

## Memory-efficient Image Patch Distribution Modeling for Industrial Anomaly Detection

Ki-Ju Kim\*, Gye-Young Kim\*\*

\*Student, School of Software, Soongsil University, Seoul, Korea

\*\*Professor, School of Software, Soongsil University, Seoul, Korea

### [Abstract]

Defective products can cause substantial economic losses and even threaten human safety; therefore, early defect detection prior to shipment is crucial. In conventional memory-bank-based approaches that model normal data, however, memory consumption increases sharply with the number of images and their resolution when all normal samples are used for modeling. In real industrial environments, large-scale and high-resolution images are often required for stable inspection, which limits practical deployment. To address this issue, we propose an online cluster-based memory bank. The proposed method processes normal images sequentially and models them in a memory-efficient manner by clustering their features online. Experiments on the Real-IAD dataset show that our method achieves an image-level AUROC of 90.6% while reducing memory usage by more than 90%. Furthermore, enabling high-resolution processing with 512×512 inputs improves the image-level AUROC by about 1.4 percentage points and increases the PRO metric by about 5 percentage points compared to the 224×224 baseline.

▶ **Key words:** Computer vision, Image processing, Industrial anomaly detection, Online clustering, Memory bank

### [요 약]

불량품은 경제적 손실과 인명 피해를 초래할 수 있기 때문에, 제품 출하 이전 단계에서 불량품을 조기에 탐지하는 것은 중요하다. 정상 데이터를 모델링하여 높은 불량 탐지 정확도를 보인 기존 연구에서, 전체 정상 데이터를 대상으로 모델링할 경우 영상 수와 해상도에 비례하여 메모리 사용량이 급격히 증가한다. 실제 산업 환경에서는 안정적인 탐지를 위해 대규모 및 고해상도 영상이 필요하므로, 이러한 한계는 실제 적용에 큰 제약으로 작용한다. 본 논문은 이 문제를 해결하기 위해 온라인 군집 메모리 बैं크 방법을 제안한다. 제안 방법은 정상 영상을 순차적으로 처리하며 특징을 군집화 방식을 통해 메모리 효율적인 방법으로 모델링한다. 실험 결과, 제안한 방법은 Real-IAD 데이터셋에서 약 90.6%의 image-AUROC을 달성하면서 메모리 사용량을 약 90% 이상 절감하였다. 또한 고해상도 영상 처리가 가능해지면서, 512×512 해상도 접근으로 기존 224×224 해상도 대비 약 1.4%의 Image-AUROC 향상 및 5%의 PRO 성능 향상을 보였다.

▶ **주제어:** 컴퓨터 비전, 영상 처리, 제조공정 불량품 탐지, 온라인 군집화, 메모리 बैं크

- First Author: Ki-Ju Kim, Corresponding Author: Gye-Young Kim
- \*Ki-Ju Kim (kimkyju@naver.com), School of Software, Soongsil University
- \*\*Gye-Young Kim (gykim11@ssu.ac.kr), School of Software, Soongsil University
- Received: 2026. 02. 03, Revised: 2026. 04. 13, Accepted: 2026. 04. 15.
- This paper is an extension of the paper presented at the 73rd Winter Conference of the Korean Computer Information Society in 2026 ("Memory-efficient Image Patch Distribution Modeling for Industrial Anomaly Detection").

## I. Introduction

산업 현장에서 발생할 수 있는 불량품으로 인한 경제적 손실과 인명 피해를 방지하기 위해, 제품 출하 이전 단계에서 조기 불량 탐지를 수행하는 것은 매우 중요하다. 전통적으로 사람이 직접 시각적 검사를 통해 불량을 식별해 왔으나, 반복적인 작업 환경에서는 피로 누적과 주관적 판단의 편차로 인해 일정한 수준의 검출 성능을 유지하기 어렵다. 이러한 한계를 보완하기 위해 최근 제조공정 불량품 탐지 분야에서는 컴퓨터 비전 기술을 활용한 영상 분석 기반의 자동 시각 검사 기법이 활발히 연구되고 있다[1][2].

그러나 산업 데이터는 보안 및 접근성 제약으로 인해 공개가 어렵고, 실제 불량 샘플의 수가 제한적이어서 딥러닝 모델 학습에 어려움이 존재한다. 이를 해결하기 위해 일부 연구에서는 GAN이나 Perlin noise를 이용해 정상 영상에 인위적 불량을 합성하는 방안이 제안되었다[3][4][5]. 하지만 이러한 합성 불량은 실제 산업 환경의 결함 특성을 충분히 반영하지 못하기 때문에 실제 현장 적용에는 한계가 있다.

이러한 이유로 정상 데이터만을 활용해 정상 패턴의 분포를 모델링하고, 입력 영상이 이 분포에서 얼마나 벗어나는지를 평가하는 비지도 학습 기반 불량 탐지 방법이 주목받고 있다[6][7][8]. 특히 이러한 접근법은 MVTec AD[9] 공개 데이터셋에서 우수한 성능을 보여 왔다. 이는 MVTec AD가 조명, 배경, 객체 위치 등이 비교적 정제된 환경에서 촬영되어 데이터 변동성이 낮고, 저해상도 영상에서도 정상 패턴의 분포를 비교적 안정적으로 학습할 수 있기 때문이다.

그러나 이러한 특성은 실제 산업 환경과는 거리가 있다. 실제 제조 현장에서는 촬영 각도, 조명, 반사, 카메라 노이즈 변화 등이 매우 크며, 결함 또한 수 mm 단위의 미세 크기로 존재하는 경우가 많다. 하지만 저해상도 영상에서는 이러한 결함이 픽셀 수준에서 소실되어 탐지 자체가 어려워질 수 있다. 예를 들어 10cm 크기의 물체에 2mm 크기의 결함이 존재할 경우, 100×100 해상도로 촬영하면 결함은 약 2픽셀 수준으로만 표현되어 정보가 쉽게 희석되기 때문에 실제 산업 결함을 안정적으로 탐지하기 위해서는 고해상도 영상의 활용이 필수적이며 다양한 촬영 조건을 수용할 수 있는 대규모 데이터셋 사용 또한 필수적이다.

