Anomaly Detection under Distribution shift

2026 Winter seminar

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

Presented By
김건우

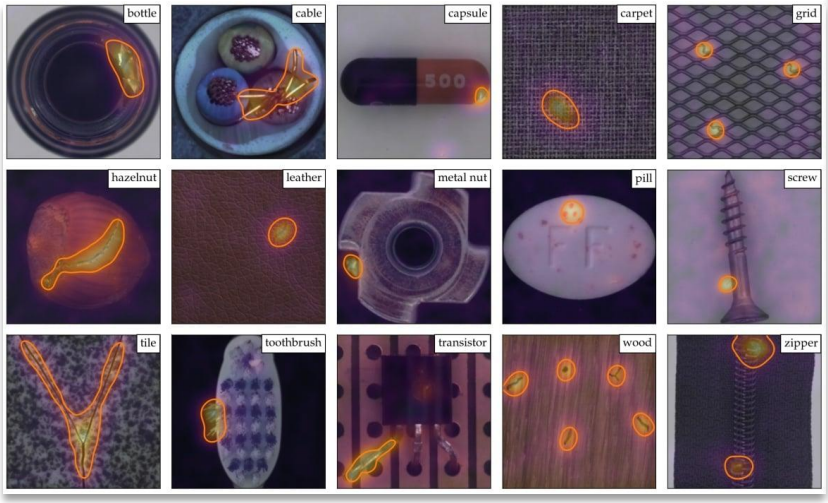
Outline

- Background
  - Anomaly detection
  - Distribution shift
- Paper 1
  - *Anomaly Detection under Distribution Shift* [ICCV 2023]
- Paper 2
  - *Filter or Compensate: Towards Invariant Representation from Distribution Shift for Anomaly Detection* [AAAI 2025]

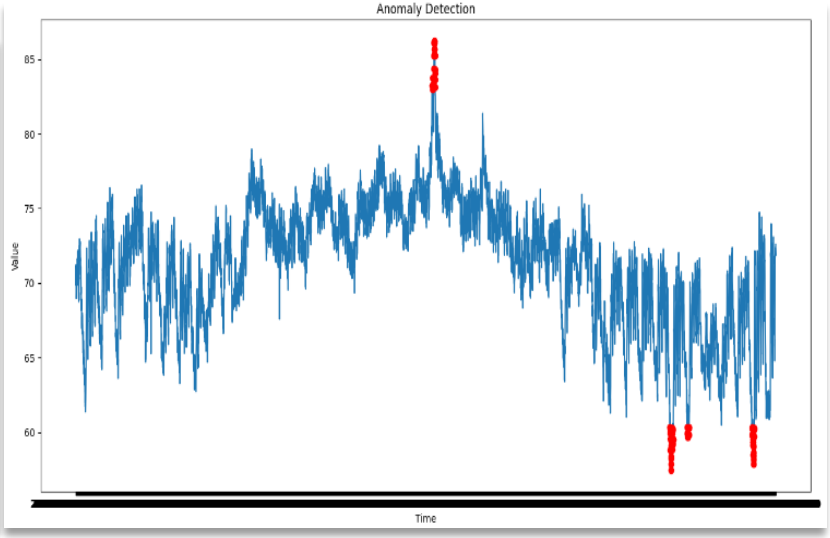
Background

Anomaly Detection(이상 탐지)

- 일반적인(Normal) 패턴에서 벗어난 이상(Anomaly)를 탐지하는 기법
- 활용 분야
  - 산업 양품 검사
  - 의료 영상
  - 보안 및 침입 탐지



< Image AD >



< Time Series AD >

Background

Distribution Shift in Anomaly Detection

- 개념
  - 기존 AD는 train distribution과 test distribution이 동일함
- 문제
  - 학습 시 보지 못한 환경적 변화(shift)를 모델이 '이상(Anomaly)'로 착각함

    ∴ False Positive의 급증

< Train data>

< Test data>

Anomaly Detection under Distribution Shift [ICCV 2023]

Introduction

Distribution shift

- In-Distribution(ID)
  - 정의
    - 모델이 학습할 때 본 데이터와 동일한 통계적 특성(Style, 조명, 배경 등)을 가진 데이터셋
  - 학습 데이터
    - 오직 "정상(Normal)" 샘플들로만 구성.
  - 테스트 데이터
    - 정상 샘플
      - ✓학습 데이터와 거의 똑같은 환경에서 찍힌 정상 제품
    - 이상(Anomaly) 샘플
      - ✓동일 환경이지만 결함(스크래치, 오염 등)이 있는 샘플
  - 전통적인 가정
    - 기존 AD 연구들은 "테스트 데이터는 항상 ID 환경에서 들어온다"고 가정

Introduction

Distribution shift

- Out-of-Distribution(OOD)
  - 정의
    - 학습 데이터와 의미적(Semantic)으로는 같지만 외형적(Photometric/Style)으로 큰 차이(Shift)가 생긴 데이터셋
  - 발생 원인
    - 새로운 조명 조건, 물체의 포즈 변화, 배경의 변경, 카메라 노이즈 등 실세계의 자연스러운 변화
  - OOD 테스트 데이터
    - 정상 샘플
      - ✓제품은 정상이지만, 조명이 너무 어둡거나 배경 색이 바뀐 경우
    - 이상 샘플
      - ✓바뀐 환경(조명/배경)에서 나타난 결함 샘플
  - 문제점
    - 기존 모델은 OOD 정상 샘플을 보고 "학습 때 본 적 없는 스타일"이라며 이상치(Anomaly)로 오인(False Positive)하는 경우가 많음

Introduction

Contribution

1. AD 분야의 새로운 문제 제기 및 벤치마크 구축
   - 문제 정의
     - 기존 Anomaly Detection(AD) 연구들이 간과했던 'Distribution Shift(분포 변화)' 상황에서의 취약성을 제기
   - 벤치마크 수립
     - 4가지 주요 데이터셋(MVTec-C, CIFAR-10-C, MNIST-M, PACS)을 활용하여 분포 변화가 발생했을 때 AD 모델의 성능을 평가할 수 있는 표준화된 실험 환경을 구축
   - 한계점 증명
     - 기존의 SOTA AD 모델이나 일반적인 OOD Generalization 기법들을 단순히 결합하는 것만으로는 이 문제를 해결할 수 없음을 실험적으로 입증

Introduction

Contribution

2. Generalized Normality Learning (GNL) 방법론 제안

- 분포 변화에 강건한(Robust) 특징을 뽑아내기 위해 GNL이라는 새로운 프레임워크를 제안
- 두 가지 핵심 메커니즘

학습 단계 (Distribution-invariant Normality Learning)

