Logical Anomaly Detection Using Segmentation Transformers

Yunfei Liang
yl42484@student.uni-lj.si
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Matic Fučka
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Danijel Skočaj
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Abstract

Industrial visual inspection aims to detect and localize defects in manufactured products. This task is commonly addressed using anomaly detection methods that model the appearance of defect-free samples. Such methods perform well on structural anomalies, such as dents and scratches, but they often struggle with logical anomalies. For logical anomalies, individual components appear normal in isolation but their composition may violate constraints of the category (when a component is missing, duplicated, or misplaced). To detect logical anomalies, we propose a two-branch method that uses a Mask2Former-based component segmenter to model both local appearance and global composition. The structural branch extracts local information from a frozen backbone's patch-level features and scores each patch against a coreset-reduced memory bank of defect-free patches. The compositional branch derives a descriptor from the predicted segmentation masks by pooling the same frozen features inside each component region, and compares each component against defect-free components of the same class in a class-partitioned memory bank. The regions establish correspondence with normality through their given class labels, without open-world object detection or inference-time bipartite matching. During inference, anomaly scores are obtained by comparing the local and compositional representations of a query image with those of defect-free training samples, and the two resulting maps are fused into a single prediction. On the MVTec LOCO dataset, the proposed method achieves an image-level AUROC of 95.7% using hand-labeled part maps released with PSAD and 92.9% using automatically generated composition maps released with SALAD, two component-based anomaly detection methods.

Keywords

anomaly detection, logical anomaly detection, semantic segmentation, visual inspection, machine vision

1 Introduction

Industrial anomaly detection learns the appearance of defect-free products and flags deviations at test time, a setting motivated by the fact that anomalous examples are rare and that the defects encountered in production are often not known in advance [3]. Most work has focused on *structural* anomalies such as scratches, dents, contaminations, or other locally visible defects. MVTec LOCO [2] introduced a more difficult setting, *logical* anomalies, where individual components may appear normal in isolation while their composition is incorrect: a component may be missing, duplicated, extra, or placed in an invalid configuration.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.115

The difficulty is that the evidence for a logical anomaly often does not reside in any single abnormal-looking region. A missing component can expose pixels that are themselves perfectly normal for the object category, leaving nothing for a purely appearance-based detector to react to. A local nearest-neighbor detector can compare what is *present* against normal appearance, but it has no direct mechanism for representing what should have been present and is not. Logical anomaly detection therefore requires a representation of expected composition in addition to local appearance.

Several families of methods address this problem. Local–global approaches add a global branch to a local anomaly detector, using image-wide reconstruction or distillation error as an indirect proxy for compositional violations; GCAD, introduced with MVTec LOCO [2], and EfficientAD [1] follow this pattern. Explicit semantic-component approaches instead segment the image into component classes and summarize the resulting composition. ComAD [11], PSAD [10], SALAD [7], and CSAD [9] belong to this family. In our experiments, PSAD supplies manually annotated part maps, while SALAD supplies automatically generated composition maps for training the segmenter. A third line of work represents an image as a set or collection of elements. SINBAD [5] models set-level feature statistics, while ObjectCore [6] and SAM-LAD [12] reason over explicit object representations. ObjectCore is particularly close to our method: it retrieves a normal object set and aligns the query objects to it with bipartite matching, so missing, extra, and visually altered objects become unmatched or high-cost entries.

Our method keeps the object-representation idea but changes the correspondence mechanism. Because each predicted component has a semantic class label, a test component of class c can be compared directly against normal components of the same class in a class-partitioned memory bank. This removes the need for an open-world detector, a separate mask-refinement stage, and inference-time Hungarian matching. The cost of this simplification is that the representation is class-level rather than instance-level: multiple objects of the same class are merged into one semantic region. To compensate for the corresponding loss of sensitivity to small local defects, we retain a separate PatchCore-style structural branch and fuse the two anomaly maps.

Our contributions are: (i) a logical anomaly detector that uses semantic class labels to establish correspondence with normal components; (ii) a complementary structural branch and calibrated map fusion; and (iii) ablations examining the effects of scoring features, segmentation quality, and component-label granularity.

