Towards Total Recall in Industrial Anomaly Detection (PatchCore, CVPR 2022)

2023. 01. 31

Presenter : Taewoong Kang
twk@deepnoid.com

Anomaly Detection

- Anomaly Detection
  - 정상 범주에서 벗어난 모든 것을 탐지(불량 검출, 이상 감지)

1-2-1. Point anomaly

첫 번째는 왼쪽 상단 그림의 point anomaly입니다.

하나의 점은 특정 사건, 거래, 이미지 한 장을 의미합니다.

따라서 point anomaly는 정상 범주에서 벗어난 특정 사건, 거래, 이미지 등을 의미합니다.

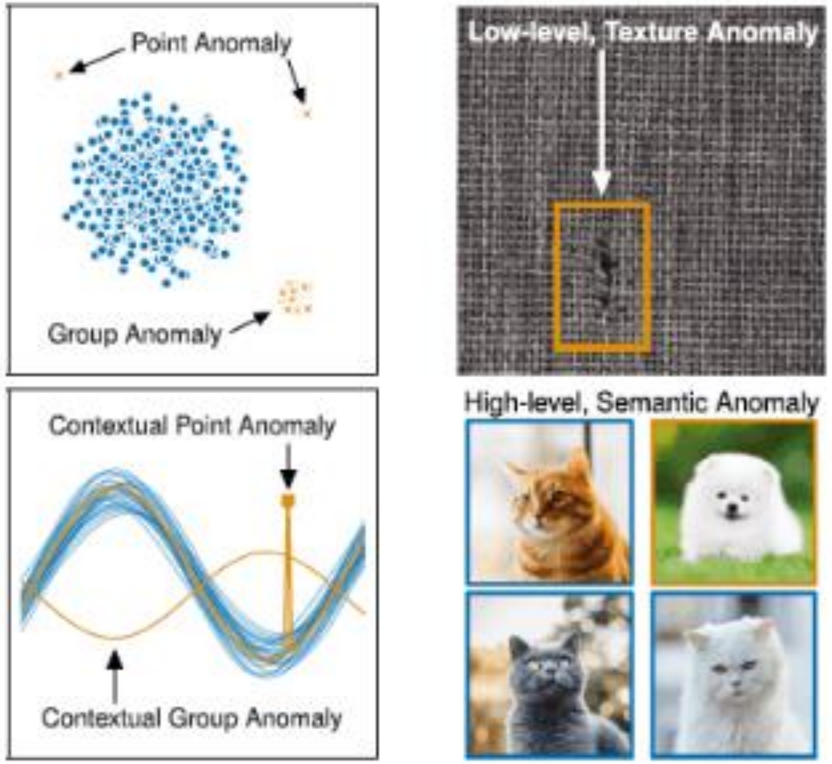
예를 들어 결재 시스템에서의 비정상적 거래, 제조 현장에서의 품질 불량 이미지 등이 여기에 해당합니다.

1-2-2. Contextual anomaly

두 번째는 왼쪽 하단 그림의 contextual anomaly입니다.

이는 point anomaly와 달리 한 점만으로는 판별할 수 없고 time series내에서 문맥을 통해 파악해야 하기 때문에 contextual anomaly라고 부릅니다.

예를 들어 주식 시장에서의 비 이상적 과열 현상, 이상 기온 현상 등이 여기에 해당합니다.



1-2-3. Group anomaly

다음은 왼쪽 그림들의 group anomaly입니다.

이는 하나의 포인트만으로는 판단할 수 없고 다량의 정보로 판단할 수 있는 anomaly를 의미합니다.

예를 들어 사이버 보안등이 여기에 해당합니다.

1-2-4. Low-level anomaly

다음은 오른쪽 상단 그림의 low-level anomaly입니다.

여기서 low-level이란 딥러닝 모델의 feature level을 의미합니다.

낮은 level의 feature는 보통 semantic 하지 않은 noise성 정보를 의미합니다.

따라서 texture 등, semantic하지 않은 특징을 갖는 anomaly 등을 의미합니다.

예를 들어 제조 환경에서의 품질검사 사례 등이 여기에 해당합니다.

1-2-5. High-level anomaly

마지막으로 오른쪽 아래 그림의 high-level anomlay입니다.

이는 low-level anomaly와는 반대로 semantic 한 의미를 갖는 anomaly를 의미합니다.