하지만 고해상도 영상을 사용할 경우 문제가 발생한다. PatchCore[8]와 같은 메모리 뱅크 기반 방법은 컨볼루션 연산을 통해 영상으로부터 특징을 추출하기 때문에 불량에 해당하는 픽셀 수가 적을 경우 불량에 해당하는 특징이

희석되어 추론 단계에서 의미 있는 특징 추출 능력이 저하된다. 또한 영상 해상도에 비례해 생성되는 패치 수가 증가하고, 그 결과 저장해야 하는 패치 특징 벡터의 수가 급격히 늘어난다. 이 경우 GPU 및 시스템 메모리를 초과하여 Out of Memory 문제가 발생할 수 있다. 이는 촬영 조건의 변동성, 미세 결함 탐지를 위한 고해상도의 필요성, 그리고 대량의 정상 데이터 요구 등을 만족하지 못하는 핵심 병목으로 작용하여 제조공정 불량 탐지의 일반화 능력을 저하시킨다. 기존 비지도 학습 기반 방법은 이러한 대규모·고해상도 데이터를 처리하는 과정에서 높은 메모리 사용량과 추론 시간 증가라는 한계를 보이며, 이는 실제 현장 적용 가능성을 크게 제한한다.

이에 본 논문에서는 이러한 한계를 극복하기 위해 대규모·고해상도 영상으로부터 안정적으로 분포를 모델링할 수 있는 온라인 클러스터 뱅크 방법을 제안한다. 제안 방법은 전체 정상 데이터 세트를 한 번에 처리하지 않고 입력 영상을 순차적으로 처리하며 군집화를 통해 분포를 점진적으로 업데이트한다. 이를 통해 메모리 사용량을 크게 줄이면서도 정상 패턴 분포를 안정적으로 모델링할 수 있으며, 단일 클래스뿐 아니라 다중 클래스 확장도 용이하다. 대규모 영상 데이터로 이루어진 Real-IAD 데이터셋[10] 실험 결과, 제안 방법은 기존 PatchCore 대비 메모리 사용량을 약 90% 이상 절감하면서도 AUROC 성능을 유지하거나 향상시키며, 대규모·고해상도 환경에서도 안정적으로 동작함을 확인하였다.

## II. Related Works

### 1. Synthetic Defect-Based Anomaly Detection

제조공정의 보안 문제 및 낮은 불량률로 인한 정상-불량 데이터 불균형 문제 등으로 불량 데이터 수집이 어려운 문제를 극복하기 위해, DREAM[3], CutPaste[4], MemSeg[5], CPR[11], Peng et al.[12] 등은 Perlin noise[13]를 통해 자연스러운 모양의 결함을 정상 영상에 임의로 합성하여 지도 학습을 통한 불량 탐지를 수행하였다. 하지만 이는 합성된 불량에 과적합 되어 일반화 능력이 떨어진다는 지적이 있다[14]. 또한 주로 사용되는 방법인 Perlin noise 등을 이용하여 불량을 합성하는 경우 실제 결함의 형태와는 다른 형태의 결함이 합성된다. 해당 예시를 Fig. 1에서 확인할 수 있다.



Fig. 1. Example of a synthesized image

따라서 보다 자연스러운 합성을 위해 GAN[15] 혹은 Diffusion[16] 기반 생성형 모델을 통해 실제와 유사한 불량을 합성하는 방법이 연구되고 있다[14][18]. 실제 결함과 유사한 패턴의 불량 영상을 만들 수 있게 된다면 제조공정 불량품 탐지 분야의 데이터 부족 문제에 크게 기여할 수 있을 것으로 기대된다.

## 2. Unsupervised Industrial Image Anomaly Detection

2.1절의 합성 결함이 실제 결함의 패턴과 다르며 합성 결함에 과적합 되어 일반화 성능이 떨어지는 문제로, 제조공정 불량 탐지 분야는 정상 영상의 특징을 학습하고 입력 영상의 특징과 비교하여 이상을 검출하는 비지도 학습 기반 접근법이 주로 사용된다. 이러한 접근은 인간이 사전에 학습한 정상 패턴에서 벗어난 변화를 감지하는 인지 메커니즘과 유사하여 다양한 형태의 이상에 대해 탐지가 가능하다.

한 예시로는 Auto Encoder[17] 형태의 신경망을 통해 정상 영상 복원을 학습하여 불량 영상을 정상 영상으로 복원하여 입력 불량 영상과 복원된 정상 영상 간 차이를 활용하여 불량을 탐지하는 재구성 기반 방법이 있다[2][20][21]. 그러나 이러한 접근은 미세한 결함을 충분히 보존하지 못하고, 네트워크가 불량 영역까지 함께 복원하는 경향을 보여 일반화 성능이 떨어진다.

또 다른 접근은 사전 학습 모델의 표현력을 활용하기 위해, Teacher-Student 구조를 이용한 지식 증류 기반 이상 탐지 방법이 연구되었다[22][23][24]. 대규모 데이터로 사전 학습된 Teacher 신경망의 정상 영상 특징 추출 능력을 묘사하기 위한 별도의 Student 신경망을 학습한다. 불량 영상이 입력되었을 경우 Teacher-Student 신경망 간 특징 추출 능력에 차이가 발생해 이 차이를 기반으로 불량을 탐지하는 방법이다. 하지만, 이 방법 또한 Student가 Teacher의 특징 추출 능력을 학습하여 특징 추출 능력 차

이가 감소하기 때문에 일반화 성능이 떨어진다는 문제가 지적되었다[25].

이러한 문제를 해결하기 위한 접근은 사전 학습 네트워크가 추출한 특징을 기반으로 정상 영상의 분포를 모델링하는 방법이 있다. 대표적인 예로 SPADE[5], PaDiM[6], PatchCore[7] 등이 있으며 이 모델들은 사전 학습 백본 네트워크를 통해 정상 영상으로부터 특징을 추출하여 입력 영상의 특징과 유사도를 비교한다. SPADE 모델의 경우 각 정상 영상을 하나의 전역 특징 벡터로 표현하여 정상 영상 분포를 모델링하고 입력영상의 특징 벡터와 최근 접 이웃과 유사도를 계산해 이상을 검출한다. PaDiM은 정상 특징맵의 각 위치별 평균과 공분산을 추정한 후, 입력 특징맵과 유사도를 비교하여 이상을 탐지한다. 그러나 이러한 전역 또는 통계 기반 접근은 영상 내 국소적인 불량 정보를 충분히 반영하기 어렵다는 한계를 가진다. 이를 보완하기 위해 PatchCore는 정상 영상으로부터 패치 특징을 추출하여 메모리 बैं크를 구축한 후 입력 패치 특징과 유사한 패치 특징과의 유사도를 비교하여 보다 정확한 불량 검출과 빠른 추론 속도를 동시에 달성하였다. 하지만 전체 패치 특징으로부터 모델링하기 때문에 영상 수에 따라 필요 메모리가 선형적으로 증가하여 메모리 부족 문제가 발생할 수 있으며[19] 메모리 문제로 인한 고해상도 접근 어려움으로 인한 미세 결함 오탐 문제가 지적된다[10]. 또한 패치 수를 단순히 줄이는 방식으로 모델링하기 때문에 유사 패치 간의 구조적 관계를 충분히 활용하지 못한다는 한계가 있다.

