Benchmarking Framework for Anomaly Localization: Towards Real-World Deployment of Automated Visual Inspection

Tryambak Gangopadhyay[a,*], Sungmin Hong[a], Sujoy Roy[a], Yash Shah[a], Lin Lee Cheong[a,*]

[a]*Amazon ML Solutions Lab, Amazon Web Services*

Abstract

Localizing defects in products is a critical component of industrial pipelines in manufacturing, retail, and many other industries to ensure consistent delivery of high 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 this research domain investigating deep learning-based anomaly localization methods. However, such proposed methods, mainly considering product-specific evaluation, cannot be directly implemented for real-world use. Therefore, despite the advancements, there is still a gap between research and deployment of those methods to real-world production environments. Implementing any automated solution for manufacturing can involve a steep upfront investment. It is important to develop an industry-friendly benchmarking framework to ensure the feasibility and robustness of an automated quality control system. In this paper, we present a new anomaly localization benchmarking framework considering different aspects - 1) understand the performance of models in a generalizable product-agnostic manner, 2) explore pros and cons to find the most optimal modeling approach, 3) develop an efficient training and inference scheme with defect-free training samples and very few defective samples for evaluation, and 4) perform an ablation study of threshold estimation techniques to determine optimal threshold level for segmentation. We release a newly-labeled dataset for the research community with product-agnostic categorization of defective product images. To the best of our knowledge, this is the first anomaly localization work on developing a benchmarking framework focusing on real-world use. We believe domain experts from different industries will find this useful and can gain valuable insights to deploy automated visual inspection in production pipelines.

Keywords: anomaly, localization, deep learning, threshold

1. Introduction

Detecting defects from product images is of profound importance in manufacturing [1, 2] to maintain high product quality and ensure cost efficiency. Once trained in recognizing what a defect-free product sample looks like, humans are generally good in these visual inspection tasks and are flexible in quickly adapting to detect different defect types. However, in industrial production, manual inspection processes can suffer from limited throughput and can be subjective with a much slower feedback loop. Though automated visual defect detection can have high upfront costs, it provides the advantages of being fast and repeatable, has lower inspection costs and the feedback loop is faster.

Given these advantages, several manufacturers are leveraging automated visual defect detection modules as part of quality check processes in production pipelines. One of the primary challenges faced by the machine vision modules is that it is difficult to consistently categorize a defect as it can come in different forms and shapes - hence can be described, at best, as an "anomaly". Anomaly detection refers to computing an anomaly score of each image for image-level classification that predicts an image is whether anomaly-free or anomalous [3]. Anomaly localization is a more complex task that involves assigning a pixel-level anomaly score to estimate a segmentation map highlighting subtle anomalous regions in an image. Anomalies can be located by either putting a bounding box [4] or classifying every pixel as anomalous or non-anomalous [5]. While the former approach is useful if it satisfies industrial production goals, the segmentation mask estimation provides pixel-wise precision of an anomalous area that enables a system to localize subtle defects. We focus on modeling approaches that can generate pixel-precise anomaly segmentation maps by masking the defect-free regions.

While both supervised and unsupervised approaches can be used for anomaly localization, there are some major disadvantages of adopting a supervised approach: 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 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 or weakly-supervised approaches. Semi-supervised approaches have been proposed for anomaly localization to efficiently compute anomaly score

*To whom correspondence may be addressed. Email: tganguly@amazon.com, lcheong@amazon.com

maps for anomalous images after being trained on defect-free images [6, 7, 8, 9].

Most of the existing anomaly localization works evaluate the efficacy of their proposed methods on individual product types from some publicly available datasets like MVTec [10]. 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. This does not help a manufacturer coming with a different product having its own set of defect types. It is also not clear what we are learning across product types and what specific characteristics of a product/defect type are suitable to be analyzed by such proposed machine vision approaches. Therefore, the proposed methods cannot be directly implemented for real-world use. For performance evaluation, most methods have utilized only threshold-independent metrics. But, this is not useful in inference scenarios, when the goal is to generate the segmentation map by masking the non-anomalous regions. Therefore, determining an optimal threshold is crucial to make a framework ready for practical use.

To alleviate these problems, we develop a benchmarking framework that maps a product/defect type combination to higher level descriptive abstractions capturing similar characteristics. The framework defines and demonstrates the characteristics of a dataset that would help us to gain insights from different anomaly localization methods. The learnings about pros and cons can be applied to a new use case or dataset. To the best of our knowledge, this is the first anomaly localization work focusing on real-world use in industrial production. Users can utilize this benchmarking framework to integrate an automated anomaly inspection process to their pipelines. Through the proposed benchmark, this work makes the following contributions:

1. We make available new annotations of anomalous product images that capture higher-level human understandable descriptions. The newly labeled versions of the datasets can enable researchers implement new experiments in this direction.

2. We highlight the disadvantages of some evaluation metrics commonly used in existing anomaly localization works and recommend metrics that are useful from the perspective of practical use. The benchmarking framework also bring out the pros and cons of each individual method and this information can guide the practitioners during implementation.

3. We focus on the importance of determining an optimal threshold value for anomaly localization. We perform a detailed ablation study of threshold-determination techniques and suggest optimal solutions specific to broad product-agnostic categorization.

4. We demonstrate how a manufacturer, with a new product having its unique set of defect types, can utilize this benchmarking framework to figure out what works best for their use case.

The paper is organized into three sections. First, we focus on the modeling approaches and datasets. Thereafter, Section 3 presents the threshold determination techniques and is followed by conclusion.

2. Proposed Benchmarking Framework Formulation

An overview of different phases of an anomaly localization pipeline is shown in Fig. 1. The training phase requires non-anomalous examples to train a machine learning model which does not learn about anomalous samples. In this work, we investigate different types of modeling approaches for the training phase. 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. As this validation phase is very rarely used in existing works, it is difficult to accurately replicate previous experiments. Also, existing datasets do not make this validation dataset explicitly available. In the inference phase, an anomaly score map is first generated to highlight the anomalous regions and thereafter, the segmentation map is computed by masking the non-anomalous pixels. The goal of the proposed benchmarking framework is to provide guidance regarding building an anomaly localization pipeline comprising three different phases to localize anomalies from images of novel products.

The proposed benchmarking framework broadly consists of three building blocks - datasets, anomaly localization methods and evaluation approaches. In this section, we dive deep into each of the building blocks. We present product-agnostic dataset categorization, anomaly localization method categorization and take an in-depth look at evaluation techniques that are more relevant for practical use.

2.1. Product-Agnostic Dataset Categorization

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

2.1.1. Background Categories

We categorize the product image types as with or without a background. In practical scenarios, a varying background can play a significant role in influencing the performance of an anomaly localization model.

With Background: For product images with a background, there is the presence of a background and in some cases, the product may constitute less number of pixels compared to the background. In many real use cases, the background can vary for the same foreground product.

Without Background: For product images without a background, there is no presence of an external background and the product itself covers all the image pixels. This is applicable for products like a carpet, a grid, a wooden floor, etc when the product itself is the background.

Sample images from the two background categories are shown in Fig. 2.

