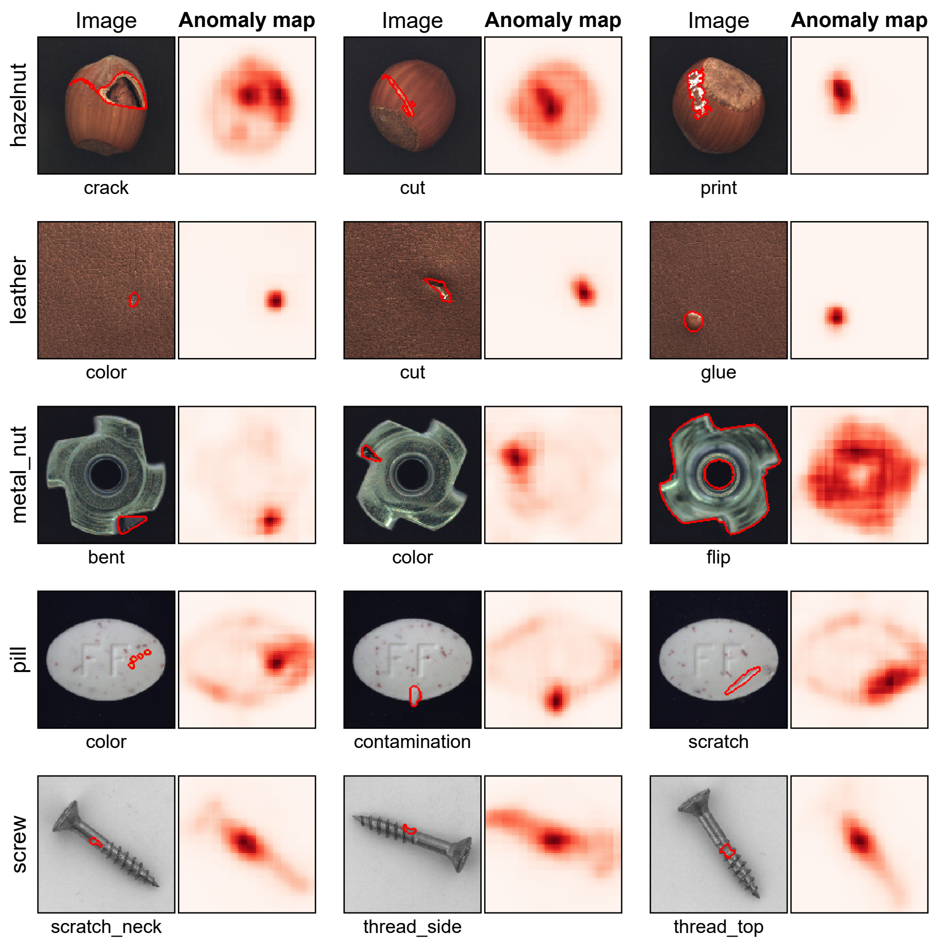

Fig. S2: **Anomaly maps generated by the proposed method.** Patch SVDD generates anomaly maps of the images in each class of MVTec AD [1] dataset. The ground truth defect annotations are depicted as red contours in the image, and the darker heatmap indicates higher anomaly scores. The name of the class is provided at the left of the image, and the type of the defect is indicated below the image.

