

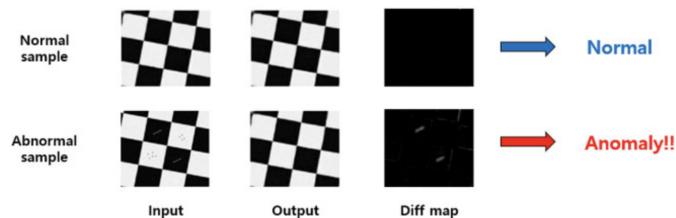
Anomaly Detection

[22'CVPR] PatchCore : Towards Total Recall in Industrial Anomaly Detection

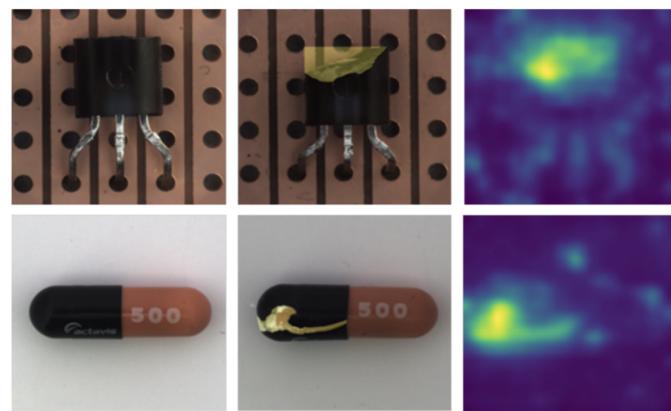
| [arxiv Link - PatchCore Anomaly Detection](#)

1 . Introduction

- Anomaly detection ??
 - Anomaly detection 방식은 어떤 데이터 안에서 측정된 값들이 정상 데이터가 가진 특징과 다른 결과가 나왔을 경우를 비정상 데이터(anomaly data)로 탐지하는 방식
 - 이상치 탐지 방식은 supervised, unsupervised, semi-supervised(one-class) 방식이 존재

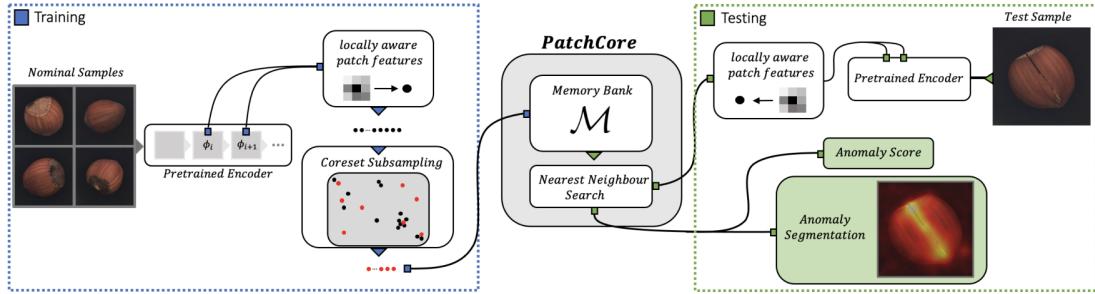


- Anomaly detection & Anomaly segmentation(localization)
 - detection 같은 경우, input image 내 이상치 포함 여부에 따라 이상치 탐지
 - segmentation 같은 경우, input image 내, pixel-level에 따라 이상치 탐지



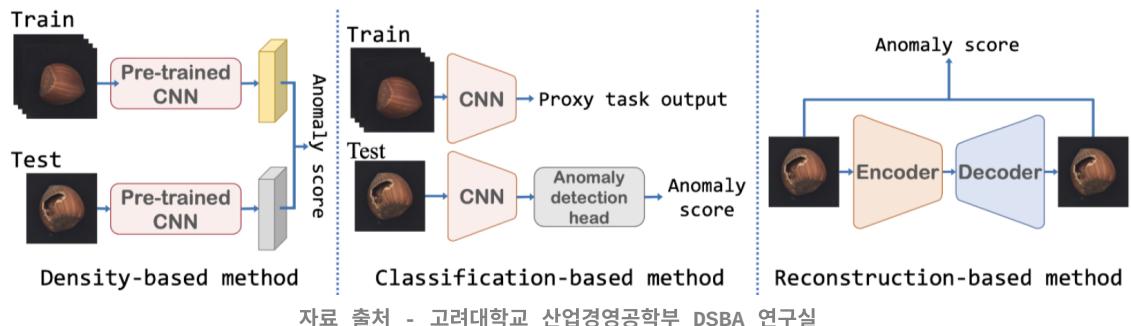
- PatchCore 방식 Anomaly detection이란?
 - imageNet 기반 Pretrain model을 적극 활용하여 학습 과정 없이, 데이터셋의 feature 추출 하여, anomaly image를 탐지 하는 방식의 모델

- 양품 데이터셋 전체를 저장하는 방식이 아닌 coreset 내 subsampling 과정을 통해 효과적으로 저장하며, 적은 양의 양품 데이터만으로도 높은 성능과 inference time(linear)을 감소
- keyword - coresnet subsampling(greedy search), faiss, memory bank...