예를 들어 그림에서와 같이 '고양이'가 normal일 때, 다른 동물인 '개'가 high-level anomaly에 해당합니다.

Anomaly Detection

- Autoencoding Model
- GMM
- Generative Adversarial training
- Invariance towards predefined physical augmentations
- Robustness of hidden features to reintroduction of reconstructions
- Prototypical memory banks
- Attention-guidance
- Structural objectives
- Constrained representation spaces

Related Works

- SPADE
  - Utilizes memory banks with separate approaches for image- and pixel-level anomaly detection
  - PatchCore-> neighbourhood-aware patch-level features, coreset subsampled

- PaDiM
  - patch-level approach (use Mahalanobis distance measure)
  - PatchCore->use locally aware patch-feature scores

Abstract

- PatchCore
  - Anomaly Detection
  - Cold-start (One-class Classification)
  - Use pretrained network (WideResNet50-backbone)
  - On training stage, use only nominal dataset
  - Localization, Detection SOTA

Overview

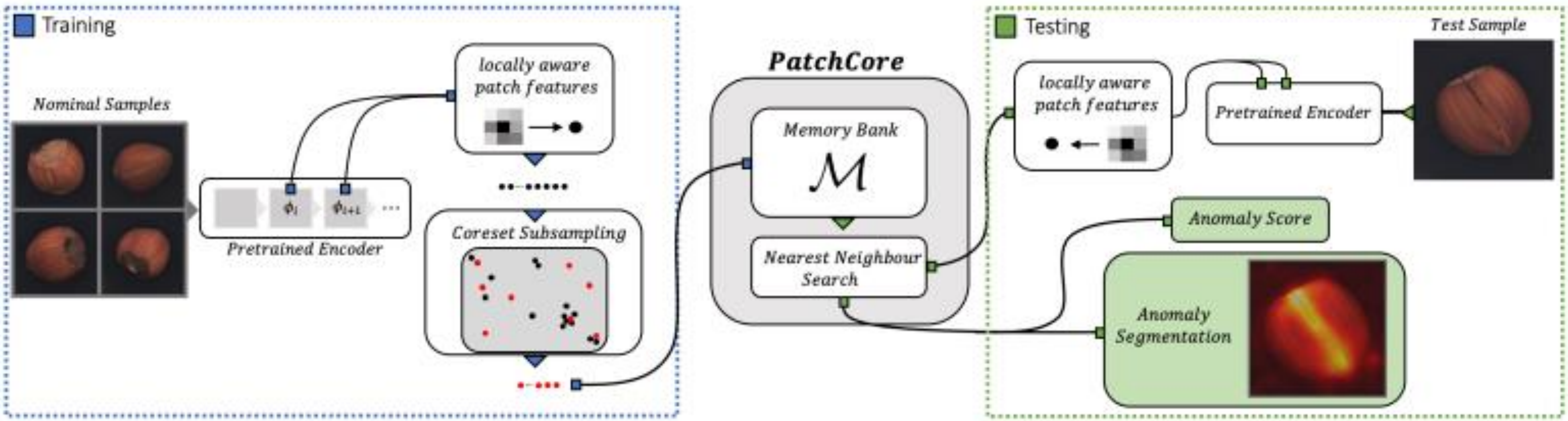


Figure 2. Overview of *PatchCore*. Nominal samples are broken down into a memory bank of neighbourhood-aware patch-level features. For reduced redundancy and inference time, this memory bank is downsampled via greedy coreset subsampling. At test time, images are classified as anomalies if at least one patch is anomalous, and pixel-level anomaly segmentation is generated by scoring each patch-feature.

DEEP NOID

Notation

- $\mathcal{X}_N$ set of all nominal images $\forall x \in \mathcal{X}_N : y_x = 0)$ $y_x \in \{0, 1\}$
- $\mathcal{X}_T$ set of samples provided at test time
- ϕ pretrained network (pretrained on Imagenet)
- $\phi_{i,j} = \phi_j(x_i)$, $x_i \in \mathcal{X}$ and hierarchy-level j of pretrained network
- $\phi_{i,j} \in \mathbb{R}^{c^* \times h^* \times w^*}$ feature map (depth , height, width)
- $\phi_{i,j}(h, w) = \phi_j(x_i, h, w) \in \mathbb{R}^{c^*}$