2 Method

The detector has two branches built around a shared frozen anomaly-scoring representation. The structural branch compares local patch descriptors against a coreset-reduced memory of normal patches. The logical branch first decomposes the image into semantic components, pools the frozen scorer features inside

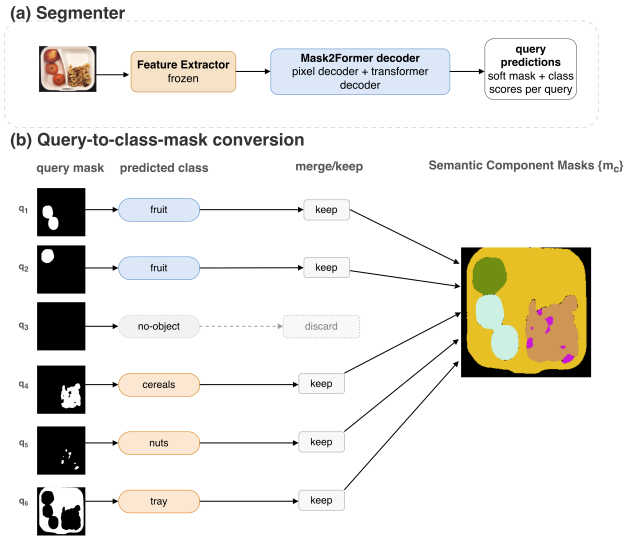

Figure 1: Conversion of Mask2Former query predictions to class masks. Active queries are assigned to their predicted semantic classes, and masks of the same class are merged by element-wise maximum to produce one soft mask m_c per detected class.

each component region, and compares the resulting descriptors against class-specific normal memories. Only the segmentation network is trained; the feature extractor used for anomaly scoring stays frozen, and both anomaly memory banks are constructed directly from normal training images.

Semantic component extraction. We use a Mask2Former-style semantic segmenter [4]. Its feature backbone is frozen, while the segmentation-specific layers are trained from component annotations. For each input image the decoder produces a fixed set of query predictions. Each query outputs a class distribution, including a no-object class, together with a soft spatial mask. Queries whose no-object probability is below 0.5 are retained and assigned to their most likely semantic class. Because several queries can predict the same class, their soft masks are merged by an element-wise maximum,

$$m_c(p) = \max_{q \in \mathcal{A}_c} m_q(p),$$

where $\mathcal{A}_c$ is the set of active queries assigned to class c. We discard merged masks whose mean soft-mask value is below 5×10^{-4} and exclude background from the component classes. The result is one soft mask m_c for each component class detected in the image (Figure 1).

Component representation. The semantic masks specify the location of each component class but do not describe its appearance. We therefore extract a feature map f from a frozen anomaly-scoring backbone and use each soft mask m_c to pool the scorer features inside the corresponding region:

$$\tilde{o}_c = \frac{1}{\sum_p m_c(p)} \sum_p m_c(p)\, f(p), \qquad o_c = \frac{\tilde{o}_c}{\|\tilde{o}_c\|_2}. \tag{1}$$

The soft mask acts as a spatial weighting function, so pixels with high mask confidence contribute more strongly to the descriptor. We additionally record the relative area $a_c = \frac{1}{HW} \sum_p m_c(p)$, which captures how much of the image is occupied by the component. An image is thus represented as the set $O = \{(o_c, m_c, a_c)\}_{c \in C}$ of detected semantic components.

Logical memory and scoring. For each anomaly-free training image, its components are added to a class-partitioned logical memory. For each class c, the bank contains the unit-normalized descriptors of all normal components of that class, their corresponding relative areas, and a mean soft mask $\bar{m}_c$. The mean mask is computed from the predicted soft masks in training images where the class is detected: $\bar{m}_c = \frac{1}{n_c} \sum_{i:\, c \in C_i} m_c^{(i)}$, where C_i denotes the classes detected in training image i and n_c is the number of training images containing a detection of class c. Images where the class is absent are excluded from this average. All masks are represented on the same 256×256 grid.

We also record the fraction of normal training images in which each class is detected. With N training images for the product category, a class is considered expected when $n_c/N \geq 0.5$. An expected class that is not detected in the test image is treated as missing.

For a test component of class c represented in the normal memory, we retrieve its nearest normal descriptor o_s^c using cosine distance alone, together with the relative area a_s^c of that same component. The semantic label therefore supplies the correspondence that ObjectCore must recover with set matching. The component cost combines the appearance difference with a penalty for the relative-area difference between the test component and its selected normal neighbor:

$$d_c = 1 - \langle o_t^c, o_s^c \rangle + \gamma \frac{|a_t^c - a_s^c|}{\max(a_t^c, a_s^c, \varepsilon)}. \tag{2}$$

The weight $\gamma = 0.25$ controls the contribution of the area penalty, while $\varepsilon = 10^{-6}$ keeps the denominator positive.