✓다양한 데이터 증강(Augmentation)을 통해 가상의 분포 변화를 생성

✓Normality-preserved Loss

- 스타일이 변하더라도 '정상성'을 나타내는 핵심 의미(Semantics)는 변하지 않도록 추상적/저수준 특징 공간에서 일관성을 학습

추론 단계 (AD-oriented Test Time Augmentation)

✓테스트 시점에 들어온 OOD 샘플의 스타일을 학습 데이터의 스타일과 맞추는 Feature Distribution Matching (FDM) 기술을 적용하여 분포 간극을 좁힘

Introduction

Contribution

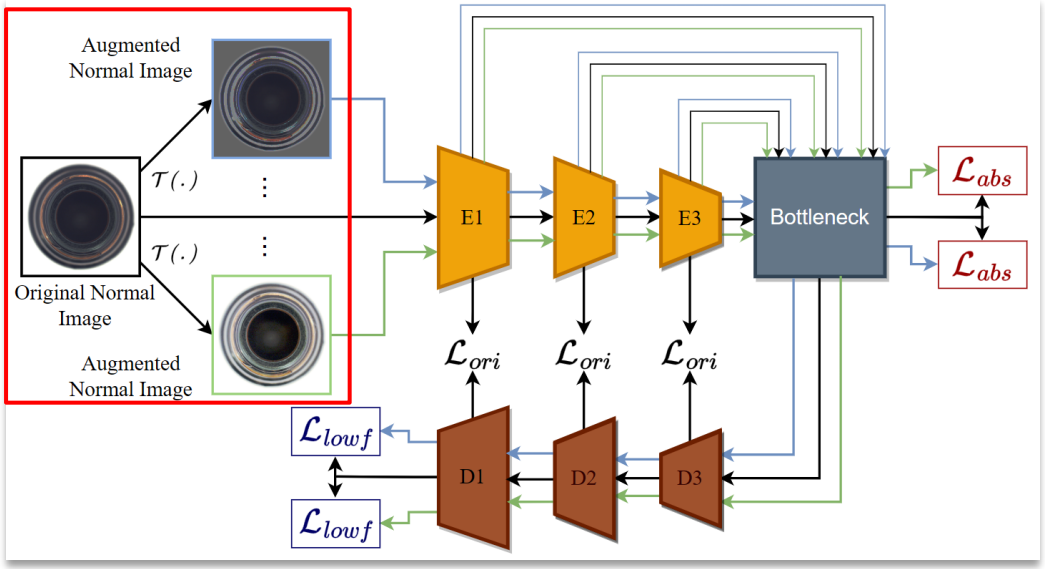
3. 압도적인 성능 향상 및 실증적 검증

- 성능 우위
  - 다양한 분포 변화 상황에서 기존 SOTA 모델들보다 AUROC 기준 10% 이상의 큰 성능 향상을 기록
- 안정성 유지
  - OOD 상황에서의 성능을 올리면서도, 원래의 ID(In-Distribution) 환경에서의 탐지 정확도를 희생하지 않고 유지한다는 점을 증명
- 강건성 증명
  - 분포 변화의 강도(Severity)가 세밀하게 변하는 환경에서도 모델이 성능 급락 없이 안정적으로 작동함을 보여줌

Method

Distribution-invariant Normality Learning (DINL)

- 핵심 기법
  - 데이터 증강(AugMix) + 유사성 유지 손실 함수(Normality-preserved Loss)
- ***Augmix***
  1. 원본 정상 이미지 x 다른 여러 증강 이미지 x'_k를 만듦
  2. Original image와 새로운 이미지 x'_k를 Encoder에 넣음

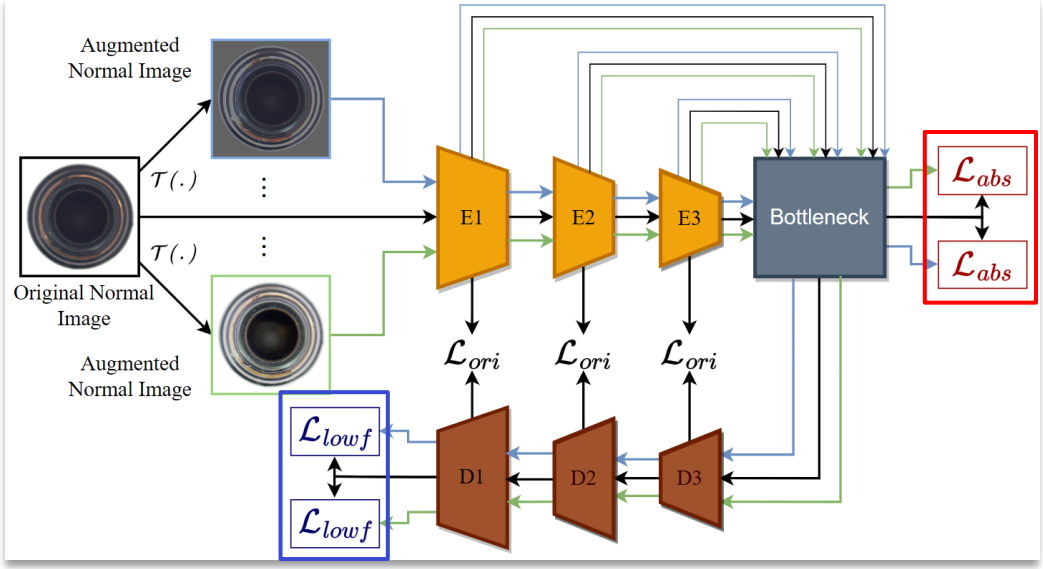


< Training >

Method

Distribution-invariant Normality Learning (DINL)

- Bottleneck
  - $\mathcal{L}_{abs}$: Global 정보인 고차원적인 Semantics 정보를 학습
- Decoder last block
  - $\mathcal{L}_{low\ f}$: Local 정보인 Edge, Texture와 같은 Structure 정보를 학습



< Training >

$$\mathcal{L}_{\text{abs}} = \sum_{k=1}^{N} \frac{1}{N} \left\{ \mathcal{L}_{sim}(\phi(x), \phi(x'_k)) \right\}$$