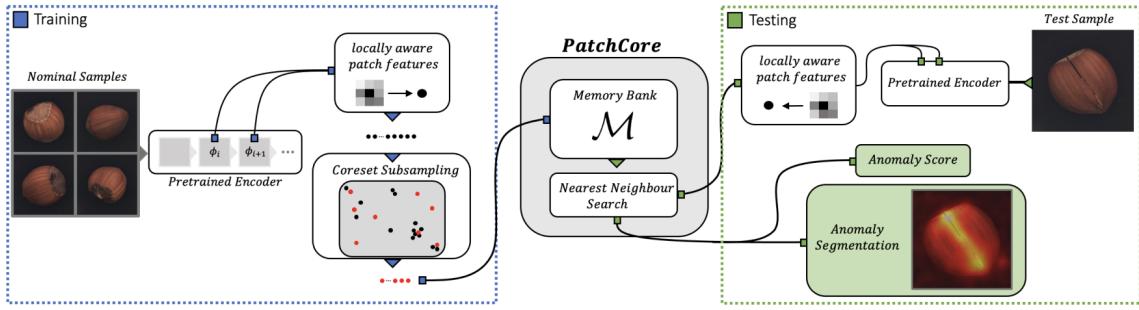


2 . Related Works & Method

- **Anomaly Detection** 방법론
 - **density-based method** : 정상 데이터의 분포를 통해 비정상 데이터를 탐지
 - pretrained network를 사용하는 방식이며, patchcore 이전에 spade, padim 방식이 존재
(PatchCore - [SPADE](#), [PaDim](#) 방식을 이용한 task)
 - SPADE - locally aware 하지 않아 주변 patch들에 대한 정보 없고 모든 feature를 사용, train feature count만큼 linear하게 inference time이 증가하는 단점
 - PaDim - 동일 위치의 patch feature끼리만 비교하므로, alignment가 맞지 않을 경우, anomaly에 대한 판단 불가
 - **classification-based method** : 지도 학습 기반이며, 사전 학습 모델을 통해 one-class classification을 적용하여 비정상 데이터 탐지
 - **Reconstruction-based method** : 정상 data pattern train 후, input sample를 재구성하여 정상적인 data pattern과의 차이를 계산하는 방식



- **PatchCore Model Structure**

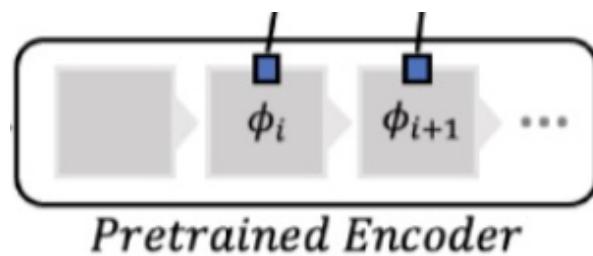


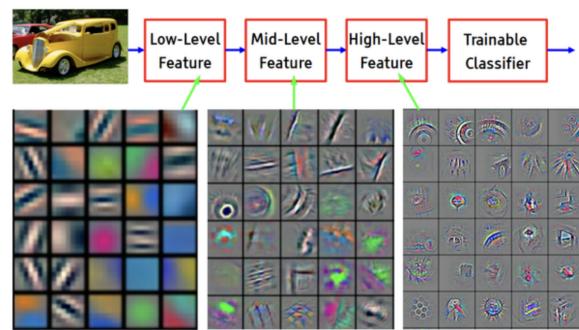
- PatchCore Model task

1. Pretrained network - feature를 추출
2. 추출된 feature들의 locally aware patch feature로 만들
3. coreset sampling을 통해 memory bank에 추가 될 patch feature를 선별 후, 추가
4. memory bank에 있는 feature들을 하나씩 꺼내, test 단계에서 anomaly score를 통해 판정

