Towards Zero-shot 3D Anomaly Localization

Yizhou Wang[1], Kuan-Chuan Peng[2], and Yun Raymond Fu[1]
[1] Northeastern University, Boston, MA, USA; [2] Mitsubishi Electric Research Laboratories, Cambridge, MA, USA
wyzjack990122@gmail.com, kpeng@merl.com, yunfu@ece.neu.edu

Smile lab — Synergetic Media Learning Lab

MITSUBISHI ELECTRIC — Changes for the Better | MITSUBISHI ELECTRIC RESEARCH LABORATORIES

Introduction & Motivation

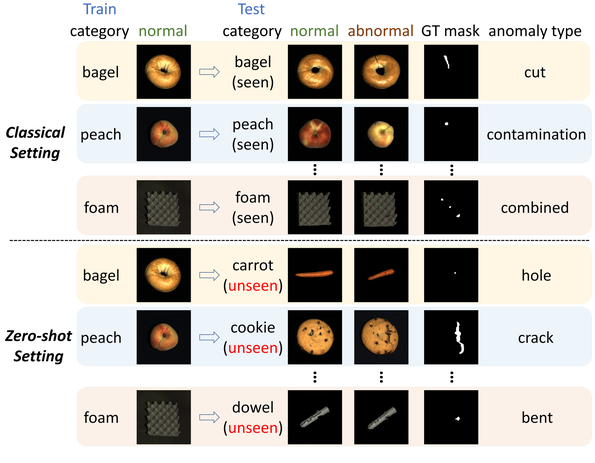
Classic (3D) Anomaly Detection [1-3] methods focus on the setting that the testing data (normal + abnormal) are from the same class and the same domain as training data. However, in real-world industrial 3D Anomaly Detection and Localization applications,

- the **normal training data** of the target objects can be **unavailable** (e.g., data privacy, export control regulations, etc.).
- the client's **data** can be **sensitive**, and the client only wants a solution that can perform well "**off-the-shelf**."

References:
[1] Paul Bergmann et al. The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization. VISAPP 2021.
[2] E. Horwitz and Y. Hoshen. An Empirical Investigation of 3D Anomaly Detection and Segmentation. arXiv 2022.
[3] Karsten Roth et al. Towards Total Recall in Industrial Anomaly Detection. CVPR 2022.

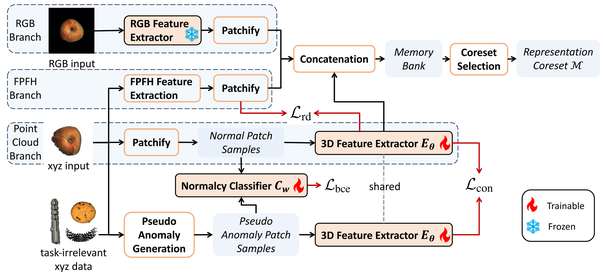
Problem Overview

We propose a new 3D anomaly detection setting: **zero-shot 3D anomaly detection**, which refrain practioners from training models for each class separately.



- During training, anomaly-free data from one class are used.
- During testing, the model is required to detect and localize anomalies from other classes.

Our Proposed Method – 3DzAL

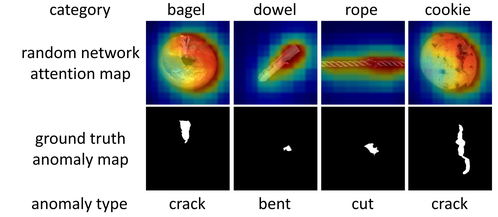


3DzAL mainly adopts 3 branches to extract features given both RGB and 3D point cloud (xyz) data of an object:

- The **RGB branch** extracts feature from 2D image data of the object using ResNet pre-trained on ImageNet.
- The **FPFH branch** extracts handcrafted FPFH features from xyz data.
- The **point cloud branch** employs a learnable network (PointNet++) to extract features. The network is trained by a patch-level **contrastive learning loss**, which takes inductive bias-based pseudo anomaly patches as negative samples and normal patches as positive samples and a **representation disentanglement loss** which pushes the FPFH features and the learned 3D features away.

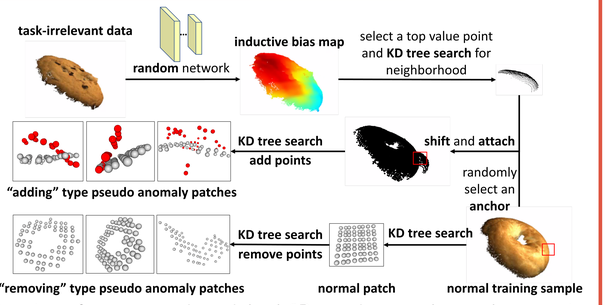
The features of the three branches are concatenated to store in the memory bank where a coreset selection is performed. In addition, a normalcy classifier is trained to classify the pseudo anomaly patch and the normal patch using the **binary cross-entropy loss**.

