Dinomaly

(The Less Is More Philosophy in Multi-Class Unsupervised Anomaly Detection)

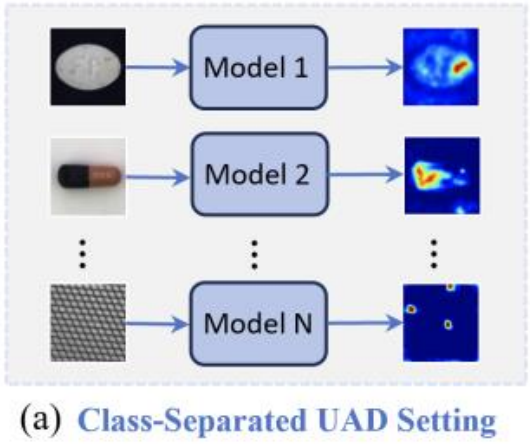
김동원

목차

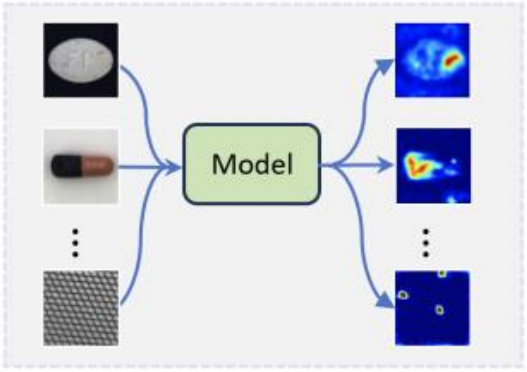
Dinomaly 개요, identity mapping

Reconstruction based anomaly detection 모델을 이용한 unified unsupervised anomaly detection 모델을 개선한 논문

기존의 class별로 다른 모델을 구축한 방법과의 성능차이가 존재하였는데, 이 차이를 간단한 구조를 추가하여 해소하였다.

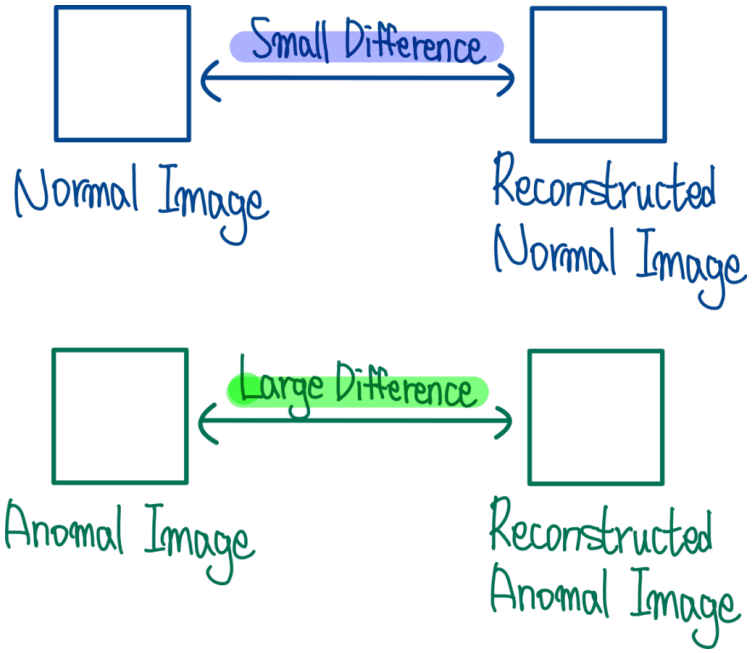


(a) Class-Separated UAD Setting



(b) Model-Unified Multi-Class UAD Setting

Reconstruction based anomaly detection



정상 이미지를 압축하고 재구성하는 과정을 학습하였다.