Locally aware patch features

- $$\mathcal{N}_p^{(h,w)} = \{(a,b) | a \in [h - \lfloor p/2 \rfloor, ..., h + \lfloor p/2 \rfloor], \\ b \in [w - \lfloor p/2 \rfloor, ..., w + \lfloor p/2 \rfloor]\},$$
- $$\phi_{i,j}\left(\mathcal{N}_p^{(h,w)}\right) = f_{\text{agg}}\left(\{\phi_{i,j}(a,b) | (a,b) \in \mathcal{N}_p^{(h,w)}\}\right)$$
- We use adaptive average pooling
- (= local smoothing, retains feature map resolution)
- $$\mathcal{P}_{s,p}(\phi_{i,j}) = \{\phi_{i,j}(\mathcal{N}_p^{(h,w)}) | \\ h, w \bmod s = 0, h < h^*, w < w^*, h, w \in \mathbb{N}\}$$
- Optional use of a striding parameters s, set to 1

DEEP NOID

Locally aware patch features

- PatchCore uses only two intermediate feature hierarchies
  - ->aggregating each elements with its corresponding patch feature at the lowest hierarchy level used
  - Avoiding features too generic or too heavily biased towards ImageNet classification

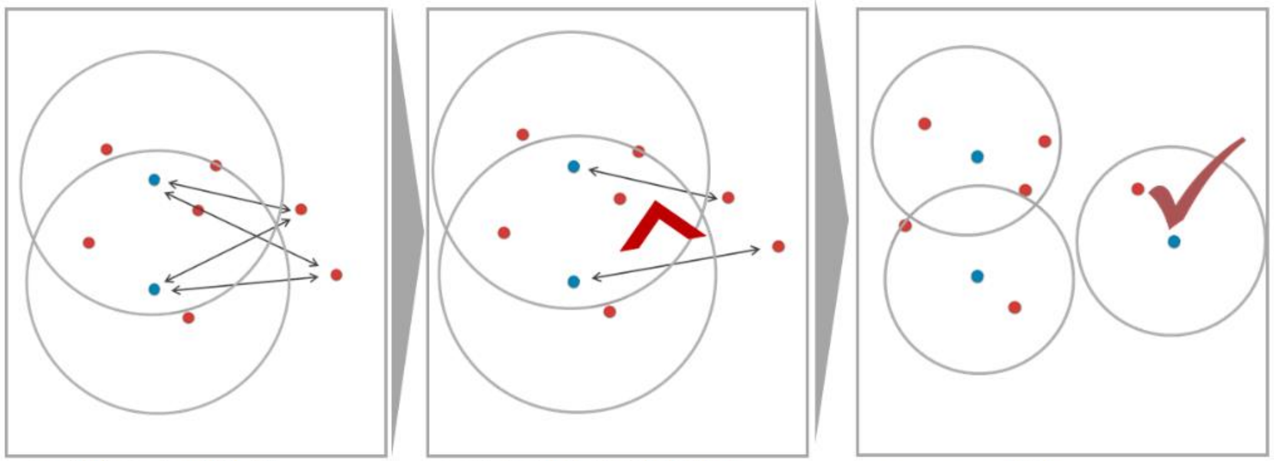
- PatchCore memory bank

$$\mathcal{M} = \bigcup_{x_i \in \mathcal{X}_N} \mathcal{P}_{s,p}(\phi_j(x_i))$$

Coreset-reduced patch-feature memory bank

- Use a coreset subsampling mechanism to reduce M
  - -> reduce Inference time while retaining performance

$$\mathcal{M}_C^* = \underset{\mathcal{M}_C \subset \mathcal{M}}{\arg\min} \max_{m \in \mathcal{M}} \min_{n \in \mathcal{M}_C} \|m - n\|_2$$



1. S^0의 Center거리 중 짧은 거리 찾기

2. 그 중 가장 긴 거리 찾기

3. Core-set 선정

Algorithm 1: *PatchCore* memory bank.

Input: Pretrained ϕ, hierarchies j, nominal data $\mathcal{X}_N$, stride s, patchsize p, coreset target l, random linear projection ψ.

Output: Patch-level Memory bank $\mathcal{M}$.

