Rethinking Benchmarking Framework of Self-Supervised Learning Approaches for Anomaly Localization

Tryambak Gangopadhyay
Amazon ML Solutions Lab
Amazon Web Services
tganguly@amazon.com

Sungmin Hong
Amazon ML Solutions Lab
Amazon Web Services
hsungmin@amazon.com

Sujoy Roy
Amazon ML Solutions Lab
Amazon Web Services
roysujo@amazon.com

Yash Shah
Amazon ML Solutions Lab
Amazon Web Services
syash@amazon.com

Lin Lee Cheong
Amazon ML Solutions Lab
Amazon Web Services
lcheong@amazon.com

Abstract

Localizing defects in products is a critical component of industrial pipelines in manufacturing, retail, and many other industries to ensure consistent delivery of the highest quality products. Automated anomaly localization systems leveraging computer vision have the potential to replace laborious and subjective manual inspection of products. Recently, there have been tremendous efforts in the domain of anomaly localization investigating self-supervised learning methods. However, despite the advancements, there is still a gap between research and deployment of those methods to real-world production environment. It is important to develop an industry-friendly benchmarking framework to understand the performance of models in a generalizable product-agnostic manner. We present a new anomaly localization benchmarking framework that maps a product/defect type combination to higher level descriptive abstractions capturing similar characteristics. We propose efficient training and inference schemes considering different aspects, including an ablation study of threshold estimation techniques. To the best of our knowledge, this is the first anomaly localization work on developing a benchmarking framework focusing on real-world use.

1 Introduction

Detecting defects from product images are of profound importance in industrial applications [1, 2, 3] to maintain high product quality and ensure cost efficiency. In an industrial production setting, manual inspection processes to detect defects can suffer from limited throughput and can be subjective with a much slower feedback loop. Automated visual defect detection methods provide the advantages of being fast and repeatable, have lower inspection costs and the feedback loop is faster compared to manual processes. Given these advantages, manufacturers are keen to leverage automated visual defect detection modules as part of quality check processes in production pipelines.

The task of anomaly localization involves assigning a pixel-level anomaly score to estimate a segmentation map highlighting subtle anomalous regions in an image. There are some major disadvantages of adopting a supervised approach for anomaly localization - 1) manual annotation of defective images is expensive, 2) all possible defect types need to be known beforehand, 3) only a limited number of defective images are available compared to the amount of defect-free

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

images as industrial production processes are optimized to minimize the number of defective samples. To alleviate these disadvantages, it is important to develop a benchmarking framework based on self-supervised approaches.

Most of the existing anomaly localization works [4, 5, 6, 7, 8] evaluate the efficacy of their proposed methods on individual product types from publicly available dataset like MVTec [9]. In such a dataset, the nature of defects are product-specific and therefore, from such evaluations, we only learn about how a proposed method works on a specific product or defect type and is hard to generalize learnings. Therefore, the proposed methods cannot be directly implemented for real-world use as those methods do not help a manufacturer coming with a different product having its own set of defect types. For example, most methods have utilized only threshold-independent metrics for performance evaluation. This is not useful during inference in a production setting where the goal is to generate the segmentation map by masking the non-anomalous regions.

To alleviate these problems, we develop a benchmarking framework that maps a product/defect type combination to higher level descriptive abstractions capturing similar characteristics. We summarize the contributions of our work here:

(1) We annotate anomalous product images in a product-agnostic manner to capture higher level human understandable descriptions.

(2) The benchmarking framework highlights the pros and cons of each individual method across product-agnostic categories and recommend a metric for practical use. These insights, alongside the recommended best practices, can provide guidance to practitioners on implementing a performant anomaly localization pipeline.

(3) We perform a detailed ablation study of threshold-determination techniques and suggest optimal solutions specific to broad product-agnostic categorization.

2 Proposed Benchmarking Framework

A typical anomaly localization pipeline consists of three phases - training, validation and inference. An illustration is provided in appendix (Fig. 1). In the training phase, anomaly-free images are used to train a model to learn representations from normal samples. The validation phase, requiring the availability of few anomalous samples, computes the optimal threshold level which is used in the inference phase to mask non-anomalous regions. The goal of the proposed benchmarking framework is to provide guidance for efficiently building an anomaly localization pipeline for practical use. The proposed benchmarking framework broadly consists of three building blocks - datasets, anomaly localization methods and evaluation approaches. In this section, we describe each building block.

