

---

# Industrial Image Anomaly Detection

---

2025.06.13

Data Mining & Quality Analytics Lab.

추창욱

# 발표자 소개

---



## ❖ 추창욱 (Changwook Chu)

- 고려대학교 산업경영공학과 대학원 재학
- Data Mining & Quality Analytics Lab. (김성범 교수님)
- M.S. Student (2023.09 ~ Present)

## ❖ Research Interest

- Diffusion Models for Time Series
- Large Language Models

## ❖ Contact

- [chaliechu117@korea.ac.kr](mailto:chaliechu117@korea.ac.kr)

# Contents

---

## ❖ Industrial Anomaly Detection

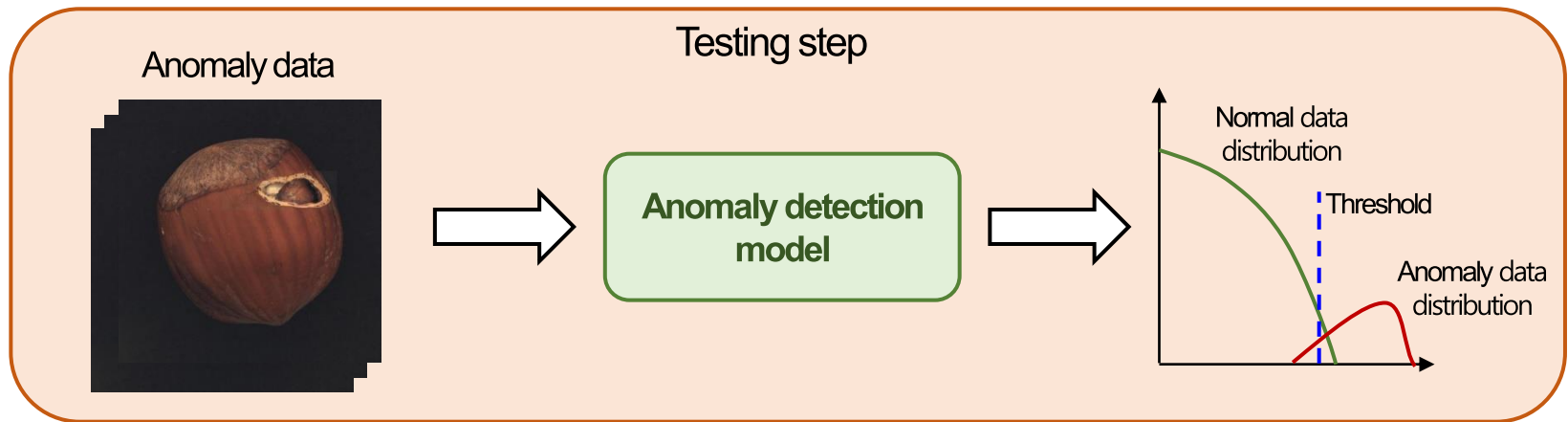
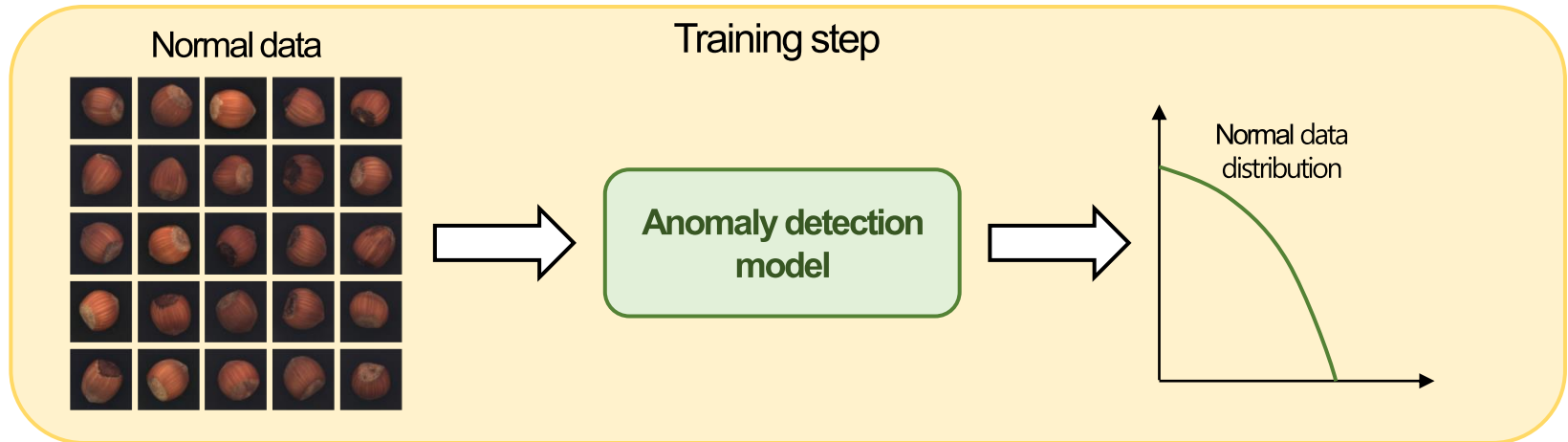
- PatchCore: Towards Total Recall in Industrial Anomaly Detection (2022, CVPR)
- FastRecon: Few-shot Industrial Anomaly Detection via Fast Feature Reconstruction (2023, ICCV)
- AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection (2024, ICLR)

## ❖ Conclusion

# Introduction

## ❖ 이미지 이상 탐지란 무엇인가?

- 현실적으로 이상 데이터를 충분히 확보하지 못하는 상황을 가정하여 정상 데이터만 학습
- 정상 데이터가 가진 특징을 기반으로 이와 다른 결과가 발생시 이상으로 탐지



# Unsupervised Anomaly Detection

## ❖ Unsupervised Anomaly Detection

- 학습 시 비정상 데이터의 사용 여부 및 레이블 유무에 따라 분류

### Supervised Anomaly Detection



- 학습 시, 레이블링 된 정상/비정상 데이터 모두 사용한 경우
- 모델 성능 평가가 직관적
- **Class imbalance** 문제가 발생

### One-class Anomaly Detection



- 학습 시, 레이블링 된 정상 데이터 사용한 경우
- 정상 데이터만 확보되는 경우 좋음
- 정상 데이터만 확보된다는 보장 X

### Unsupervised Anomaly Detection



- 학습 시, 데이터가 대부분 정상으로 이루어졌다고 **가정**
- 데이터에 레이블이 없는 경우 좋음
- 현실과 가장 유사한 상황

# Unsupervised Anomaly Detection

---

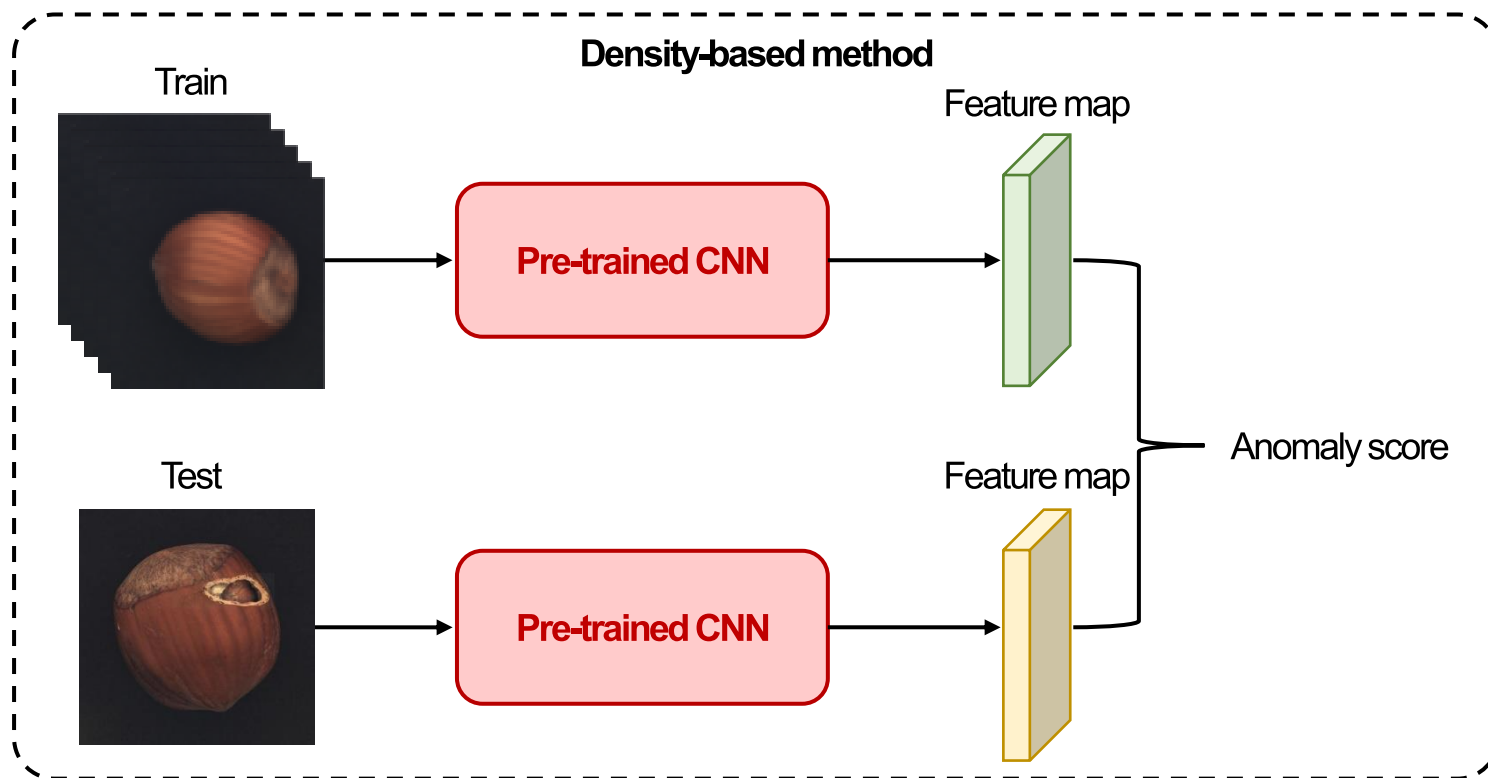
## ❖ Unsupervised Anomaly Detection

- 데이터가 대부분 정상으로 이루어졌다고 가정하고 학습 (레이블 X)
- Density-based: 정상 데이터의 분포를 통해 비정상 데이터 탐지
- Classification-based: Proxy task 정의 후 학습 진행 및 one-class classification 적용하여 비정상 탐지
- Reconstruction-based: 정상 데이터 만을 복원하도록 학습하여 비정상 데이터 추론 시 재구축된 결과와 차이로 비정상 데이터 탐지

# Unsupervised Anomaly Detection

## ❖ Unsupervised Anomaly Detection

- 데이터가 대부분 정상으로 이루어졌다고 가정하고 학습 (레이블 X)
- Density-based: 정상 데이터의 분포를 통해 비정상 데이터 탐지
  - ✓ 사전학습 된 모델 사용
  - ✓ 추가적인 학습이 필요 없음



# Unsupervised Anomaly Detection

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- CVPR 2022 발표, 2025년 6월 기준 1319회 인용

### Towards Total Recall in Industrial Anomaly Detection

Karsten Roth<sup>1,\*</sup>, Latha Pemula<sup>2</sup>, Joaquin Zepeda<sup>2</sup>, Bernhard Schölkopf<sup>2</sup>, Thomas Brox<sup>2</sup>, Peter Gehler<sup>2</sup>  
<sup>1</sup>University of Tübingen <sup>2</sup>Amazon AWS

#### Abstract

Being able to spot defective parts is a critical component in large-scale industrial manufacturing. A particular challenge that we address in this work is the cold-start problem: fit a model using nominal (non-defective) example images only. While handcrafted solutions per class are possible, the goal is to build systems that work well simultaneously on many different tasks automatically. The best performing approaches combine embeddings from ImageNet models with an outlier detection model. In this paper, we extend on this line of work and propose **PatchCore**, which uses a maximally representative memory bank of nominal patch-features. PatchCore offers competitive inference times while achieving state-of-the-art performance for both detection and localization. On the challenging, widely used MVTec AD benchmark PatchCore achieves an image-level anomaly detection AUROC score of up to 99.6%, more than halving the error compared to the next best competitor. We further report competitive results on two additional datasets and also find competitive results in the few samples regime. Code: [github.com/amazon-research/patchcore-inspection](https://github.com/amazon-research/patchcore-inspection).

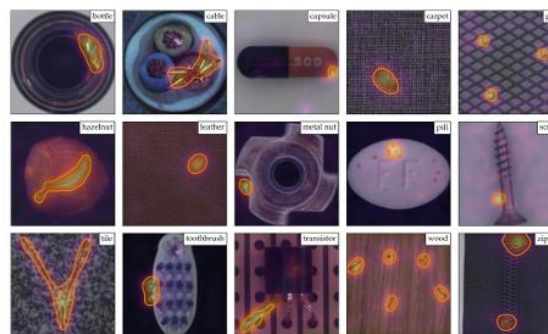


Figure 1. Examples from the MVTec benchmark datasets. Superimposed on the images are the segmentation results from PatchCore. The orange boundary denotes anomaly contours of actual segmentation maps for anomalies such as broken glass, scratches, burns or structural changes in blue-orange color gradients.

vary from subtle changes such as thin scratches to larger structural defects like missing components [5]. Some examples from the MVTec AD benchmark along with results from our proposed method are shown in Figure 1. Existing



# Related works

## ❖ Unsupervised Anomaly Detection, Density-based methods

- SPADE: Sub-Image Anomaly Detection with Deep Pyramid Correspondences (2020, arXiv, 603회 인용)
- PaDiM : a Patch Distribution Modeling Framework for Anomaly Detection and Localization (2021, ICPR, 1058회 인용)

## Sub-Image Anomaly Detection with Deep Pyramid Correspondences

Niv Cohen and Yedid Hoshen

School of Computer Science and Engineering  
The Hebrew University of Jerusalem, Israel.  
{niv.cohen2,yedid.hoshen}@mail.huji.ac.il

**Abstract.** Nearest neighbor (kNN) methods utilizing deep pre-trained features exhibit very strong anomaly detection performance when applied to entire images. A limitation of kNN methods is the lack of segmentation map describing where the anomaly lies inside the image. In this work we present a novel anomaly segmentation approach based on alignment between an anomalous image and a constant number of the similar normal images. Our method, Semantic Pyramid Anomaly Detection (SPADE) uses correspondences based on a multi-resolution feature pyramid. SPADE is shown to achieve state-of-the-art performance on unsupervised anomaly detection and localization while requiring virtually no training time.

## PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization

Thomas Defard, Aleksandr Setkov, Angélique Loesch, Romaric Audigier

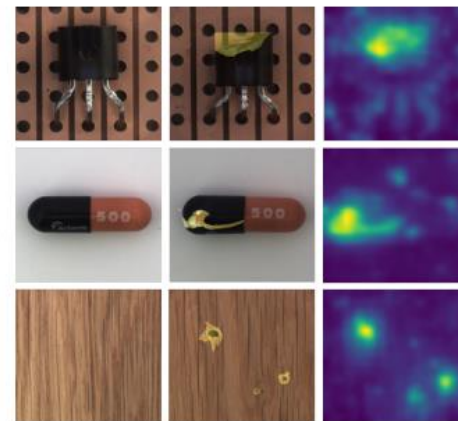
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

thomas.defard@imt-atlantique.net, {aleksandr.setkov, angelique.loesch, romaric.audigier}@cea.fr

*Abstract*—We present a new framework for Patch Distribution Modeling, PaDiM, to concurrently detect and localize anomalies in images in a one-class learning setting. PaDiM makes use of a pretrained convolutional neural network (CNN) for patch embedding, and of multivariate Gaussian distributions to get a probabilistic representation of the normal class. It also exploits correlations between the different semantic levels of CNN to better localize anomalies. PaDiM outperforms current state-of-the-art approaches for both anomaly detection and localization on the MVTEC AD and STC datasets. To match real-world visual industrial inspection, we extend the evaluation protocol to assess performance of anomaly localization algorithms on non-aligned dataset. The state-of-the-art performance and low complexity of PaDiM make it a good candidate for many industrial applications.

### I. INTRODUCTION

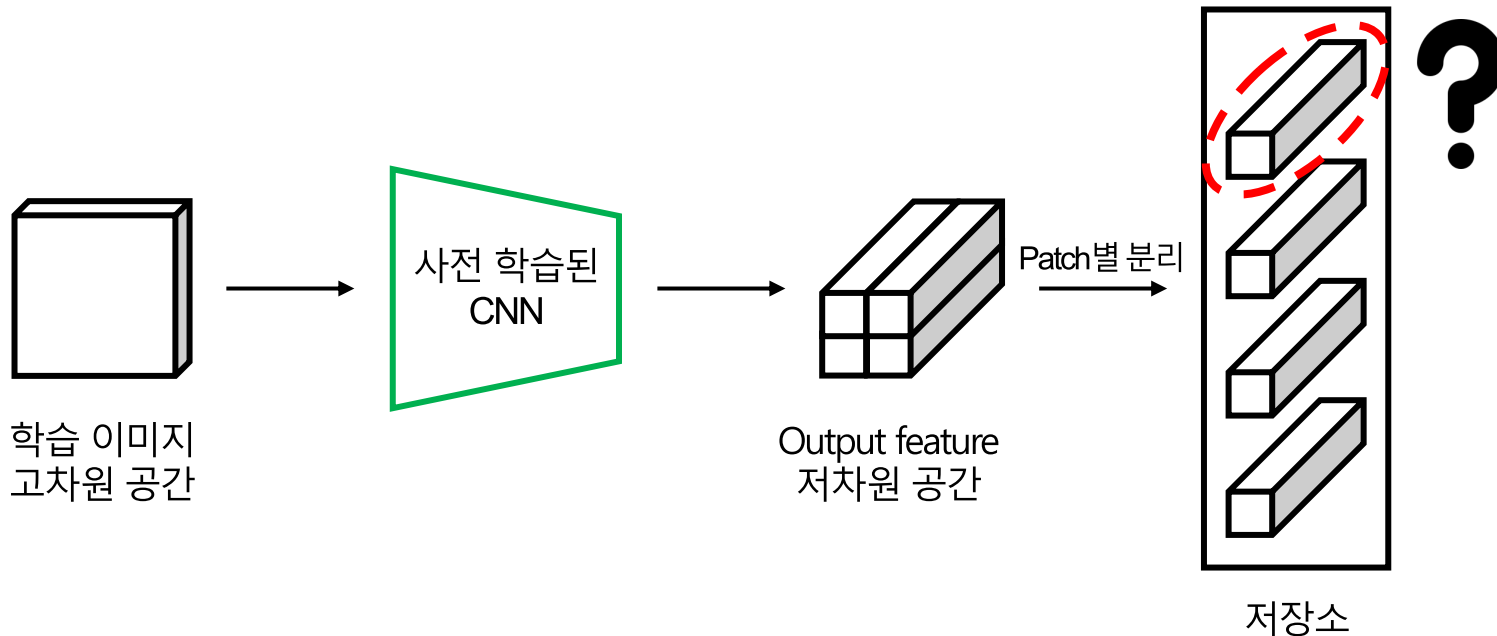
Humans are able to detect heterogeneous or unexpected patterns in a set of homogeneous natural images. This task is known as anomaly or novelty detection and has a large number of applications, among which visual industrial inspections.



# Related works

## ❖ SPADE: Sub-Image Anomaly Detection with Deep Pyramid Correspondences

- 사전 학습된 네트워크로 추출한 feature와 비교
- 어떤 feature map을 사용할 것이냐?
- Anomaly score는 어떻게 측정할 것이냐?