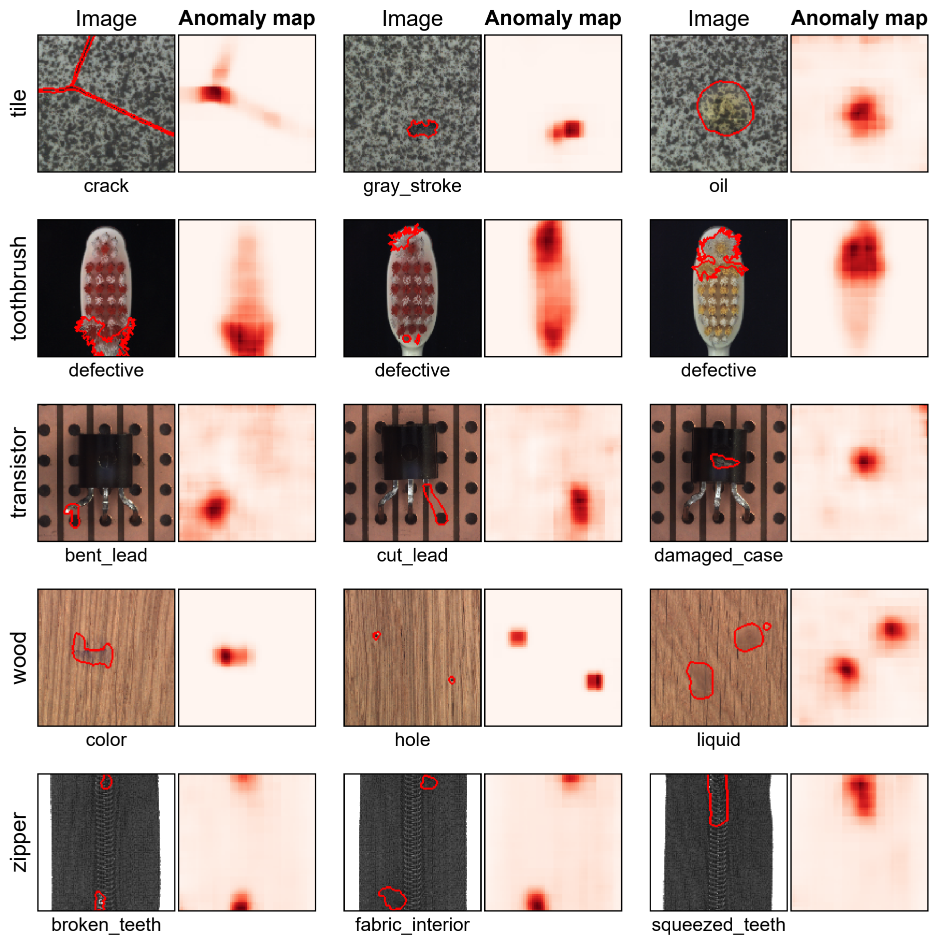

Fig. S3: **Anomaly maps generated by the proposed method.** Patch SVDD generates anomaly maps of the images in each class of MVTec AD [1] dataset. The ground truth defect annotations are depicted as red contours in the image, and the darker heatmap indicates higher anomaly scores. The name of the class is provided at the left of the image, and the type of the defect is indicated below the image.

S3 Implementation details

S3.1 Dataset

The dataset in the study, MVTec AD [1], consists of 15-class industrial images. Each class is categorized as either an *object*[1] or *texture*[2]. Each class contains 60 to 390 normal train images and 40 to 167 test images. Test images include both normal and abnormal examples, and the defects of the abnormal images are annotated at the pixel level in the form of binary masks. We downsampled every image to a resolution of 256×256. Gray-scale images are converted to RGB images by replicating the single channel to three. No data augmentation method (e.g., horizontal flip, rotation) was used for the training.

S3.2 Networks

Two neural networks are used throughout the study: an encoder and a classifier. The encoder is composed of convolutional layers only. The classifier is a two-layered MLP model having 128 hidden units per layer, and the input to the classifier is a subtraction of the features of the two patches. The activation function for both networks is a LeakyReLU [2] with a $\alpha = 0.1$. Please refer to the code[3] for the detailed architecture of the networks.

As proposed in Section 3.3 of the main paper, the encoder has a hierarchical structure. The receptive field of the encoder is $K = 64$, and that of the embedded smaller encoder is $K = 32$. Patch SVDD divides the images into patches with a size K and a stride S. The values for the strides are $S = 16$ and $S = 4$ for the encoders with $K = 64$ and $K = 32$, respectively.

S3.3 Environments

The experiments throughout the study were conducted on a machine equipped with an Intel i7-5930K CPU and an NVIDIA GeForce RTX 2080 Ti GPU. The code is implemented in python 3.7 and PyTorch [3].

References

1. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR. (2019)
2. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML. (2013)
3. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)

---

[1] bottle, cable, capsule, hazelnut, metal_nut, pill, screw, toothbrush, transistor, and zipper

[2] carpet, grid, leather, tile, and wood

[3] `https://github.com/nuclearboy95/Anomaly-Detection-PatchSVDD-PyTorch`