Algorithm:

$\mathcal{M} \leftarrow \{\}$

for $x_i \in \mathcal{X}_N$ **do**

  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{P}_{s,p}(\phi_j(x_i))$

end

/* Apply greedy coreset selection. */

$\mathcal{M}_C \leftarrow \{\}$

for $i \in [0, ..., l-1]$ **do**

  $m_i \leftarrow \underset{m \in \mathcal{M} - \mathcal{M}_C}{\arg\max} \min_{n \in \mathcal{M}_C} \|\psi(m) - \psi(n)\|_2$

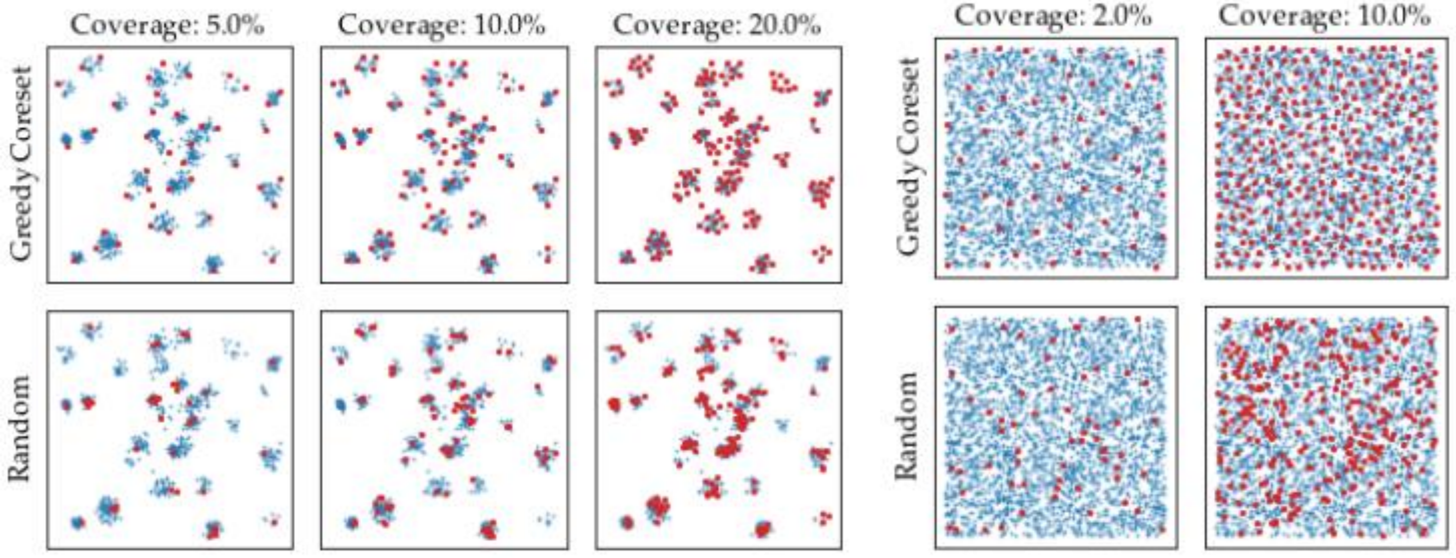
  $\mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{m_i\}$

end

$\mathcal{M} \leftarrow \mathcal{M}_C$

Coreset-reduced patch-feature memory bank

- Use a coreset subsampling mechanism to reduce M
  - -〉 reduce Inference time while retaining performance



(a) (b)

Anomaly Detection with PatchCore

- s : image-level anomaly score
- s^* : maximum distance score
- $$m^{\text{test},*}, m^* = \underset{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})}{\arg\max} \ \underset{m \in \mathcal{M}}{\arg\min} \left\| m^{\text{test}} - m \right\|_2$$
  $$s^* = \left\| m^{\text{test},*} - m^* \right\|_2 .$$

- if far from neighbouring samples and thereby an already rare nominal occurrence,
- Increase the anomaly score

- $$s = \left(1 - \frac{\exp \| m^{\text{test},*} - m^* \|_2}{\sum_{m \in \mathcal{N}_b(m^*)} \exp \| m^{\text{test},*} - m \|_2}\right) \cdot s^*$$

Anomaly detection on MVTec AD

- AUROC
  - Area Under the Receiver Operator Curve
  - (Detection)
- PRO
  - Per Region Overlap
  - (Localization)
- F1 score
  - 2 *Recall *Precision/ (recall +precision)
  - Recall = TP/TP+FN, Precision = TP/TP+FP

Table 4. PatchCore-1% with higher resolution/larger backbones/ensembles. The coreset subsampling allows for computationally expensive setups while still retaining fast inference.