2.1 Product-Agnostic Dataset Categorization

It is important to understand the performance of the anomaly localization methods over different products on a generalized setting beyond the product-specific analyses. We perform two types of product-agnostic dataset categorization.

Background Categories: We categorize each product image as with or without background. In industrial pipelines, image acquisition processes along with product size may decide what type of image will be input to the model. This background-type categorization is important as in practical scenarios, a varying background can play a significant role in influencing the performance of a model. For product images with background, there is the presence of a background, and, for product images without background, the product itself covers all the image pixels.

Defect Categories: We propose to categorize product images into four broad product-agnostic defect categories. This type of defect categorization allows us to compare the efficacy of different anomaly localization methods across products instead of being product specific. While most images can be labeled to the following defect categories, for some images, it is not straightforward to label and such corner cases can exist. For those corner case images, we label those into more than one label.

(1) Structural: Represented by distorted or missing object parts or some damage to the product structure. Generally, a structural defect is not a sublte defect and involves considerable damage. Some examples of structural defects are holes, bends, missing parts, etc.

(2) Surface: Restricted to smaller regions on the surface of the products requiring relatively lesser repair effort, and, unlike structural defects, not always make the product unusable. Some examples of surface defects are scratches, dents, iron rust, etc.

(3) Contamination: Contamination defects indicate the presence of some foreign material which are not part of the original product (normal image). Some examples of contamination defects are glue slip, dust, dirt, etc.

(4) Combined: Combination of the above three types of defects characterised by the presence of multiple connected components in the ground truth segmentation map - for example, a hole in a contaminated background.

We utilize two publicly available datasets (MVTec [10], BTAD [11]) to generate a product-agnostic dataset.

MVTec Anomaly Detection Dataset (MVTec): MVTec is a comprehensive multi-object dataset providing pixel-precise ground truth segmentation maps [10, 9]. The MVTec dataset comprises about 5.3k high-resolution color images with 15 product types.

BeanTech Anomaly Detection Dataset (BTAD): BTAD [11] is a public dataset consisting of about 1.8k high resolution images of three industrial products. BTAD has the same setting as MVTec.

Annotation: For each anomalous sample in MVTec and BTAD, considering the ground truth segmentation map, the actual anomalous image and its corresponding normal product image are compared to label that sample to a defect category. A team of annotators worked on this and the entire labeling effort was manual. By utilizing a custom built UI tool, an annotator could compare the image and map in the tool itself to select the appropriate defect category. The tool ensured that the labeling process is efficient, accurate with faster feedback loop.

Experimental Protocol: Also, any benchmarking dataset should comprise distinct training, validation and test datasets for each product. In existing publicly available datasets [9, 11], a validation dataset is not explicitly identified. Estimating an optimal threshold based only on anomaly-free images does not work well [10]. Having a validation set, comprising a set of anomalous images, is necessary for optimal threshold estimation. For each product of MVTec and BTAD, the training set comprises defect-free images and the test set consists of both anomalous and non-anomalous images. For the anomalous images, pixel-precise ground truth segmentation maps are provided. We take out few examples from each test set with a split of 0.3 to create the validation set.

2.2 Anomaly Localization Method Categorization

Previous works have proposed self-supervised anomaly localization methods leveraging different architectures. The SSL methods learn the distribution of normal images and estimate anomaly in an anomalous image from the learned distribution. Depending on how an anomaly score map is generated, we can categorize the methods into four broad categories: reconstruction based [12, 13, 14, 15, 4], attribution map based [16, 17, 5], patch similarity based [6, 18, 19, 20, 21], and normalizing flow based [22, 23, 7]. Based on the reported state-of-the-art performance of existing works, we select a representative approach from each category:

(1) Reconstruction Based - Autoencoder (AE): Based on [4] utilizing SSIM as the loss. We train the model so that it can learn to reconstruct an anomaly-free image as closely as possible. During inference, an anomaly score map is generated based on the pixel-wise SSIM loss between test image and (defect-free) reconstructed version.

(2) Attribution Map Based - Knowledge Distillation (KD): In KD [5], training is done using a student-teacher approach to make the student network learn about normal images. During inference, the discrepancy between the teacher and student networks' intermediate activations is used to localize the anomalies.

