VDSL 세미나

Patch-wise Anomaly Detection

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
석사과정 이정훈

2022. 4. 8.

Outline

- Introduction
  - What is Anomaly?
- Patch SVDD
  - Patch-level SVDD for Anomaly Detection and Segmentation (ACCV 2020)
- Multi-Scale Patch-Based Learning
  - Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation (WACV 2022)
- PatchCore
  - Towards Total Recall in Industrial Anomaly Detection (arXiv 2021)
- Conclusion

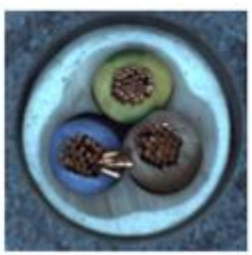
Introduction

- What is Anomaly?
  - Outlier
    - 데이터의 전체적인 패턴에서 벗어난 관측치
  - Novelty
    - 본질적인 데이터는 같지만 유형이 다른 관측치
  - Anomaly
    - 대부분의 데이터와 본질적인 특성이 다른 관측치 전혀 다른 방식으로 생성되었을 것으로 추정되는 관측치

- Why interested in Anomaly?
  - 제조산업에서 생산된 제품 내 결함은 상품 가치를 떨어뜨림
  - Anomaly(i.e. defect)가 하나라도 존재한다면 상품성이 없는 것인가?
    - 그러면, 어느 정도의 anomaly가 수용 가능한 범위(수준) 인가?
      - product specification needs to be met based on customer needs

Introduction

- What is Anomaly Detection and Segmentation?
  - Anomaly detection
    - Binary classification problem to determine whether an input contains an anomaly
    - Usually formulated as a one-class classification because abnormal examples are either inaccessible or insufficient to model distribution during the training
  - Anomaly segmentation
    - On image data, detected anomalies can also be localized, and anomaly segmentation problem is to localize the anomalies at the pixel level

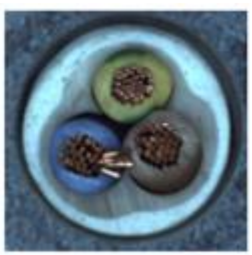


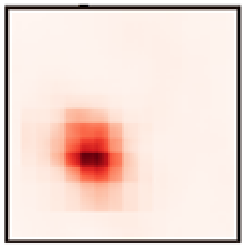
normal

abnormal

input 이미지 내 이상치 포함 여부를 판단

[Anomaly Detection]





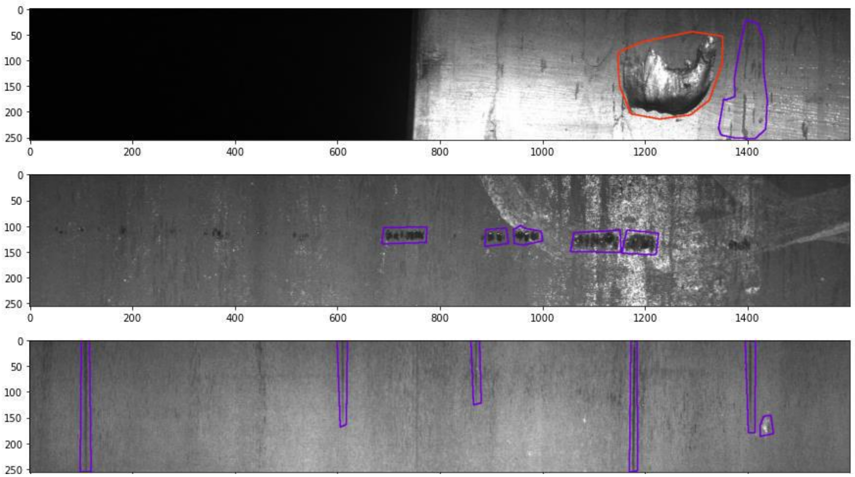
input 이미지 내 pixel-level 비정상 여부를 탐지

[Anomaly Segmentation]

Introduction

- Real World Application
  - In iron & steel making industry
    - In the process of transforming raw materials into steel plate, impurities or defects can be surfaced by the process of milling
    - Since it is hard to find impurities inside the plate, every plate goes through the inspection process to sort out the product that are out of specification

[Real-time surface inspection system]



[Kaggle competition* - Can you detect and classify defects in steel?]

*https://www.kaggle.com/competitions/severstal-steel-defect-detection/overview

Introduction

- Anomaly 연구 트렌드
  - Reconstruction 방식
    - GAN, AE 등을 사용하여 normal data를 reconstruction 하도록 network 학습
    - Test 단계에서 reconstruction loss가 클수록 anomalous 하다고 판단
  - Classification 방식
    - One-class classification 방식으로 normal feature 들이 한 점에 모이도록 hypersphere 학습
    - Test 단계에서는 train feature의 중심과 멀리 있을수록 anomalous 하다고 판단
  - Feature matching 방식 ★
    - Feature의 distance 또는 확률 분포 기반으로 anomal을 판단
    - pre-trained model 사용, 추출된 feature와 얼마나 멀리 떨어져 있는지로 anomaly score 측정
  - Probablistic 방식 ★
    - Normalizing Flow를 사용하여 image → gaussian으로 가는 invertible 함수를 학습
    - Test 이미지에 대해 직접적으로 확률 값을 예측

