Blind Localization and Clustering of Anomalies in Textures

Supplementary Material

S1. Summary

This supplementary material includes additional insights regarding the proposed anomaly clustering method in the form of visualizations, comparisons, evaluations, and details useful for the reproducibility of our results.

S2. Detailed clustering evaluation

In Table 5, we present the metrics' breakdown at the level of texture classes for MVTec AD [10]. MTD and Leaves datasets are not included since they contain a single texture class only.

S3. Additional ablation results.

In Table 6, we present an ablation study in a similar manner to Table 3 from the main text. The difference is that these results are obtained under k-means clustering as the final step instead of agglomerative clustering with Ward linkage. The results show that our contributions hold independent from a particular choice of feature-clustering method.

To develop our blind anomaly localization system, we build on FCA [5] as a state-of-the-art zero-shot anomaly localization method. We argue that this zero-shot detection synergizes well with the VAE reconstruction, fully capitalizing on the information in the residual maps (computed according to Equation (1)). In Table 7, we substantiate our decision by replacing FCA as the residual maps processor with different ways to compare the original features to the reconstruction. We compare against various approaches from the relevant literature: L^1 and L^2 norm [6, 14], SSIM [8], and Rec-grad [51, 52].

There are, of course, many other levels on which we could analyze our pipeline, including the effect of different hyperparameters. However, we observe that most parameters of our method, such as the number of layers in the VAE and feature refiner H, number of neighbors k and margin for contrastive learning, as well as the optimization parameters, have limited effect on the final performance or may offer benefits only for specific datasets. In any case, we did not perform an exhaustive hyperparameter search, and a different configuration could potentially yield better results.

S4. Automatic threshold estimation

Our contrastive learning formulation assumes binary anomaly maps. In order to maintain a fully unsupervised method, we propose an algorithm to estimate the threshold t for binarization. One can observe from the blind anomaly localization results (Table 2) that image-level anomaly scores are reasonably reliable (83% on MTD and almost a perfect result on MVTec AD). Therefore, if we knew in advance the proportion of normal samples in the input set, we could estimate the threshold as the corresponding quantile in the predicted image-level anomaly scores.

However, in a fully unsupervised scenario, this proportion is also unknown. To estimate this value, we perform an initial k-means clustering using descriptors D_i (see Equation (2)). We then find the normality cluster by taking the group with the smallest average anomaly score. The size of this cluster divided by the total number of images is finally used as the ratio of normal images to compute the threshold t.

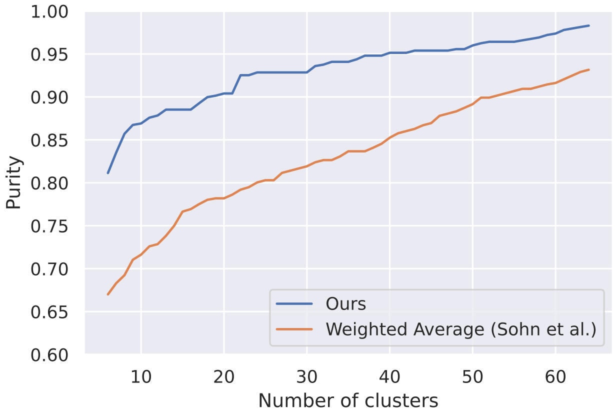

Figure 6. Visualization of the increase of purity with respect to over-clustering. The purity is averaged over all MVTec AD textures.

S5. Unknown number of anomaly types

The quantitative and qualitative results in the main paper are obtained under the common assumption that the number of anomaly types is known in advance. This is made primarily to facilitate comparison across different methods. However, in practice, the anomaly types residing in the input set might be unknown.

We note that up to the final feature-based clustering, our algorithm is not constrained to a predefined number of classes[1]. This offers great flexibility as the image descriptors can be used in other ways such as the automatic discovery of the number of clusters or as a tool to reduce manual labeling efforts (through over-clustering). We show this in Figure 6 by plotting the purity as a function of the number of clusters.