Let $\mathcal{M}$ denote classes present in both the test image and the normal memory, $\mathcal{E}$ classes detected in the test image but never detected in training, and $\mathcal{S}$ expected normal classes that are missing from the test image. The logical anomaly map is

$$A_L = \sum_{c \in \mathcal{M}} m_t^c\, d_c + \sum_{c \in \mathcal{E}} m_t^c + \sum_{c \in \mathcal{S}} \bar{m}_c. \tag{3}$$

Matched components therefore contribute their test masks weighted by their matching costs. Extra components contribute their own masks directly. A missing component has no test mask, so its mean normal mask is used to localize the region where the component is normally expected.

Structural branch and fusion. The logical descriptor is intentionally coarse: a scratch occupying only a small fraction of a component may barely change its mask-pooled appearance and will not change the component vocabulary. We therefore retain a separate PatchCore-style structural branch [13]. Each spatial feature of the same frozen scorer is treated as a patch descriptor, with no local aggregation in the final $p = 1$ configuration. Normal patch descriptors are reduced to a 100,000-entry bank with greedy k-center coreset subsampling. At test time each patch is scored by squared Euclidean distance to its nearest normal patch descriptor, producing the structural anomaly map A_S.

Because the two branches have different score distributions, each map is calibrated using the 99.5th percentile estimated from anomaly-free validation images. The calibrated maps are combined by a weighted sum, followed by Gaussian smoothing:

$$A = G_\sigma * \left(w_S \frac{A_S}{s_S} + w_L \frac{A_L}{s_L} \right),$$

Table 1: Combined image-level AUROC (%) on MVTec LOCO for the strongest and closest baselines. Rows above the divider use automatically generated component maps or none; rows below use PSAD's hand-labeled maps (SALAD†: SALAD trained with them). B.box = breakfast box; J.bottle = juice bottle; Pins = pushpins; S.bag = screw bag; S.conn. = splicing connectors.

Method	B.box	J.bottle	Pins	S.bag	S.conn.	Avg.
ComAD+PC [11]	86.4	96.6	93.4	80.2	94.1	90.1
SAM-LAD [12]	91.0	97.6	88.2	86.6	90.0	90.7
CSAD [9]	92.8	95.3	98.7	96.6	93.5	95.3
SALAD [7]	89.3	99.7	99.4	95.0	97.3	96.1
Proposed method (SALAD maps)	96.9	95.9	88.5	91.9	91.2	92.9
PSAD [10]	92.5	98.7	94.9	97.5	90.6	94.9
SALAD† [7]	94.2	99.3	99.1	96.6	97.0	97.2
Proposed method (PSAD maps)	96.4	98.5	94.5	96.0	93.3	95.7

Table 2: Branch isolation at the reference configuration.

Maps	Configuration	$\text{AUROC}_{\text{log}}$	$\text{AUROC}_{\text{str}}$	Mean	sPRO
–	structural only	0.7338	0.9740	0.8539	0.7082
PSAD	logical only	0.9399	0.7498	0.8448	0.5517
PSAD	fused	**0.9588**	0.9560	**0.9574**	**0.7693**
SALAD	logical only	0.8703	0.7579	0.8141	0.2986
SALAD	fused	**0.8793**	**0.9783**	**0.9288**	0.6207

Table 3: Selected one-factor-at-a-time ablations. Δ is the change in mean AUROC from the corresponding reference configuration.

Maps	Change	Mean AUROC	Δ
PSAD	scorer RADIO-L → WRN50	0.8711	-0.0863
PSAD	greedy → random coreset	0.9458	-0.0115
PSAD	γ : 0.25 → 1.0	0.9394	-0.0180
PSAD	MSDef → FPN	0.9545	-0.0029
SALAD	scorer RADIO-L → WRN50	0.8816	-0.0472
SALAD	greedy → random coreset	0.9200	-0.0087

and the image-level anomaly score is the maximum value of the fused map. The reference configuration uses $w_S = w_L = 1$ and $\sigma = 4$.

3 Experiments

Dataset and evaluation. All experiments use the five product categories of MVTec LOCO [2]. The training and validation splits contain only anomaly-free images, while the test split contains normal images together with structural and logical anomalies. We use two released component-annotation sources. SALAD [7] provides automatically generated composition maps with 18 relatively coarse classes, whereas PSAD [10] provides 53 hand-labeled part classes, some of which encode position. The same anomaly pipeline is used with both sources; no new component annotations are introduced for these experiments.

