AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models *[AAAI 24]*

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, Jinqiao Wang
Foundation Model Research Center, Institute of Automation,
University of Chinese Academy of Sciences, Objecteye Inc., Wuhan AI Research,

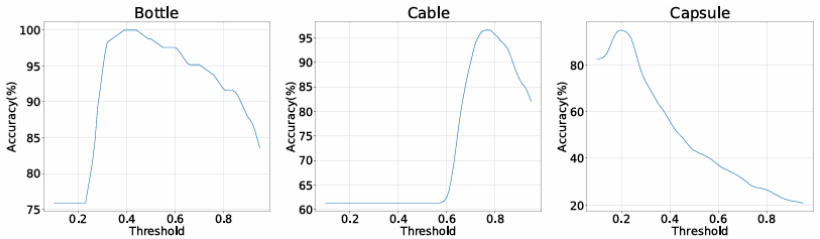
Sunghyun Ahn
skd@yonsei.ac.kr

<2024/07/17>

1 AnomalyGPT

AnomalyGPT

- 기존의 IAD 모델들은 단순히 Anomaly Score를 제공함
- 기존의 LVLM 모델들은 domain-specific한 IAD task를 하기 어려움
- LVLM을 통해 설명 가능한 IAD를 수행하는 AnomalyGPT를 제안함



→ 클래스마다 적절한 threshold가 다르기 때문에,
정상, 비정상을 판단하기 위해서는 ML 사전 지식이 필요함.
따라서 기존 IAD 방식은 실제 제조 현장에서 부적합함

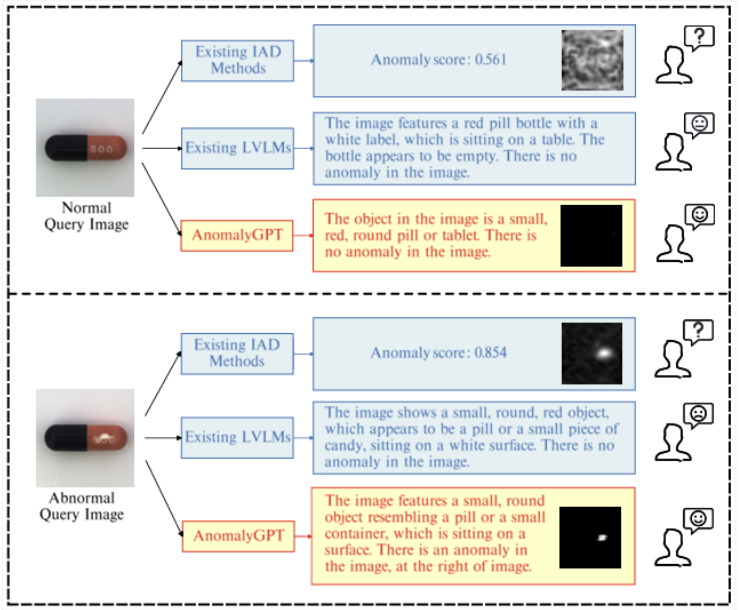
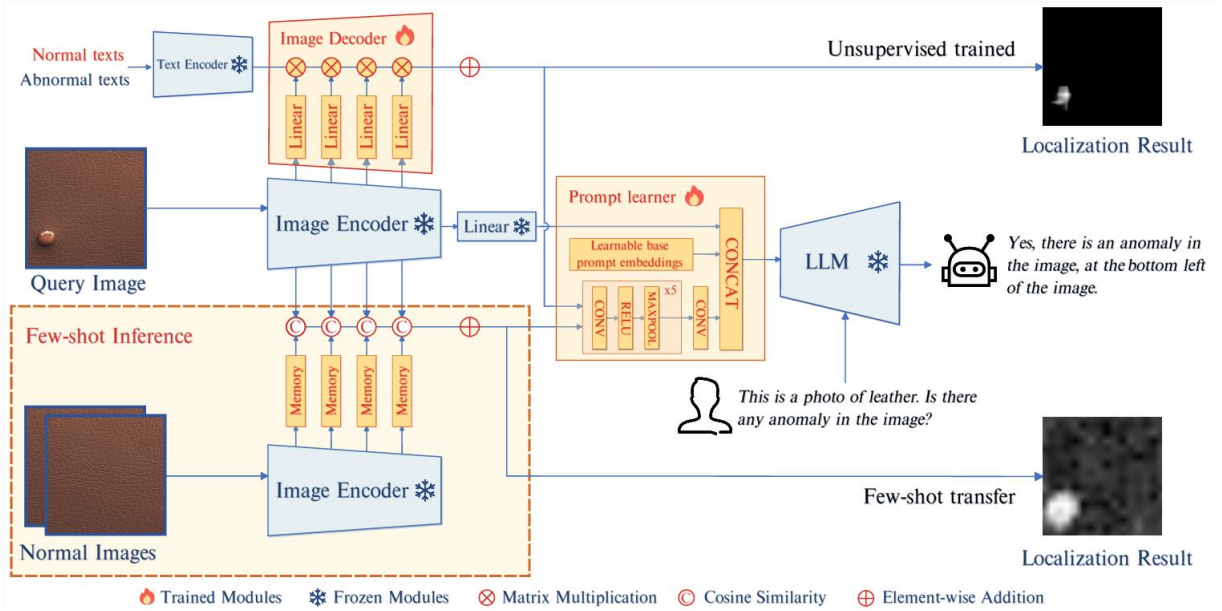


