Data-aware Anomaly Detection

2025 summer seminar

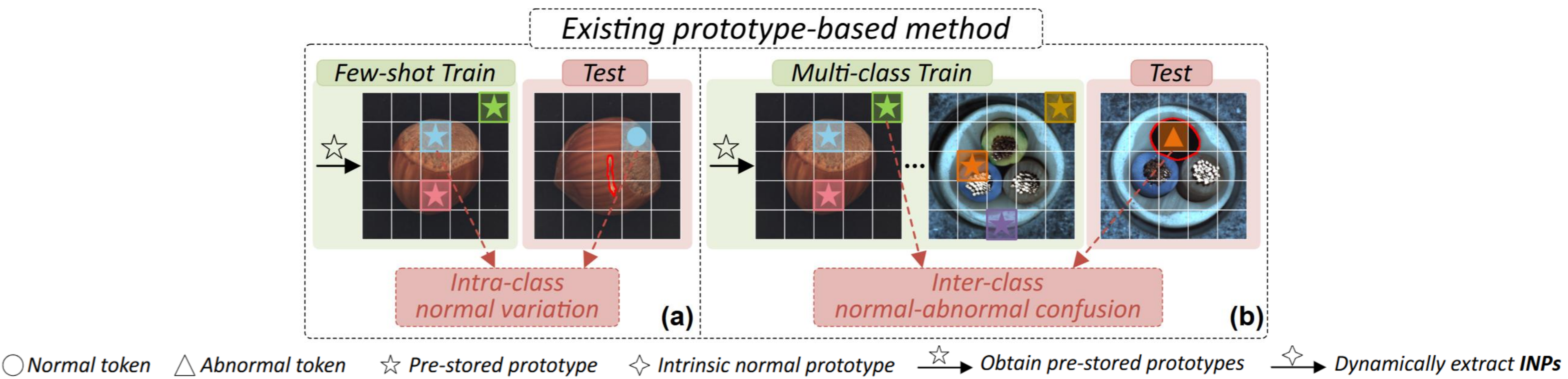
Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
Jimin Roh

Exploring Intrinsic Normal Prototypes within a Single Image for Universal Anomaly Detection
[CVPR'25]

Introduction

- Problem formulation
  - Intra-class normal variation
    - 동일 class 내에서도 정상(normal) 상태가 다양하게 나타남
    - Few-shot AD에서는 train set의 normal과 test set의 normal이 달라 성능 저하를 유발
  - Inter-class normal variation
    - 서로 다른 class 간에도 정상 상태가 다르게 나타남
    - 서로 다른 class의 patch를 비교할 때 normal과 anomaly를 혼동할 수 있음

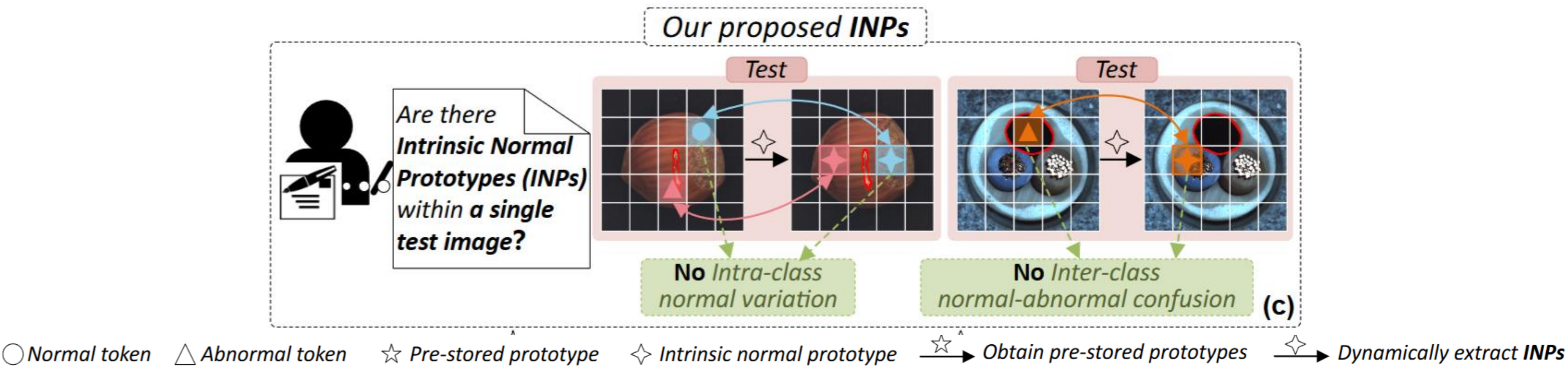


(a): 동일한 *hazelnut* 클래스에서도 **train과 test 간 normal patch의 특성이 달라질 수 있음**
(b): *hazelnut* 클래스의 **normal patch**가 cable 클래스의 **anomaly patch와 혼동**될 수 있음

1) Luo, Wei, et al. "Exploring intrinsic normal prototypes within a single image for universal anomaly detection." *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025.

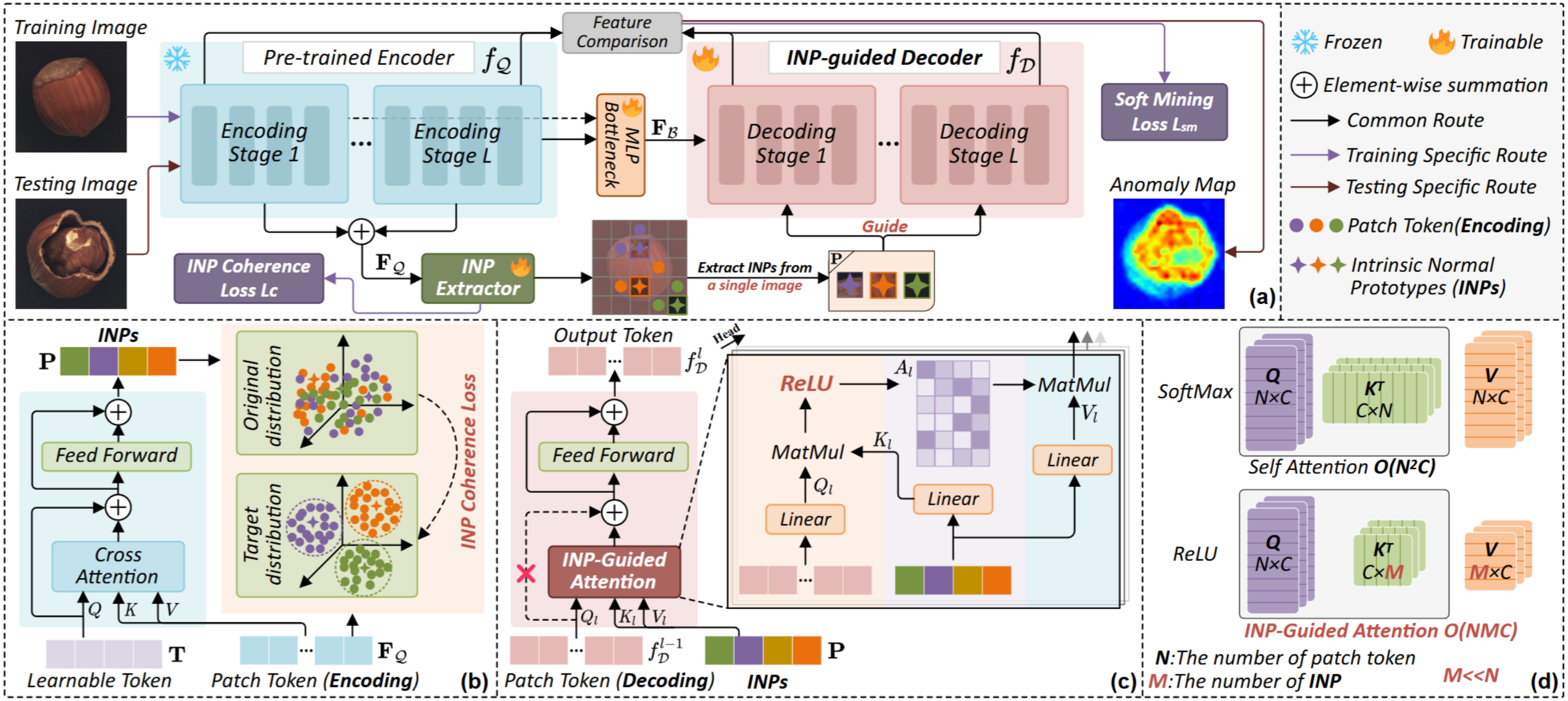
Introduction

- Problem formulation
  - Intrinsic Normal Prototype (INP)
    - 해결방안
      - Test 이미지 내에서 내재된 정상 패턴을 동적으로 추출하여 사용
    - 기대 효과
      - No intra-class normal variation
        - ✓클래스 내부의 정상 패턴 다양성을 극복
      - No inter-class normal-abnormal confusion
        - ✓서로 다른 클래스 간 정상-이상 혼동 방지



Method

- Overall architecture
  - 전체적으로 encoder, bottleneck, decoder 구조로 전형적인 reconstruction 기반 AD 모델 구조
  - 추가적으로 INP extractor와 INP-aware cross attention 모듈을 추가하여 성능을 개선