정상 이미지를 재구성하였을 경우 원본과 비슷할 것이고, 이상 이미지는 원본과의 차이가 많이 존재할 것이라는 바탕으로 이상탐지를 수행함

이 방법은 identity mapping, identical shortcuts문제가 발생

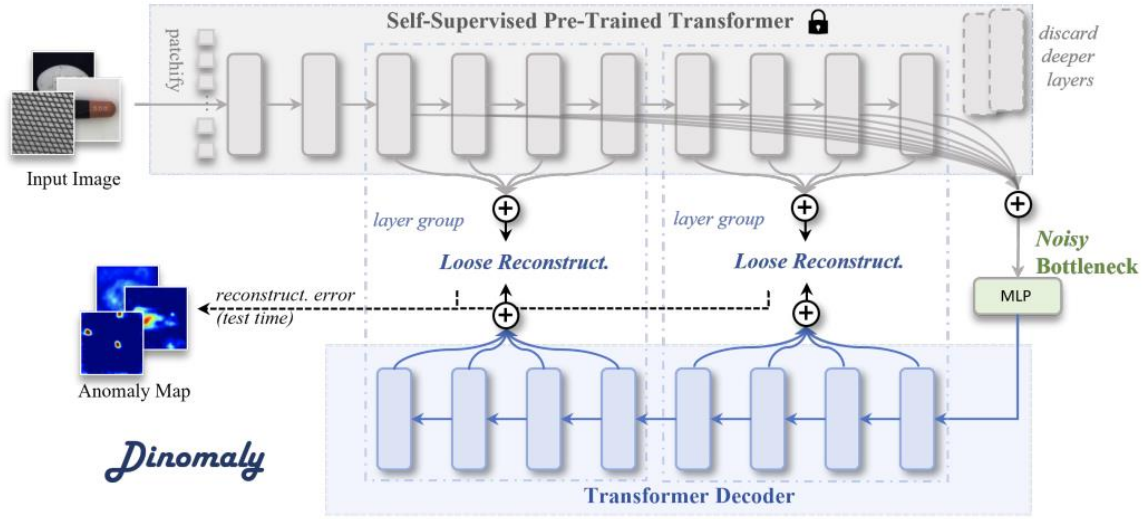
이상 이미지도 복원을 잘해내는 문제를 의미

정상데이터의 다양성을 학습하며 이상 이미지의 unseen pattern에 대해서도 일반화 시켜서 발생하였다.

Dinomaly 구조

Identity mapping문제를 over-generalization 문제로 정의 하여 decoder의 학습을 어렵게 하기위한 여러 방법을 추가하였다.

Noisy Bottleneck, Linear Attention, Loose, Constraint, Loose Loss의 방법을 추가하였다.



Frame work encoder의 각각 다른 scale의 feature map, 8 layer의 값을 모으는 MLP구조이다.

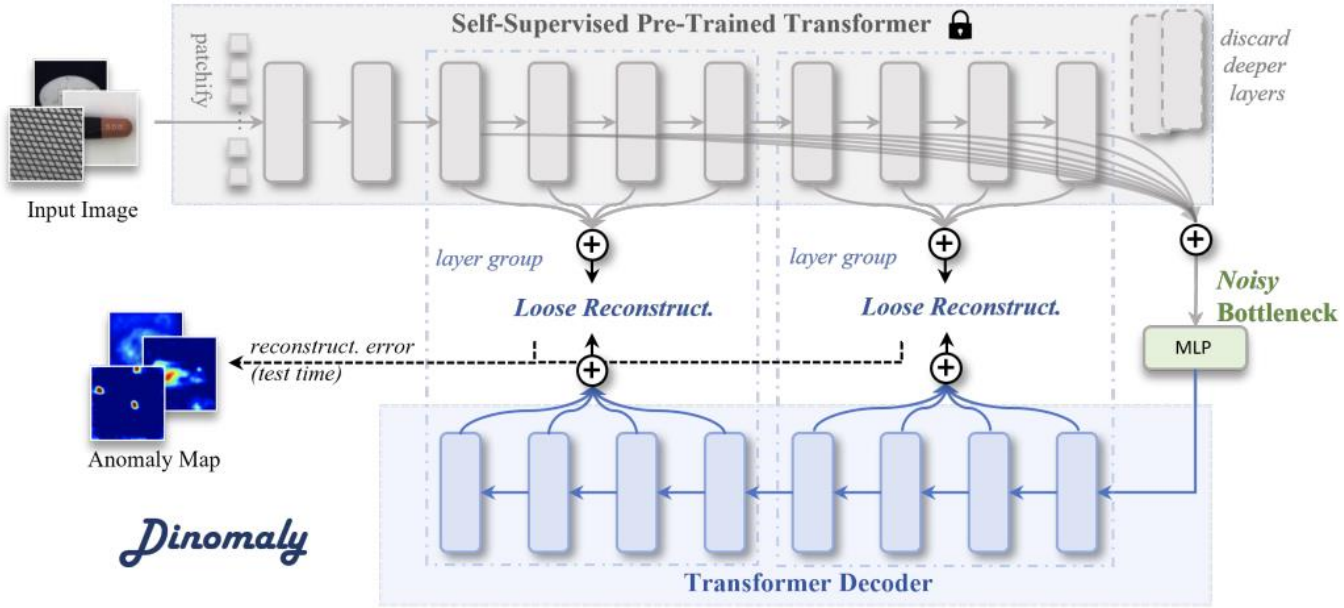
Decoder는 train시에는 mlp의 값을 통해 encode의 8 layer의 feature map을 코사인 유사도가 커지도록 재구성하는 것을 학습하고, 추론시에는 재구성하였을 때 정상부분은 잘 재구성하지만 이상 부분에 대해서는 재구성이 잘 안 되도록하는 것이다.