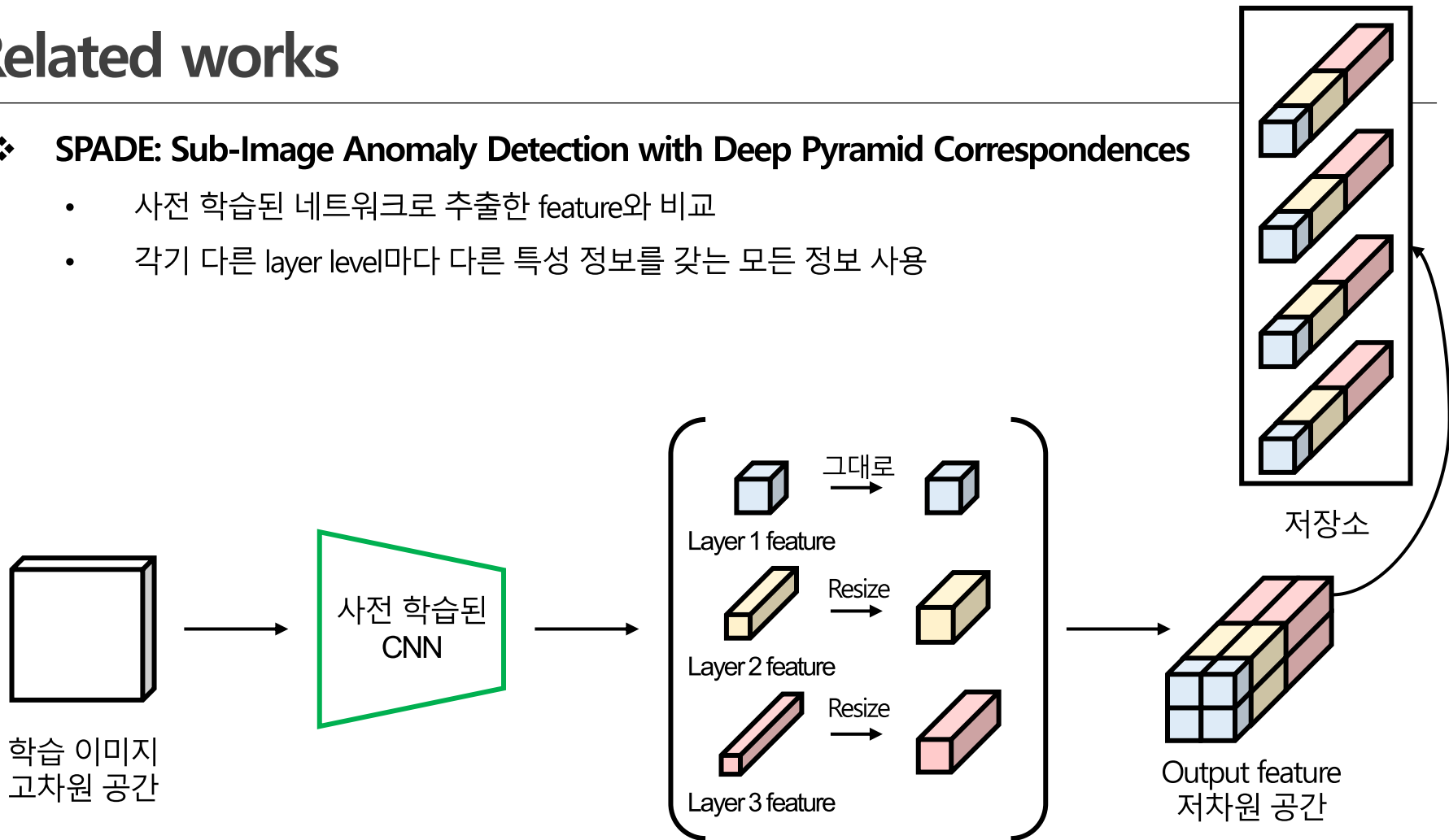


[SPADE 동작 방식

# Related works

## ❖ SPADE: Sub-Image Anomaly Detection with Deep Pyramid Correspondences

- 사전 학습된 네트워크로 추출한 feature와 비교
- 각기 다른 layer level마다 다른 특성 정보를 갖는 모든 정보 사용

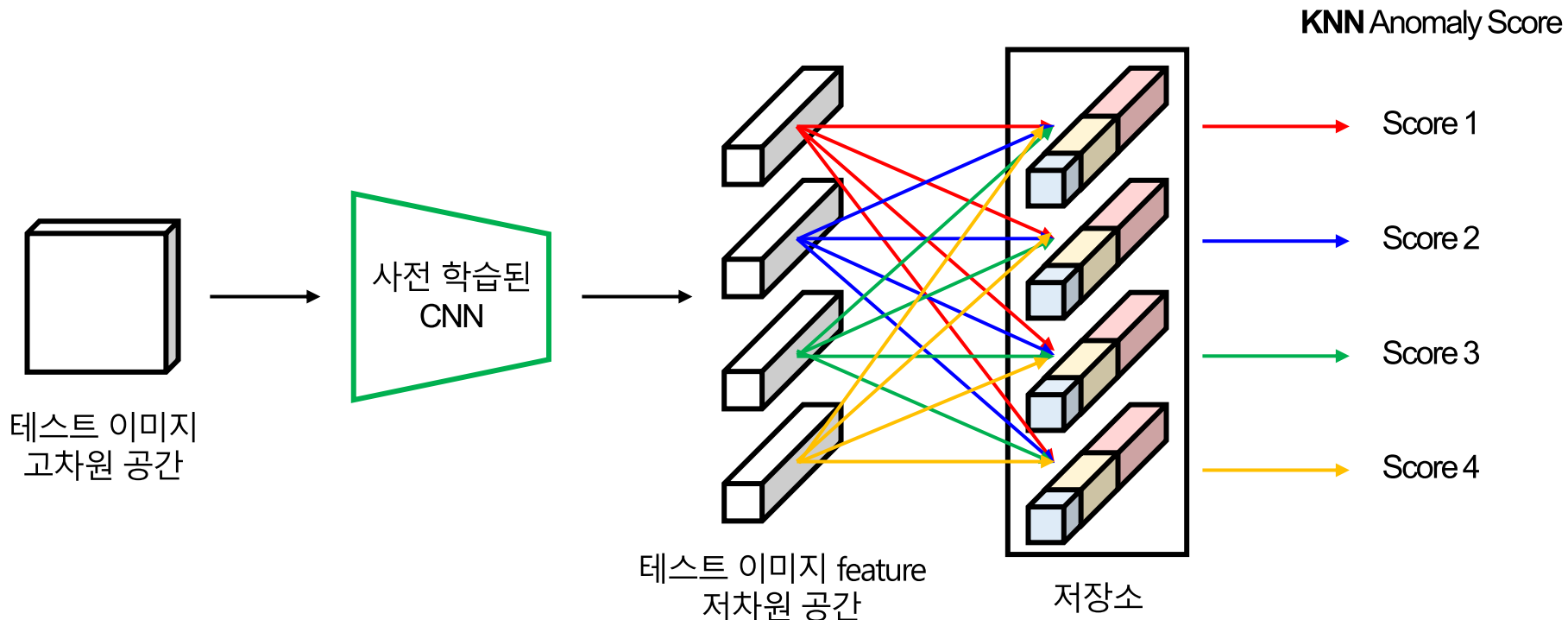


[SPADE 동작 방식

# Related works

## ❖ SPADE: Sub-Image Anomaly Detection with Deep Pyramid Correspondences

- 사전 학습된 네트워크로 추출한 feature를 있는 그대로 사용
- 각 patch별로 저장소의 모든 patch와 비교
- KNN을 사용하여 anomaly score 측정

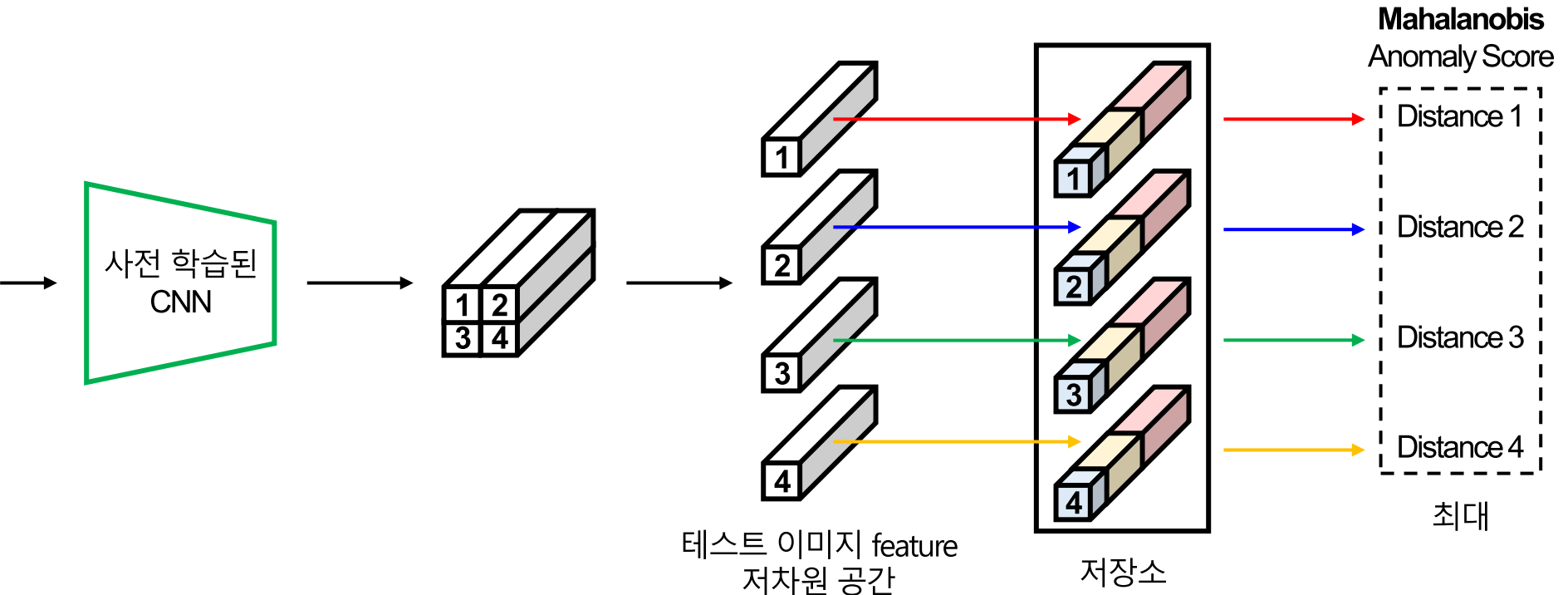


[SPADE 동작 방식

# Related works

## ❖ PaDiM : a Patch Distribution Modeling Framework for Anomaly Detection and Localization

- 사전 학습된 네트워크로 추출한 feature와 비교
- SPADE와 달리 같은 위치의 patch들끼리만 score 측정
- 마할라노비스 거리를 사용하여 거리 측정 후 가장 먼 거리를 anomaly score로 사용

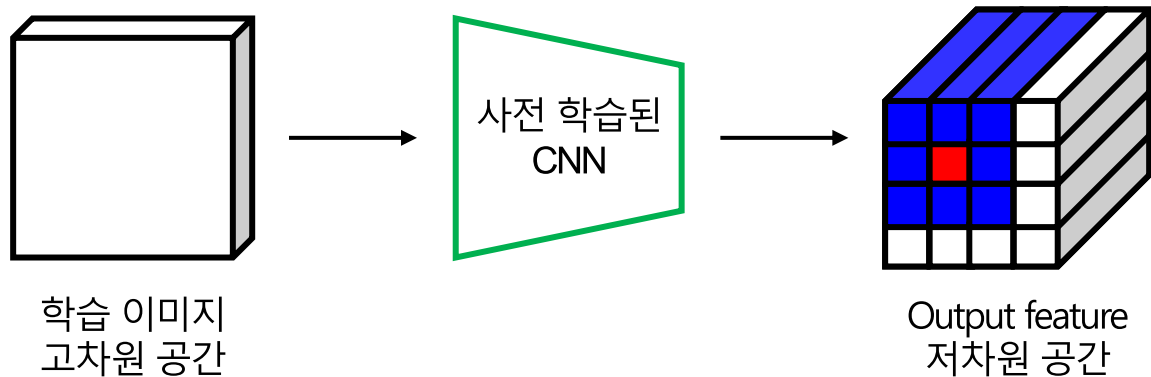


[PaDiM 동작 방식

# Related works

## ❖ SPADE, PaDiM 그리고 PatchCore

- 세가지 모델의 **공통점**
  - ✓ 사전 학습된 모델을 사용해서 추출하여 사용
- SPADE와 PaDiM의 **한계점**
  - ✓ SPADE: 사전 학습된 모델로 추출한 feature를 그대로 사용 (Locally aware 하지 않음)
  - ✓ PaDiM: 동일 위치의 patch feature 끼리 비교 (patch 별 alignment가 맞지 않으면 치명적)

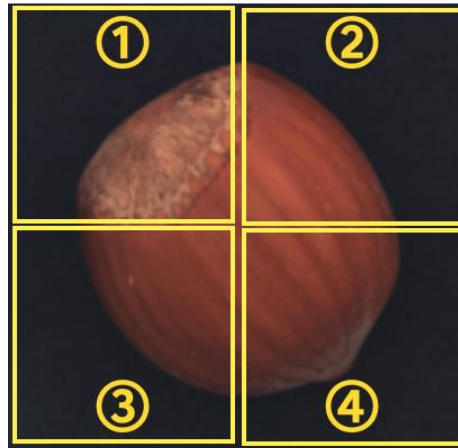


[SPADE 문제점]

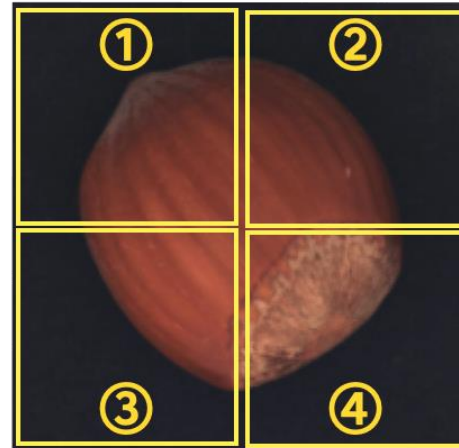
# Related works

## ❖ SPADE, PaDiM 그리고 PatchCore

- 세가지 모델의 **공통점**
  - ✓ 사전 학습된 모델을 사용해서 추출하여 사용
- SPADE와 PaDiM의 **한계점**
  - ✓ SPADE: 사전 학습된 모델로 추출한 feature를 그대로 사용 (Locally aware 하지 않음)
  - ✓ PaDiM: 동일 위치의 patch feature 끼리 비교 (patch 별 alignment가 맞지 않으면 치명적)



학습용 정상 이미지



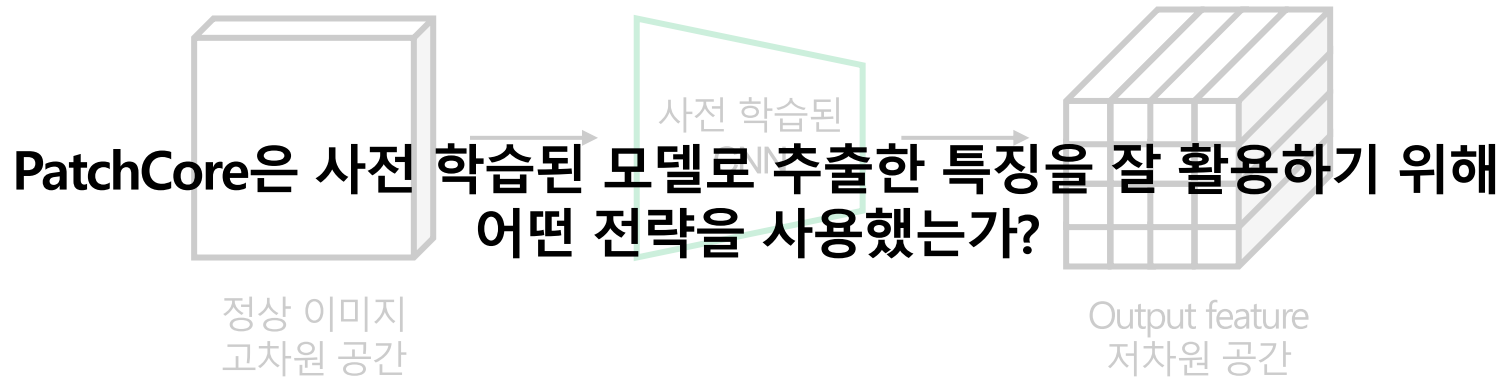
테스트 이미지

[PaDiM 문제점]

# Related works

## ❖ SPADE, PaDiM 그리고 PatchCore

- 세가지 모델의 **공통점**
  - ✓ 사전 학습된 모델을 사용해서 추출하여 사용
- SPADE와 PaDiM의 **한계점**
  - ✓ SPADE: 사전 학습된 모델로 추출한 feature를 그대로 사용 (Locally aware 하지 않음)
  - ✓ PaDiM: 동일 위치의 patch feature 끼리 비교 (patch 별 alignment가 맞지 않으면 치명적)





# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- PatchCore의 핵심
  - ✓ Locally aware한 patch feature 만들기
  - ✓ Coreset subsampling을 하여 계산 시간 줄이기

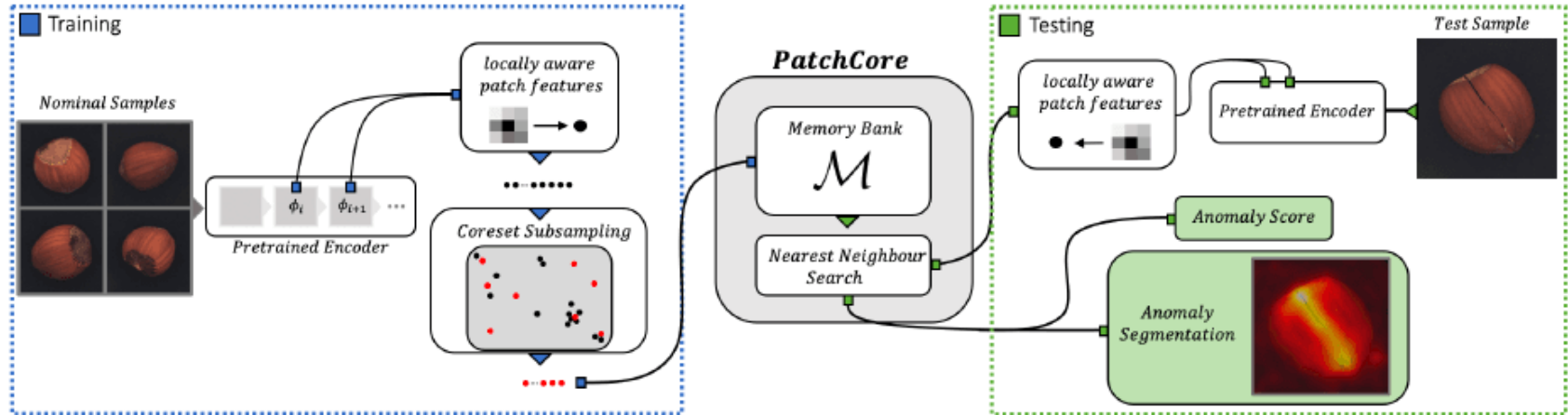


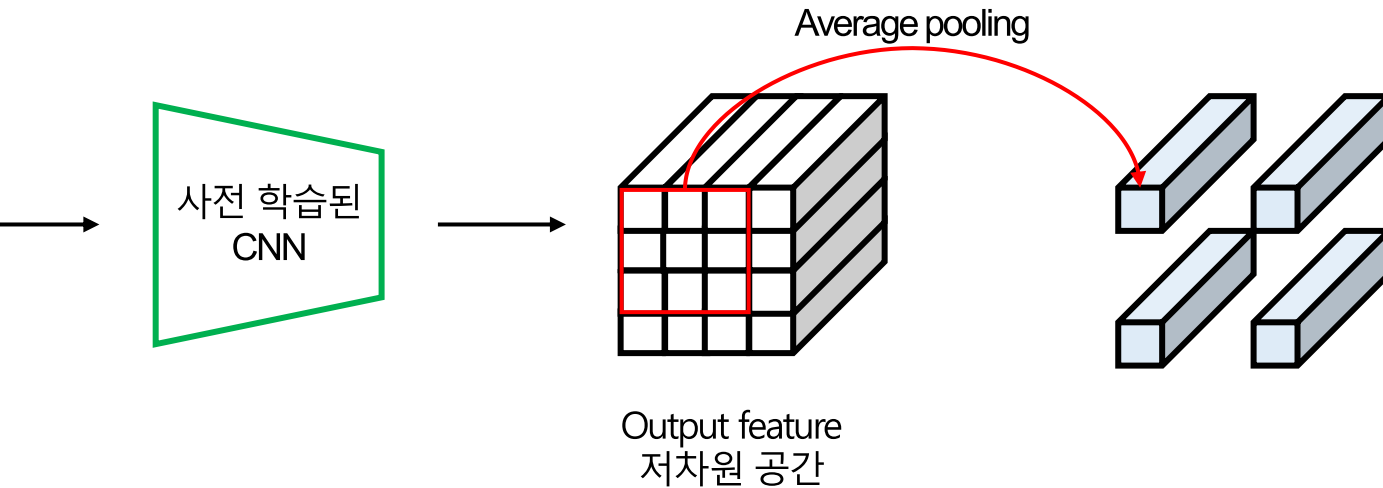
Figure 2: Overview of *PatchCore*. During training, nominal samples are broken down into a memory bank of neighbourhood-aware patch-level features. For reduced redundancy and inference time, this memory bank is downsampled via greedy coreset subsampling. At test time, images are classified as anomalies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

[PatchCore 동작 방식]

# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

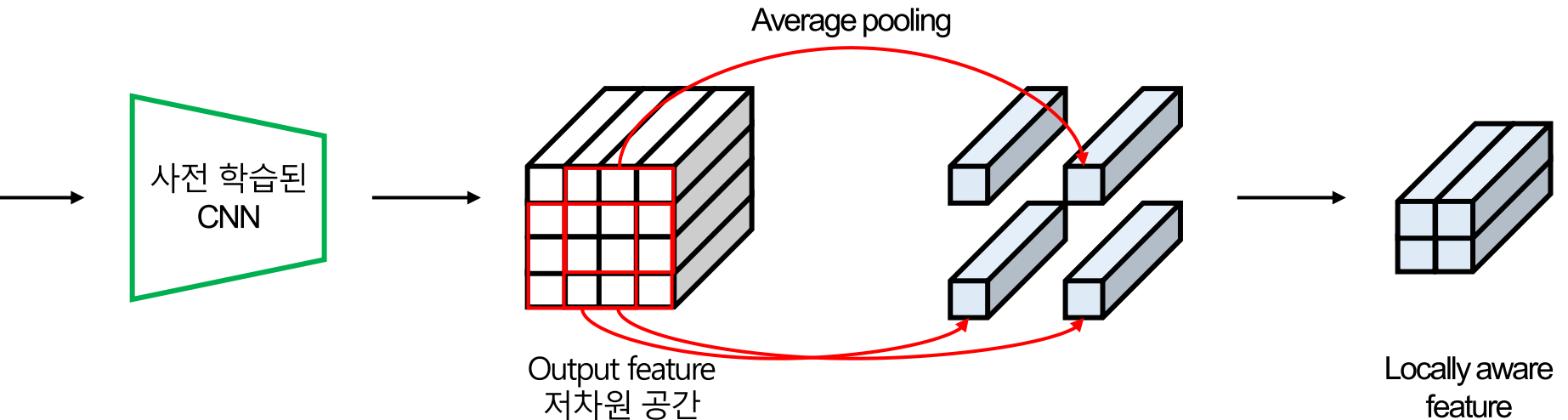
- Locally aware한 patch feature 만들기
- Average pooling 기법을 통해 간단하게 해결
- 주변의 정보를 반영한 feature가 생성됨



# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

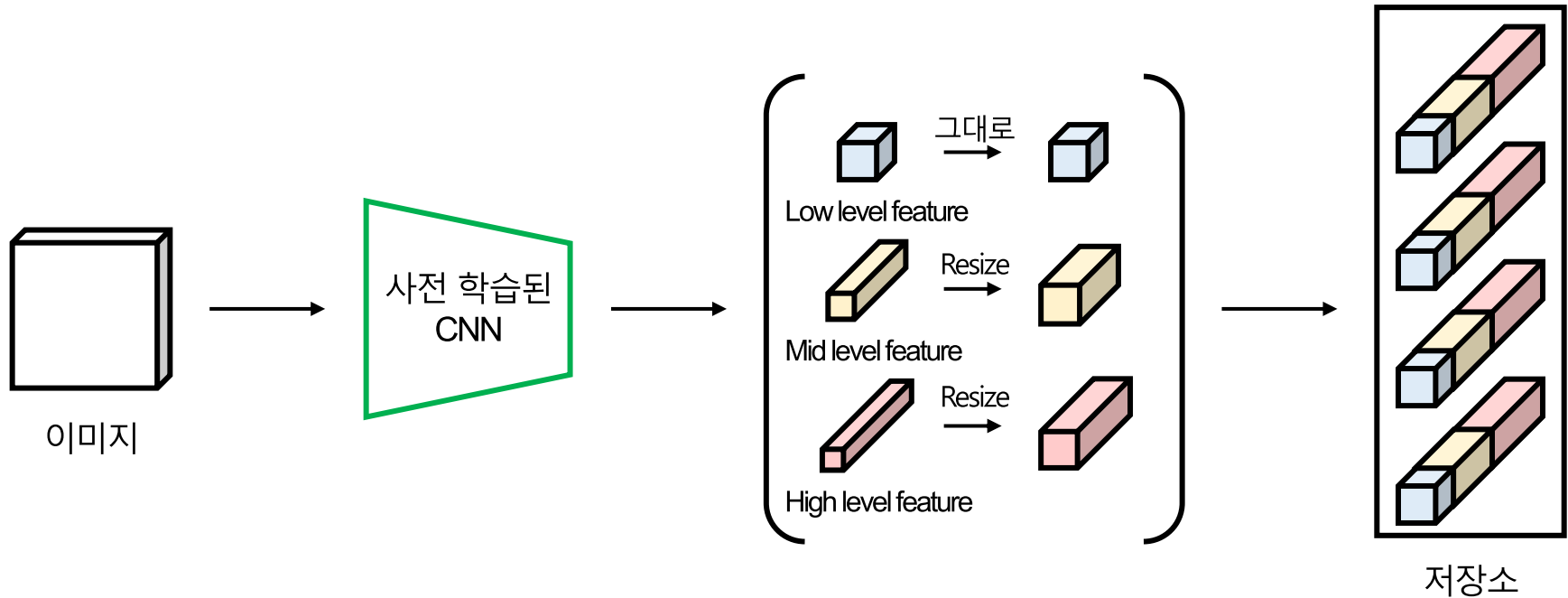
- Locally aware한 patch feature 만들기 (SPADE와 PaDiM 한계점 개선)
- Average pooling 기법을 통해 간단하게 해결
- 주변의 정보를 반영한 feature가 생성됨



# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- Locally aware한 patch feature 만들기 (SPADE와 PaDiM 한계점 개선)
- Mid level feature를 사용하여야 한다고 주장
  - ✓ High level feature은 위치 정보가 너무 많이 사라짐
  - ✓ ImageNet Classification task에 너무 특화된 정보만 남음

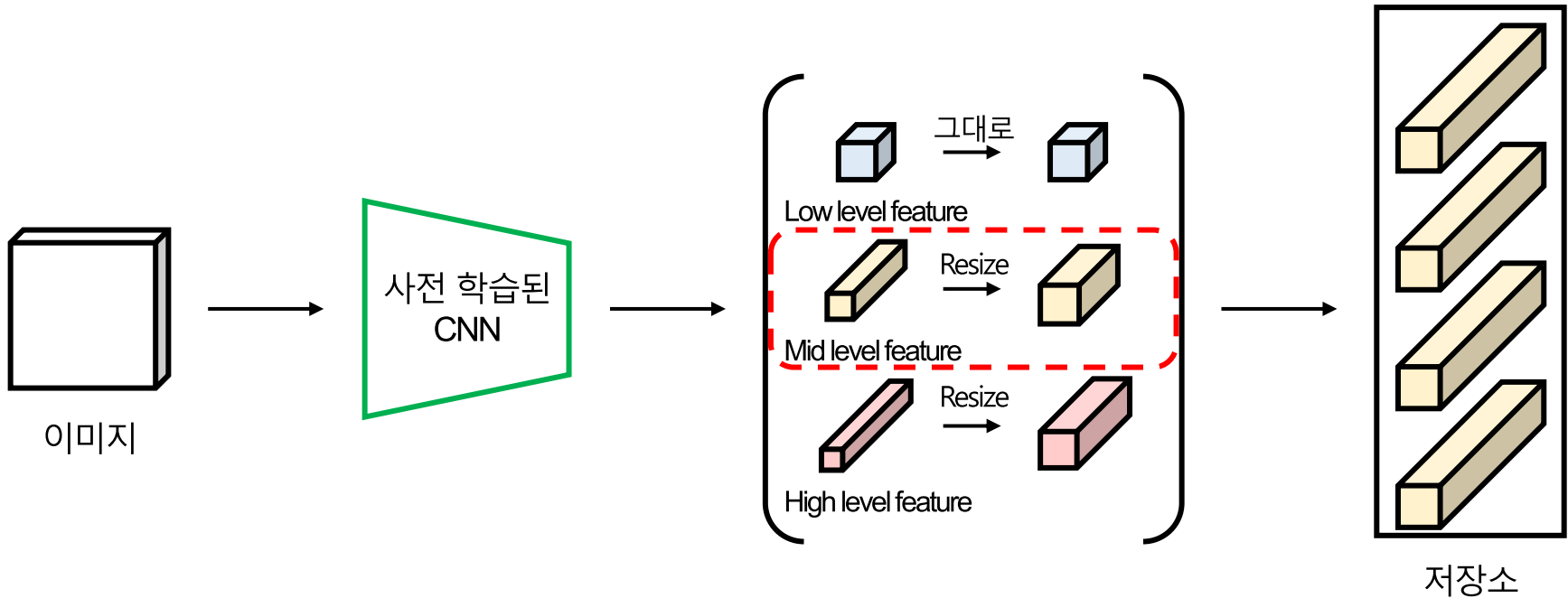


[SPAD & PaDiM feature 추출 방식]

# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- Locally aware한 patch feature 만들기 (SPADE와 PaDiM 한계점 개선)
- Mid level feature를 사용하여야 한다고 주장
  - ✓ High level feature은 위치 정보가 너무 많이 사라짐
  - ✓ ImageNet Classification task에 너무 특화된 정보만 남음

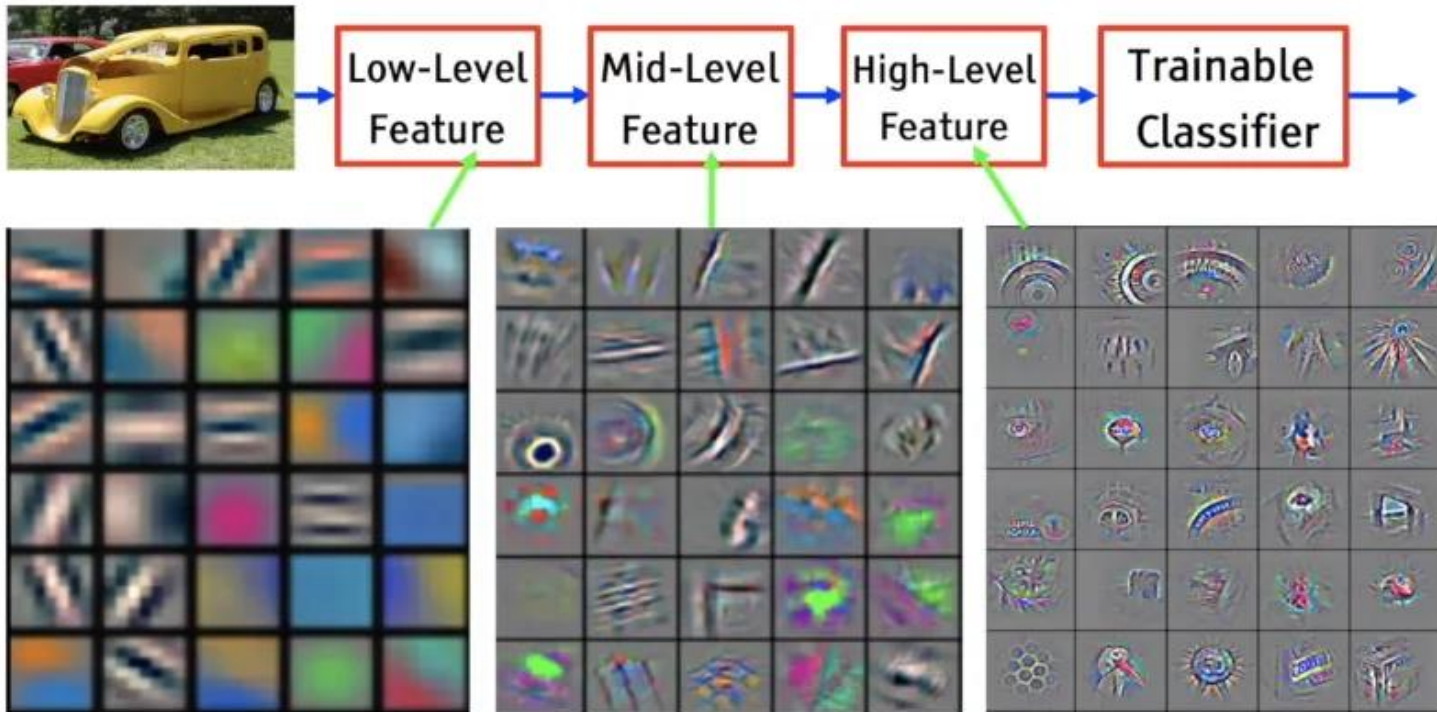


[PatchCore feature 추출 방식

# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- Locally aware한 patch feature 만들기 (SPADE와 PaDiM 한계점 개선)
- Mid level feature를 사용하여야 한다고 주장
  - ✓ High level feature은 위치 정보가 너무 많이 사라짐
  - ✓ ImageNet Classification task에 너무 특화된 정보만 남음

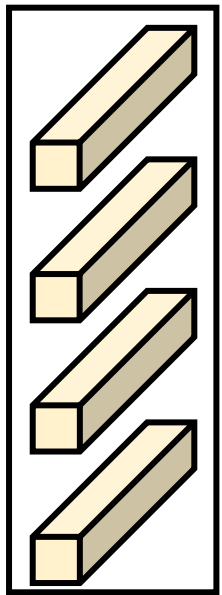


[PatchCore feature 추출 방식]

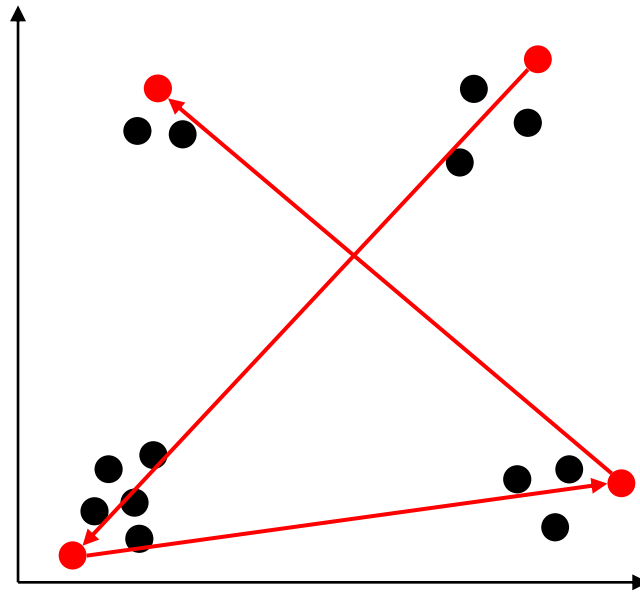
# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

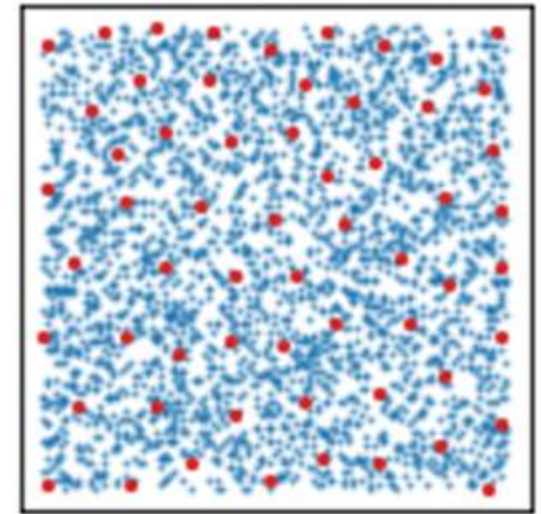
- Coreset subsampling을 하여 계산 시간 줄이기
- 군집 내 대표되는 특징들만 사용하자



저장소



2차원 상의 feature

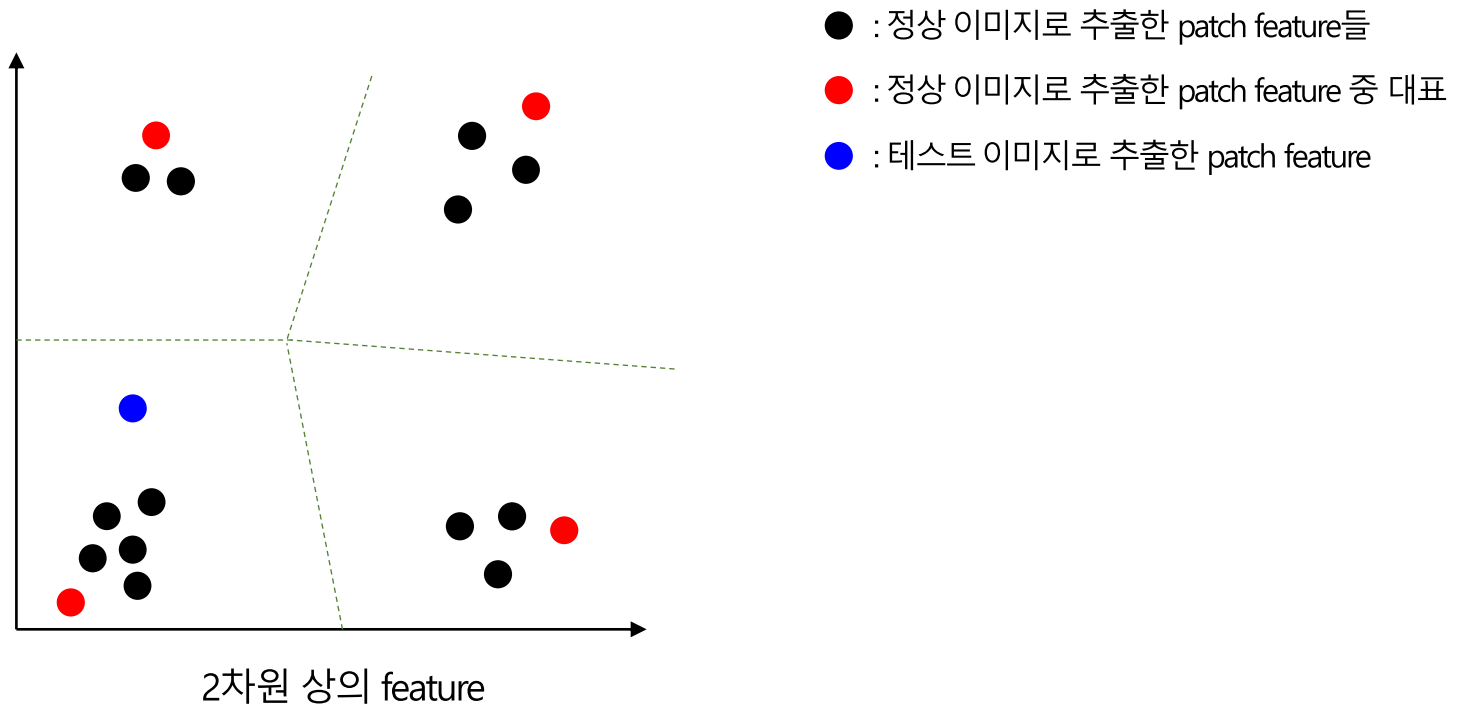


논문에 기재된 그림

# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- KNN 거리 기반 anomaly score 사용
- 이웃 중 가장 먼 거리를 anomaly score로 사용

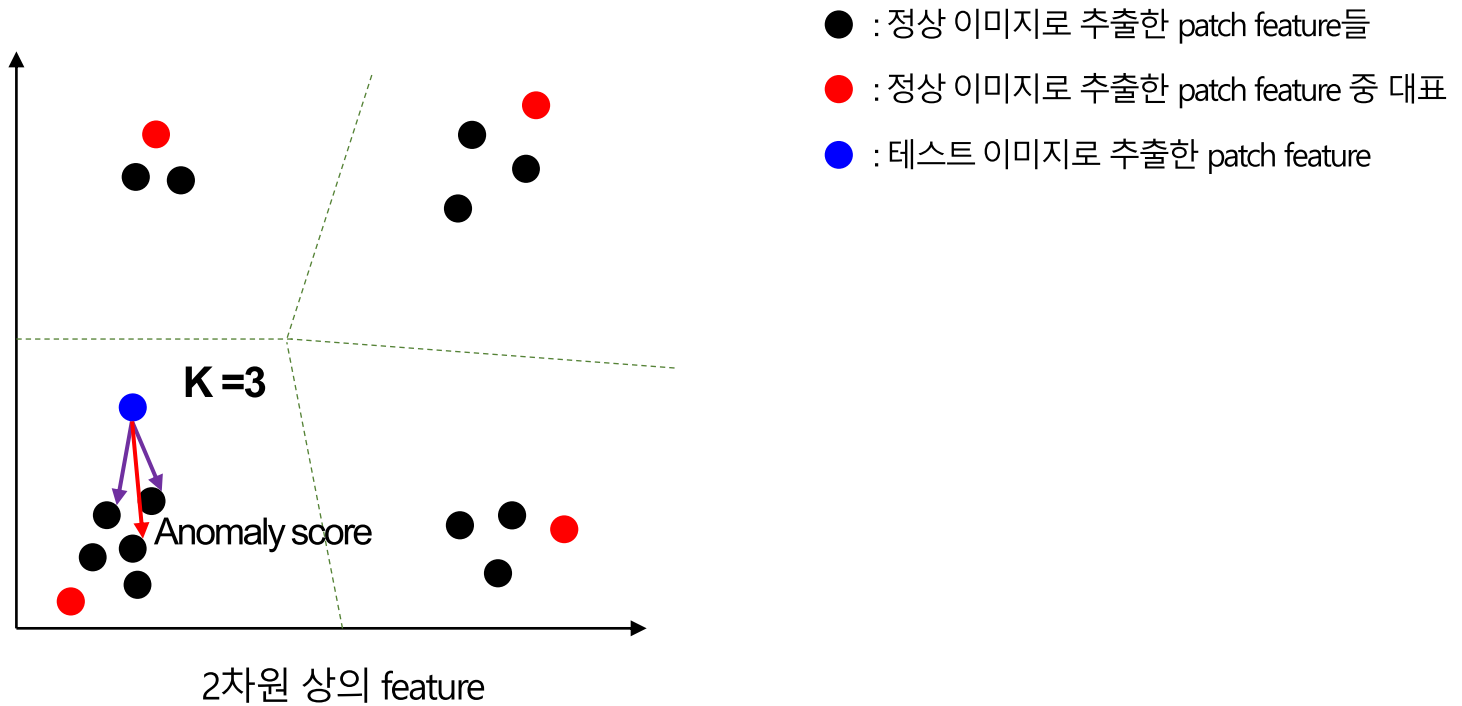




# Method

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- KNN 거리 기반 anomaly score 사용
- 이웃 중 가장 먼 거리를 anomaly score로 사용



# Experiments

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- Memory bank를 1%만 사용해도 기존 방법론에 비해 높은 성능
- Image-AUROC, Pixel-AUROC 그리고 AUPRO를 평가 지표로 사용
- PatchCore – n%: 특징의 n%만큼의 coreset을 사용
- 모든 평가 지표는 높을 수록 좋은 지표임

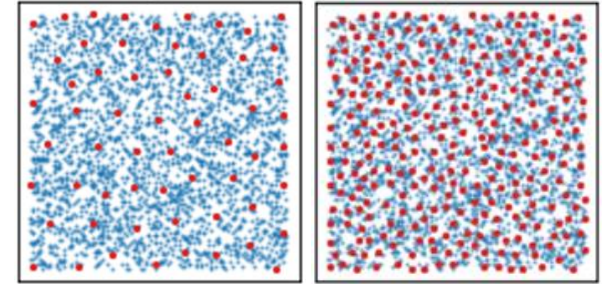


Image-AUROC

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

Pixel-AUROC

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

AUPRO

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

# Few-shot Anomaly Detection

## ❖ Few-shot Anomaly Detection

- 적은 수의 정상 이미지만을 활용

### One-class Anomaly Detection



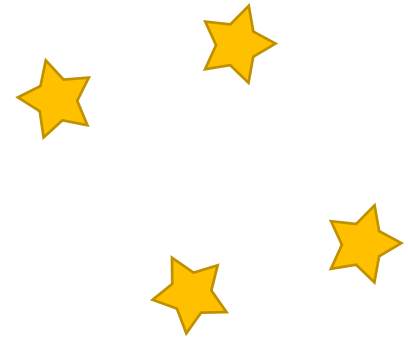
- 학습 시, 레이블링 된 정상 데이터 사용한 경우
- 정상 데이터만 확보되는 경우 좋음
- 정상 데이터만 확보된다는 보장 X

### Unsupervised Anomaly Detection



- 학습 시, 데이터가 대부분 정상으로 이루어졌다고 **가정**
- 비정상이 포함된 학습 가능성 ↑

### Few-shot Anomaly Detection



- 정상 데이터만을 활용
- 적은 수의 정상 이미지를 활용
- 현실성이 있으면서 확실한 방법

# Few-shot Anomaly Detection

---

- ❖ **FastRecon: Few-shot Industrial Anomaly Detection via Fast Feature Reconstruction**
  - ICCV 2023 게재

