AUPIMO: Redefining Anomaly Localization Benchmarks with High Speed and Low Tolerance

Joao P. C. Bertoldo[1]
jpcbertoldo@minesparis.psl.eu
Dick Ameln[2]
dick.ameln@intel.com
Ashwin Vaidya[2]
ashwin.vaidya@intel.com
Samet Akçay[2]
samet.akcay@intel.com

[1] Mines Paris, PSL University, Centre for mathematical morphology (CMM), 77300 Fontainebleau, France
[2] Intel

Abstract

Recent advances in anomaly localization research have seen AUROC and AUPRO scores on public benchmark datasets like MVTec and VisA converge towards perfect recall. However, high AUROC and AUPRO scores do not always reflect qualitative performance, which limits the validity of these metrics. We argue that the lack of an adequate and domain-specific metric restrains progression of the field, and we revisit the evaluation procedure in anomaly localization. In response, we propose the Area Under the Per-IMage Overlap (AUPIMO) as a recall metric that introduces two major distinctions. First, it employs a validation scheme based solely on normal images, which avoids biasing the evaluation towards known anomalies. Second, recall scores are assigned *per image*, which is fast to compute and enables more comprehensive analyses (*e.g.* cross-image performance variance and statistical tests). Our experiments (27 datasets, 8 models) show that the stricter task imposed by AUPIMO redefines anomaly localization benchmarks: current algorithms are not suitable for all datasets, problem-specific model choice is advisable, and MVTec AD and VisA have *not* been near-solved. Available on GitHub[1].

1 Introduction

Anomaly Detection (AD) is a machine learning task based on *normal* patterns, meaning they are not of special interest at inference time. As such, the model must identify deviations from the patterns observed in the training set, *i.e. anomalies*. Within this domain, Visual Anomaly Detection focuses on image or video-related applications, including both the detection of anomalies in images (answering the question, "Does this image contain an anomalous structure?") and the more precise task of anomaly localization or segmentation, where the goal is

[1] github.com/jpcbertoldo/aupimo

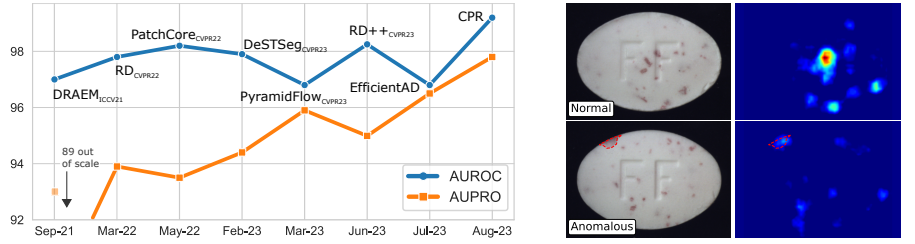


Figure 1: Left: performance on MVTec AD over time, approaching a near 100% performance plateau. Right: images from the dataset Pill (left column) and their inferred anomaly maps (right column; higher values mean anomalous; JET colormap) from the best performing model in this dataset (EfficientAD; see Appendix D), with 98.7% AUROC and 96.7% AUPRO. The normal image (top) has higher anomaly scores than the anomaly (bottom).

to determine if specific pixels belong to an anomaly. Our emphasis is on anomaly localization in image applications (other modalities are out of the scope of this paper, but extensions of our work are possible and briefly discussed in Sec. 6).

Anomaly localization research has achieved significant progress, partly thanks to the increased availability of suitable datasets [4, 5, 12, 17, 28]. In particular, MVTec Anomaly Detection (MVTec AD) [4] and Visual Anomaly (VisA) [28] comprise (together) 27 datasets (22 object and 5 texture-oriented) with high-resolution images and pixel-level annotations.

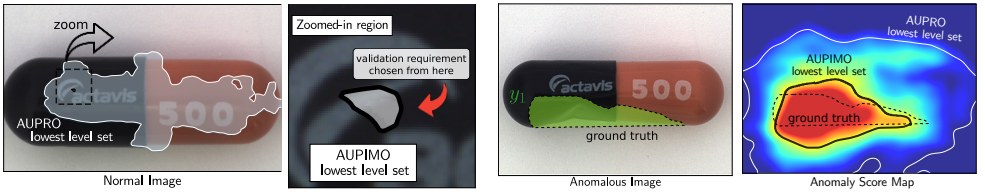
AUROC [10] and AUPRO [5] – respectively, the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) and Per-Region Overlap (PRO) curves (see Sec. 3.1) – have been used to evaluate anomaly localization, but it has been observed that the extreme class imbalance at pixel level inflates the scores produced by these metrics[2] [19, 23]. As a result, the performance numbers on MVTec AD and VisA reported in the literature are converging towards 100% (Fig. 1, left), giving the impression that these datasets have been solved. Meanwhile, even the top performing models often fail to localize anomalous regions in some of the more challenging samples from these datasets while raising many False Positives (FPs) (*i.e.* a normal pattern wrongly flagged as anomalous in Fig. 1, right).

