CSAD: Unsupervised Component Segmentation for Logical Anomaly Detection

Yu-Hsuan Hsieh
ss111062646@gapp.nthu.edu.tw
Shang-Hong Lai
lai@cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan

Abstract

To improve logical anomaly detection, some previous works have integrated segmentation techniques with conventional anomaly detection methods. Although these methods are effective, they frequently lead to unsatisfactory segmentation results and require manual annotations. To address these drawbacks, we develop an unsupervised component segmentation technique that leverages foundation models to autonomously generate training labels for a lightweight segmentation network without human labeling. Integrating this new segmentation technique with our proposed Patch Histogram module and the Local-Global Student-Teacher (LGST) module, we achieve a detection AUROC of 95.3% in the MVTec LOCO AD dataset, which surpasses previous SOTA methods. Furthermore, our proposed method provides lower latency and higher throughput than most existing approaches.

1 Introduction

In industrial anomaly detection (AD), previous works [2, 26, 31, 32, 36] have demonstrated excellent performance in various datasets, like MVTec AD [3] dataset and VisA [39] dataset. However, the research on logical anomaly detection remains underdeveloped. In this anomaly detection task, we focus on identifying violations of underlying logical constraints in images, such as incorrect quantities, arrangements, and combinations of object components. Although previous efforts [20, 25] have achieved progress by segmenting components and calculating their areas or quantities, they struggle to recognize components with similar textures or even require manual annotations.

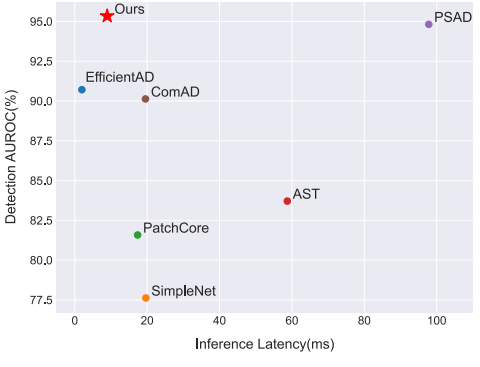

Figure 1: The speed-performance plot on the MVTec LOCO AD benchmark. The x- and y-axis indicate inference latency and average detection AUROC, respectively.

In the field of image segmentation, foundation models, such as the Segment Anything Model (SAM) [21] and Grounding DINO [24], are making notable strides. SAM excels in isolating objects within images using points or boxes as prompts. However, this model cannot identify objects from general text descriptions. Grounding DINO is an open-set object detection model that detects objects based on text prompts and is trained on a vast dataset that includes detection and visual grounding data. Although it has a strong zero-shot object detection performance, it requires textual input. By combining these models, Grounded-SAM [29] can segment any semantic object in the image via text prompts in the wild with high segmentation quality. However, direct application of these models to industrial images does not yield satisfactory results [20], as these methods falter when encountering objects absent during the training phase, significantly impeding their applicability of component segmentation. Furthermore, certain industrial products, such as long and short screws in the "screw bag" category, cannot be consistently segmented using only the term "screw." To address this challenge, we exploit multiple foundation models to generate semantic pseudo-labels of object components and train a segmentation network in a semi-supervised setting. Training a lightweight segmentation network allows efficient segmentation of industrial images without relying on heavy foundation models in the inference stage. Our contributions can be summarized as follows:

- We develop an unsupervised method to generate semantic pseudo-label maps for training a lightweight component segmentation model for a specific logical anomaly detection task without human labeling.
- We propose a Patch Histogram module based on an unsupervised image segmentation network trained from semantic pseudo-labels that can effectively detect both positional and quantity abnormalities of the components in an image.
- We develop a Local-Global Student-Teacher(LGST) module to detect both small- and large-scale anomalies.
- Our approach achieves state-of-the-art performance in identifying logical and structural anomalies, with lower latency and higher throughput than most existing methods.