## **FastRecon: Few-shot Industrial Anomaly Detection via Fast Feature Reconstruction**

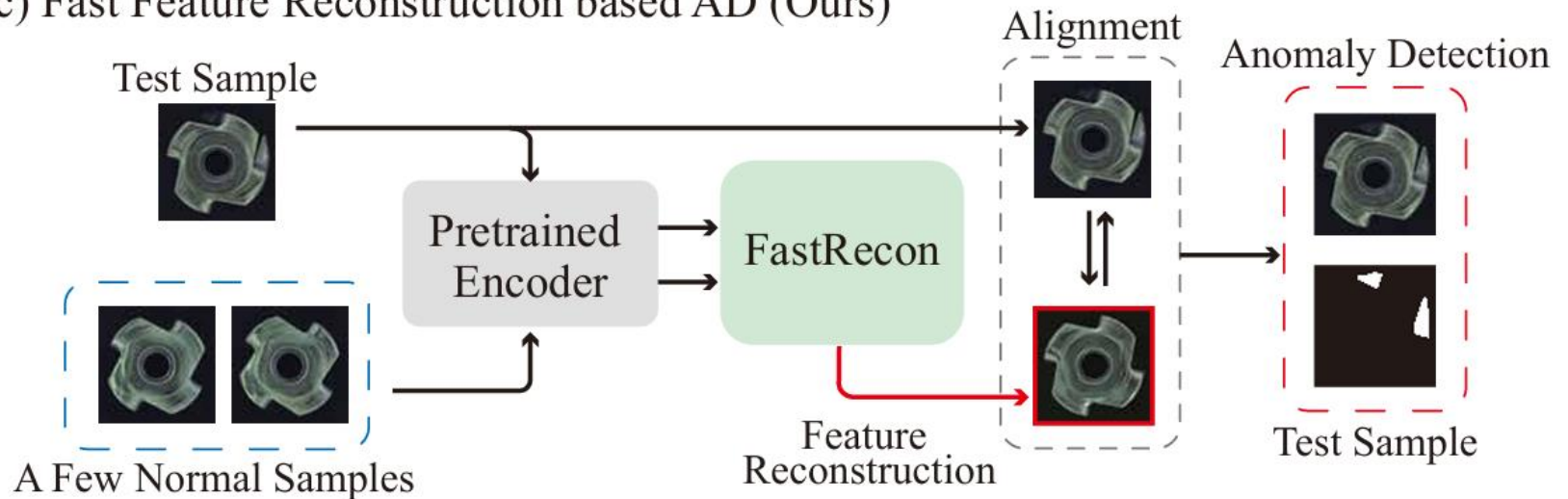
Zheng Fang<sup>1,2†</sup> Xiaoyang Wang<sup>1,2†</sup> Haocheng Li<sup>3</sup> Jiejie Liu<sup>1,3</sup> Qiugui Hu<sup>3</sup> Jimin Xiao<sup>1\*</sup>  
<sup>1</sup>XJTU <sup>2</sup>Metavisioncn <sup>3</sup>Dinnar Automation Technology  
{fz.jun26th, haocheng.li789}@gmail.com,  
{xiaoyang.wang20, jiejie.liu22}@student.xjtu.edu.cn, jimmin.xiao@xjtu.edu.cn

# Few-shot Anomaly Detection

## ❖ Motivation

- 실제 산업 환경에서 정상 데이터만으로도 빠르게 이상을 탐지하는 이상탐지 기법이 필요
  - ✓ Few-shot: 정상 데이터만으로도 빠르게 이상 탐지 가능
  - ✓ Training-free: 별도의 학습이 필요 없어 빠른 추론
  - ✓ 분포 정규화 기반 회귀: 새로운 정규화 항 도입

### (c) Fast Feature Reconstruction based AD (Ours)



[FastRecon 방법론]

# Few-shot Anomaly Detection

## ❖ Previous Industrial Anomaly Detection

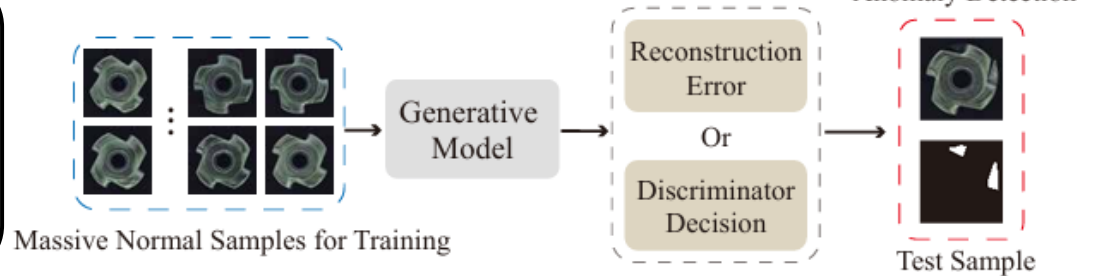
- 기존 industrial anomaly detection들은 대부분 복잡성이 높거나 계산 비용이 높음

### 재구성 기반 방법론

오토인코더 등 생성 모델로  
정상 데이터 분포 학습

**복잡한 학습 과정과 대량의 정상 데이터 요구**

(a) Vanilla Reconstruction-based AD

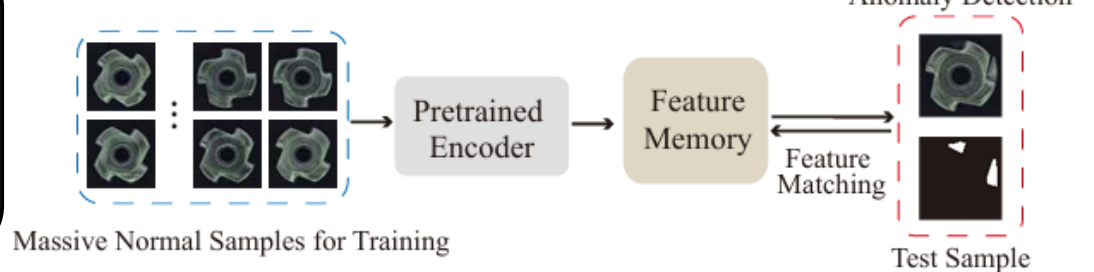


### 특징 기반 방법론

사전 학습된 네트워크에서 추출한  
특징을 이용, 정상 메모리 बैं크

**피쳐 검색 과정 계산 비용이 높음, 복잡성 높음**

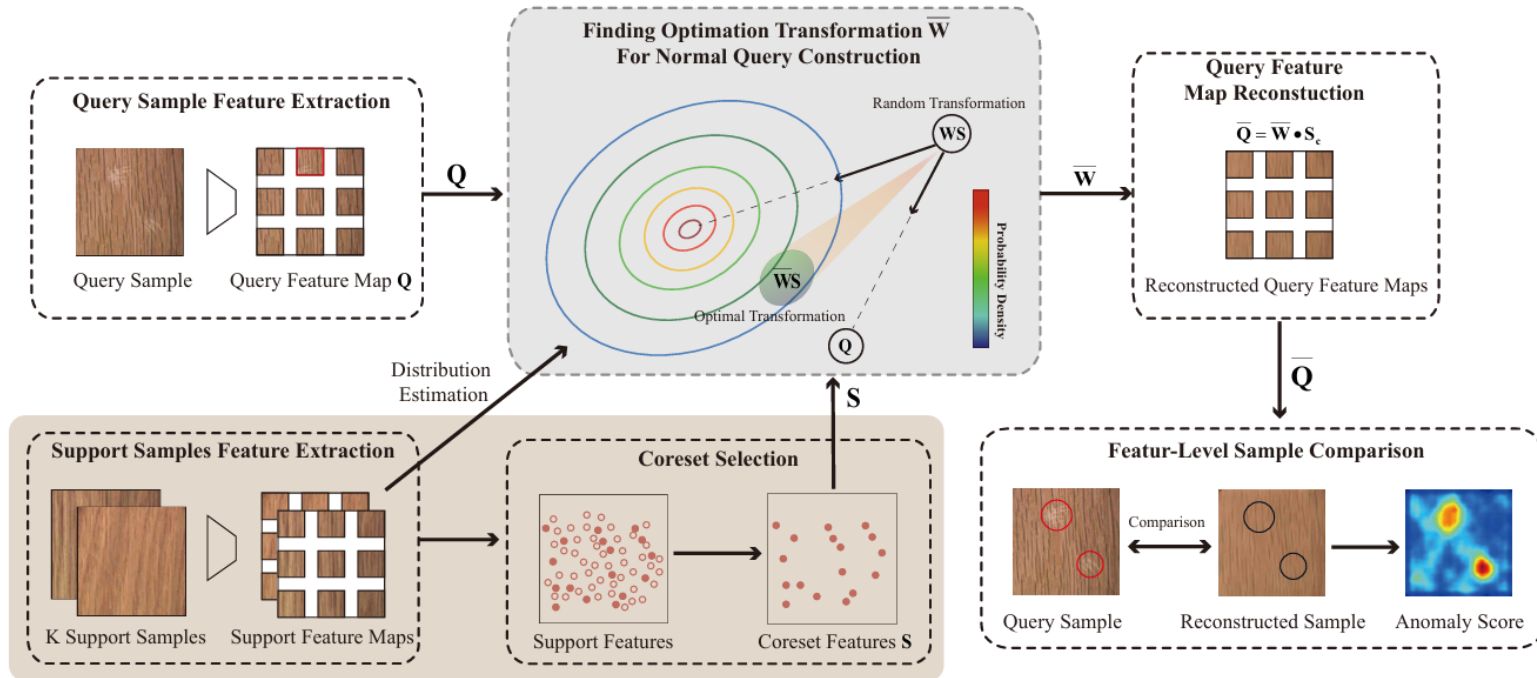
(b) Vanilla Feature-matching-based AD



# Method

## ❖ 3 composition of FastRecon

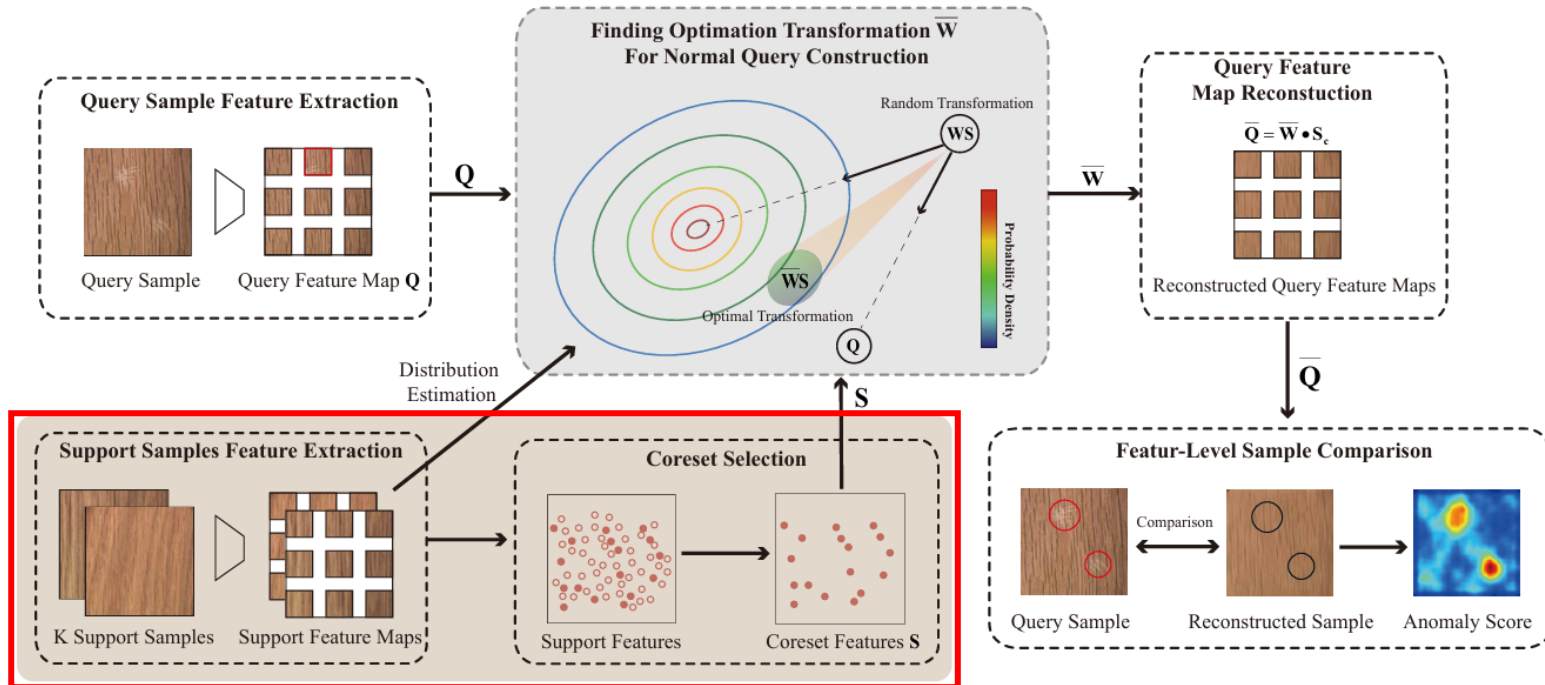
- 특징 추출 및 코어셋 선택 과정
- 분포 정규화 기반 회귀
- 재구성 기반 이상탐지



# Method

## ❖ 3 composition of FastRecon

- 특징 추출 및 코어셋 선택 과정
  - ✓ K개의 정상 이미지들에서만 대표 특징을 추출하여 시간과 비용을 줄임
  - ✓ 코어셋 사용으로 비용을 다시 한번 줄임

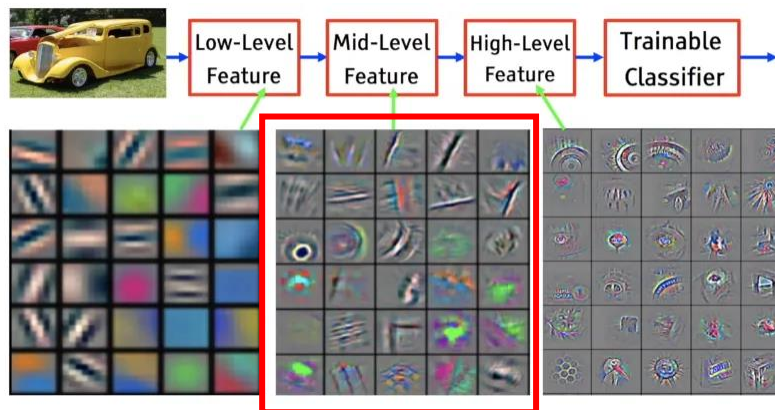
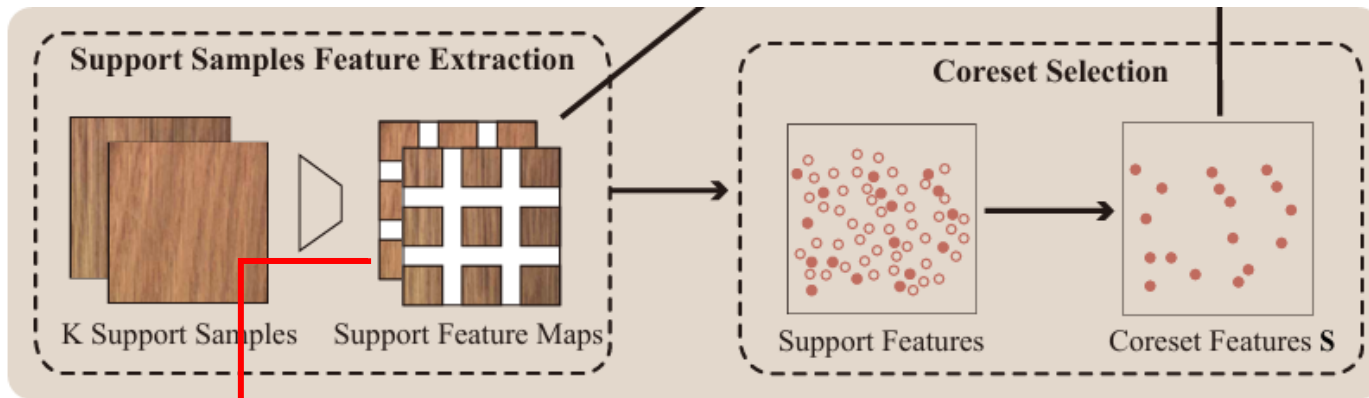




# Method

## ❖ 특징 추출 및 코어셋 선택 과정

- 특징 추출
  - ✓ 사전학습 된 ResNet-50을 사용하여 이미지로 부터 중간 계층의 특징맵 추출

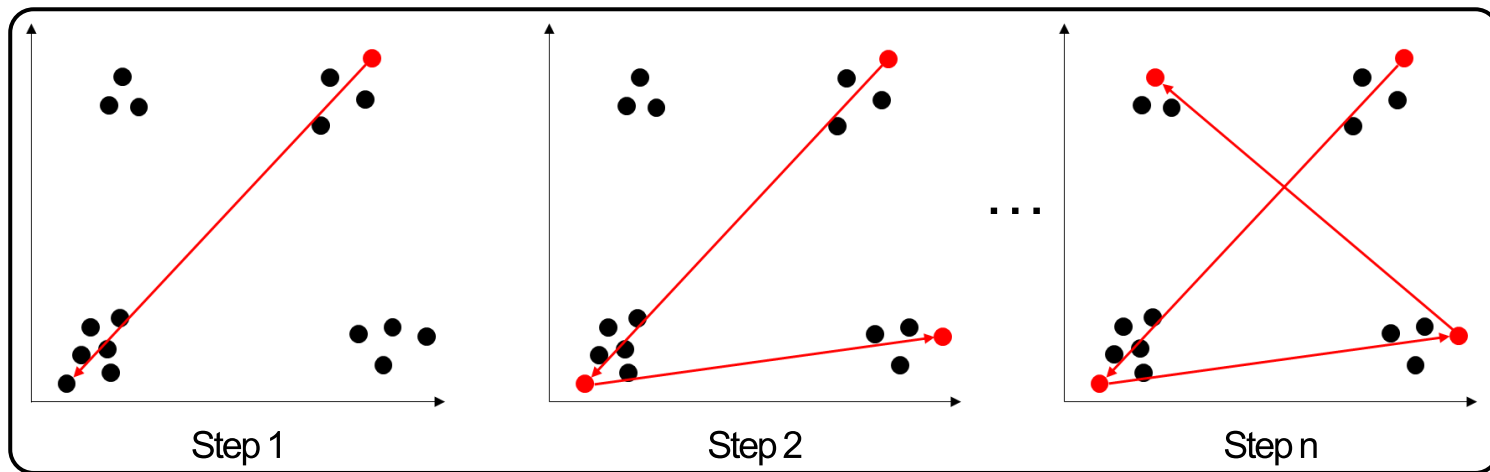
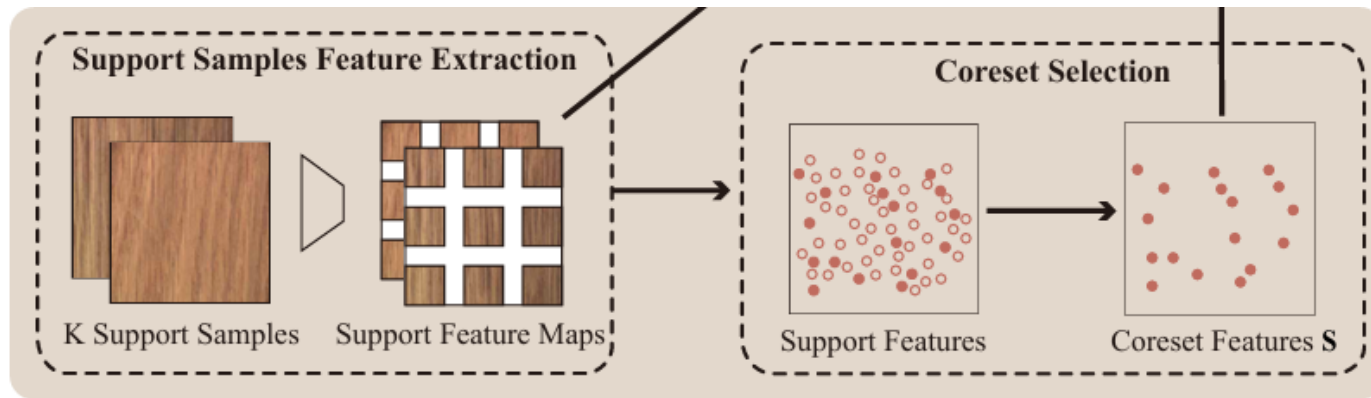


- High level feature은 위치 정보가 너무 많이 사라짐
- 특정 task에 너무 특화된 정보만 남음

# Method

## ❖ 특징 추출 및 코어셋 선택 과정

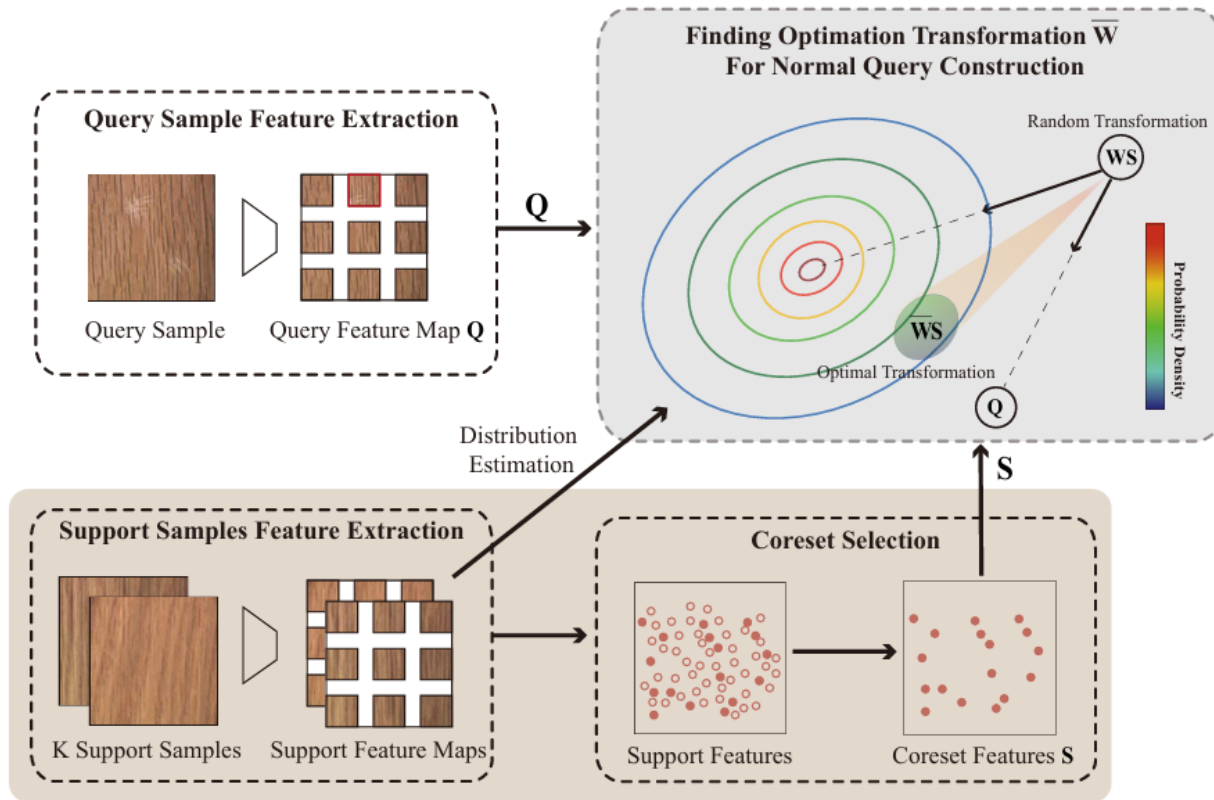
- 코어셋 선택 과정
  - ✓ 피쳐맵: 2차원 행렬로 재구성 ( $H \times W \times D \rightarrow R \times D$ )
  - ✓ 코어셋 선택: 소수의 정상 이미지 특징들 중, 대표성이 높은 특징