## III. Method

### 1. Model Overview

Fig. 2는 제안하는 모델의 구조를 그림으로 나타낸 것이다. 사전 학습 백본 네트워크를 통해 영상의 특징을 추출하며 추출한 패치 특징 벡터를 순차적으로 패치 특징 벡터 집합인 메모리 बैं크에 입력한다. 이때 하이퍼파라미터 임계 거리를 사용하여 입력 시 최근접 샘플과 L2 거리를 계산하고 임계 거리 이내라면 평균 벡터를 갱신하는 방법으로 온라인 군집화하여 정상 패치 분포를 모델링한다. 이 과정을 반복하며 최종 메모리 बैं크를 구축하고 입력 영상의 특징 벡터를 메모리 बैं크에 입력하여 메모리 बैं크 내 최근접 정상 대표 벡터와 L2 거리를 계산하고 이 거리를 불량 점수로 사용한다. 각 내용은 이후 3.2절과 3.3절, 3.4절에서 상세히 설명한다.

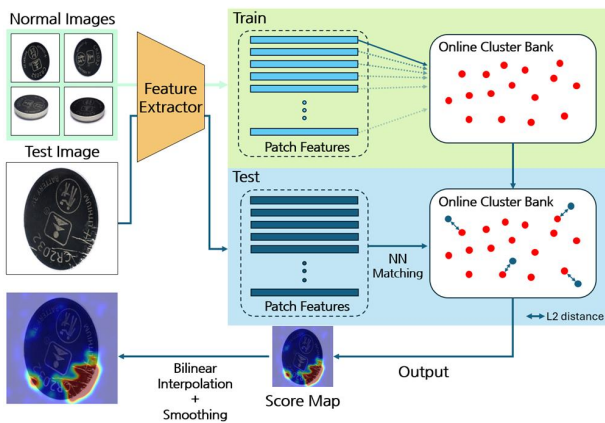


Fig. 2. Architecture Overview

## 2. Feature Vector Construction

산업 불량 탐지 분야에서는 정상 데이터와 불량 데이터 간의 불균형 및 데이터 수 부족으로 인해 모델 학습이 어려운 경우가 많다. 이러한 이유로 많은 연구에서는 ImageNet[25] 데이터로 사전 학습된 CNN[26] 기반 모델 [27][28][29]을 특징 추출기로 사용한다[30][31]. 사전 학습된 ResNet[27] 등의 CNN 모델은 두 가지 측면에서 장점을 갖는다. 첫째, 픽셀 단위 변화에 민감한 컨볼루션 연산을 통해 다양한 객체의 영상으로부터 저수준에서 고수준까지의 풍부한 지역 특징을 효과적으로 추출할 수 있다. 둘째, ViT 기반 모델[32]에 비해 FLOPS가 수 배에서, 많게는 수십 배 낮아 특징 추출 속도가 빠르므로 실시간 추론에 유리하다. 따라서 이러한 CNN 기반 사전 학습 모델을 사용하면 데이터 부족 문제를 완화하면서도 대규모 영상의 신속한 추론이 가능하다.

CNN 모델은 입력 영상에 일정 크기의 커널을 적용하여 커널 내부의 픽셀들로부터 컨볼루션 연산을 수행하고, 그 결과로 해당 영역의 특징 스칼라값을 출력한다. 커널은 영상의 왼쪽에서 오른쪽으로, 위쪽에서 아래쪽으로 슬라이딩 하면서 하나의 커널 당 2차원 특징맵 (h,w)을 생성한다. 여러 개의 커널을 사용할 경우, 각 커널에서 생성된 특징맵들이 3차원 채널 방향으로 결합되어 최종적으로 (h,w,c) 형태의 3차원 특징맵이 된다. 각 좌표 (h,w)에서의 c-차원 벡터는 입력 영상의 동일한 영역에서 여러 개의 커널을 통해 컨볼루션 연산한 결과이므로 해당 위치를 반영하는 지역 특징 벡터이며, 입력 영상의 국소 영역 정보를 표현한다. 산업 불량 검사의 경우, 불량 픽셀은 영상 전체에서 극히 작은 비율을 차지하기 때문에, 전체 영상 단위의 평균적 표현을 사용할 경우 이러한 불량 정보가 쉽게 희석될 수 있다. 따라서 본 논문에서는 (h,w)에서의 특징 벡터를

직접 활용하여, 국소 영역 간의 세밀한 비교를 수행함으로써 미세한 불량도 효과적으로 탐지할 수 있도록 한다.

본 논문은 WideResNet-50[33]을 특징 추출기로 사용한다. 기존 연구인 PatchCore는 ResNet 특징 추출기의 layer2와 layer3의 중간 수준 특징을 결합하고 average pooling을 적용한 특징이 MVTec-AD 벤치마크 데이터셋에서 정상과 불량을 효과적으로 구분할 수 있음을 보였다. MVTec-AD의 경우 불량 영역이 육안으로 확인할 수 있는 크기이지만, Real-IAD 데이터셋의 경우 미세한 불량이 포함되어 있어 넓은 영역에서 비교할 경우 불량에 해당하는 정보가 희석될 수 있다. 이에 본 논문은 PatchCore와 동일하게 WideResNet의 Layer2와 Layer3의 출력 특징맵을 결합한 중간 수준 특징맵을 활용하되, 특징맵에 average pooling을 적용하지 않고 보다 좁은 영역의 지역 특징 벡터를 활용하도록 설계한다. 본 논문은 WideResNet-50의 중간 수준 특징맵으로부터, 입력 해상도 224×224 기준 정상 영상 한 장당 28×28 개의 1,536차원 지역 특징 벡터를 추출한다.