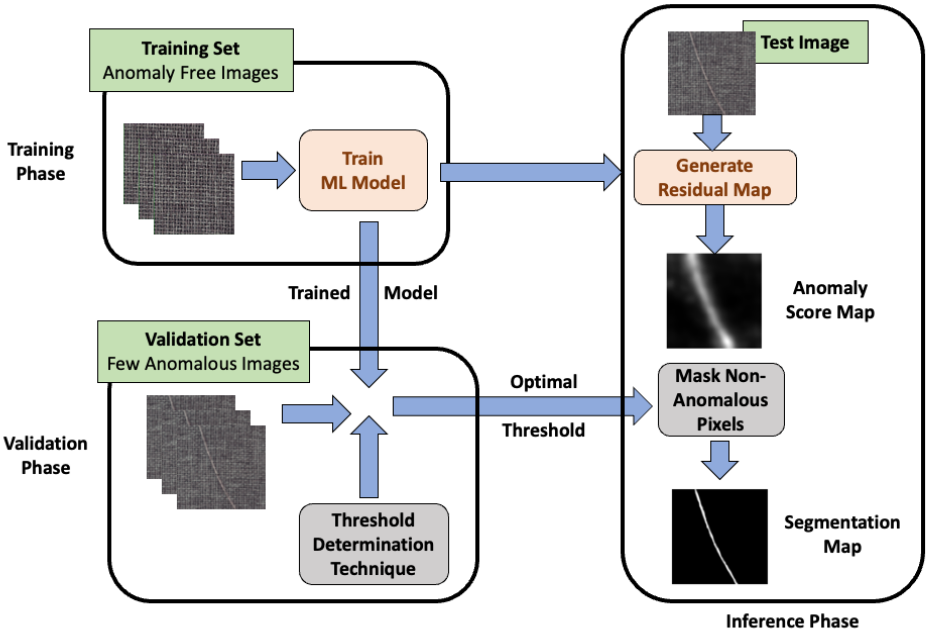

Figure 1: Illustration of different phases of an anomaly localization pipeline. While the training phase needs only anomaly-free images, the validation phase requires only very few samples of anomalous images. The optimal threshold determined during the validation phase is utilized to generate the segmentation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

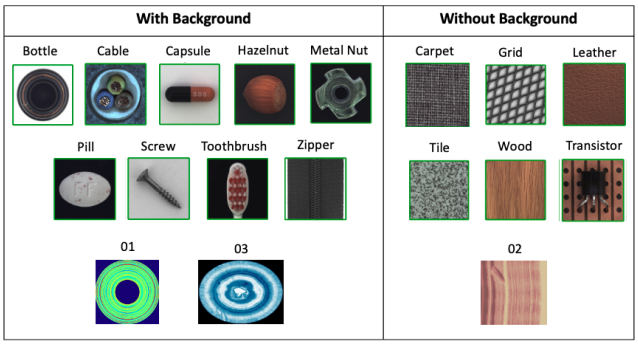

Figure 2: Background Categories. Sample images from MVTec [11, 10] and BTAD [12] datasets.

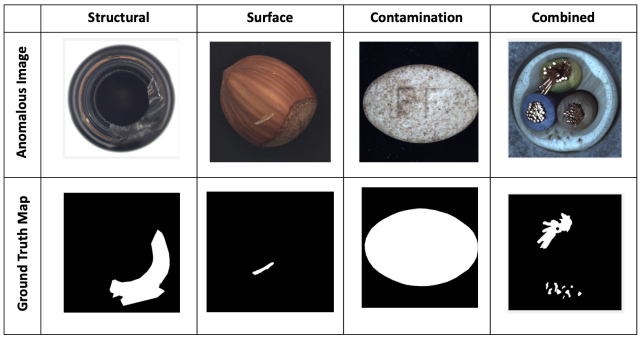

Figure 3: Defect Categories. Sample labeled anomalous images from MVTec [11, 10] dataset.

2.1.2. *Defect Categories*

We propose to categorize product images into four broad product-agnostic defect categories, namely, (1) structural, (2) surface, (3) contamination, and (4) combined. This type of defect categorization allows us to compare the efficacy of different anomaly localization methods across products instead of being product specific. Thereafter, for a new product, with similar defect categorization, the analysis can help us to understand which method works better.

Structural: Structural defects are represented by distorted or missing object parts or some damage to the product structure. In Fig. 3 an example of structural defect from a product is shown - we observe that a large part of the product is broken here. Generally, a structural defect is not a sublte defect, rather it involves considerable damage to the product structure. Some examples of structural defects are holes, bends, missing parts, etc.

Surface: Surface defects are mostly restricted to smaller regions on the surface of the products. These defects require relatively lesser repair effort, and, unlike structural defects, they do not always make the product unusable. However, for some refined products like wafer (a thin slice of semiconductor), surface defects can render those products useless. Some examples of surface defects are scratches, dents, iron rust, etc.

Contamination: Contamination defects indicate the presence of some foreign material that is not part of the original product (normal image). In general, fixing these defects don't require much effort and can happen at various scales. Some examples of contamination defects are glue slip, dust, dirt, etc.

Combined: In several cases, defects can be a combination of

the above three types of defects. In such scenarios, it is hard to clearly categorize an anomalous image into a specific defect category. A combined defect can be characterised by the presence of multiple connected components in the ground truth segmentation map - for example, a hole in a contaminated background.

The defect categories, to some extent, highlight the degree of a damage and the complexity of a solution/repair to fix a defect. In general, for a solid object, structural defects are the hardest to fix, followed by surface and contamination defects, respectively.

Apart from product-agnostic categorization, we highlight another important aspect of dataset in anomaly localization benchmarking framework. Any benchmarking dataset should comprise distinct training, validation and test datasets for each product. In existing publicly available datasets [10, 12], a validation dataset is not explicitly identified. A method may take a cross-validation approach by splitting the training data into training and validation sets. However, when the training images are all normal images, estimating an optimal threshold based only on anomaly-free images will not work well [11]. Having a validation set, comprising a set of anomalous images, is necessary for optimal threshold estimation.

2.2. Anomaly Localization Method Categorization

As illustrated in Fig. 1, the training phase in an anomaly localization pipeline involves training a machine learning model using anomaly-free images. The trained model is utilized to generate anomaly score map in the inference phase. Previous works have proposed anomaly localization methods leveraging different architectures and theories on modeling the distribution of normal images and estimating anomaly from the learned distribution. Depending on how an anomaly score map is generated, we can categorize the methods into four broad categories: reconstruction based [6], attribution map based [7], patch similarity based [8] and normalizing flow based [9].

Reconstruction-based: Approaches based on reconstruction error are essentially trained on defect-free images and generate an anomaly score map based on the differences between test image and (defect-free) reconstructed version. The underlying mechanism is that these methods are trained to learn to generate a defect-free image and if the test image has any localized defects it should be highlighted in the difference. Generating a reconstructed version of a test image can be done using modified GANs [13, 14, 15] and different autoencoder variants [16, 17]. These models assimilate global knowledge of all pixels in image and hence, are not expected to work well for subtle defects. Using variational autoencoders [16] instead of deterministic convolutional autoencoders doesn't significantly improve unsupervised segmentation performance [18].

Attribution-Map-based: Attribution-map-based frameworks [19, 20, 7] utilize gradients to explain predictions of deep learning models. These methods are based on ideas from path attribution methods like integrated gradients [21] which use gradient-based ways to explain a model's predictions in terms of features. The intuition is to attribute the output loss to input pixels and the loss is an estimation of deviation from normal. Although these approaches seem intuitive, in some scenarios, these methods can end up attributing the loss to a large portion of the entire product image, especially when the product is the background or when the defect is camouflaged with the background.

Patch-Similarity-based: For localization approaches based on patch similarity, first, distributions are learned from defect-free patches and thereafter, during inference, some patch-wise statistics of test image are compared with the learned distributions to predict if a patch is anomalous or not. These approaches follow the principle of learning a distribution for representations of defect-free patches and finding a robust way to detect outliers based on similarity matching. There are issues concerning how well these methods can scale for a large collection of patches and whether the methods can miss out on a global perspective. But, several existing works [8, 5, 22, 23, 24] have proposed methods which can work around those issues and have reported good results.

Normalizing-Flow-based: Normalizing flows are networks capable of learning complex transformations between data distributions and probability densities [25]. A valid probability distribution is obtained after the initial density flows through a sequence of invertible mappings. This type of neural network was proposed to alleviate the problems of GANs [26] and VAEs [16] by implementing invertible functions. Features extracted by convolutional neural networks can be leveraged to estimate density using normalizing flows. To adapt normalizing flows for low dimensional data, DifferNet uses a multi-scale feature extractor [27]. CFLOW-AD [9], a model based on conditional normalizing flow, extends DifferNet to achieve pixel-level anomaly localization.

2.3. Evaluation Approach

To properly evaluate the performance of different approaches, correctly choosing a set of metrics is important. The previous works in this domain have used a wide variety of metrics, some of which may not be ideal to estimate the localization performance. Most of the previous works have reported localization performance using the AUROC (Area Under Receiver Operating Characteristic Curve) metric. In certain scenarios, ROC curve (plotting FPR Vs TPR) may not be a reliable metric. For example, if the defect is a subtle defect (with wide spread of masked region in the segmentation map), False Positive (FP) can be low with a high True Negative (TN) leading to low False Positve Rate (FPR) value. The AUROC value will not be a good estimate of the performance in such a scenario. We discuss more about choosing the right metrics in later sections. Also, a threshold-independent metric like AU-ROC can only be used for validation and not for inference in a production setting.