We argue that the anomaly localization literature urges a metric well-suited to its unique characteristic: the positive (anomalous) class is unknown beforehand and may have an unlimited number of modes. While anomalous samples (even of different types) are available in public datasets, the goal of an AD model is to detect *any* type of anomaly. Our work emphasizes on this unsupervised nature of the problem to build a performance metric that does *not* depend on anomalies available at hand to avoid a bias towards known anomalies.

In response, we present the **Area Under the Per-Image Overlap (AUPIMO)** curve (Sec. 3.2). It relies on a clear separation of normal and anomalous images for, respectively, validation and evaluation of models – thus avoiding class imbalance-related issues. Its strict validation requirement sets a more challenging task in-line with the latest advances in the field. Our work provides means to comprehensively compare models with image-specific evaluation scores and, along with the standard procedure proposed in Sec. 4, tackles cross-paper comparison issues. In summary, our work presents the following contributions:

1. A validation-evaluation framework based on strict low tolerance for FPs on normal images only, which avoids conditioning the model behavior on known anomalies, thus

[2]The term "metric" is used as a synonym for "performance measure" in this paper. It does *not* refer to the mathematical concept of distance in a metric space.



(a) AUPIMO's integration bound is chosen so false positive regions in normal images are small. Zoomed-in region: the lowest (*i.e.* largest) level set seen by AUPIMO in a normal image is insignificant compared to the structure of the image (more examples in Appendix A). AUPRO's equivalent is larger as it is chosen to yield recall-achievable results (*i.e.* based on the anomalies).

(b) Left: anomalous image and its ground truth annotation mask (green region means anomalous). Right: anomaly map (JET colormap; blue/red means lower/higher anomaly score). The upper bound level sets are the lowest level sets seen by each metric. Their areas under the curve (AUCs) correspond to the average recall of the level sets above them (*i.e.* inside these contours).

Figure 2: AUPRO and AUPIMO's upper bounds visualized as level sets from the anomaly score maps. Solid contours are level sets at thresholds yielding the maximum FPR in AUPRO (white) and AUPIMO (black). Images from the dataset MVTec AD/ Capsule.

providing a recall measure consistent with AD's unsupervised nature (Sec. 3.3);

2. Per-image recall scoring, enabling the analysis of cross-image performance variance and high-speed execution at high resolution both on CPU and GPU (Sec. 5).
3. Empirical evidence suggesting that MVTec AD and VisA datasets have *not* been near-solved and that problem-specific model choice is advisable (Sec. 5).

2 Related Work

AUROC is a threshold-independent metric for binary classifiers [10], and it is widely used to assess anomaly localization, treating it as a pixel-level binary classification. However, it has recently been argued that, in real-world applications, full or partial localization of anomalous regions is more relevant than pixel accuracy [4, 27]. Furthermore, it has been shown that AUROC is not suitable for anomaly localization datasets due to the extreme class imbalance [19, 23], prompting the exploration of other evaluation metrics in the field [4, 19, 27].

Bergmann et al. [4] proposed a ROC-inspired curve called Per-Region Overlap (PRO). At each binarization threshold, it measures the region-scoped recall averaged across all anomalous regions available in the test set. Notably, AUPRO excludes thresholds yielding False Positive Rate (FPR) values above 30% in the computation of the area under the PRO curve to force the metric to operate over a range of meaningful thresholds.

Recent studies have proposed metrics that index the thresholds based on recall instead of FPR. Rafiei et al. [19] observed that the high pixel-level class imbalance in MVTec AD and similar anomaly localization datasets challenges the effectiveness of AUROC and AUPRO for model comparison. They concluded that the area under the Precision-Recall (PR) curve is a more suitable metric for AD as it is conditioned on the positive class (anomalous). Alternatively, other authors [11, 28] have used the F_1-max score, which is the best achievable F_1 (harmonic mean of recall and precision), implying an anomaly score threshold choice. Zhang et al. [27] proposed the Instance Average Precision (IAP), a modified version of the PR curve where recall is defined at the region-level, counting a region as detected if at least

Table 1: Notation.