# Method

## ❖ 분포 정규화 기반 회귀

- 정상 코어셋 피쳐 (S)를 이용하여 주어진 쿼리 이미지에서 추출한 Q와 유사한 찾기 위함
- 결함 부분이 포함되지 않은 정상 버전의 피쳐 Q를 재구성하기 위한 변환행렬 W를 찾고자 함

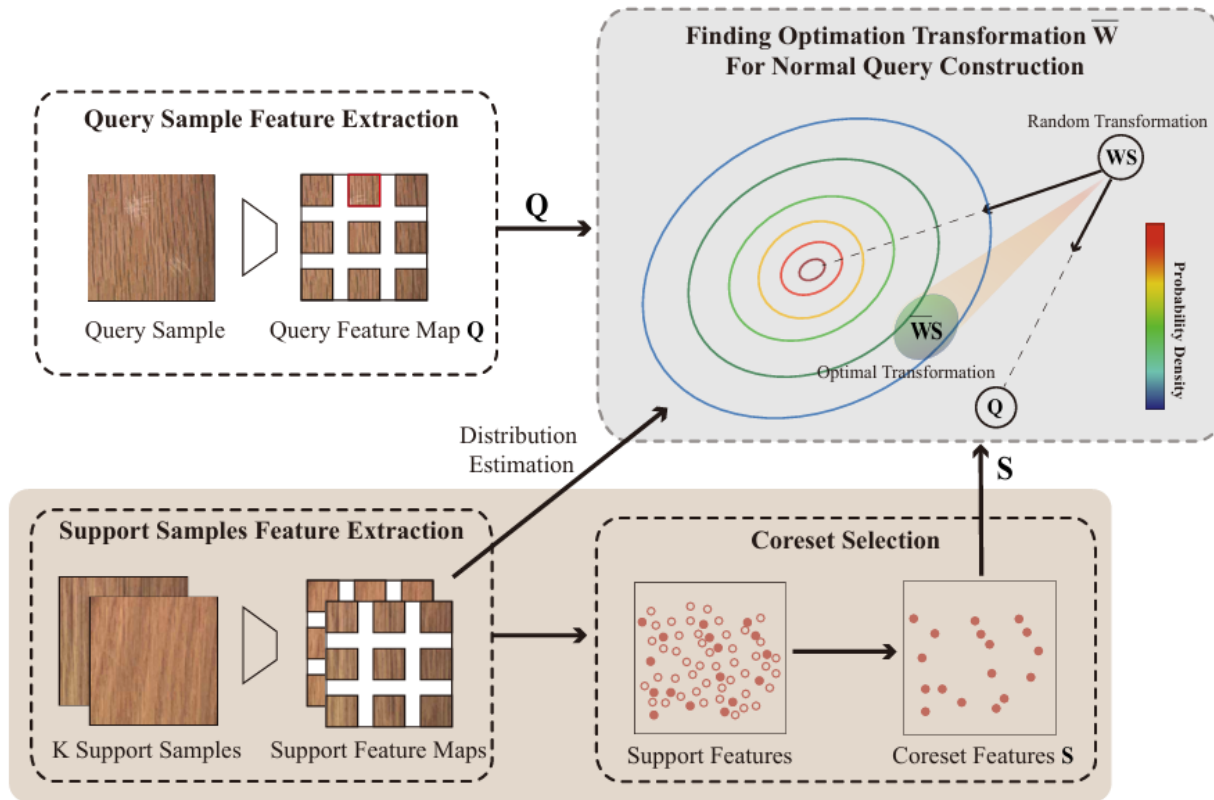




# Method

## ❖ 분포 정규화 기반 회귀

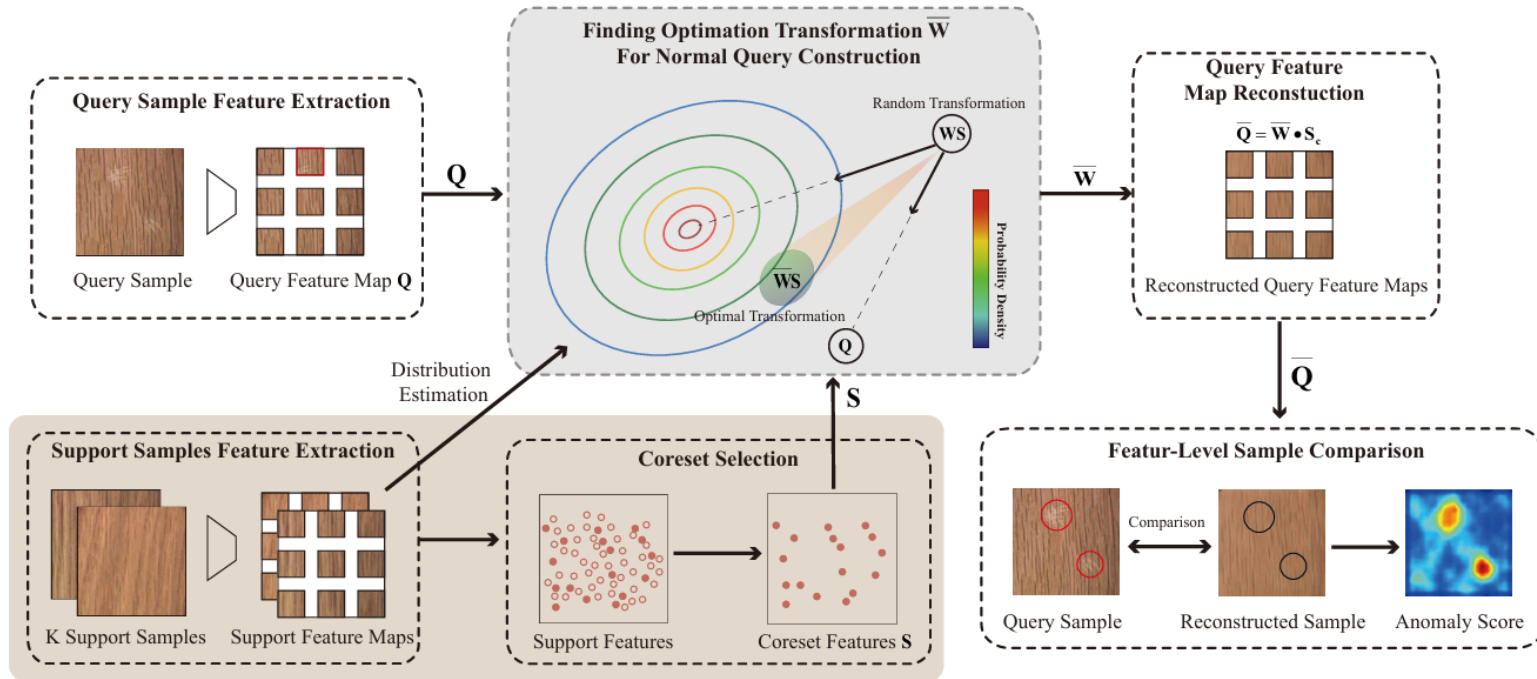
- 단순히 원본 쿼리와 가까워 지는 것을 넘어, **정상 데이터의 특성을 유지하도록 강제함**
- 이상 영역이 두드러짐



# Method

## ❖ 재구성 기반 이상탐지

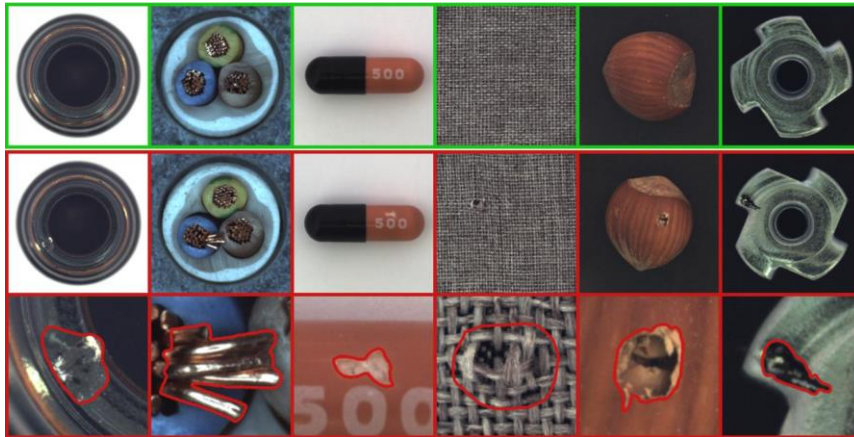
- 최적의 변환 ( $\bar{W}$ )을 사용하여 정상 샘플 기반으로 쿼리 이미지 정상 버전 재구성 ( $\bar{Q} = \bar{W}S$ )



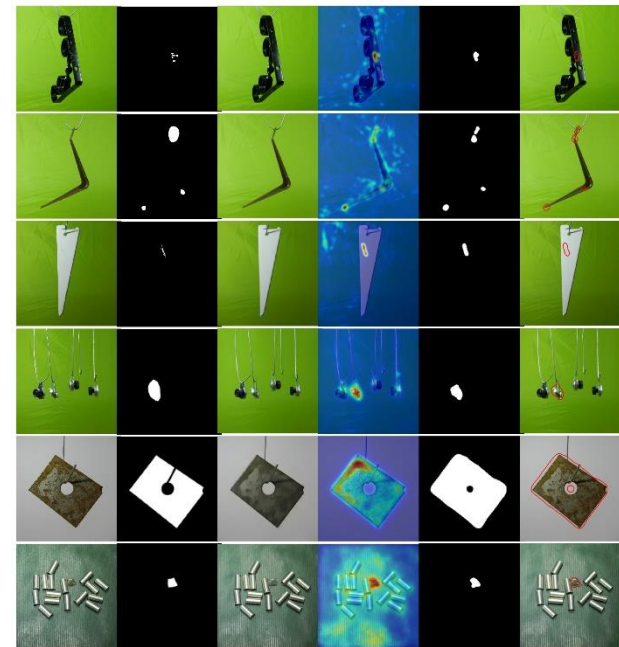
# Experiments

## ❖ 데이터셋 및 평가 지표

- 데이터셋
  - ✓ MVTec AD: 15 카테고리
  - ✓ MPDD: 금속 제품 이미지로 이루어진 데이터셋
- 평가지표
  - ✓ AUROC: 이미지 단위 및 픽셀 단위로 성능 평가



MVTec



MPDD

# Experiments

## ❖ Few-shot 기반 AD와 비교

- 2, 4, 8 shot에 대한 실험 진행
- 모든 카테고리에 대한 AUROC 평균값
- 모든 지표는 높을 수록 좋음
- 모든 k 셋팅에 대해 가장 우수한 성능

Dataset	k	AUROC at Image-level						AUROC at Pixel-level			
		TDG [23]	DiffNet [19]	RegAD [13]	PaDiM [7]	PatchCore [18]	FastRcon (ours)	RegAD [13]	PaDiM [7]	PatchCore [18]	FastRcon (ours)
MVTec	2	71.20	80.60	85.70	78.90	87.81	<b>90.97</b>	94.60	90.50	94.75	<b>95.86</b>
	4	72.70	81.30	88.20	71.60	89.49	<b>94.24</b>	95.80	80.20	94.99	<b>96.98</b>
	8	75.20	83.20	91.20	75.30	94.31	<b>95.19</b>	96.70	80.50	95.60	<b>97.27</b>
MPDD	2	60.30	60.20	63.40	-	59.55	<b>73.65</b>	93.20	-	79.15	<b>97.03</b>
	4	63.50	63.30	68.30	-	59.78	<b>79.85</b>	93.90	-	79.82	<b>97.60</b>
	8	68.20	68.50	71.90	-	59.95	<b>82.50</b>	95.10	-	80.30	<b>97.92</b>



# Experiments

## ❖ Few-shot 기반 AD와 비교

- 4-shot에 대한 실험 진행
- 두가지 데이터셋의 각 카테고리에 대한 성능
- 대부분 FastRecon이 가장 좋은 성능을 보임

Category	RegAD [13]		PatchCore [18]		FastRecon	
	Image	Pixel	Image	Pixel	Image	Pixel
Bottle	99.40	98.40	<b>99.60</b>	<b>98.60</b>	99.44	98.50
Cable	76.10	92.70	<b>97.40</b>	<b>97.90</b>	93.79	96.12
Capsule	72.40	97.60	66.30	97.70	<b>90.07</b>	<b>98.96</b>
Carpet	97.90	98.90	99.00	99.00	<b>99.90</b>	<b>99.15</b>
Grid	<b>91.20</b>	85.70	63.00	70.60	88.81	<b>86.32</b>
Hazelnut	95.80	98.00	92.80	97.00	<b>99.32</b>	<b>98.59</b>
Leather	<b>100.00</b>	99.10	<b>100.00</b>	96.90	<b>100.00</b>	<b>99.20</b>
Metal_Nut	94.60	97.80	94.70	97.00	<b>99.12</b>	<b>98.72</b>
Pill	80.80	97.40	89.00	96.90	<b>93.48</b>	<b>98.32</b>
Screw	56.60	95.00	54.10	92.10	<b>62.46</b>	<b>97.10</b>
Tile	95.50	94.90	<b>100.00</b>	96.00	<b>100.00</b>	<b>96.73</b>
Toothbrush	90.90	98.50	<b>95.20</b>	98.80	93.61	<b>99.04</b>
Transistor	85.20	93.80	<b>98.40</b>	<b>95.00</b>	97.29	94.18
Wood	98.60	94.70	97.40	93.10	<b>99.29</b>	<b>94.94</b>
Zipper	88.50	94.00	95.50	98.30	<b>96.95</b>	<b>98.89</b>
Average	88.20	95.80	89.49	94.99	<b>94.24</b>	<b>96.98</b>

Table 2: FSAD categorical performance comparisons in AUROC (%)  $\uparrow$  on MVTec dataset. Results are from the setting of shot  $k = 4$ . The best results are in bold. Our method achieves the best performance in most categories.

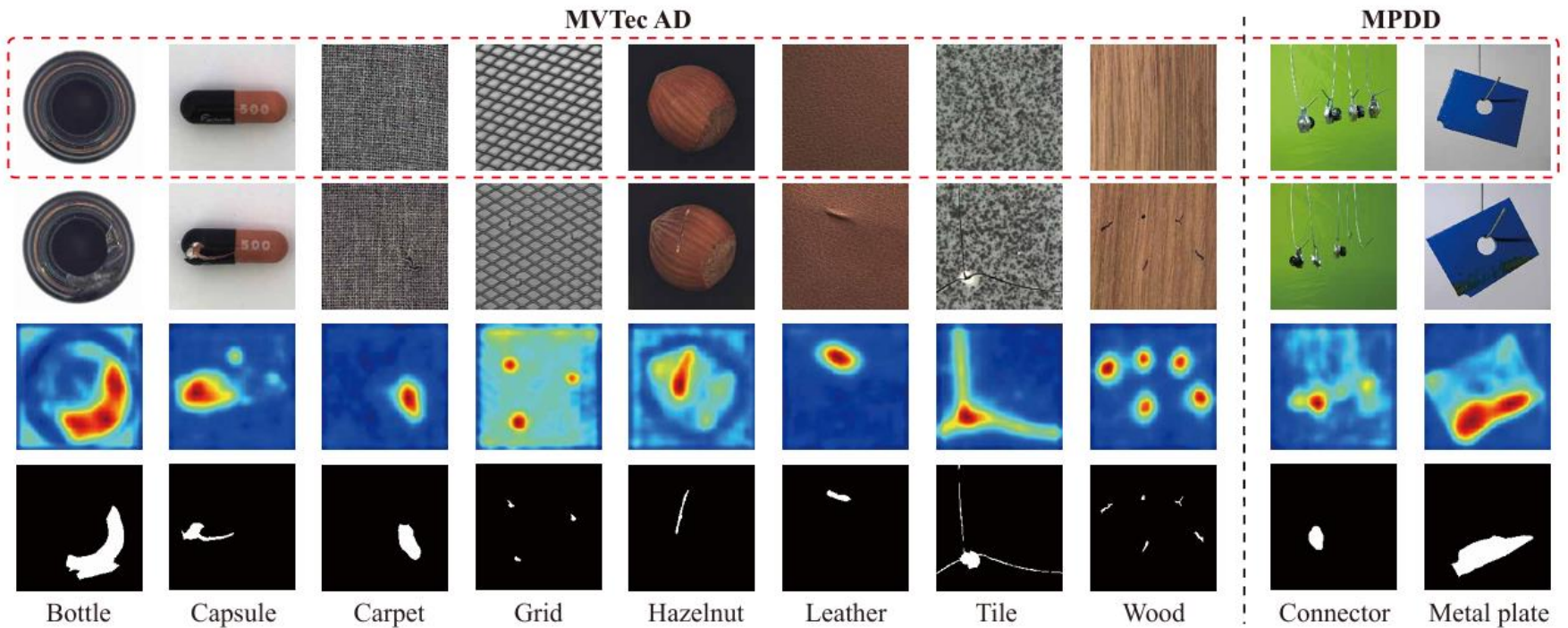
Category	RegAD [13]		PatchCore [18]		FastRecon	
	Image	Pixel	Image	Pixel	Image	Pixel
Bracket_black	63.80	-	58.90	79.10	<b>71.81</b>	<b>95.54</b>
Bracket_brown	66.10	-	<b>70.80</b>	77.30	63.34	<b>95.93</b>
Bracket_white	59.30	-	<b>70.70</b>	69.30	69.44	<b>98.89</b>
Connector	77.20	-	59.40	86.40	<b>97.62</b>	<b>98.04</b>
Metal_plate	78.60	-	64.40	86.70	<b>100.00</b>	<b>99.29</b>
Tubes	67.50	-	34.50	80.10	<b>76.90</b>	<b>97.89</b>
Average	68.30	93.90	59.78	79.82	<b>79.85</b>	<b>97.60</b>

Table 3: FSAD categorical performance comparisons in AUROC (%)  $\uparrow$  on MPDD dataset. Results are from the setting of shot  $k = 4$ . The best results are in bold. Our method achieves the best performance in most categories.