Inductive Bias of Random Networks



We feed the xyz data of abnormal examples as the input of a **randomly initialized** and **untrained** ResNet-50, and visualize the attention maps. These maps show that the random network has the inductive bias of covering the locations of interest, including the locations shown in the ground truth.

Pseudo Anomaly Generation



Overview of our proposed patch-level 3D pseudo anomaly sample generation process for both "adding" and "removing" type anomalies.

Training Objectives

contrastive learning loss:

$$\mathcal{L}_{con} = \sum_{x_j \in \mathcal{X}_p} \frac{-1}{|\mathcal{P}(x_j)|} * \sum_{x_p \in \mathcal{P}(x_j)} \log \frac{\exp\left(\frac{E_\theta(x_j)\cdot E_\theta(x_p)}{T\cdot \|E_\theta(x_j)\|_2 \cdot \|E_\theta(x_p)\|_2}\right)}{\sum_{x_n \in \mathcal{N}(x_j)} \exp\left(\frac{E_\theta(x_j)\cdot E_\theta(x_n)}{T\cdot \|E_\theta(x_j)\|_2 \cdot \|E_\theta(x_n)\|_2}\right)}$$

representation disentanglement loss:

$$\mathcal{L}_{rd} = \cos(F(x), E_\theta(x)) = \frac{F(x)\cdot E_\theta(x)}{\|F(x)\|_2 \cdot \|E_\theta(x)\|_2}$$

final loss for 3D feature extractor:

$$\mathcal{L} = w_{con}\cdot\mathcal{L}_{con} + w_{rd}\cdot\mathcal{L}_{rd}$$

binary cross entropy loss for the normalcy classifier:

$$\mathcal{L}_{bce} = -\frac{1}{N}\sum_{i=1}^{N} \log(p(x_i|w))\cdot y_i + \log(1-p(x_i|w))\cdot(1-y_i)$$

Anomaly Scores

distance-based score:

$$f(x^{test,*}), f^* = \underset{f(x^{test}) \in f(\mathcal{X}^{test})}{\operatorname{argmax}} \underset{f \in \mathcal{M}}{\operatorname{argmin}} \|f(x^{test}) - f\|_2, \quad S^* = \|f(x^{test,*}) - f^*\|_2.$$

$$S_{dist}(X^{test}) = \left(1 - \frac{\exp\|f(x^{test,*}) - f^*\|_2}{\sum_{f \in \mathcal{N}_b(f^*)} \exp\|f(x^{test,*}) - f\|_2}\right) \cdot S^*$$

adversarial perturbation:

$$\tilde{x}^{test} = x^{test} + \eta(-\nabla_{x^{test}} \log(\hat{p}(x^{test}|w)))$$

classification-based score:

$$S_{cls}(\tilde{X}^{test}) = \max_{x^{test} \in \tilde{X}^{test}} p(\tilde{x}^{test}|w)$$

final score:

$$S(X^{test}) = w_d \cdot S_{dist}(X^{test}) + w_c \cdot S_{cls}(\tilde{X}^{test})$$

Quantitative Results

3DzAL outperforms the state-of-the-art methods on the MVTec 3D-AD dataset for the zero-shot 3D anomaly detection and localization tasks.

train\test	bagel	cable	carrot	cookie	dowel	foam	peach	potato	rope	tire	mean (3DzAL)	BTF	3DSR
bagel	-	78.6	91.9	89.3	81.6	48.3	91.2	96.5	82.0	86.7	**82.9** (↑ 2.4)	80.5	7.9
cable	31.5	-	87.1	48.6	81.0	56.6	65.4	89.4	81.8	80.7	**69.1** (↑ 0.9)	68.2	6.6
carrot	45.4	77.2	-	52.9	82.3	46.3	67.3	91.3	84.4	89.3	**70.7** (↑ 2.1)	68.6	21.7
cookie	70.6	76.8	91.5	-	81.0	46.5	82.9	91.7	84.4	89.2	**79.4** (↑ 5.5)	73.9	8.3
dowel	15.1	76.7	89.8	20.8	-	46.4	49.0	84.5	82.3	89.3	**61.5** (↑ 4.1)	57.4	35.4
foam	25.6	77.6	86.0	9.4	80.1	-	57.0	79.4	79.8	83.9	**64.3** (↑ 3.3)	61.0	0.5
peach	81.3	78.8	92.4	84.8	82.7	51.8	-	97.9	82.9	89.2	**82.4** (↑ 3.5)	78.9	14.2
potato	78.0	78.1	96.7	80.0	81.5	46.4	88.8	-	82.7	88.1	**80.0** (↑ 1.7)	78.3	13.2
rope	13.4	76.3	87.7	9.0	80.5	45.8	47.7	82.7	-	89.4	**59.2** (↑ 7.2)	52.0	19.3
tire	14.9	76.7	87.8	6.5	80.8	47.0	48.4	83.7	80.8	-	**58.5** (↑ 4.7)	53.8	23.8