## 3. Memory Bank Compression via Online Clustering

3.2절에서 추출한 특징맵은 정상 영상 한 장당 28×28×c×4 bytes의 메모리를 필요로 한다. 여기서 c는 특징맵의 채널 수를 말한다. 정상 영상의 수가 N개일 경우 총 필요한 메모리는 N×28×28×c×4 bytes가 되며, 정상 영상의 수가 증가함에 따라 메모리 사용량이 선형적으로 증가하여 Out Of Memory(OOM) 문제가 발생한다. PatchCore는 저차원 투영을 통해 메모리 사용량을 줄여 이러한 문제를 예방하고자 하였으나 24GB VRAM의 3090GPU 환경에서 약 5,000장의 영상을 처리할 때 OOM 문제가 발생하였다. 더 낮은 차원으로 투영하면 요구 메모리량을 줄일 수 있지만 투영 차원이 감소할수록 샘플 간 L2 거리 보존성이 저하되어 거리 기반 이상 탐지의 정확도에 부정적인 영향을 미칠 수 있다. 따라서 본 논문은 유사한 패치 간 정보를 압축하여 메모리 문제를 억제할 수 있도록 군집화 알고리즘을 사용한다. 군집화 알고리즘의 대표적인 예로 k-means, ISODATA, DBSCAN 등이 있지만, 전체 특징 벡터로 처리할 경우 기존 방법과 동일한 OOM 문제가 발생할 수 있다. 전체 특징 벡터 접근이 아닌 온라인 k-means를 통해 OOM 문제를 회피하며 군집화할 수 있지만 이 경우 군집 개수 파라미터 k를 지정해야 한다. 하지만 특징 공간에서 군집이 몇 개인지 알 수 없기 때

문에 적합하지 않다.

따라서 본 논문은 Welford's 온라인 알고리즘[35]을 사용하여 메모리 बैं크를 군집화하도록 구성한다. 군집화를 통해 유사한 특징 벡터들을 집약함으로써 메모리 사용량을 크게 절감하고, 동시에 불필요한 비교 연산을 줄여 추론 속도 향상을 기대한다. Welford's online algorithm은 새로운 입력이 들어올 때마다 평균을 갱신하는 방식으로 정의되며 다음 [식 1]로 표현된다:

$$\mu = \mu + \frac{x - \mu}{n} \quad [\text{식 1}]$$

여기서  $\mu$ 는 시점  $n$ 까지의 평균,  $x$ 는 새로운 입력 벡터를 의미한다. 본 논문은 이 공식을 군집 대표 벡터 갱신 과정에 적용한다. 구체적으로, 흡수 반경  $r$ 을 하이퍼 파라미터로 설정하고 각 지역 특징 벡터  $v$ 를 메모리 बैं크  $M$ 에 순차적으로 입력하면서 반경 내에 존재하는 가장 가까운 대표 벡터를 탐색한다. 만약 해당 반경 내에 대표 벡터가 존재하면, 해당 대표 벡터는 새로운 입력 벡터를 흡수하여 평균을 갱신한다. 반대로, 반경 내에 대표 벡터가 존재하지 않으면, 새로운 입력 벡터가 새로운 군집의 대표 벡터로 메모리 बैं크에 등록된다. 반경을  $r$ 이라고 했을 때 이를 수식으로 표현하면 다음 식과 같다:

$$k^* = \arg \min_{1 \leq k \leq K} \|x - \mu_k\|_2 \quad [\text{식 2}]$$

$$d(x, \mu_{k^*}) = \|x - \mu_{k^*}\|_2 \quad [\text{식 3}]$$

$$M = \begin{cases} (M - (\mu_k, n)) \cup (\mu_k + \frac{x - \mu_k}{n_k + 1}, n + 1), & d(x, \mu_k) \leq r \\ M \cup (x, 1), & d(x, \mu_k) > r \end{cases} \quad [\text{식 4}]$$

[식 2]는 최근접 이웃 매칭에 대한 수식이며 [식 3]은 매칭된 최근접 대표 벡터와의 L2 거리 계산식이다. [식 4]는 흡수 반경을 기준으로 매칭된 대표 벡터가 입력 벡터를 흡수하며 평균 벡터로 이동하거나 새로운 대표 벡터의 추가에 대한 식이다. 또한 전체 메모리 बैं크 구축 과정을 의사 코드로 표현하면 [Algorithm 1]과 같다. 이 과정은 입력 데이터가 순차적으로 주어지며 일정한 메모리 한도 내에서 수행이 가능하다. 결과적으로 유사한 특징 벡터들을 효과적으로 집약하여 메모리 사용량을 크게 절감할 수 있도록 한다.

Algorithm 1. Online cluster memory bank

---

```

Input: Training Images  $\{I_i\} \{i = 1 \dots N\}$ 
         Feature extractor  $\psi(\cdot)$ 
         Radius  $r$ 

Output: Memory bank
          $M = \{(\mu_k, n_k)\} \{k = 1 \dots K\}$ 

Algorithm:
 $K \leftarrow 0$ 
 $M \leftarrow \emptyset$ 
for each image  $I$  in  $\{I_i\}$  do
     $F \leftarrow \psi(I)$ 
     $X \leftarrow \text{reshape } F \text{ into } \{x_j\}_{j=1}^N$ 
    for each feature  $x$  in  $X$  do
        if  $K > 0$  then
             $k^* \leftarrow \arg \min_{1 \leq k \leq K} \|x - \mu_k\|_2$ 
             $d^* \leftarrow \|x - \mu_{k^*}\|_2$ 
        else
             $d^* \leftarrow \infty$ 
        end if
        if  $d^* \leq r$  then
             $n_{k^*} \leftarrow n_{k^*} + 1$ 
             $\mu_{k^*} \leftarrow \mu_{k^*} + (x - \mu_{k^*}) / \mu_{k^*}$ 
        else
             $K \leftarrow K + 1$ 
             $\mu_K \leftarrow x$ 
             $n_K \leftarrow 1$ 
             $M \leftarrow M \cup \{(\mu_K, n_K)\}$ 
        end if
    end for
end for

return  $M$ 
    
```

---

Fig. 3은 2차원 예제 데이터에 본 알고리즘을 적용한 결과를 시각화한 것이다. 파란 점이 입력 벡터이며 빨간 점이 알고리즘에 의해 계산된 대표 벡터들이다. 첫 번째 이미지가 원본 데이터이며 2, 3, 4번째 이미지는 점점 커지는 흡수 반경 파라미터 설정에 의한 결과를 시각화한 것이다. 푸른색 점이 한 데이터 샘플이며 붉은색 점이 [Algorithm 1]에 의해 평균화된 지역 대표 벡터이다. 흡수 반경의 크기에 따라 군집의 개수와 대표 벡터의 분포가 달라지며, 각 대표 벡터는 원본 데이터의 일정 지역의 평균 정보를 잘 반영하고 있음을 확인할 수 있다.