2 Related Work

2.1 Conventional AD methods

Recently, anomaly detection has predominantly followed the unsupervised setting, wherein the model is trained solely with normal samples and is tested against normal and abnormal samples. Some works [9, 11, 17] model deep features with multivariate Gaussian distributions and compute the Mahalanobis distance to the training set as the anomaly score. Using kNN-based anomaly detection on deep features extracted by a pretrained neural network [1, 19, 27, 30], this approach offers a significant advantage as it does not require training. Student-teacher-based (ST-based) methods [4, 12, 15, 32, 33] were developed based on the assumption that a student model, trained on normal samples only, will exhibit a different feature distribution in anomalous regions compared to a pretrained teacher model.

2.2 Logical AD methods

THFR [16] is an ST-based method that employs bottleneck compression and utilizes normal images as templates to preserve and restore features in anomalous images. DSKD [37] con-

structs two reverse knowledge distillation models: the local student, which reconstructs low-level features to identify structural anomalies, and the global student, which leverages global context to detect logical anomalies. EfficientAD [2] implements a knowledge distillation framework composed of a teacher distilled from a pretrained encoder, a student, and an autoencoder. Both the student-teacher and autoencoder-student pairs are specifically designed to detect small and large-scale anomalies, respectively. Both of the following methods utilize segmentation to improve the performance in detecting logical anomalies. ComAD [25] introduces an unsupervised segmentation method using DINO [7] as a feature extractor. This method segments images by clustering the features, models normal features of object components with a memory bank, and compares the region features of test samples with those of training samples to detect logical anomalies effectively. PSAD [20] trains a segmentation network in a semi-supervised setting using human-annotated ground-truth label maps. It constructs three memory banks: a standard PatchCore [30] patch feature memory bank, a class histogram memory bank that stores class distribution histograms, and a class composition memory bank that aggregates class embeddings. The anomaly detection is then performed by aggregating the anomaly scores from these three memory banks.

3 Methodology

3.1 Semantic Pseudo-label Generation

Component-level Segmentation

A primary challenge with unsupervised segmentation is the tendency toward under- or over-segmentation. To address this problem, we implement two modes in the semantic pseudo-label generation process: fine-grained and coarse-grained modes. Typically, the coarse-grained mode generates enough semantic labels for logical anomaly detection. However, the fine-grained mode is employed when object components are too complex to discern using simple image tags. In our experiments, we use the fine-grained mode for the "screw bag" and the "juice bottle," while the coarse-grained mode is used for other categories. Here, components of the same class are defined as segments sharing similar visual features.

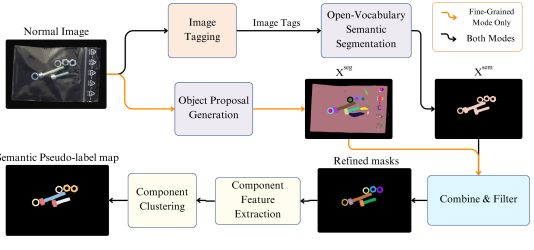

Figure 2: Proposed semantic pseudo-label generation that generates semantic pseudo-labels from normal images only.

Figure 2 shows the procedure of our semantic pseudo-label generation. To create a semantic-aware component segmentation, we initially employ an open-vocabulary semantic segmentation model and utilize an object proposal generation model to segment each component within the image more precisely.

Image Tags Generation We utilize an image-tagging model, Recognize Anything++ (RAM++) [18] model, to automatically generate text prompts for semantic segmentation. RAM++, an advanced iteration of the original Recognize Anything(RAM) [38] image tagging foundation model, is trained for the open-set image tagging challenge. It operates by using a normal image from each category as input. Subsequently, the generated tags are

manually refined to remove non-noun or background tags such as "attach" and "connect" found in the "splicing connector" category and "zip-lock bag" from "screw bag." A list of tags for all categories is available in the supplementary material.