Method	Carpet			Grid			Leather			Tile			Wood		
	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1
SPICE [37]	0.202	0.031	0.231	0.197	0.033	0.166	0.373	0.153	0.016	0.284	0.490	0.176	0.284	0.130	0.282
STEGO [26]	0.271	0.167	0.402	0.123	0.002	0.282	0.338	0.202	0.395	0.790	0.707	0.701	0.423	0.274	0.515
Average [42]	0.287	0.138	0.392	0.158	0.033	0.326	0.398	0.218	0.465	0.288	0.157	0.444	0.231	0.066	0.384
Max. Hausdorff [42]	**0.660**	**0.586**	**0.795**	0.129	0.018	0.308	0.725	0.652	0.762	0.932	0.914	0.957	0.678	0.500	0.716
Weighted Average [42]	0.656	0.576	0.647	0.143	0.018	0.304	0.778	0.674	0.704	0.933	0.921	0.957	**0.860**	**0.815**	0.921
Ours	0.617	0.458	0.590	**0.710**	**0.620**	**0.718**	**0.832**	**0.779**	**0.823**	**0.947**	**0.936**	**0.974**	0.846	0.788	**0.926**

Table 5. Detailed breakdown of the metrics generated by different anomaly clustering methods on the MVTec AD [10] textures. The results for the algorithms introduced by Sohn *et al*. are lifted from the original publication [42].

MVTec AD textures	NMI	ARI	F_1
Ours w/o VAE, CL	0.667	0.589	0.745
Ours w/o VAE	0.773	0.678	0.801
Ours w/o CL	0.694	0.605	0.743
Ours	**0.778**	**0.701**	**0.809**
MTD	NMI	ARI	F_1
Ours w/o VAE, CL	0.148	0.109	0.411
Ours w/o VAE	0.154	0.181	0.499
Ours w/o CL	0.201	0.253	0.560
Ours	**0.211**	**0.347**	**0.649**
Leaves	NMI	ARI	F_1
Ours w/o VAE, CL	0.456	0.426	0.635
Ours w/o VAE	0.568	0.561	0.735
Ours w/o CL	0.484	0.412	0.659
Ours	**0.689**	**0.704**	**0.801**

Table 6. Ablation results when using k-means for the final feature-level clustering.

MVTec AD textures	PRO	AUROC_p	AUROC_i
VAE + L^1	95.24	97.73	98.41
VAE + L^2	94.68	97.39	98.02
VAE + SSIM	96.73	98.73	99.77
VAE + Rec-grad	82.00	90.33	94.82
VAE + FCA (ours)	**97.50**	**99.02**	**99.93**

Table 7. Results for FCA ablation. The VAE feature reconstruction yields the best results when combined with FCA as the residual maps processor.

S6. Qualitative pixel-level results.

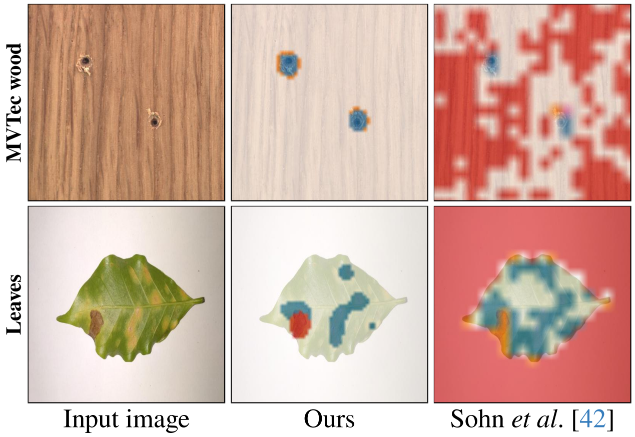

Figure 7. Qualitative comparison for pixel-level clustering on unseen images. We depict the predicted anomaly types with the following colors for wood: blue – hole, orange – scratch, red – color stain, and for leaves: blue – cercospora, orange – miner, red – phoma. In both cases white is used for pixels classified as *normal*.

Our contrastive training enables us to leverage the improved features for each pixel in the input image. In Figure 7, we show that one can use the features to perform clustering at a pixel level, essentially achieving multi-class anomaly segmentation/localization. For this experiment we use unseen images (not used for contrastive learning or training the VAE): an image from the *combined* subcategory of the wood texture in MVTec AD and a leaf texture that contains two different anomaly types. The wood texture initially had two types of anomalies; however, we cropped the image to include only one type (holes) because the other (knot) was not part of the fitting set (*i.e*., the subcategories: hole, scratch, color, liquid, and normal).

To create the pixel-level labeling for new images at inference time, we perform the following. The image-level descriptors of the fitting set are clustered using k-means to obtain the clusters' centers. Afterward, we apply the feature extractor F on the new images and apply the additional layers H trained using CL. These improved features are then compared with the image-level cluster centers and assigned the label corresponding to the closest point. We apply the analogous steps to compare with the weighted average method of Sohn *et al*. [42].