Thinking from the perspective of practical use, we believe that evaluation should be two fold in these localization systems - validation and inference metrics. While the validation metrics don't require a threshold value, the computation of inference metrics need an optimal threshold. In other words, the validation and inference metrics can be referred to

as threshold-independent and threshold-dependent metrics, respectively. Also, there needs to be a common consensus about which metrics are most optimal for practitioners to use in industrial production. To ensure that metrics can properly assess the performance, validation and inference metrics are accordingly selected in our benchmarking framework.

2.3.1. Validation Metrics

AU-ROC: The Receiver Operating Characteristic (ROC) curve is one of the most widely used evaluation tools for classification and segmentation. The ROC curve is plotted by the True Positive Rate (TPR) against the False Positive Rate (FPR) with varying discrimination thresholds [28]. The Area Under ROC curve (AU-ROC) measures the probability of positive samples being ranked higher than negative samples. For binary segmentation, the AU-ROC can be estimated as follows [29],

$$AU\text{-}ROC = \frac{\sum_{t_0 \in \mathcal{D}^0} \sum_{t_1 \in \mathcal{D}^1} \mathbf{1}[f(t_0) < f(t_1)]}{|\mathcal{D}^0||\mathcal{D}^1|}, \quad (1)$$

where f, $\mathcal{D}^0$, $\mathcal{D}^1$, and $\mathbf{1}[\cdot]$ are the predictor function that returns the predicted probability, the negative sample set, the positive sample set, and the indicator function, respectively.

AU-IoU: The Intersection over Union (IoU) score is defined as follows,

$$IoU(t) = \frac{TP(t)}{TP(t) + FP(t) + FN(t)}, \quad (2)$$

where t, TP, FP, and FN are the threshold, false positives, and false negatives, respectively. Compared to the F1 score, it penalizes the false predictions more. The IoU curve plots the IoU against the FPR which shows the trade-off between the false positives and IoU over varying thresholds. The Area Under IoU curve (AU-IoU) score is the trapezoidal area of the IoU curve,

$$AU\text{-}IoU = auc_{tr}(IoU\text{-}FPR(t)) \approx \int_t IoU\text{-}FPR(t)dt, \quad (3)$$

where $IoU\text{-}FPR(t)$ is the point on the IoU curve with corresponding IoU and FPR at the threshold t.

AU-PR: The Precision-Recall (PR) curve shows precision, $Precision(t) = \frac{TP(t)}{PP(t)}$, against recall, $Recall(t) = \frac{TP(t)}{P}$ with varying thresholds t, where TP, PP, and P are true positives, predicted positives, and positives, respectively. It is widely used when there is class imbalance between positive and negative samples because the AU-ROC may not represent the quality of a predictor well with severe class imbalance. In practice, the Area Under PR curve (AU-PR) is computed by the trapezoidal area of the PR curve because the standard linear interpolation overestimates the AU-PR [30],

$$AU\text{-}PR = auc_{tr}(PR(t)) \approx \int_t PR(t)dt, \quad (4)$$

where τ, auc_{tr}, and $PR(t)$ is a prefixed threshold interval, the trapezoidal area function, and the Precision-Recall point at the threshold t.

AU-PRO: Per-region overlap (PRO) measures the region-wise TPR [31, 32, 10]. It is often used for the evaluation of anomaly localization where subregion-wise overlap is more important than the pixel-wise overlap to identify the anomalous areas [10].

$$PRO(t) = \frac{1}{N_c} \sum_i \sum_k \frac{|PP_i(t) \cap C_{i,k}|}{|C_{i,k}|}, \quad (5)$$

where $C_{i,k}$ is the k^{th} connected component in the i^{th} ground truth segmentation label. The PRO curve plots the PRO against the FPR with varying thresholds, t. It can be viewed as the region-level ROC curve. The Area Under PRO curve (AU-PRO) is calculated similarly to the AU-PR and AU-IoU by calculating the trapezoidal area of the PRO curve,

$$AU\text{-}PRO = auc_{tr}(PRO\text{-}FPR(t)) \approx \int_t PRO\text{-}FPR(t)dt, \quad (6)$$

where $PRO\text{-}FPR(t)$ is the point on the PRO curve with corresponding PRO and FPR at the threshold t.

In anomaly detection datasets, only a small percentage of all pixels are generally labeled as anomalous. The amount of non-anomalous pixels is much higher. In industrial pipelines, large false positive rate (FPR) can lead to a large amount of non-anomalous products to be wrongly rejected after being detected as anomalous. Previous works reported that computing the area under curves up to a certain limit (30%) of FPR can be more appropriate [32, 10, 12, 8, 33, 5]. For AU-ROC, AU-IoU and AU-PRO, we compute area up to 30% FPR and then normalize the resulting area to ensure that maximum value is 1. For AU-PR, the entire area is computed as PR curve does not involve FPR.

2.3.2. Inference Metrics

IoU: Intersection Over Union (IoU) estimates the amount of overlap between the predicted and ground truth segmentation maps. The IoU metric (Eq. 2) is not dependent on True Negative (TN) and can therefore be a more suitable metric to evaluate localization performance.

F1 Score: F1 Score, designed to handle data imbalances, is based on precision and recall. Precision is the ratio between True Positive (TP) pixels and number of pixels predicted as positive. Here, positive refer to anomalous pixels. High precision indicates that a model has low False Positive (FP) value - the pixels predicted as anomalous are very likely to be actually anomalous. Recall is the ratio between TP and actual number of positive pixels. High recall means that the model mostly succeeds in finding the anomalous regions. F1 Score, harmonic mean of precision and recall, is a metric which finds a balance between precision and recall. High value of F1 Score implies high values of both precision and recall.

FPR: False positive refers to predicting the positive event (anomaly) falsely when there is no event (anomaly). False positive rate (FPR), also called false alarm, summarizes how often positive (or anomaly) is predicted when the actual is negative (or non-anomalous). It is the ratio between the predicted number of false positive (anomalous) pixels and the actual number of negative (non-anomalous) pixels. In other words, it quantifies the percentage of non-anomalous pixels falsely predicted as anomalous.

Table 1: Annotation details for defect categories of each product from MVTec and BTAD.

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

Table 2: Number of defective samples for each boundary category. Products from MVTec and BTAD.

Dataset	With Background	Without Background
MVTec	836	422
BTAD	90	200
Including Both	926	622

PRO: As mentioned before, Per-Region-Overlap (PRO) is a region-level metric. The ground truth segmentation map is segregated into connected components and for each connected component, TPR is computed. TPR values across the connected components are averaged to get PRO metric. The advantage of PRO is that it gives equal importance to each anomalous region irrespective of whether it is large or subtle defect. For this reason PRO can be a more appropriate threshold-dependent metric than TPR but PRO can also suffer from some disadvantages as with TPR when TP is high and FN is low for subtle defects in an image.

3. Experimental Setup

3.1. Generating a Product-Agnostic Dataset

We described our proposed product-agnostic categorizations in Section 2.1. We utilize two publicly available datasets (MVTec [11], BTAD [12]) to generate a product-agnostic dataset.

3.1.1. Datasets

MVTec Anomaly Detection Dataset (MVTec): The MVTec Anomaly Detection dataset is a comprehensive multi-object dataset providing pixel-precise ground truth segmentation maps [11]. The MVTec dataset comprises about 5.3k high-resolution color images with 15 product types. Five products represent different types of textures and the remaining ten products are objects. Some objects can be rigid (bottle, metal nut) while some (cable) can be deformable. For each product, the training set comprises defect-free images and the test set comprises of both anomalous and non-anomalous images. For the anomalous images, pixel-precise ground truth segmentation maps are provided.

BeanTech Anomaly Detection Dataset (BTAD): BTAD is a public dataset consisting of about 1.8k high resolution images of three industrial products. BTAD has the same setting as MVTec [12]. Similar to MVTec, the training set comprises non-anomalous images and the test set includes both non-anomalous and anomalous images.