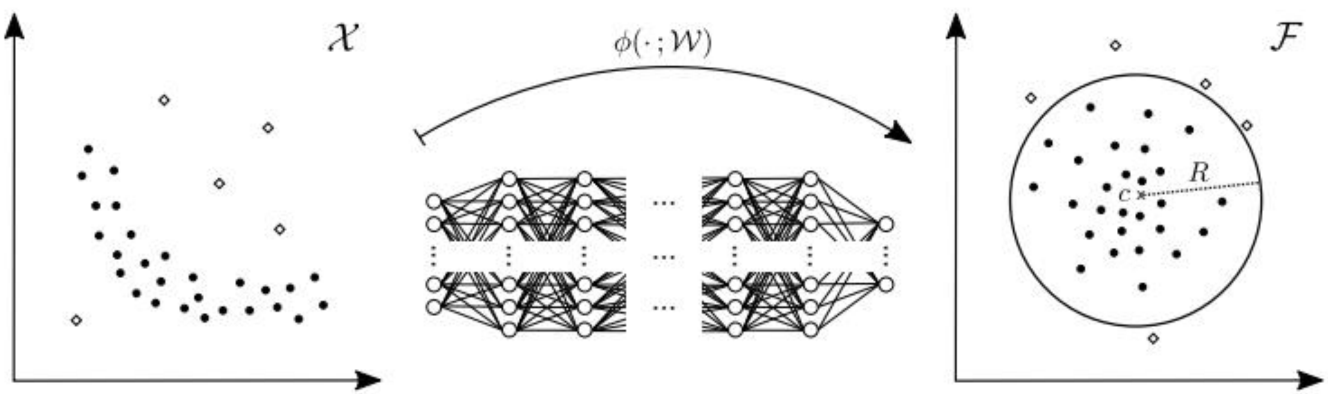
MVTec AD Benchmark

Rank	Model	Detection AUROC	Segmentation AUROC	Overall AUC	Segmentation AUPRO	Extra Training Data	Paper	Code	Result	Year	Tags
1	Fastflow	99.4	98.5			✓	FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows			2021	Transformer ResNet
2	PatchCore	99.1	98.1			✓	Towards Total Recall in Industrial Anomaly Detection			2021	ResNet
3	CS-Flow	98.7				✓	Fully Convolutional Cross-Scale-Flows for Image-based Defect Detection			2021	EfficientNet
⋮											
6	CFLOW-AD	98.26	98.62		94.6	✓	CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows			2021	
⋮											
13	AnoSeg	96	97			×	AnoSeg: Anomaly Segmentation Network Using Self-Supervised Learning			2021	
⋮											
22	Patch-SVDD	92.1	95.7			×	Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation			2020	

Patch SVDD

1) DMJ Tax et al. "Support vector data description" (Machine learning 2004)
2) Lukas Ruff et al. "Deep One-Class Classification" (PMLR 2018)

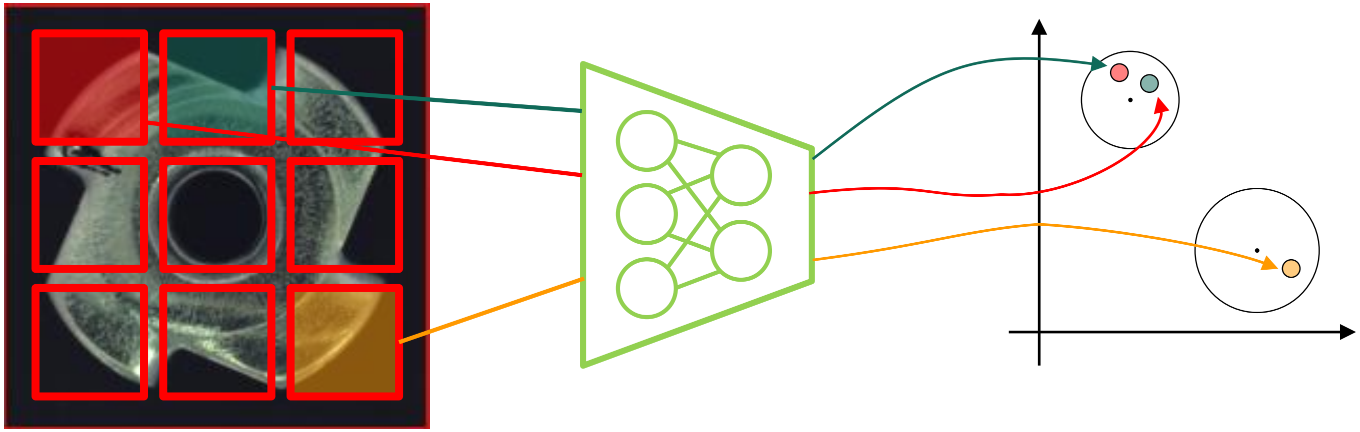
- Patch-level SVDD for Anomaly Detection and Segmentation
  - What is SVDD (Support Vector Data Description) [1]
    - SVDD is a classic algorithm used for one-class classification
    - SVDD searches for a data enclosing hypersphere of smallest size in the feature space
  - What is Deep SVDD [2]
    - 이미지를 input으로 활용하여 정상 데이터끼리 유사하게 임베딩 되도록 학습
    - 기존 kernel-based SVDD와 다르게 딥러닝을 기반으로 학습한 feature space에서 정상 데이터를 둘러싸는 가장 작은 구를 찾고, 해당 경계면을 기반으로 이상치를 탐지함



Patch SVDD

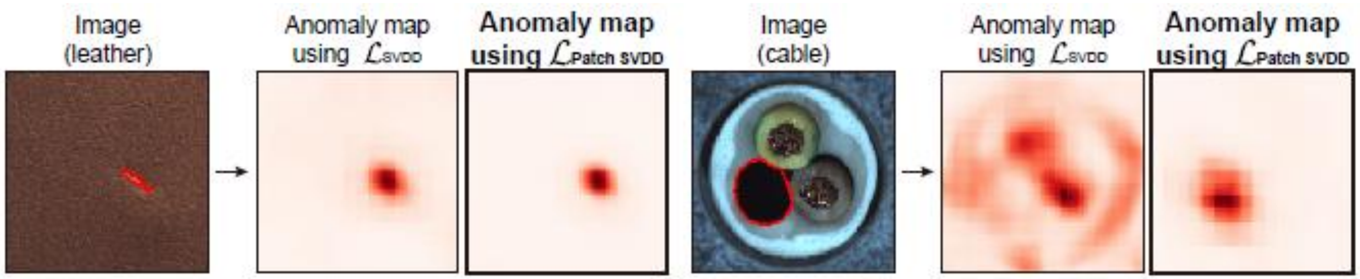
1) Jihun Yi et al. “Patch-level SVDD for Anomaly Detection and Segmentation” (ACCV 2020)

- Patch-level SVDD for Anomaly Detection and Segmentation [1]
  - What is the advantage of patch-wise detection?
    - Deep SVDD to Patch-wise Deep SVDD
    - Self-supervised representation learning
    - Hierarchical encoding

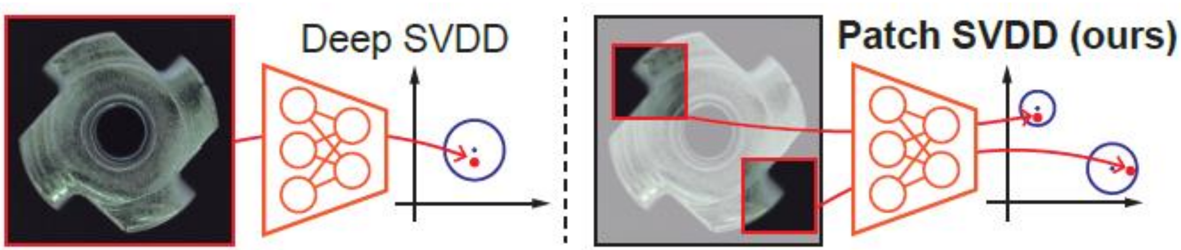


Patch SVDD

- Patch-level SVDD for Anomaly Detection and Segmentation
  - Patch SVDD is an extension of Deep SVDD to a patch-wise detection method
    - With a relatively simple image, SVDD localizes the defect well
    - By contrast, the detection performance of SVDD is poor for the images with high complexity