- Training

- locally aware patch features
 - Mid-level feature를 통한 local patch feature 사용
 - 이전 anomaly detection 같은 경우, high level feature를 사용하였으나, space data loss(high level로 갈수록 위치 정보 손실), pretrain bias의 빈도가 높음
 - patchcore에서는 위치 정보 및 imagenet bias의 비중을 낮추기 위해, mid level feature를 사용
 - patchcore는 intermediate feature hierarchies i, i+1만을 사용하는 이유는 user feature의 generality를 방지하고, spatial resolution을 보존하기 위함



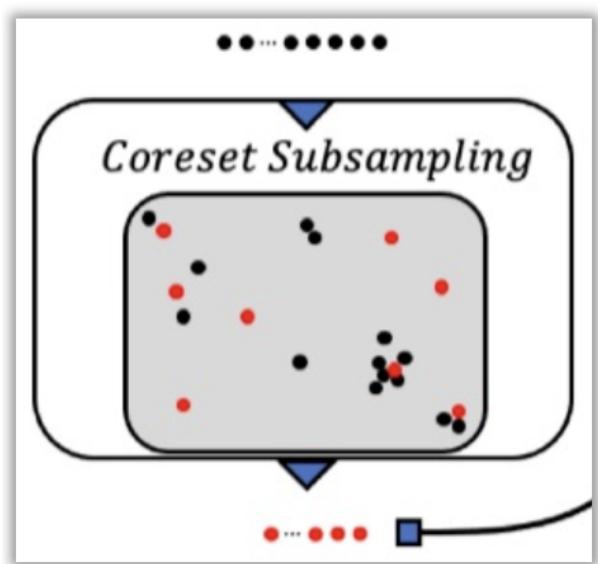


자료 출처 -

<https://pdfs.semanticscholar.org/3154/d217c6fca87aecd99f47bdd6ed9b2be47c0c.pdf?ref=dataroots.ghost.io>

- coresnet subsampling

- SPADE 방식처럼 global averaging 한 후, feature를 memory bank에 모두 넣으면 무거움
- Greedy Search 방법으로 subsampling을 수행



- PatchCore

- memory bank (pre - coresnet subsampling)
 - Patchcore 방식은 k nearest neighbor 방식으로 anomaly score를 계산 하기에, memory bank 안에 feature들 많으면 많을수록 computation 하는 과정이 오래 걸림 (prev - spade 방식이 추출된 feature들을 memory 안에 모두 저장, inference time, 연산량 이슈)

Algorithm 1: PatchCore memory bank.

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

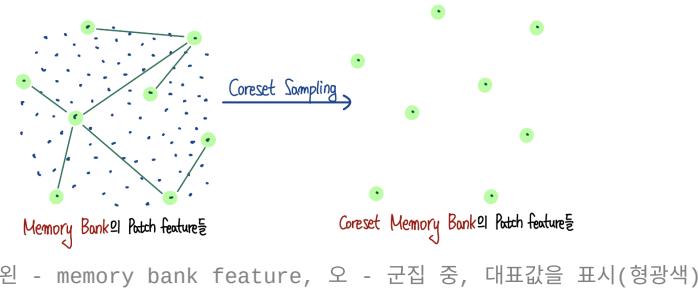
Output: Patch-level Memory bank \mathcal{M} .

Algorithm:

```

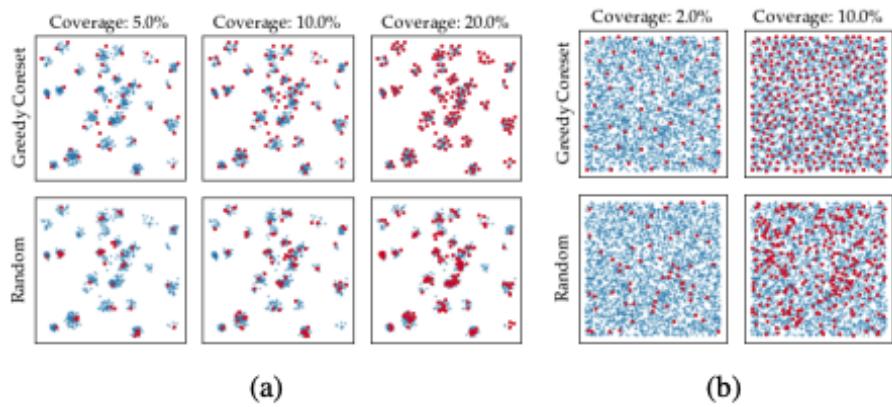
 $\mathcal{M} \leftarrow \{\}$ 
for  $x_i \in \mathcal{X}_N$  do
|    $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{P}_{s,p}(\phi_j(x_i))$ 
end
/* Apply greedy coresset selection. */
 $\mathcal{M}_C \leftarrow \{\}$ 
for  $i \in [0, \dots, l - 1]$  do
|    $m_i \leftarrow \arg \max_{m \in \mathcal{M} - \mathcal{M}_C} \min_{n \in \mathcal{M}_C} \|\psi(m) - \psi(n)\|_2$ 
|    $\mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{m_i\}$ 
end
 $\mathcal{M} \leftarrow \mathcal{M}_C$ 