Mask Refinement Next, for a training image X_i , we use the text prompts previously created and Grounded-SAM as our open-vocabulary semantic segmentation model to generate a semantic segmentation map X_i^{sem} . Additionally, we employ the automatic mask generation feature of the SAM model as our object proposal generation model to produce all possible masks of components, denoted by X_i^{seg}, allowing segmentation of each object component. In the categories operating under the fine-grained mode, an algorithm is implemented to filter noise in X_i^{seg} using X_i^{sem}, producing refined masks X_i^{ref}. A detailed description of this algorithm can be found in the supplementary material. The refined masks are identical to X_i^{sem} for categories operating in the coarse-grained mode.

Component Feature Extraction After obtaining the refined masks, we need to cluster them into semantic pseudo-labels by their visual features. Since we expect the components with the same shape and texture to belong to the same cluster, we need to eliminate the influence of rotation on an object component. To achieve this, we crop each segment and resize the diagonal of the minimum bounding rectangle of the segment to 64x64. After that, we apply rotation augmentation on each component and extract their corresponding features from the fourth layer of a pretrained CNN, with the rotation-augmented feature map denoted by $f^{rot} \in \mathbb{R}^{R\times C\times H\times W}$, where R is the number of rotations, and C, H, W represent feature channel, height, and width, respectively. We take the average of the feature maps over H, W, and R dimensions to form a rotation-invariant feature vector $f^{com} \in \mathbb{R}^C$ representing the component.

Component Clustering We cluster all components in the training samples by their f^{com} using MeanShift [10] to avoid the cluster number selection. Assuming that each component should appear in all images, we discard any clusters with less than α members. We denote the number of these filtered clusters as N_{cls}, and the semantic pseudo-label map Y_i with each pixel value representing its class number. In our experiments, we defined α as half of the training samples.

Filtering Unreliable Semantic Pseudo-label Maps To ensure greater consistency across semantic pseudo-label maps, we compute a class histogram $Hist(Y_i) \in \mathbb{R}^{N_{cls}}$ as described in Equation (1)

$$Hist(Y_i)[k] = \frac{1}{W\times H}\sum_{x=1}^{W}\sum_{y=1}^{H}\mathbb{I}(Y_i(x,y)=k),\ \text{where } k=1,2,\ldots,N_{cls} \tag{1}$$

where $\mathbb{I}(Y_i(x,y)=k)$ is an indicator function that equals 1 if $Y_i(x,y)=k$ and 0 otherwise. Assuming that the size of the object components remains consistent across normal images, the corresponding class histogram should be similar. We apply HDBSCAN [6], an enhanced version of DBSCAN [12], to the histograms to eliminate incorrect semantic pseudo-label maps, discarding those that do not belong to the largest cluster.

To this end, we obtain N_l normal images with high-quality semantic pseudo-label maps (labeled images), denoted as $\{X_1^l, ..., X_{N_l}^l\}$, the corresponding semantic pseudo-label maps are $\{Y_1^l, ..., Y_{N_l}^l\}$, and the rest of the unlabeled images are $\{X_1^u, ..., X_{N_u}^u\}$.

3.2 Component Segmentation

Segmentation Network Architecture We utilize a DeepLabV3+ [8] decoder that processes feature maps of multiple scales drawn from the intermediate layers of a pretrained CNN. The overall architecture of the segmentation network is a U-shape CNN with skip connections. Example images of the segmentation result are shown in Figure 3.

Logical Synthetic Anomalies (LSA) Training with normal images only for component segmentation may cause the model to overfit to component locations. In other words, it may predict labels based on the component's position rather than its semantic features. To address this problem, we propose Logical Synthetic Anomalies (LSA). This augmentation places additional components into the image to simulate logical anomalies, such as incorrect component counts or misplacements. In this way, we can increase the diversity of the training set for component segmentation, especially since the segmentation network relies only on a subset of normal images for supervised training. Example images can be found in the supplementary material.