< INP-former 전체 아키텍처 >

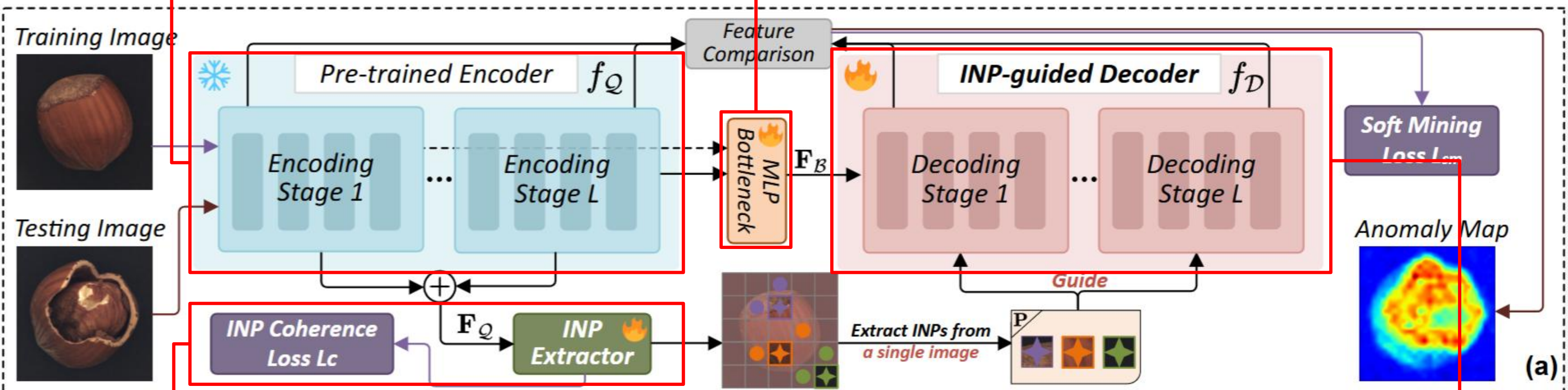
서강대학교 SOGANG UNIVERSITY

Method

- Overall architecture
  - 전체 동작 과정 요약
    - **Step 1.** 입력 이미지에 대해 pre-trained encoder(DINOv2)를 사용해 level별 patch tokens를 추출
    - **Step 2.1.** 추출된 patch tokens와 INPs를 INP extractor에 입력하여 정상 패턴을 학습
    - **Step 2.2.** Patch tokens와 INPs를 결합해 bottleneck에 전달
    - **Step 3.** Bottleneck 출력과 INPs를 decoder에 입력해 patch tokens를 재구성

Step 1. 입력 이미지에 대한 feature 추출

Step 2.2. level 별 feature 통합



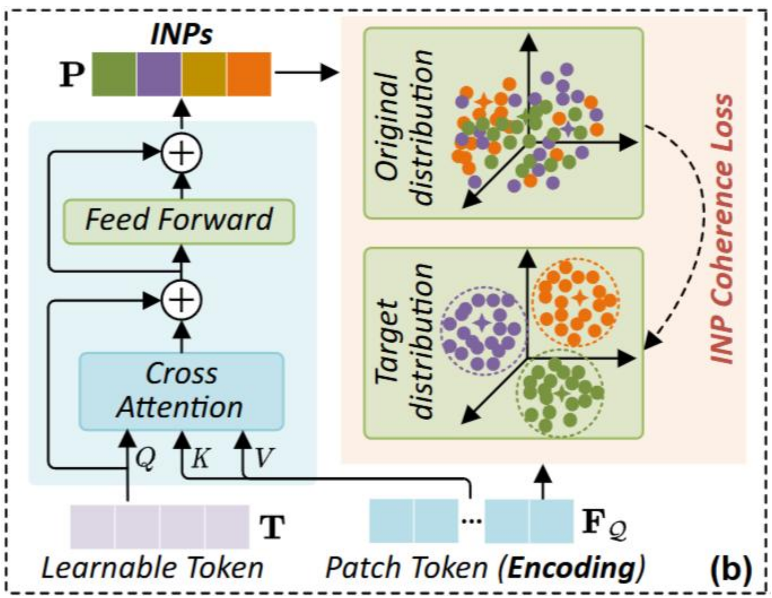
< INP-former encoder-decoder 구조 >

Step 2.1. 대표 정상 패턴 prototype 추출

Step 3. 정상 패턴을 고려한 feature reconstruction 수행

Method

- INP Extractor
  - 동작 과정
    - **Step 1.** Test 이미지의 patch token과 학습 가능한 learnable token을 초기화
    - **Step 2.** 두 token들을 cross-attention ViT의 입력으로 사용하여 INP token을 생성
    - **Step 3.** INP token과 patch token과의 coherence loss를 통해 INP token이 정상 패턴을 학습하도록 유도



< INP extractor 동작 과정 >

결론적으로 INPs와 patch token이 유사해지도록 학습

$$\underbrace{Patch\ token\ (H'W', D) \approx INPs\ (6, D)}_{\text{대표성만 학습}}$$

$$d_i = \min_{m \in \{1,\ldots,M\}} \mathcal{S}(\mathbf{F}_Q(i), p_m)$$

$$\mathcal{L}_c = \frac{1}{N} \sum_{i=1}^{N} d_i$$

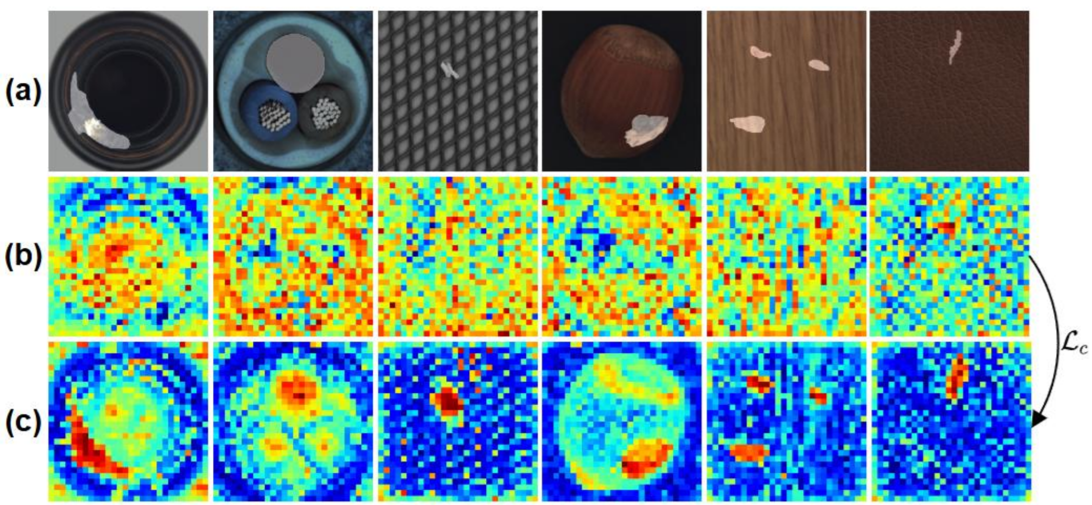
where $S(\cdot,\cdot)$ denotes the cosine distance

< INP coherence loss 수식 >

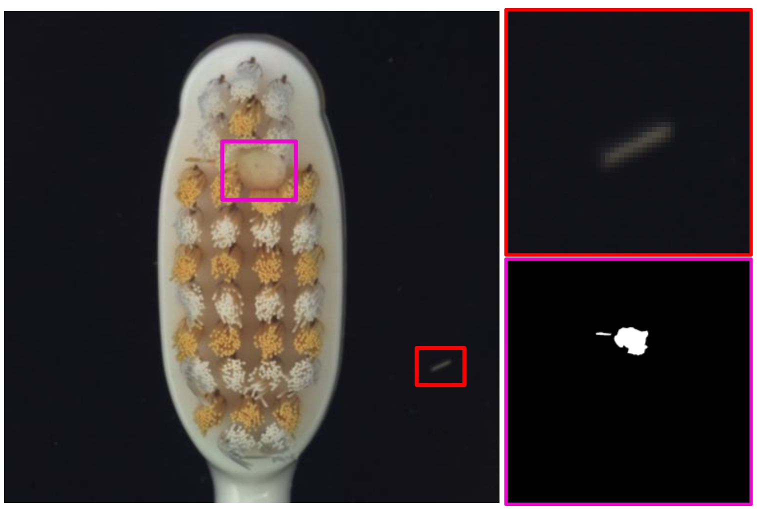
1) Luo, Wei, et al. "Exploring intrinsic normal prototypes within a single image for universal anomaly detection." *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025.

Method

- INP Extractor
  - 한계점
    - INP coherence loss는 입력 이미지와 INPs 간의 거리를 최소화하도록 학습되며, 이를 통해 test 이미지의 abnormal region을 효과적으로 검출 가능
    - 그러나 **background noise나 weak normal region**에 대해서는 구분 성능이 떨어지는 한계가 존재
    - 특히, 배경 잡음이 강하거나 **정상 패턴의 경계가 불명확한 경우**, 이상 영역과 혼동할 수 있음



(a): Input image, GT
(b): Attention map (w/o Coherence loss)
(c): Attention map (w/ Coherence loss)

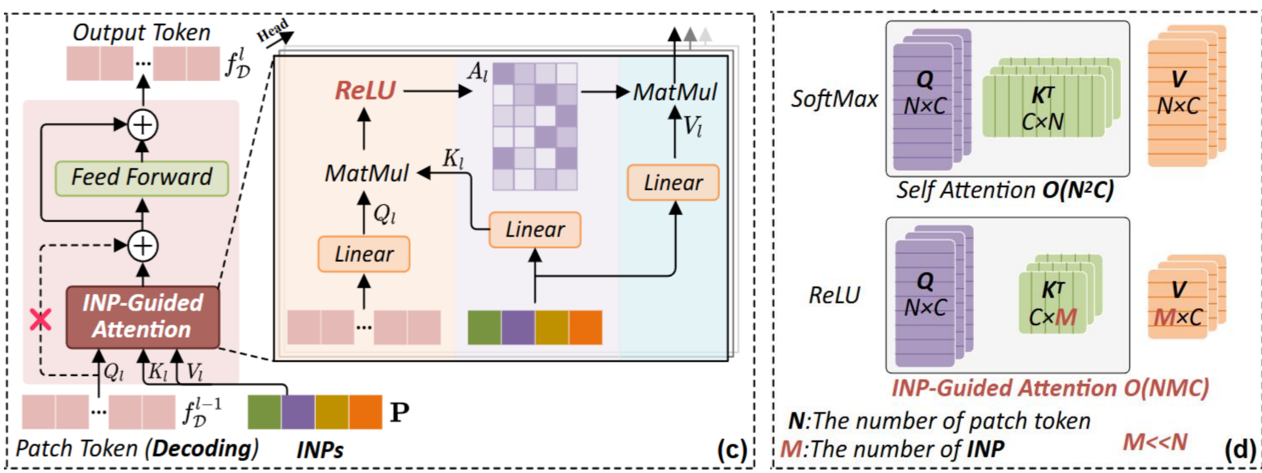


< Background noise 예시 >

1) Luo, Wei, et al. "Exploring intrinsic normal prototypes within a single image for universal anomaly detection." *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025.