# Experiments

## ❖ Few-shot 기반 AD와 비교

- 이상을 잘 탐지하고 있는 모습을 확인



# Zero-shot Anomaly Detection

## ❖ Zero-shot Anomaly Detection (ZSAD)

- 학습 데이터와 테스트 데이터가 서로 다른 도메인일 때 적용 가능

### One-class Anomaly Detection



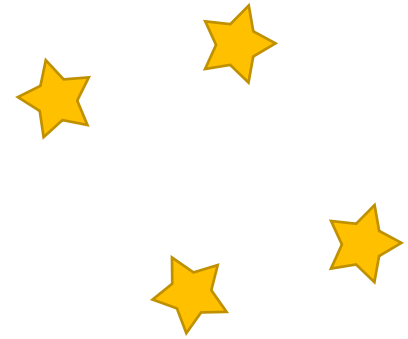
- 학습 시, 레이블링 된 정상 데이터 사용한 경우
- 정상 데이터만 확보되는 경우 좋음
- 정상 데이터만 확보된다는 보장 X

### Unsupervised Anomaly Detection



- 학습 시, 데이터가 대부분 정상으로 이루어졌다고 가정
- 현실과 가장 유사한 상황
- 비정상이 포함된 학습 가능성 ↑

### Few-shot Anomaly Detection



- 정상 데이터만을 활용
- 적은 수의 정상 이미지를 활용
- 현실성이 있으면서 확실한 방법

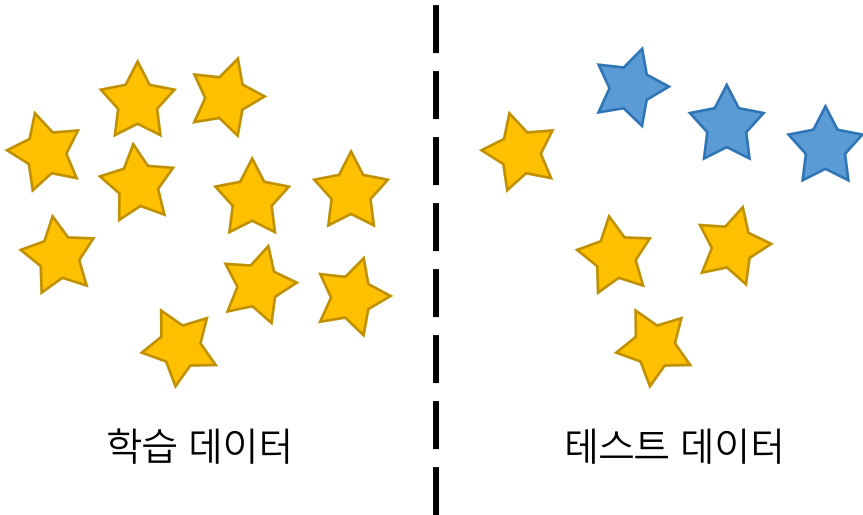
# Zero-shot Anomaly Detection

## ❖ Zero-shot Anomaly Detection (ZSAD)

- 학습 데이터와 테스트 데이터가 서로 다른 도메인일 때 적용 가능
  - ✓ 데이터 프라이버시 정책으로 학습데이터를 활용할 수 없는 경우
  - ✓ 타겟 도메인과 관련된 학습 데이터를 갖고 있지 못한 경우

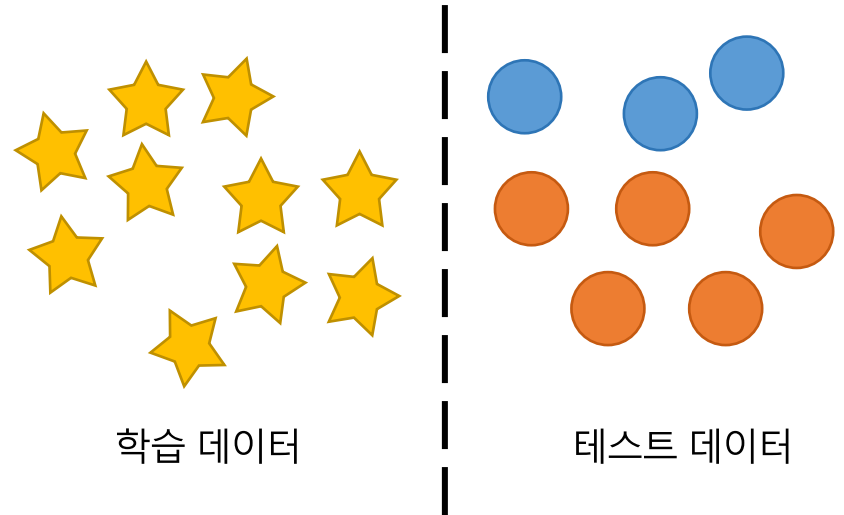
기존 방법론

- 학습 데이터와 테스트 데이터의 도메인이 같음



Zero-shot 방법론

- 학습 데이터와 테스트 데이터의 도메인이 다름



# Zero-shot Anomaly Detection

## ❖ Zero-shot Anomaly Detection (ZSAD)

- 학습 데이터와 테스트 데이터가 서로 다른 도메인일 때 적용 가능
  - ✓ 데이터 프라이버시 정책으로 학습데이터를 활용할 수 없는 경우
  - ✓ 타겟 도메인과 관련된 학습 데이터를 갖고 있지 못한 경우

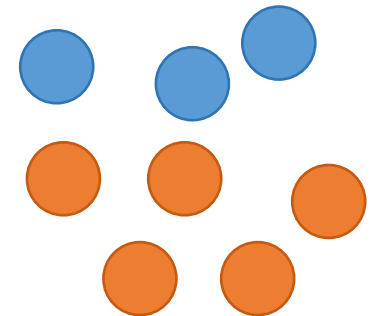
다양한 도메인에 걸쳐 이상치를  
어떻게 일반화해야 할 것인가?

### Zero-shot 방법론

- 학습 데이터와 테스트 데이터의 도메인이 다름



학습 데이터

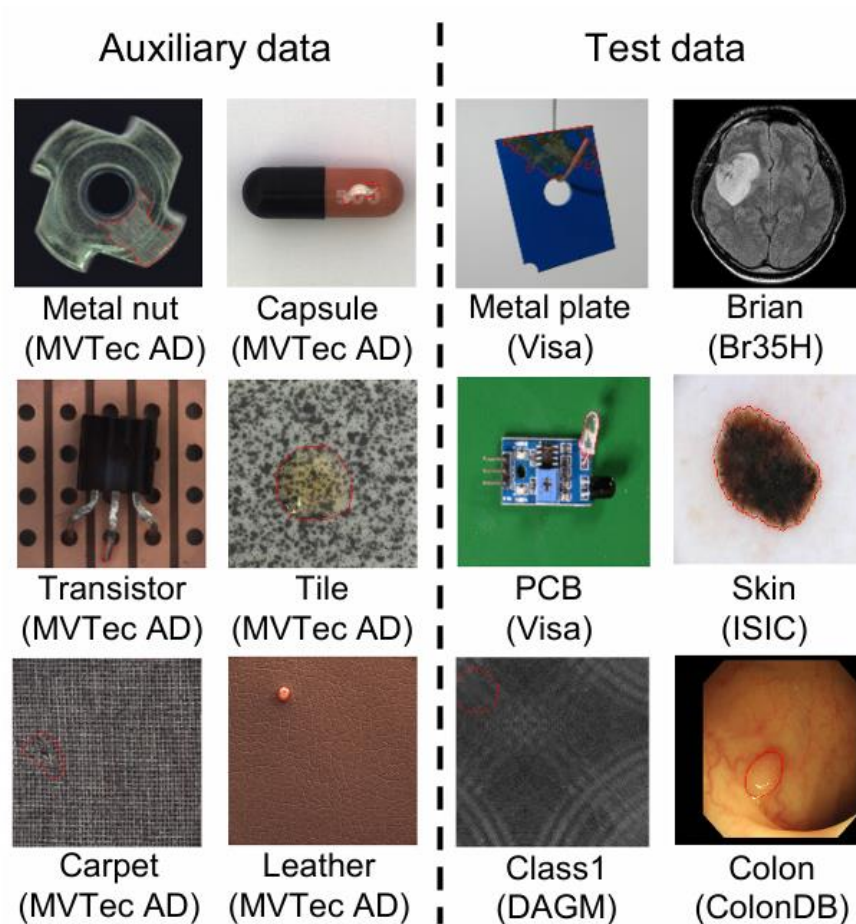


테스트 데이터

# Zero-shot Anomaly Detection

## ❖ Zero-shot Anomaly Detection (ZSAD)

- 테스트 이미지에는 학습에 사용한 이미지와 유사한 이상 패턴을 가지고 있는 경우가 있음
- 이상 영역을 탐지할 수 있을 것이라는 가정



# Zero-shot Anomaly Detection

---

- ❖ **AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection**
  - ICLR 2024 게재

## ANOMALYCLIP: OBJECT-AGNOSTIC PROMPT LEARNING FOR ZERO-SHOT ANOMALY DETECTION

**Qihang Zhou<sup>1\*</sup>, Guansong Pang<sup>2\*</sup>, Yu Tian<sup>3</sup>, Shibo He<sup>1†</sup>, Jiming Chen<sup>1†</sup>**

<sup>1</sup>Zhejiang University   <sup>2</sup>Singapore Management University   <sup>3</sup>Harvard University

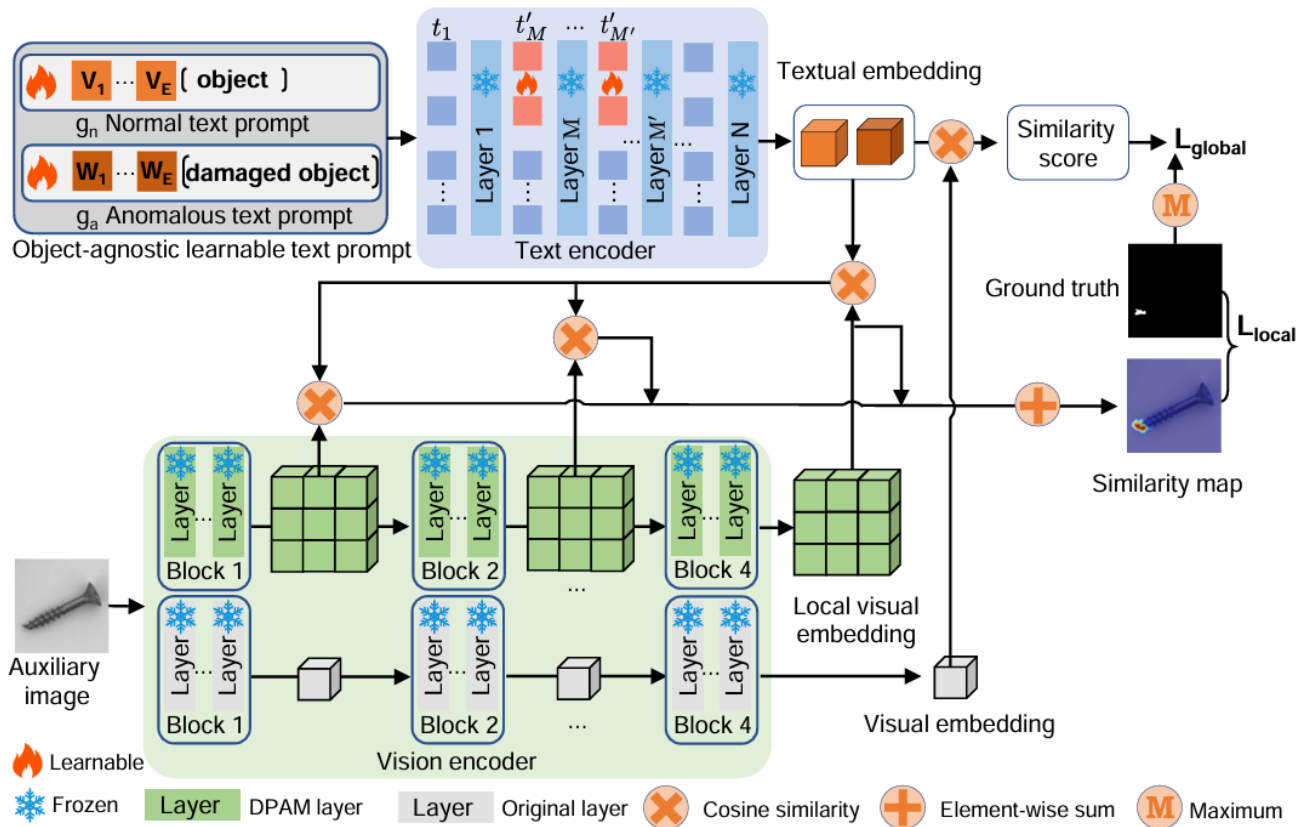
<sup>1</sup>{zqhang, s18he, cjm}@zju.edu.cn   <sup>2</sup>gspang@smu.edu.sg

<sup>3</sup>ytian11@meei.harvard.edu

# Method

## ❖ Overview of AnomalyCLIP

- Contrastive Language-Image Pre-Training (CLIP) 모델을 활용
- 이미지의 전체적인 특징과 세부적인 특징을 동시에 고려해서 텍스트와 함께 학습

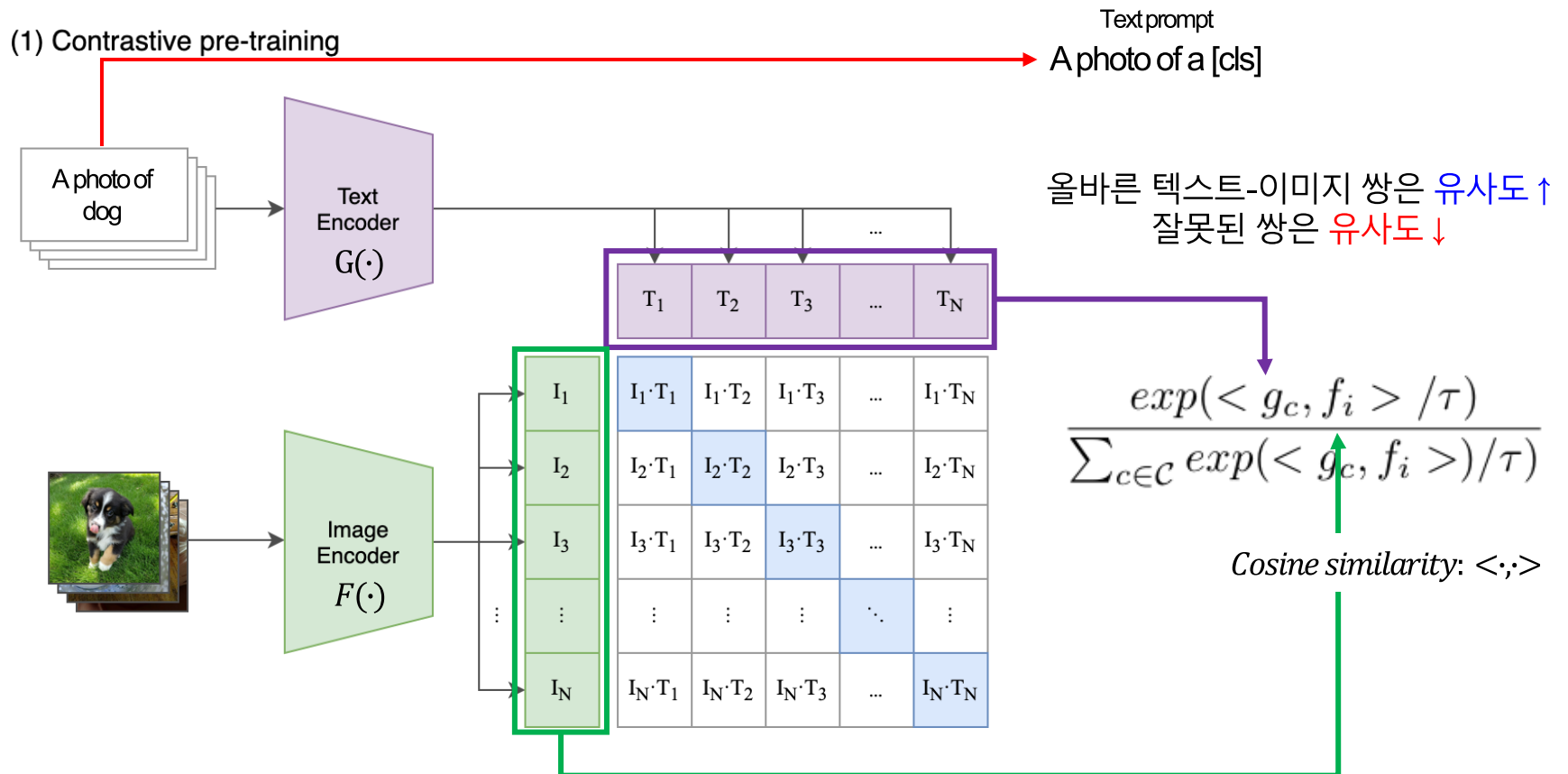




# Method

## ❖ What is CLIP?

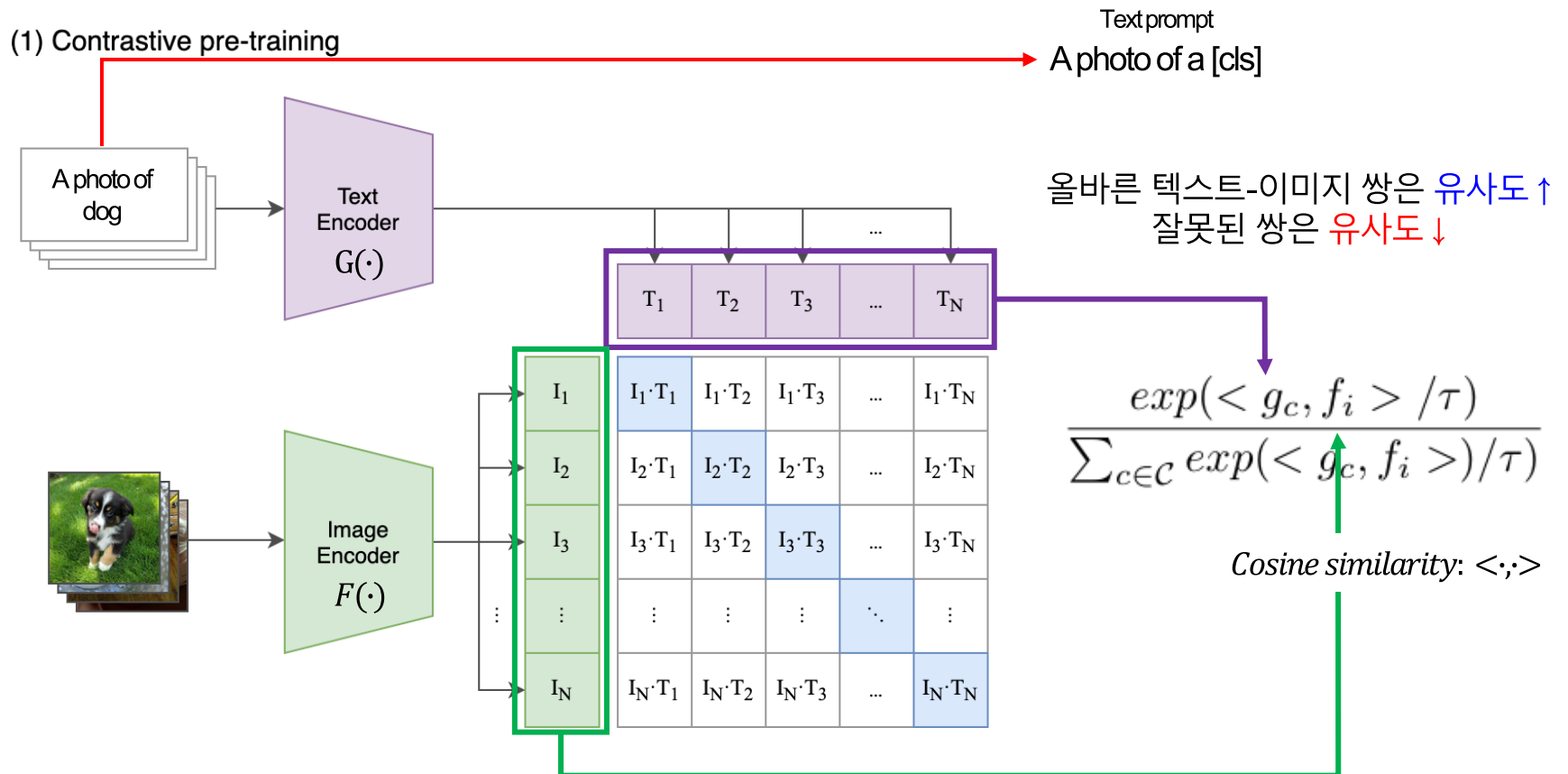
- 거대 vision language model인 CLIP을 활용
- 강력한 일반화 성능을 보이기 때문에 ZSAD 연구들에서 활발하게 사용



# Method

## ❖ CLIP in Anomaly Detection

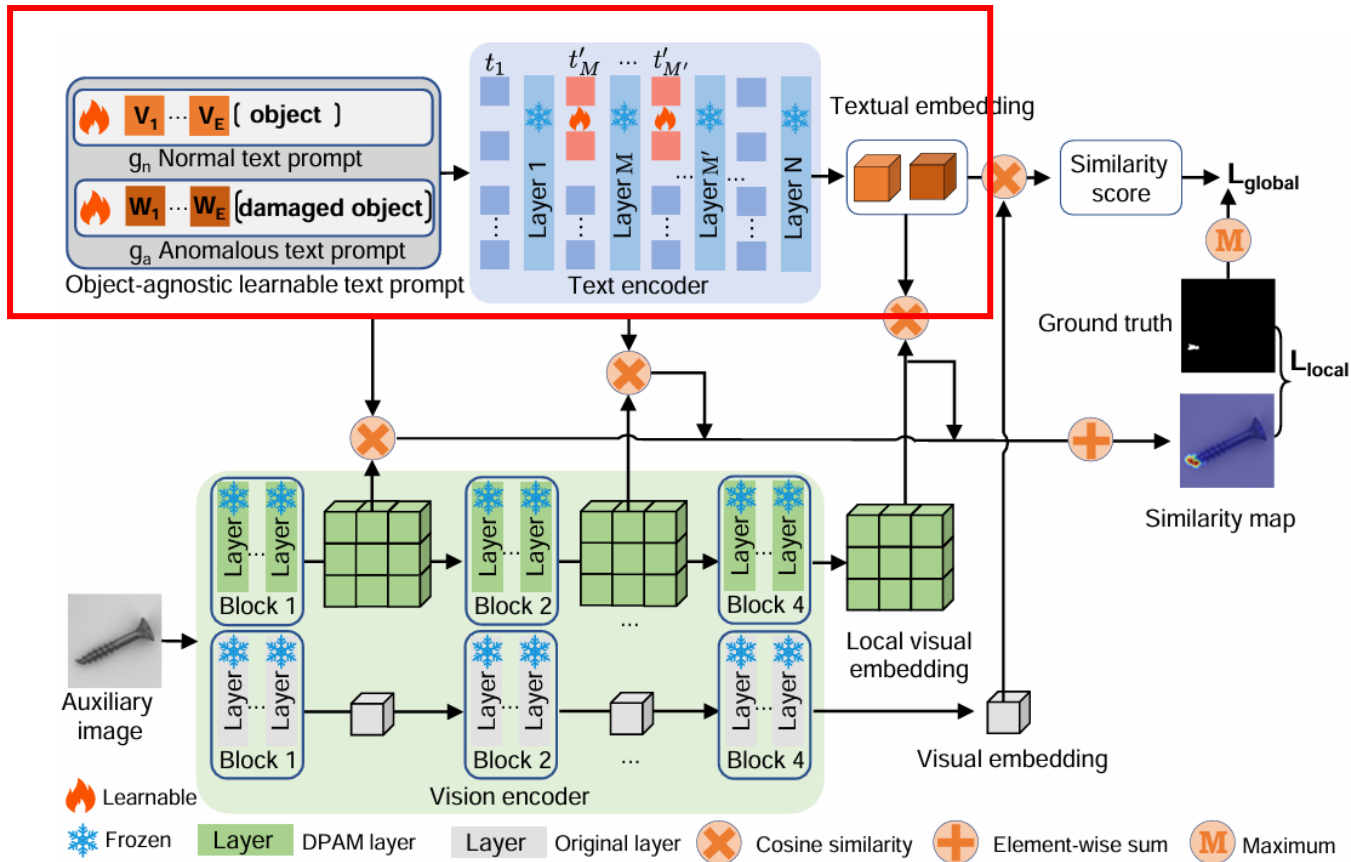
- 이미지 내 **정상/이상 구별보다는 어떤 class인지 분류하는데 특화**되어 있음
- **Anomaly Detection을 수행할 수 있는 전략**이 필요



# Method

## ❖ Overview of AnomalyCLIP

- Contrastive Language-Image Pre-Training (CLIP) 모델을 활용
- 이미지의 전체적인 특징과 세부적인 특징을 동시에 고려해서 텍스트와 함께 학습



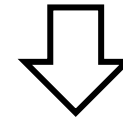
# Method

## ❖ Object-Aware Text Prompt

- CLIP에서 사용하는 텍스트 프롬프트 “A photo of [cls]”는 이상치 탐지에 맞지 않음
  - ✓ 이상치 유형에 대한 프롬프트로 변경
  - ✓ A phot of [cls] with scratches
- 더욱 일반화된 이상치 탐지용 프롬프트가 필요함



CLIP text prompt: “A photo of capsule”



“A photo of capsule with scratches”

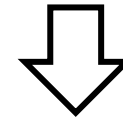
# Method

## ❖ Object-Aware Text Prompt

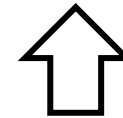
- CLIP에서 사용하는 텍스트 프롬프트 “A photo of [cls]”는 이상치 탐지에 맞지 않음
  - ✓ 이상치 유형에 대한 프롬프트로 변경
  - ✓ A phot of [cls] with scratches
- 더욱 일반화된 이상치 탐지용 프롬프트가 필요함



CLIP text prompt: “A photo of capsule”



“A photo of capsule with scratches”



**불가능함**

수많은 이상치 유형들

scratch, hole, crack, dent, contamination, misplacement, col or deviation, surface defect ...