3.1.2. Product-Agnostic Annotations

Background Categories: Analyzing based on background categories helps us understand how the nature of imaging conditions can influence vision-based systems in detecting defects. For example, when product images have background, localizing defects along the edge of products can be challenging. In industrial pipelines, while encountering different types of products, image acquisition processes along with product size may decide what type of image will be used as input to the automated inspection system. This type of image-type categorization helps us to analyse from that perspective while designing the framework. Product-wise examples for the image categories are shown in Fig. 2. Table 2 demonstrates number of defective samples for each boundary category.

Defect Categories: As we mentioned earlier, getting insights on a broader level is crucial to enable implementation of auto-

mated inspection systems across pipelines dealing with different defects. Therefore, it is important to understand the efficacy over all product types instead of being product specific. We annotate each anomalous image from MVTec and BTAD datasets into four defect categories - structural, surface, contamination and combined. For each anomalous sample, 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. Table 1 shows number of defective samples for each defect category across products.

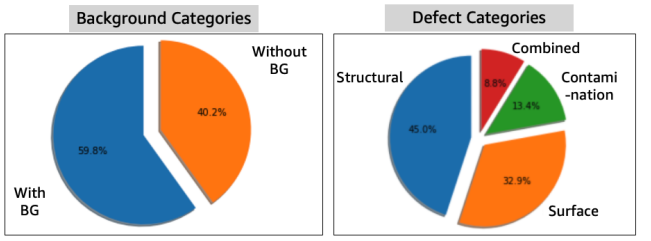

Figure 4: Pie charts showing overall statistics of the product-agnostic categories.

Pie charts in Fig. 4 depict overall numerical proportions of the product-agnostic categories. We observe that number of samples annotated with structural defects is the most followed by surface defects. Product images having external background are more in proportion than product images without background.

3.2. Representative Anomaly Localization Methods

We choose to evaluate a few representative models based on a broad categorization of methods previously described in Section 2.2. Based on the reported state-of-the-art performance of existing works, we select a representative approach from each category, namely, 1) Autoencoder (AE) [6], 2) Knowledge Distillation (KD) [7], 3) Patch Distribution Model (PaDiM) [8], and 4) Conditional normalization Flow (CFLOW) [9]. Next, we provide brief explanation of the selected modeling approaches.

3.2.1. Autoencoder (AE)

Autoencoders when implemented with a structural similarity metric (SSIM) loss function [34, 35] can improve anomaly localization performance [6]. In this work, autoencoder modeling approach is based on [6] utilizing SSIM as the loss.

Training: To train the autoencoder for a particular product type, the dataset is first augmented by generating 10,000 defect-free patches of size 128 x 128. Following the training procedure of [6], we train the model so that it can learn to reconstruct an anomaly-free image as closely as possible.

Inference: During inference with an anomalous sample, the autoencoder model (Fig. 5) still reconstructs a normal version

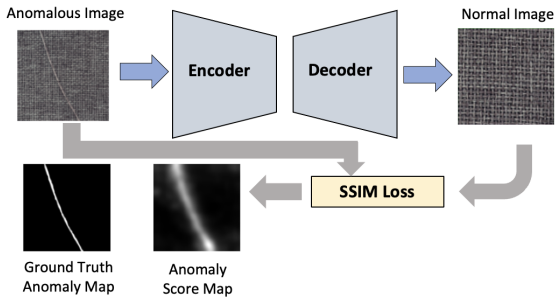

Figure 5: Autoencoder Modeling Approach.

of that. With a sliding window approach, SSIM is computed between the test image and its reconstruction at each pixel location to compute the anomaly score map highlighting the anomalous regions.

3.2.2. Knowledge Distillation (KD)

In Knowledge Distillation (KD) approach [7], based on ideas from attribution methods, the output loss, an estimation of anomaly in the image, is attributed to the input pixels.

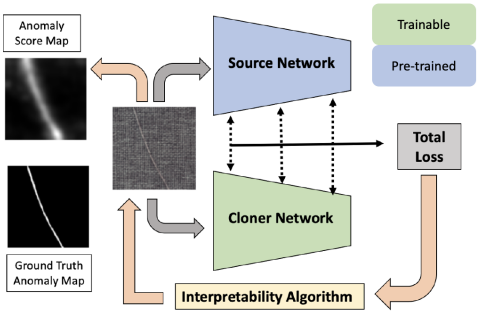

Figure 6: Knowledge Distillation Modeling Approach.

Training: The training [7] is done using a student-teacher approach with a source network and a cloner network as shown in Fig. 6. The source network, pre-trained on Image Net, is used to educate the cloner network in various abstraction levels. It is assumed that the source network has knowledge about both anomalous and non-anomalous images. But, the cloner network, trained on normal samples can only learn about normal images.

Inference: During inference with an anomalous sample, the cloner network cannot recognize it as it has been trained on normal images only. The discrepancy between the expert and cloner networks' intermediate activations is used to localize the anomalies. The anomaly score map is computed by taking gradients of the total loss to find anomalous pixels causing an increase in its value.

3.2.3. Patch Distribution Modeling (PaDiM)

Patch Distribution Modeling (PaDiM) [8] gets a probabilistic representation of the normal class using multivariate Gaussian

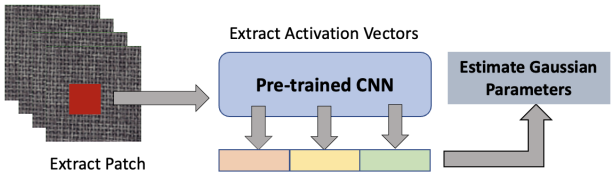

Figure 7: PaDiM Modeling Approach.

distributions and correlations between different semantic levels of a pretrained CNN.

Training: For training PaDiM, each image is segregated into multiple patches. For each patch position, activations are extracted from different abstraction levels of a pre-trained network. After implementing random dimensionality reduction, Gaussian parameters are learned from the training samples (Fig. 7).

Inference: During inference, a test image is divided into multiple patches and for each patch, activation values are extracted using the pre-trained CNN. Mahalanobis distance is computed between the embedding and the learned distribution to compute pixel-wise anomaly scores.

3.2.4. Conditional Normalizing Flow (CFLOW)

The conditional normalizing flow anomaly detector (CFLOW-AD) models the distribution of normal images during training by leveraging a pre-trained encoder and the conditional normalizing flow (CFLOW) decoder architecture [9]. The encoder is pre-trained with a large image dataset for classification (e.g., ImageNet [36]) and used as a feature extractor [37]. To maximize the ability of the pre-trained encoder with the various receptive field sizes, the authors adopted the multi-scale feature pyramid pooling approach to pool multiple feature vectors representing input image's characteristics at different scales [38].

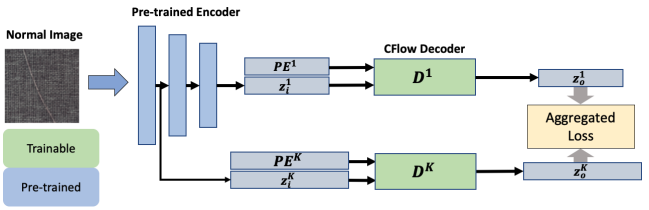

Figure 8: CFlow-AD Modeling Approach.

Training: During training, the CFLOW decoder (Fig. 8) models the distribution of normal images via the distribution of latent vectors of the decoder with the feature vectors that were pooled at multiple scales as input. To incorporate the spatial information of the feature vectors, the decoder comprised the concatenation of the latent vectors and the conditional positional vector.

Inference: At the inference, the CFLOW decoder takes the pooled feature vectors by the multi-scale feature pyramid pooling from the pre-trained encoder as inputs. The probability maps are estimated at each scale. The probability maps are rescaled to an input image size by bilinear interpolation and aggregated to estimate the final anomaly score map.