Method

- INP-Guided Decoder
  - 추출된 INPs를 활용하여 feature를 재구성
    - Test 이미지에서 추출한 INPs를 기준으로 patch tokens를 재구성하여 anomaly map을 생성
  - 효율적인 attention 구조 적용
    - Encoder와 동일한 구조를 기반으로 기존 self-attention을 INP-guided attention으로 대체
    - INP-guided attention은 계산량과 메모리 사용량을 크게 줄이면서도 성능을 유지



< INP-aware decoder 동작 과정 >

Table 1. Comparison of **computational cost** and **memory usage**.

	Number of multiplicaiton and addition	
Calculation	Vanilla Self Attention	**INP-Guided Attention**
$A_l = Q_l(K_l)^T$	943 496 960	**7 220 640**
$f_{\mathcal{D}}^{l-1'} = A_l V_l$	943 509 504	**6 623 232**
	Memory usage (MB)	
$Q_l/K_l/V_l/A_l$	2.30/2.30/2.30/2.34	2.30/**0.018/0.018/0.018**

< Self-attention/ INP-guided attention 비교 표 >

Experiment

- Implementation details
  - 환경 세팅
    - GPU: RTX 4090 1개
    - Resize: 448 x 448
    - Learning rate: 1e-4
    - Epochs: 200
    - Scheduler: StableAdamW
- Universal anomaly detection
  - 실험
    - Single-class anomaly detection
    - Multi-class anomaly detection
    - Few/zero-shot anomaly detection
    - Super-multi-class anomaly detection

Experiment

- Single-class AD

Table 4. **Single class** anomaly detection performance on different AD datasets. The best in **bold**, the second-highest is underlined.

Dataset →	MVTec-AD [2]			VisA [51]			Real-IAD [41]		
Method ↓	I-AUROC	P-AP	AUPRO	I-AUROC	P-AP	AUPRO	I-AUROC	P-AP	AUPRO
PatchCore [38]	99.1	56.1	93.5	95.1	40.1	91.2	89.4	-	91.5
RD4AD [10]	98.5	58.0	93.9	96.0	27.7	70.9	87.1	-	93.8
SimpleNet [30]	99.6	54.8	90.0	96.8	36.3	88.7	88.5	-	84.6
Dinomaly [17]	**99.7**	68.9	95.0	**98.9**	**50.7**	**95.1**	92.0	45.2	95.1
INP-Former	**99.7**	**70.2**	**95.4**	98.5	49.2	93.8	**92.1**	**48.1**	**95.6**

- Multi-class AD

Table 2. **Multi-class** anomaly detection performance on different AD datasets. The best in **bold**, the second-highest is underlined.

Dataset →	MVTec-AD [2]		VisA [51]		Real-IAD [41]	
Metric →	Image-level(I-AUROC/I-AP/I-F1_max)		Pixel-level(P-AUROC/P-AP/P-F1_max/AUPRO)			
Method ↓	Image-level	Pixel-level	Image-level	Pixel-level	Image-level	Pixel-level
RD4AD [10]	94.6/96.5/95.2	96.1/48.6/53.8/91.1	92.4/92.4/89.6	98.1/38.0/42.6/91.8	82.4/79.0/73.9	97.3/25.0/32.7/89.6
UniAD [46]	96.5/98.8/96.2	96.8/43.4/49.5/90.7	88.8/90.8/85.8	98.3/33.7/39.0/85.5	83.0/80.9/74.3	97.3/21.1/29.2/86.7
SimpleNet [30]	95.3/98.4/95.8	96.9/45.9/49.7/86.5	87.2/87.0/81.8	96.8/34.7/37.8/81.4	57.2/53.4/61.5	75.7/2.8/6.5/39.0
DeSTSeg [47]	89.2/95.5/91.6	93.1/54.3/50.9/64.8	88.9/89.0/85.2	96.1/39.6/43.4/67.4	82.3/79.2/73.2	94.6/37.9/41.7/40.6
DiAD [19]	97.2/99.0/96.5	96.8/52.6/55.5/90.7	86.8/88.3/85.1	96.0/26.1/33.0/75.2	75.6/66.4/69.9	88.0/2.9/7.1/58.1
MambaAD [18]	98.6/99.6/97.8	97.7/56.3/59.2/93.1	94.3/94.5/89.4	98.5/39.4/44.0/91.0	86.3/84.6/77.0	98.5/33.0/38.7/90.5
Dinomaly [17]	99.6/99.8/99.0	98.4/69.3/69.2/94.8	98.7/98.9/96.2	98.7/**53.2/55.7/94.5**	89.3/86.8/80.2	98.8/42.8/47.1/93.9
INP-Former	**99.7/99.9/99.2**	**98.5/71.0/69.7/94.9**	**98.9/99.0/96.6**	**98.9**/51.2/54.7/94.4	**90.5/88.1/81.5**	**99.0/47.5/50.3/95.0**

1) Luo, Wei, et al. "Exploring intrinsic normal prototypes within a single image for universal anomaly detection." *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025.

Experiment

- Few/Zero-shot AD
  - 단일 이미지에서 정상 패턴을 추출하는 능력
    - Test 이미지에서 INPs를 추출한 뒤 입력 feature와 비교하여 defect를 검출함
    - 이러한 방식 덕분에 few-shot 및 zero-shot 설정 모두에서 활용 가능

Table 3. **Few-shot (4-shot)** anomaly detection performance on different AD datasets. The best in **bold**, the second-highest is underlined. † indicates the results we reproduced using publicly available code.

Dataset →	MVTec-AD [2]		VisA [51]		Real-IAD [41]	
Method ↓	Image-level	Pixel-level	Image-level	Pixel-level	Image-level	Pixel-level
SPADE [7]	84.8/92.5/91.5	92.7/-/46.2/87.0	81.7/83.4/82.1	96.6/-/43.6/87.3	50.8†/45.8†/61.2†	59.5†/0.2†/0.5†/19.2†
PaDiM [9]	80.4/90.5/90.2	92.6/-/46.1/81.3	72.8/75.6/78.0	93.2/-/24.6/72.6	60.3†/53.5†/64.0†	90.9†/2.1†/5.1†/67.6†
PatchCore [38]	88.8/94.5/92.6	94.3/-/55.0/84.3	85.3/87.5/84.3	96.8/-/43.9/84.9	66.0†/62.2†/65.2†	92.9†/9.8†/16.1†/68.6†
WinCLIP [24]	95.2/97.3/94.7	96.2/-/59.5/89.0	87.3/88.8/84.2	97.2/-/47.0/87.6	73.0†/61.8†/61.0†	93.8†/13.3†/21.0†/76.4†
PromptAD [28]	96.6/-/-	96.5/-/-/90.5	89.1/-/-	97.4/-/-/86.2	59.7†/43.5†/52.9†	86.9†/8.7†/16.1†/61.9†
INP-Former	**97.6/98.6/97.0**	**97.0/65.9/65.6/92.9**	**96.4/96.0/93.0**	**97.7/49.3/54.3/93.1**	**76.7/72.3/71.7**	**97.3/32.2/36.7/89.0**

Table S8. **Zero-shot** anomaly detection performance on different AD datasets. The best in **bold**.

Dataset →	MVTec-AD [2]		VisA [51]	
Metric →	Image-level(I-AUROC/I-AP/I-F1_max) Pixel-level(P-AUROC/P-AP/P-F1_max/AUPRO)			
Method ↓	Image-level	Pixel-level	Image-level	Pixel-level
WinCLIP [24]	**91.8/96.5/92.9**	85.1/-/31.7/64.6	**78.1/81.2/79.0**	79.6/-/**14.8**/56.8
INP-Former	80.8/90.7/89.1	**88.0/36.1/39.5/76.9**	67.5/71.6/75.0	**88.7/7.8**/11.8/**67.2**

Experiment

- Super-multi-class AD
  - 3가지 데이터셋을 통합해 학습 및 평가를 진행
    - 각 데이터셋별 성능과 비교했을 때, 소폭의 성능 감소만 나타남
  - INP extractor의 강건성 확인
    - INP extractor가 정상 패턴을 효과적으로 추출하기 때문에, 다양한 데이터셋 통합 상황에서도 높은 성능을 유지

Table S5. **Super-multi-class** anomaly detection performance on different AD datasets. Δ represents the performance change of INP-Former in the super-multi-class setting relative to the multi-class setting.