# Method

---

## ❖ Object-Aware Text Prompt

- $[V]_i, [W]_i$ : 정상과 비정상 텍스트 프롬프트 템플릿의 학습 가능한 word embedding
- “[*damaged*][*cls*]”를 적용해 결함을 쉽게 탐지할 수 있는 포괄적인 anomaly semantic 프롬프트 적용

정상에 대한 프롬프트: “[ $V_1$ ][ $V_2$ ] ... [ $V_E$ ][*cls*]”

이상치에 대한 프롬프트: “[ $W_1$ ][ $W_2$ ] ... [ $W_E$ ][*damaged*][*cls*]”

# Method

---

## ❖ Object-Agnostic Text Prompt

- `[cls]`를 입력하는 것 또한 ZSAD의 성능을 떨어뜨릴 우려가 있음
- Object-Agnostic한 프롬프트 제안 → 객체에 비의존적인 프롬프트
- 데이터셋에 이상치 유형에 대한 정보가 없거나, 새로운 class에 대한 데이터가 들어와도 강건함

정상에 대한 프롬프트: "[V<sub>1</sub>][V<sub>2</sub>] ... [V<sub>E</sub>][cls]"

이상치에 대한 프롬프트: "[W<sub>1</sub>][W<sub>2</sub>] ... [W<sub>E</sub>][damaged][cls]"

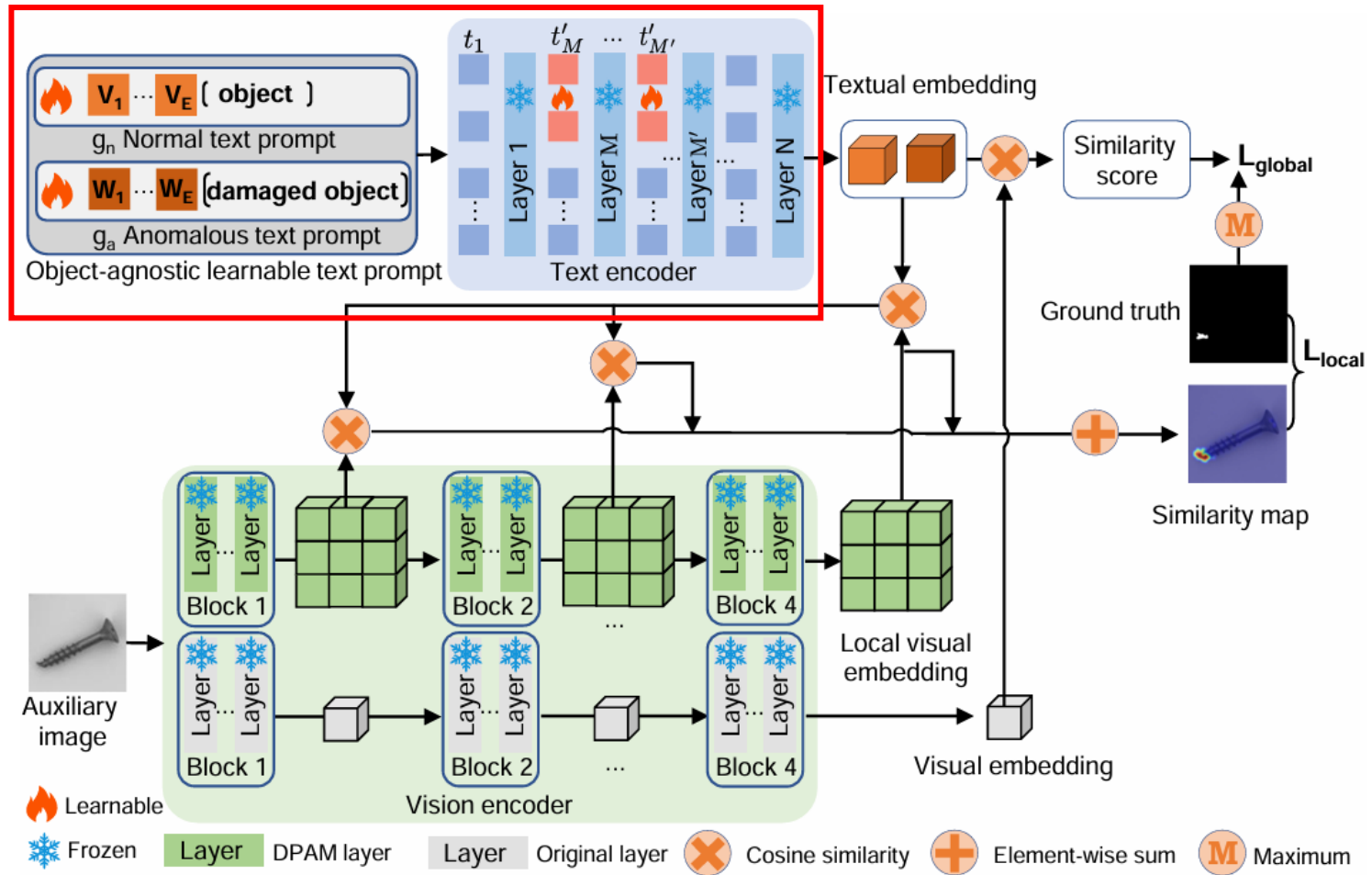


정상에 대한 프롬프트: "[V<sub>1</sub>][V<sub>2</sub>] ... [V<sub>E</sub>][**object**]"

이상치에 대한 프롬프트: "[W<sub>1</sub>][W<sub>2</sub>] ... [W<sub>E</sub>][damaged][**object**]"

# Method

## ❖ Object-Agnostic Text Prompt

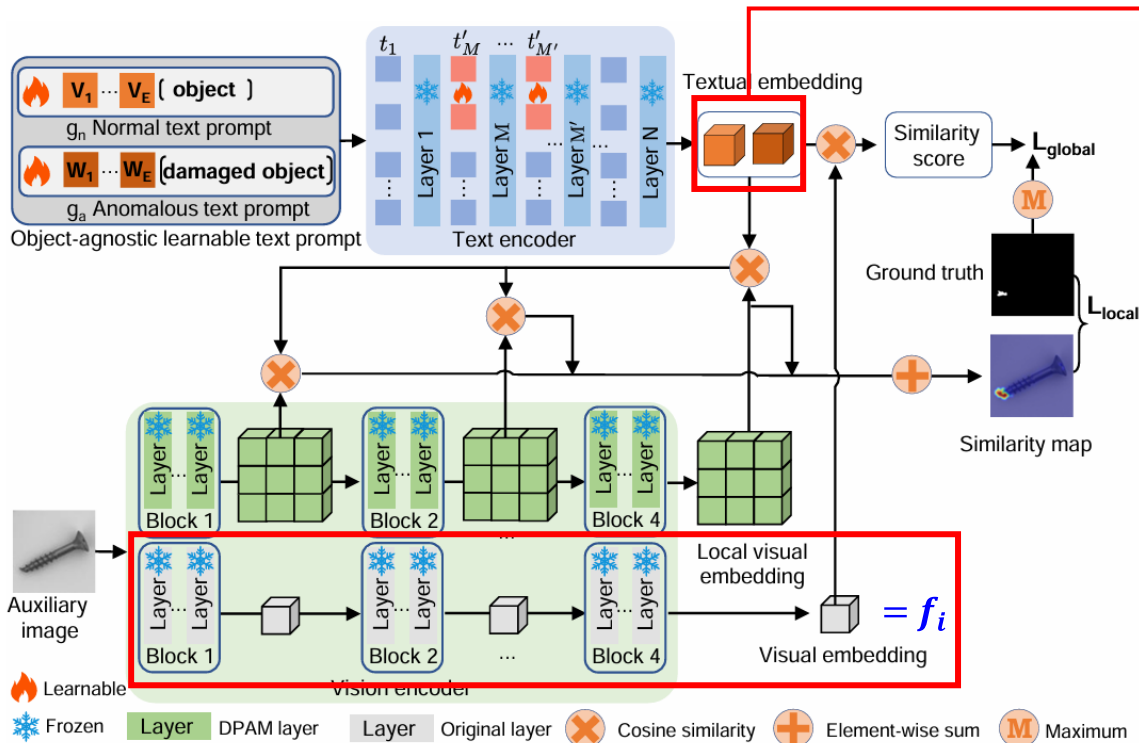




# Method

## ❖ Global context optimization

- 앞선 object-agnostic text prompt를 어떻게 학습할 것인가
- 이상치 탐지는 global과 local 관점에서 이상치를 탐지해야 함 ( $L_{total} = L_{global} + L_{local}$ )
  - ✓ Global loss ( $L_{global}$ ): 이미지 전체의 임베딩과 텍스트 프롬프트 임베딩 간의 코사인 유사도
  - ✓ Local loss ( $L_{local}$ ): 이미지에서 뽑은 각 패치와 텍스트 프롬프트 임베딩 간의 비교



$$\frac{e(\langle g_n, f_i \rangle) / \tau}{(e(\langle g_n, f_i \rangle) / \tau) + (e(\langle g_a, f_i \rangle) / \tau)}$$

정상이냐 이상치냐

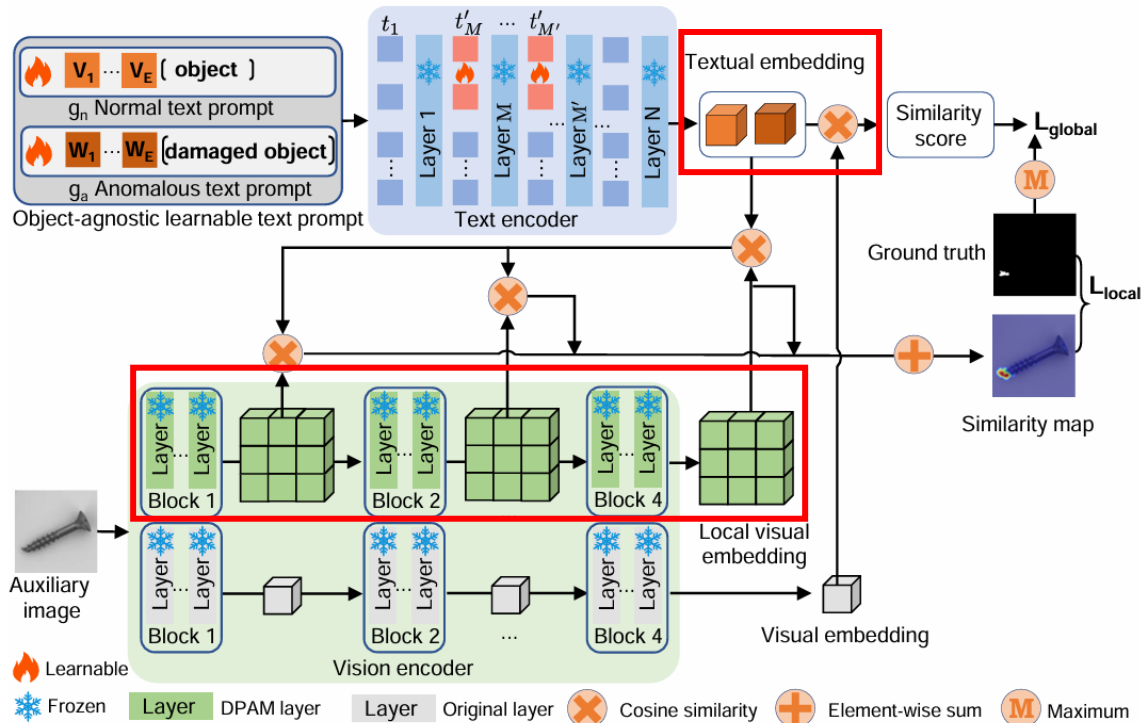
Cosine similarity:  $\langle \cdot, \cdot \rangle$

[Global loss 계산 방법]

# Method

## ❖ Glocal context optimization

- 앞선 object-agnostic text prompt를 어떻게 학습할 것인가
- 이상치 탐지는 global과 local 관점에서 이상치를 탐지해야 함
  - ✓ Global loss ( $L_{global}$ ): 이미지 전체의 임베딩과 텍스트 프롬프트 임베딩 간의 코사인 유사도
  - ✓ Local loss ( $L_{local}$ ): 이미지에서 뽑은 각 패치와 텍스트 프롬프트 임베딩 간의 비교

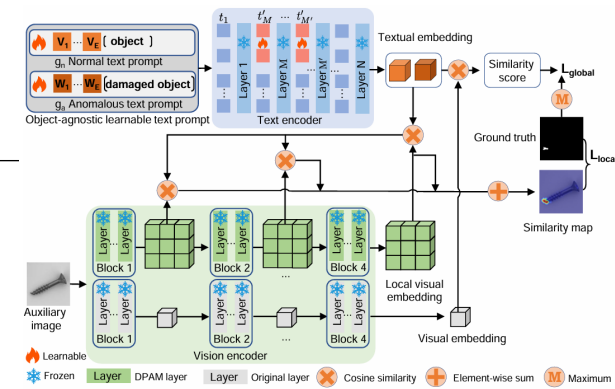


[Local loss 계산 방법]

# Method

## ❖ Glocal context optimization

- 앞선 object-agnostic text prompt를 어떻게 학습할 것인가
- 이상치 탐지는 global과 local 관점에서 이상치를 탐지해야 함
  - ✓ Global loss ( $L_{global}$ ): 이미지 전체의 임베딩과 텍스트 프롬프트 임베딩 간의 코사인 유사도
  - ✓ Local loss ( $L_{local}$ ): 이미지에서 뽑은 각 패치와 텍스트 프롬프트 임베딩 간의 비교



- 텍스트 임베딩
  - ✓  $g_n$ : 정상 프롬프트 임베딩 벡터
  - ✓  $g_a$ : 이상 프롬프트 임베딩 벡터
- 지역 이미지 임베딩
  - ✓  $f_i^{(j,k)}$ : 이미지의 위치  $(j,k)$ 에 해당하는 패치 임베딩
- Ground Truth 이상 마스크  $S$ 
  - ✓  $S_{j,k} = 1 \rightarrow$  이상
  - ✓  $S_{j,k} = 0 \rightarrow$  정상

정상 확률

$$S_n^{(j,k)} = \langle g_n, f_i^{(j,k)} \rangle$$

이상 확률

$$S_a^{(j,k)} = \langle g_a, f_i^{(j,k)} \rangle$$

Cosine similarity:  $\langle \cdot, \cdot \rangle$

$$L_{local} = Focal \left( Up \left( \left[ S_n^{(j,k)} S_a^{(j,k)} \right] \right), S_{j,k} \right) + Dice \left( Up \left( S_n^{(j,k)} \right), 1 - S_{j,k} \right) + Dice \left( Up \left( S_a^{(j,k)} \right), S_{j,k} \right)$$

예측된 값에 대한 확신 정도를 나타냄

정상영역에 대해 겹치는 정도

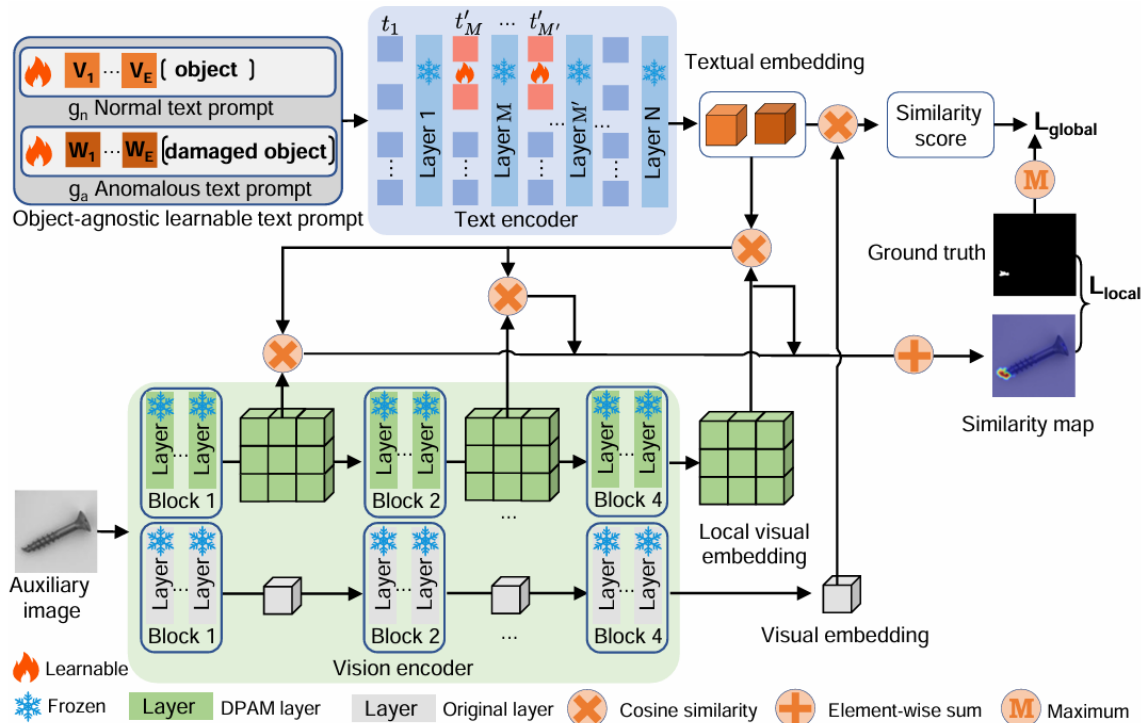
이상영역에 대해 겹치는 정도

[local loss 계산 방법]

# Method

## ❖ Glocal context optimization

- 앞선 object-agnostic text prompt를 어떻게 학습할 것인가
- 이상치 탐지는 global과 local 관점에서 이상치를 탐지해야 함
  - ✓ Global loss ( $L_{global}$ ): 이미지 전체의 임베딩과 텍스트 프롬프트 임베딩 간의 코사인 유사도
  - ✓ Local loss ( $L_{local}$ ): 이미지에서 뽑은 각 패치와 텍스트 프롬프트 임베딩 간의 비교



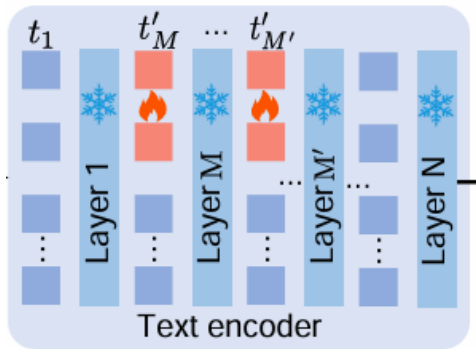
$$L_{total} = L_{global} + \lambda \sum_M L_{local}^M$$

[최종 Loss]

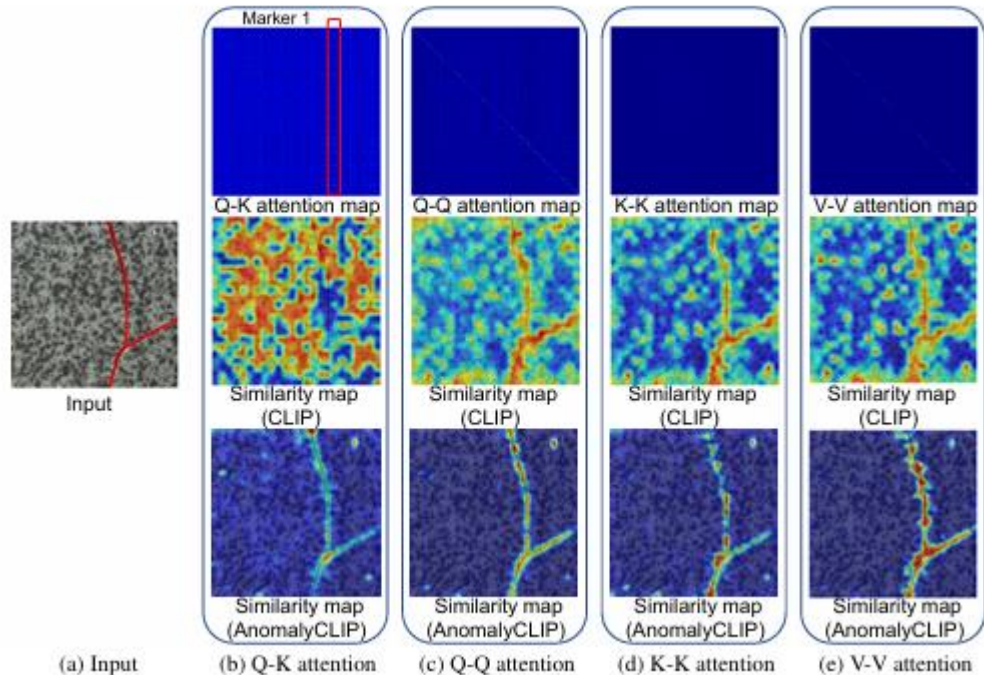
# Method

## ❖ Refinement

- Textual space refinement: 기존 토큰 일부를 학습 가능한 토큰으로 대체
  - ✓ 정상과 이상을 더욱 잘 구별할 수 있도록 미세조정
- Local visual space refinement: Q-K self attention을 Q-Q, K-K, V-V self attention으로 대체 (DPAM)
  - ✓ Attention map이 대각선 중심으로 집중되도록 유도, 특정 위치의 지역적 특징을 잘 유지하도록 함
  - ✓ Segmentation 성능 향상
  - ✓ 특히, V-V가 가장 높은 성능