3.3. Threshold Estimation

A threshold to estimate a binary anomaly segmentation label map from a gray scale anomaly score map is essential in the real-world industry environment. However, it is often overlooked in many anomaly localization papers and not properly evaluated as the validation (threshold-independent) metrics are often considered to be good enough to compare the performance of the model [10, 9]. While anomaly score map allots an anomaly score for each pixel, evaluation should also be done using a segmentation map which masks the non-anomalous regions. The process of generating this mask given an anomaly score for each pixel has not been investigated much in existing works. Some existing works either apply a predefined threshold [39] or derive one threshold from the test data [40, 9]. While former is not an optimal way to determine threshold, for the latter, a validation set should be used as the test data should be unseen. Some works have determined threshold using a particular approach without doing an ablation study of using some other methods. Selecting an optimal threshold after optimizing the right evaluation metric can improve the performance significantly. Our proposed benchmark will provide a detailed analysis on threshold-dependent (inference) metrics comparing four models with five different threshold estimation techniques. To determine a threshold, for each product, we require a validation set comprising only a few number of anomalous samples. Next, we explain the validation set formation and the threshold estimation techniques.

3.3.1. Data Split into Validation and Test Sets

In our benchmark we consider the availability of a small amount of validation data that belongs to the same distribution as the test data. For the MVTec and BTAD dataset, we take out few examples from the test set to create the validation set. We use a split of 0.3 to generate the validation set from the test set of each product. Alongside evaluation, this validation dataset is leveraged to determine optimal threshold levels using different techniques.

3.3.2. Threshold Estimation Techniques

We propose to estimate the thresholds using four different techniques. Illustrations of threshold estimation for MVTec product bottle using model KD are shown in Fig. 9. Additionally, we also determine a baseline threshold using Otsu's method.

Otsu's Method: Otsu's method [41] is based on image histograms and segments regions by minimizing variance on each class (object, background). The main idea is to divide the image histogram into two clusters. In other words, the optimal threshold level is the score that divides the histogram into two parts ensuring that distributions of scores falling under the same class have minimum variance and distributions of scores falling under different classes have maximum variance. It is important

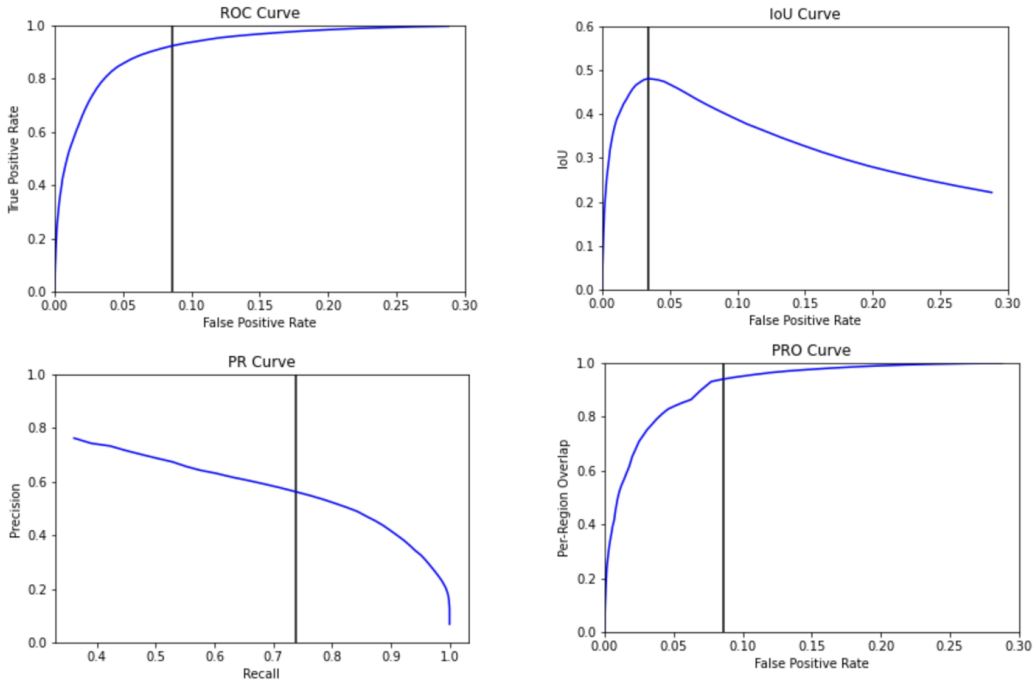

Figure 9: Illustrations of threshold estimation. In the ROC, IoU and PRO curves, the vertical lines show the FPR values corresponding to the optimal thresholds. In the PR curve, the vertical line highlights the recall value corresponding to the optimal threshold.

to note here that, Otsu's thresholding method does not require ground truth segmentation maps.

Using ROC Curve: To plot the Receiver Operating Characteristic (ROC) curve, different threshold values are used to compute the true positive rate (TPR) and false positive rate (FPR). FPR and TPR are plotted on the x-axis and y-axis respectively (Fig. 9). The optimal threshold finds the balance between FPR and TPR. 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.

Using IoU Curve: In the IoU curve, we plot the FPR on the x-axis and Intersection Over Union (IoU) on the y-axis for different threshold values as shown in Fig. 9. The optimal threshold finds the balance by maximizing IoU and simultaneously ensuring that FPR is not high. It is determined by maximizing the Geometric Mean of IoU and $1 - FPR$.

Using PR Curve As shown in Fig. 9, for the Precision Recall (PR) curve, we plot recall and precision on the x-axis and y-axis respectively. Different threshold values are used in ascending order to compute precision and recall values. To find a balance of precision and recall, the F1 Score (described earlier) can be maximized. For each threshold value, F1 scores are computed and the threshold corresponding to the maximum F1 score is the optimal one.

Using PRO Curve In the PRO curve (Fig. 9, FPR and per region overlap (PRO) are plotted on the x-axis and y-axis respectively. As explained before, PRO is a region level metric averaging TPR values across multiple connected components in the ground truth map. The optimal threshold value, with the maximum Geometric Mean of PRO and $1 - FPR$, seeks a balance between PRO and FPR.

4. Results

In this section, we present results from the validation and inference phase. To get the aggregated results across different products for the product-agnostic categories and for the subsequent analysis, we don't consider the product bottle from MVTec. We keep aside the product bottle to demonstrate later how this benchmarking framework can be leveraged for a new product.

4.1. Validation Phase

After training each model product-wise, the trained model and validation set of each product are utilized to compute the optimal thresholds and validation metrics. The optimal threshold values obtained using our proposed techniques are shown in supplementary material (Table 1 and Table 2 for MVTec and BTAD dataset respectively). We notice that, overall, Otsu's threshold values are lower than IoU thresholds across different models and products. This is an indication that the IoU threshold will have less chances than Otsu's threshold in overpredicting anomalous regions. Considering results across models, we find that the threshold values from the AE model are in general higher than the other models. Threshold-determination is an important step in our benchmarking framework and in later sections, we dive deeper to find definite answers regarding the advantages of threshold determination techniques.

During the validation phase, apart from computing optimal thresholds from four different curves (ROC, IoU, PR, PRO),

Table 3: Validation Set performance for product-agnostic defect categories.

Category	Model	AU-ROC	AU-IoU	AU-PR	AU-PRO
Structural	AE	0.436	0.085	0.195	0.432
	KD	0.758	0.179	0.422	0.768
	PaDiM	**0.826**	**0.200**	**0.568**	0.813
	CFLOW	0.825	0.155	0.494	**0.814**
Surface	AE	0.465	0.062	0.143	0.461
	KD	0.801	0.153	0.407	0.826
	PaDiM	0.842	**0.175**	**0.537**	0.839
	CFLOW	**0.884**	0.149	0.528	**0.876**
Contami-nation	AE	0.176	0.029	0.106	0.175
	KD	0.742	0.163	0.428	0.748
	PaDiM	0.858	**0.219**	0.584	0.859
	CFLOW	**0.922**	0.195	**0.602**	**0.918**
Combined	AE	0.436	0.128	0.227	0.409
	KD	0.681	0.227	0.409	0.696
	PaDiM	**0.754**	**0.250**	**0.518**	**0.712**
	CFLOW	0.714	0.189	0.450	0.644

Table 4: Validation Set performance for product-agnostic background categories.