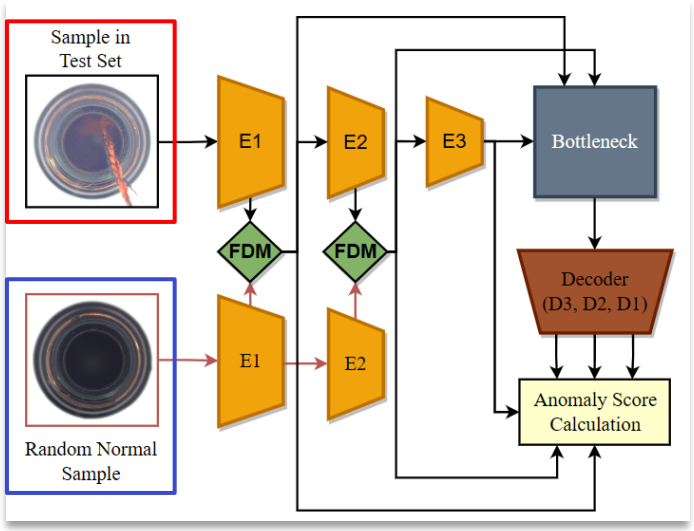
$$\mathcal{L}_{lowf} = \sum_{k=1}^{N} \frac{1}{N} \left\{ \mathcal{L}_{sim}(\omega(\phi(x)), \omega(\phi(x'_k))) \right\}$$

- $\phi(x)$: Feature vector
- $\omega(\phi(x))$: decoder의 마지막 출력 feature

Method

AD-oriented Test Time Augmentation (ATTA)

- EFDM (Exact Feature Distribution Matching)
  - Test 중 데이터 분포 간의 불일치 문제를 해결
  - Sample in Test Set
    - Normal인지 Anormal인지 모르는 샘플(ODD일 가능성이 높음)
  - Random Normal Sample
    - ID Normal data를 무작위로 뽑음



< Inference >

Method

AD-oriented Test Time Augmentation (ATTA)

- FDM (Feature Distribution Matching)
  - Test feature(C)와 random하게 뽑아온 normal feature (V)의 값들을 오름차순 정렬

$$\text{Test feature}(C): \begin{matrix} 30 & 10 & 40 \\ 5 & 14 & 42 \\ 20 & 60 & 50 \end{matrix}, \text{Random normal feature } (V): \begin{matrix} 150 & 200 & 105 \\ 350 & 360 & 180 \\ 400 & 700 & 600 \end{matrix}$$

✓ Test feature(C): [5, 10,14, 20, 30, 40, 42, 50, 60]

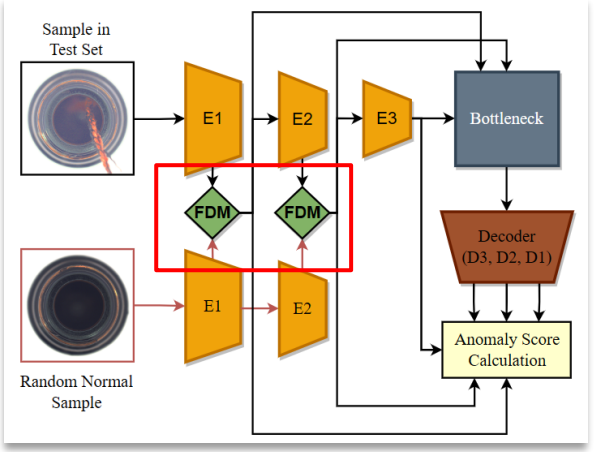
✓ Random normal feature (V): [105, 150, 200, 180, 350, 360, 400, 600, 700]

  - 정렬된 순서에 맞춰 teste feature 값을 normal feature의 값으로 보정

$$\text{If, } \alpha = 0.5 \text{) FDM}(C): \begin{matrix} 190 & 80 & 200 \\ 55 & 107 & 221 \\ 100 & 380 & 325 \end{matrix}$$

$$\text{FDM}(\mathcal{C}, \mathcal{V}, \alpha) : \mathcal{C}_{\tau_i} = (1-\alpha)\mathcal{C}_{\tau_i} + \alpha\mathcal{V}_{\kappa_i}$$

Feature의 내용을 완전히 바꾸는 것이 아닌
OOD의 상황을 익숙한 ID 상황에 가깝게 만들어주는 것!



< Inference >

Experiment

Result

	ID	OOD			
Method	MVTec	Brightness	Contrast	Blur	Noise
Deep SVDD	69.98	55.18	50.07	68.82	59.11
f-AnoGAN	75.65	48.36	49.29	37.98	39.10
KDAD	85.50	83.81	64.03	84.17	82.04
RD4AD	**98.64**	96.50	94.12	**98.9**	90.14
Augmix	96.29	95.10	94.51	95.39	90.99
Mixstyle	98.58	96.60	94.45	98.27	88.92
EFDM	98.64	96.78	94.77	98.25	89.29
Augmix+Mixstyle	96.78	96.86	94.57	98.73	90.12
Augmix+EFDM	97.04	96.83	95.21	98.11	90.18
Jigsaw	73.97	73.36	67.88	73.88	72.60
GNL (Ours)	97.99	**97.43**	**97.46**	97.77	**94.10**

< AUROC on MVTec >

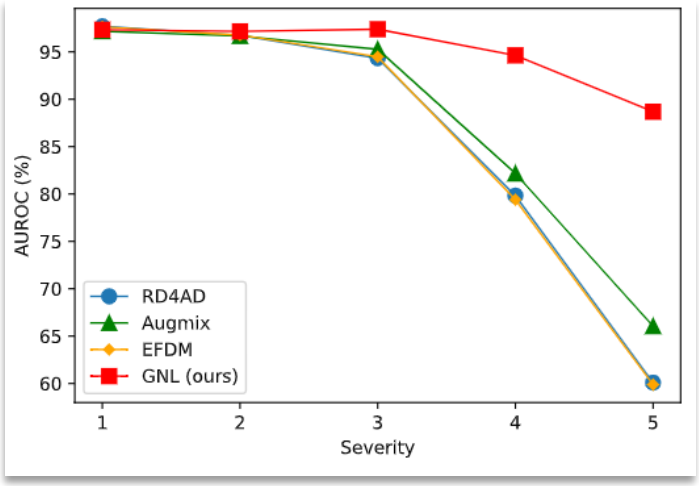
	ID	OOD			
Method	CIFAR	Brightness	Contrast	Blur	Noise
Deep SVDD	64.62	59.13	55.94	62.13	54.46
f-AnoGAN	70.25	54.62	57.23	60.74	51.76
KDAD	84.21	75.91	64.37	63.49	56.87
RD4AD	**84.62**	75.89	65.34	66.67	58.82
Augmix	82.83	74.15	62.48	**66.92**	57.36
Mixstyle	83.68	76.07	63.87	65.74	57.74
EFDM	83.92	76.19	63.92	64.81	57.63
Augmix+Mixstyle	83.87	76.02	65.55	63.89	58.04
Augmix+EFDM	82.96	75.73	64.39	63.83	57.14
Jigsaw	71.29	66.86	61.45	60.12	55.29
Ours	82.29	**77.94**	**66.13**	64.04	**61.51**