Dataset →	MVTec-AD [2]		VisA [51]		Real-IAD [41]	
Metric →	Image-level(I-AUROC/I-AP/I-F1_max)			Pixel-level(P-AUROC/P-AP/P-F1_max/AUPRO)		
Setting ↓	Image-level	Pixel-level	Image-level	Pixel-level	Image-level	Pixel-level
Multi-Class	99.7/99.9/99.2	98.5/71.0/69.7/94.9	98.9/99.0/96.6	98.9/51.2/54.7/94.4	90.5/88.1/81.5	99.0/47.5/50.3/95.0
Super-Multi-Class	99.5/99.8/98.9	98.1/69.2/68.1/94.2	97.3/97.8/94.1	98.4/51.4/54.7/92.4	89.8/87.4/80.5	98.9/45.2/48.6/94.4
Δ	0.2↓/0.1↓/0.3↓	0.4↓/1.8↓/1.6↓/0.7↓	1.6↓/1.2↓/2.5↓	0.5↓/0.2↑/0.0/2.0↓	0.7↓/0.7↓/1.0↓	0.1↓/2.3↓/1.7↓/0.6↓

Experiment

- Qualitative results
  - 제안한 방법의 이상 탐지 성능을 정성적으로 확인하기 위해, 세 가지 데이터셋(MVTec-AD, VisA, Real-IAD)에서의 anomaly localization 결과를 시각화
  - 각 데이터셋의 다양한 클래스에서 입력 이미지와 이상 영역의 anomaly map을 비교
  - 모든 데이터셋에서 정상 영역과 이상 영역이 명확하게 구분되며, 작은 이상 패턴도 잘 탐지함을 확인

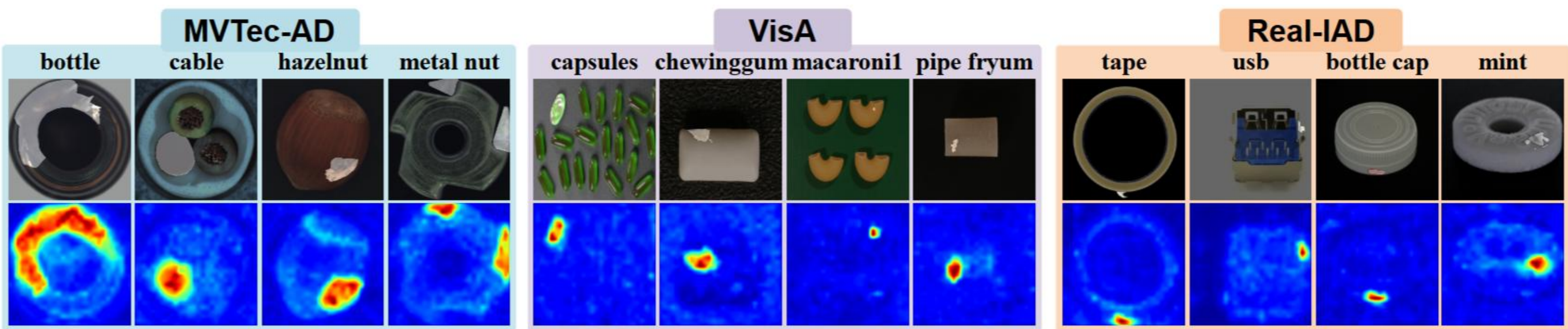


Figure 3. **Qualitative results of anomaly localization** on the MVTec-AD [2], VisA [51], and Real-IAD [41] datasets for **multi-class anomaly detection**. The first row presents the input images with their ground truth, while the second row displays the corresponding anomaly maps.

Experiment

- Ablation
  - Intrinsic normal prototype 분석
    - INPs attention map 시각화
      - 6개의 attention map 모두 **defect와는 관계없이 서로 다른 정상패턴을 학습**
    - INPs 개수에 따른 성능 변화 분석
      - INP 개수가 4 이상일 때 학습이 안정화되며 성능이 향상됨
      - 그러나 개수가 너무 많아지면 **abnormal token까지 학습해 오히려 성능이 감소함**

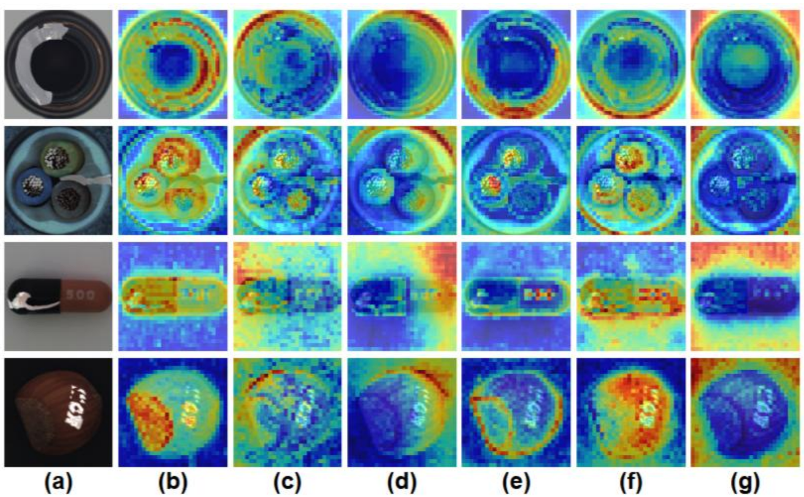


Figure 7. **Visualizations of INPs.** (a) Input anomalous image and ground truth. (b)-(g) Attention maps of six different INPs.

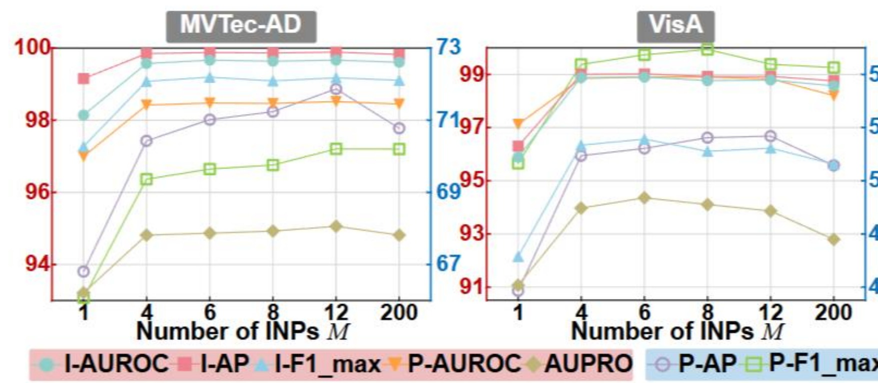


Figure 6. **Influence of the number of INPs** M on model performance across the MVTec-AD [2] and VisA [51] datasets. Pixel-level AP and F1_max use the right vertical axis, while the other metrics share the left vertical axis.

Limitation

- Logical anomaly detection
  - Defect type: misplaced
    - Cable처럼 misplaced 영역이 background distribution과 다르면 검출 가능
    - 하지만 Transistor처럼 misplaced 영역이 background와 유사하면 검출 어려움
      - INP extractor가 anomaly 영역을 INPs로 추출하고, INP-guided decoder에서 이를 복원하기 때문에 background와 유사한 패턴은 구분하기 어려움

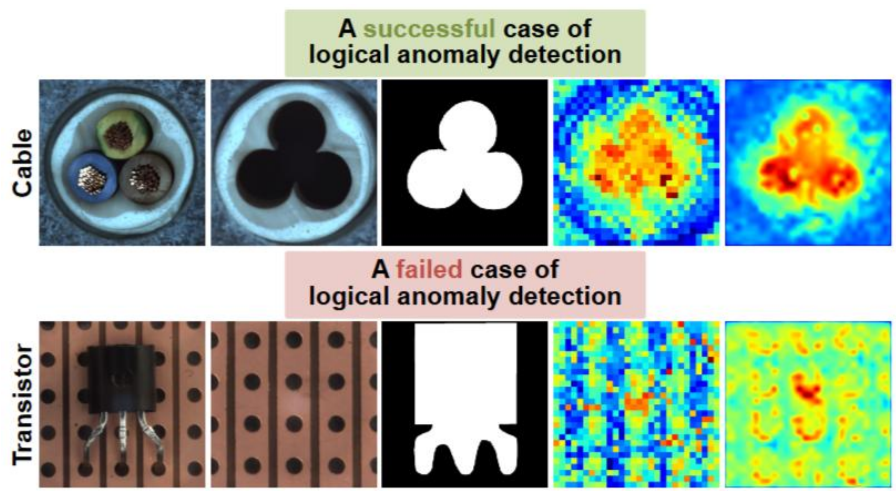
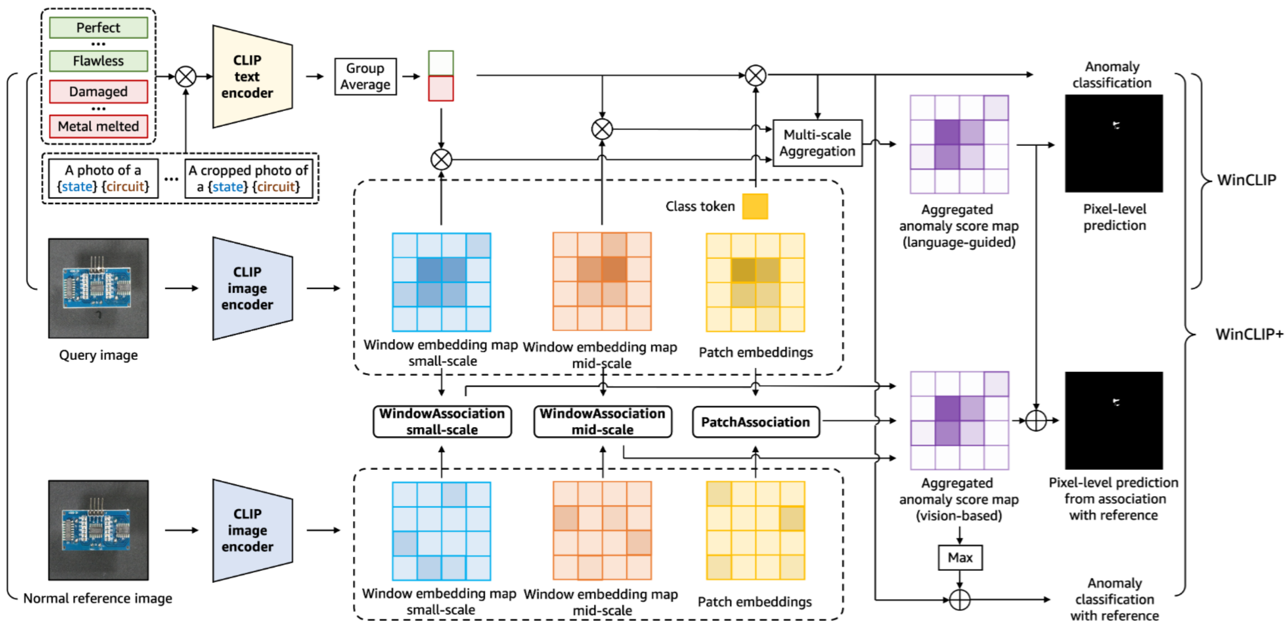


Figure S2. **Limitation of proposed method in detecting logical anomalies similar to the background.** From left to right: Normal Image, Input Anomaly, Ground Truth, Distance Map, and Predicted Anomaly Map.