Category	Model	AU-ROC	AU-IoU	AU-PR	AU-PRO
With Background	AE	0.475	0.073	0.149	0.476
	KD	0.805	0.154	0.364	0.813
	PaDiM	0.861	**0.174**	**0.541**	0.852
	CFLOW	**0.873**	0.138	0.491	**0.863**
Without Background	AE	0.327	0.083	0.207	0.312
	KD	0.695	0.205	0.487	0.717
	PaDiM	0.776	**0.238**	**0.574**	0.761
	CFLOW	**0.793**	0.196	0.540	**0.769**

we can also get an estimate of the model performance using the validation curves. Table 3 shows the validation set performance for the product-agnostic defect categories in terms of AU-ROC, AU-IoU, AU-PR and AU-PRO. We notice that in terms of AU-ROC and AU-PR, PaDiM and CFLOW outperform other models. When we use AU-IoU as the metric, PaDiM demonstrates better performance for all four defect categories. We find that CFLOW comparatively performs better when evaluated using AU-PRO. Overall, AE fails to achieve comparable performance for all the defect categories. Interestingly, we observe that, KD performs better than CFLOW for three categories when AU-IoU is used as the metric. Similar observations can be made from the validation set performance of the product-agnostic background categories, shown in Table 4. In terms of AU-ROC and AU-PRO, CFLOW slightly outperforms PaDiM. PaDiM has higher AU-IoU scores for both background categories followed by KD. Therefore, there remains this open question regarding the sub-optimal performance of CFLOW in terms of AU-IoU when it is showing better results with other metrics. This motivates us to understand the right choice of evaluation metrics and leads us back to what we have discussed before - that validation or threshold-independent metrics may not be the correct metrics for evaluation. Evaluation should also be done on the predicted segmentation maps using inference or threshold-dependent metrics to draw clear conclusions.

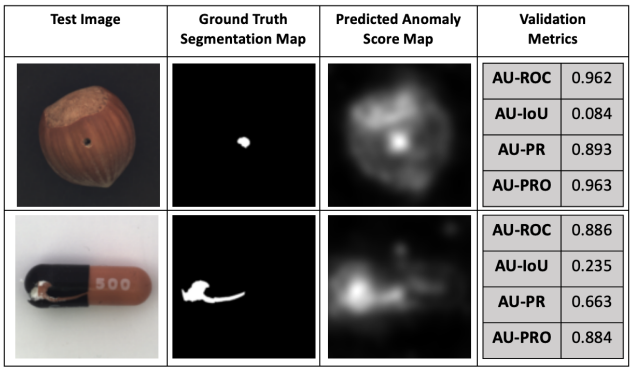

Figure 10: Anomaly score map results for two validation images from MVTec dataset. The top row and bottom row show anomalous images of hazelnut and capsule respectively. Though the bottom row demonstrates comparatively better localization performance, we observe that the values of AU-ROC, AU-PR and AU-PRO are lower for the bottom row. Only the value of AU-IoU is higher which seems to be coherent with the relative performance.

Fig. 10 shows anomaly score map results for two sample images from validation sets of hazelnut and capsule. From visual comparison, the predicted anomaly score map for capsule image is localizing anomalies better. In spite of a comparatively better localization performance, the AU-ROC, AU-PR and AU-PRO scores are lower for the bottom row (capsule) compared to the top row (hazelnut). AU-IoU seems to be a more reliable metric here. But, it is also quite clear from Fig. 10 that without the masking of non-anomalous pixels by using a threshold, it is quite difficult to localize the anomalies accurately with low false positives. Comparing performance based on segmentation maps using inference metrics is therefore important.

4.2. Inference Phase

Next, we analyse the product-agnostic performance of the different models in terms of inference metrics. Fig. 11 shows results from two test images from which we observe that though the PRO value is 1.0 for both, segmentation performance varies significantly which is correctly captured with changing IoU scores. This proves that PRO is not the correct metric for evaluation. In another way of analysis, from Fig. 12, it is illustrated that even though the F1 score remains roughly the same and the FPR deteriorates slightly, the segmentation performance for the bottom row is actually better - correctly manifested by IoU scores. This confirms that IoU is a more dependable inference metric for anomaly localization. We utilize IoU as the metric for performance comparison in Table 5 and Table 6.

Product types can vary across datasets - so it is important to summarize the performance in terms of product-agnostic categories. As mentioned in previous sections, we use two types of categories - defect and background. Table 5 shows the comparison of four methods in terms of the IoU metric using different threshold values. From Table 5, we observe that PaDiM demonstrates better overall performance than the other models. When the defects are structural, PaDiM (IoU scores of 0.307

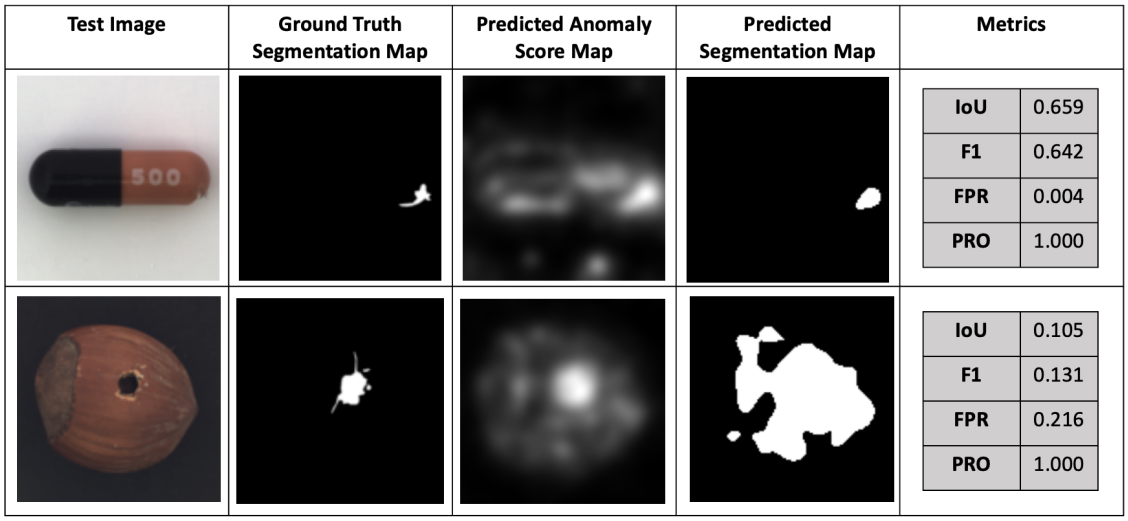

Figure 11: Segmentation map results for two test images from the MVTec dataset with contrasting performances in terms of IoU. The top row demonstrates an anomalous image from capsule where the segmentation map is computed from the score map using the IoU threshold. The bottom row shows an anomalous image from the hazelnut dataset where the segmentation map is computed using Otsu's threshold. It is interesting to note here that for both samples, PRO score is 1.0 comparing the predicted and ground truth map. The other scores, especially IoU varies a lot depending on the performance. This shows why PRO is not a right metric to evaluate segmentation maps.

Table 5: Test set performance in terms of IoU for product-agnostic defect categories. The highest IoU value for each category is highlighted.

Category	Model	Otsu's Thres	ROC Thres	IoU Thres	PR Thres	PRO Thres
Structural	AE	0.092	0.091	0.118	0.116	0.088
	KD	0.165	0.192	0.240	0.236	0.201
	PaDiM	0.195	0.234	**0.307**	0.284	0.232
	CFLOW	0.082	0.205	0.288	0.217	0.203
Surface	AE	0.059	0.059	0.086	0.082	0.056
	KD	0.087	0.135	0.206	0.210	0.149
	PaDiM	0.114	0.138	0.221	0.195	0.137
	CFLOW	0.094	0.210	**0.297**	0.242	0.202
Contamination	AE	0.062	0.057	0.071	0.077	0.057
	KD	0.189	0.198	0.281	0.280	0.217
	PaDiM	0.217	0.230	**0.325**	0.280	0.248
	CFLOW	0.116	0.225	0.282	0.222	0.233
Combined	AE	0.147	0.148	0.181	0.171	0.146
	KD	0.181	0.238	0.257	0.234	0.250
	PaDiM	0.224	0.263	**0.294**	0.281	0.265
	CFLOW	0.066	0.158	0.257	0.149	0.162

and 0.284) is a better choice than CFLOW (IoU = 0.288) or KD (IoU = 0.240). For surface defects, CFLOW (IoU scores of 0.297 and 0.242) outperforms PaDiM (IoU = 0.221) and KD (IoU = 0.206). For contamination defects, we notice that PaDiM demonstrates better performance (IoU = 0.325) and the performance of CFLOW and KD are almost similar (with best IoU scores of 0.282 and 0.281 respectively). Similar observations can be made for combined defects - the highest IoU score from both KD and CFLOW is 0.257. The best performance for combined defects is from PaDiM with an IoU score of 0.294. While AE mostly fails to detect anomalous regions, it shows considerable performance (IoU = 0.181) only for combined defects.