(3) Patch Similarity Based - Patch Distribution Model (PaDiM): PaDiM [6] learns a representation of the normal class using multivariate Gaussian distributions and correlations between different semantic levels of a pretrained CNN. During inference, Mahalanobis distance is computed between the predicted embedding and the learned distribution to compute patch-wise anomaly scores.

(4) Normalizing Flow Based - Conditional Normalization Flow (CFLOW): CFLOW [7] models the distribution of normal images during training by leveraging a pre-trained encoder and a CFLOW decoder. During inference, the probability maps, estimated at different scales, are rescaled and aggregated to estimate the final anomaly score map.

2.3 Evaluation Approach

Most of the previous works have reported results using threshold-independent metrics like AUROC which is not ideal to estimate localization performance used to drive decision-making. These metrics can only be used for validation and not for inference in a production setting. In high-volume manufacturing applications, ROC curve (plotting FPR Vs TPR) is not be a reliable metric especially when FPR values are low (resulting from high true negative) for subtle defects. In this work, we focus on an inference-ready (threshold-dependent) metric for evaluation. Intersection Over Union (IoU) estimates the amount of overlap between the predicted and ground truth segmentation maps. The IoU metric is not dependent on True Negative (TN) and can therefore be a more suitable metric to evaluate localization performance.

2.4 Threshold Estimation

We propose to estimate the thresholds using three different techniques.

ROC-T: In ROC curve, different threshold values are used to compute TPR and FPR. The threshold which gives the maximum geometric mean of TPR and $1 - FPR$ is considered to be the optimal one maximizing TPR and minimizing FPR. The optimal threshold finds the balance between FPR and TPR.

IoU-T: In the IoU curve, we plot FPR on the x-axis and IoU on the y-axis for different threshold values. The optimal threshold is determined by maximizing the geometric mean of IoU and $1-FPR$. It tries to maximize IoU and simultaneously ensure that FPR is not high.

PR-T: In Precision Recall (PR) curve, we plot recall and precision on the x-axis and y-axis respectively. To find a balance of precision and recall, F1 Score can be maximized. For each threshold value, F1 scores are computed and the threshold corresponding to the maximum F1 score is considered to be the optimal one.

Illustrations of threshold estimation are provided in appendix (Fig. 4).

Table 1: Product-Agnostic performance in terms of IoU for test sets of MVTec and BTAD datasets.

		AE			KD			PaDiM			CFLOW		
Category	Dataset	ROC-T	IoU-T	PR-T	ROC-T	IoU-T	PR-T	ROC-T	IoU-T	PR-T	ROC-T	IoU-T	PR-T
Defect Categories													
Structural	MVTec	0.071	0.099	0.100	0.207	0.257	0.254	0.254	**0.345**	0.314	0.199	0.277	0.213
	BTAD	0.177	0.190	0.184	0.227	0.229	0.214	0.225	0.188	**0.243**	0.074	0.110	0.067
Surface	MVTec	0.048	0.077	0.074	0.190	0.291	0.295	0.177	**0.321**	0.281	0.210	0.306	0.253
	BTAD	0.118	0.130	0.124	0.028	0.038	0.042	0.058	0.013	0.023	0.204	**0.224**	0.171
Contami-nation	MVTec	0.054	0.067	0.075	0.224	0.298	0.294	0.247	**0.314**	0.273	0.204	0.255	0.201
	BTAD	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Combined	MVTec	0.078	0.126	0.116	0.215	0.244	0.223	0.249	**0.331**	0.304	0.257	0.271	0.155
	BTAD	0.192	0.216	0.206	0.316	0.306	0.283	**0.318**	0.186	0.220	0.114	0.104	0.088
Background Categories													
With Background	MVTec	0.073	0.098	0.104	0.200	0.257	0.256	0.216	**0.316**	0.293	0.166	0.277	0.199
	BTAD	0.062	0.071	0.068	0.169	0.199	0.200	0.190	0.149	**0.241**	0.071	0.098	0.059
Without Background	MVTec	0.041	0.073	0.062	0.220	0.303	0.297	0.262	**0.367**	0.311	0.271	0.294	0.257
	BTAD	0.216	0.236	0.227	0.077	0.072	0.066	0.098	0.041	0.034	0.271	**0.287**	0.220