We report image-level AUROC separately for logical and structural anomaly subsets, their unweighted mean, and saturated per-region overlap (sPRO) at FPR≤0.05, using the official MVTec LOCO evaluator. The segmenter is trained separately for each category for 100 epochs with AdamW using a frozen RADIO v2.5-B backbone [8] and a multi-scale deformable-attention pixel decoder. Anomaly scoring uses a frozen RADIO v2.5-L at 1024 px, squared L_2 distance in the structural branch, no local aggregation, a greedy 100,000-entry structural coreset, area weight $\gamma = 0.25$, and the fusion settings above.

Comparison to prior work. Table 1 summarizes the closest and strongest baselines. With PSAD component maps, the fused detector reaches 95.7% combined AUROC, above PSAD's own 94.9% and within 1.5 percentage points of the supervised SALAD† result of 97.2%. With automatically generated SALAD maps, the same pipeline reaches 92.9%, below SALAD's 96.1% result. ObjectCore, the closest object-representation reference, reports 90.8% in its full-shot setting; both of our configurations exceed it. The comparison therefore shows that semantic class correspondence can support competitive logical anomaly detection, although the result depends strongly on the component vocabulary used to train the segmenter.

Branch contributions. The two branches behave as intended (Table 2). The structural branch reaches 0.974 AUROC on structural anomalies but only 0.734 on logical anomalies. The logical branch shows the opposite pattern, reaching 0.940 logical AUROC with PSAD maps but only 0.750 structural AUROC. Fusing the two branches increases mean AUROC by more than 0.07 over either isolated branch for both annotation sources.

Main ablations. Before interpreting small changes, we estimate the variation caused by segmenter training randomness across three seeds in earlier PSAD experiments. The standard deviation in mean AUROC ranges from 0.0018 to 0.0054. We use these runs as a practical reference when interpreting small differences and treat changes of around 0.005 or less cautiously.

Among the ablations in Table 3, the frozen anomaly-scoring backbone has by far the largest effect. Replacing RADIO v2.5-L with WideResNet-50-2 reduces mean AUROC by 0.086 with PSAD maps and 0.047 with SALAD maps. By comparison, replacing the multi-scale deformable-attention decoder with an FPN or replacing the segmenter backbone produces much smaller changes in mean AUROC. This indicates that, in these experiments, the pretrained representation used for anomaly scoring matters much more than the architecture used to generate the component masks.

Greedy coreset selection gives a small but consistent gain over random subsampling. The relative-area term matters mainly with the finer PSAD vocabulary: increasing γ from 0.25 to 1.0 reduces mean AUROC by 0.018 on PSAD, while the SALAD result is essentially unchanged. These differences are consistent with the role of area as an indirect cue for count or occupancy when a semantic class merges multiple instances.

Segmentation quality is not the main bottleneck. We additionally test whether improving semantic segmentation improves anomaly detection. Full backbone fine-tuning, late-stage fine-tuning, and LoRA adaptation increase strict segmentation mIoU by roughly 0.006–0.011, yet none improves mean AUROC by more than 0.005 once configurations are compared at the same area-term setting. One possible explanation is that the anomaly detector never consumes the final hard segmentation map directly. Instead, the predicted soft masks are used to pool features from a separate frozen scorer, so moderate changes in hard pixel agreement can leave the pooled component descriptors nearly unchanged.

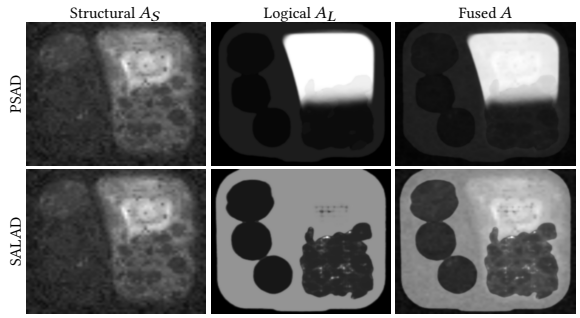

Figure 2: Breakfast box with granola missing from the upper-right compartment. Rows show results using PSAD and SALAD component maps. The structural map A_S is identical in both rows because the structural branch does not use component maps. Each panel is normalized to its own maximum. With PSAD maps, the missing granola component is localized using its mean normal mask $\bar{m}_c$; with SALAD maps, the logical response is over the whole box.