Symbol	Description	Symbol	Description
M, j	Number and index of pixels in an image	$\mathbf{a} \in \mathbb{R}_+^M$	Anomaly score map
$\neg, \wedge$	Pointwise logical negation/AND	$\mathbf{y} \in \{0,1\}^M$	Ground truth (GT) mask
$\lvert\cdot\rvert$	Cardinality of a set or number of 1s in a mask	$\mathbf{r} \in \{0,1\}^M$	Region mask
$\mathbf{a} \geq t$	Binarization of $\mathbf{a}$ by t	$t \in \mathbb{R}_+$	Threshold
L, U	Integration lower/upper bounds	$\mathcal{A}$, $\mathcal{Y}$, $\mathcal{R}$	Sets of $\mathbf{a}$, $\mathbf{y}$, and $\mathbf{r}$

half of its pixels are correctly detected. This alternative recall metric is further used as a validation requirement (threshold choice) and the pixel-level precision is used to compare models (precision-at-k%-recall).

AUPIMO uses a validation criterium based only on normal images to avoid a bias towards detectable anomalies. As detailed in Sec. 3, we advocate in favor of normal-only validation to build an evaluation score in line with AD's unsupervised nature, while using recall only to rate models. Finally, AUPIMO uses image-scoped metrics, preserving the structured information from the images and making its computation significantly faster (Fig. 5a).

3 Metrics

We define a framework to compare AUROC and AUPRO (Sec. 3.1), introduce our new metric (Sec. 3.2), and discuss its properties (Sec. 3.3). Key notation is listed in Tab. 1.

Our goal is to compare a model's output $\mathbf{a}$ (an anomaly score map; higher means more likely to be anomalous) with its ground truth mask $\mathbf{y}$ (0 and 1 labels indicate "normal" and "anomalous" respectively), illustrated in Fig. 2b. We define $\mathbf{r}$ as a region in $\mathbf{y}$ such that instances do not overlap (maximally connected components). All metrics are *pixel-wise* (one score/annotation per pixel), not *image-wise* (one score/annotation per image) since our focus is to measure whether a model can detect anomalous structures *within an image*. We define the False Positive Rate (FPR) and True Positive Rate (TPR), *i.e.* recall, across three scopes: **set** (all pixels in all images confounded; subscript s), **per-image** (all pixels in an image; subscript i), and **per-region** (pixels in a single anomalous region; subscript r):

$$\mathrm{F_s} : t \mapsto \frac{\sum_{\mathbf{y}\in\mathcal{Y}} \lvert(\mathbf{a} \geq t) \wedge (\neg\mathbf{y})\rvert}{\sum_{\mathbf{y}\in\mathcal{Y}} \lvert\neg\mathbf{y}\rvert} \qquad \mathrm{T_s} : t \mapsto \frac{\sum_{\mathbf{y}\in\mathcal{Y}} \lvert(\mathbf{a} \geq t) \wedge \mathbf{y}\rvert}{\sum_{\mathbf{y}\in\mathcal{Y}} \lvert\mathbf{y}\rvert} \tag{1}$$

$$\mathrm{F_i} : t \mapsto \lvert(\mathbf{a} \geq t) \wedge (\neg\mathbf{y})\rvert \,/\, \lvert\neg\mathbf{y}\rvert \qquad \mathrm{T_i} : t \mapsto \lvert(\mathbf{a} \geq t) \wedge \mathbf{y}\rvert \,/\, \lvert\mathbf{y}\rvert \tag{2}$$

$$\mathrm{T_r} : t \mapsto \lvert(\mathbf{a} \geq t) \wedge \mathbf{r}\rvert \,/\, \lvert\mathbf{r}\rvert \quad . \tag{3}$$

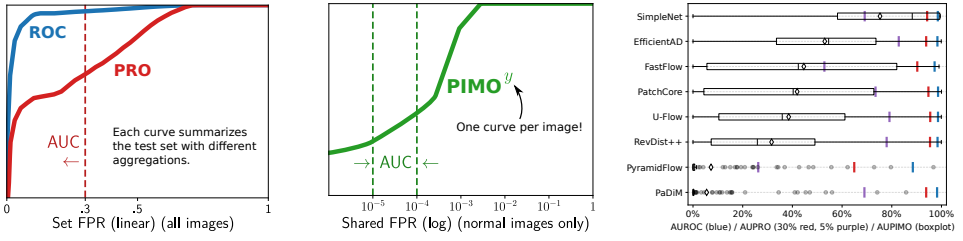
Instances at each scope ($\mathbf{r}$, $\mathbf{y}$, and $\mathbf{a}$) are ommited in the notation for brevity.