Dinomaly 구조

Noisy bottleneck: dropout을 학습에 잡음을 추가

Unified class 모델을 구성하므로 이미지의 패턴이나 정상적인 패턴도 다양성을 가지면서 이상데이터의 unseen pattern에 대해서도 일반화 시키는 문제가 발생하는데, 이를 해결하기 위한 방법으로 제안되었다.

복원문제를 복구문제로 전환 시키는 방법으로 해결하게 되었는데, encoder의 특징에 추가적인 잡음을 넣어 feature map을 재구성하는 과정에서 잡음에 대해서도 denoising 과정이 추가로 동반되도록 설계하였다.



Dinomaly 구조

Linear Attention: softmax대신 linear한 activation을 사용하므로 attention이 집중되지 않도록

Linear attention 을 통해 모든 부분의 정보가 layer가 깊어져도 오래 존재하며, 보지 못한 패턴에 대해서는 복원가능성을 줄이고, 연산복잡도는 줄어들게 되는 장점이 있다.

concerning the number of tokens [26]. By substituting Softmax operation with a simple activation function $\phi(\cdot)$ (usually $\phi(x) = \text{elu}(x) + 1$), we can change the computation order from $(\mathbf{QK}^T)\mathbf{V}$ to $\mathbf{Q}(\mathbf{K}^T\mathbf{V})$. Formally, Linear Attention (LA) is given as:

$$\text{LA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\phi(\mathbf{Q})\phi(\mathbf{K}^T))\mathbf{V} = \phi(\mathbf{Q})(\phi(\mathbf{K}^T)\mathbf{V}) , \tag{3}$$

where the computation complexity is reduced to $\mathcal{O}(Nd^2)$ from $\mathcal{O}(N^2d)$. The trade-off between complexity and expressiveness is a dilemma. Previous studies [15, 48] at-