Figure 1. Comparison between our AnomalyGPT, existing IAD methods and existing LVLMs. Existing IAD methods can only provide anomaly scores and need manually threshold setting, while existing LVLMs cannot detect anomalies in the image. AnomalyGPT can not only provide information about the image but also indicate the presence and location of anomaly.

1 AnomalyGPT

Contributions

- Anomaly Localization을 수행할 수 있는 Image Decoder를 제안함
- LLM이 적절하게 이상 탐지를 할 수 있도록 프롬프트를 학습하는 Prompt learner를 제안함
- Decoder 학습없이 Few-normal-shot으로 Anomaly Detection, Localization을 하는 방법을 제안함



1 AnomalyGPT

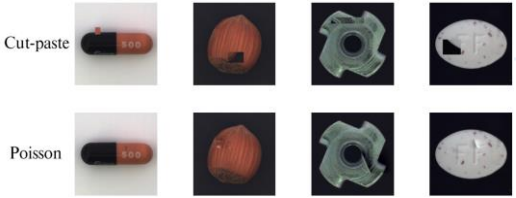


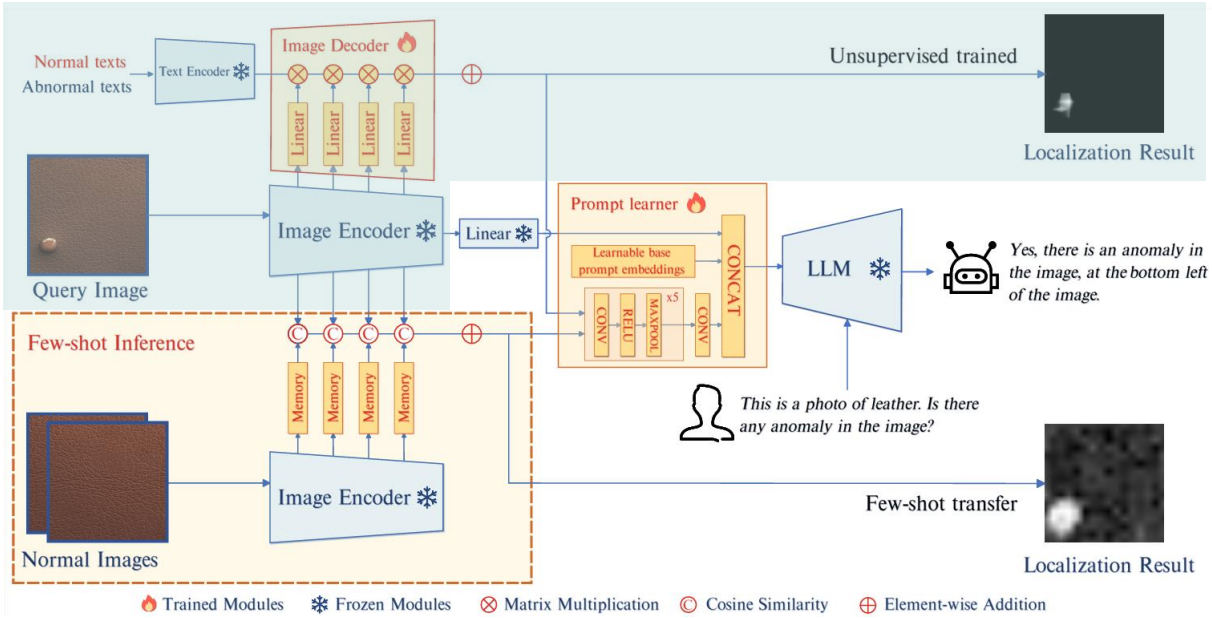
Image Decoder

- Pretrained Image Encoder의 8, 16, 24, 32 layers를 활용하여 각 stage의 패치마다 text와 유사도를 계산함
- Abnormal text와 유사도가 큰 패치들의 위치가 anomaly 영역이 됨
- Focal Loss와 Dice Loss를 이용하여 Anomaly Localization을 학습함