Metric→	AUROC	pwAUROC	PRO
DenseN-201 & RNext-101 & WRN-101 (2+3), Imagesize 320			
Score ↑	**99.6**	98.2	94.9
Error ↓	**0.4**	1.8	5.6
WRN-101 (2+3), Imagesize 280			
Score ↑	99.4	98.2	94.4
Error ↓	0.6	1.8	5.6
WRN-101 (1+2+3), Imagesize 280			
Score ↑	99.2	**98.4**	**95.0**
Error ↓	0.8	**1.6**	**5.0**

Table 1. Anomaly Detection Performance (AUROC) on MVTec AD [5]. PaDiM* denotes a result from [14] with problem-specific backbone selection. The total count of misclassifications was determined as the sum of false-positive and false-negative predictions given a F1-optimal threshold. We did not have individual anomaly scores for competing methods so could compute this number only for *PatchCore*.

Method	SPADE [10]	PatchSVDD [56]	DifferNet [42]	PaDiM [14]	Mah.AD [40]	PaDiM* [14]	PatchCore−25%	PatchCore−10%	PatchCore−1%
AUROC ↑	85.5	92.1	94.9	95.3	95.8	97.9	**99.1**	99.0	99.0
Error ↓	14.5	7.9	5.1	4.7	4.2	2.1	**0.9**	1.0	1.0
Misclassifications ↓	-	-	-	-	-	-	**42**	47	49

Table S1. Anomaly Detection Performance (AUROC) on MVTec AD [5]. PaDiM* denotes a result from [14] with a backbone specifically selected for the task of image-level anomaly detection, which we could not reproduce.

↓ Method \Dataset →	**Avg**	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
GeoTrans [20]	67.2	74.4	78.3	67.0	43.7	61.9	35.9	84.1	81.3	63.0	50.0	41.7	97.2	86.9	61.1	82.0
GANomaly [2]	76.2	89.2	75.7	73.2	69.9	70.8	78.5	84.2	70.0	74.3	74.6	79.4	65.3	79.2	83.4	74.5
DSEBM [58]	70.9	81.8	68.5	59.4	41.3	71.7	76.2	41.6	67.9	80.6	99.9	69.0	78.1	74.1	95.2	58.4
OCSVM [3]	71.9	99.0	80.3	54.4	62.7	41.0	91.1	88.0	61.1	72.9	74.7	87.6	61.9	56.7	95.3	51.7
ITAE [25]	83.9	94.1	83.2	68.1	70.6	88.3	85.5	86.2	66.7	78.6	**100**	73.5	**100**	84.3	92.3	87.6
SPADE [10]	85.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CAVGA-R_w [52]	90	96	92	93	88	84	97	89	82	86	81	97	89	99	79	96
PatchSVDD [56]	92.1	98.6	90.3	76.7	92.9	94.6	92.0	90.9	94.0	86.1	81.3	97.8	**100**	91.5	96.5	97.9
DifferNet [42]	94.9	99.0	95.9	86.9	92.9	84.0	99.3	97.1	96.1	88.8	96.3	99.4	98.6	91.1	**99.8**	95.1
PaDiM [14]	95.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MahalanobisAD [40]	95.8	**100**	95.0	95.1	**100**	89.7	99.1	**100**	94.7	88.7	85.2	**99.8**	96.9	95.5	99.6	97.9
PaDiM* [14]	97.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchCore−25	**99.1**	**100**	**99.5**	**98.1**	**98.7**	98.2	**100**	**100**	**100**	96.6	98.1	98.7	**100**	**100**	99.2	99.4
PatchCore−10	99.0	100	99.4	97.8	98.7	97.9	**100**	**100**	**100**	96.0	97.0	98.9	99.7	**100**	99.0	**99.5**
PatchCore−1	99.0	100	99.3	98.0	98.0	**98.6**	**100**	**100**	99.7	**97.0**	96.4	99.4	**100**	99.9	99.2	99.2