Loss Functions For the training of the segmentation model, we employ Cross Entropy loss $\mathcal{L}_{CE}$, Dice loss $\mathcal{L}_{Dice}$ [28], and Focal loss $\mathcal{L}_{Focal}$ [23] for labeled images. Following the approach described in PSAD, histogram matching loss $\mathcal{L}_{Hist}$ is applied to unlabeled images. The loss measures the difference between the class histogram of the input image and that of a randomly sampled labeled image, i.e.,

$$\mathcal{L}_{hist} = \frac{1}{N_{cls}} \sum_{k=1}^{N_{cls}} \|Hist(Y_i)[k] - Hist(P_i)[k]\| \tag{2}$$

where P_i is the label map predicted by the segmentation network from X_i. Furthermore, the entropy loss $\mathcal{L}_{Ent}$ [34] is used to reduce the uncertainty of the prediction. The total loss for training the segmentation network model is given in Equation (3).

$$\mathcal{L}_{seg} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{Dice} + \lambda_3 \mathcal{L}_{Focal} + \lambda_4 \mathcal{L}_{Hist} + \lambda_5 \mathcal{L}_{Ent} \tag{3}$$

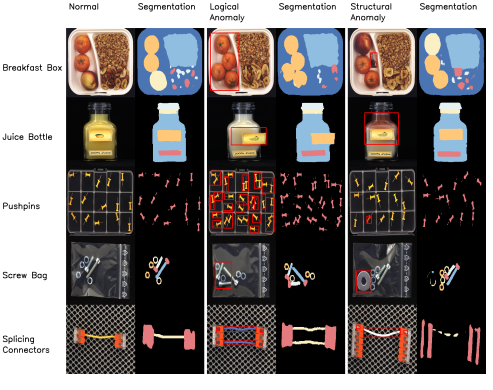

Figure 3: Segmentation result of five categories from MVTec LOCO AD. The red bounding box indicates the anomalous region of the image, and the color in the segmentation image represents the class label of the pixel.

3.3 Patch Histogram

Class Histogram Using the segmentation map P_i predicted by the segmentation network, we can identify logical anomalies by comparing the class histogram of a given image with that of normal images. By calculating the Mahalanobis distance from the class histogram of the testing sample to the class histograms of the training samples, we obtain the class histogram anomaly score as follows:

$$M(P_i) = \sqrt{(Hist(P_i) - \mu_{hist})^T \sum_{hist}^{-1} (Hist(P_i) - \mu_{hist})} \tag{4}$$

where μ_{hist} and $\sum_{hist}$ represent the mean and covariance matrix of class histograms estimated from training samples.

Patch Histogram By analyzing the class histogram of the image, we can identify logical anomalies, including incorrect quantities of object components and some structural inconsistencies. However, some logical anomalies cannot be detected through class histograms. For instance, as illustrated in Figure 4, in "splicing connectors," a type of logical anomaly is "Wrong_cable_location," where a wire is connected to an incorrect socket. This type of anomaly cannot be identified by the class histogram, as the distribution of different component classes remains consistent with that of a normal image.

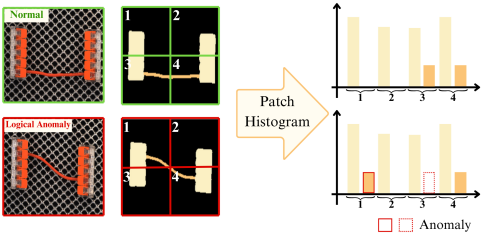

Figure 4: An example illustrating the effectiveness of patch histograms in addressing position-related logical anomalies.

To address this limitation, we introduce the patch histogram. This technique divides the predicted segmentation map into a $P \times P$ grid, constructs a class histogram for each grid cell, and concatenates these histograms to form a patch histogram vector $H_i^p \in \mathbb{R}^{(Cls \times P \times P)}$. The calculation of the anomaly score follows the same method as Equation (2).