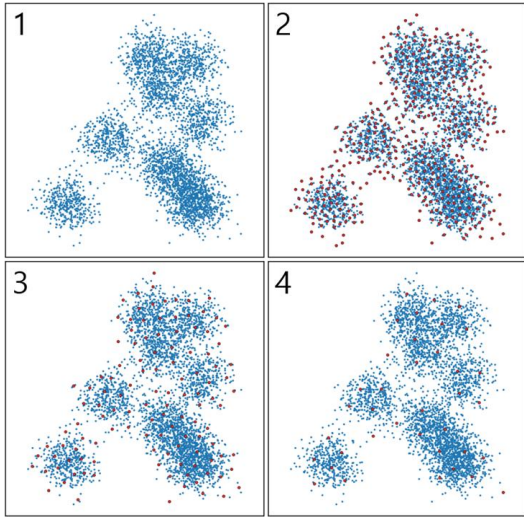


Fig. 3. Example Results of [Algorithm 1]

Real-IAD 전체 훈련 데이터셋 약 36,000장의 영상 데이터로부터 PatchCore는 최소 약 175GB 이상의 VRAM을 필요로 하지만, 제안 방법으로 메모리 बैं크를 구축할 시 약 3GB 정도의 적은 VRAM으로 메모리 बैं크를 구축할 수 있다. 데이터셋의 크기 증가에 따라 PatchCore는 요구 메모리량이 선형적으로 증가함에 따라 OOM 문제 발생 가능성이 높아진다. 반면에, 제안하는 방법은 선형적인 요구 메모리 증가와는 다르게 군집화를 통해 요구 메모리량이 수렴하게 되어 대규모 데이터셋에 적용이 용이하다. 또한 클래스별로 메모리 बैं크를 구축할 경우 단순한 클래스는 적은 수의 대표 벡터로, 객체별 물체 내부의 픽셀 변화가 큰 경우, 보다 많은 대표 벡터로 메모리 बैं크가 구축되기 때문에 일반적인 방법과는 다르게 도메인 맞춤형 메모리 बैं크를 구축할 수 있다는 장점을 갖는다.

#### 4. Inference

3.2절에서 구축한 메모리 बैं크 각 대표 벡터는 흡수 반경 내 정상 지역 특징 벡터들의 평균 정보를 내포하고 있다. 이를 활용해 추론 단계에서는 입력 영상으로부터 특징 맵을 추출한 후, 각 지역 특징 벡터  $x_j$ 를 메모리 बैं크에 입력하여 최근접 이웃을 탐색한다. 각 지역 특징 벡터별 가장 유사한 대표 벡터와 매칭하고 두 벡터 간 L2 거리  $d$ 를 계산하여 유사도를 측정한다. 이는 다음의 [식 5]와 [식 6]으로 표현된다:

$$k^* = \arg \min_{1 \leq k \leq K} \|x_j - \mu_k\|_2 \quad [\text{식 5}]$$

$$d_j = \|x_j - \mu_{k^*}\|_2 \quad [\text{식 6}]$$

입력 벡터가 정상 영역에 해당할 경우, 매칭된 대표 벡터와의 L2 거리는 작게 나타나며, 반대로 불량 영역의 벡터는 대표 벡터로부터 멀리 위치하게 되어 큰 L2 거리를 갖는다. 따라서 각 지역 특징 벡터에 대한 거리는 지역 단위 불량 가능성으로 해석할 수 있다.

PaDiM과 유사하게, 군집화를 통한 대표 벡터에서 매칭되는 정상 특징 벡터간 공분산 정보를 활용해 마할라노비스 거리를 사용할 수 있지만 1,536차원의 고차원 공간에서는 안정적인 추론을 위해 매우 많은 샘플을 필요로 한다. 실험적으로, 마할라노비스 거리 사용 시 L2 거리보다 정확도가 떨어졌으므로 L2 거리를 사용해 추론한다.

한 지역 특징 벡터라도 불량에 해당하는 경우 해당 영상을 불량으로 판정한다. 이를 위해 모든 지역 특징 벡터와 매칭된 대표 벡터 간의 거리 집합  $J$  중 최대 거리를 입력 영상의 불량 점수로 정의한다. 이는 [식 7]로 표현된다:

$$s(I) = \max_{1 \leq j \leq J} d_j \quad [\text{식 7}]$$

또한, 각 지역 특징 벡터의 거리를 기반으로 2차원 불량 가능성 히트맵을 생성하여 입력 영상의 해상도에 맞게 양 선형 보간을 통한 업샘플링 후 부드러운 히트맵을 위해 가우시안 필터를 적용해 히트맵을 스무딩한다. 이를 통해 픽셀 단위 탐지 정확도를 측정하며 시각화를 통해 불량 발생 위치를 직관적으로 확인할 수 있다.

## IV. Experiments

### 1. Experimental Setup

#### 1.1 Dataset

다양한 각도에서 촬영된 대규모 산업 데이터에 대한 실험을 위해 Real-IAD 데이터셋을 사용하였다. Real-IAD는 총 30개의 클래스로 구성되어 있으며 36,465 장의 학습 데이터, 114,585장의 테스트 데이터로 이루어져 있으며 각 클래스는 약 1,200장의 학습 데이터와 3,800장의 테스트 데이터로 이루어져 있다. 본 논문은 클래스별로 각각의 메모리 बैं크를 구축하여 성능을 측정하였다. 또한 전체 클래스를 대상으로 실험하여 모든 데이터셋을 대상으로 메모리 बैं크를 구축하고 그 안정성을 검증한다.