MultiADS: Defect-aware Supervision for Multi-type Anomaly Detection and Segmentation in Zero-Shot Learning [ICCV'25]

Background

- CLIP-based Anomaly Detection
  - Training phase
    - Text embedding과 image embedding 간의 similarity map을 계산하고, 이를 GT mask와 비교해 loss를 통해 **text-image간의 alignment를 학습**하도록 함
  - Inference phase
    - 학습된 alignment 능력을 기반으로 **target image와 text prompt 간의 similarity를 계산**해 normal/abnormal을 판별



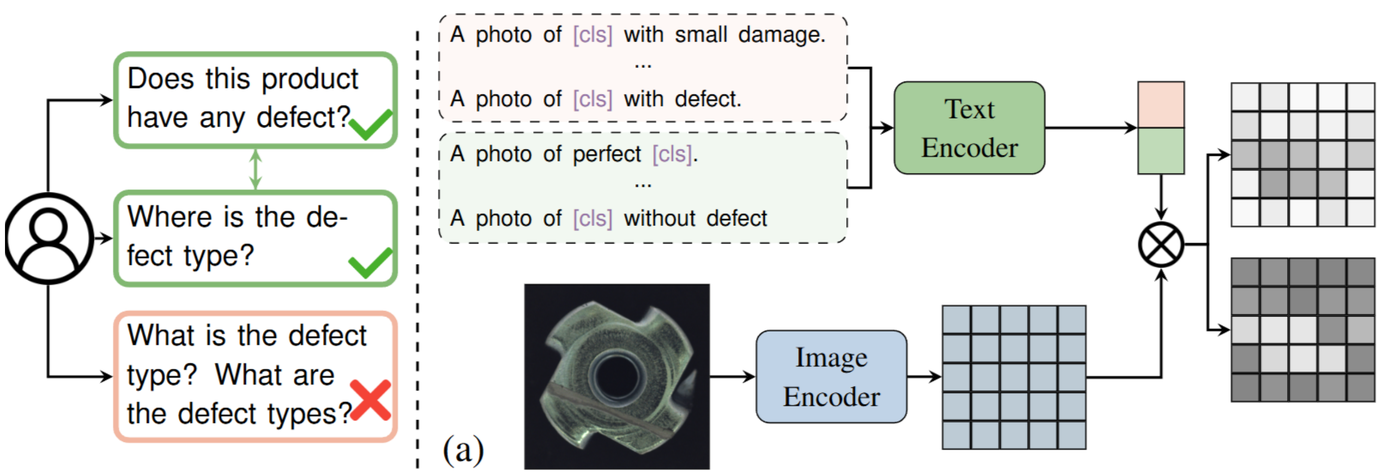
< WinCLIP 전체 아키텍처 >

서강대학교 SOGANG UNIVERSITY

VDS LAB

Introduction

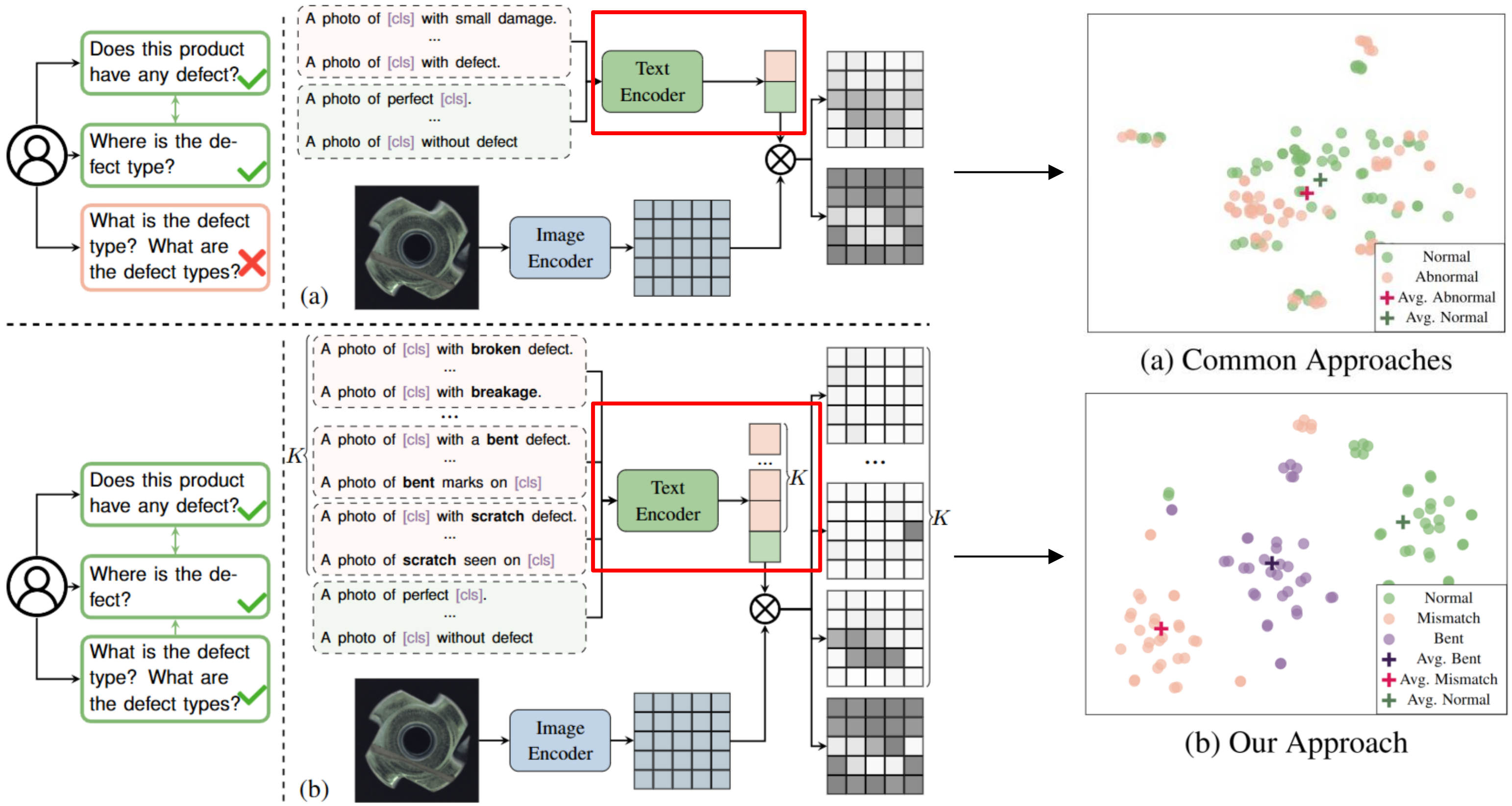
- Problem formulation
  - Defect type estimation의 중요성
    - Defect의 유형(type)을 알면 보다 **정확하고 효과적인 대응**이 가능
    - 또한 생산 라인에서 발생하는 결함에 대해 **근본적인 대처**가 가능
  - 이전 연구들의 문제점
    - Defect의 유형을 구별하는 것보다 defect를 검출하는데 초점을 맞춤
    - Pre-trained vision-language model에 **내재된 결함 유형 지식을 활용하지 못함**
    - 특정 도메인에서의 fine-tuning은 **overfitting을 유발해 중요한 지식을 상실함**



< 이전 연구들의 접근 방식 >

Introduction

- Problem formulation
  - 이전 연구들의 모델 구조 문제점
    - 기존 방법들은 각 text prompt별로 최종적으로 **average pooling을 적용해** image embedding과의 similarity를 계산. 하지만 이러한 average pooling 과정에서 중요한 정보가 손실될 수 있음