3.4 Local-Global Student-Teacher(LGST)

Model Architecture Similar to the architecture of EfficientAD [2], our Local-Global Student-Teacher(LGST) branch comprises two student models and one teacher network as shown in Figure 5. The student model's primary goal is to learn the normal feature distribution from the teacher. Since the students are not exposed to abnormal feature distributions, the feature output in anomalous regions will differ between the teacher and students. To detect small-scale anomalies, the local student network extracts features with a 33x33 receptive field. In contrast, the global student utilizes a bottleneck design that reduces the spatial dimension to 1x1, making the receptive field cover the entire image. This dual approach effectively captures both small-scale and large-scale anomalies.

Difference between LGST and EfficientAD In the EfficientAD framework, the local student is required to output a feature map with double the channels. Half of these channels are dedicated to computing the local anomaly map, and the other half to computing the global anomaly map. In our LGST module, we avoid doubling the channels. Instead, after the

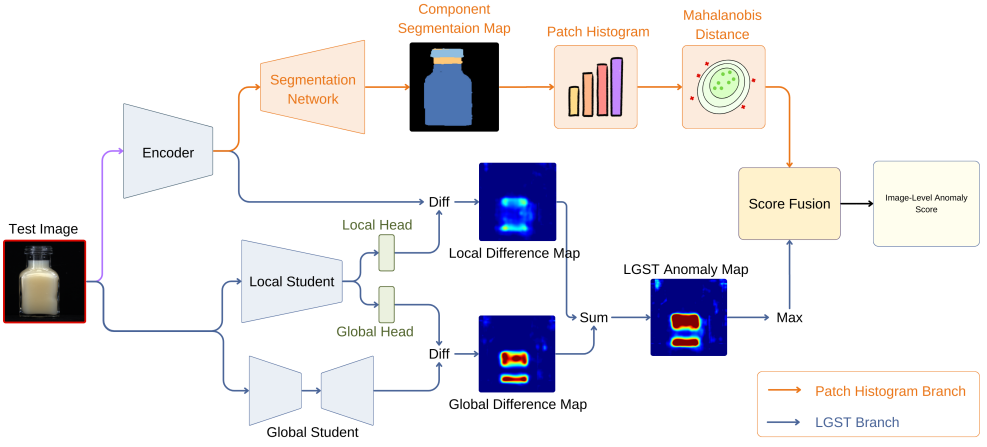

Figure 5: **Overall architecture of the proposed CSAD in the inference stage.** It consists of two branches: a Patch Histogram branch that detects anomalies using component segmentation and an LGST branch that detects both small and large-scale anomalies.

local student, we employ two separate heads, each consisting of one layer of 1x1 convolution. These heads generate two feature maps, each retaining the channel size the same as the teacher's output. This modification can reduce computational cost while only slightly affecting performance. Next, we replace the distilled PDN with a pretrained CNN, which allows us to make use of both shallow and deep features for the LGST and the segmentation network in a single forward pass, significantly reducing overall latency.

3.5 Anomaly Score

Score Fusion To effectively combine the image-level anomaly scores derived from both the Patch Histogram and LGST branches, a direct summation is not feasible due to differences in scale and variation. To tackle this issue, we normalize Patch Histogram anomaly scores and LGST anomaly scores separately before summation. The normalized score, denoted $\hat{S}$, is computed using the formula $\hat{S} = (S - \mu_s)/\sigma_s$, where S is the original anomaly score, μ_s and σ_s denote the trimmed mean and the trimmed standard deviation, respectively, calculated from the anomaly scores of the validation set with the trimmed range equal to (20%, 80%). This range helps to filter out the extreme values. The final anomaly score is the summation of the normalized Patch Histogram anomaly score and normalized LGST anomaly score.

4 Experiments

4.1 Implementation Details

Segmentation Network We utilize features extracted from the first and third layers of the ImageNet [13] pretrained WideResNet-50 [35] as inputs for the DeepLabV3 [8] decoder. All images are resized to 256x256. The weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ are empirically assigned to 0.5, 10, 1, 10, and 1, respectively. Over 120 epochs, training focuses exclusively on the segmentation network decoder, with the initial 50 epochs using only labeled images.