#### 1.2 Evaluation Metrics

제조공정 불량 탐지 분야에서 널리 사용되는 AUROC와 PRO를 평가지표로 사용한다. 영상 단위에서는 [식 7]의

각 영상 최대 거리  $s(I)$ 를 사용하여 ground truth 라벨과의 AUROC를 계산하고, 이를 영상 단위 이상 탐지 정확도로 정의한다. 영상 단위 AUROC는 불량 영상 검출 정확도이며 마스크의 불량 영역마다 겹치는 비율을 나타내는 픽셀 단위 Per-Region Overlap (PRO)를 통해 segmentation 정확도를 분석한다. 이러한 평가지표는 영상 기반 불량 탐지 벤치마크에서 공통적으로 사용되는 평가지표이며 다양한 연구들이 핵심 지표로 사용하고 있다.

### 1.3 Experimental Environment

본 논문은 단일 NVIDIA RTX 3090 GPU 환경에서 실험하였으며 Pytorch 프레임워크를 사용하였다. 이 외의 세부 사항은 Table 1에서 확인할 수 있다.

Table 1. Experimental Environment

CPU	Intel(R) Core(TM) i9-10900K CPU@3.70Ghz
GPU	NVIDIA GeForce RTX 3090 24GB
RAM	64 GB
OS	Ubuntu 22.04.2 LTS
Framework	Pytorch

### 1.4 Parameter Settings

흡수 반경  $r$  파라미터 설정을 위해 정상 영상의 지역 특징 벡터 간 1-NN과의 L2 거리 분포를 분석하였다. 클래스별 백분위수  $p_{99}$ 를 계산하였으며, 약 20~30 범위 내에 최근접 이웃이 존재함을 확인하였다. 따라서 이 범위 내의 값을 기준으로 메모리 बैं크를 구축하였다.

이 하이퍼 파라미터는 높을수록 메모리 बैं크의 크기가 작아져 메모리 및 추론 속도 측면에서 장점이 있지만 정확도와 trade-off 관계이다.

## 2. Experimental Results

$r$ 을 26으로 설정했을 때, 총 약 100만 개의 지역 특징 벡터가 객체별 내부 픽셀 변화가 작은 단순한 클래스의 경우 약 1천 개의 대표 벡터로, 내부 픽셀 변화가 많은 복잡한 클래스의 경우 약 2만 개의 대표 벡터로 압축되었다. 이는 PatchCore[7]가 메모리 बैं크 구축에 최소 약 6GB 이상의 VRAM을 필요로 하지만 제안하는 방법으로는 약 0.12GB VRAM 이내로 메모리 बैं크를 구축할 수 있음을 의미한다.

Table 2에서 제안 방법의 Real-IAD 데이터셋 각각의 클래스별 실험 결과를 확인할 수 있다. 각각의 클래스별로 실험 후 결과를 평균 내었으며 데이터는 256×256 해상도로 양선형 보간 후 중앙을 기준으로 224×224 크롭하였다.

추론 속도 관점에서, PatchCore 모델은 약 50FPS 속도를 보였으며 제안 모델은 평균 약 350FPS의 추론 속도를 보이며 실시간 불량 탐지 적용 가능성을 확인하였다.

Table 2. Average Results on Real-IAD (each class)

Model	Image AUROC	PRO
PatchCore[8]	90.4	92.5
PaDim[7]	86.6	90.0
SimpleNet[36]	91.7	88.9
DeSTSeg[37]	89.2	90.3
RD[24]	87.6	94.4
UniAD[38]	82.7	86.0
Proposed	90.6	92.0

Table 3은 224×224 해상도 전체 클래스에 대한 제안 방법 각 파라미터별 실험 결과이다. 전체 36,465 장의 정상 데이터를 모두 사용하여 메모리 बैं크를 구축하였으며, PatchCore 방법이 전체 데이터로부터 메모리 बैं크 구축에 약 175GB의 VRAM을 필요로 하는 것에 비해 제안 방법은  $r=26$  기준 약 14만 개의 대표 벡터로 압축되어 0.9GB 정도의 VRAM만을 필요로 하였으며 메모리 बैं크 기반 방법의 다중 클래스 불량 탐지로의 확장 가능성을 확인하였다.

Table 3. Results on Real-IAD (all classes)

Parameter	Image AUROC	PRO	FPS
$r=25$	90.9	92.3	33.9
$r=26$	90.8	92.1	51.4
$r=27$	90.6	91.8	69.8
$r=28$	90.2	91.5	72.1
$r=29$	89.9	91.2	118.3
$r=30$	89.5	90.8	119.5
$r=32$	88.5	89.8	180.6
$r=35$	86.3	87.9	275.0
$r=40$	80.3	83.3	377.8

또한 Real-IAD는 미세 결함 탐지를 위해 고해상도 접근이 필요함을 지적하였다. 하지만 해상도가 커짐에 따라 요구 메모리가 선형적으로 증가하여 기존 방법은 고해상도 접근이 어렵다. 예로 PatchCore 등 기존 연구의 방법을 512×512 해상도로 접근할 경우 기존 224×224 해상도 접근보다 약 5.2배 많은 메모리를 요구한다. 제안 방법은 온라인 군집화를 통해 메모리 효율적인 메모리 बैं크를 구축하면서 512×512 해상도 접근이 가능하였으며 Table 4는  $r=28$ 일 때 512×512 해상도의 실험 결과와  $r=26$ 일 때 224×224 해상도의 실험 결과 비교이다. 고해상도 접근으로 정확도가 향상되었으며 512×512 해상도 접근으로도 평균 약 48FPS 추론 속도를 보였다. Fig. 4는 제안 방법

의 추론 결과 시각화이다.

Table 4. Experimental Results Comparison by Resolution (each class)

Resolution	Image AUROC	PRO
224×224	90.6	92.0
512×512	92.0	97.0

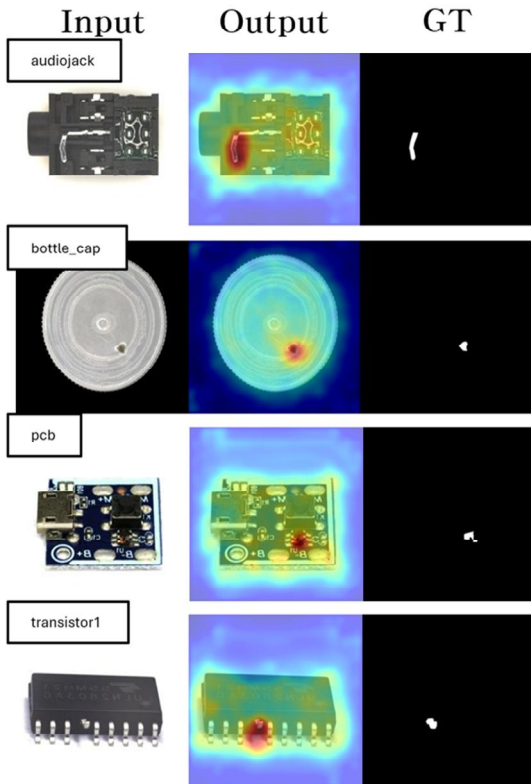


Fig. 4. Visualization of Results

대표 벡터로 압축하여 0.03GB만의 메모리를 필요로 한다. 또한 제안 방법은 해상도 증가에 따라 요구 메모리가 선형적으로 증가하지 않기 때문에 더 큰 해상도 처리가 가능하여 보다 정확한 미세 결함 탐지가 가능하며 이는 실험 결과 분석을 통해 확인할 수 있다.