< AUROC on CIFAR - 10 >

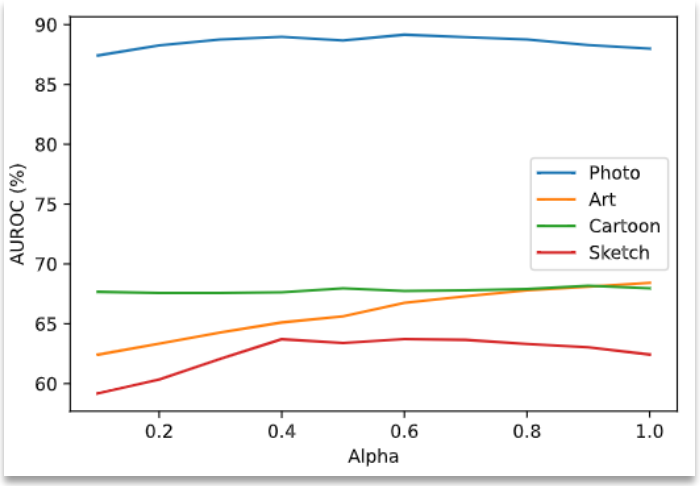
Experiment

Result

$$\mathrm{FDM}(\mathcal{C}, \mathcal{V}, \alpha) : \mathcal{C}_{\tau_i} = (1-\alpha)\mathcal{C}_{\tau_i} + \alpha\mathcal{V}_{\kappa_i}$$



< varying severityof the 'Contrast' corruption >



< AUROC results using varying α>

Filter or Compensate: Towards Invariant Representation from Distribution Shift for Anomaly Detection

[AAAI 2025]

Introduction

선행 연구에 대한 지적 GNL(Generalized Normality Learning)

- 선행 연구 내용
  - Train 과 inference 단계 모두에서 학습 데이터(ID)와 분포가 다른(OOD) 데이터 사이의 간극을 최소화하여 다양한 분포 변화에도 견고한 이상 탐지를 목표로 함
- 문제점
  - GNL은 단순히 서로 다른 증강(Augmentation) 뷰 사이의 일관성만을 강조
    - 각 분포가 가진 고유한 특징들을 너무 강제로 지워버림
      - ✓이전 블록들의 고차원적인 시맨틱 정보까지 손실되는 부작용이 발생
- 해결 방안
  - GNL은 분포 차이를 무시하려 했음
    - FiCo는 student network에 부족한 분포 정보를 채워넣음(Compensate)
      - ✓Teacher network와의 간극을 메움

Method

Distribution-Invariant Filter Module

1. S-T network 간의 정렬 문제 해결
   - DiSCo 모듈
     - 분포 특화 정보를 보충해 주는 분포 특화 보상
       - ✓교사와 학생 네트워크 사이의 정보 차이를 줄였습니다.

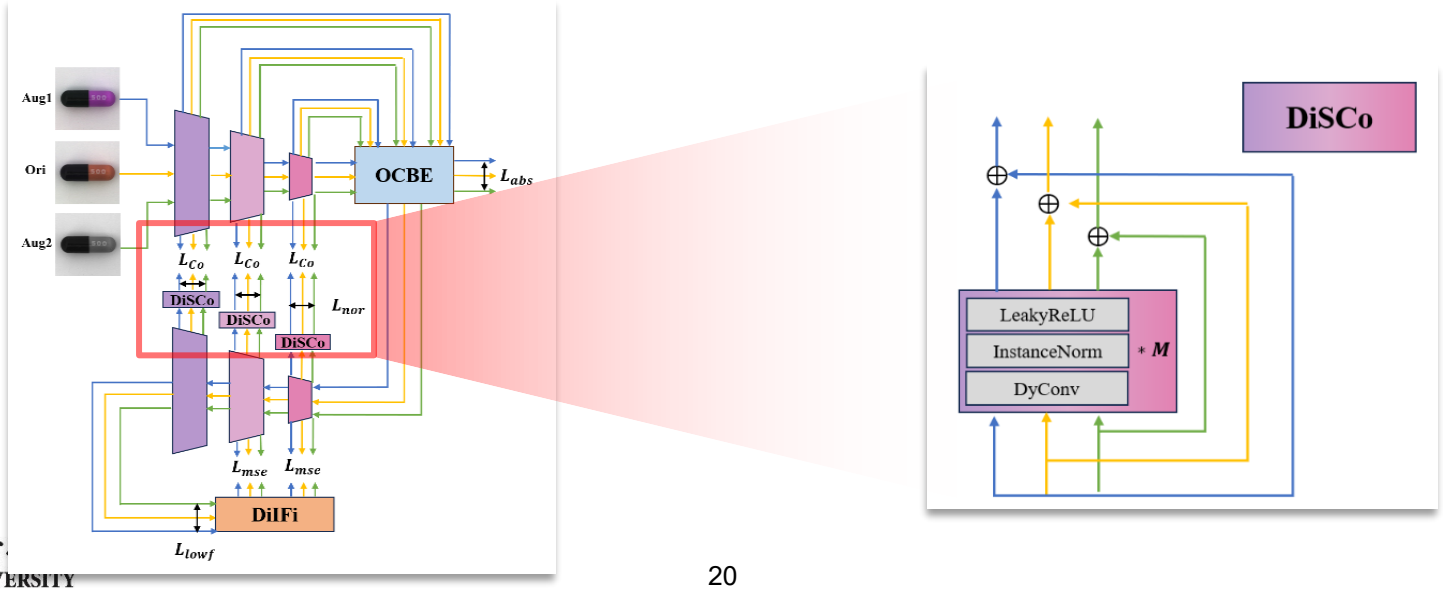
2. 분포 불변 정상성 학습을 위한 새로운 프레임워크
   - DiIFi 모듈
     - 이상 패턴(anomalous patterns)과 분포 특화 정보(distribution-specific information)를 모두 걸러내는 분포 불변 필터
       - ✓분포 변화에 상관없는 정상성을 얻음

3. 광범위한 시나리오에서의 압도적인 성능 입증
   - MVTec, PACS, CIFAR-10 등 3가지 이상의 AD 벤치마크 실험을 통해 OOD 상황뿐만 아니라 ID에서도 기존 RD 기반 모델들을 뛰어넘는 성능을 보임

Method

DiSCo (Distribution-Specific Compensation Module)