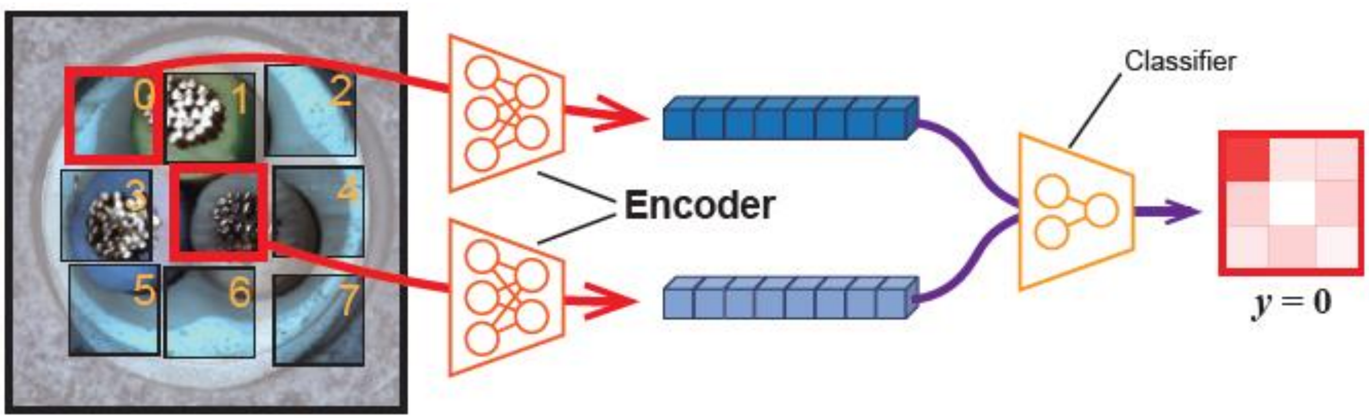


    - Patch SVDD performs inspection on every patch to localize the positions of defects
      - Since patches have high intra-class variation, Patch SVDD maps the features of dissimilar patches to multiple centers and allocates patches to each center



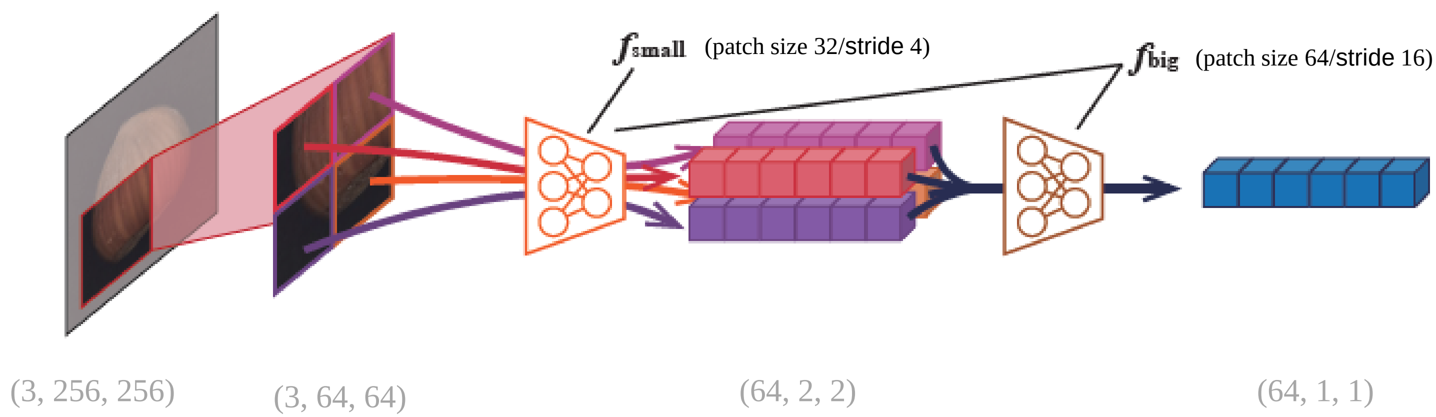
Patch SVDD

- Patch-level SVDD for Anomaly Detection and Segmentation
  - Self-supervised representation learning
    - Encoder가 feature extractor로서의 역할을 더 잘 수행하기 위해 SSL 적용
    - 이미지 내 임의의 patch p1 생성 후 3x3 grid 내 8개의 이웃하는 patch, $y = \{0, \ldots, 7\}$, 중 랜덤하게 하나를 선택하여 patch 2로 지정, classifier가 position을 예측하도록 학습
      - The size of the patch is the same as the receptive field of the encoder



Patch SVDD

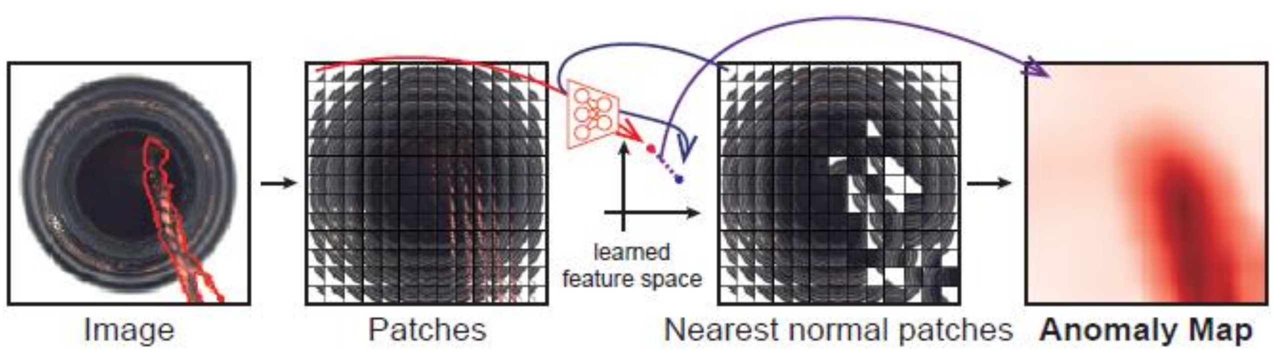
- Patch-level SVDD for Anomaly Detection and Segmentation
  - Hierarchical Encoding
    - 크기가 다른 anomaly를 잘 찾아내기 위해, 여러 개의 receptive field로 encoding 수행
    - Input patch p를 2x2 grid로 나누고, 각각을 f_{small}을 통해 embedding
    - Embedding을 다시 concatenate 시킨 후 f_{big}을 통해 한번 더 embedding 시킴