하지만 CNN 기반 방법은 영상의 지역적인 픽셀 변화에 민감하게 반응하기 때문에 배경에 작은 먼지 등이 있다면 그 영역이 불량으로 탐지되어 실제 불량 탐지의 정확도를 저해한다. Fig. 5는 배경으로 인한 오탐 발생 사례 시각화이다. 입력 영상의 붉은색 박스 배경 영역의 노이즈로 인해 오탐이 발생한 것을 확인할 수 있다. 기존 메모리 뱅크 방법의 단점인 배경으로 인한 오탐은 향후 연구를 통해 해결해야 할 과제로 남아 있다. 또한, 1-NN간 L2 거리를 기반으로 의미 있는 정확도 향상을 달성하였지만, 고차원 벡터 연산이기에 정상인 경우도 큰 L2 거리를 가지고 있기에 분별력이 저하되어 오탐이 발생할 가능성이 있다. 각 지역 대표 벡터별 통계적인 정보를 활용하여 이를 기반으로 정상-불량 간 구분을 보다 정확하게 한다면 보다 높은 정확도를 달성할 수 있을 것으로 기대된다.

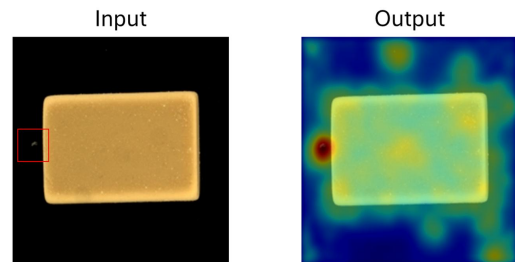


Fig. 5. Example of Background-induced False Positive

### V. Conclusions

본 논문은 기존 메모리 뱅크 방법에 온라인 군집화 방법을 적용하여 메모리 부족 문제를 해결하고, 대규모 영상 데이터를 활용해 불량 탐지의 일반화 성능을 높일 수 있는 온라인 클러스터 뱅크 모델을 제안하였다. 또한 대규모 영상 데이터 활용 능력을 기반으로 기존 메모리 뱅크가 가지는 추가 학습의 불필요, 빠른 추론 속도에 대한 장점을 유지하면서 단일 클래스, 저해상도 접근만을 사용하던 메모리 뱅크 기반 방법을 다중 클래스, 고해상도 접근으로 확장하였다. Table 4의 실험에서, 512×512 해상도 접근 시 PatchCore의 방법으로는 클래스별로 약 512만 개의 특징 벡터를 사용하여 31.4GB의 메모리를 필요로 하는 반면, 제안 방법은 약 512만 개의 특징 벡터를 최대 약 5천 개의

### ACKNOWLEDGEMENT

This work was supported by the Industrial Strategic Technology Development Program-Electronic Components Industry Technology Development Program (IT Convergence in Key Industries))(No. RS-2025-02317274, Development and Validation of a Laboratory-Based Unit Quality Inspection System for Enhancing Tire Manufacturing Quality and Driving Stability, and an All-in-One Integrated Inspection System for Production Quality Management in Manufacturing Sites) funded By the Ministry of Trade Industry & Energy(MOTIE, Korea)