Table 6: Test set performance in terms of IoU for product-agnostic background categories. The highest IoU value for each category is highlighted.

Category	Model	Otsu's Thres	ROC Thres	IoU Thres	PR Thres	PRO Thres
With Background	AE	0.071	0.073	0.098	0.102	0.069
	KD	0.131	0.177	0.236	0.237	0.185
	PaDiM	0.148	0.195	**0.286**	0.272	0.192
	CFLOW	0.058	0.172	**0.286**	0.203	0.169
Without Background	AE	0.105	0.101	0.128	0.118	0.100
	KD	0.156	0.175	0.230	0.224	0.192
	PaDiM	0.206	0.212	0.268	0.224	0.220
	CFLOW	0.146	0.271	**0.293**	0.254	0.270

The performances for product-agnostic background categories are shown in Table 6. For images with background, both PaDiM and CFLOW demonstrate highest IoU score of 0.286. The best IoU score from KD is 0.236. When the product images don't have any external background, CFLOW (IoU = 0.293) performs better in comparison to PaDiM (IoU = 0.268) and KD (IoU = 0.230). Considering the threshold determination techniques, from Table 5 and Table 6, we find that estimating threshold from IoU curve is the most accurate approach followed by threshold from PR curve.

5. Discussion

Product-Agnostic Categorization: One of the key contributions of this work is to go beyond product-specific evaluation and analysis to enable a more generalized way of comparing the model performance. While products can vary in industrial

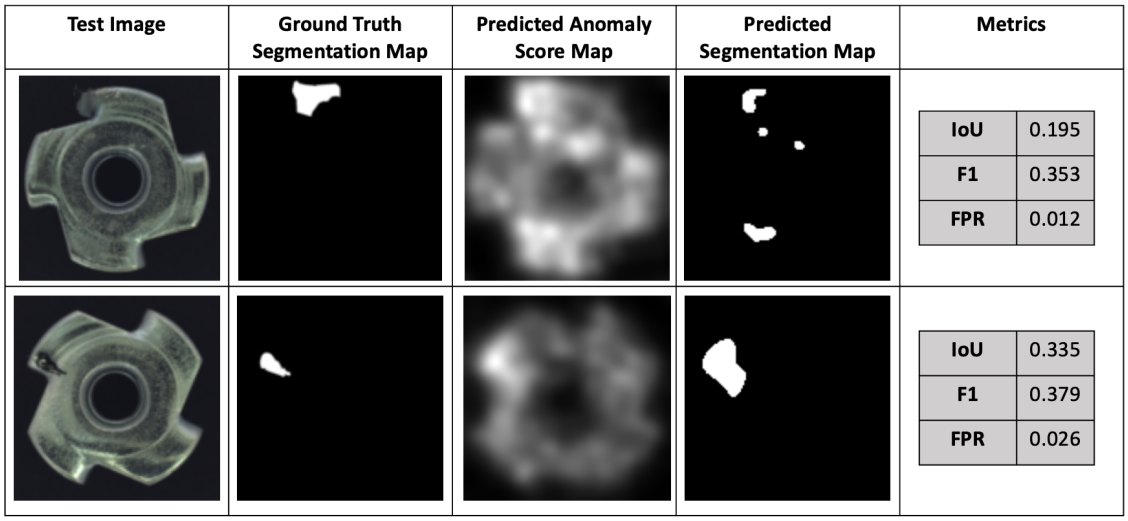

Figure 12: Segmentation map results for two test images from MVTec product Metal Nut with contrasting performances in terms of IoU. The IoU scores for the top and bottom row are 0.195 and 0.335 respectively. Though the F1 values are very close for both rows and FPR rather increases for the bottom row, IoU value correctly indicates that for the bottom row the performance is actually better. This shows that IoU is a more reliable metric.

pipelines across different sectors, defects generally will be seen to fall among some common categories. In developing this benchmarking framework, we consider two types of product-agnostic categorization - background and defect. Categorizing in terms of defect is done with a detailed labeling effort - a team of annotators worked on the task to generate image-wise labels. We release these labels as part of the supplementary materials. These re-labeled datasets can be very useful for practitioners to think along the same lines and researchers can develop new benchmarks in a product-agnostic way.

Threshold Estimation: Threshold estimation is another aspect we focus majorly in this work to make an anomaly localization system ready for practical use. This has been overlooked by previous works and a detailed ablation study was required. Utilizing five different threshold determination techniques, we perform an ablation study considering all the evaluation metrics. We use a few defective samples as the validation set for the threshold estimation step. From the ablation study, we observe that using the IoU curve is the most efficient threshold determination approach followed by the threshold from PR curve. Estimating an optimal threshold is an extremely important step in building a localization pipeline for real-world use. The insights gained from this work can be very useful to find the optimal threshold when deploying an automated visual inspection system in industrial pipelines.

Evaluation: We focus on how to correctly evaluate this type of framework with different types of metrics. This has also been mostly overlooked by previous work and a common understanding was required regarding the correct way of evaluation. While we utilize both validation (threshold-independent) and inference (threshold-dependent) metrics, we highlight that evaluating with inference metrics is extremely crucial as it involves predicted segmentation maps. In previous sections, we analyze with examples to highlight the disadvantages of validation metrics. We also demonstrate with examples that IoU is a more reliable inference metric to estimate segmentation performance and that some other inference metrics like PRO is not suitable. The development of this benchmarking framework brings more clarity in the evaluation process which can contribute immensely towards enabling automated anomaly localization.

Choice of Modeling Approach: We investigate modeling approaches from four different broad categories. From experiments, we find that PaDiM demonstrates better performance in most cases. CFLOW and KD also perform well in predicting segmentation maps correctly. Comparatively AE fails to perform well in most experiments with some rare exceptions. The benchmarking framework highlights the optimal model choice for different defect and background categories. We recommend that using PaDiM and estimating threshold from the IoU curve is the most efficient way to start with for a manufacturer coming with a new product. If computational resource is a constraint, we recommend using KD which shows faster inference speeds in our experiments. If surface defects are more likely to appear in the product, CFLOW can be preferred over PaDiM.

Benchmarking a New Product: As we mentioned earlier, we kept aside one product (bottle) to demonstrate how a manufacturer can benchmark a new product. The illustration of the benchmarking process with this hold-out product is shown in Fig. 13. We have 209 defect-free images and 63 anomalous product images. The first step is to categorize the anomalous images as per the product-agnostic categorization. Out of 63 images, 41 images demonstrate structural defects and 21 images fall under contamination category. With this proportion of defects, we can observe from Table 5 that PaDiM will be the appropriate modeling approach. The preferred threshold determination approach should be the threshold from the IoU curve. Also, the product images have external background and fall under 'with background' category (Table 6). The PaDiM model is trained using the normal (defect-free) images. Thereafter,