$$M = Upsample\left(\sum_{i=1}^{4} softmax(\tilde{F}^{i}_{patch} F^{T}_{text})\right)$$

$$L_{focal} = -\frac{1}{n}\sum_{i=1}^{n}(1-p_i)^{\gamma} log(p_i),$$

$$L_{dice} = -\frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{n} \hat{y}_i^2},$$



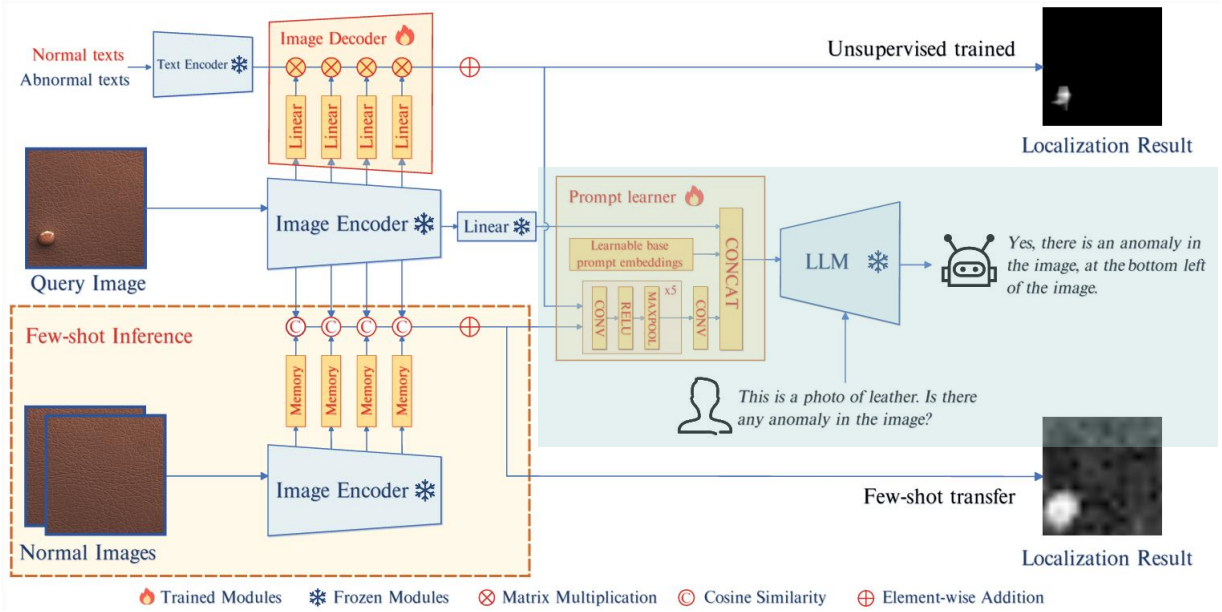
1 AnomalyGPT

Human: <Img> E_{img} </Img> E_{prompt} [Image Description] Is there any anomaly in the image? ### Assistant:

Class	Image description
Bottle	This is a photo of a bottle for anomaly detection, which should be round and without any damage, flaw, defect, scratch, hole or broken part.

Prompt learner

- Image embedding, Localization result, learnable prompt을 입력받아 LLM의 입력 프롬프트를 제작함
- LLM이 출력하는 텍스트와 정답 텍스트 간의 Cross Entropy를 낮추는 방법으로 학습함
- LLM parameter finetuning을 하지 않는 대신 prompt를 튜닝함으로써 LLM의 사전 지식을 잃지 않고 활용 가능



$$E_{img} \in \mathbb{R}^{C_{emb}}$$
$$E_{base} \in \mathbb{R}^{n_1 \times C_{emb}}$$
$$E_{dec} \in \mathbb{R}^{n_2 \times C_{emb}},$$
$$E_{prompt} \in \mathbb{R}^{(n_1+n_2) \times C_{emb}}$$
$$L_{ce} = -\sum_{i=1}^{n} y_i log(p_i),$$