Patch SVDD

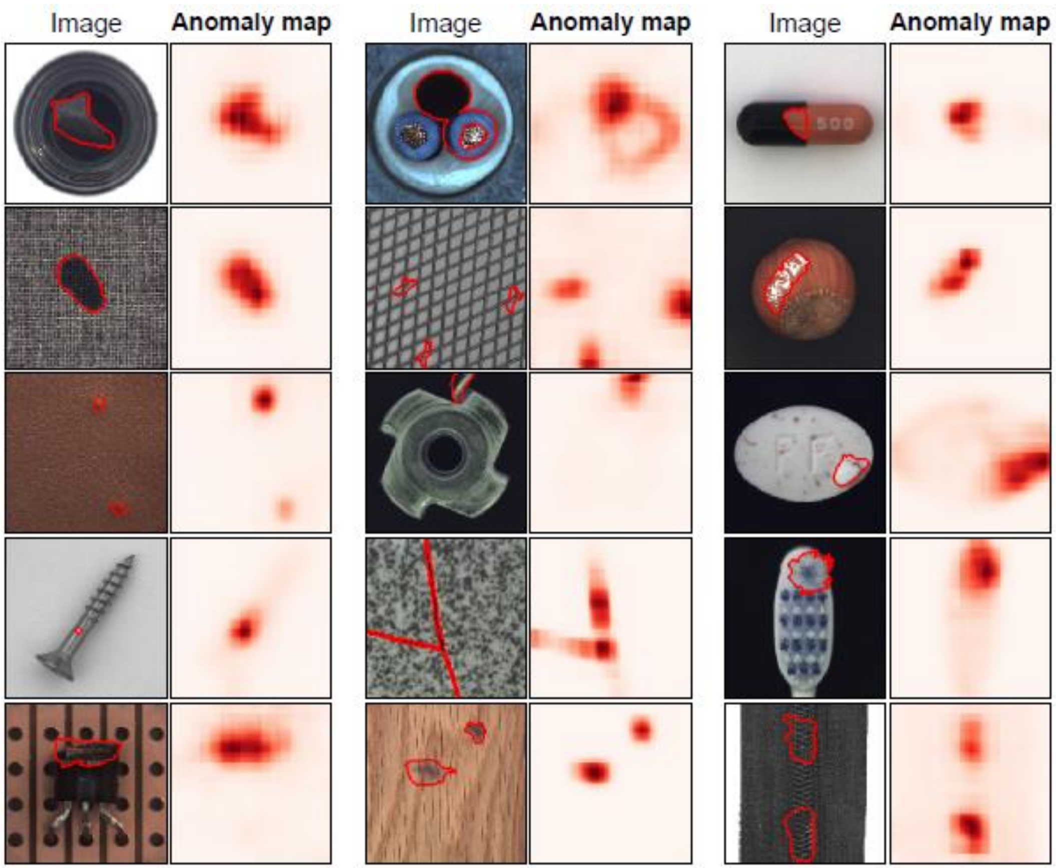
*NGT : Neighborhood Graph and Tree for Indexing High-dimensional Data (https://github.com/yahoojapan/NGT)

- Patch-level SVDD for Anomaly Detection and Segmentation
  - Generating anomaly map
    - 학습된 encoder로 train/test data에 대해 patch embedding을 추출
      - Patch(size: 64)로 sliding 하며(stride: 16) embedding map 생성
    - 전체 train data의 patch embedding 중에서 test patch embedding과 유사한 것을 탐색
      - For nearest neihgbor search, NGT* is used (0.48 sec per image)
    - Feature space 상의 nearest normal patch 까지의 L2 distance로 anomaly score 정의
      - Patch-wise calculated anomaly scores are then distributed to the pixels (decoding 과정)
        - ✓ Pixels receive the average anomaly scores of every patch to which they belong
    - f_{big}, f_{small}을 통해 생성된 anomaly map들은 성분곱을 통해 aggregate 후 anomaly map 생성



Patch SVDD

- Patch-level SVDD for Anomaly Detection and Segmentation
  - Results (MVTec AD dataset)



Classes	Patch SVDD Det.	Patch SVDD 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**

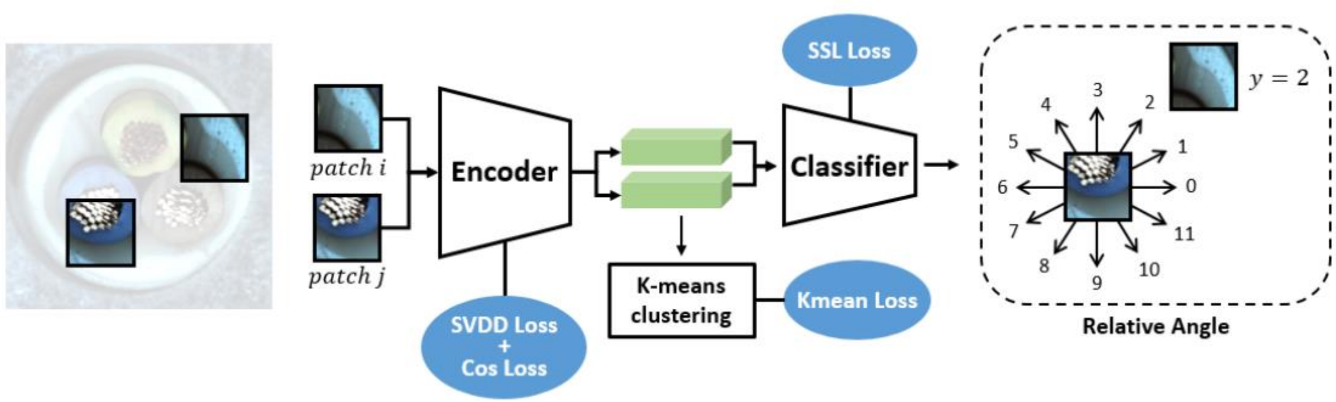
Why introduce Multi-Scale Patch-Based Learning?