```



원 - memory bank feature, 오 - 군집 중, 대표값을 표시(형광색)

- 최초 sample을 고르고, 이후 patch feature들 중 가장 멀리 떨어져 있는 patch feature를 선택하는 과정을 반복 하여, 군집 데이터들의 대표값들을 선정
- uniform distribution test는 아래 그림과 같이 greedy search algorithm이 종복 없이 sampling 되어 있는 것을 볼 수 있음.



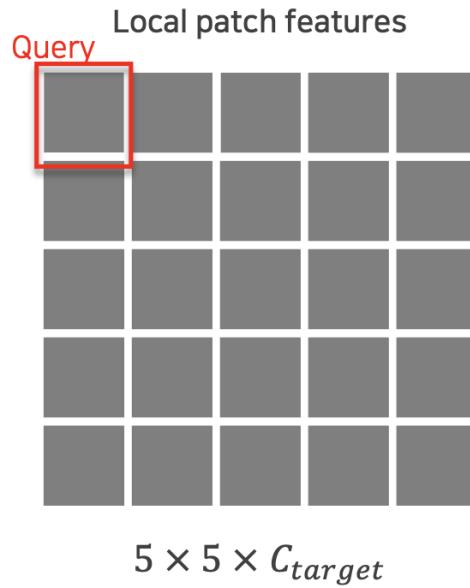
random, greedy search 방식

- Testing
 - detection and segmentation(localization)
 - detection - image level 이므로 $\arg \max$, \min
(test image 내에서 anomaly score가 가장 높은 값을 anomaly score로 사용)

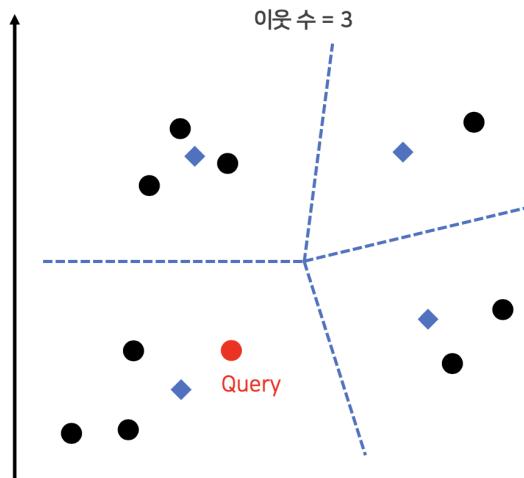
- seg, localization - pixel level 단위이며, 공간 정보를 가지고 있어 해당 공간에 대응되는 patch들끼리 비교를 하면 anomaly score 산출 가능

$$m^{\text{test},*}, m^* = \arg \max_{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})} \arg \min_{m \in \mathcal{M}} \|m^{\text{test}} - m\|_2 \\ s^* = \|m^{\text{test},*} - m^*\|_2.$$

- Test image에 대한 local patch features 추출하여 각 patch features를 memory bank에 query로 사용

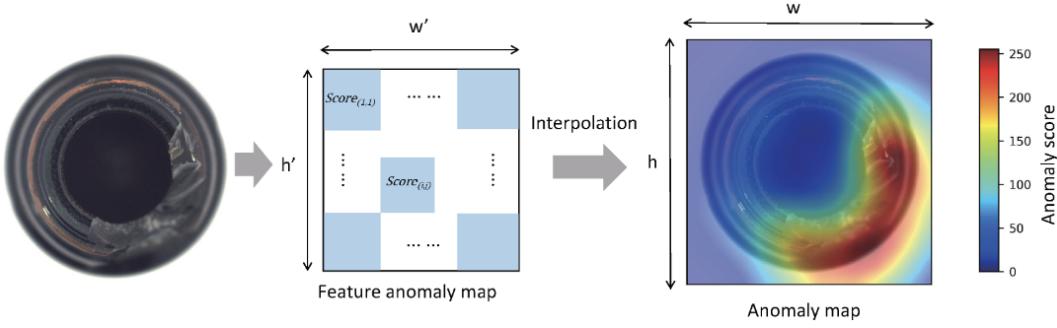


- Nearest Neighbor 방법을 통해 query와 coresset에 대한 distance 를 기준으로 anomaly score 계산
- Nearest Neighbor는 faiss를 통해 빠르게 연산 가능
- Faiss는 모든 feature를 비교하지 않고, cluster를 활용하여 효율적인 similarity 계산 가능함