MVTec LOCO AD	Category	No Segmentation				Supervised	Unsupervised	
		SimpleNet[26]	PatchCore[30]	AST[31]	EfficientAD[2]	PSAD*[20]	ComAD[25]	CSAD(Ours)
Logical Anomalies (LA)	Breakfast Box	77.1	74.8	80.0	85.5	**100.0**	91.1	94.4
	Juice Bottle	87.8	93.9	91.6	98.4	**99.1**	95.0	94.9
	Pushpins	69.0	63.6	65.1	97.7	**100.0**	95.7	99.5
	Screw Bag	51.6	57.8	80.1	56.7	99.3	71.9	**99.9**
	Splicing Connectors	72.0	79.2	81.8	**95.5**	91.9	93.3	94.8
	Average(Logical)	71.5	73.9	79.7	86.8	**98.1**	89.4	96.7
Structural Anomalies (SA)	Breakfast Box	80.9	80.1	79.9	88.4	84.9	81.6	**91.1**
	Juice Bottle	90.4	98.5	95.5	**99.7**	98.2	98.2	95.6
	Pushpins	81.6	87.9	77.8	96.1	89.8	91.1	**97.8**
	Screw Bag	83.3	92.0	**95.9**	90.7	95.7	88.5	93.2
	Splicing Connectors	82.6	88.0	89.4	**98.5**	89.3	94.9	92.2
	Average(Structural)	83.7	89.3	87.7	**94.7**	91.6	90.9	94.0
Total Average		77.6	81.6	83.7	90.7	94.8	90.1	**95.3**
Throughput		114.3	109.9	28.2	**2369.8**	10.2	69.9	321.8
Latency		19.6	17.3	58.7	**1.9**	97.8	19.5	8.9

Table 1: Comparison of MVTec LOCO performance with state-of-the-art methods, as measured by image AUROC. The top row indicates which type of component segmentation is used. The mark * denotes that the throughput is measured with a batch size equal to 1 since PSAD does not provide a batch inference. The highest and second highest scores are highlighted in **bold** and underline, respectively.

LGST Our training scheme for the LGST branch is the same as that used in EfficientAD. Images are resized to 256×256, and the feature map of one image extracted from all subnetworks in LGST has the same shape of [512,56,56].

4.2 Experimental Setup

We evaluate our method on the MVTec LOCO AD [6] dataset and compare its performance with SimpleNet [26], PatchCore [30], AST [31], EfficientAD [2], PSAD [20] and ComAD [25], focusing on their detection performance, latency, and throughput. All experiments are run on a PC equipped with an RTX-3090 GPU and a Core i5-13600K CPU.

Following established methods, we use AUROC as our evaluation metric. For the speed analysis, we measure latency with a batch size of 1 and report the mean latency of 500 runs of inference. The throughput is calculated by $throughput = (batch\_size \times runs)/total\_time$ with 500 runs and a batch size of 8.

4.3 Experimental Results

Performance on MVTec LOCO AD As shown in Figure 1 of the MVTec LOCO benchmark, our method achieves an AUROC of 94.0% for structural anomalies and 96.7% for logical anomalies, showcasing superior detection capabilities. With a remarkable total average AUROC of 95.3%, our approach excels in achieving high accuracy on both logical and structural anomalies simultaneously.

Latency and Throughput Table 1 presents a speed comparison between our method and other approaches. Although our method does not surpass the impressive latency and throughput of EfficientAD, the latency of our method is just 8.9 milliseconds, demonstrating a considerable improvement over other segmentation-based models like PSAD and ComAD. The throughput of our method is also noteworthy at 321.8 images per second, indicating a competitive capability for high-speed anomaly detection. Figure 1 is the speed-performance plot of our and other SOTA methods. It shows that our method reaches a new SOTA perfor-

mance with the highest anomaly detection accuracy and lower latency than most existing approaches.