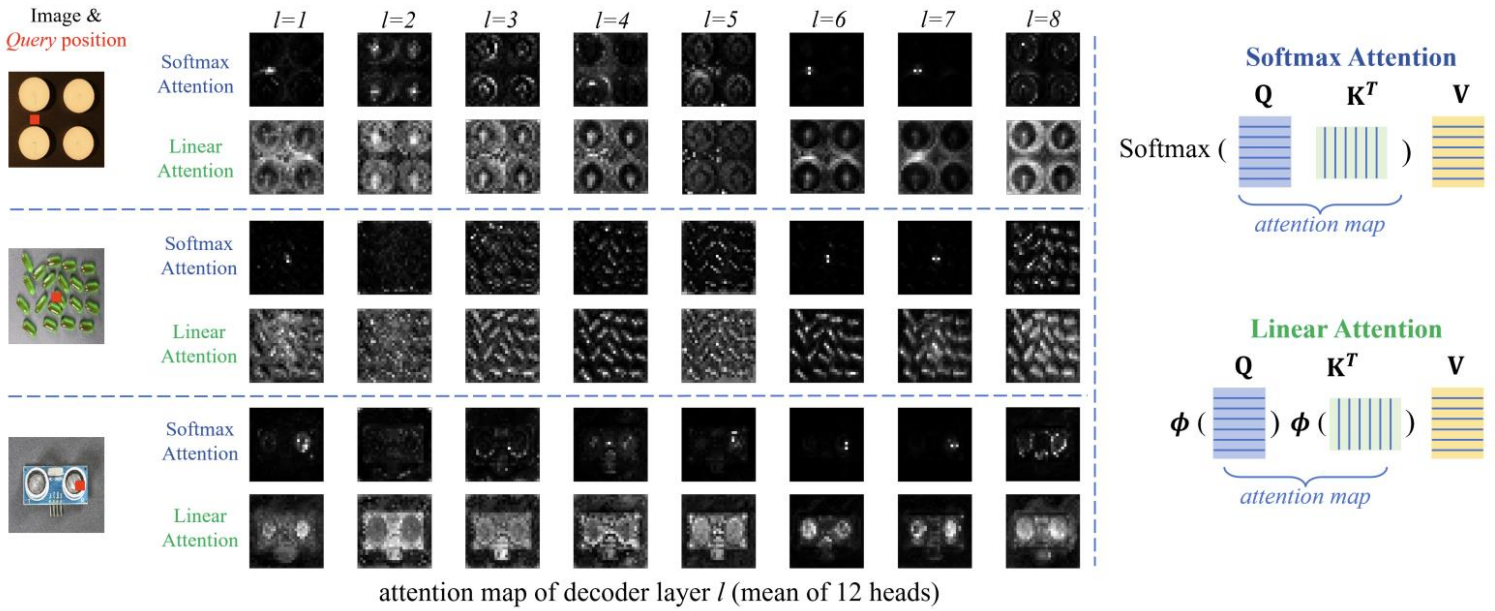
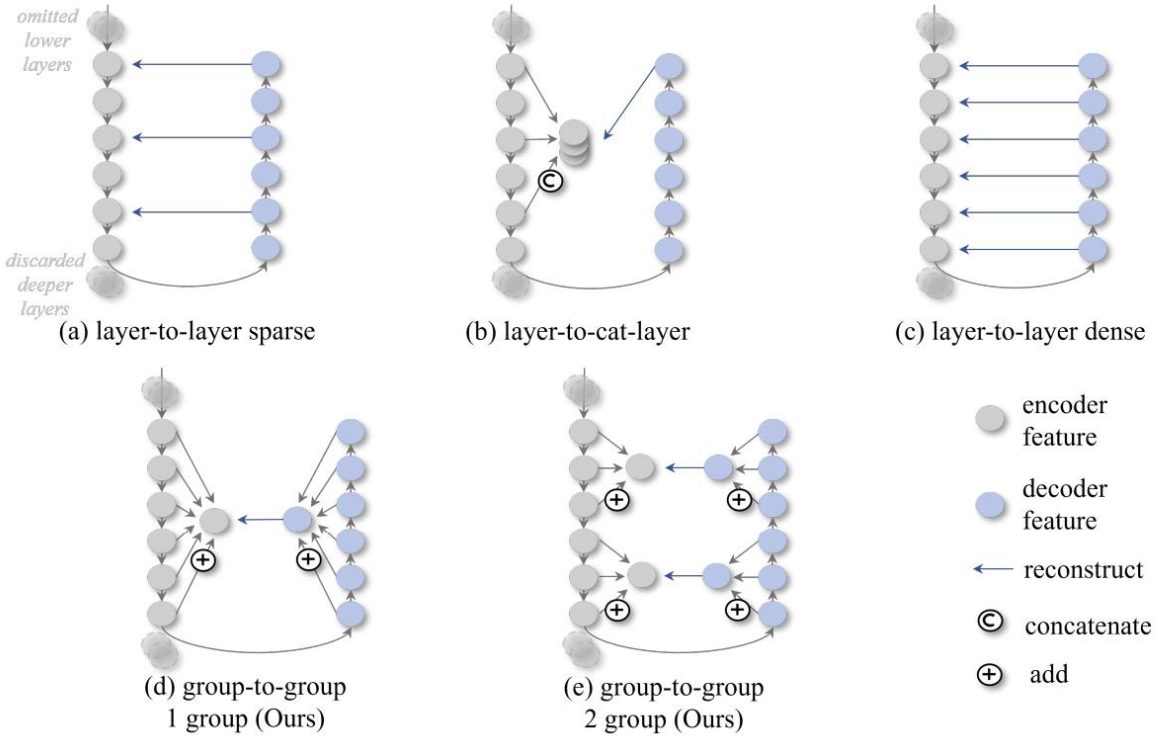