[Textual space refinement]



[Local visual space refinement]

# Experiments

## ❖ Main result – Industrial domain

- 모든 지표는 높을 수록 좋음
  - ✓ AUROC: 정상과 이상을 얼마나 잘 구분했는지?
  - ✓ AP (Average Precision): 이상이라고 예측한 것들 중 얼마나 맞았는지?
  - ✓ PRO (Per-Region Overlap): 픽셀 단위로 이상 영역의 위치를 얼마나 잘 맞췄는가?
- MVTec AD 테스트 데이터셋으로 미세조정 후 다른 도메인 데이터셋에 대해 실험 진행
- MVTec AD 평가 시에는 VisA 데이터셋의 테스트 데이터로 평가

Task	Category	Datasets	C	CLIP	CLIP-AC	WinCLIP	VAND	CoOp	AnomalyCLIP
Image-level (AUROC, AP)	Obj & texture	MVTec AD	15	(74.1, 87.6)	(71.5, 86.4)	(91.8, 96.5) <sup>†</sup>	(86.1, 93.5) <sup>†</sup>	(88.8, 94.8)	(91.5, 96.2)
		VisA	12	(66.4, 71.5)	(65.0, 70.1)	(78.1, 81.2) <sup>†</sup>	(78.0, 81.4) <sup>†</sup>	(62.8, 68.1)	(82.1, 85.4)
	Obj	MPDD	6	(54.3, 65.4)	(56.2, 66.0)	(63.6, 69.9)	(73.0, 80.2)	(55.1, 64.2)	(77.0, 82.0)
		BTAD	3	(34.5, 52.5)	(51.0, 62.1)	(68.2, 70.9)	(73.6, 68.6)	(66.8, 77.4)	(88.3, 87.3)
		SDD	1	(65.7, 45.2)	(65.2, 45.7)	(84.3, 77.4)	(79.8, 71.4)	(74.9, 65.1)	(84.7, 80.0)
	Texture	DAGM	10	(79.6, 59.0)	(82.5, 63.7)	(91.8, 79.5)	(94.4, 83.8)	(87.5, 74.6)	(97.5, 92.3)
		DTD-Synthetic	12	(71.6, 85.7)	(66.8, 83.2)	(93.2, 92.6)	(86.4, 95.0)	(-, -)	(93.5, 97.0)
Pixel-level (AUROC, PRO)	Obj & texture	MVTec AD	15	(38.4, 11.3)	(38.2, 11.6)	(85.1, 64.6) <sup>†</sup>	(87.6, 44.0) <sup>†</sup>	(33.3, 6.7)	(91.1, 81.4)
		VisA	12	(46.6, 14.8)	(47.8, 17.3)	(79.6, 56.8) <sup>†</sup>	(94.2, 86.8) <sup>†</sup>	(24.2, 3.8)	(95.5, 87.0)
	Obj	MPDD	6	(62.1, 33.0)	(58.7, 29.1)	(76.4, 48.9)	(94.1, 83.2)	(15.4, 2.3)	(96.5, 88.7)
		BTAD	3	(30.6, 4.4)	(32.8, 8.3)	(72.7, 27.3)	(60.8, 25.0)	(28.6, 3.8)	(94.2, 74.8)
		SDD	1	(39.0, 8.9)	(32.5, 5.8)	(68.8, 24.2)	(79.8, 65.1)	(28.9, 7.1)	(90.6, 67.8)
	Texture	DAGM	10	(28.2, 2.9)	(32.7, 4.8)	(87.6, 65.7)	(82.4, 66.2)	(17.5, 2.1)	(95.6, 91.0)
		DTD-Synthetic	12	(33.9, 12.5)	(23.7, 5.5)	(83.9, 57.8)	(95.3, 86.9)	(-, -)	(97.9, 92.3)

# Experiments

## ❖ Main result – Medical domain

- 모든 지표는 높을 수록 좋음
  - ✓ AUROC: 정상과 이상을 얼마나 잘 구분했는지?
  - ✓ AP (Average Precision): 이상이라고 예측한 것들 중 얼마나 맞았는지?
  - ✓ PRO (Per-Region Overlap): 픽셀 단위로 이상 영역의 위치를 얼마나 잘 맞췄는가?
- MVTEC AD 테스트 데이터셋으로 미세조정 후 다른 도메인 데이터셋에 대해 실험 진행
- MVTEC AD 평가 시에는 VisA 데이터셋의 테스트 데이터로 평가

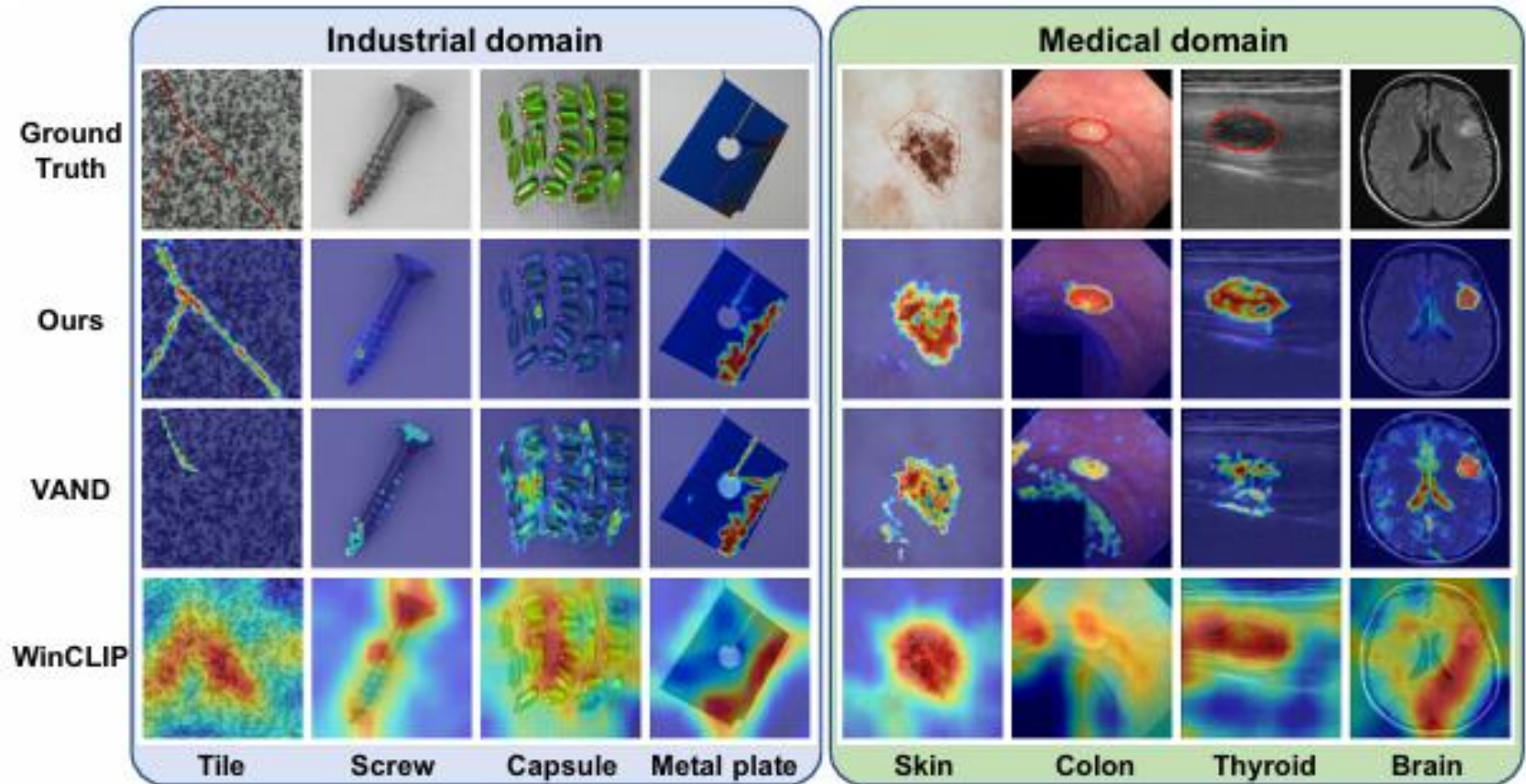
Task	Category	Datasets	C	CLIP	CLIP-AC	WinCLIP	VAND	CoOp	AnomalyCLIP
Image-level (AUROC, AP)	Brain	HeadCT	1	(56.5, 58.4)	(60.0, 60.7)	(81.8, 80.2)	(89.1, 89.4)	(78.4, 78.8)	(93.4, 91.6)
		BrainMRI	1	(73.9, 81.7)	(80.6, 86.4)	(86.6, 91.5)	(89.3, 90.9)	(61.3, 44.9)	(90.3, 92.2)
		Br35H	1	(78.4, 78.8)	(82.7, 81.3)	(80.5, 82.2)	(93.1, 92.9)	(86.0, 87.5)	(94.6, 94.7)
	Chest	COVID-19	1	(73.7, 42.4)	(75.0, 45.9)	(66.4, 42.9)	(15.5, 8.5)	(25.3, 9.2)	(80.1, 58.7)
Pixel-level (AUROC, PRO)	Skin	ISIC	1	(33.1, 5.8)	(36.0, 7.7)	(83.3, 55.1)	(89.4, 77.2)	(51.7, 15.9)	(89.7, 78.4)
		CVC-ColonDB	1	(49.5, 15.8)	(49.5, 11.5)	(70.3, 32.5)	(78.4, 64.6)	(40.5, 2.6)	(81.9, 71.3)
	Colon	CVC-ClinicDB	1	(47.5, 18.9)	(48.5, 12.6)	(51.2, 13.8)	(80.5, 60.7)	(34.8, 2.4)	(82.9, 67.8)
		Kvasir	1	(44.6, 17.7)	(45.0, 16.8)	(69.7, 24.5)	(75.0, 36.2)	(44.1, 3.5)	(78.9, 45.6)
		Endo	1	(45.2, 15.9)	(46.6, 12.6)	(68.2, 28.3)	(81.9, 54.9)	(40.6, 3.9)	(84.1, 63.6)
	Thyroid	TN3K	1	(42.3, 7.3)	(35.6, 5.2)	(70.7, 39.8)	(73.6, 37.8)	(34.0, 9.5)	(81.5, 50.4)



# Experiments

## ❖ Main result

- 이상 영역을 세부적으로 잡을 수록 높은 성능
- Ground truth 사진에서 빨간 표시가 이상 영역임





# Conclusion

## ❖ Industrial Image Anomaly Detection

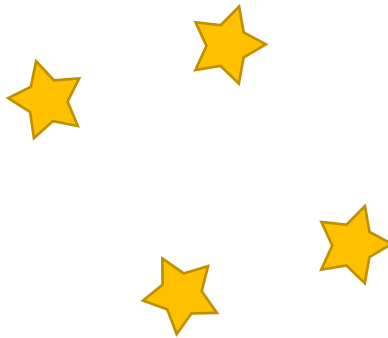
- 대량의 정상 데이터를 활용하여 모델을 학습하는 전통적인 방식의 문제점을 해결하고자 함

### Unsupervised Anomaly Detection



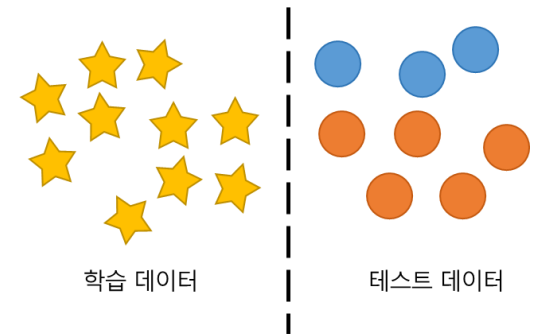
- 학습 시, 데이터가 대부분 정상으로 이루어졌다고 **가정**
- 현실과 가장 유사한 상황
- 비정상이 포함된 학습 가능성 ↑

### Few-shot Anomaly Detection



- 정상 데이터만을 활용
- **적은 수**의 정상 이미지를 활용
- 현실성이 있으면서 확실한 방법

### Zero-shot Anomaly Detection



- 학습 데이터와 테스트 데이터가 **다른 도메인**일 때 적용 가능
- 학습 데이터를 **확보할 수 없는 경우**

# Conclusion

## ❖ PatchCore: Towards Total Recall in Industrial Anomaly Detection

- 사전 학습된 모델로 뽑은 feature을 효율적으로 사용
  - ✓ 사전 학습된 모델로부터 뽑은 feature를 locally aware하게 만듦
  - ✓ Coreset subsampling을 통해 효율적인 계산이 가능
- 매우 높은 성능을 보여줌

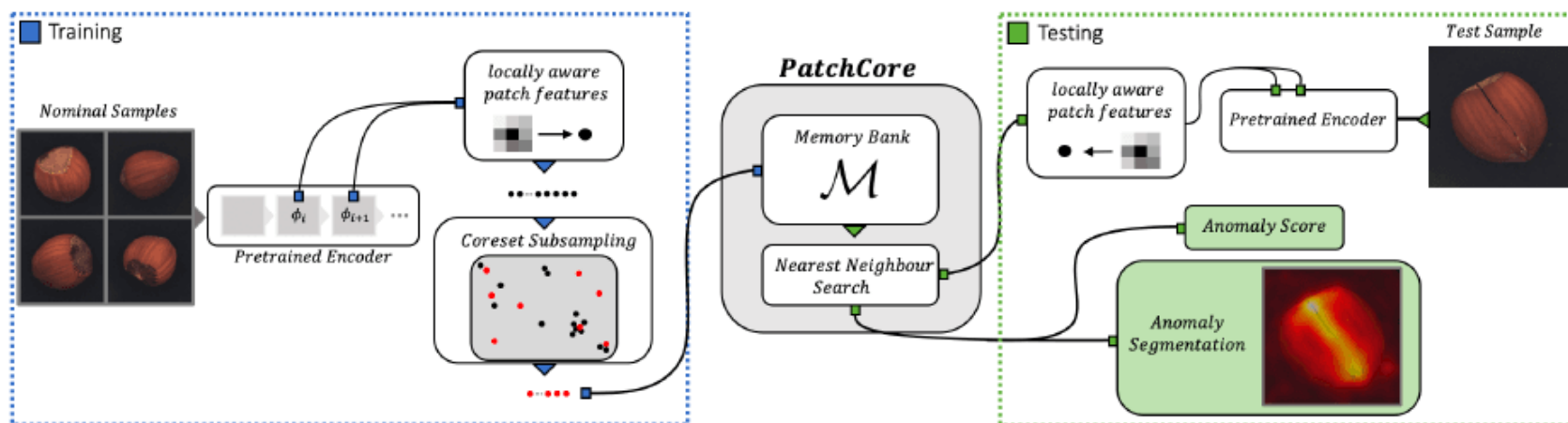
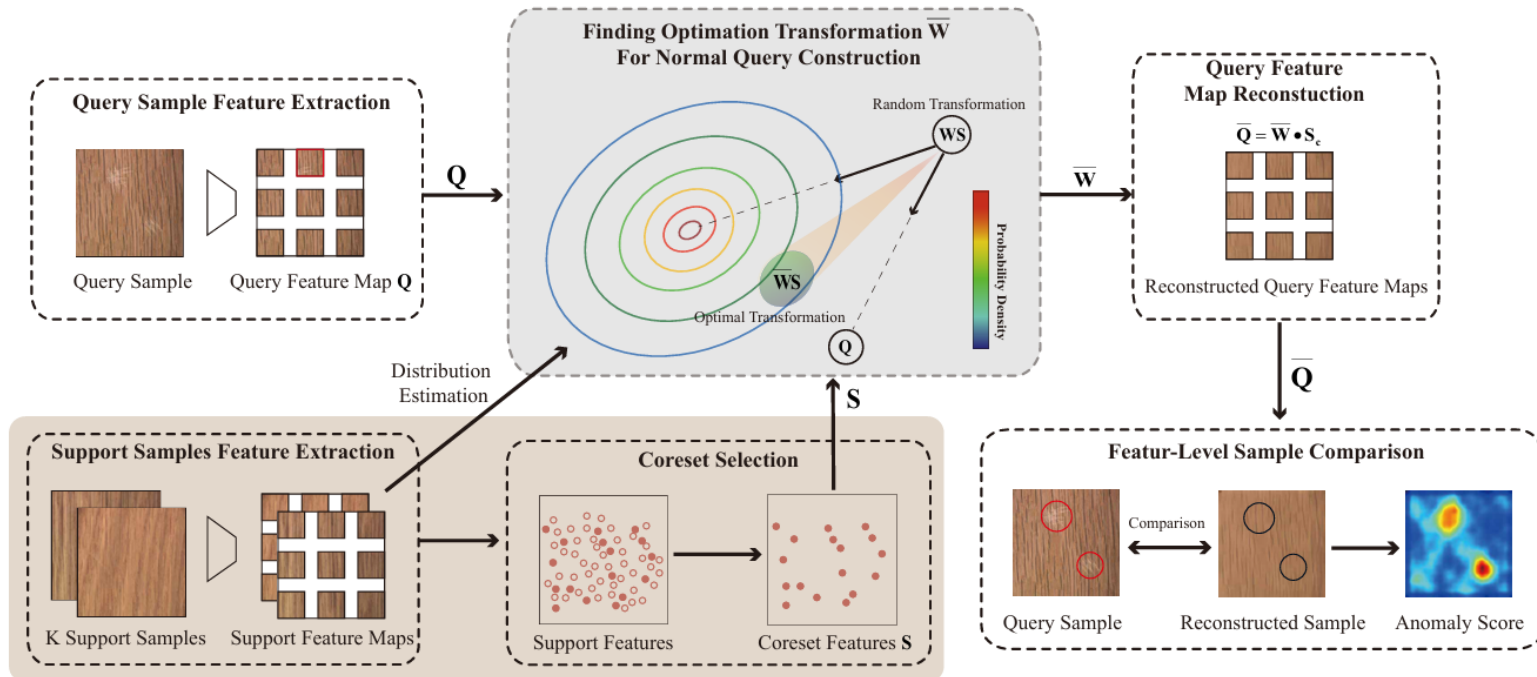


Figure 2: Overview of *PatchCore*. During training, nominal samples are broken down into a memory bank of neighbourhood-aware patch-level features. For reduced redundancy and inference time, this memory bank is downsampled via greedy coreset subsampling. At test time, images are classified as anomalies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

# Conclusion

## ❖ FastRecon: Few-shot Industrial Anomaly Detection via Fast Feature Reconstruction

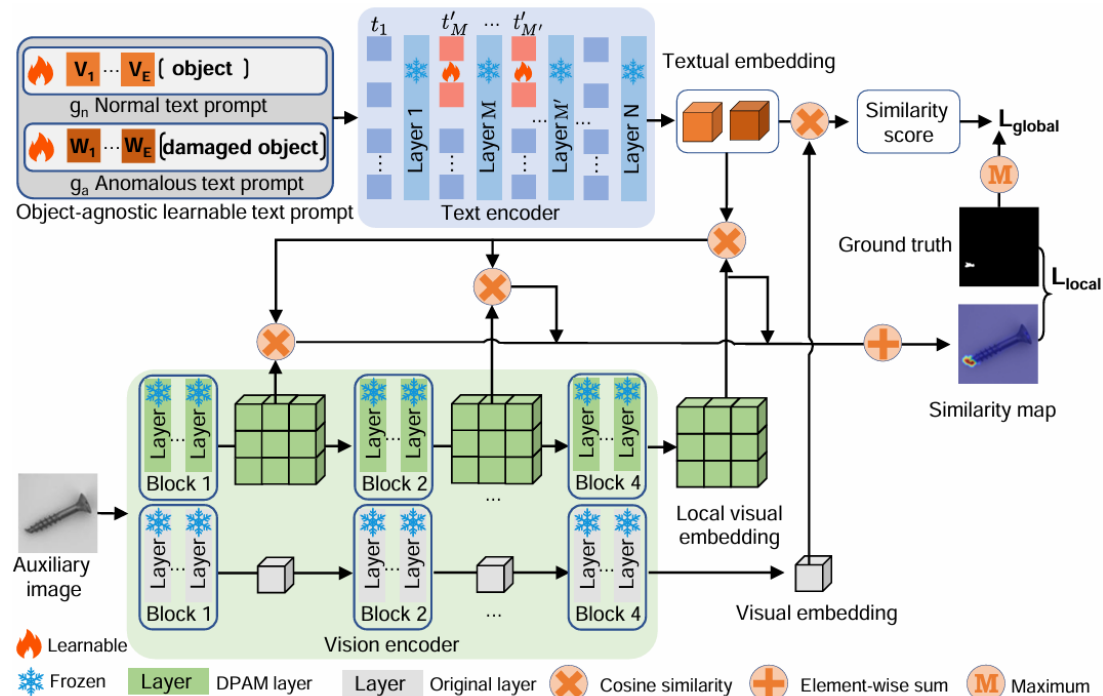
- 소수의 정상 샘플을 활용하여 효과적인 이상탐지를 수행할 수 있는 방법론 제안
  - ✓ 데이터 효율성: 적은 데이터로도 높은 탐지 성능
  - ✓ 훈련 불필요: 사전 학습된 네트워크 활용
  - ✓ 분포 정규화를 통해 재구성 개선: 재구성된 피처가 정상 특징을 유지하도록 설계



# Conclusion

## ❖ AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection

- CLIP의 ZSAD에 대한 낮은 일반화 능력을 개선
  - ✓ Object-agnostic Prompt를 통해 정상/이상 분류 성능을 높임
  - ✓ Object-agnostic Prompt를 학습하기 위한 Glocal Loss 활용
  - ✓ Refinement를 통해 Anomaly Detection task에 더욱 맞게끔 사용
- ZSAD 성능이 높아짐을 보여줌



---

**고맙습니다**

---