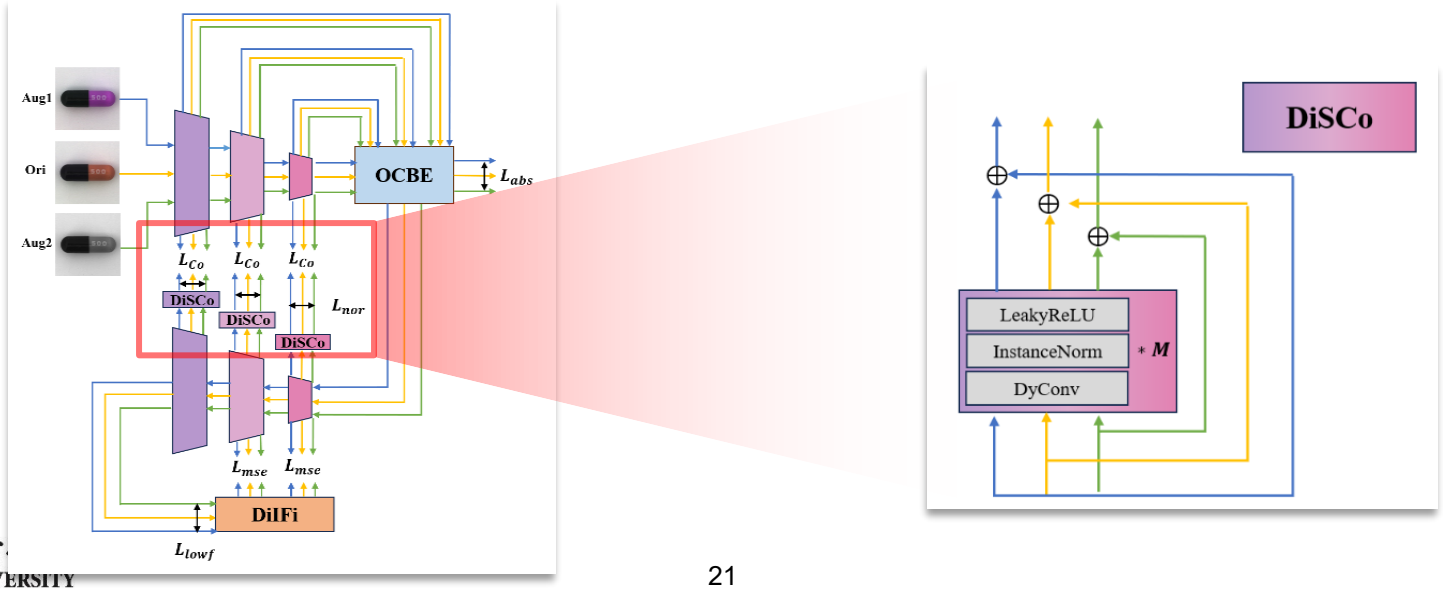
- 핵심 역할
  - 정보 비정렬 해결
    - S-T Network 사이의 Alignment을 보장하기 위해 Student에서 누락된 '분포 특화 정보'(Noise, style 등)를 직접 보충
  - 오검출 방지
    - 테스트 환경의 변화(분포 변화)를 모델이 이상치(Anomaly)로 착각하여 오작동하는 것을 방지하는 역할을 함



Method

DiSCo (Distribution-Specific Compensation Module)

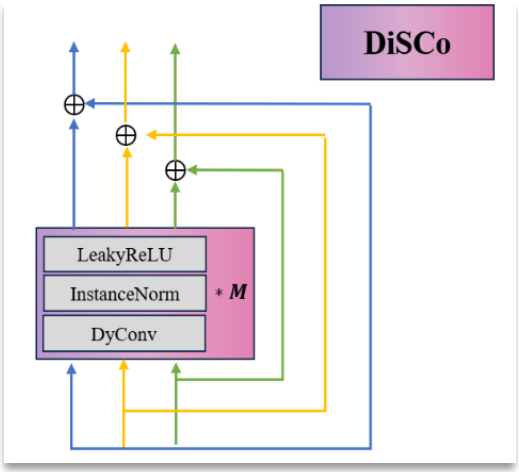
- 작동 원리
  - 동적 특징 추출
    - Image 마다 다른 다양한 특징 분포를 처리하기 위해 Dynamic Convolution(DyConv)을 사용하여 스타일 정보 정규화에 강한 Instance Normalization을 결합합니다.
  - 잔차 학습 (Residual Learning)
    - 기존의 특징 정보를 보존하면서 부족한 정보만 더하기 위해 Shortcut(지름길 연결) 방식을 채택
    - 테스트 환경의 변화(분포 변화)를 모델이 Anomaly로 착각하여 오작동하는 것을 방지하



Method

DiSCo (Distribution-Specific Compensation Module)

- 실행 순서
  - 입력: Student Network에서 생성된 feature map을 DiSCo 모듈에 입력
  - DyConv
    - Input 특성에 따라 여러 개의 Conv에 서로 다른 Attention을 주어 동적으로 필터를 생성
  - InstanceNorm (Instance Normalization)
    - 각 이미지 샘플의 채널별로 평균과 표준편차를 계산하여 정규화
  - LeakyReLU
    - 비선형성 부여 및 정보 손실 방지

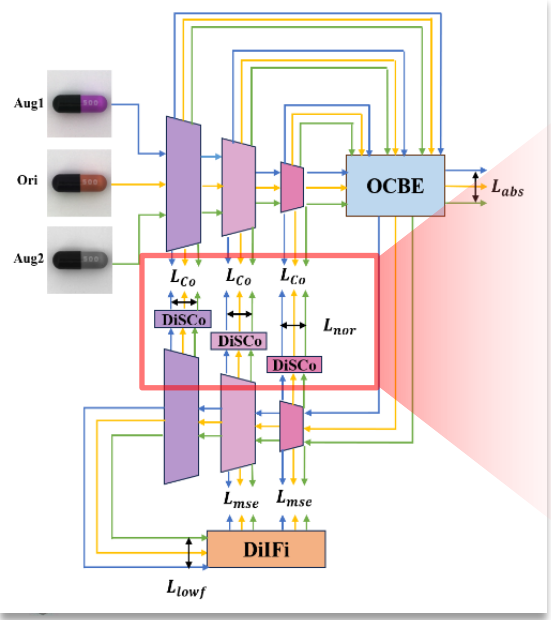


< DiSCo >

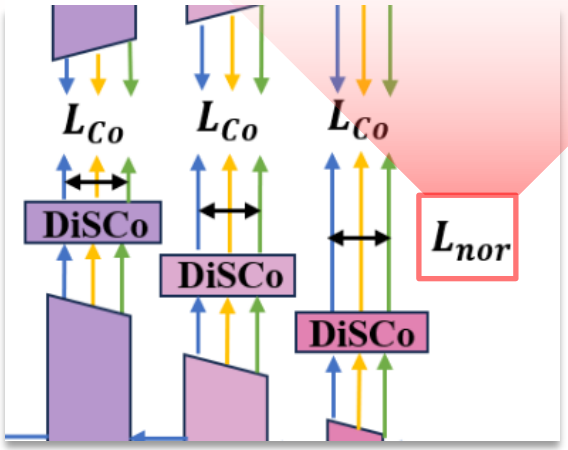
Method

DiSCo (Distribution-Specific Compensation Module)