Figure 3. The decoder attention map (min-max to 0-1 for visualization) of Dinomaly with vanilla Softmax Attention *vs.* Linear Attention.

Dinomaly 구조

Loose constraint: encoder를 decode가 overfit하게 학습할 경우 identity mapping이 발생한다고 생각하여 encoder의 정보를 그룹으로 합쳐서 decoder에게 제공하는 방법



Decoder의 자유도를 높여주므로 학습의 복잡성이 증가하고 보지 못한 패턴에 대해서 encode와 다르게 반응할 것이고 이것이 identity mapping을 해결할 것이라고 생각함

Dinomaly 구조

Loose Loss: 과도한 학습을 방지하여 identity mapping을 줄이도록 한 방법.

$$\mathcal{L}_{global-hm} = \mathcal{D}_{cos}(\mathcal{F}(f_E), \mathcal{F}(\hat{f_D})), \quad (4)$$

$$\hat{f_D}(h,w) = \begin{cases} sg(f_D(h,w))_{0.1}, & \text{if } \mathcal{D}_{cos}(f_D, f_E) < k\%_{batch} \\ f_D(h,w), & \text{else} \end{cases} \quad (5)$$

$$\mathcal{D}_{cos}(a,b) = 1 - \frac{a^T \cdot b}{\|a\| \ \|b\|}, \quad (6)$$

where $\mathcal{D}_{cos}$ denotes cosine distance, $\mathcal{F}(\cdot)$ denotes flatten

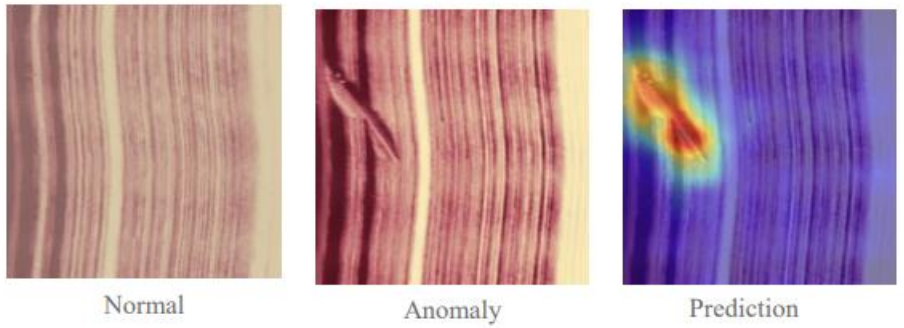
Feature point (h,w)에 대해 k%이하의 cosine distance를 가지는 경우 가중치를 0.1배 감소시킨다.

Dinomaly 실험

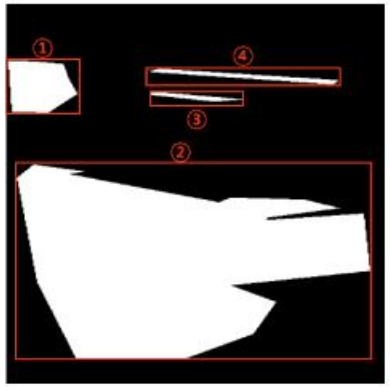
Image level, pixel level에서 AUROC, AP, F1-max, AUPRO metric을 사용하였다.