3.1 Precursors: AUROC and AUPRO

The ROC and PRO curves (Fig. 3a) can be defined as

$$\mathrm{ROC} : t \mapsto \left(\mathrm{F_s}(t)\,,\, \mathrm{T_s}(t)\right) \quad \text{and} \quad \mathrm{PRO} : t \mapsto \left(\mathrm{F_s}(t)\,,\, \overline{\mathrm{T_r}}(t)\right) \quad , \tag{4}$$

where $\overline{\mathrm{T_r}} : t \mapsto \frac{1}{\lvert\mathcal{R}\rvert} \sum_{\mathbf{r}\in\mathcal{R}} \mathrm{T_r^r}(t)$ is the average Region TPR; $\mathrm{T_r^r}$ refers to the $\mathrm{T_r}$ applied to the instance $\mathbf{r}$ and $\mathcal{R}$ is the set of all $\mathbf{r}$ from all $\mathbf{y} \in \mathcal{Y}$. Both curves trace the trade-off between



(a) ROC and PRO curves (b) PIMO curve (c) MVTec AD / Zipper

Figure 3: (a, b) ROC, PRO, and PIMO curves. The y-axes are TPR metrics: ROC uses the set TPR (all anomalous pixels from all images confounded); PRO uses the region-scoped TPR averaged across all regions from all images; PIMO uses the image-scoped TPR keeping one curve per anomalous image (no cross-instance averaging). The x-axes are FPR metrics shared by all instances (*i.e.* anom. regions for PRO and anom. images for PIMO), which indexes the binarization thresholds. ROC and PRO use the set FPR (all normal pixels from all images confounded) in linear scale. PIMO uses the image-scoped FPR averaged accross normal images only in log scale. The curves are summarized by their (normalized) area under the curve (AUC), with different integration ranges: AUROC in $[0,1]$, AUPRO in $[0,0.3]$[3], and AUPIMO in $[10^{-5},10^{-4}]$. (c) Benchmark on dataset MVTec AD / Zipper shows how their AUCs differ.

False Positives (FPs) and True Positive (TP)s across all potential binarization thresholds. Both use the Set FPR as the x-axis, but different recall measures as the y-axis, reflecting distinct Region TPR aggregation strategies. PRO calculates the arithmetic average (equal weight to each region). ROC uses the Set TPR, which is equivalent to averaging the Region TPRs with region size weighting. Their respective AUCs, AUROC and AUPRO, summarize the curves into a single score:

$$\text{AUROC} = \int_0^1 \mathrm{T_s}\left(\mathrm{F_s^{-1}}(z)\right) \mathrm{d}z \quad \text{and} \quad \text{AUPRO} = \frac{1}{U}\int_0^U \overline{\mathrm{T_r}}\left(\mathrm{F_s^{-1}}(z)\right) \mathrm{d}z\,, \tag{5}$$

where $\mathrm{F_s^{-1}}$ is the inverse of $\mathrm{F_s}$. In practice, they are computed using the trapezoidal rule with discrete curves given by a sequence of anomaly score thresholds.

AUPRO is restricted to thresholds such that $\mathrm{F_s}(t) \in [0,U]$ (*i.e.* to the left of the vertical line in Fig. 3a), where U is the upper bound FPR. This means that AUPRO only accounts for recall values obtained from level sets higher than (*i.e.* inside) the white level set in the anomaly score map in Fig. 2. The default value of $U = 30\%$[3] is based on the intuition that at such FPR levels the segmentation contours of the anomalies are no longer meaningful [5], so that should be the "worst case". From this perspective, the FPR restriction in AUPRO acts as a model validation – an implicit requirement since a partial threshold choice is imposed.

3.2 Our Approach: AUPIMO

PRO measures region-scoped recall at each binarization threshold, which are indexed by an FPR metric (the x-axis) shared by all region instances. We generalize this idea and employ the term *Shared* FPR ($\mathrm{F_{sh}}$) to refer to "*any* FP measure shared by all anomalous instances."

[3] We also considered a AUPRO with $U = 5\%$ (noted $\text{AUPRO}_{5\%}$) in our experiments for the sake of making the metric more challenging.

In our approach, the Set FPR used as x-axis by ROC and PRO is replaced by the average Image FPR on normal images only: $\mathrm{F_{sh}} : t \mapsto \frac{1}{|\mathcal{Y}^0|} \sum_{\mathbf{y} \in \mathcal{Y}^0} \mathrm{F_i^y}(t)$, where $\mathcal{Y}^0 \subset \mathcal{Y}$ contains only and all normal images in $\mathcal{Y}$, and $\mathrm{F_i^y}$ refers to $\mathrm{F_i}$ computed on instance $\mathbf{y}$. This design choice is a major counterpoint with previous approaches, and its implications are discussed in Sec. 3.3. The **Per-Image Overlap (PIMO)** curve (Fig. 3b) and its AUC are defined as

$$\mathrm{PIMO^y} : t \mapsto \left(\log\left(\mathrm{F_{sh}}(t)\right), \mathrm{T_i}(t)\right) \quad \text{and} \quad \mathrm{AUPIMO^y} = \int_{\log(L)}^{\log(U)} \frac{\mathrm{T_i}\left(\mathrm{F_{sh}^{-1}}(z)\right)}{\log\left(U/L\right)} \mathrm{d}\log(z) \quad , \quad (6)$$

where the integration bounds have default values $L = 10^{-5}$ and $U = 10^{-4}$. To have a better resolution at low FPR levels, the x-axis is in log-scale, and the term $1/\log(U/L)$ normalizes the integral's score to $[0, 1]$. Contrasting with AUROC and AUPRO, which define a single score for the entire test set, we keep one score per image (superscript $^{\mathbf{y}}$).