## REFERENCES

- [1] Tao, X., Gong, X., Zhang, X., Yan, S., and Adak, C., “Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, Article 5018021, pp. 1–21, Aug 2022. DOI: 10.1109/TIM.2022.3196436
- [2] J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin, “Deep industrial image anomaly detection: A survey,” *Mach. Intell. Res.*, vol. 21, no. 1, pp. 104–135, Jan 2024. DOI: 10.1007/s11633-023-1459-z
- [3] V. Zavrtnik, M. Kristan, and D. Skočaj, “DRAEM: A discriminatively trained reconstruction embedding for surface anomaly detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 8310–8319, Oct 2021. DOI: 10.1109/ICCV48922.2021.00822
- [4] E. Li, Z. Yang, W. Deng, and J. Hu, “CutPaste: Self-supervised learning for anomaly detection and localization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9664–9673, Jun 2021. DOI: 10.1109/CVPR46437.2021.00954
- [5] M. Yang, P. Wu, J. Liu, and H. Feng, “MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities,” *Eng. Appl. Artif. Intell.*, vol. 120, Art. no. 105885, Mar 2023. DOI: 10.1016/j.engappai.2023.105835
- [6] Niv Cohen and Yedid Hoshen, “Sub-image anomaly detection with deep pyramid correspondences,” *CoRR*, May 2020. DOI: 10.48550/arXiv.2005.02357
- [7] Thomas Defard, Aleksandr Setkov, Angélique Loesch, and Romaric Audigier, “Padim: A patch distribution modeling framework for anomaly detection and localization,” *Pattern Recognition. ICPR International Workshops and Challenges*, pp. 475–489, 2021. DOI: 10.1007/978-3-030-68799-1\_35
- [8] K. Roth, L. Pemula, J. C. Ntumba, L. Ruff, B. Scholkopf, T. Brox, and M. Maire, “Towards total recall in industrial anomaly detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 14318–14328, Jun 2022. DOI: 10.1109/CVPR52688.2022.01392
- [9] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “MVTecAD – A comprehensive real-world dataset for unsupervised anomaly detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9592–9600, Jun 2019. DOI: 10.1109/CVPR.2019.00982
- [10] C. Wang, W. Zhu, B.-B. Gao, Z. Gan, J. Zhang, Z. Gu, S. Qian, M. Chen, and L. Ma, “Real-IAD: A real-world multi-viewdataset for benchmarking versatile industrial anomaly detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 22883–22892, Jun 2024. DOI: 10.1109/CVPR52733.2024.02159
- [11] H. Li, J. Hu, B. Li, H. Chen, Y. Zheng, and C. Shen, “Target Before Shooting: Accurate Anomaly Detection and Localization Under One Millisecond via Cascade Patch Retrieval,” *IEEE Trans. Image Process.*, vol. 33, pp. 5606–5621, Sep 2024. DOI: 10.1109/TIP.2024.3448263
- [12] T. Peng, Y. Zheng, L. Zhao, and E. Zheng, “Industrial product surface anomaly detection with realistic synthetic anomalies based on defect map prediction,” *Sensors*, vol. 24, no. 1, 264, Jan 2024. DOI: 10.3390/s24010264
- [13] Perlin, K., “An Image Synthesizer,” *Computer Graphics (Proc.SIGGRAPH)*, vol. 19, no. 3, pp. 287–296, Jul 1985. DOI: 10.1145/325165.325247
- [14] X. Zhang, M. Xu, and X. Zhou, “RealNet: A feature selection network with realistic synthetic anomaly for anomaly detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 16699–16709, Jun 2024. DOI: 10.1109/CVPR52733.2024.01580
- [15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., “Generative Adversarial Nets,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, pp. 2672–2680, 2014.
- [16] Ho, J., Jain, A., and Abbeel, P., “Denoising Diffusion Probabilistic Models,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6840–6851, Dec 2020.
- [17] Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J., “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction,” in *Proc. International Conference on Artificial Neural Networks (ICANN)*, pp. 52–59, 2011. DOI: 10.1007/978-3-642-21735-7\_7
- [18] Sharma, A. K., Gupta, R., and Wang, Y., “Generative Defect Synthesis for Enhancing Industrial Anomaly Detection,” *Proc. CVPR Workshop on Generative Vision for Industrial Anomaly Detection*, Jun 2025.
- [19] W. Zhu, C. Wang, B.-B. Gao, J. Zhang, G. Jiang, J. Hu, Z. Gan, L. Wang, Z. Zhou, L. Cheng, Y. Pan, B. Peng, M. Chi, and L. Ma, “Real-IAD Variety: Pushing industrial anomaly detection dataset to a modern era,” *arXiv preprint*, Nov 2025. DOI: 10.48550/arXiv.2511.00540
- [20] V. Zavrtnik, M. Kristan, and D. Skočaj, “Reconstruction by inpainting for visual anomaly detection,” *Pattern Recognit.*, vol. 112, Art. no. 107706, Apr 2021. DOI: 10.1016/j.patcog.2020.107706
- [21] Z. Fang, X. Wang, H. Li, J. Liu, Q. Hu, and J. Xiao, “FastRecon: Few-shot industrial anomaly detection via fast feature reconstruction,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 17435–17444, Oct 2023. DOI: 10.1109/ICCV51070.2023.01603
- [22] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “Uninformed Students: Student-Teacher anomaly detection with discriminative latent embeddings,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4183–4192, Jun 2020. DOI: 10.1109/CVPR42600.2020.00424
- [23] G. Wang, S. Han, E. Ding, and D. Huang, “Student-Teacher Feature Pyramid Matching for anomaly detection,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, pp. 1–15, 2021.

- [24] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 9737-9746, Jun 2022. DOI: 10.1109/CVPR52688.2022.00951
- [25] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Asymmetric Student-Teacher Networks for Industrial Anomaly Detection," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), pp. 2592-2602, Jan. 2023. DOI: 10.1109/WACV56688.2023.00262.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 248-255, Jun 2009. DOI: 10.1109/CVPR.2009.5206848
- [27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov 1998. DOI: 10.1109/5.726791
- [28] He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, Jun 2016. DOI: 10.1109/CVPR.2016.90
- [29] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q., "Densely Connected Convolutional Networks," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708, Jul 2017. DOI: 10.1109/CVPR.2017.243
- [30] Tan, M. and Le, Q., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Proc. International Conference on Machine Learning (ICML), vol. 97 of Proceedings of Machine Learning Research, pp. 6105-6114, 2019.
- [31] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-Time unsupervised anomaly detection with localization via conditional normalizing flows," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), pp. 1819-1828, Jan 2022. DOI: 10.1109/WACV51458.2022.00188
- [32] Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., and Wu, L., "FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows," arXiv preprint, Nov. 2021. DOI: 10.48550/arXiv.2111.07677
- [33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Represent. (ICLR), pp. 1-21, May 2021.
- [34] S. Zagoruyko and N. Komodakis, "Wide residual networks," in Proc. Brit. Mach. Vis. Conf. (BMVC), pp. 1-12, Sep 2016. DOI: 10.5244/C.30.87
- [35] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," Technometrics, vol. 4, no. 3, pp. 419-420, Aug 1962. DOI: 10.2307/1266577
- [36] A. Batzner, L. Heckler, and R. König, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 20402-20411, Jun 2023. DOI: 10.1109/CVPR52729.2023.01958.
- [37] H. Deng, X. Li, and Y. Liu, "DestSeg: Segmentation-Guided Denoising Student-Teacher for Anomaly Detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 3914-3923, Jun 2023. DOI: 10.1109/CVPR52729.2023.00379.
- [38] Z. You, L. Cui, Y. Shen, K. Yang, X. Lu, Y. Zheng, and X. Le, "A Unified Model for Multi-Class Anomaly Detection," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 35, pp. 4571-4584, 2022.

## Authors



Ki-Ju Kim received the B.S. degree in multimedia engineering from Hannam University, Korea, in 2023, and the M.S. degree in Department of Software from Soongsil University, Korea, in 2026.

His research interests include computer vision, image processing, and image anomaly detection.



Gye-Young Kim (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Soongsil University, Seoul, South Korea, in 1990, 1992, and 1996, respectively, with a focus on

computer vision. Prof. Kim worked with the Electronics and Telecommunications Research Institute (ETRI) and Korea Electric Power Corporation (KEPCO). He was a Visiting Professor with Indian Institute of Technology Bombay (IITB), in 2007, and a Visiting Scholar with the University of California (UCI), Irvine, from 2014 to 2015. He is currently a Professor with the School of Software, Soongsil University. His research interests include computer vision, machine learning, ultra-high speed camera recognition, and 3D vision.