- anomaly score

- image level - patch들 중 가장 큰 anomaly score를 image에 대한 anomaly score
- pixel level - local patch에 대한 resolution을 source(원본) Image size로 interpolation 후 gaussian smoothing을 적용



3 . Experiments

- anomaly detection and segmentation
 - performance auroc or pixel-wise auroc - MVTec data set

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

Method	AE _{SIM} [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

- coresnet sampling 방법론 성능

- 이전 sota WideResNet50을 backbone으로 사용한 PaDiM이었으나, patchcore 등장 이후 바뀜
- 적은 데이터 수에도 이전 방법론보다 높은 성능을 보임
- higher sample efficiency (학습 수가 적어도 sota level의 성능 확인)

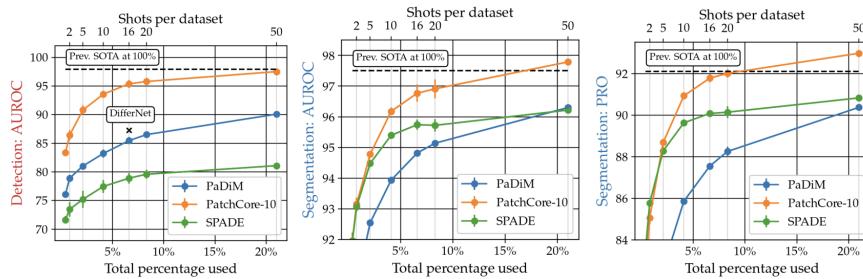


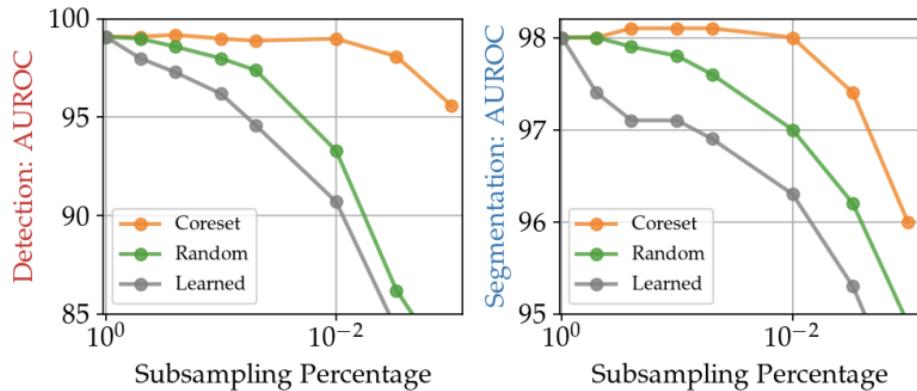
Figure 6. PatchCore shows notably higher sample-efficiency than competitors, matching the previous state-of-the-art with a fraction of nominal training data. Note that PaDiM and SPADE were reimplemented with WideResNet50 for comparability.

- patch core에서 coresnet sampling을 진행해도 비교적 성능 하락이 적으며, 초기엔 오히려 성능이

잠시 오르는 부분을 아래 그래프에서 확인 가능 (이전 random, learned 방식보다 우수한 성능)

- memory bank size를 줄여도 성능 유지 (inference time 감소)

Method	<i>PatchCore</i> -100	<i>PatchCore</i> -10	<i>PatchCore</i> -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	<i>PatchCore</i> -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



4 . Conclusion

- patchcore는 density base method 방식이며, 이전 sota anomaly detection 방식보다 가볍고 우수한 성능이 보장된다.
- feature extract를 다양한 방법론으로 접근 하거나, image alingment 방식을 개선할 새로운 방법론이 더 나오면 산업 분야에 더 우수한 성능? General하게 적용 가능 할 것 같습니다.

5 . Q&A - END (~)

6 . summary

normal sample train 과정을 통해 얻은 feature를 patch 형태로 저장, coresset 방식을 통해 memory bank 경량화 작업을 한다.

test sample이 들어 왔을 때, patch를 뽑고 knn로 비교를 해서 anomaly score를 구하고, image 또는 pixel level 단으로 anomaly를 비교해서 det 또는 seg로 사용 가능하다.

그리고 corset 과정을 통해 얻은 memory bank로 인해 sample efficiency하고 linear 하지만 핵심적인 dataset 기반으로 fast한 anomaly detection model을 구성 할 수 있다.