AUPRO

- AUPRO
  - GT mask의 모든 anomalous region을 bounding box로 구분하고, 각각의 bounding box에서 pixel AUROC를 계산하고 이들의 평균을 계산



Anomaly Segmentation



AUPRO

Dinomaly 실험

Multi class UAD모델과 class separated UAD에서 모두 가장 좋은 성능을 보였다.

Dateset	Method	Image-level			Pixel-level			
		AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
MVTec-AD [3]	RD4AD [10]	94.6	96.5	95.2	96.1	48.6	53.8	91.1
	SimpleNet [34]	95.3	98.4	95.8	96.9	45.9	49.7	86.5
	DeSTSeg [67]	89.2	95.5	91.6	93.1	54.3	50.9	64.8
	UniAD [60]†	96.5	98.8	96.2	96.8	43.4	49.5	90.7
	ReContrast [14]†	98.3	99.4	97.6	97.1	60.2	61.5	93.2
	DiAD [18]†	97.2	99.0	96.5	96.8	52.6	55.5	90.7
	ViTAD [65]†	98.3	99.4	97.3	97.7	55.3	58.7	91.4
	MambaAD [17]†	98.6	99.6	97.8	97.7	56.3	59.2	93.1
	Dinomaly (Ours)	**99.6**	**99.8**	**99.0**	**98.4**	**69.3**	**69.2**	**94.8**
VisA [70]	RD4AD [10]	92.4	92.4	89.6	98.1	38.0	42.6	91.8
	SimpleNet [34]	87.2	87.0	81.8	96.8	34.7	37.8	81.4
	DeSTSeg [67]	88.9	89.0	85.2	96.1	39.6	43.4	67.4
	UniAD [60]†	88.8	90.8	85.8	98.3	33.7	39.0	85.5
	ReContrast [14]†	95.5	96.4	92.0	98.5	47.9	50.6	91.9
	DiAD [18]†	86.8	88.3	85.1	96.0	26.1	33.0	75.2
	ViTAD [65]†	90.5	91.7	86.3	98.2	36.6	41.1	85.1
	MambaAD [17]†	94.3	94.5	89.4	98.5	39.4	44.0	91.0
	Dinomaly (Ours)	**98.7**	**98.9**	**96.2**	**98.7**	**53.2**	**55.7**	**94.5**
Real-IAD [54]	RD4AD [10]	82.4	79.0	73.9	97.3	25.0	32.7	89.6
	SimpleNet [34]	57.2	53.4	61.5	75.7	2.8	6.5	39.0
	DeSTSeg [67]	82.3	79.2	73.2	94.6	37.9	41.7	40.6
	UniAD [60]†	83.0	80.9	74.3	97.3	21.1	29.2	86.7
	ReContrast [14]†	86.4	84.2	77.4	97.8	31.6	38.2	91.8
	DiAD [18]†	75.6	66.4	69.9	88.0	2.9	7.1	58.1
	ViTAD [65]†	82.3	79.4	73.4	96.9	26.7	34.9	84.9
	MambaAD [17]†	86.3	84.6	77.0	98.5	33.0	38.7	90.5
	Dinomaly (Ours)	**89.3**	**86.8**	**80.2**	**98.8**	**42.8**	**47.1**	**93.9**

Method	MVTec-AD [3]			VisA [70]			Real-IAD [54]		
	I-AUROC	P-AUROC	P-AUPRO	I-AUROC	P-AUROC	P-AUPRO	I-AUROC	P-AUROC	P-AUPRO
***Dinomaly** (MUAD)*	*99.6*	*98.4*	*94.8*	*98.7*	*98.7*	*94.5*	*89.3*	*98.8*	*93.9*
Dinomaly	**99.7**	**99.9**	**95.0**	**98.9**	**98.9**	**95.1**	**92.0**	**99.1**	**95.1**
RD4AD [10]	98.5	97.8	93.9	96.0	90.1	70.9	87.1	n/a	93.8
PatchCore [45]	99.1	98.1	93.5	94.7	98.5	91.8	89.4	n/a	91.5
SimpleNet [34]	99.6	98.1	90.0	97.1	98.2	90.7	88.5	n/a	84.6