- L_{nor} (Normality Consistency Loss)
  - 목적
    - 정상성 붕괴(Normality Collapsing) 방지
  - Original Image와 Augmentation Image간의 Cosine Similarity
    - Noise, light가 바뀌어도 image가 가진 배경 분포 특징을 유지하기 위해



$$\mathcal{L}_{nor} = \sum_{n=1}^{N} \{1 - \frac{(C_1(f^{D_1}))^T \cdot (C_1(f_n^{D_1}))}{\|C_1(f^{D_1})\| \|C_1(f_n^{D_1})\|}\}$$



- C_1: Student network의 마지막 Block
- f^{D_1}: Ori Image의 Decoder 출력 feature
- $f_n^{D_1}$: Aug Image의 Decoder 출력 feature

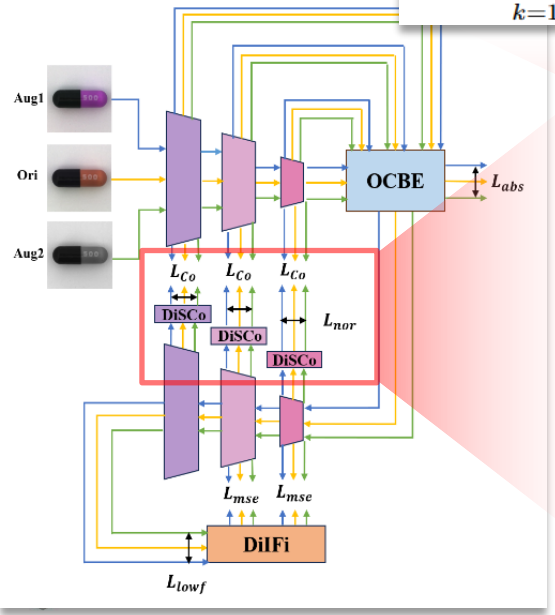
VDS LAB

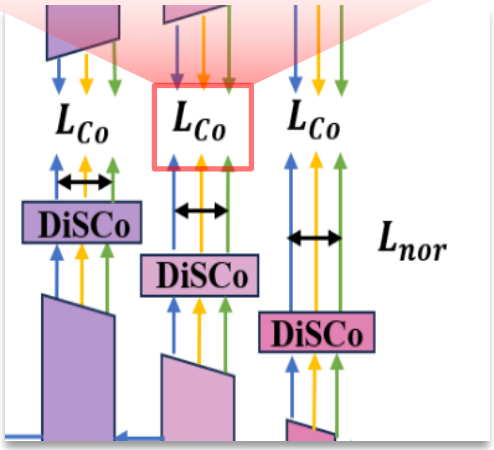
Method

DiSCo (Distribution-Specific Compensation Module)

- L_{Co} (Distribution-Specific Compensation Loss)
  - 목적
    - T-Network와 S-Network 사이의 Misalignment 문제를 해결
  - Encoder's feature map과 decoder's feature map간의 Cosine Similarity
    - Ori와 Aug 동시에 진행하여 일반화 성능 향상

$$\mathcal{L}_{Co} = \sum_{k=1}^{K}\{1-\frac{f^{E_k}\cdot f_F^{D_k}}{\|f^{E_k}\|\|f_F^{D_k}\|}\}+\alpha\sum_{n=1}^{N}\sum_{k=1}^{K}\{1-\frac{f_n^{E_k}\cdot f_{F,n}^{D_k}}{\|f_n^{E_k}\|\|f_{F,n}^{D_k}\|}\}$$





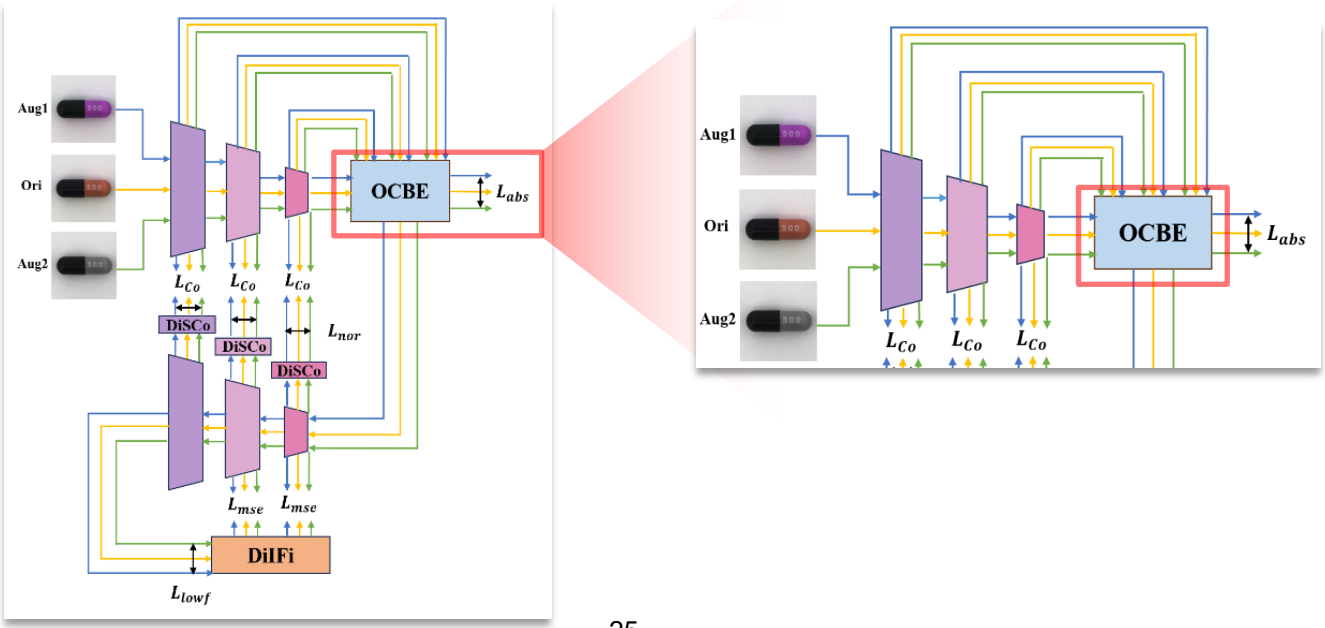
- f^{E_k}: Ori image가 Encoder에서 나온 feature map
- $f_F^{D_k}$: Ori image Decoder에서 나온 최종 feature map
- $f_n^{E_k}$: Aug image가 Encoder에서 나온 feature map
- $f_{F,n}^{D_k}$: Aug image가 Decoder에서 나온 최종 feature map