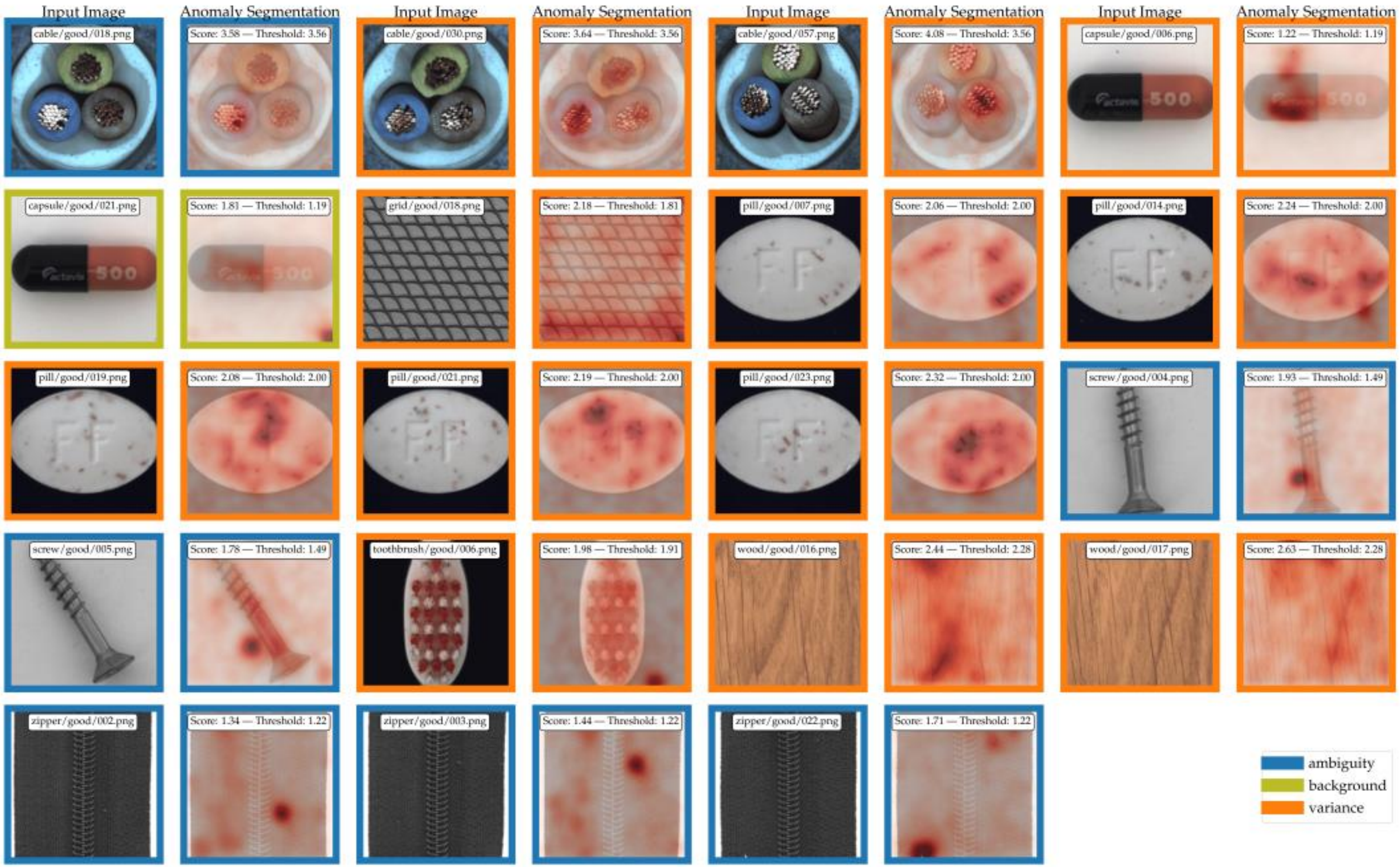


Figure S1. Visualization of remaining false positive classifications (under F1-optimal thresholding). Colors denote different error sources. Orange denotes high degrees of nominal variance mistaken for anomalies, blue denotes misclassifications due to anomalies in the labelling context and olive denotes variance in the background mistaken for anomalous content.

Anomaly detection on MVTec AD

Table 2. Anomaly Segmentation Performance (pixelwise AUROC) on MVTec AD [5].

Method	AE_{SSIM} [5]	γ-VAE + grad. [15]	CAVGA-R_w [52]	PatchSVDD [56]	SPADE [10]	PaDiM [14]	PatchCore−25%	PatchCore−10%	PatchCore−1%
AUROC ↑	87	88.8	89	95.7	96.0	97.5	**98.1**	**98.1**	98.0
Error ↓	13	11.2	11	4.3	4.0	2.5	**1.9**	**1.9**	2.0

Table 3. Anomaly Detection Performance on MVTec AD [5] as measured in PRO [%] [5, 10].