3 Results

After training each model product-wise, the trained model and validation set of each product are utilized to compute the optimal threshold values. The models are trained using one Tesla K80 GPU with the default hyper-parameters suggested for the modeling approaches [4, 5, 6, 7]. In the validation phase, for each product-model combination, we compute three threshold values which are utilized during inference to compute IoU. Product types can vary across datasets - so it is important to summarize the performance in terms of product-agnostic categories. Table 1 shows the comparison

of four methods using three threshold values in terms of IoU metric. From Table 1, we observe that PaDiM demonstrates better overall performance than the other models. For structural, surface and combined defect categories of MVTec, we find that PaDiM is followed by CLFOW and KD. And, for the contamination category, PaDiM still performs better followed by KD and CFLOW. From BTAD, no presence of contamination defect is detected during annotation. For structural and combined defect categories of BTAD, PaDiM shows the best IoU scores, but for surface defects, CFLOW outperforms the other models. From the ablation study, we observe that using IoU curve is the most efficient threshold determination approach followed by threshold from PR curve.

For images with background, PaDiM is the best performing model for both MVTec and BTAD. When the product images don't have any external background, CFLOW performs better for BTAD. We also notice that for MVTec product images without background, PaDiM outperforms KD, which is followed closely by CFLOW.

4 Conclusion

Through product-agnostic evaluation, our analysis enables a more generalized way of comparing model performance. Overall, PaDiM demonstrates better performance than the other three models. We recommend that using PaDiM and estimating threshold from IoU curve is the most efficient way to start with for a manufacturer coming with a new product. While KD and CFLOW also perform well, comparatively, AE fails to accurately detect anomalous regions in most experiments. If computational resource is a constraint, we recommend using KD which shows faster inference speeds in our experiments. The insights gained from developing this framework can help practitioners in deploying automated anomaly localization in industrial pipelines. It will also encourage other researchers to perform new experiments using self-supervised learning approaches in similar directions. In future, we plan to include other datasets to build on the existing defect categories by adding new defect types.

References

[1] Z. Ren, F. Fang, N. Yan, and Y. Wu. State of the art in defect detection based on machine vision. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 2021.

[2] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang. Using deep learning to detect defects in manufacturing A comprehensive survey and current challenges. *materials*, 2020.

[3] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021.

[4] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

[5] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021.

[6] 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.

[7] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022.

[8] 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.

[9] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

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

[11] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021.

[12] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer, 2018.

[13] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019.

[14] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019.

[15] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI brainlesion workshop*, pages 161–169. Springer, 2018.

[16] David Zimmerer, Jens Petersen, Simon AA Kohl, and Klaus H Maier-Hein. A case for the score: Identifying image anomalies using variational autoencoder gradients. *arXiv preprint arXiv:1912.00003*, 2019.

[17] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. *arXiv preprint arXiv:2002.03734*, 2020.

[18] Chin-Chia Tsai, Tsung-Hsuan Wu, and Shang-Hong Lai. Multi-scale patch-based representation learning for image anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3992–4000, 2022.

[19] Kaitai Zhang, Bin Wang, and C-C Jay Kuo. Pedenet: Image anomaly localization via patch embedding and density estimation. *Pattern Recognition Letters*, 153:144–150, 2022.

[20] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *arXiv preprint arXiv:2206.04325*, 2022.

[21] 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.

[22] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[23] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please check Section 3.
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We have included information to reproduce the results.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