VDS LAB

Method

OCBE (One-Class Bottleneck Embedding)

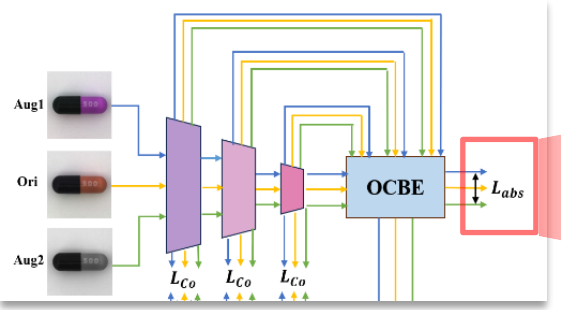
- 핵심 역할
  - Multi-scale Feature Aggregation
    - Encoder의 3개 레이어 출력을 16×16 해상도로 통합
      - ✓다양한 스케일의 정보를 하나로 합침
  - Bottleneck Compression
    - 3072 → 2048 channel로 정보 압축



Method

OCBE (One-Class Bottleneck Embedding)

- L_{abs} (Distribution-Specific Compensation Loss)
  - 역할
    - Distribution-Invariant Normality
      - ✓밝기, 색감, 노이즈(Augmented View)가 있어도 OCBE(병목 지점)을 통과한 뒤에는 똑같은 '정상 상태'의 값으로 나와야 한다고 강제
        - 여기서 앞서 비판했던 문제가 발생
          - Aug에서 가지고 있는 정보까지 ori 기준으로 맞추게 됨



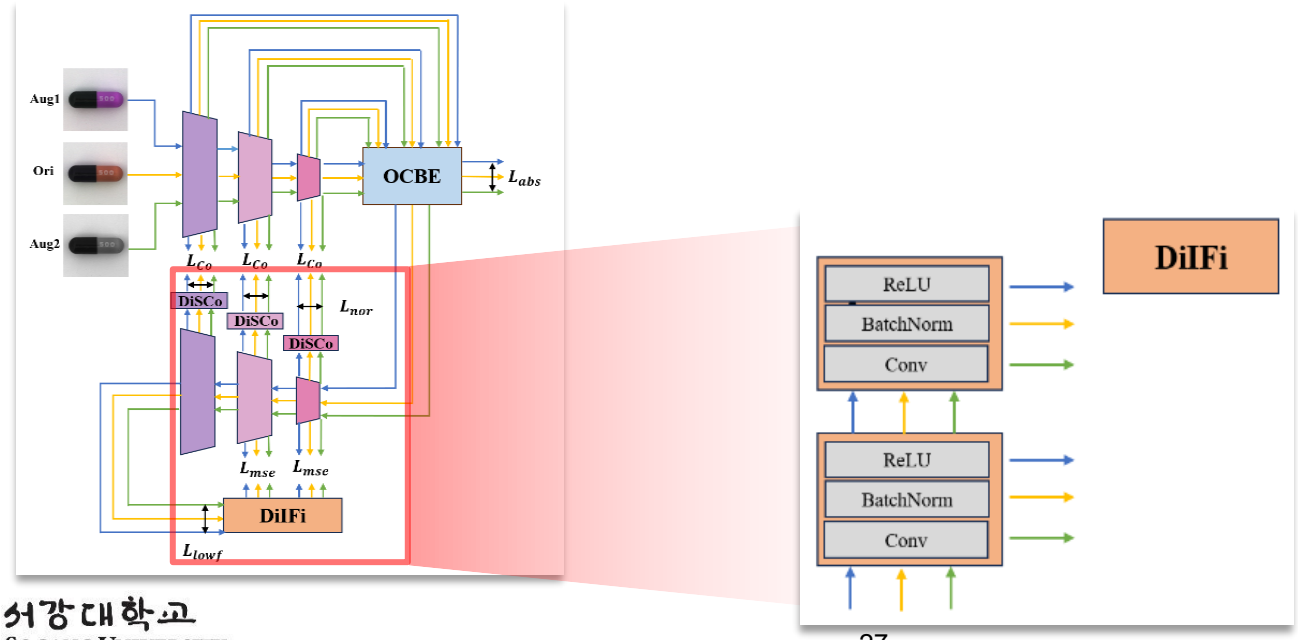
$$\mathcal{L}_{abs} = \sum_{n=1}^{N} \{1 - \frac{(f^{\phi})^T \cdot f_n^{\phi}}{\|f^{\phi}\| \|f_n^{\phi}\|}\}$$

- f^{ϕ}: OCBE를 거친 ori image의 임베딩 벡터
- f_n^{ϕ}: OCBE를 거친 aug image의 임베딩 벡터

Method

DilFi (Distribution-Invariant Filter)

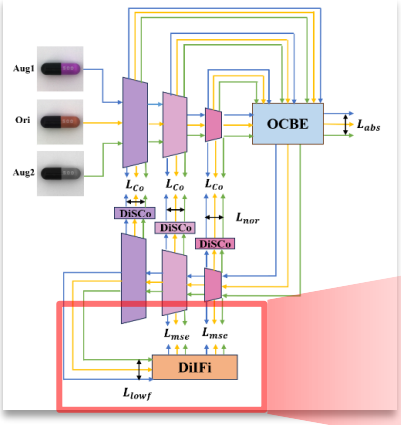
- 핵심 역할
  - Multi-scale Filtering
    - 마지막 출력 단계에서만 noise를 거르던 기존 방식(GNL)과 달리 decoder의 모든 층에서 동시에 필터링을 수행
  - Normality 보존 (Semantic Protection)
    - 앞단의 DisCo에서 Noise에 대한 loss를 구했다면 여기서는 normal semantic 정보에 집중

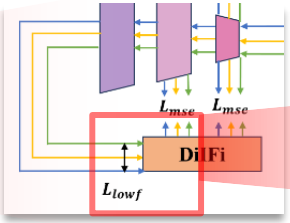


Method

DilFi (Distribution-Invariant Filter)

- L_{Lowf} (Distribution-Specific Compensation Loss)
  - 역할
    - Distribution shift를 정확하게 정의하는 기준
      - ✓마지막 decoder block의 local 정보로 어떤 것이 noise(방해요소)인지 object의 특성인지를 정의함
    - 이전 decoder에 필터링할 기준 전달
      - ✓중간 블록(Block 2, 3)들은 정보가 너무 복잡해서 스스로 노이즈를 찾아내기 어려움
      - ✓마지막 블록에서 정의하는 noise와 object의 정보를 전달