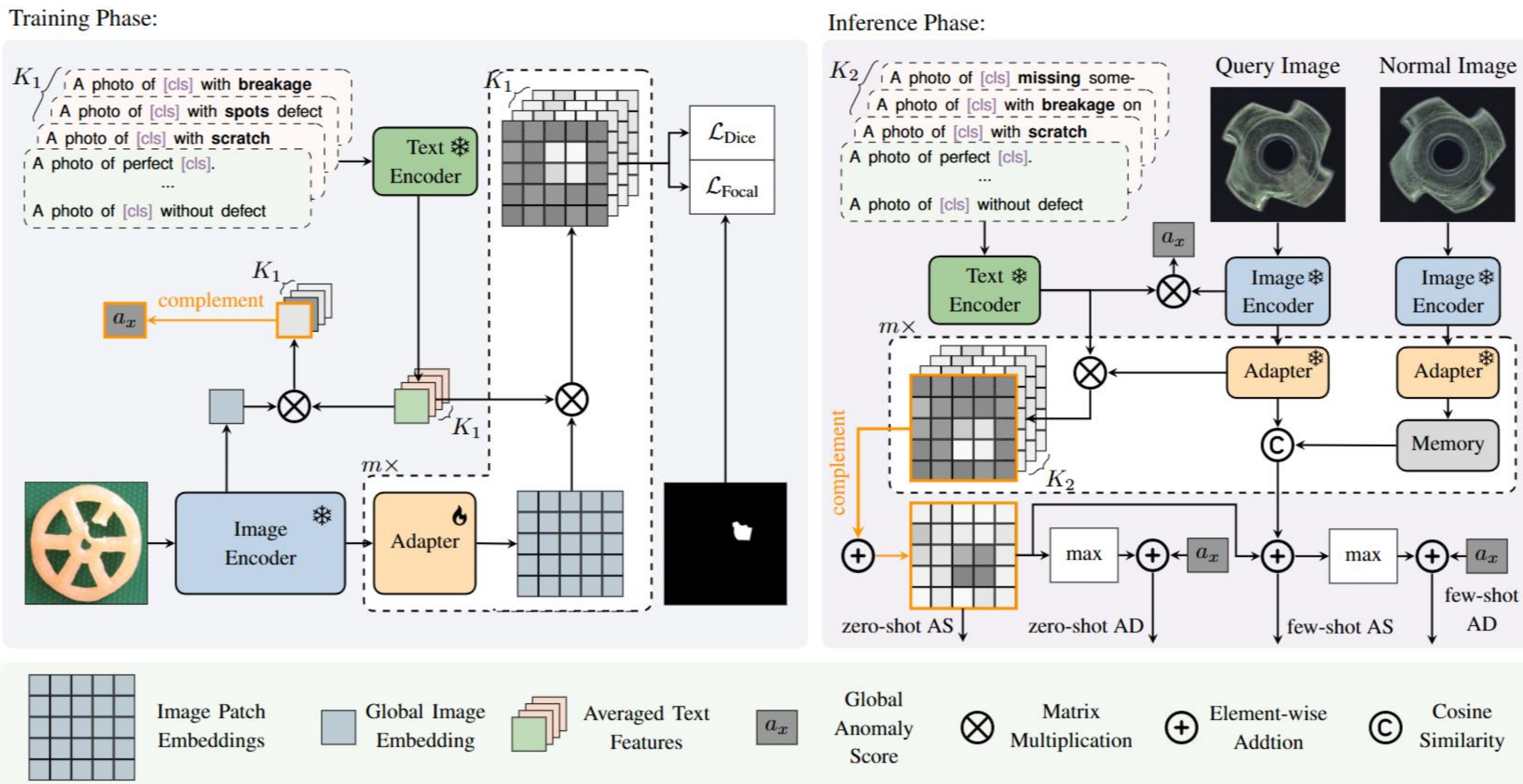


< 모델 구조 비교(좌), 각 모델들의 text embedding 시각화(우) >

1) Sadikaj, Ylli, et al. "MultiADS: Defect-aware Supervision for Multi-type Anomaly Detection and Segmentation in Zero-Shot Learning." *arXiv preprint arXiv:2504.06740* (2025).

Method

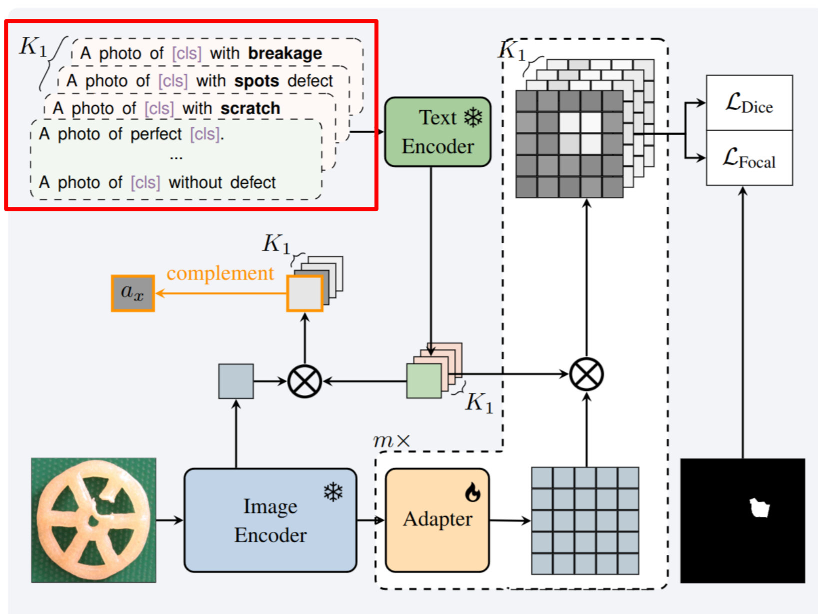
- Overall architecture
  - 전체적으로 **CLIP 기반 Anomaly Detection 모델 구조**를 따름
  - Few-shot 및 Zero-shot AD를 모두 수행할 수 있도록 inference 단계 설계



< **MultiADS 전체 파이프라인** >

Method

- 학습 과정
  - **Step 1.** Text prompt 구성
    - 데이터셋에서 발생 가능한 **모든 결함 유형**을 사전에 정의
    - 각 결함 유형에 대해 **여러 형태의 text prompt**를 생성
    - Training set과 test set의 prompt 종류는 서로 다르게 설정

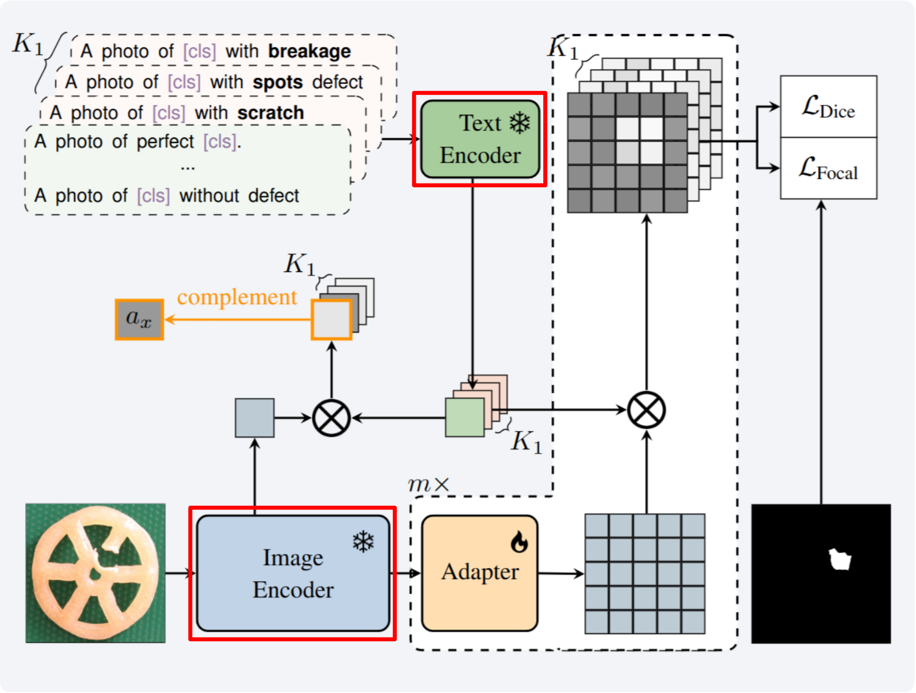


Defect Type	Defect-Aware Text Prompts	Defect Type	Defect-Aware Text Prompts
Bent	"*[cls]* has a bent defect" "flawed *[cls]* with a bent lead" "a bend found in *[cls]*" "*[cls]* has a slight curve defect" "*[cls]* with noticeable bending" "a bent wire on *[cls]*"	Broken	"*[cls]* with a breakage defect" "broken *[cls]*" "*[cls]* with broken defect" "*[cls]* shows breakage" "broken or cracked areas on *[cls]*" "visible breakage on *[cls]*"
Bubble	"*[cls]* with bubbles defect" "bubbles seen on *[cls]*" "*[cls]* with bubble marks" "air bubbles in *[cls]*" "*[cls]* contains bubble defects" "small bubbles on *[cls]* surface"	Burnt	"*[cls]* with a burnt defect" "*[cls]* shows burn marks" "burnt areas on *[cls]*" "*[cls]* with signs of burning" "scorch marks on *[cls]*" "*[cls]* appears slightly burnt"
Chip	"*[cls]* with chip defect" "*[cls]* with fragment broken defect" "chipped areas on *[cls]*" "*[cls]* with chipped parts" "broken fragments on *[cls]*" "chip marks found on *[cls]*"	Crack	"*[cls]* with a crack defect" "*[cls]* has a visible crack" "cracked areas on *[cls]*" "*[cls]* with surface cracking" "fine cracks found on *[cls]*" "*[cls]* shows crack lines"
Damage	"*[cls]* has a damaged defect" "flawed *[cls]* with damage" "*[cls]* shows signs of damage" "damage found on *[cls]*" "*[cls]* with visible wear and tear" "*[cls]* with structural damage"	Extra	"*[cls]* with extra thing" "*[cls]* has a defect with extra thing" "extra material on *[cls]*" "*[cls]* contains additional pieces" "*[cls]* with extra component defect" "unwanted additions on *[cls]*"
Hole	"*[cls]* has a hole defect" "a hole on *[cls]*" "visible hole on *[cls]*" "*[cls]* has small punctures" "*[cls]* shows perforations" "hole present on *[cls]*"	Melded	"*[cls]* with melded defect" "melded parts on *[cls]*" "*[cls]* has fused areas" "fused spots on *[cls]*" "melded areas on *[cls]*" "*[cls]* with melded material"
Melt	"*[cls]* with melt defect" "melted areas on *[cls]*" "*[cls]* shows melting" "signs of melting on *[cls]*" "*[cls]* with melted spots" "*[cls]* has a melted appearance"	Missing	"*[cls]* with a missing defect" "flawed *[cls]* with something missing" "*[cls]* has missing parts" "missing components on *[cls]*" "absent pieces in *[cls]*" "*[cls]* is incomplete"
Partical	"*[cls]* with particles defect" "*[cls]* has foreign particles" "small particles on *[cls]*" "*[cls]* with unwanted particles" "contaminants found on *[cls]*" "*[cls]* with visible particles"	Scratch	"*[cls]* has a scratch defect" "flawed *[cls]* with a scratch" "scratches visible on *[cls]*" "*[cls]* has surface scratches" "small scratches found on *[cls]*" "*[cls]* with scratch marks"
Spot	"*[cls]* with spot defect" "spots visible on *[cls]*" "flawed *[cls]* with spots" "*[cls]* with visible spotting" "*[cls]* shows small spots" "surface spots on *[cls]*"	Stuck	"*[cls]* with a stuck defect" "*[cls]* stuck together" "*[cls]* has stuck parts" "adhesive issue causing *[cls]* to stick" "*[cls]* is partially stuck" "*[cls]* with adhesion defect"
Weird Wick	"*[cls]* with a weird wick defect" "*[cls]* has an unusual wick" "the wick on *[cls]* appears odd" "*[cls]* with a strangely shaped wick" "irregular wick found on *[cls]*" "odd wick defect on *[cls]*"	Wrong Place	"*[cls]* with defect that something on wrong place" "*[cls]* has a misplaced defect" "flawed *[cls]* with misplacing" "misaligned part on *[cls]*" "*[cls]* shows parts out of place" "misplacement detected on *[cls]*"