Model	LA	SA	Mean
PSAD	**94.2**	71.1	82.7
ComAD	87.7	74.6	81.2
PatchHist	91.4	**75.4**	**83.4**

Table 2: MVTec LOCO AD performance comparison of segmentation modules of different methods and different settings measured in AUROC. LA and SA denote logical anomalies and structural anomalies, respectively.

Augmentation	LA	SA	Mean
None	89.2	71.0	80.1
CutPaste	89.0	70.1	79.6
LSA	**89.9**	**73.7**	**81.8**

Table 3: MVTec LOCO AD performance comparison of different augmentation methods used in training the segmentation network. The detection performance is measured in AUROC.

Segmentation Branch In Table 2, we compare our Patch Histogram branch with other segmentation-based methods. We only compare the segmentation-related modules. In detail, we report the score of ComAD excludes the PatchCore branch and the score of PSAD with only the histogram memory bank; the scores are retrieved from their original papers. As for our method, we report the score using only the Patch Histogram branch. PSAD attains the highest scores in detecting logical anomalies through precise human annotations. Meanwhile, our Patch Histogram has achieved an AUROC of 75.4% in structural anomalies, outperforming other techniques and securing the highest average score of 83.4% AUROC. These results prove the effectiveness of our Patch Histogram design in detecting both types of anomalies.

Augmentation of Semantic Pseudo-label Map Table 3 shows the comparison of using LSA and CutPaste [22] during training the segmentation network. The result demonstrates that our LSA augmentation improves the segmentation network's robustness. Unlike CutPaste, which randomly cuts and pastes image patches, LSA augments images and labels while preserving the semantic integrity of the components. For instance, in the "screw bag" category, the model can not distinguish between long screws and short screws with just partial screws in CutPaste-augmented images, thereby impairing anomaly detection performance.

4.4 Ablation Study

Patch Size of Patch Histogram Table 4 illustrates that each patch size combination in the histogram delivers consistently similar performance across various settings, significantly improving the detection of both logical and structural anomalies compared to that without using patch histogram (patch size=256). However, as the patch size decreases, there is a

notable increase in latency. Considering latency and performance, we select 256+128 as our setting. The score fusion for different patch sizes is the same as that described in Sec 3.5.

Impact of Different Components and Settings of CSAD Table 5 shows the performance and speed of different settings of CSAD. From the first three rows, we compared our local student design with that of EfficientAD, the result shows that our design can reduce latency with a small performance drop. From the fourth and fifth rows, LSA improves both logical and structural detection performance. The last two rows indicate that using either the class histogram or the patch histogram results in a comparable logical anomaly detection in CSAD, we attribute this to the limitation as noted in the supplementary material. In contrast, using patch histograms can improve the detection performance of structural anomalies due to the component area variation in some structural anomalies.

Patch Size				LA	SA	Mean	Latency(ms)
256	128	85	64				
✓				89.9	73.7	81.8	**4.2**
✓	✓			91.4	75.4	83.4	5.6
✓	✓	✓		**91.6**	**76.0**	**83.8**	10.1
✓	✓		✓	90.8	75.2	83.0	9.8
✓	✓	✓	✓	91.1	75.7	83.4	14.1

Table 4: Performance and speed of different patch size combinations in the Patch Histogram module. The detection performance is measured in AUROC.

Setting			LA	SA	Mean	Latency(ms)
Local Student	PatchHist	LSA				
Same	✘	✘	72.5	63.5	68.06	**6.51**
DoubleChannel	✘	✘	88.0	92.8	90.40	6.89
Heads	✘	✘	87.7	93.1	90.38	6.73
Heads	256	✘	96.5	92.4	94.45	7.84
Heads	256	✓	**96.7**	93.8	95.25	7.84
Heads	256+128	✓	**96.7**	**94.0**	**95.34**	8.96

Table 5: Impact of different components and settings of CSAD. In the design of the local student, "Same" means that the output feature maps of the local student used to calculate local and global difference maps are the same, "DoubleChannel" is the design of EfficientAD, and "Heads" is the design of CSAD. The detection performance is measured in AUROC.