3.3 AUPIMO's properties

AUPIMO significantly diverges from its predecessors by: (1) considering only normal instances for validation and using a stricter requirement (integration range in the x-axis), (2) evaluating metrics at the image scope, and (3) calculating individual scores for each image. This section discusses the implications and advantages of these design choices.

Bias-free validation AUROC is a threshold-independent metric, which limits its usage in real-world applications that require threshold selection for inference. AUPRO addresses this by imposing an FPR restriction, which selects a range of valid thresholds, thus carrying an implicit model validation based on the Set FPR. AUPIMO uses a similar strategy, but – to produce a bias-free score – we propose that the validation metric (x-axis of the curve) should only use normal images, while anomalous images are only used for evaluation.

AD is often viewed as a binary classification problem, yet this simplification is misleading. While the normal class is well-defined by the training set, the anomalous class is, by definition, unknown, unbounded, thus inherently multi-modal. Public datasets (*e.g.* MVTec AD and VisA) provide various types of anomalies, but the objective in AD is to detect *any* type of anomaly. As the positive class in AD can have an unlimited number of modes, we argue that an evaluation metric in benchmarks should avoid conditioning the model behavior (*i.e.* creating a bias, *e.g.* selecting a threshold range) based on *known* anomalies.

The x-axis in AUPIMO ($\mathrm{F_{sh}}$) is built only from normal images, which can be reasonably assumed from the same distribution as the training set. In this framework, the variance of the normal class coming from acquisition conditions, sensor noise, *etc.* is accounted for in the validation metric ($\mathrm{F_{sh}}$). By ensuring that these variations are not falsely detected, the model's capacity to detect anomalies is isolated from the normal class's variability. This essential change avoids biasing the evaluation metric towards available anomalies, which is consistent with the unsupervised nature of AD. Note that an alternative AUPRO could be defined in the same way, but AUPIMO carries additional advantages discussed below.

Anomaly-dependent metrics The Area Under the Precision-Recall (AUPR) and its variant Instance Average Precision (IAP) [27] use recall measures on the x-axis and precision on the y-axis. Similar to the AUCs defined in Sec. 3.1 and Sec. 3.2, they express the average of the y-axis over a range of thresholds, which are indexed by the x-axis. Using the recall as x-axis biases the metric in favor of detectable anomalies, making the metric sensitive to the

distribution of known anomalies. The threshold at the integration lower bound is the maximum full-recall threshold, making them sensitive to hard anomalies[4] – while not revealing them. Conversely, easy anomalies can be over-represented because low-recall thresholds are coverered – *i.e.* unnecessarily high thresholds are accounted for.

The F_1-max score and IAP further choose, respectively, optimal and minimum thresholds based on the recall. Similarly, AUPRO validates models using anomalous images as well because it restricts the Set FPR (Eq. (1)), which encompasses all test images (thus the normal-annotated pixels in anomalous images). While such threshold choices are useful for practical applications, we argue that benchmarks should prefer bias-free metrics so that model comparison is more consistent across different datasets and applications.

Finally, AUPIMO's validation is insentive to imprecisions in the anomaly annotations – *i.e.* when only loose bounding box annotations are available. Other model conditioning criteria – as in F_1-max and IAP in particular – carry pixel-level imprecision but AUPIMO is not affected because normal images are only annotated at the image level.

Low tolerance From an application perspective, anomalies are expected to contain information deserving the user's attention. A high FPR can lead to user frustration and diminish trust in the model. To tighten evaluation, we restrict the FPR range in AUPIMO to be between 10^{-5} and 10^{-4} for datasets like MVTec AD and VisA. At such levels, the FP regions in normal images are small compared to the structures seen in the images (see Fig. 2 and Appendix A). An AUPIMO score can be interpreted as the "*average segmentation recall in an anomalous image given that the model (nearly) does not yield FP regions in normal images*". These default values were chosen to establish a challenging task in-line with recent advances in research, but they can be adapted to application-specific needs.

AUPRO vs. AUPIMO Fig. 2 shows a visual comparison between AUPRO and AUPIMO. The upper bound in AUPRO is chosen from a precionsion-inspired criterion ("beyond that point the anomaly segmentations are no longer useful"), so the FP regions on normal images can be large. In contrast, AUPIMO chooses a more conservative upper bound. The model conditioning in AUPIMO ensures that FP regions in normal images are insignificant. As a result, its recall on the anomalous region (on the right in Fig. 2b) is lower than AUPRO's – which is expected.