[1] This excludes the heuristic for finding the threshold t; see Section S4

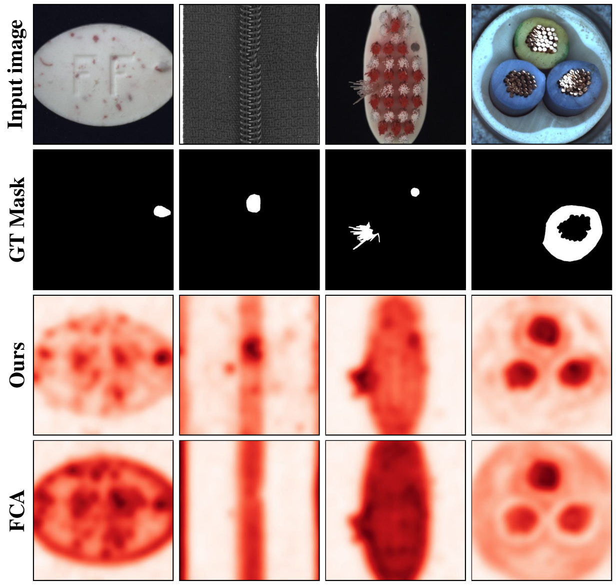

Figure 8. Qualitative comparison on MVTec AD object classes. The first three examples present structural defects that are reasonably well identified despite the non-homogeneity of the images. In the last column, there is a logical anomaly (wrong cable color) which our method fails to detect.

S7. Discussion on the application domain

As reflected in the MVTec AD dataset [10], there are two different classes of images on which anomaly detection is often applied: textures and generic objects. Depending on the class, anomalies tend to manifest themselves in very different ways; textures contain more subtle, structural anomalies, whereas objects have more semantic or logical anomalies. Structural anomalies, as found in textures, are better suited for zero-shot [4, 5] and blind anomaly detection [50]. Our blind anomaly localization algorithm is designed to work on textures by employing a zero-shot anomaly localization approach (FCA) as a subroutine. Our VAE-based improvement significantly enhances the zero-shot results, especially on more complex textures; however, we still inherit the stationarity assumption of the underlying zero-shot method. For completeness, we show results on a few objects from MVTec in Figure 8. Our blind anomaly localization decidedly improves upon the base zero-shot model. Nonetheless, the background acts as a distractor and diminishes the capability of the anomaly localization; our approach is also not suited to detect logical anomalies such as the cable swap (last column in Figure 8).

Importantly, our contrastive learning contribution is complementary to the method used for blind anomaly localization. If we assume the solution for generating the anomaly maps A_i is given, our feature fine-tuning method can be used to perform anomaly clustering for arbitrary types of images.

S8. Details on the evaluation of baselines

In this section, we describe in detail how we obtained the results for the prior work which we compare against in the main paper experiments.

For the main comparison of clustering performance in Table 1, we evaluate against seven methods. The results for SelFormaly [32] are taken from the original paper as no code is shared by the authors. We note that we outperform SelFormaly despite their having a supervisory advantage. The results for SCAN [44] and the approaches introduced by Sohn *et al*. [42] for MVTecAD textures and MTD are taken from [42], whereas the metrics for the Leaves dataset we compute ourselves. We evaluate SPICE [37] using the public official implementation shared by the authors. To ensure fairness, use the same WideResnet [48] feature extractor base for SCAN [44] and SPICE [37] as for our method. In order to evaluate STEGO, which is an unsupervised semantic segmentation method, we must design a mechanism to assign a single label per image. Firstly, we find the most frequently predicted (pixel-level) label and mark it as the normal class. We then use the ground truth number of normal samples (K) to the advantage of STEGO by taking the first K images with most pixels assigned to the normal class; these images are labeled as normal. For the rest of the dataset, we assign the image label as the most frequent non-normal label predicted by STEGO.

To evaluate blind anomaly localization (BAL), we compare against the following baselines. DRAEM [49] and CFA [31] as normality-supervised methods to analyze the performance when the training set has anomaly contamination ($\approx 75\%$, depending on the dataset). For convenience, we use the implementation from anomalib [1]. For ILTM [38] and the method of Zhang *et al*. [50], we use the results from the respective publications. ILTM was evaluated in a slightly different way, as the authors used the test set to make train, validation, and test splits and to control the ratio of anomalous samples. Nonetheless, as Zhang *et al*. report the results with 80% anomaly ratio, we argue that their evaluation methodology is largely comparable to ours and the results are representative. To compare with the anomaly maps generated by the weighted average method of Sohn *et al*. [42], we use our implementation of their paper. For this experiment, we run their method at a higher resolution (512×512) as opposed to the one suggested in the paper (256×256) since we observed that this increases the fidelity of anomaly localization.