5 Conclusion

In this paper, we presented a segmentation-based approach that significantly improves industrial anomaly detection. We achieved accurate component segmentation without human annotations by exploiting multiple foundation models. Integrating the component segmentation network with the Patch Histogram and LGST modules, our method outperforms the current state-of-the-art methods with higher accuracy and lower latency, as demonstrated in our experiments on the MVTec LOCO AD dataset.

Acknowledgments

This work was supported in part by the National Science and Technology Council, Taiwan under grants NSTC 111-2221-E-007-106-MY3 and NSTC 112-2634-F-007 002.

References

[1] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Pni: industrial anomaly detection using position and neighborhood information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6373–6383, 2023.

[2] Kilian Batzner, Lars Heckler, and Rebecca König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 128–138, 2024.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. doi: 10.1109/CVPR.2019.00982.

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. doi: 10.1109/cvpr42600.2020.00424. URL http://dx.doi.org/10.1109/CVPR42600.2020.00424.

[5] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022.

[6] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[9] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020.

[10] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[11] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021.

[12] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9737–9746, 2022.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[15] Zhihao Gu, Liang Liu, Xu Chen, Ran Yi, Jiangning Zhang, Yabiao Wang, Chengjie Wang, Annan Shu, Guannan Jiang, and Lizhuang Ma. Remembering normality: Memory-guided knowledge distillation for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16401–16409, 2023.

[16] Hewei Guo, Liping Ren, Jingjing Fu, Yuwang Wang, Zhizheng Zhang, Cuiling Lan, Haoqian Wang, and Xinwen Hou. Template-guided hierarchical feature restoration for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6447–6458, 2023.

[17] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratlin, and Yanfeng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision (ECCV)*, 2022.

[18] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-set image tagging with multi-grained text supervision. *arXiv e-prints*, pages arXiv–2310, 2023.

[19] Jeeho Hyun, Sangyun Kim, Giyoung Jeon, Seung Hwan Kim, Kyunghoon Bae, and Byung Jun Kang. Reconpatch: Contrastive patch representation learning for industrial anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2052–2061, 2024.

[20] Soopil Kim, Sion An, Philip Chikontwe, Myeongkyun Kang, Ehsan Adeli, Kilian M Pohl, and Sang Hyun Park. Few shot part segmentation reveals compositional logic for industrial anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8591–8599, 2024.

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[22] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[24] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[25] Tongkun Liu, Bing Li, Xiao Du, Bingke Jiang, Xiao Jin, Liuyi Jin, and Zhuo Zhao. Component-aware anomaly detection framework for adjustable and logical industrial visual inspection. *Advanced Engineering Informatics*, 58:102161, 2023. ISSN 1474-0346. doi: https://doi.org/10.1016/j.aei.2023.102161. URL https://www.sciencedirect.com/science/article/pii/S1474034623002896.

[26] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023.

[27] Declan McIntosh and Alexandra Branzan Albu. Inter-realization channels: Unsupervised anomaly detection beyond one-class classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6285–6295, 2023.

[28] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[29] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[30] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022.

[31] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2592–2602, 2023.

[32] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan Duong, Chanh D Tr Nguyen, and Steven QH Truong. Revisiting reverse distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24511–24520, 2023.

[33] Shenzhi Wang, Liwei Wu, Lei Cui, and Yujun Shen. Glancing at the patch: Anomaly localization with global and local feature comparison. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 254–263, June 2021.

[34] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4248–4257, 2022.

[35] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[36] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021.

[37] Jie Zhang, Masanori Suganuma, and Takayuki Okatani. Contextual affinity distillation for image anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 149–158, 2024.

[38] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023.

[39] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. *arXiv preprint arXiv:2207.14315*, 2022.