- Patch SVDD recap
  - Patch SVDD enables SVDD to do anomaly segmentation and at the same time improves the anomaly detection performance significantly
    - Uses SVDD loss to train an encoder to gather semantically similar patches and makes the embeddings of adjacent patches still distinguishable enough by adopting the SSL method
    - Mapping the features of adjacent patches together benefits the patches with similar structures
      - predicting relative position of two patches can be confusing by cases with texture images
    - To overcome, Patch SVDD proposed to increase the weight of SVDD loss for texture images and decrease it for object images
    - Unfortunately, information extracted from the images tends to be insufficient by only adjusting the weights of losses

- Extension of Patch SVDD → Multi-Scale Patch-Based Learning
  - Multi-scale patch-based architecture for different levels of representation learning
  - New loss function for better feature representation learning

1) Chin-Chia Tsai et al. "Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation" (WACV 2022)

- Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation [1]
  - Multi-scale patch-based architecture for different levels of representation learning
    - Framework is mainly composed of 3 encoders with different architectures and 3 classifiers
      - 3 encoders : Enc_{64}, Enc_{32}, Enc_{16} + 3 classifiers
    - 각각의 encoder에서 추출된 feature 들은 normal pattern의 feature representation을 더 잘 학습하기 위해 k-means 적용, 유사 pattern을 가진 patch들을 cluster 하고 classifier는 두 neighboring patch 간 상대적 각도를 예측하도록 학습됨
      - 이미지에서 64, 32, 16 size의 패치 두 개를 각각 선택 후 Enc_{64}, Enc_{32}, Enc_{16} 각각 통과



Multi-Scale Patch-Based Learning

- Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation
  - New loss function(Cos loss & Kmean loss) besides SVDD loss and SSL loss
    - Cos loss : expect the two patches with larger distance to be semantically less similar
      - Add the Cos loss to strengthen the information extracted from the patches

$$\mathcal{L}_{\text{Cos}} = \sum_{(i,j,k)\in T} Sim\_cos(Enc_\theta(P_i), Enc_\theta(P_k)) - Sim\_cos(Enc_\theta(P_i), Enc_\theta(P_j)),$$

    - Kmeans loss : to gather patches with similar patterns
      - Embeddings of the patches in the same cluster are expected to be closer to the center of that cluster as much as possible (k = 50 적용 시 가장 좋은 성능을 보임)

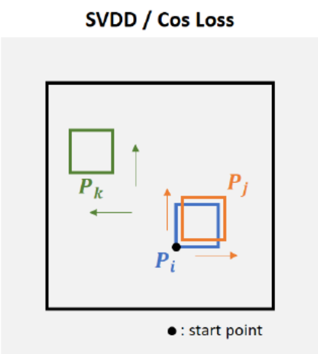
$$\mathcal{L}_{\text{Kmeans}} = \sum_{r} \min_{k} \|Enc_\theta(P_r) - c_k\|_2,$$

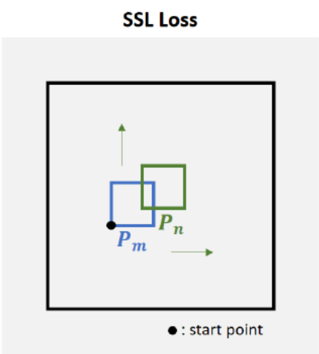
    - Overall loss :

$$\mathcal{L}_{\text{all}} = \lambda(L_{\text{SVDD}} + L_{\text{Cos}}) + L_{\text{SSL}} + L_{\text{Kmeans}},$$

Multi-Scale Patch-Based Learning

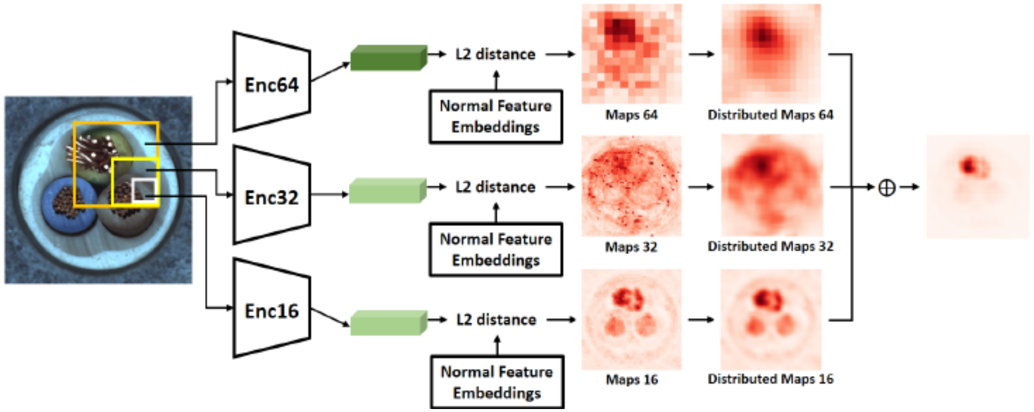
- Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation
  - Patch Selection
    - SVDD loss & Cos loss
      - Randomly select a patch P_i, P_j is randomly selected from the ±4 pixels adjacent patches to P_i for the patch size 64 (±2 for patch size 32, ±1 for 16)
      - P_k is randomly selected from the patches which are further away from P_i
    - SSL loss
      - Randomly select a patch P_m first, then select a P_n nearly a patch size away from P_m along a direction randomly selected from the 12 pre-defined angles {0◦, 30◦, 60◦, ..., 330◦}
    - Kmeans loss
      - Selects patches P_r from the image randomly
    - For the whole training process, the total number of patches, patch pairs, or patch triplets selected for each loss is set to 100 for each image