Dinomaly 실험

논문에서 추가한 Noisy bottleneck(NB), linear attention(LA), loss constraint(LC), loose loss(LL)에 따른 성능 향상 정도를 확인

NB	LA	LC	LL	Image-level			Pixel-level			
				AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
				98.41	99.09	97.41	97.18	62.96	63.82	92.95
✓				99.06	99.54	98.31	97.62	66.22	66.70	93.71
	✓			98.54	99.21	97.62	97.20	62.94	63.73	93.09
		✓		98.35	99.04	97.43	97.10	61.05	62.73	92.60
			✓	99.03	99.45	98.19	97.62	64.10	64.96	93.34
✓	✓			99.27	99.62	98.63	97.85	67.36	67.33	94.16
✓		✓		99.50	99.72	98.87	98.14	68.16	68.24	94.23
✓		✓	✓	99.52	99.73	98.92	98.20	68.25	68.34	94.17
✓	✓	✓		99.57	**99.78**	99.00	98.20	67.93	68.21	94.50
✓	✓	✓	✓	**99.60**	**99.78**	**99.04**	**98.35**	**69.29**	**69.17**	**94.79**

NB, LL의 경우 직접적으로 성능향상에 기여하지만 LA와 LC는 NB가 존재할 경우에만 성능 향상에 기여할 수 있다.

Dinomaly 실험

Model scalability

모델별로 성능이 잘 나오는 backbon이 존재하는 것은 이전의 RD4AD, ViTAD에서 확인 하였는데 Dinomaly에서는 ViT-small에서 SOTA성능이 나왔고, ViT-Large에서는 더 좋은 성능을 보였다.

Arch.	Params	MACs	Im/s	Image-level			Pixel-level			
				AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
ViT-Small	37.4M	26.3G	153.6	99.26	99.67	98.72	98.07	68.29	67.78	94.36
ViT-Base†	148.0M	104.7G	58.1	<u>99.60</u>	<u>99.78</u>	<u>99.04</u>	<u>98.35</u>	<u>69.29</u>	<u>69.17</u>	<u>94.79</u>
ViT-Large	275.3M	413.5G	24.2	**99.77**	**99.92**	**99.45**	**98.54**	**70.53**	**70.04**	**95.09**

input scalability

이미지 크기가 커지면 성능이 저하되는 다른 모델과 다르게 dinomaly는 이미지 크기가 변해도 비슷하게 좋은 성능을 보여준다

Method	Input Size	Image-Level	Pixel-Level
RD4AD	256^2†	**94.6**/96.5/**96.1**	**96.1**/48.6/53.8/**91.1**
	320^2	93.2/**96.9**/95.6	95.7/**55.1**/**57.5**/**91.1**
	384^2	91.9/96.2/95.0	94.9/52.1/55.3/90.8
ReContrast	256^2†	**98.3**/**99.4**/**97.6**	**97.1**/60.2/61.5/93.2
	320^2	98.2/99.2/97.5	96.8/**61.8**/**62.6**/**93.3**
	384^2	95.2/98.0/96.4	96.5/57.7/59.5/92.6
Dinomaly	280^2	**99.6**/**99.8**/**99.3**	98.2/65.2/66.3/93.6
	336^2	**99.6**/**99.8**/99.2	98.3/67.2/67.8/94.2
	392^2†	**99.6**/**99.8**/99.0	**98.4**/**69.3**/**69.2**/**94.8**

Dinomaly 실험

ViT Foundations

이 논문의 모델에서는 DINOv2-R을 이용하여 ViT모델을 사전학습한 모델을 사용하였는데, DINO방식 이외에도 여러 방법을 이용하였을 경우의 성능을 측정하였다.