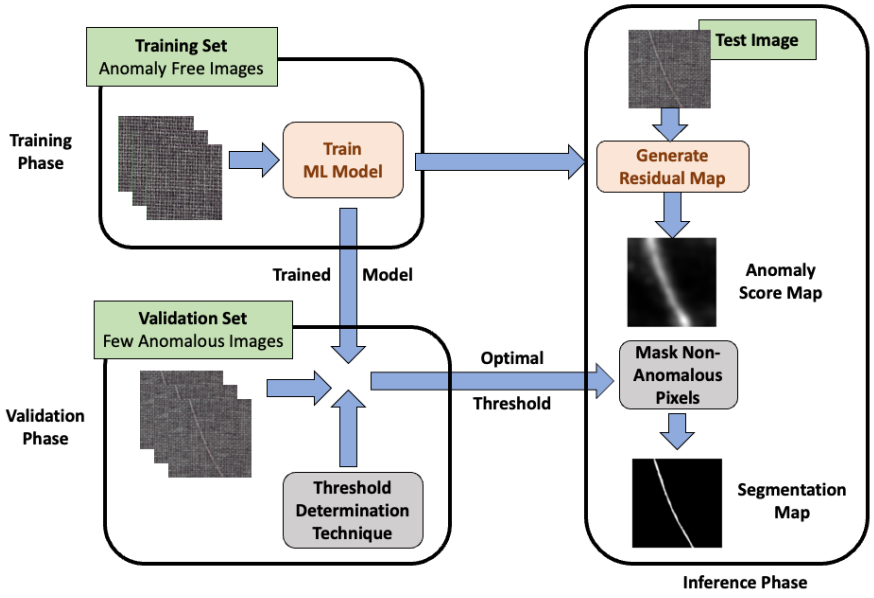


Figure 1: Illustration of different phases of an anomaly localization pipeline. While the training phase needs only anomaly-free images, validation phase requires only very few samples of anomalous images. The optimal threshold determined during the validation phase is utilized to generate the segmentation map from anomaly score map in the inference phase.

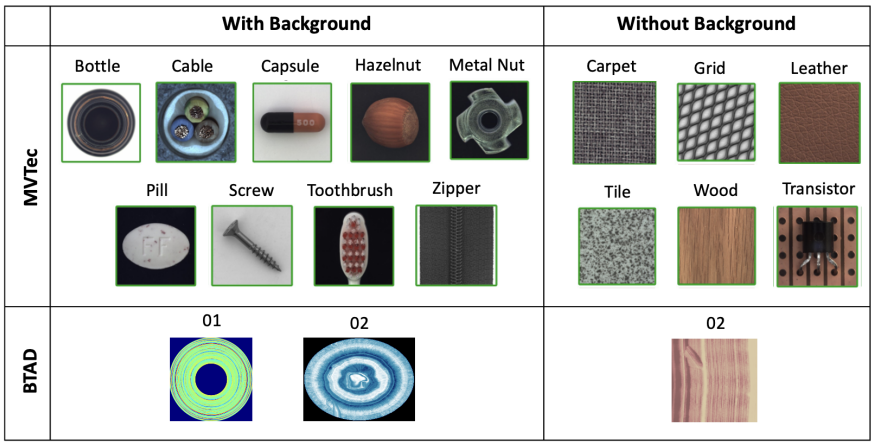


Figure 2: Background Categories. Sample images from MVTec [10, 9] and BTAD [11] datasets.

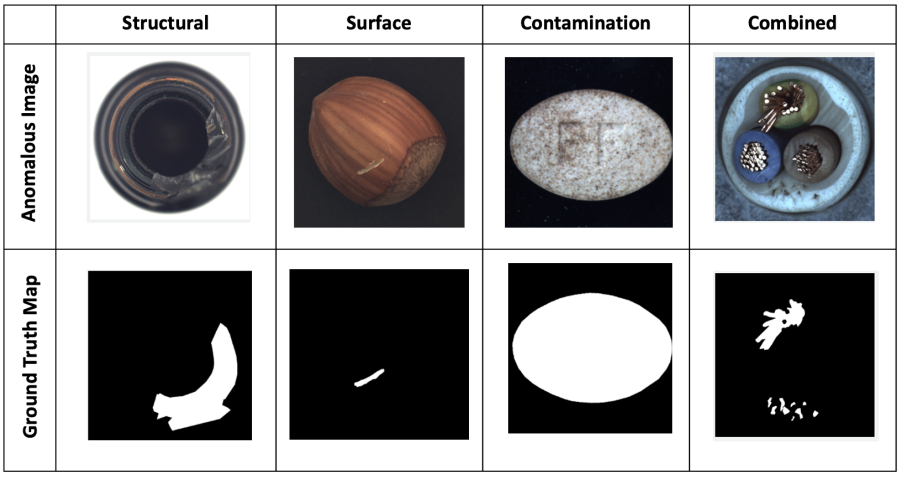


Figure 3: Defect Categories. Sample labeled anomalous images from MVTec [10, 9] dataset.