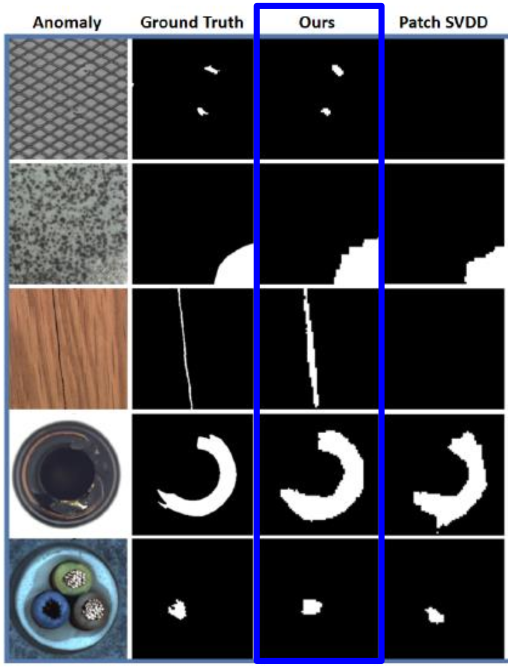
Multi-Scale Patch-Based Learning

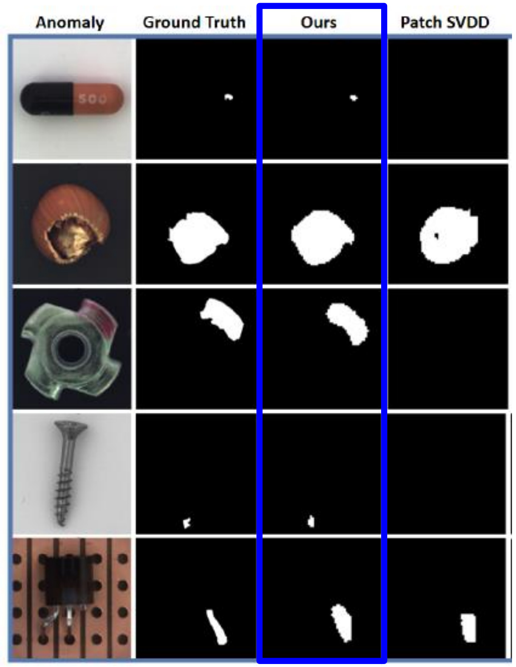
- Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation
  - Inference Stage
    - Test 이미지를 patch size가 64, 32, 16인 patch가 overlap 되게 한 후 split 하고, 이 patch들로부터 학습된 encoder를 활용 feature 추출
      - 각 patch 별 추출된 feature embedding과 training dataset의 normal feature embedding 사이의 shortest L2-distance를 계산하여 test 이미지의 abnormality 판단
    - Anomaly Segmentation
      - Patch-wise로 계산된 anomaly score를 overlapped된 pixel에 배포하여 각 patch size에 대한 anomaly map을 생성하고, element-wise addition을 통해 세 개의 anomaly map을 aggregate 시킴



Multi-Scale Patch-Based Learning

- Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation
  - Inference Stage (Anomaly Detection)
    - Anomaly map에 window size 8x8 and stride 4의 sliding window를 활용. 각 window 내 score를 평균하여 local score를 판단하고, local score의 maximum 값을 test 이미지의 final anomaly score로 정함
  - Results (MVTec AD dataset) : Detection 98.1%, Segmentation 98.1%





PatchSVDD
- Detection 92.1%
- Segmentation 95.7%

1) Karsten Roth et al. "Towards total recall in industrial anomaly detection" (arXiv 2021)

- Towards Total Recall in Industrial Anomaly Detection [1]
  {온전한 기억회복, 전부다 기억남}
  - Introduction
    - Recent development on anomaly detection is utilizing the pre-trained ImageNet model
    - Use ImageNet classification model without tuning, for learning common deep representation of target distribution
      - Fine-grained defect segmentation은 high resolution feature를 사용
      - 반면, full image-level anomaly detection은 higher abstraction level을 주로 사용
      - ImageNet으로 trained 된 모델의 higher level feature는 industrial environment에서 요구하는 abstract feature와 유사하지 않음
        - ✓ ImageNet trained very deep and abstract features concentrate on natural image classification
  - PatchCore introduces below method
    - Locally aware patch feature - memory bank of patch-level features
    - Coreset-reduced patch feature memory bank
    - Anomaly Detection with the nominal patch-feature memory bank

PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Locally aware patch feature
    - PatchCore uses a network ϕ pre-trained on ImageNet
      - Using the last level representation loses more localized nominal(non-defective) information
        - ✓ Anomalies encountered at test time are not known a priori, this becomes detrimental to the downstream anomaly detection performance
      - Deep and abstract features are biased towards the task of natural image classification
        - ✓ Has only little overlap with the industrial anomaly detection
    - Use a memory bank of patch-level features comprising mid-level feature representations
      - Exploits ResNet-like architectures (e.g. ResNet-50) with hierarchy level j $\in$ {1, 2, 3, 4}

$$\phi_{i,j} \in \mathbb{R}^{c^* \times h^* \times w^*} \qquad \phi_{i,j}(h,w) = \phi_j(x_i, h, w) \in \mathbb{R}^{c^*}$$

- ✓ Pixel coordinate (h, w)를 넣어주면, c^* 만큼의 feature vector 가 나옴
- ✓ 하나의 vector에 대해서 feature vector를 뽑는게 아니라 neighborhood 라는 개념을 도입

PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Locally aware patch feature – neighborhood
    - Neighborhood 에서 feature를 뽑게 되면, feature tensor 가 나오게 되며 이를 patch 형태로 관리

$$\mathcal{N}_p^{(h,w)} = \{(a,b) | a \in [h - \lfloor p/2 \rfloor, ..., h + \lfloor p/2 \rfloor], \\ b \in [w - \lfloor p/2 \rfloor, ..., w + \lfloor p/2 \rfloor]\}$$

- h,w를 중심에 두고 patch size(p) 만큼의 neighborhood 가 그려짐
- e.g. 3x3 neighborhood → feature network 에 넣으면, 3x3xd 개의 feature vector 가 나옴

$$\phi_{i,j}\left(\mathcal{N}_p^{(h,w)}\right) = f_{\text{agg}}\left(\{\phi_{i,j}(a,b) | (a,b) \in \mathcal{N}_p^{(h,w)}\}\right)$$

- 이를 합쳐서 f_{agg} 수행, adaptive average pooling을 활용해서 나오는 feature channel의 개수를 제어
  - ✓ Adaptive average pooling* has local smoothing effect over each individual feature map

*Adaptive average pooling is simply an average pooling operation that, given an input and output dimensionality, calculates the correct kernel size necessary to produce an output of the given dimensionality from the given input

PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Locally aware patch feature - memory bank of patch-level features
    - 다시 말해, 3x3 neighborhood → feature network 에 넣으면 3x3xd 개의 feature tensor, 전체 이미지에 대해서 feature tensor를 다 구해준 것을 Patch 라고 함
      - For a feature map tensor $\phi_{i,j}$, its locally aware patch-feature collection $P_{s,p}(\phi_{i,j})$ is

$$\mathcal{P}_{s,p}(\phi_{i,j}) = \{\phi_{i,j}(\mathcal{N}_p^{(h,w)})| \\ h, w \bmod s = 0, h < h^*, w < w^*, h, w \in \mathbb{N}\}$$