< VisA dataset의 모든 결함 종류 >

Method

- 학습 과정
  - **Step 2.** Text/Image embedding
    - CLIP encoder를 통해 text embedding vector z^t와 image embedding vector z^x, e_i^p를 추출
      - Text embedding z^t 은 각 결함 유형에 대해 average pooling을 적용하여 k_1개를 생성
      - Image patch e_i^p 는 각 level 별로 4개의 feature map을 가져옴

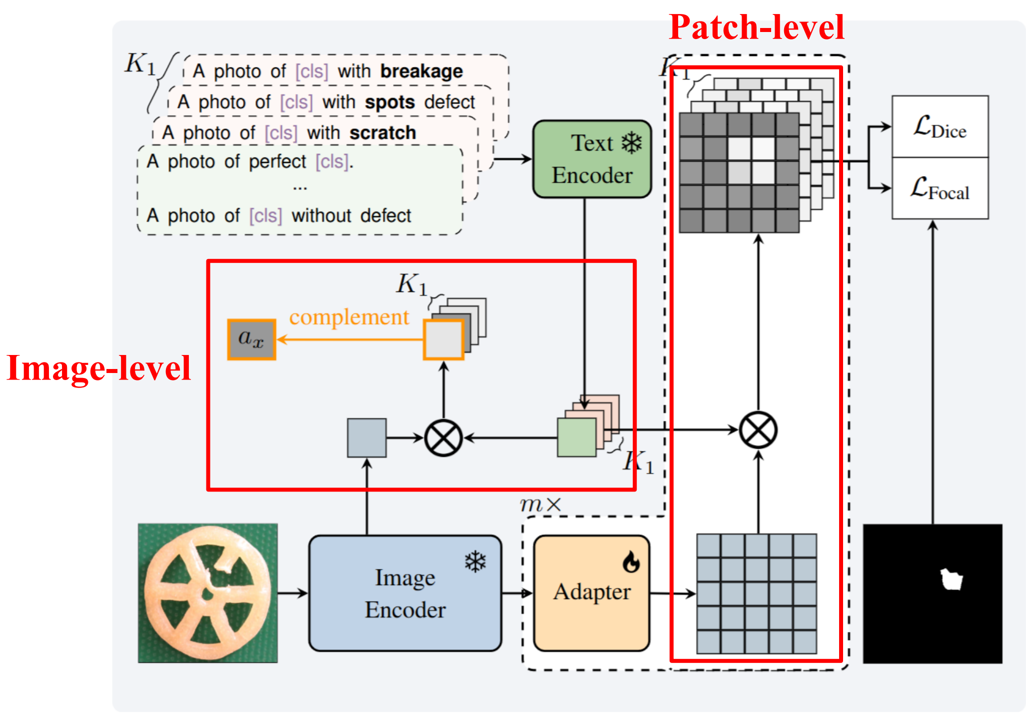


서강대학교 SOGANG UNIVERSITY

VDS LAB

Method

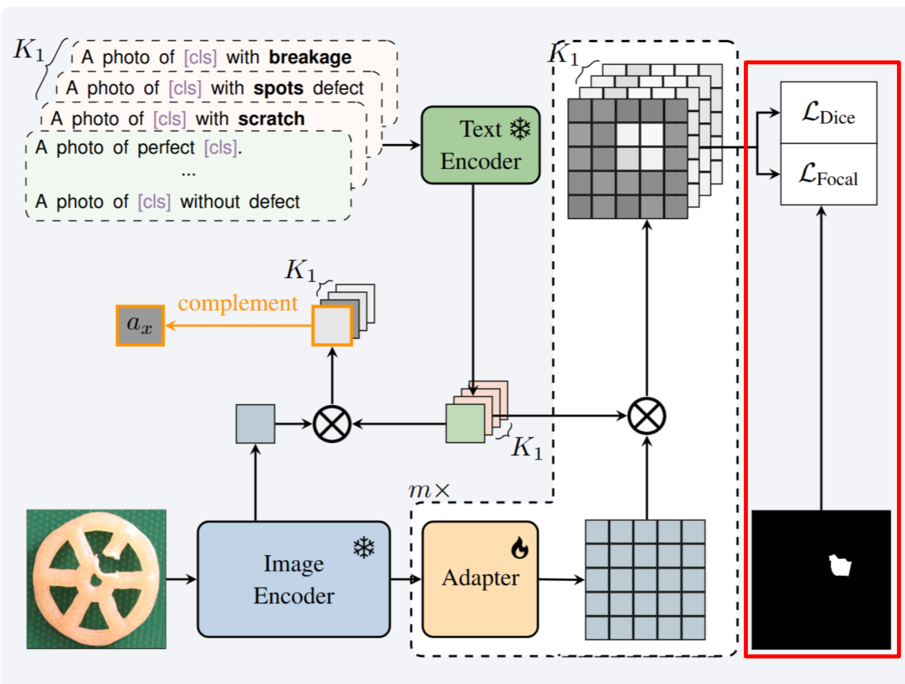
- 학습 과정
  - **Step 3.** Aligning Image Patches and Text Prompts
    - CLIP encoder를 통해 text embedding vector z^t와 image embedding z^x, e_i^p를 추출
      - z^t : 각 결함 유형별로 average pooling을 적용해 k_1개의 text embedding 생성
      - e_i^p : 이미지의 각 level에서 4개의 patch token을 추출하고, single linear layer를 통과해 생성



$\otimes$: Matrix Multiplication

Method

- 학습 과정
  - **Step 4.** Training objective
    - CLIP encoder를 통해 text embedding vector z^t와 image embedding vector z^x, e_i^p를 추출
      - Text embedding z^t 은 각 결함 유형에 대해 average pooling을 적용하여 k_1개를 생성
      - Image patch e_i^p 는 각 level 별로 4개의 feature map을 가져옴



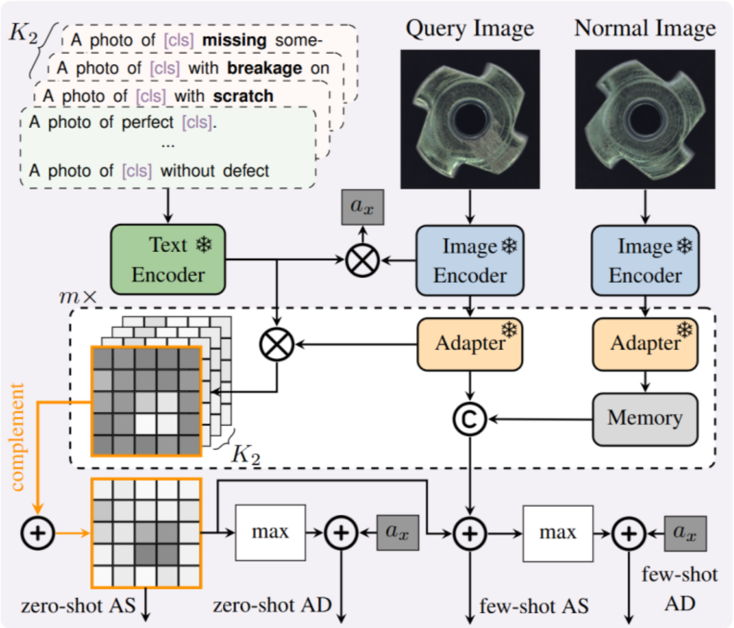
$$\mathcal{L} = \sum_{i=1}^{m} \mathcal{L}_{\text{focal}}(UP(\boldsymbol{S}_i), \boldsymbol{M}'_x) + \mathcal{L}_{\text{dice}}(\boldsymbol{1} - UP(\boldsymbol{S}_i)[0], \boldsymbol{M}_x), \quad (2)$$

< **최종 focal / dice loss 수식** >

1) Sadikaj, Ylli, et al. "MultiADS: Defect-aware Supervision for Multi-type Anomaly Detection and Segmentation in Zero-Shot Learning." *arXiv preprint arXiv:2504.06740* (2025).

Method

- 평가 과정
  - **Zero-shot AD**
    - Text embedding vector와 query image embedding vector 간 similarity를 계산해 **anomaly map 생성**
    - anomaly map의 **최댓값과 평균 anomaly score**를 결합해 최종 anomaly score로 사용
  - **Few-shot AD**
    - Zero-shot AD와 동일하게 진행한 후, **normal image**에 대해서도 image embedding vector를 추출
    - query image embedding과 cosine distance로 **similarity map 생성 후,** text–query image 간 similarity map과 평균을 내어 최종 anomaly map 생성



Experiment

- Implementation details
  - 환경 세팅
    - CLIP: ViT-L-14-336 from OpenCLIP (LAION-400M E32)
    - Learning rate: 1e-4
    - Batch size: 8
    - Features are selected from layers: 6, 12, 18, and 24.