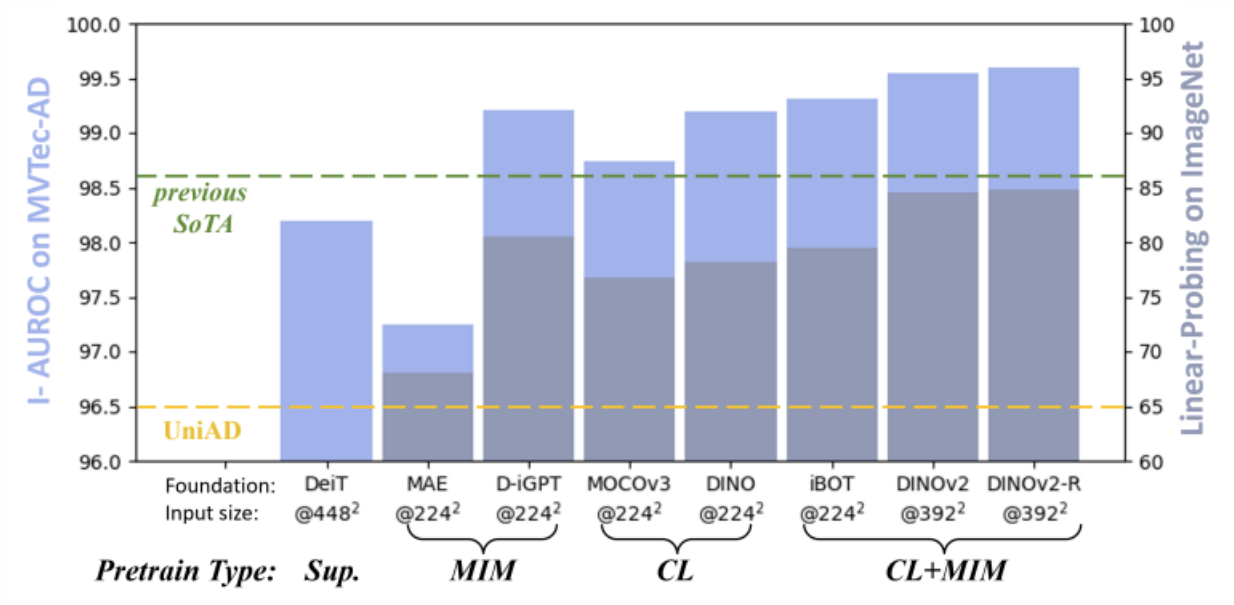


Figure 5. Image-level AUROC of Dinomaly equipped with various ViT foundations, and their linear-probing accuracy on ImageNet. MIM: Masked Image Modeling. CL: Contrastive Learning.

Dinomaly 실험

Attention vs convolution

Attention대신 convolution을 사용하였을 경우 성능비교
Decoder의 attention 대신 convolution으로 대체하였고, inverted bottleneck block을 사용하였다.

Spatial Mixer	Image-level			Pixel-level			
	AUROC	AP	F_1-max	AUROC	AP	F_1-max	AUPRO
ConvBlock 3×3	99.45	99.63	98.64	98.05	65.35	68.07	94.17
ConvBlock 5×5	99.41	99.62	98.86	97.99	66.64	67.47	94.24
ConvBlock 7×7	99.42	99.65	98.86	98.01	67.57	67.94	94.45
Softmax Attention	99.52	99.73	98.92	98.20	68.25	68.34	94.17
Softmax Attention w/ Neighbour-Mask $n = 1$	99.51	99.71	98.90	98.17	67.86	67.92	94.27
Softmax Attention w/ Neighbour-Mask $n = 3$	99.56	99.76	99.05	98.28	69.26	68.17	94.50
Linear Attention	**99.60**	99.78	99.04	98.35	69.29	69.17	**94.79**
Linear Attention w/ Neighbour-Mask $n = 1$	**99.60**	99.78	99.04	98.32	68.77	68.72	94.75
Linear Attention w/ Neighbour-Mask $n = 3$	**99.60**	**99.80**	**99.14**	**98.38**	**69.65**	**69.38**	94.70

Convolution보다 attentio을 사용하였을 때 성능이 좋았다. 하지만 covolution을 이용한 모델도 SOTA성능을 보였다.

참고자료link

AUPRO 설명이미지 – VDS LAB(an introduction to anomaly detection)