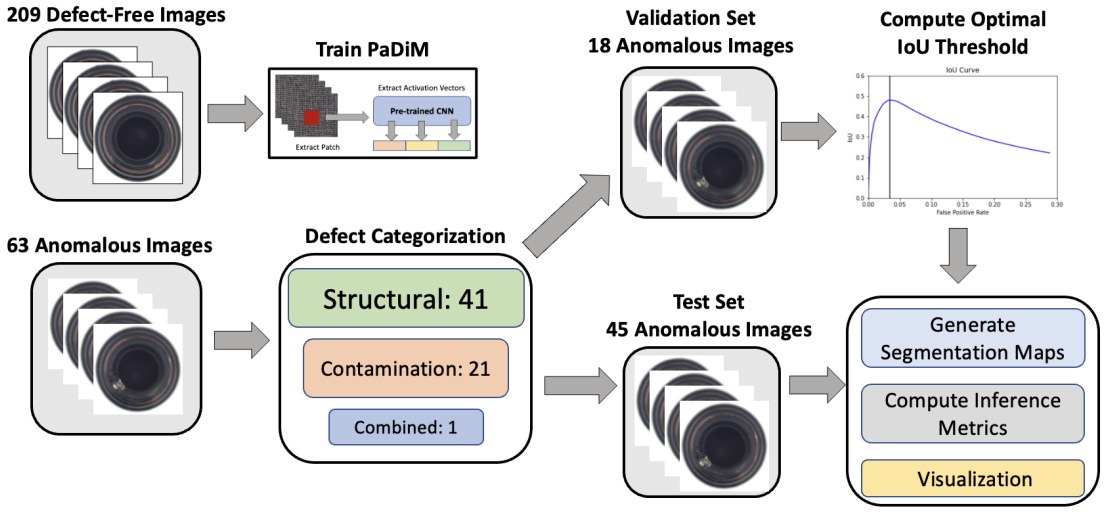

Figure 13: Illustration of the process of benchmarking a new product. We use the product bottle from MVTec for this illustration. The choices of modeling approach (PaDiM) and threshold determination technique (IoU Thres) are based on the learnt insights from our proposed benchmarking framework.

the set of anomalous images is split into validation and test set. The validation set is utilized to estimate the optimal threshold from the IoU curve. After threshold estimation, in the inference phase, we generate segmentation maps using the trained model along with the threshold value. This is followed by evaluation using IoU and visualization of some segmentation maps to confirm the results from a domain perspective.

6. Conclusion

We develop a benchmarking framework focusing on making anomaly localization systems ready for practical use. We generate a new product-agnostic dataset with annotations of anomalous product images capturing higher level features. We perform experiments with four different modeling approaches along with an ablation study of different threshold estimation techniques. We dive deeper to understand the correct way of evaluation for the localization task. The insights gained from developing this framework can help practitioners in deploying automated anomaly localization in industrial pipelines. Accurate detection of defects will eliminate production of defective products and improve the efficiency of manufacturers in delivering highest-quality products to their customers. This work will also encourage other researchers to perform new experiments in similar directions. In future, we plan to include other datasets to build on the existing defect categories by adding new ones. Another future research direction can be to focus on the localization performance for rare defects in industrial pipelines.

Supplementary Material

See the supplementary material for additional results.

Authors' Contributions

T.G., L.L.C., S.R. and Y.S. conceptualized this machine learning work. T.G. built the pipelines and generated results for AE, KD and PaDiM models. S.H. built the pipeline and generated results for CFLOW model. T.G. built the evaluation pipeline common for all models. T.G. prepared the figures except the CFLOW figure which S.H. prepared. T.G., S.H., and S.R. interpreted the results and performed the analysis. T.G., S.H. and S.R. wrote the manuscript with feedback from L.L.C. and Y.S. All authors have contributed in reviewing the manuscript.

Acknowledgements

The authors acknowledge the ML Solutions Lab labeling team for their great effort in providing the image-wise annotations for different MVTec and BTAD products. The authors especially would like to thank Abishek Karthikeyan, Nandini Krishnan and Arunram for driving the labeling process.

References

[1] Z. Ren, F. Fang, N. Yan, 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, S. Tang, Using deep learning to detect defects in manufacturing A comprehensive survey and current challenges, materials (2020).

[3] P. Burlina, N. Joshi, I. Wang, et al., Where's wally now? deep generative and discriminative embeddings for novelty detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11507–11516.

[4] Z. Jin, Z. Zhang, J. Ott, G. X. Gu, Precise localization and semantic segmentation detection of printing conditions in fused filament fabrication technologies using machine learning, Additive Manufacturing 37 (2021) 101696.

[5] C.-C. Tsai, T.-H. Wu, S.-H. 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, 2022, pp. 3992–4000.

[6] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, C. Steger, Improving unsupervised defect segmentation by applying structural similarity to autoencoders, arXiv preprint arXiv:1807.02011 (2018).

[7] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, H. R. Rabiee, Multiresolution knowledge distillation for anomaly detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 14902–14912.

[8] T. Defard, A. Setkov, A. Loesch, R. Audigier, Padim: a patch distribution modeling framework for anomaly detection and localization, in: International Conference on Pattern Recognition, Springer, 2021, pp. 475–489.

[9] D. Gudovskiy, S. Ishizaka, K. 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, 2022, pp. 98–107.

[10] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection, International Journal of Computer Vision 129 (2021) 1038–1059.

[11] P. Bergmann, M. Fauser, D. Sattlegger, C. 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, 2019, pp. 9592–9600.

[12] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, G. L. Foresti, Vt-adl: A vision transformer network for image anomaly detection and localization, in: 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), IEEE, 2021, pp. 01–06.

[13] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Asian conference on computer vision, Springer, 2018, pp. 622–637.

[14] P. Perera, R. Nallapati, B. 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, 2019, pp. 2898–2906.

[15] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-anogan: Fast unsupervised anomaly detection with generative adversarial networks, Medical image analysis 54 (2019) 30–44.

[16] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[17] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, O. Camps, Towards visually explaining variational autoencoders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8642–8651.

[18] C. Baur, B. Wiestler, S. Albarqouni, N. Navab, Deep autoencoding models for unsupervised anomaly segmentation in brain mr images, in: International MICCAI brainlesion workshop, Springer, 2018, pp. 161–169.

[19] D. Zimmerer, J. Petersen, S. A. Kohl, K. H. Maier-Hein, A case for the score: Identifying image anomalies using variational autoencoder gradients, arXiv preprint arXiv:1912.00003 (2019).

[20] D. Dehaene, O. Frigo, S. Combrexelle, P. Eline, Iterative energy-based projection on a normal data manifold for anomaly localization, arXiv preprint arXiv:2002.03734 (2020).

[21] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: International conference on machine learning, PMLR, 2017, pp. 3319–3328.

[22] K. Zhang, B. Wang, C.-C. J. Kuo, Pedenet: Image anomaly localization via patch embedding and density estimation, Pattern Recognition Letters 153 (2022) 144–150.

[23] S. Lee, S. Lee, B. C. Song, Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization, arXiv preprint arXiv:2206.04325 (2022).

[24] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, P. Gehler, Towards total recall in industrial anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14318–14328.

[25] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International conference on machine learning, PMLR, 2015, pp. 1530–1538.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[27] M. Rudolph, B. Wandt, B. 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, 2021, pp. 1907–1916.

[28] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (2006) 861–874.

[29] T. Calders, S. Jaroszewicz, Efficient auc optimization for classification, in: European conference on principles of data mining and knowledge discovery, Springer, 2007, pp. 42–53.

[30] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd international conference on Machine learning, 2006, pp. 233–240.

[31] P. Napoletano, F. Piccoli, R. Schettini, Anomaly detection in nanofibrous materials by cnn-based self-similarity, Sensors 18 (2018) 209.

[32] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4183–4192.

[33] J.-C. Wu, D.-J. Chen, C.-S. Fuh, T.-L. Liu, Learning unsupervised metaformer for anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4369–4378.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE transactions on image processing 13 (2004) 600–612.

[35] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss functions for neural networks for image processing, arXiv preprint arXiv:1511.08861 (2015).

[36] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).

[37] R. Schirrmeister, Y. Zhou, T. Ball, D. Zhang, Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features, Advances in Neural Information Processing Systems 33 (2020) 21038–21049.

[38] J. Masci, U. Meier, G. Fricout, J. Schmidhuber, Multi-scale pyramidal pooling network for generic steel defect classification, in: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, 2013, pp. 1–8.

[39] S. Venkataramanan, K.-C. Peng, R. V. Singh, A. Mahalanobis, Attention guided anomaly localization in images, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII, Springer, 2020, pp. 485–503.

[40] J. Song, K. Kong, Y.-I. Park, S.-G. Kim, S.-J. Kang, Anoseg: anomaly segmentation network using self-supervised learning, arXiv preprint arXiv:2110.03396 (2021).

[41] N. Otsu, A threshold selection method from gray-level histograms, IEEE transactions on systems, man, and cybernetics 9 (1979) 62–66.