1 AnomalyGPT

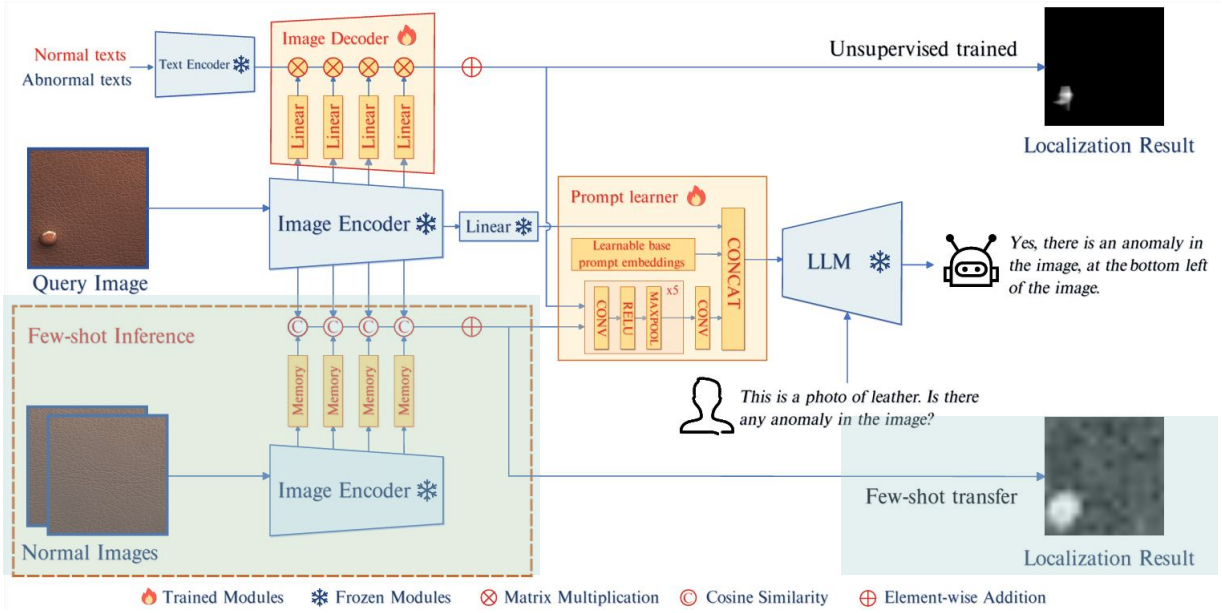
$$M = Upsample\left(\sum_{i=1}^{4}\left(1 - max(F_{patch}^{i} \cdot B^{i^T})\right)\right)$$

$$F_{patch}^{i} \in \mathbb{R}^{H_i \times W_i \times C_i}$$

$$B^{i} \in \mathbb{R}^{N \times C_i}$$

Few-shot Inference

- Few-normal-shot을 메모리에 저장하고, Query Image patches와 유사도를 계산하여, 가장 높은 유사도 점수를 채택함
- 1에서 유사도 점수를 빼는 방식으로 Localization Result를 제작할 수 있음
- Image Decoder를 이용하지 않고 Prompt learner만 이용해서 LLM에 입력함



1 AnomalyGPT

Ablation studies

- Image Embedding과 Image Description만을 이용한 LLM은 72.2%의 정확도를 보임 (학습 X)
- Decoder를 이용하면 Localization Results를 활용 가능하므로 Image-AUC와 Pixel-AUC를 계산 가능함
- Decoder와 Prompt learner를 학습 후 LLM을 활용하면 성능이 약 20% 향상됨

Decoder	Prompt learner	LLM	LoRA	MVTec-AD (unsupervised)			VisA (1-shot)		
				Image-AUC	Pixel-AUC	Accuracy	Image-AUC	Pixel-AUC	Accuracy
		✓		-	-	72.2	-	-	56.5
	✓	✓		-	-	73.4	-	-	56.6
		✓	✓	-	-	79.8	-	-	63.4
✓		✓		97.1	90.9	72.2	85.8	96.2	56.5
✓		✓	✓	97.1	90.9	84.2	85.8	96.2	64.7
✓	✓	✓	✓	96.0	88.1	83.9	85.8	**96.5**	72.7
✓				97.1	90.9	90.3	85.8	96.2	75.4
✓	✓	✓		**97.4**	**93.1**	**93.3**	**87.4**	96.2	**77.4**

Table 4. Results of ablation studies. The ✓ in "Decoder" and "Prompt learner" columns indicate module inclusion. The ✓ in "LLM" column denotes whether use LLM for inference and the ✓ in "LoRA" column denotes whether use LoRA to fine-tune LLM. In settings without LLM, the maximum anomaly score from normal samples is used as the classification threshold. In settings without decoder, due to the sole textual output from the LLM, we cannot compute image-level and pixel-level AUC.