- Anomaly detection
  - 실험
    - Few/zero-shot anomaly detection
    - Super-multi-class anomaly detection

Experiment

- Few/zero-shot anomaly detection
  - Few/Zero-shot 성능 모두 최신 모델들과 비교 시, 우수한 성능을 보임
  - 특히 Image-level 성능은 모든 데이터셋에서 최고 성능을 보임

Table 21. Few-shot anomaly detection and segmentation on the VisA Datasets. April-GAN baseline and our model are trained on the MVTec-AD dataset. (- denotes the results for this metric are not reported in the original paper; **bold** represents the best performer)

Settings		k=1					k=2				
VisA		Pixel-Level		Image-Level			Pixel-Level		Image-Level		
Method	Venue	AUROC	AUPRO	AUROC	F1-max	AP	AUROC	AUPRO	AUROC	F1-max	AP
PaDiM	ICPR21	89.9	64.3	62.8	75.3	68.3	92.0	70.1	67.4	75.7	71.6
CoOp	IJCV22	-	-	-	-	-	-	-	83.5	-	-
PatchCore	CVPR23	95.4	80.5	79.9	81.7	82.8	96.1	82.6	81.6	82.5	84.8
WinCLIP	CVPR23	96.4	85.1	83.8	83.1	85.1	96.8	86.2	84.6	83.0	85.8
April-GAN	CVPR23	96.0	90.0	91.2	86.9	93.3	96.2	90.1	92.2	87.7	94.2
PromptAD	CVPR24	96.7	-	86.9	-	-	97.1	-	88.3	-	-
InCTRL	CVPR24	-	-	-	-	-	-	-	87.7	-	-
AnomalyGPT	AAAI24	96.2	-	87.4	-	-	96.4	-	88.6	-	-
MultiADS (ours)		**97.1**	**92.7**	91.9	**88.3**	93.1	**97.2**	**93.1**	**93.3**	**89.5**	93.9
MultiADS-F (ours)		96.6	91.7	**92**	88.1	**93.9**	96.7	91.9	92.8	88.5	**94.4**

Settings		k=4					k=8				
VisA		Pixel-Level		Image-Level			Pixel-Level		Image-Level		
Method	Venue	AUROC	AUPRO	AUROC	F1-max	AP	AUROC	AUPRO	AUROC	F1-max	AP
PaDiM	ICPR21	93.2	72.6	72.8	78.0	75.6	-	-	78.1	-	-
CoOp	IJCV22	-	-	84.2*	-	-	-	-	84.8	-	-
PatchCore	CVPR23	96.8	84.9	85.3	84.3	87.5	-	-	87.3	-	-
WinCLIP	CVPR23	97.2	87.6	87.3	84.2	88.8	-	-	88.0	-	-
April-GAN	CVPR23	96.2	90.2	92.6	88.4	94.5	96.3	90.2	92.7	88.5	94.6
PromptAD	CVPR24	**97.4**	-	89.1	-	-	-	-	-	-	-
InCTRL	CVPR24	-	-	90.2*	-	-	-	-	90.4	-	-
AnomalyGPT	AAAI24	96.7	-	90.6	-	-	-	-	-	-	-
MultiADS (ours)		96.9	91.1	**93.3**	**89.7**	94.3	**97.4**	**93.5**	**94.7**	**91.3**	94.9
MultiADS-F (ours)		97.0	**91.5**	92.8	88.5	**94.6**	96.9	92.1	93.8	89.5	**95.1**

Table 4. Zero-shot anomaly detection and segmentation. (Bold represents best performer; underline indicates second best performer, * means results are taken from papers)

ZSAD			Pixel-Level		Image-Level	
Dataset	Method	Venue	AUROC	AUPRO	AUROC	AP
VisA	CLIP*	ICML21	46.6	14.8	66.4	71.5
	CLIP-AC*	ICML21	47.8	17.3	65.0	70.1
	CoOp*	IJCV22	24.2	3.8	62.8	68.1
	CoCoOp*	CVPR22	93.6	-	78.1	-
	WinCLIP	CVPR23	79.6	56.8	78.1	81.2
	April-GAN	CVPR23	94.2	86.8	78.0	81.4
	AnomalyCLIP	CVPR24	**95.5**	87.0	82.1	85.4
	AdaCLIP	ECCV24	95	-	75.4	79.3
	MultiADS (ours)		95	**89.7**	**83.6**	**86.9**
	MultiADS-F (ours)		94.5	87.4	82.5	86.5
MPDD	CLIP*	ICML21	62.1	33.0	54.3	65.4
	CLIP-AC*	ICML21	58.7	29.1	56.2	66.0
	CoOp*	IJCV22	15.4	2.3	55.1	64.2
	CoCoOp*	CVPR22	95.2	-	61	-
	WinCLIP	CVPR23	76.4	48.9	63.6	69.9
	April-GAN	CVPR23	94.1	83.2	73.0	80.2
	AnomalyCLIP	CVPR24	**96.5**	88.7	77.0	**82.0**
	AdaCLIP	ECCV24	96.3		66.3	75
	MultiADS (ours)		95.8	**89.7**	78.3	78.4
	MultiADS-F (ours)		96.3	89.5	**79.7**	80.5
MAD-sim	WinCLIP	CVPR23	77.6	55.8	54.3	90.2
	April-GAN	CVPR23	80.4	61.5	56	91
	AnomalyCLIP	CVPR24	77.9	40.1	54.6	90.9
	AdaCLIP	ECCV24	85.7	-	55.2	90.5
	MultiADS (ours) / MultiADS-F (ours)		**88.0**	**74.2**	**57.1**	**94.4**
MAD-real	WinCLIP	CVPR23	60.5	26.9	64.1	87.6
	April-GAN	CVPR23	88.2	69.5	62.9	87.7
	AnomalyCLIP	CVPR24	88.3	65.1	66.8	90
	AdaCLIP	ECCV24	85.7	-	59	86.5
	MultiADS (ours)		89.7	74.0	78.3	**92.9**
	MultiADS-F (ours)		**90.7**	**75.2**	**78.5**	**92.9**
Real-IAD	WinCLIP	CVPR23	87.1	59.9	75	72.3
	April-GAN	CVPR23	96	86.8	75.7	73.5
	AnomalyCLIP	CVPR24	96.2	85.7	78.4	76.7
	AdaCLIP	ECCV24	95.3	-	70.1	68.5
	MultiADS (ours)		**96.6**	87.1	**78.7**	**79.1**
	MultiADS-F (ours)		96.3	**87.2**	78.2	78.5

Experiment

- Ablation
  - Knowledge Base for Anomalies(KBA)검증 실험
    - KBA를 기반으로 생성한 prompt의 효과를 검증하는 실험을 진행
    - KBA를 사용한 경우, 사용하지 않은 경우보다 성능이 향상됨 (VisA에서 6.6%, MAD-sim에서 1.0% 향상)
  - Seen/Unseen defect type 별 성능 분석
    - 학습 단계에서 관찰된 결함 유형(scratch, hole 등)은 쉽게 검출됨
    - 학습 단계에서 관찰되지 않은 결함 유형(mismatch, stuck 등)은 상대적으로 낮은 정확도를 보임
    - 그러나 학습 데이터에 존재하지 않는 결함에 대해서도 일정 수준 이상의 일반화 성능을 유지함

Table 3. Ablation studies on the role of KBA for MTAS

	MVTec → VisA			MVTec → MAD-sim		
KBA	AUROC	F1-score	AP	AUROC	F1-score	AP
-	87.0	22.1	23.6	91.1	25.1	26.5
✓	93.6	22.3	24.8	92.1	27.9	31.5

< KBA 적용 여부에 따른 성능 비교 >

(a) VisA

	Defects	ROC	F1-Score	AP
-	Extra	94.07	2.11	0.15
-	Stuck	91.54	10.51	7.76
✓	Bent	96.53	6.07	7.74
✓	Hole	99.55	12.64	25.19

(b) Real-IAD

	Defects	ROC	F1-Score	AP
-	Pit	97.08	6.15	1.01
✓	Contamin.	90.03	6.12	1.86
✓	Scratch	92.63	4.37	2.96
✓	Damage	96.61	6.31	9.75

(c) MAD-sim

	Defects	ROC	F1-Score	AP
-	Burrs	95.56	1.18	1.67
✓	Missing	86.52	2.56	3.08
✓	Stains	98.19	15.02	9.92

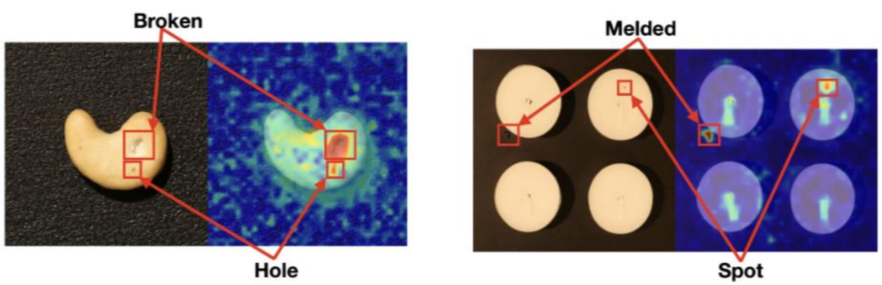
(d) MPDD

	Defects	ROC	F1-Score	AP
-	Mismatch	88.44	2.56	1.04
-	Flattening	96.72	36.06	8.33
✓	Scratch	96.67	26.99	20.26

< Seen/Unseen 결함 유형별 세부 성능 분석 >

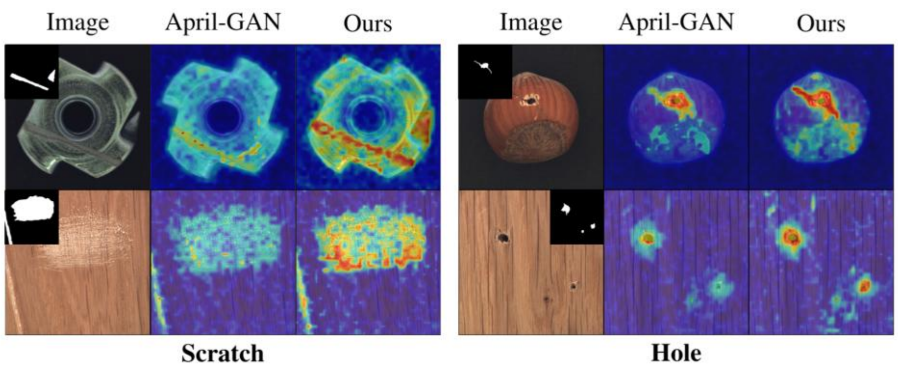
Experiment

- Ablation
  - 단일 이미지 내 multi-defect type 검출 시각화
    - 하나의 이미지 안에 존재하는 서로 다른 유형의 결함들을 동시에 검출 가능함을 시각적으로 확인
  - 비교 모델과의 정성적 성능 비교
    - April-GAN 등 기존 방법과의 시각적 결과를 비교
    - 제안된 방법이 복합적이고 미세한 결함까지 잘 검출하는 성능을 보여줌
    - Anomalous region에 anomaly score가 더 높으며, 이는 normal과 abnormal의 distribution 차이가 더 크다는 것을 의미함



(a) Broken and Hole defects. (b) Melded and Spot defects.

< 단일 이미지 내 multi-defect type 검출 시각화 >



< 비교 모델과의 정성적 성능 비교>

감사합니다.