Table 1. The detailed pixel-level AUPRO (%) of 3DzAL under the zero-shot setting. The best and second-best performances are highlighted in **bold** and underline (the gain of 3DzAL over the best baseline is also reported). 3DzAL outperforms BTF and 3DSR in all of the categories.

train\test	bagel	cable	carrot	cookie	dowel	foam	peach	potato	rope	tire	mean (3DzAL)	BTF	3DSR
bagel	-	57.9	71.8	68.9	57.5	58.1	56.1	69.0	47.3	55.9	**60.3** (↑ 6.9)	53.4	46.1
cable	52.5	-	52.3	49.6	51.6	72.4	46.4	48.2	44.2	59.9	**53.0** (↑ 2.9)	50.1	47.7
carrot	51.6	54.1	-	53.5	57.9	54.9	52.5	48.8	48.1	47.1	**52.1** (↑ 1.3)	50.8	46.0
cookie	40.2	50.0	55.5	-	55.3	60.1	46.0	47.3	35.9	59.6	**50.0** (↓ 0.6)	49.5	**50.6**
dowel	54.9	57.0	42.4	51.8	-	57.6	50.4	55.9	58.9	50.4	**53.3** (↑ 2.0)	51.3	47.3
foam	60.9	48.8	46.7	47.0	52.7	-	49.0	48.5	50.0	62.4	**51.8** (↓ 3.8)	49.9	**55.6**
peach	46.5	49.2	67.5	49.3	55.0	54.8	-	79.7	58.1	52.6	**57.0** (↑ 1.4)	55.6	45.0
potato	43.8	50.5	73.8	43.2	52.4	55.6	49.7	-	46.2	48.3	**51.5** (↓ 0.6)	**52.1**	49.7
rope	48.8	47.6	54.6	42.0	41.9	52.8	46.7	45.7	-	61.8	**49.1** (↓ 1.0)	49.1	**50.1**
tire	48.0	52.1	48.8	45.5	51.6	63.5	53.6	50.0	56.3	-	**52.2** (↑ 0.8)	50.5	51.4

Table 2. The detailed image-level AUROC of 3DzAL under the zero-shot setting. The best and second-best performances are highlighted in **bold** and underline (the gain of 3DzAL over the best baseline is also reported). 3DzAL outperforms BTF and 3DSR in most of the categories.

Combining both the adding-point and removing-point type in pseudo anomaly generation achieves the best performance. For each cell, the numbers correspond to the cases when the training class is bagel/potato/rope.

pseudo anomaly generation type		pixel-level	image-level
adding points	removing points	AUPRO (%)	AUROC (%)
✓	✗	80.3 / 79.3 / 58.8	53.8 / 52.6 / 49.1
✗	✓	81.0 / 79.6 / 58.9	54.0 / 52.8 / 49.1
✓	✓	**82.9 / 80.0 / 59.2**	**60.3 / 51.5 / 49.1**

The ablation on CNN weight initialization (WI) type shows that random-initialized CNN for inductive-bias-based pseudo anomaly generation leads to the best performance. For each cell, the numbers correspond to the cases when the training class is bagel/peach/rope.

method	pixel-level AUPRO (%)	image-level AUROC (%)
BTF (baseline)	80.5 / 78.9 / 52.0	53.4 / 55.6 / 49.1
3DzAL (pretrained CNN WI)	81.4 / 79.6 / 57.8	56.7 / 56.0 / 49.1
3DzAL (random CNN WI)	**82.9 / 82.4 / 59.2**	**60.3 / 57.0 / 49.1**

3DzAL also works for training data with multiple classes. For each cell, the first/second number is pixel-level AUPRO (%)/image-level AUROC(%).

training classes \ method	BTF	3DzAL
bagel + cable	80.3 / 53.4	**80.9** (↑ 0.6) / **53.7** (↑ 0.3)
carrot + cookie	79.4 / 50.8	**83.7** (↑ 4.3) / **55.4** (↑ 4.6)
dowel + foam	78.5 / 51.3	**80.3** (↑ 1.8) / **54.0** (↑ 2.7)