Image-scoped metrics Note that the set-scoped metrics in AUROC and AUPRO are ill-suited for images because information within each image is disregarded (all pixels are confounded). AUPIMO avoids this problem by only using image-scoped metrics (*i.e.* ratios of pixels within each image). Image-scoped measures account for image structure, are fast to compute (Fig. 5a), and are robust to noisy annotations (see Fig. 5b).

Image-specific scores Since each curve/score refers to an image file, it is easy to index scores to instances[5]. Achieving the same with region-based scores would require more metadata, and finding connected regions is implementation-sensitive. For instance, Anomalib's [1] CPU and GPU-based implementations are from `opencv-python` [7] and `kornia` [21], and the AUPRO scores slightly differ. Per-image scores enable fine-grained analyses otherwise impossible with AUROC and AUPRO. Score distributions (*e.g.* Fig. 3c) – instead of single-valued scores – provide insight into performance variance, which we exploit to select representative samples for qualitative analysis in Appendix D. Finally, it also enables

[4] Reminder: lower threshold means higher recall, so the anomalies with lowest anomaly score are the hardest.

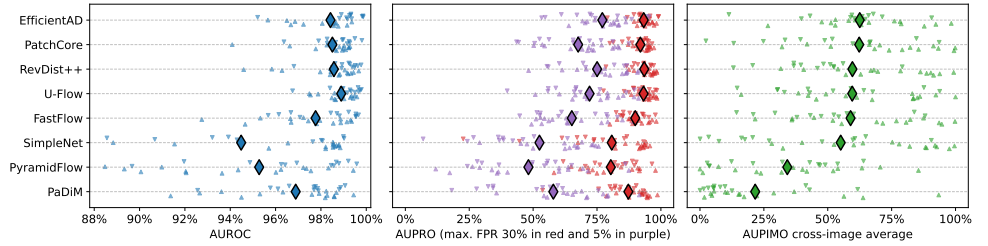


Figure 4: Dataset-wise comparison. Each triangle is a set-scoped score (AUROC, AUPRO, and $\text{AUPRO}_{5\%}$) or a cross-image statistic (average AUPIMO) from a dataset in MVTec AD ($\triangle$) or VisA ($\triangledown$). Diamonds are cross-dataset averages (all confounded). Plots have different x-axis scales. AUPIMO reveals that all models have a large cross-problem variance, meaning that none of the models is robust to all problems.

the use of statistical tests, which we showcase in an ablation study in Appendix C.1.

4 Experimental Setup

We benchmark the datasets from MVTec AD and VisA with State-of-the-Art (SOTA) models to compare the performances reported in terms of AUROC, AUPRO, and AUPIMO. We also report AUPRO with $U = 5\%$ ($\text{AUPRO}_{5\%}$) for the sake of comparing with a more challenging alternative of that metric.

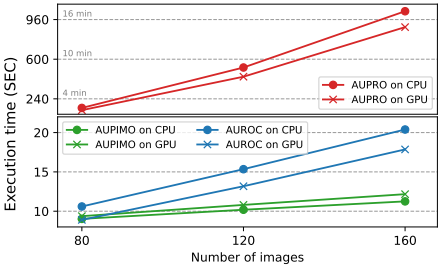
We reproduce a selection of models: PaDiM [8] from ICPR 2021, PatchCore [22] from CVPR 2022, SimpleNet [15], PyramidFlow[6] [14], and RevDist++ [25] from CVPR 2023, along with the recently published models UFlow [24], FastFlow [26], and EfficientAD [2]. Our aim is to ensure a comprehensive evaluation with a set of different algorithm families. This selection includes methods based on memory bank (PatchCore), reconstruction (SimpleNet), student-teacher framework (RevDist++, EfficientAD), probability density modelling (PaDiM), and normalizing flows (FastFlow, PyramidFlow, UFlow).

All models were trained with 256×256 images (downsampled with bilinear interpolation, no center crop), and with the hyperparameters reported in the original papers. We used the official implementations or Anomalib [1]. The implementations of AUROC and AUPRO are from Anomalib [1]. Details provided in Appendix D.

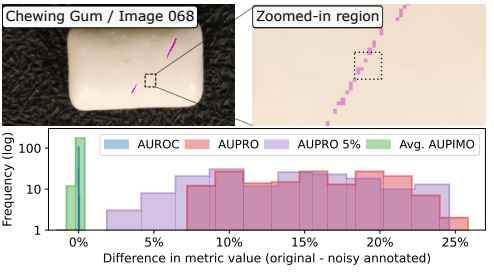
Cross-paper comparisons in the anomaly localization literature often have conflicting evaluation procedures. We aim to tackle this issue by proposing our evaluation guidelines as a standard: (1) compute test set metrics at the annotations' full resolution with bilinear interpolation for resizing the anomaly score maps if necessary; (2) do *not* apply crop to the input images; (3) publish per-image scores[5]; (4) (ideally) report the score distribution (*e.g.* boxplots as in Fig. 3c). Details in Appendix D.