Table 2: Annotation details for defect categories of each product from MVTec and BTAD. To determine optimal threshold levels, for each product, we select a validation set (30%) from the total number of anomalous samples shown in this table.

Dataset	Product	Structural	Surface	Contamination	Combined	Total
MVTec	Bottle	41	0	21	1	63
	Cable	81	0	0	11	92
	Capsule	50	62	0	0	112
	Carpet	34	17	37	1	89
	Grid	25	0	32	0	57
	Hazelnut	35	35	0	0	70
	Leather	37	37	18	0	92
	Metal Nut	46	43	0	4	93
	Pill	28	46	50	17	141
	Screw	65	40	0	18	123
	Tile	33	15	36	0	84
	Toothbrush	20	1	3	6	30
	Transistor	37	3	0	0	40
	Wood	11	28	10	11	60
	Zipper	69	33	0	17	119
BTAD	01	24	12	0	13	49
	02	30	135	0	35	200
	03	33	4	1	3	41
Including Both		699	511	208	137	1555

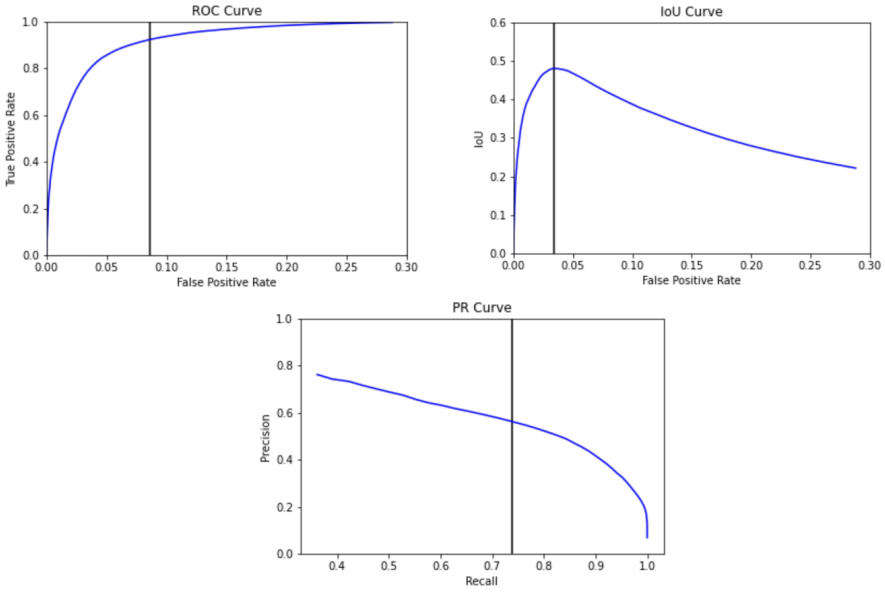


Figure 4: Illustrations of threshold estimation for MVTec product Bottle using model KD. In the ROC and IoU curves, the vertical lines show the FPR values corresponding to the optimal thresholds. In the PR curve, the vertical line highlights the recall value corresponding to the optimal threshold.

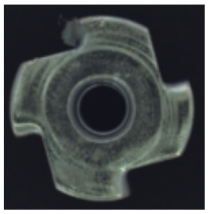
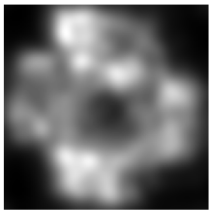
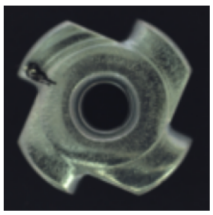
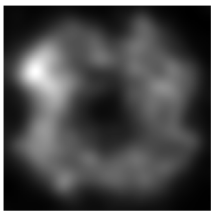
Test Image	Ground Truth Segmentation Map	Predicted Anomaly Score Map	Predicted Segmentation Map	Metrics
				IoU 0.195 F1 0.353 FPR 0.012
				IoU 0.335 F1 0.379 FPR 0.026

Figure 5: Segmentation map results for two test images from MVTec product Metal Nut with contrasting performances in terms of IoU. The IoU scores for the top and bottom row are 0.195 and 0.335 respectively. Though the F1 values are very close for both rows and False Positve Rate (FPR) rather increases for the bottom row, IoU value correctly indicates that for the bottom row the performance is actually better. This shows that IoU is a more reliable metric.