    - To retain the generality of used features as well as the spatial resolution, two intermediate feature hierarchies j and j+1 are used (i.e. 2 & 3)

    - 마지막으로, patch를 all nominal sample 대해 다 구해주게 되면 이를 memory bank 라고 함

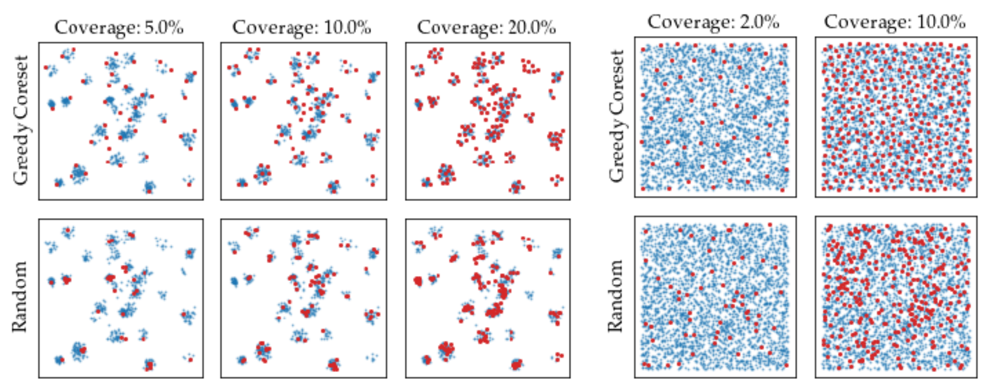
$$\mathcal{M} = \bigcup_{x_i \in \mathcal{X}_N} \mathcal{P}_{s,p}(\phi_j(x_i))$$

PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Coreset-reduced patch feature memory bank
    - For increasing sizes of Xn, memory bank becomes exceedingly large, and with it both inference time to evaluate novel test data and required storage get large
    - Memory bank uses a coreset sub-sampling mechanism to reduce inference time
      - Minimax facility location coreset selection is used to ensure approximately similar coverage of the memory back coreset in patch-level feature space

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

      - Random subsampling will lose significant information available in M encoded in the coverage of nominal features (coreset subsampling better approximates the spatial support)



(a) (b)

PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Anomaly Detection with the nominal patch-feature memory bank
    - Image-level anomaly score는 test 이미지가 가지고 있는 patch 중 가장 anomaly score가 높은 patch와 training 과정에서 사용된 가장 가까운 sampl을 비교해서 distanc가 얼마나 멀리 떨어져 있는지 비교

$$m^{\text{test},*}, m^* = \underset{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})}{\arg\max} \; \underset{m \in \mathcal{M}}{\arg\min} \left\| m^{\text{test}} - m \right\|_2$$

$$s^* = \left\| m^{\text{test},*} - m^* \right\|_2 .$$

※ Test sample에 가장 가까운 normal sample의 주변에 normal sample 들이 많이 있으면, normal 한 patch 라고 판단, 반대로 그 주변에 normal sample이 거의 없으면 anomalous 하다라고 판단하여 anomaly score를 높여주는 방식으로 weight를 줌

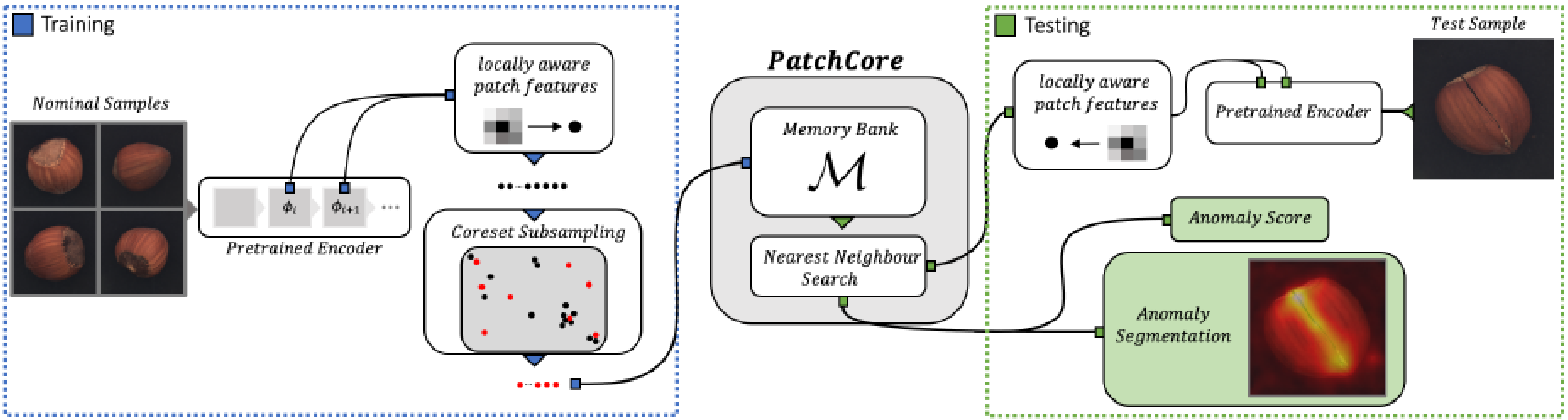
$$s = \left(1 - \frac{\exp \left\| m^{\text{test},*} - m^* \right\|_2}{\sum_{m \in \mathcal{N}_b(m^*)} \exp \left\| m^{\text{test},*} - m \right\|_2} \right) \cdot s^*$$

    - Segmentation requires the computation of the anomaly score for each patch through the arg max-operation by realigning computed patch anomaly scores based on their repective spatial location

※ Pixel 단위로 비교를 하기 때문에 공간 정보를 기반으로, 해당 공간에 대응되는 patch 들끼리 직접 비교하면 anomaly score 비교 가능

PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - PatchCore recap



– During training, 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 on patch is anomalous, and pixel-level anomaly segmentation is generated by scoring each patch-feature