Method	AE_{SSIM} [5]	Student [6]	SPADE [10]	PaDiM [14]	PatchCore−25%	PatchCore−10%	PatchCore−1%
PRO ↑	69.4	85.7	91.7	92.1	93.4	**93.5**	93.1
Error ↓	30.6	14.3	8.3	7.9	6.6	**6.5**	6.9

Inference Time

- Lower Inference time
- Higher Performance

Table 5. Mean inference time per image on MVTec AD. Scores are (image AUROC, pixel AUROC, PRO metric).

Method	*PatchCore*−100%	*PatchCore*−10%	*PatchCore*−1%
Scores	(99.1, 98.0, 93.3)	(99.0, 98.1, 93.5)	(99.0, 98.0, 93.1)
Time (s)	0.6	0.22	0.17

Method	*PatchCore*−100% + IVFPQ	SPADE	PaDiM
Scores	(98.0, 97.9, 93.0)	(85.3, 96.6, 91.5)	(95.4, 97.3, 91.8)
Time (s)	0.2	0.66	0.19

D E E P
N O I D

Locally aware patch-features and hierarchies

- Neighbourhood size p = 3
- Hierarchy level -〉 2+3

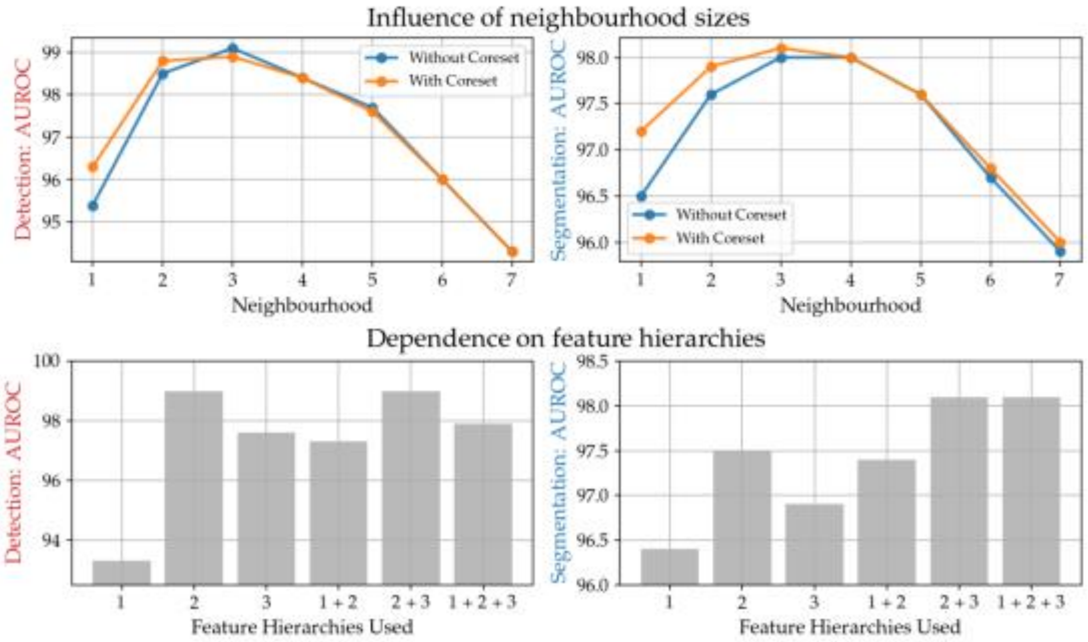


Figure 4. Local awareness and network feature depths vs. detection performance. PRO score results in the supplementary.