[5] A standard format is proposed in Appendix D and implemented in our repository.

[6] Our AUPRO results significantly differ from PyramidFlow's paper. Their implementation has higher scores because it does not apply the maximum FPR (30%) as proposed by [5] https://github.com/gasharper/PyramidFlow (commit `6977d5a`), see function `compute_pro_score_fast` in the file `util.py`.



(a) Execution time



(b) Robustness

Figure 5: (a) Execution time of metrics on MVTec AD / Screw dataset (image resolution of 1024×1024; average times over 3 runs). (b, top) An anomalous sample from the dataset VisA / Chewing Gum superimposed with its annotation (pink) shows meaningless, tiny (even 1-pixel) regions (the mask has *not* been downsampled). (b, bottom) Robustness to noisy annotation. Histograms show the distribution of the difference between the scores without and with the synthetic mistakes (closer to zero is better).

5 Results

In this section we comment on the results of a single dataset (Fig. 3c), present a summary across all datasets (Fig. 4), and compare AUROC, AUPRO, and AUPIMO in terms of the execution time and robustness to noisy annotation. Due to the space constraints, additional results are available in Appendix C and the benchmarks from all datasets in MVTec AD and VisA are documented in Appendix D.

Benchmark on MVTec AD / Zipper Fig. 3c illustrates two common observations in our benchmarks. First, it shows how AUROC and AUPRO fail to reveal differences between models (*e.g.* differences of 0.1% and 0.4% between the two best models). While $\text{AUPRO}_{5\%}$ amplifies the differences, AUPIMO's strict validation causes the best model to stand out more clearly. Note that $\text{AUPRO}_{5\%}$ and AUPIMO show different rankings, which might be attributed to how they weight small anomalies differently. Second, image-specific performance often has large variance and the best models have left-skewed AUPIMO distributions – *c.f.* the best models per dataset in Appendix D.3. In Fig. 3c for example, several models have worst and best-case samples at 0% and 100% AUPIMO respectively. Fortunately, AUPIMO provides the means to investigate this by programatically identifying specific instances or anomaly types not well-detected by a model.

Cross-dataset analysis Fig. 4 reveals two key insights regarding the SOTA in anomaly localization. First, the benchmark datasets from MVTec AD and VisA still have room for improvement. While $\text{AUPRO}_{5\%}$'s (purple) stricter validation is more challenging, AUPIMO (green) reveals that even the best models have failure cases when constrained to low FP tolerance. We argue that setting such a challenging standard will push the next generation of models to achieve a more trustworthy task: high anomaly recall with near-zero false positives. Second, none of the models consistently achieves reasonable performance across all datasets. For example, despite PatchCore's high performance in many problems, it performs poorly on VisA / Macaroni 2 (details in Appendix D.3). Meanwhile, EfficientAD has a reasonable performance on this dataset, thus the dataset is not unsolvable with the current models. This provides a useful insight for practitioners: problem-specific model choice is

highly advised because a model's failure in one dataset does not imply failure in another one.

Execution time Having computationally efficient metrics is essential to enable fast iterations and not create computational bottlenecks in research and development. Fig. 5a shows that AUROC and AUPIMO have comparable execution time, but AUPRO is significantly slower both on CPU and GPU. The main reason is that AUPRO requires connected component analysis, while AUROC and AUPIMO do not. AUPIMO's implementation relies on simple operations, enabling the use of `numba` [43] to further accelerate the computation (reported execution times include the just-in-time compilation). The GPU used was an NVIDIA GeForce RTX 3090 and the CPU was an Intel Core i9-10980XE. Note that the chosen model does not influence the execution time because the anomaly score maps are precomputed.

Robustness In real-world use-cases, high-quality annotation is hard to acquire or even to define. Fig. 5b shows an example of a ground truth mask where noisy regions can be seen. We found this issue to be prevalent in VisA (more examples in Appendix B). In the PRO curve, these tiny regions have the same weight as the actual anomalous regions. In contrast, AUPIMO is more robust to this issue due to their limited contribution to the overall image score. Fig. 5b demonstrates this in an experiment with artificially added noise. Random mistakes mimicking statistics from VisA are added to the datasets in MVTec AD. We generate one noisy mask for each anomalous mask by adding randomly shaped anomalous regions to it. The number and size of the noisy regions are randomly sampled with probabilities matching the statistics of the VisA dataset (average frequencies from Tab. 2 in Appendix B).