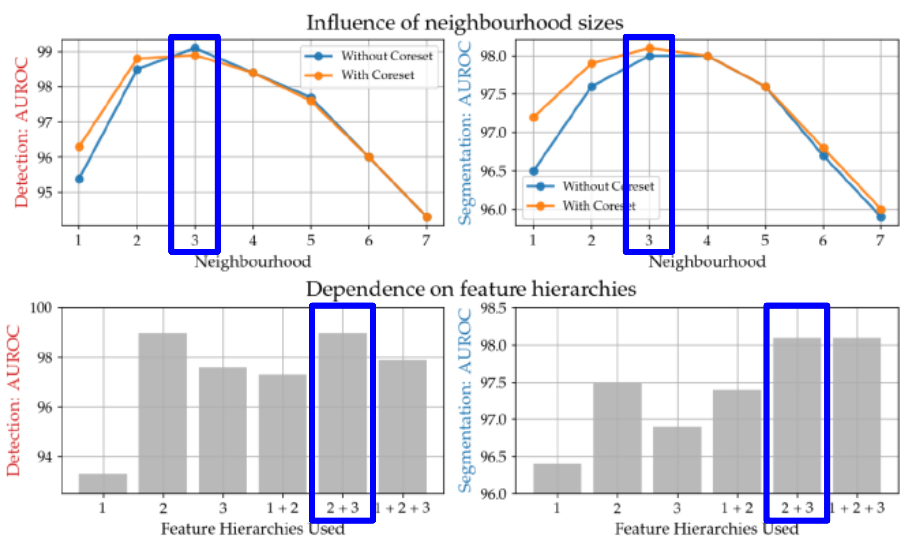
PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Results (MVTech AD dataset)
    - Anomaly Detection Performance (AUROC)

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

    - Anomaly Segmentation Performance (pixelwise AUROC)

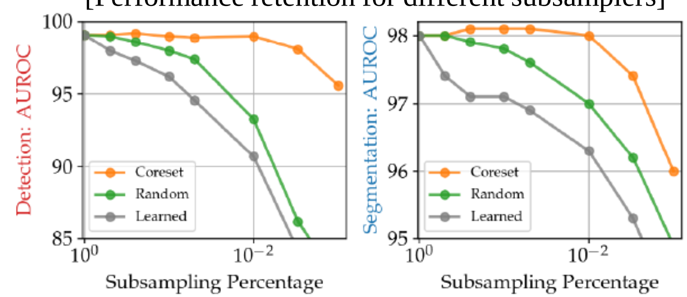
Method	AE_{SSIM} [5]	γ-VAE + grad. [15]	CAVGA-R_w [52]	PatchSVDD [56]	SPADE [10]	PaDiM [14]	PatchCore−25%	PatchCore−10%	PatchCore−1%
AUROC ↑	87	88.8	89	95.7	96.0	97.5	**98.1**	**98.1**	98.0
Error ↓	13	11.2	11	4.3	4.0	2.5	**1.9**	**1.9**	2.0



[Mean inference time per image]

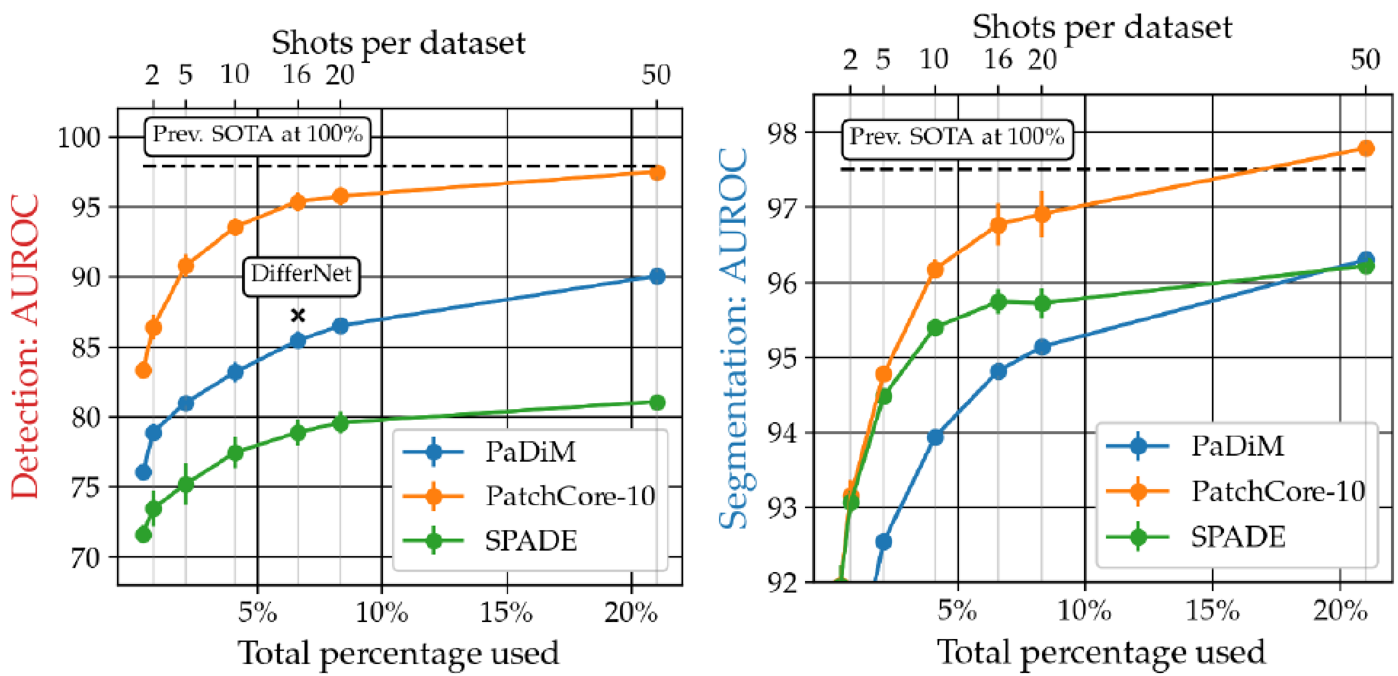
Method	*PatchCore*−100	*PatchCore*−10	*PatchCore*−1
Scores	(99.1, 98.0, 93.3)	(99.0, 98.1, 93.5)	(99.0, 98.0, 93.1)
Time (s)	0.6	0.22	0.17
Method	*PatchCore*−100 + IVFPQ	SPADE	PaDiM
Scores	(98.0, 97.9, 93.0)	(85.3, 96.6, 91.5)	(95.4, 97.3, 91.8)
Time (s)	0.2	0.66	0.19

[Performance retention for different subsamplers]



PatchCore

- Towards Total Recall in Industrial Anomaly Detection
  - Results (MVTech AD dataset)
    - Shows notably higher sample-efficiency than competing methods, retaining strong performance with only few samples per class and matches the previous state-of-the-art with only a fraction of nominal training samples



감사합니다.