DEEP NOID

Coreset-reduced patch-feature memory bank

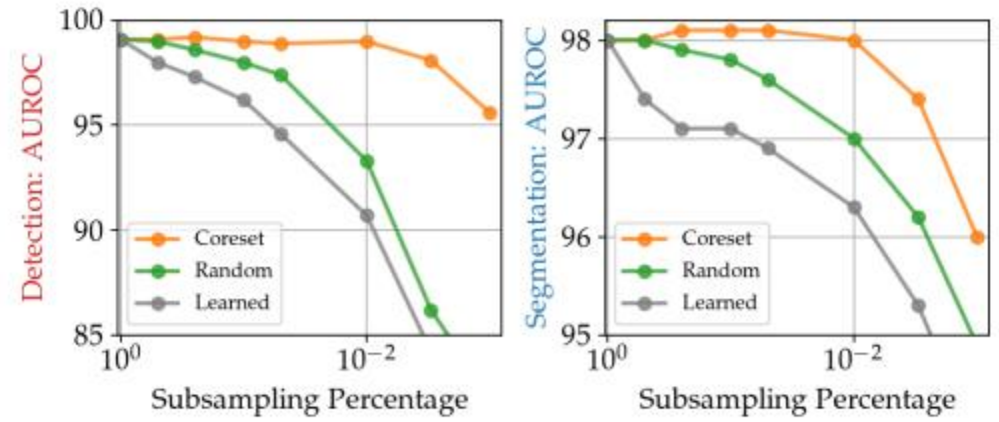


Figure 5. Performance retention for different subsamplers, results for PRO score in the supplementary.

$$\mathcal{L}_{\text{rec}}(m_i) = \left\| m_i - \sum_{p_k \in \mathcal{P}} \frac{e^{\|m_i - p_k\|_2}}{\sum_{p_j \in \mathcal{P}} e^{\|m_i - p_j\|}} p_k \right\|_2^2$$

$p_i \in \mathcal{P} \subset \mathbb{R}^d$ with $|\mathcal{P}| = p_{\text{target}} \cdot |\mathcal{M}|$

DEEP NOID

Low shot anomaly detection

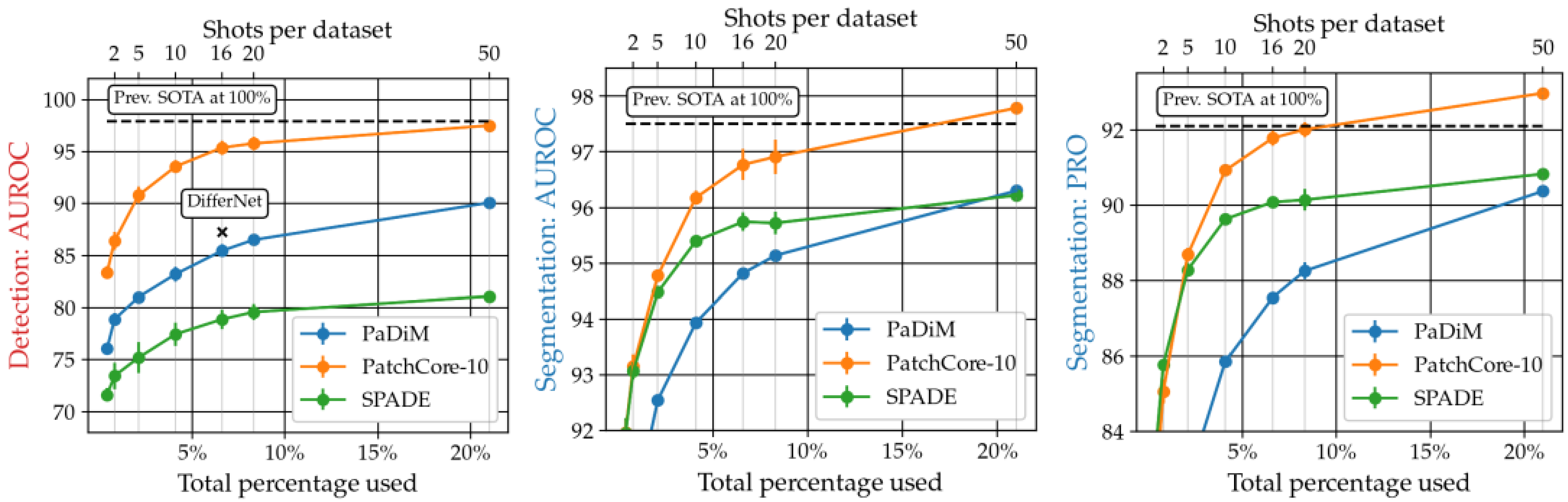


Figure 6. *PatchCore* shows notably higher sample-efficiency than competitors, matching the previous state-of-the-art with a fraction of nominal training data. Note that PaDiM and SPADE where reimplemented with WideResNet50 for comparability.

Conclusion

- A SOTA cold-start image anomaly detection and localization system with low computational cost on industrial anomaly detection benchmarks
- On MVTec, achieve AUROC over 99% with highest sample efficiency

Limitations

- Applicability is generally limited by the transferability of the pretrained features leveraged

Q & A

감사합니다