Annotation granularity and position. The annotation source has a much larger effect on the logical branch. At the reference setting, PSAD reaches 0.9588 logical AUROC whereas SALAD reaches 0.8793. The gap is largest for `pushpins` and `screw_bag`. The finer PSAD vocabulary contains more part classes and, for some categories, position-coded labels. Consequently, some arrangement violations that are difficult to express with a coarse semantic class become explicit missing- or extra-class events in the class-partitioned representation. Figure 2 illustrates this on a breakfast box with the granola missing: with PSAD maps the absent class is detected as missing and A_L concentrates on the empty compartment, whereas with SALAD maps the logical response is spread over the whole box.

Because the class-level representation discards some spatial arrangement information, we also tested an explicit class-conditional position term based on the centroid of each predicted component. This extension does not help: increasing its weight decreases mean AUROC on both annotation sources, and at $\gamma_{\text{pos}} = 1$ the mean is roughly 0.075 lower for PSAD and 0.062 lower for SALAD than with the term disabled. The penalty raises scores on normal images as well, reducing the separation between normal and anomalous examples. We therefore use $\gamma_{\text{pos}} = 0$ in the reference model.

Fusion sensitivity. The optimal branch balance depends on the annotation source. With PSAD maps, reducing the logical weight to $w_L = 0.5$ increases mean AUROC from 0.9574 to 0.9650 and also improves sPRO, whereas the same change reduces the SALAD result. We keep equal weights in the reference configuration because MVTec LOCO provides no anomalous validation set on which to choose the branch weights without using the test data.

4 Conclusion

We presented a two-branch anomaly detector in which semantic class labels provide correspondence between a test image and normality. The logical branch uses mask-pooled component descriptors and class-partitioned memories, while a PatchCore-style structural branch captures local appearance faults.

The main limitation is that one semantic mask is retained per class, so multiple same-class instances are not represented independently. Intra-class count is therefore captured only indirectly through pooled appearance and relative area. A natural extension is to convert the semantic segmenter to panoptic output so that individual instances remain separate. The scorer experiments also suggest that the two branches may prefer different frozen feature extractors, making branch-specific scorers another useful direction for future work.

We train the segmenter using component maps released with PSAD and SALAD and create no additional annotations. Applying the method to new categories requires manual labeling effort or an automatic component-annotation pipeline. Our experiments cover only MVTec LOCO, and we do not evaluate runtime. The code is not currently publicly available.

Acknowledgements

This work was in part supported by the ARIS research projects MUXAD (J2-60055) and AI4Science (GC-0001), research programme P2-0214 and the supercomputing network SLING (ARNES, EuroHPC Vega).

References

[1] Kilian Batzner, Lars Heckler, and Rebecca König. 2024. EfficientAD: accurate visual anomaly detection at millisecond-level latencies. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 127–137.

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. 2022. Beyond dents and scratches: logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130, (Apr. 2022).

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2019. MVTec AD – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9592–9600.

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1280–1289.

[5] Niv Cohen, Issar Tzachor, and Yedid Hoshen. 2024. Set features for anomaly detection. *Transactions on Machine Learning Research*.

[6] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. 2026. ObjectCore – efficient few-shot logical anomaly detection using object representations. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3857–3867.

[7] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. 2025. SALAD – semantics-aware logical anomaly detection. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, 21843–21852.

[8] Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. 2025. RADIOv2.5: improved baselines for agglomerative vision foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22487–22497.

[9] Yu-Hsuan Hsieh and Shang-Hong Lai. 2024. CSAD: unsupervised component segmentation for logical anomaly detection. *arXiv preprint arXiv:2408.15628*.

[10] Soopil Kim, Sion An, Philip Chikontwe, Myeongkyun Kang, Ehsan Adeli, Kilian M Pohl, and Sang Hyun Park. 2024. Few shot part segmentation reveals compositional logic for industrial anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence* number 8. Vol. 38, 8591–8599.

[11] Tongkun Liu, Bing Li, Xiao Du, Bingke Jiang, Xiao Jin, Liuyi Jin, and Zhuo Zhao. 2023. Component-aware anomaly detection framework for adjustable and logical industrial visual inspection. *Advanced Engineering Informatics*, 58, 102161.

[12] Yun Peng, Xiao Lin, Nachuan Ma, Jiayuan Du, Chuangwei Liu, Chengju Liu, and Qijun Chen. 2025. SAM-LAD: Segment Anything Model meets zero-shot logic anomaly detection. *Knowledge-Based Systems*, 314, 113176.

[13] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. 2022. Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14298–14308.