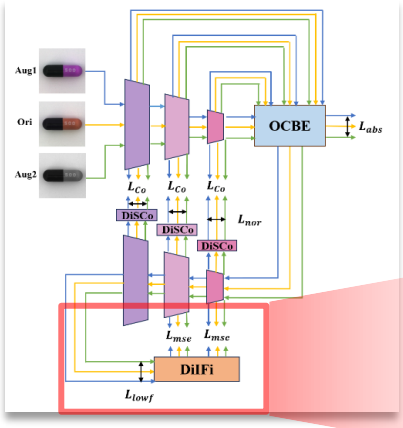
- f^{D_1}: Ori image가 decoder 가장 마지막 block에서 나온 feature map
- $f_n^{D_1}$: Aug image가 decoder 가장 마지막 block에서 나온 feature map

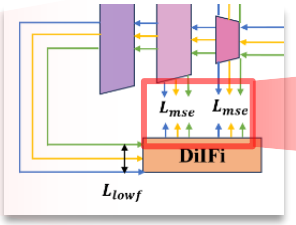
$$\mathcal{L}_{lowf} = \sum_{n=1}^{N}\{1 - \frac{(f^{D_1})^T \cdot (f_n^{D_1})}{\|f^{D_1}\|\|f_n^{D_1}\|}\}$$

Method

DilFi (Distribution-Invariant Filter)

- L_{mse} (Distribution-Specific Compensation Loss)
  - 역할
    - 필터링 강도 강화 (Filter Strengthening)
      - ✓ DilFi를 통해 얻은 noise(방해요소)와 object의 특성을 decoder에 전달
        - Filtering 기준을 업데이트
      - ✓Decoder의 Distribution을 판단하는 능력을 향상





- $C_k(f^{D_k})$: Ori image가 decoder에서 k block에서 나온 feature map
- $f_k^{D_1}$: DilFi를 통해 나온 ori image의 feature map (Distribution shift을 정의하는 기준)
- $C_k(f_n^{D_k})$: Aug image가 decoder에서 k block에서 나온 feature map
- $f_{k,n}^{D_1}$: DilFi를 통해 나온 Aug image의 feature map (Distribution shift을 정의하는 기준)

$$\mathcal{L}_{mse} = \sum_{k=2}^{K} \left(C_k(f^{D_k}) - f_k^{D_1}\right)^2 + \sum_{n=1}^{N} \sum_{k=2}^{K} \left(C_k(f_n^{D_k}) - f_{k,n}^{D_1}\right)^2$$

Experiment

Result

Method	ID	OOD				Avg.
	Ori	Br	Co	Bl	No	
Deep SVDD (Ruff et al. 2018)	70.0	55.2	50.1	68.8	59.1	60.6
f-AnoGAN (Schlegl et al. 2019)	75.7	48.4	49.3	38.0	39.1	50.1
KD (Salehi et al. 2021)	85.5	83.8	64.0	84.2	82.0	79.9
PatchCore† (Roth et al. 2022)	99.1	96.0	92.1	97.2	93.9	95.7
SimpleNet† (Liu et al. 2023)	99.4	90.6	71.7	91.6	76.1	85.9
PNI† (Bae, Lee, and Kim 2023)	99.6	87.8	67.6	90.2	66.1	82.3
RealNet† (Zhang, Xu, and Zhou 2024)	**99.7**	92.3	95.4	95.6	76.7	91.9
RD (Deng and Li 2022)	98.6	96.5	94.1	**98.9**	90.1	95.7
RD++† (Tien et al. 2023)	98.7	96.1	95.2	98.2	84.4	94.5
GNL (Cao, Zhu, and Pang 2023)	98.0	97.4	97.5	97.8	94.1	97.0
GNL† (Cao, Zhu, and Pang 2023)	97.7	97.2	96.5	97.0	93.7	96.4
FiCo (ours)	98.8	**97.9**	**97.9**	98.5	**95.0**	**97.6**

< AUROC on MVTec >

Method	ID	OOD			Avg.
	P	A	C	S	
Deep SVDD (Ruff et al. 2018)	40.9	53.4	41.2	39.5	43.8
f-AnoGAN (Schlegl et al. 2019)	61.3	50.2	52.4	**63.8**	56.9
KD (Salehi et al. 2021)	88.2	62.9	62.6	51.4	66.3
PatchCore† (Roth et al. 2022)	77.5	57.5	56.5	52.1	60.9
SimpleNet† (Liu et al. 2023)	**91.6**	62.3	54.8	47.5	64.1
RD (Deng and Li 2022)	81.5	61.1	60.3	55.1	64.5
RD++† (Tien et al. 2023)	86.9	61.7	65.2	60.6	68.6
GNL (Cao, Zhu, and Pang 2023)	87.7	65.6	68.0	62.4	70.9
GNL† (Cao, Zhu, and Pang 2023)	87.5	64.8	68.3	58.1	69.7
FiCo (ours)	89.7	**67.6**	**70.9**	62.3	**72.6**

< AUROC on PACS >

Experiment

Result

Method	ID	OOD				Avg.
	Ori	Br	Co	Bl	No	
Deep SVDD (Ruff et al. 2018)	64.6	59.1	55.9	62.1	54.5	59.3
f-AnoGAN (Schlegl et al. 2019)	70.3	54.6	57.2	60.7	51.8	58.9
KD (Salehi et al. 2021)	84.2	75.9	64.4	63.5	56.9	69.0
PatchCore† (Roth et al. 2022)	80.6	72.9	63.0	57.7	55.5	65.9
RD (Deng and Li 2022)	**84.6**	75.9	65.3	**66.7**	58.8	70.3
RD++† (Tien et al. 2023)	80.3	75.9	66.9	60.3	63.3	69.3
GNL (Cao, Zhu, and Pang 2023)	82.3	77.9	66.1	64.0	61.5	70.4
GNL† (Cao, Zhu, and Pang 2023)	79.2	76.9	67.5	63.2	**64.6**	70.3
FiCo (ours)	80.5	**77.8**	**69.2**	63.8	64.4	**71.1**

< AUROC on CIFAR - 10 >

Method	ID	OOD			Avg.
	P	A	C	S	
GNL (Cao, Zhu, and Pang 2023)	87.5	64.8	68.3	58.1	69.7
DiSCo	88.2	64.2	69.0	59.2	70.1
DiSCo + DiIFi	89.5	65.5	70.5	61.6	71.7
FiCo	**89.7**	**67.6**	**70.9**	**62.3**	**72.6**

< Effectiveness of different component >

Thanks :)