6 Conclusion

We introduced AUPIMO: a novel recall metric tailored for anomaly localization addressing the limitations of its predecessors (AUROC and AUPRO) and formalizing a validation-evaluation framework. As a guiding principle, it was proposed that the validation step should only depend on normal images to avoid biasing the model behaviour towards known anomalies, thus making the metric consistent with the unsupervised nature of AD. Finally, a stringent false positive restriction is proposed to establish a more challenging task on contemporary benchmark datasets and expose differences between models.

AUPIMO is built with image-scoped metrics and enables simple assignment of image-specific scores. As demonstrated, these design choices offer advantages in terms of computational efficiency (see Fig. 5a), fine-grained performance analysis (see Fig. 3c and Appendix C.1), and resilience against noisy annotation (see Fig. 5b).

Evaluating eight recent models on 27 datasets with AUPIMO revealed a significant insights about the SOTA in anomaly localization. We show evidence that problem-specific model selection is highly advised, raising further questions for future research. Namely, can one identify dataset traits causing a model to succeed or fail? Or conversely, which model features should one look for to succeed on a specific problem?

Limitations In this paper we focused on (2D) image anomaly localization, but AUPIMO can be easily adapted to 3D imaging (*e.g.* X-ray tomography), 3D point clouds (*e.g.* LiDAR), and video-based applications (a proof of concept is shown in Appendix C.3). Other domains like times series would require more careful adaptation, which is left for future work. As a recall metric, the notion of segmentation quality is not covered by AUPIMO, but Appendix C.4 briefly discusses alternatives based on the same validation-evaluation principle.

7 Acknowledgements

This research has been conducted during Google Summer of Code 2023[7] (GSoC 2023) at OpenVINO (Intel). We would like to thank the OpenVINO team for their support and feedback during the project. We would like to thank Matías Tailanian for having collaborated by training the UFlow models and providing the evaluation results for the benchmark.

References

[1] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. Anomalib: A Deep Learning Library for Anomaly Detection. In *ICIP*, pages 1706–1710, 2022.

[2] Kilian Batzner, Lars Heckler, and Rebecca König. EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies, 2023.

[3] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should We Really Use Post-Hoc Tests Based on Mean-Ranks? *Journal of Machine Learning Research*, 17(5):1–10, 2016.

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *CVPR*, pages 9592–9600, 2019.

[5] 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. *IJCV*, 129(4):1038–1059, 2021.

[6] Jakob Božič, Domen Tabernik, and Danijel Skočaj. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Computers in Industry*, 129:103459, 2021.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. In *ICPR*, pages 475–489, 2021.

[9] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[10] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8): 861–874, 2006.

[11] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In *CVPR*, pages 19606–19616, 2023.

[7] https://summerofcode.withgoogle.com/archive/2023/projects/SPMopugd.

[12] Renato A. Krohling, Guilherme J. M. Esgario, and José A. Ventura. BRACOL - A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests, 2019.

[13] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: a LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

[14] Jiarui Lei, Xiaobo Hu, Yue Wang, and Dong Liu. PyramidFlow: High-Resolution Defect Contrastive Localization Using Pyramid Normalizing Flow. In *CVPR*, pages 14143–14152, 2023.

[15] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. SimpleNet: A Simple Network for Image Anomaly Detection and Localization. In *CVPR*, pages 20402–20411, 2023.

[16] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, pages 1975–1981, 2010.

[17] 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, 2021.

[18] Mantini Pranav, Li Zhenggang, and Shah Shishir K. A day on campus - an anomaly detection dataset for events in a single camera. In *ACCV*, 2020.

[19] Mehdi Rafiei, Toby P. Breckon, and Alexandros Iosifidis. On Pixel-level Performance Assessment in Anomaly Detection, 2023. URL http://arxiv.org/abs/2310.16435.

[20] Bharathkumar Ramachandra and Michael J. Jones. Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2558–2567, 2020.

[21] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3674–3683, 2020.

[22] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards Total Recall in Industrial Anomaly Detection. In *CVPR*, pages 14318–14328, 2022.

[23] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3):e0118432, 2015.

[24] Matías Tailanian, Álvaro Pardo, and Pablo Musé. U-Flow: A U-shaped Normalizing Flow for Anomaly Detection with Unsupervised Threshold, 2023. URL http://arxiv.org/abs/2211.12353.

[25] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan T. M. Duong, Chanh D. Tr Nguyen, and Steven Q. H. Truong. Revisiting Reverse Distillation for Anomaly Detection. In *CVPR*, pages 24511–24520, 2023.

[26] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows, 2021. URL http://arxiv.org/abs/2111.07677.

[27] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *CVPR*, pages 3914–3923, 2023.

[28] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, pages 392–408, 2022.