1 AnomalyGPT

Few-shot IAD results

- 기존 IAD 방식과 비교해서 매우 좋은 성능을 보임
- Decoder를 학습하지 않아(Localization results가 부정확함) Unsupervised 방식보다는 부족한 Accuracy를 보임

Setup	Method	MVTec-AD			VisA		
		Image-AUC	Pixel-AUC	Accuracy	Image-AUC	Pixel-AUC	Accuracy
1-shot	SPADE	81.0 ± 2.0	91.2 ± 0.4	-	79.5 ± 4.0	95.6 ± 0.4	-
	PaDiM	76.6 ± 3.1	89.3 ± 0.9	-	62.8 ± 5.4	89.9 ± 0.8	-
	PatchCore	83.4 ± 3.0	92.0 ± 1.0	-	79.9 ± 2.9	95.4 ± 0.6	-
	WinCLIP	93.1 ± 2.0	95.2 ± 0.5	-	83.8 ± 4.0	**96.4 ± 0.4**	-
	AnomalyGPT (ours)	**94.1 ± 1.1**	**95.3 ± 0.1**	**86.1 ± 1.1**	**87.4 ± 0.8**	96.2 ± 0.1	**77.4 ± 1.0**
2-shot	SPADE	82.9 ± 2.6	92.0 ± 0.3	-	80.7 ± 5.0	96.2 ± 0.4	-
	PaDiM	78.9 ± 3.1	91.3 ± 0.7	-	67.4 ± 5.1	92.0 ± 0.7	-
	PatchCore	86.3 ± 3.3	93.3 ± 0.6	-	81.6 ± 4.0	96.1 ± 0.5	-
	WinCLIP	94.4 ± 1.3	**96.0 ± 0.3**	-	84.6 ± 2.4	**96.8 ± 0.3**	-
	AnomalyGPT (ours)	**95.5 ± 0.8**	95.6 ± 0.2	**84.8 ± 0.8**	**88.6 ± 0.7**	96.4 ± 0.1	**77.5 ± 0.3**
4-shot	SPADE	84.8 ± 2.5	92.7 ± 0.3	-	81.7 ± 3.4	96.6 ± 0.3	-
	PaDiM	80.4 ± 2.5	92.6 ± 0.7	-	72.8 ± 2.9	93.2 ± 0.5	-
	PatchCore	88.8 ± 2.6	94.3 ± 0.5	-	85.3 ± 2.1	96.8 ± 0.3	-
	WinCLIP	95.2 ± 1.3	96.2 ± 0.3	-	87.3 ± 1.8	**97.2 ± 0.2**	-
	AnomalyGPT (ours)	**96.3 ± 0.3**	**96.2 ± 0.1**	**85.0 ± 0.3**	**90.6 ± 0.7**	96.7 ± 0.1	**77.7 ± 0.4**

Table 2. Few-shot IAD results on MVTec-AD and VisA datasets. Results are listed as the average of 5 runs and the best-performing method is in **bold**. The results for SPADE, PaDiM, PatchCore and WinCLIP are reported from [11].

1 AnomalyGPT

Qualitative Examples

(Abnormal) 

Is there any anomaly in the image?

PandaGPT: Yes, there is an anomaly in the image. The image shows a person's hand holding a knife, but the knife appears to be missing from the hand. This is an unusual and unnatural detail that deviates from the expected scenario of a person holding a knife.

LLaVA: Yes, there is an anomaly in the image. There is a blue arrow pointing to a section of the wood grain that appears to be different from the rest of the wood.

MiniGPT-4: No, there is no anomaly in the image. The wood grain pattern is consistent throughout the surface of the wood.

AnomalyGPT: Yes, there is an anomaly in the image. 

Where is the anomaly?

PandaGPT: The anomaly is in the middle of the image, where the person's hand is holding a knife but the knife is missing from the hand.

LLaVA: The anomaly is located in the wood grain, specifically in the section of the wood that the blue arrow is pointing to.

MiniGPT-4: There is no anomaly in the image. The wood grain pattern is consistent throughout the surface of the wood.

AnomalyGPT: The anomaly is at the left of the image.

Thank You